GPMS: A Generative Projection Mapping System

Figure i *Projection Concept* [1]

University of Central Florida
Department of Electrical and Computer Engineering

Senior Design I
Dr. Chan

Group B
Victoria Moreno - Computer Engineering
Declan Carter - Computer Engineering
Francisco Soriano - Computer Engineering

Committee
Reza Abdolvand, Saleem Sawahneh, George Atia

Table of Content

List of Figures

List of Tables

Chapter 1: Executive Summary

The project introduced by the authors was a Generative Projection Mapping System (*GPMS*) - an intriguing technology aimed at delivering projection mapping for everyday use. The authors desired to make immersive storytelling and entertainment accessible to hobbyists and entertainers alike. The GPMS initiative seeked to bridge the gap between high-end, professional projection mapping displays utilized by various entertainment giants and the consumer market. The motivation is fueled by the potential to enhance personal and communal experiences through the innovative use of generative AI and computer vision algorithms.

GPMS was designed as a portable system capable of transforming any 2D surface into a dynamic visual storytelling canvas. By accepting user inputs, GPMS leveraged generative AI to produce images that were then projected onto surfaces, aligning with their unique features. The device was engineered to recognize the structure's vertices, edges, and designs, which ensured that generated images perfectly complemented the physical space. GPMS combined a projector with a responsive touchscreen monitor for easy use along with advanced image processing algorithms for seamless image integration.

The primary goal of GPMS was to offer a user-friendly, customizable projection experience that allowed for a high degree of personalization in how images were matched and displayed against targeted structures. Key objectives included both ensuring that the projector accurately displayed images based on user prompts and that it allowed for precise alignment of generated images with the structure's specific features.

GPMS has broad applications that promise to revolutionize how stories are told and how experiences are shared among the general public. Furthermore, this system will make projection mapping an accessible tool for creativity. The project's ambition extended beyond its current scope, with plans to collaborate with local theme parks and potentially establish a startup, significantly impacting the entertainment industry and consumer technology.

GPMS aimed to set a new and unique standard in immersive visual storytelling. This system represents a significant step towards making advanced projection mapping accessible, which offers a blend of creativity, innovation, and personalization that will certainly enrich everyday experiences.

Chapter 2: Project Description

2.1 Motivation and Background

The authors' interest in projection mapping stemmed from their passion for immersive storytelling and entertainment. However, due to the artistry, time, and technical nuance which are often required to tailor-make projection mapping shows, such displays are typically beyond the grasp of everyday hobbyists or entertainers. While theme park giants like Disney and Universal continue to improve large, multi-projector systems to render grand scenes onto their existing architecture, more has to be done to put this technology into everyday consumers' hands. Individuals already look to purchase Sky Projectors to throw the night sky onto their ceiling, Christmas lights to celebrate the holidays, or disposable paper products to decorate their walls for festive occasions. However, through the combination of state-of-the-art computer vision algorithms and generative AI models, the authors looked to introduce a new and unique product to allow people to share the joy of any occasion.

The authors proposed a portable generative projection mapping system that can map AI generated images onto any unique surface to create a themed and immersive environment. This system used generative AI to create themes that precisely fit user input. This was done by GPMS aligning the major image features with the existing vertices on the surface. This process showed to be scalable using a central and distinct server to generate these images. As a result, the system can be employed relatively cheaply. The generative AI aspect of GPMS allowed users to let their audiences experience immersive storytelling.

The authors took inspiration from the historical presence of projection mapping in local entertainment venues. In 2011, Disney World, located in Orlando, Florida, began projecting images onto Cinderella's castle with the help of 3D scanning and projection mapping. The emerging projections ran from 2011 to 2012 and even displayed visitors' photos onto Cinderella's castle throughout select shows. In light of the increase in the versatility and stability of projection mapping, Disney World has continued incorporating more creative and new projection shows for years to come. [2]

Additionally, Universal Orlando has transformed its projection mapping techniques through two major breakthroughs within the last four years. The first breakthrough was the development of high-resolution projectors capable of beaming bright and crystal-clear images over long ranges. The second breakthrough was projection mapping which ensured that projected images would not be distorted when beamed onto uneven structures such as the Hogwarts Castle, one of Universal's more notable attractions. [3]

As entertainment venues increase their use of projection mapping, a unique business opportunity arose for the authors. More specifically, after the final demonstration, the authors plan to propose GPMS with a 'buy' option to local entertainment venues in Orlando, FL and they also plan to create their own startup from scratch.

Projection mapping, being relatively recent, has been refined to project images that can wrap themselves around various surface types for the images to look like they are part of the surfaces themselves. As projection mapping advancements continued to grow and expand, GPMS aimed to directly engage and provide storytellers in all environments with a fresh way of delivering messages through the use of generative AI.

2.2 Related Work

2.2.1 Dynamic Projection Mapping Research

A projection mapping solution presented by *Addison Sandvik* of the University of California Polytech State University outlined a similar approach to GPMS. This project demonstrated a proof-of-concept for a flexible projection mapping technique, which allowed for a surface to be placed anywhere within the projector's field of view for accurate image projection. This technique employed a surface equipped with infrared lights and a camera outfitted with an infrared filter to be able to identify the location of the surface. Then, this same technique utilized a projective transformation to convert points from the camera's perspective into corresponding points in the projector's output. [4]

Sandvik's work demonstrated a few basic implementations of computer vision aspects, such as 'lens distortion correction' using a pre-calculated matrix to flatten the rounded image and 'image calibration/alignment' by transforming the corner points of the projected image to where the camera was able to view them. Both of these implementations were essential aspects of the computer vision stack necessary for GPMS. Additionally, in a few key ways, the authors' implementation simplified a few critical aspects given that their projection surfaces were *static*, unlike the dynamic mapping proposed by Sandvik. So, select pieces of Sandvik's project were well adapted to suit the needs of any of the calibration issues that GPMS experienced.

2.2.2 Industry Products: Luma Box

Many products in the industry do projection mapping, and some are even AI-generated. The best existing example of a product like this is 'Luma Box'. Luma Box introduces an innovative approach to projection mapping through its product, Luma Map. This tool is designed to facilitate the creation of visually engaging projections that can be tailored to fit the unique contours of any surface, such as that of a house. The versatility of Luma Map lies in its ability to enhance physical features with light and shadows while integrating AI-generated themes.

Moreover, Luma Map presents a relevant case study for a project that employs AI to generate images for projection onto buildings, thereby augmenting their appearance and/or introducing thematic elements. It exemplifies how technology can merge seamlessly with generative AI to redefine spaces, offering insights into the potential applications and benefits of similar AI-driven projects in projection mapping. [5]

2.2.3 Stable Diffusion

‘Stable Diffusion’ is a state-of-the-art deep learning model developed by ‘Stability AI’ that has revolutionized the field of text-to-image generation. It is a latent diffusion model trained on a vast dataset of *text-image* pairs, enabling it to generate highly realistic and diverse images from textual descriptions. [6]

At its core, Stable Diffusion is based on a type of generative model called a ‘diffusion model’. Diffusion models work by learning to reverse a gradual noising process applied to training images. During the training phase, the model is shown images that are progressively degraded with noise, and it learns to recover the original clean images from the noisy versions. [^]

The key innovation of Stable Diffusion lies in its ability to combine this diffusion process with a powerful text encoder, such as the ‘Transformer’ architecture, which allows it to understand and translate textual descriptions into meaningful visual representations. The model learns to associate specific words and phrases with visual concepts, enabling it to generate images that accurately reflect the content and style described in the text prompt. [6]

Stable Diffusion employs a technique called ‘latent diffusion’, which operates in a compressed latent space rather than directly on pixel values. This approach enables the model to capture high-level semantic information and generate images with improved quality and efficiency compared to previous text-to-image models. [6]

The architecture of Stable Diffusion consists of an encoder, a series of diffusion steps, and a decoder. The encoder maps the input text into a latent representation, which is then iteratively refined through the diffusion process. At each step, the model predicts the noise that needs to be removed to reconstruct the original image. The decoder then maps the final latent representation back into the pixel space, generating the output image. [6]

One of the key advantages of Stable Diffusion is its ability to generate diverse and creative images from a given text prompt. By sampling from the learned distribution of the model, it can produce multiple plausible visual interpretations of the same textual description. This allows users to explore a wide range of creative possibilities and generate unique and compelling images. [6]

Stable Diffusion has found applications in various domains, including art, design, gaming, and visual storytelling. It has the potential to assist artists, designers, and content creators in generating novel and inspiring visual content, as well as enabling new forms of creative expression and exploration.

2.2.4 Stable Diffusion with ControlNet

While Stable Diffusion excels at generating images based on textual prompts, it can be challenging to accurately convey specific visual elements or spatial arrangements through words alone. To address this limitation and provide users with more control over the generated images, a group of researchers from Stanford University introduced *ControlNet* in 2023.

ControlNet is “a neural network architecture that enables the incorporation of spatially localized input conditions into a pre-trained text-to-image diffusion model, such as Stable Diffusion, through efficient fine-tuning.” (Zhang et al., 2023) By allowing users to provide an additional input image alongside the text prompt, ControlNet enhances the user's ability to guide the image generation process and achieve more precise results. For example, if a user wants to generate an image of a superhero in a specific pose, providing an input image of the desired pose would be more effective than attempting to describe the pose in words. ControlNet leverages this visual information to constrain and direct the image generation process, resulting in outputs that more closely align with the user's intentions/requests. [7]

A key aspect of ControlNet is its use of a hypernetwork architecture. Instead of retraining the entire Stable Diffusion model from scratch, ControlNet’s makers were adamant in reusing the model. ControlNet acts as a hypernetwork, a neural network that generates weights for another network. [8] This approach allows for efficient adaptation and specialization while preserving the knowledge and capabilities of the original model.

The diagram below, taken from the researchers' paper, illustrates the architecture of ControlNet and its integration with Stable Diffusion:

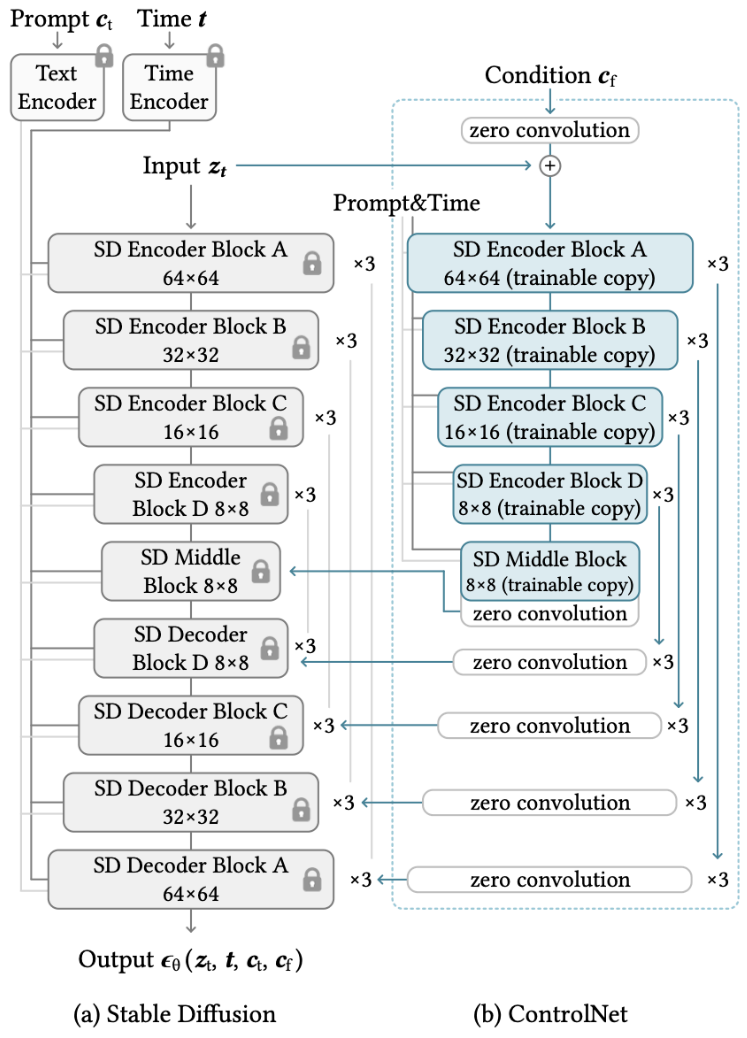


Figure 2.2.4 *Stable Diffusion with ControlNet*

In this architecture, ControlNet consists of a trainable copy of the encoders and the middle block from the Stable Diffusion model. The Stable Diffusion model itself remains locked, ensuring that its pre-trained weights are not altered during the fine-tuning process. ControlNet utilizes zero convolution layers in its initial stages to mitigate the impact of harmful noise during training. The weights generated by ControlNet's layers are then passed to the decoders of the Stable Diffusion model. The middle block of ControlNet also contributes weights to the corresponding middle block of Stable Diffusion, which are summed together before propagating to the subsequent decoder blocks. This iterative process of combining weights from ControlNet and Stable Diffusion continues through the decoder blocks until the final output image is generated. [7]

By selectively injecting learned weights into specific parts of the Stable Diffusion model, ControlNet enables more precise control over the image generation process while also leveraging the pre-trained model's knowledge and capabilities. The results presented in the researchers' report demonstrate the effectiveness of ControlNet in enhancing the controllability and flexibility of Stable Diffusion. By preserving the essential features of the input structure and incorporating them into the generated image, ControlNet opens up exciting possibilities for user-guided image generation across various domains, including art, design, and creative applications. This combination represents a significant advancement in the field of generative AI, empowering users with greater control and expressiveness in creating visually coherent and semantically meaningful images.

2.3 Project Overview

2.3.1 Basic Goals

The authors' project was centered around creating a system, GPMS, with the primary goal of constructing an immersive storytelling environment. The overarching objective was to dynamically generate and project images onto a designated structure in alignment with its unique features. These structures' features include, but are not limited to, pillars, windows, bricks, or even just a drawing. Therefore, the camera utilized within the system must be of sufficient quality to best capture the features. GPMS must be able to adeptly detect vertices, edges, and designs on the given target structure by intelligently leveraging sophisticated computer vision techniques.

GPMS integrated a responsive touchscreen monitor to facilitate seamless user input and display the respective generated AI image content. Users will be able to input text/prompts, which will then directly initiate the generation of images that will lastly be intelligently projected onto the unique structure/surface. The authors' cutting-edge system harnesses the power of generative AI to interpret user inputs, along with capabilities to craft images that align seamlessly with the unique structure's features.

For example, during the winter, one user may ask for a Christmas design dedicated for a house. As a result, the generated image could possibly include candy canes for the front pillars and snowflakes on the featureless areas. Alternatively, if the surface was a drawing of a dog, the output generated image could include that same dog with a Santa hat on its head.

GPMS required a high-powered projector for optimal visibility for users. Additionally, in order to calibrate images, users had to manually select the edges of the projected display each time it was used on a new structure. User calibration was achieved through the touch screen monitor by moving the image's corners to the proper areas on the structure with a magnifying tool to aid users in achieving maximum precision. The user will know if the calibration was set correctly because the image will have been projected on the structure as they were calibrating.

2.3.2 Advanced Goal

In the author's pursuit of elevating user customization, the overarching goal was to increase the control users have over image representation, which aimed for a significant level of precision. Their specific objective was the implementation of sensitivity and threshold controls to achieve this goal. These controls will have empowered users, allowing them to intricately adjust the outline of the structure according to their unique preferences. For example, if the structure was a brick wall, but the user did not want their projected image to be based on each brick's outline, they could turn the sensitivity down. However, if the structure is a drawing of someone's face, the user could increase the sensitivity, allowing for the inclusion of each and every specific feature on the face.

This transformative experience was facilitated through a user-friendly interface, where individuals could precisely determine the desired similarity between the generated image and the structure. The innovative approach proposed by the authors ensured a seamless and personalized customization experience, which set a new user engagement and satisfaction standard.

2.3.3 Stretch Goals

As part of the authors' stretch goals, they aimed to allow users to highlight specific structural features. Their primary stretch objective was to design an intuitive interface that allows users to effortlessly select and emphasize particular elements, ensuring a personalized visual experience. Users then would be able to achieve this by choosing features on the picture of the structure through the monitor. As a result, the output image generated would then be able to closely align with the selected features.

Their second goal was to streamline the user experience. More specifically, the objective was to integrate an automatic calibration process in the device, adapting to the features of walls or selected structures with minimal user input. Autocalibration ensures optimal image display without unnecessary complexity. Given these enhancements, all the user would have to do is set up the device and then the image would only need to adequately align with all the features of the structure.

Their third goal resided in the area of power. More specifically, to power GPMS, certain voltage/current requirements must be met. The Power Supply Unit (PSU) most likely has a larger output voltage/current than what GPMS may need. Therefore, designing a buck-down regulator, housed within a custom Printed Circuit Board (PCB), designed to drop down the output PSU voltage/current to the appropriate specifications of GPMS would be necessary.

Below is a summarization of the specific types and descriptions of each of the authors' objectives.

- **Basic Objectives**
  - The projector seamlessly displays an image corresponding to the user input.
  - The generated image intelligently aligns with distinct features on the targeted structure.
- **Advanced Objective**
  - Implementation of an input method to adjust sensitivity and threshold for image outlines.
- **Stretch Objectives**
  - Develop a user-friendly frontend that empowers users to highlight features on the structure for emphasis.
  - Automatic calibration of the device to the specific features present on the wall.
  - Design a buck-down regulator within a custom PCB to drop the PSU output voltage/current to GMPS input specifications.

2.4 Requirements Specification

2.4.1 Target Project Specifications

The target specifications, which are listed in table format below, are additionally displayed in the House of Quality (HOQ) below. The specifications in the table below were derived with the main motivation of assembling GPMS efficiently.

Category	Target
Size	7 inches x 10 inches x 8 inches
Weight	~10 pounds
Network Scalability	1 Server : Many Connections
Visual Quality	1920 x 1080 pixels
Recovery Time	< 10 seconds
Projection Range	> 6 feet
Calibration Autonomy	100% Fully Autonomous*
Elapsed Time	< 5 minutes
Alignment Accuracy	< 8 mm
AI Generation Time	< 1.5 minutes/image

Table 2.4.1 *Target Project Specifications*

*Note: Complete autonomy is an advanced goal.

2.4.2 Elapsed Time

The importance of assessing elapsed time in this project lies in its role in ensuring a swift and efficient user experience from start to finish. Keeping the setup and image projection process within the set timeframe of under five minutes, was essential for maintaining usability, especially in scenarios where time-sensitive deployment would be necessary. By minimizing the time required for setup and configuration, users could then quickly initiate image projections without unnecessary delays, which is especially valuable in dynamic settings or during live presentations.

In assessing elapsed time, the project aimed to streamline the process from setup to final image projection to be achieved within less than 5 minutes. Testing would lead the authors to conduct multiple trials to record the time taken from start to finish. Each test run followed a sequence that began with setting up the projector, calibrating the projection area, adjusting necessary settings such as fine-tuning threshold values, entering the desired input prompt, and finally selecting an image for projection. This baseline process measured the time needed to get everything prepared and aligned before initiating the AI generation component.

When the AI image generation process was included, the overall elapsed time naturally extended, but it still comfortably remained within the desired target. This testing approach helped verify that the setup process was both efficient and repeatable, ensuring that, regardless of minor variations across trials, the device could consistently achieve the complete workflow within the targeted time frame.

2.4.3 Visual Quality

Ensuring a visual quality of 1920 x 1080 pixels within GPMS was paramount for delivering immersive and high-fidelity projections that captivate audiences and convey content with clarity and precision. With a resolution of 1920 x 1080 pixels, GPMS could confidently render various images and graphics with exceptional detail, allowing for the faithful reproduction of intricate designs, vibrant colors, and subtle textures. Whether it was projecting dynamic animations, lifelike visuals, or informative graphics, the system's high-resolution output ensured every pixel contributed to a visually stunning experience that leaves a lasting impression on all viewers.

Moreover, a visual quality of 1920 x 1080 pixels enabled GPMS to accommodate a wide range of content types and formats, from high-definition videos to detailed graphics and text. This versatility was essential for catering to diverse creative visions and content requirements, whether the system is used for artistic installations, corporate presentations, or educational purposes in University settings. By delivering crisp and clear visuals with a resolution that met or exceeded standard high-definition specifications, GPMS ensured that content creators have the flexibility and freedom to unleash their creativity and realize their vision with uncompromising visual fidelity.

Furthermore, a resolution of 1920 x 1080 pixels enhanced audience engagement and immersion by creating a seamless and immersive viewing experience that transported viewers into the heart of the content. Whether it was exploring virtual worlds, experiencing interactive narratives, or witnessing breathtaking visuals, audiences could now fully immerse themselves in the projected content without distraction or loss of detail. This heightened level of immersion not only enhanced the impact of the storytelling experience but also fostered a deeper connection between viewers and the content, leaving a lasting impression that resonated long after the projection had ended. By prioritizing visual quality of 1920 x 1080 pixels, GPMS elevated the art of projection mapping to new heights, setting the stage for unforgettable visual experiences that pushed the boundaries of creativity and technology.

2.4.4 Alignment Accuracy

The alignment accuracy test assesses how precisely the projected image aligns with the intended target surface. This accuracy is critical for ensuring that images are displayed exactly as intended, maintaining sharpness and clarity without any distortion or offset. During testing, measurements were taken at multiple points on the projection surface to gauge the consistency of alignment.

Analyzing this data provided insight into the GPMS's precision, with each measurement contributing to an overall understanding of its accuracy and consistency. High alignment accuracy not only enhances the visual quality by minimizing distortion but also ensures that images are projected in the exact intended locations.

This precision supports reliable, high-quality image projections. Achieving precise alignment would prove to be essential for the quality and reliability of GPMS, as even minor deviations could potentially impact visual fidelity and reduce the effectiveness of the display. The system's ability to consistently meet and exceed alignment expectations ensures a dependable foundation for high-quality image projections, essential for GPMS where accuracy directly affects user experience.

2.4.5 AI Generation Time

AI generation time was a crucial metric for assessing how quickly the system can produce AI generated images in response to user prompts, impacting both the efficiency and responsiveness of the device. A shorter generation time means that users experience minimal delay, which would be especially important for applications where quick image outputs are essential to maintain flow and user engagement.

In testing, the authors found that the data collected showed that AI generation times were consistently within a targeted range, demonstrating stable and reliable AI performance. This stability was important because it allowed for predictable timing, ensuring users can depend on a responsive system. Although the generation time itself was promising, the project considers the complete delivery time, as transmitting images back to the user also affects the total experience. By continuously monitoring these factors, the project ensured that it aligns with its efficiency and speed goals, enhancing both user satisfaction and system reliability.

2.4.6 Projection Range

Ensuring a projection range of greater than 6 feet within GPMS was fundamental to accommodating diverse spatial configurations and delivering immersive visual experiences across various environments. With a robust projection range, the system effectively covered larger surfaces, which allowed for greater flexibility in venue selection and setup. Whether it was projecting onto expansive building facades, wide interior spaces, or outdoor landscapes, a generous projection range ensured that the GPMS can adapt to different spatial constraints while maintaining optimal image quality and clarity.

Moreover, a projection range exceeding 6 feet enhances accessibility and audience engagement by enabling projections to be visible from a decent distance. This was particularly crucial for outdoor installations, large-scale events, holiday events, gatherings, or venues with expansive seating arrangements. By reaching farther distances, GPMS expanded its reach to a broader audience, ensuring that everyone within the viewing area can fully immerse themselves in the projected content. Whether it was a captivating visual narrative, interactive art display, or informative presentation, an extended projection range maximized the system's impact and ensured that the intended message or experience reaches its audience effectively.

Furthermore, a robust projection range within GPMS opened up possibilities for creative expression and innovative storytelling techniques. Artists, designers, and content creators can now leverage the system's extended reach to experiment with larger-than-life projections, immersive environments, and interactive installations that transcend traditional boundaries. From immersive art installations that transform entire cityscapes to interactive experiences that invite audience participation from afar, a projection range exceeding 6 feet can empower creators to push the boundaries of storytelling and engage audiences in unforgettable ways. By prioritizing a generous projection range, GPMS became a versatile tool for artistic expression, entertainment, education, and cultural enrichment, enriching the lives of audiences worldwide.

2.4.7 Recover Time

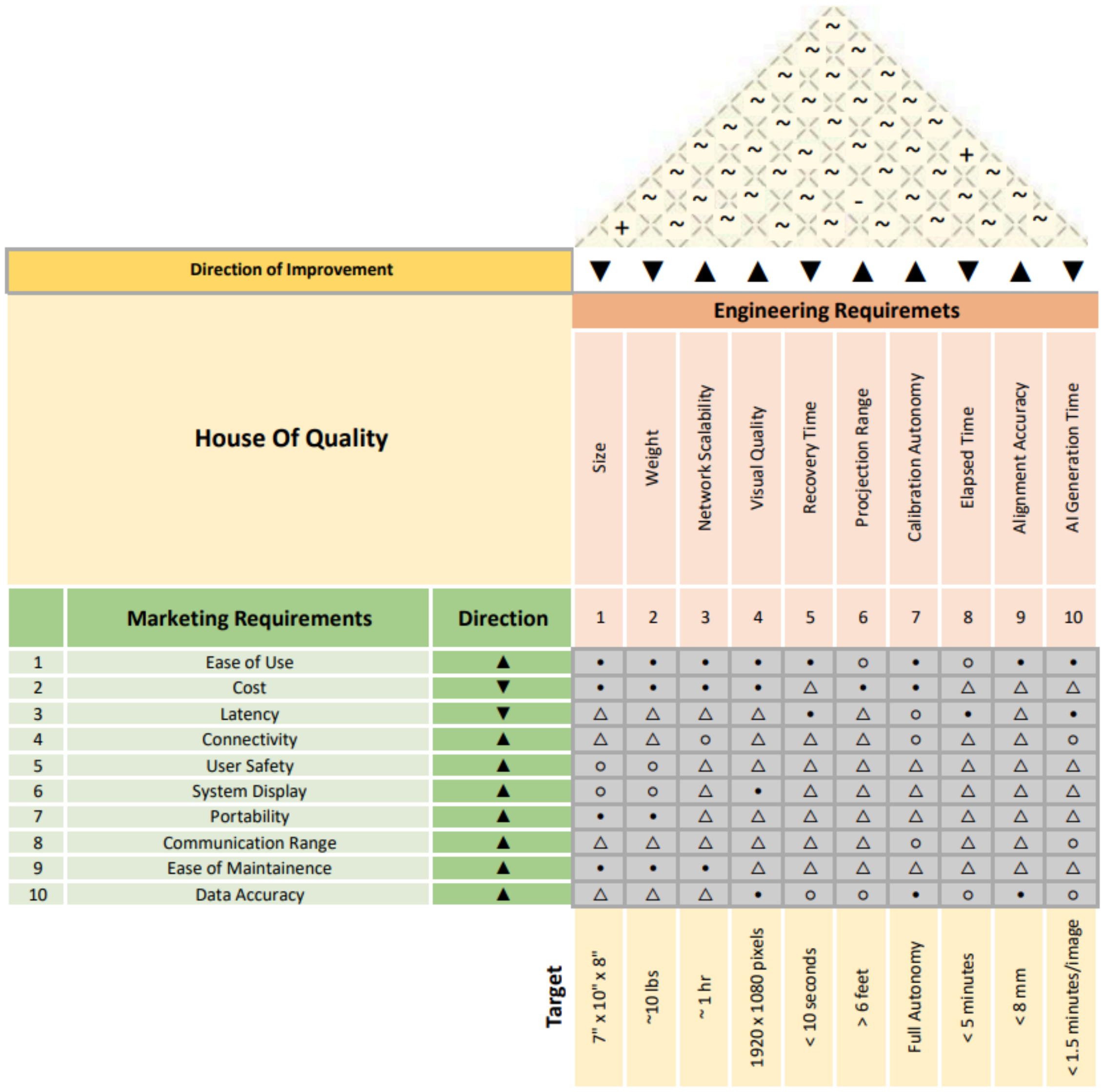
Maintaining a recovery time of less than 10 seconds within GPMS was essential for ensuring uninterrupted operation and seamless user experiences. In dynamic environments where technical glitches or interruptions occurred, a swift recovery time proved to be paramount for minimizing disruptions and maintaining the continuity of the storytelling experience. Whether it's a momentary power outage, software glitch, or even hardware malfunction, the ability of GPMS to swiftly recover within 10 seconds ensured that users could quickly resume their interactions with the system without warranting any significant downtime and/or frustration.

Furthermore, a rapid recovery time was crucial for preserving audience immersion and engagement during live events or performances. In scenarios where the GPMS is used for real-time projection mapping during concerts, theatrical productions, or interactive installations, any downtime could certainly have a large potential to take away from the overall experience and unfortunately disrupt the audience's engagement momentum. By minimizing the recovery time to no more than 10 seconds, GPMS can now seamlessly adapt to any unforeseen challenges, ensuring that the show goes on without missing a beat. This firm reliability enhanced the credibility of GPMS as a solid professional-grade tool for immersive storytelling and live entertainment applications.

Moreover, a quick recovery time within the GPMS will prove to bolster confidence among users and event organizers, fostering trust in the system's reliability and performance. Whether it has been deployed for corporate presentations, experiential marketing campaigns, or cultural events, the ability of the GPMS to swiftly recover from technical hiccups instills peace of mind and reassures stakeholders that their investment in the technology is well-founded. By prioritizing a recovery time of less than 10 seconds, GPMS demonstrated it has ample resilience and readiness to overcome challenges, which presented itself as a dependable solution for delivering impactful visual experiences in any setting.

2.4.8 House of Quality

The House of Quality (HOQ) below depicts the Target Specifications listed in Table 2.4.1. In addition, Engineering Requirements, Marketing Requirements and their respective direction of improvements and relationship with each other are listed. The correlations between the Engineering Requirements are also listed in the relationship matrix/roof of the HOQ.



House Of Quality

	Direction of Improvement		▼	▼	▲	▲	▼	▲	▲	▼	▲	▼
			Engineering Requiremets									
			Size	Weight	Network Scalability	Visual Quality	Recovery Time	Procjection Range	Calibration Autonomy	Elapsed Time	Alignment Accuracy	AI Generation Time
	Marketing Requirements	**Direction**	1	2	3	4	5	6	7	8	9	10
1	Ease of Use	▲	•	•	•	•	•	○	•	○	•	•
2	Cost	▼	•	•	•	•	△	•	•	△	△	△
3	Latency	▼	△	△	△	△	•	△	○	•	△	•
4	Connectivity	▲	△	△	○	△	△	△	○	△	△	○
5	User Safety	▲	○	○	△	△	△	△	△	△	△	△
6	System Display	▲	○	○	△	•	△	△	△	△	△	△
7	Portability	▲	•	•	△	△	△	△	△	△	△	△
8	Communication Range	▲	△	△	△	△	△	△	○	△	△	○
9	Ease of Maintainence	▲	•	•	•	△	△	△	△	△	△	△
10	Data Accuracy	▲	△	△	△	•	○	○	•	○	•	○
		Target	7" x 10" x 8"	~10 lbs	~ 1 hr	1920 x 1080 pixels	< 10 seconds	> 6 feet	Full Autonomy	< 5 minutes	< 8 mm	< 1.5 minutes/image

Figure 2.4.8 *House of Quality*

2.5 Diagrams

2.5.1 Diagram Overview

GPMS was essentially composed of a projector, camera, and touch screen monitor attached to a hardware enclosure along with additional hardware to allow onboard calibration of the image to the target surface. The overall system can be displayed via the Hardware and Software Diagrams listed in Figure 2.5.2 and Figure 2.5.3, respectively.

2.5.2 Hardware Diagram

The Hardware Diagram is listed below with wireless connections being depicted with a dotted line and wired connection being depicted with a solid connection type. Additionally, the Hardware Enclosure is depicted as the two opaque rectangles which hold the three important Hardware components.

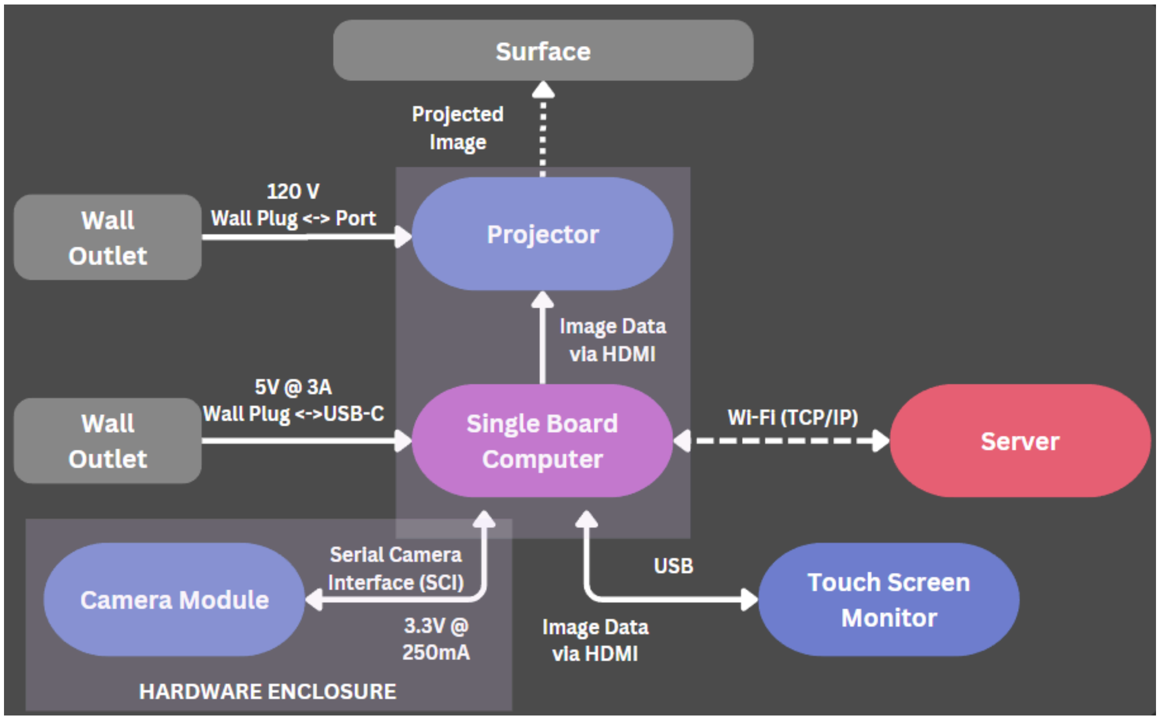


Figure 2.5.2 *Hardware Block Diagram*

The hardware components of GPMS began with assembling and integrating selected parts essential to the project's functionality. At the core of the system was a Single Board Computer—the Raspberry Pi 5—serving as the central processing unit for GPMS. This Raspberry Pi 5 is powered through a USB-A to USB-C connection by a power supply unit, specifically a power bank that provides the necessary 5V at 3A.

A touchscreen monitor connects to the Pi, sending input data via USB and receiving image data back through an HDMI connection. The camera module links to the Pi via the Serial Camera Interface (SCI), capturing images that it sends to the Pi. The camera is powered directly by the Pi with a 3.3V, 250mA supply. Once the images are processed, the Pi connects over Wi-Fi to a local server using TCP/IP protocols to generate AI images, which are then sent back to the Pi. The Pi forwards this image data via HDMI to the projector, which is connected to a wall outlet, projecting the AI-generated image onto a target surface.

The hardware design includes a 3D-printed enclosure made from PLA filament, positioned atop the projector and secured with Velcro. This enclosure houses the touchscreen monitor, Raspberry Pi 5, and camera. Designed to account for minor variances in 3D printing dimensions, the touchscreen monitor is flush with the enclosure's top, resembling a phone screen, while the Raspberry Pi 5 is securely attached to the enclosure floor with Velcro to prevent movement. The camera is fastened to the front of the enclosure above the projector while also being parallel on the same plane with it, ensuring consistent alignment between the captured image and projected display.

Additionally, the enclosure conceals all cabling associated with the touchscreen, minimizing cable clutter for users. It includes a side opening for external hardware cables to connect internally to the Pi. The power bank sits beneath the projector, providing stability and keeping the top assembly lightweight. This setup ensured both a streamlined appearance and solid functional organization of GPMS hardware components.

2.5.3 Software Diagram

The Software Diagram is listed below with three main separated sections in charge of facilitating the software integration of GPMS. The sections part of the software process include the 'Main Loop', 'AI Server' and 'Computer Vision Module'.

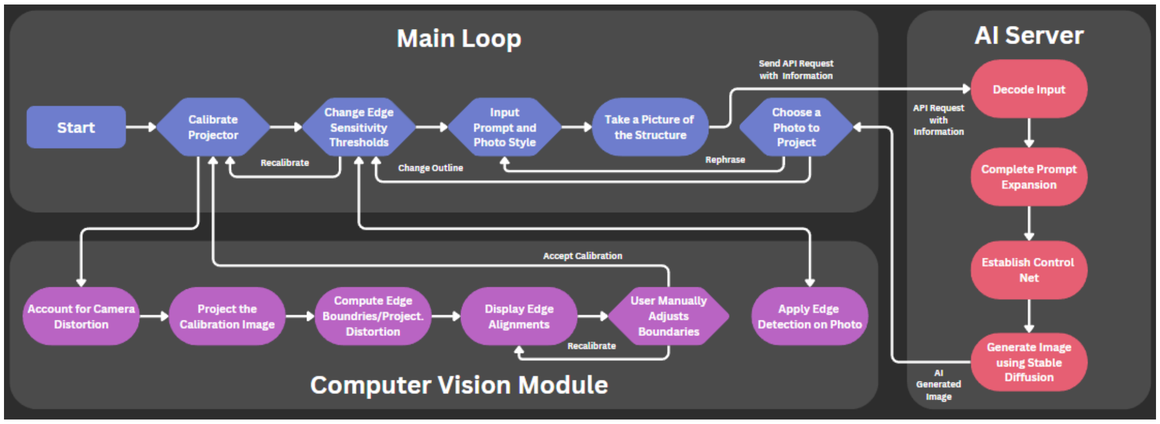

Main Loop
Start
Calibrate Projector
Change Edge Sensitivity Thresholds
Input Prompt and Photo Style
Take a Picture of the Structure
Choose a Photo to Project
Send API Request with Information
Recalibrate
Change Outline
Rephrase
AI Server
Decode Input
API Request with Information
Complete Prompt Expansion
Establish Control Net
Generate Image using Stable Diffusion
AI Generated Image
Accept Calibration
Account for Camera Distortion
Project the Calibration Image
Compute Edge Boundries/Project. Distortion
Display Edge Alignments
User Manually Adjusts Boundaries
Recalibrate
Apply Edge Detection on Photo
Computer Vision Module

Figure 2.5.3 *Software Block Diagram*

The GPMS software application was structured around three core components: the Main Loop, the Computer Vision Module, and the AI Server. Together, these elements enabled users to generate context-aware projections based on text prompts. At startup, the application's first task was camera calibration to correct lens distortion and define the projection area, ensuring that projected images would align accurately with the target surface.

Following calibration, users adjusted edge detection thresholds within the Computer Vision Module, enabling the system to recognize the boundaries of the projection space. With calibration and edge detection complete, users could then enter a descriptive prompt for their desired image, which, along with processed surface data, was sent to the AI Server. Using Stable Diffusion technology, the server generated two unique images to return to the user based on this input. Users then selected their preferred image for seamless mapping onto the target surface, achieving a precise and visually aligned projection.

The Main Loop acted as the primary user interface. Here, users began by designating the corners of the projection area for calibration, followed by setting sensitivity thresholds for edge detection. After confirming alignment with the projected boundaries, users entered a descriptive prompt of their choice and selected a photo style (Realistic or Animated). Then, this information was sent to the AI Server, which returned two AI-generated image options. Users could then choose an image to be projected accurately onto the calibrated area.

In the Computer Vision Module, lens distortion was corrected first. The system then projected a calibration image onto the surface, defining viewable boundaries regardless of GPMS's position and/or angle, with adjustments made to ensure a perfect fit to the surface. Edge detection was then activated, allowing users to fine-tune alignment and confirm the setup before finalizing calibration.

The AI Server handled the final processing step. A still frame of the projection surface, combined with the user's prompt and metadata, was sent to a Python server via HTTPS POST. The server expanded the prompt as needed and created a “control net” to guide the Stable Diffusion model (specifically, Stable Diffusion XL) during image generation. The control net preserved the primary edges and features of the original surface, ensuring consistency across image iterations. Once the final image was generated, it was sent back to GPMS for projection, resulting in a seamless user experience powered by advanced computer vision and AI integration.

2.6 GPMS - An Illustration

The image below was generated using ChatGPT-4 to fuse the elements of modern technology into a single device ultimately mirroring the vision of what GPMS would look like. [1]

This image features a projector with dual-camera, capturing multiple views. It is sleek and futuristic, with a silver and black color scheme common in contemporary technology devices. The ports and ventilation grills lend to high functionality and processing power, which would be necessary for high-definition projection onto any 3D surface. This image is a visual representation and is currently not available in the consumer market. It was intended to be used solely for brief illustrative purposes during technical presentations of all disciplines and to further envision future technology.

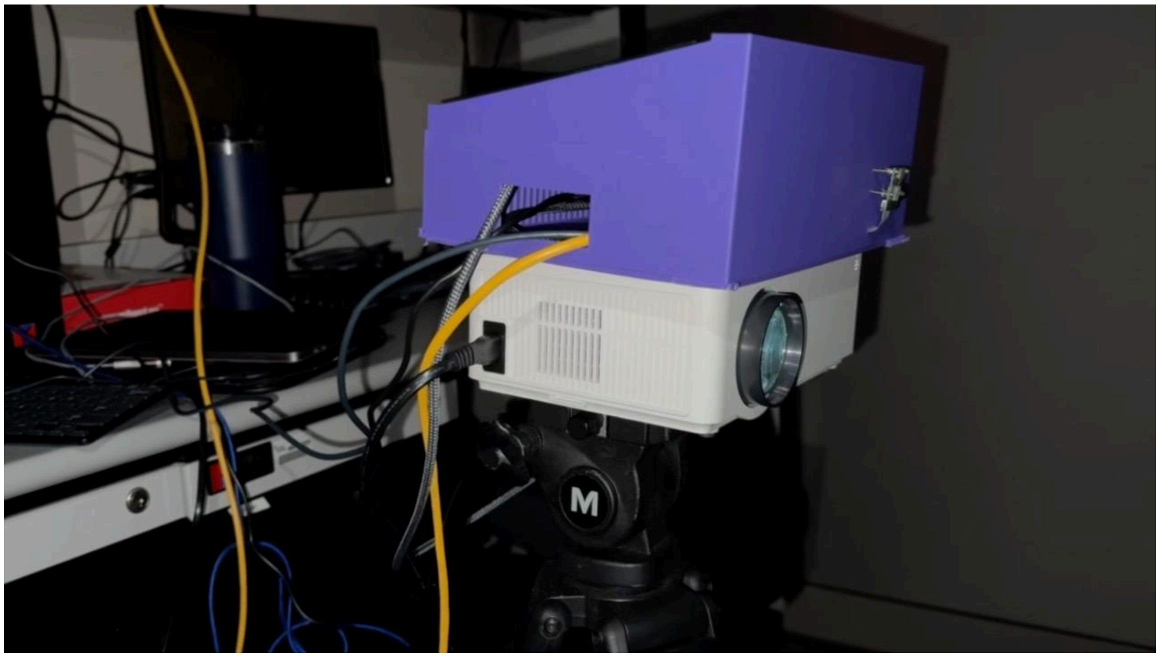


Figure 2.6 *GPMS - An Illustration*

Chapter 3 Research

3.1 Hardware Integration Strategies

In the field of projection technology, there are two primary approaches to hardware integration: a self-contained solution that provides all necessary components/hardware and a user-dependent model that relies on user-provided hardware. The user-dependent model, exemplified by products like LumaBox, requires users to download the software onto their devices, capture images of structures using their own equipment, and then transfer these images to a computer for processing. The computer, in turn, is connected to a projector to display the final output.

Alternatively, a self-contained solution bundles all necessary components - the projector, webcam, computer, and monitor - into a single, integrated product. This approach allows for more intricate low-level work, ensuring seamless communication between the components and simplifying the user experience, particularly for those who may not be technologically adept.

As a team from the College of Electrical and Computer Engineering at the University of Central Florida, the authors focused on creating a comprehensive hardware solution. They firmly believed that including all necessary components within the product would not only align with their academic focus but also enhance the user experience by being able to reduce the need for technical expertise and ensuring compatibility between components.

Central to the authors' self-contained approach was the integration of a webcam mounted above the projector while also being parallel on the same plane with it. This design choice means that users only need to position the projector correctly and specify when the image should be captured, simplifying the setup process.

However, the authors also considered the potential benefits and drawbacks of user-dependent solutions, as outlined in the following table:

Aspect	User-Contained Solutions	User-Dependent Solutions
Compatibility	Optimized for seamless compatibility between components	Requires compatibility with a wide range of user devices, potentially leading to unforeseen issues
User Complexity	Simplifies the user experience by reducing the need for additional device configuration	Adds complexity at the software and hardware levels, requiring users to

		manage multiple devices and configurations
Software Requirements	Tailored to the integrated hardware, minimizing software complexity	Could necessitate the development of platform-agnostic applications, increasing the workload and complexity
Implementation	Straightforward, focusing on the integrated components without adding layers of complexity in software and hardware management	Introduces significant complexity, requiring compatibility management for various user-provided components
Scalability	Limited scalability due to the fixed configuration of integrated components	Highly scalable, as users can upgrade their own hardware components as needed

Table 3.1 *User-Contained vs User-Dependent Solutions*

After careful consideration, the authors determined that the benefits of a self-contained solution outweighed the potential advantages of relying on user-dependent solutions. By providing a complete, fully integrated product, the authors were able to ensure a streamlined, user-friendly experience while minimizing compatibility issues and software complexity. This approach aligned with their academic focus, simplified the user experience, and set their project apart by offering a superior, turnkey solution that optimized performance through carefully selected and integrated components.

3.2 Hardware Comparison and Selection

3.2.1 MCU vs FPGA

The authors were torn at first when deciding whether they should proceed with an MCU or an FPGA. After discussing with various faculty mentors and talking through their experiences with the two options, the authors drew various conclusions.

In deciding between a MCU and a Field-Programmable Gate Array (FPGA) for GPMS, there were distinct trade-offs that were heavily considered. While FPGAs offer superior processing power and precision tailored for specialized tasks, they come with higher costs, increased power consumption, and complexity in programming. Conversely, to the benefit of GPMS, MCUs prioritize low power consumption and user-friendliness, featuring readily available peripherals, extensive documentation, and support for common programming languages, simplifying development and debugging processes. [9]

The choice between an FPGA and MCU hinges on factors such as power efficiency, ease of use, and scalability which will be further explored throughout the research stage of the project. MCUs offer a straightforward development process with built-in features for stability and standardized architecture, making them ideal for rapid prototyping and easy integration. In contrast, FPGAs actually demand specialized knowledge and customized solutions for error handling, yet they provide unparalleled scalability for future system upgrades. Additionally, while MCUs boast environmental robustness with built-in temperature sensors and watchdog timers, FPGAs may require additional measures to mitigate sensitivity to environmental factors like temperature and electromagnetic interference. [9]

Opting for an MCU over an FPGA can offer distinct advantages in specific project scenarios, particularly those requiring a balance of computational power, versatility, and ease of use. The table below encompasses a direct comparison between an MCU and an FPGA. [9]

Feature	MCU	FPGA
Computational Power	Sufficient for a wide range of tasks, including multimedia processing, networking, and computing	Superior, tailored for specialized tasks
Power Consumption	Low, prioritizing energy efficiency	Higher, due to enhanced processing capabilities
Programming Complexity	User-friendly; supports common programming languages, simplifying development and debugging	Complex; requires specialized knowledge for programming and debugging
Development Process	Straightforward with built-in features for stability, extensive documentation, and support	Demands customized solutions for error handling, scalability
Scalability	Standardized architecture, ideal for rapid prototyping and integration	Unparalleled, offers flexibility for future system upgrades

Environmental Robustness	Built-in sensors and timers for temperature and stability	May require additional measures for environmental factors
Operating System	Linux-based, providing a familiar environment with extensive software support	Not specified; often requires custom software solutions
Connectivity and Peripherals	Extensive, including USB, HDMI, Ethernet, Wi-Fi, Bluetooth	Not specified; customization might be necessary for specific needs
Cost	Generally lower, making it affordable and accessible	Higher, reflecting the advanced capabilities and flexibility

Table 3.2.1 *MCU vs FPGA*

After careful consideration, the authors opted to use an MCU. Overall, the combination of computational power, software compatibility, connectivity options, and affordability made it the compelling choice for GPMS requiring a flexible and user-friendly MCU solution, particularly when compared to the specialized and potentially more complex nature of alternatives such as FPGAs. [9]

The MCU presented several compelling reasons for its suitability in the context of GPMS such as its ability to provide sufficient computational capabilities for various tasks, including multimedia processing, networking, and general-purpose computing. [9]

Additionally, the MCU can operate on different Operating Systems (OS), which provides a familiar computing environment and extensive software support. This compatibility with different operating systems enables developers to leverage a vast ecosystem of software libraries, tools, and applications, facilitating rapid prototyping and development. [9]

Moreover, the MCU offered an opportunity to include a rich set of built-in peripherals and connectivity options, including USB ports, HDMI output, Ethernet, Wi-Fi, and Bluetooth. These features make it highly versatile, allowing for seamless integration with various devices and systems, such as sensors, displays, and network interfaces. [9]

Overall, the combination of computational power, software compatibility, connectivity options, and affordability made the MCU the compelling choice for

GPMS, requiring a flexible and user-friendly solution, particularly when compared to the specialized and potentially the more complex nature of alternatives such as FPGAs. [9]

3.2.2 MCU vs Development Board

When moving forward with the development of the project, the authors explored the possibilities of using either an MCU housed on a custom PCB or a Development Board. Therefore, the authors specifically researched a few aspects surrounding the two options including Hardware Integration, Processing Power, Rich Peripheral Support, Software Development Environment, and Community Support and Resources.

Development boards often come with integrated components such as high-quality cameras. This integration reduces the complexity of sourcing and interfacing separate hardware components, ensuring compatibility and streamlined development.

Creating an immersive storytelling environment with dynamic image generation and sophisticated computer vision techniques requires significant processing power. Development boards typically feature more powerful processors than simple MCUs on a custom PCB, allowing for faster image processing, AI inference, and real-time interactions with all users.

The project required interaction with various peripherals such as cameras, touchscreens, and projectors. Since development boards offer a wide range of peripheral interfaces and support for protocols like SPI, I2C, and HDMI, it was the clear choice as it was easier to connect and communicate with external devices.

Additionally, development boards often come with comprehensive Software Development Kits (SDKs) and libraries tailored to the specific hardware features. These SDKs provide Application Programming Interfaces (APIs) and tools for rapid prototyping, debugging, and optimizing the software, to accelerate the development process and reduce time-to-market.

Notably, development boards, especially popular ones, benefit from a large community of developers and enthusiasts. This community provides valuable resources such as documentation, tutorials, sample code, and forums for troubleshooting and sharing knowledge, easing the learning curve and fostering innovation.

A development board offered several advantages over a simple MCU for the goals outlined in the project which are highlighted in the table below.

Aspect	Development Board	MCU
Processing Power	Typically higher, often featuring multicore processors	Limited processing power, single-core
Integrated Peripherals	Comprehensive, including sensors, displays, etc.	Basic, may require additional components
Connectivity Options	Extensive, including Wi-Fi, Bluetooth, Ethernet, etc.	Limited, may require external modules
Development Environment	Rich software development kits (SDKs) and libraries	Basic toolchains, limited development support
Community Support	Large community with extensive resources	Limited community, fewer resources
Cost	Higher due to integrated features and performance	Lower, especially for simple MCU setups

Table 3.2.2 *Development Board vs MCU Comparison*

Given that development boards offered a comprehensive solution for the GPMS project by integrating essential hardware components such as high-quality cameras, touchscreens, and powerful projectors, the authors decided to use a development board. This integration effectively streamlined hardware setup and also ensured compatibility, simplifying the development process.

In addition to integrated hardware, development boards provide superior processing power compared to simple MCUs. This enhanced capability enables faster image processing and AI inference, crucial for creating an immersive storytelling environment with dynamic image generation.

Furthermore, development boards offer comprehensive peripheral support, which reduce the complexity of interfacing external devices like cameras and touchscreens. Coupled with robust SDKs and libraries tailored to their hardware features, development boards facilitate efficient software development, debugging, and optimization. Supported by a large community of developers and enthusiasts, development boards provide valuable resources and support, which made them the preferred choice for efficiently realizing the ambitious goals of the GPMS project. [10] [11]

3.2.3 Development Board Comparison

After narrowing down the options, the authors compared a Raspberry Pi 5, ESP32-WROOM-32, and MSP-EXP430fr6989. Each of the development boards are comprehensively listed below in the respective tables.

The **Raspberry Pi 5** has the following stand out features: [12]

- 2×micro HDMI
- 2×USB 2.0
- 2×USB 3.0
- 2×CSI camera/DSI display ports
- Single-lane PCIe FFC connector
- UART connector
- RTC battery connector
- Four-pin JST-SH PWM fan connector
- PoE+-capable Gigabit Ethernet (1Gb/s)
- 2.4/5GHz dual-band 802.11ac Wi-Fi 5 (300Mb/s)
- Bluetooth 5, Bluetooth Low Energy (BLE)
- microSD card slot
- USB-C power (5V, 5A (25W) or 5V, 3A (15W) with a 600mA peripheral limit)

The **ESP32-WROOM-32** has the following stand out features: [13]

- Wi-Fi Protocols
    - 802.11 b/g/n (802.11n up to 150 Mbps)
    - A-MPDU and A-MSDU aggregation with 0.4 µs guard interval support
    - Center frequency range of operating channel: 2412 ~ 2484 MHz
- Bluetooth
    - Protocols: Bluetooth v4.2 BR/EDR and Bluetooth LE specification
- Radio
    - NZIF receiver with sensitivity of -97 dBm
    - Class-1, class-2, and class-3 transmitter
    - Adaptive Frequency Hopping (AFH)
- Hardware:
    - Module interfaces: SD card, UART, SPI, SDIO, I2C, LED PWM, Motor PWM, I2S, IR, pulse counter, 39 GPIO, capacitive touch sensor, ADC, DAC, Two-Wire Automotive Interface (TWAI®)
    - Integrated crystal: 40 MHz crystal
    - Integrated SPI flash: 4 MB
    - Operating voltage/Power supply: 3.0 V ~ 3.6 V
    - Operating current (Average): 80 mA
    - Minimum current delivered by power supply: 500 mA
    - Recommended operating ambient temperature range: -40 °C ~ +85

°C
    - Package size: 18 mm × 25.5 mm × 3.10 mm
    - Moisture sensitivity level (MSL): Level 3

The **MSP-EXP430fr6989** has the following stand out features to integrate with the custom PCB: [14]

- MSP ULP FRAM-based MSP430FR6989 16-bit MCU
    - 100 uA/MHz active mode and 350 nA standby with RTC and 3.7 pF crystal
    - Certified ULPBench score of 109
    - 128 KB FRAM
    - 16-Bit RISC architecture up to 8-MHz FRAM access/ 16MHz system clock speed
    - 320-segment LCD controller
    - Extended Scan Interface
    - 16 channel 12-bit ADC
    - Comparator
    - 5 Timers
    - Direct memory access
    - 256-bit AES
    - 83 GPIO
- EnergyTrace++™ Technology available for ultra-low-power debugging
- 40 pin LaunchPad standard leveraging the BoosterPack ecosystem
- Onboard eZ-FET emulation
- 2 buttons and 2 LEDs for User Interaction
- Segmented LCD
- Pins for direct access to the Extended Scan Interface

It is important to note that the overall project deliverable will benefit most from the development board, that can produce higher processing power, contain greater memory capacity, allow for direct video output, provide an operating system and multimedia support, and host various peripheral options. The comparisons of these factors are listed in the table below.

Main Factors	**Raspberry Pi 5**	**ESP32-WROOM-32**	**MSP-EXP430fr6989**
Processor	Broadcom BCM2712 processor	Single/Dual-Core 32-bit LX6 Microprocessor up to 240 MHz	MSP ULP FRAM-based MSP430FR6989 16-bit MCU
Memory	4GB, 8GB, 16GB LPDDR4X	4 MB SPI flash	128 KB FRAM

Wireless Connectivity	2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE	802.11 b/g/n (802.11n up to 150 Mbps), Bluetooth v4.2 BR/EDR and Bluetooth LE specification	N/A
Ports	2×micro HDMI 2×USB 2.0 2×USB 3.0 2×CSI camera/DSI display ports	N/A	N/A
GPIO	Standard 40-pin GPIO header (fully backwards-compatible with previous boards)	39 GPIO, capacitive touch sensor	83 GPIO
Video Output	Dual Micro HDMI: Dual 4Kp60 display support. MIPI DSI: 2-lane display port for touchscreen. MIPI CSI: 2-lane camera port for Pi cameras. 4-pole AV: Analog audio + composite video output	None	N/A
Operating System and Multimedia Support	H.265: 4Kp60 hardware decode. H.264: 1080p60 decode, 1080p30 encode. Graphics: OpenGL ES 3.1 support for improved 3D rendering and graphics	N/A	N/A
Power Supply	USB-C power	Operating	N/A

	(5V, 5A (25W) or 5V, 3A (15W) with a 600mA peripheral limit)	voltage/Power supply: 3.0 V ~ 3.6 V	
Operating Temperature	0°C to 80°C	-40 °C ~ +85 °C	N/A

Table 3.2.3 *Development Board Comparison*

In essence, the Raspberry Pi 5 is the most powerful development board despite the fact that it consumes the most power. However, the Raspberry Pi 5 is the optimal solution for the project because it has higher processing power, greater memory capacity, direct video output, operating system and multimedia support, and various peripheral options. These features make it more suitable for complex projects that require high computational power and versatility. It also has a large community and extensive documentation, which can be very helpful during the development process. As a result, the authors decided to use a Raspberry Pi 5 for GPMS.

In summary, despite the ESP32-32-WROOM-32 and MSP-EXP430fr6989 being two versatile and powerful microcontrollers for a select range of IoT applications, it does not suit the GPMS sufficiently. On the other hand, the Raspberry Pi 5 lends itself as the more suitable development board for GPMS.

3.2.4 Projector Comparison

The tables below encapsulates the technical comparison of technology, color production, brightness, throw ratio, suitability for well-lit rooms, features for educational and business environments, and an overall technical and holistic recommendation for GPMS.

The projector that would be used for GPMS would serve as an essential part of the entire system. Francisco initially temporarily owned an Epson PowerLite 585W ultra short throw projector. Therefore, before the comparison the authors decided to test the functionality of the projector already owned and available. After testing its functionality, the authors collectively decided to not use it.

The decision not to use the ultra throw projector, specifically the Epson PowerLite 585W, stemmed from considerations regarding image distortion and the practicality of correcting it. When projecting onto a surface, especially in unconventional settings like corners or non-flat surfaces, there can be significant image warping. The distance and angle at which the projector emits light relative to the surface can cause distortion. In this case, the positioning of the Epson PowerLite 585W might result in warping that could distort the image, particularly in the corners.

The authors determined that the camera's angle relative to the surface is crucial for capturing images with minimal distortion. By ensuring that the camera angle was perpendicular to the surface, they aimed to capture an even image with as little distortion as possible. This suggested a deliberate effort to optimize the setup for image clarity and accuracy.

To counteract the distortion caused by the projector's angle and the surface's geometry, a corner transformation was necessary. This process involves digitally altering the image to correct for the distortion caused by the unique projection setup. However, implementing corner transformation can be complex and may not always yield satisfactory results, especially if the distortion is severe. Furthermore, adjusting the image through warping and transforms proved to be challenging, requiring precise calibration and could also potentially lead to image degradation or artifacts.

Considering these factors, the authors concluded that using a different projector better suited to the setup would be more practical and efficient. The alternative projector would offer features or specifications that align more closely with the desired imaging requirements, minimizing the need for extensive digital correction or compromising on image quality. Ultimately, the goal of GPMS was to achieve the best possible immersive AI generated image projection with minimal distortion and technical complications. A technical comparison of projectors is in the table below. [15] [16] [17] [18]

Feature	**M8-F Auking**	**HAPPRUN H1**	**Panseba**
Price (USD)	57.98	79.99	99.99
Brightness (ANSI)	136	249	350
Native Resolution	1920 x 1080	1920 x 1080	1920 x 1080
Display Resolution	1080p	1080p	1080p
Connectivity	HDMI/USB/VGA	HDMI/USB	HDMI/USB
Maximum Image Size (Inches)	200	200	300
LED Life (Hours)	55,000	100,000	100,000
WiFi Connection	No	No	Yes

Bluetooth Connection	No	Yes	Yes
Weight (Pounds)	2.87	3.4	3.3
Dimensions (Inches)	7.95 x 6.3 x 2.91	9.3 x 8.1 x 3.5	9.0 x 7.0 x 3.0
Contrast Ratio	2000:1	10,000:1	10,000:1
Max Throw Distance (Feet)	18.5	22.97	16.4

Table 3.2.4 *Technical Projector Comparison*

The authors collectively selected the Panseba projector. Given the comparison in Table 3.2.4, the Panseba projector stands out for several reasons boasting several notable features which set it apart from its competitors such as the M8-F Auking and HAPPRUN H1.

With a brightness of 350 ANSI, the Panseba projector stood out as significantly brighter than both the M8-F Auking and HAPPRUN H1. This brightness ensured a clearer and more vibrant image, particularly in well-lit environments.

In terms of maximum image size, the Panseba projector can project up to 300 inches which surpasses both the M8-F Auking and HAPPRUN H1. This larger projection size enhances the immersive viewing experience.

One distinct advantage of the Panseba projector was its WiFi connection, which was a feature lacking in the other two projectors. This WiFi connectivity provides greater flexibility for content streaming and device connectivity.

The Panseba projector boasts an impressive LED life of 100,000 hours which equals the HAPPRUN H1 but surpasses the M8-F Auking by a considerable margin. This extended LED life meant that the projector could operate for longer periods of time before requiring a bulb replacement. The extent of the LED life that the Panseba projector offers was imperative for GPMS to operate for a substantial period of time.

For seamless connectivity with Bluetooth devices such as speakers or headphones, the Panseba projector offers a Bluetooth connection, akin to the HAPPRUN H1. In addition to its impressive features, the Panseba projector's compact size further enhances its appeal, making it an excellent choice for portability. Measuring just 9 by 7 by 3 inches, this projector is compact and lightweight, making it easy to transport from one location to another.

The compact dimensions of the Panseba projector did not compromise its performance or features. Despite its small size, it still offered impressive brightness, maximum image size, connectivity options like WiFi and Bluetooth, and a long LED life. This combination of portability and functionality made the Panseba projector a versatile solution for various applications specific to those of GPMS.

Finally, the Panseba projector's contrast ratio of 10,000:1 matched that of the HAPPRUN H1 but exceeded the M8-F Auking's. A higher contrast ratio translates to superior color depth, resulting in a more dynamic and vibrant image quality. While the Panseba projector was more expensive, these features justified the higher price point and made it a superior choice based on the table provided.

3.2.5 Camera Comparison

Selecting the appropriate camera type was crucial for the optimal computer vision integration and quality of the GPMS device. The camera was integral to the functionality of the system, capturing high-quality images necessary for the software to perform accurate analyses. Therefore, the camera must not only have been compatible with the Raspberry Pi 5 but also meet specific performance criteria, including resolution, frame rate, and low-light capabilities. This section compares various camera types, evaluating their suitability based on these essential parameters to ensure the chosen camera will be able to enhance the effectiveness and efficiency of the GPMS device.

For GPMS, two distinct camera data interfaces were compared: Universal Serial Bus (USB) and Camera Serial Interface (CSI). The following table compares these interfaces with the features they provide. [19]

Feature	USB Camera	CSI Camera
Compatibility	High compatibility with many devices	Primarily compatible with Raspberry Pi models
Connection	Connects via USB port	Connects through a dedicated CSI port
Ease of Setup	Generally plug-and-play	Requires specific setup on Raspberry Pi

Data Transfer Speed	Slower compared to CSI; depends on USB type	Faster, designed for high data rates
Power Consumption	Typically higher	Generally lower
Flexibility	More flexibility in positioning	Fixed positioning close to the Raspberry Pi
Image Quality	Varies widely; generally lower than CSI	Typically higher, optimized for image capture
Cost	Can be less expensive	Often more expensive but offers better quality

Table 3.2.5.1 *Camera Interface Comparison*

The comparison table between USB cameras and CSI cameras reveals distinct advantages and trade-offs for each type. USB cameras offer high compatibility with various devices and are generally easier to set up due to their plug-and-play nature. However, they may suffer from slower data transfer speeds and higher power consumption compared to CSI cameras. On the other hand, CSI cameras, while primarily compatible with Raspberry Pi models and requiring a more specific setup, provide faster data transfer rates and superior image quality. These cameras are also more energy-efficient but tend to be more expensive.

However, taking into account the above features, the authors all decided that a higher quality image and smaller power footprint than a standard USB camera made a camera featuring CSI a clear and straightforward decision for the project.

To enhance the capabilities of the GPMS device, incorporating a high-quality camera using CSI was imperative. This section focuses on two prominent CSI cameras: the OV5640 and the IMX477. Each camera brings unique advantages to image capturing tasks, tailored to different performance needs and technical requirements. Below is a comparative overview of their key features.

Feature	OV5640	IMX477
Resolution	5 Megapixels (2592x1944)	12.3 Megapixels (4056x3040)
Sensor Type	CMOS	CMOS
Frame Rate	15 fps at full resolution	60 fps at full resolution
Lens Compatibility	Fixed lens	Interchangeable lenses supported
Video Output	Supports 1080p at 30 fps	Supports 4K at 30 fps
Price Range	Budget-friendly	Premium pricing

Table 3.2.5.2 *Overall Camera Comparison*

The OV5640 and IMX477 cameras both offer robust solutions for projects requiring high-resolution imaging through the CSI interface. The OV5640 is a budget-friendly option suitable for applications requiring standard resolution and frame rates, while the IMX477 caters to more advanced needs with its higher resolution, faster frame rates, and support for interchangeable lenses.

However, it was important to note the application at hand and the limited computing power of the Raspberry Pi 5, specifically when looking to process a video feed at 4k 30 fps. While the resolution could easily be downscaled, the quality offered by the IMX477 seemed to be overboard, especially on GPMS whose cost is an important constraint/factor. These reasons have led the authors to ultimately choose the OV5640 for their camera selection.

3.2.6 Monitor Comparison

To ensure a seamless user experience and maintain the compact nature of the GPMS device, it was essential to integrate a monitor that connects directly to the Raspberry Pi 5 and serve as the frontend interface. The primary requirement for this monitor was that it needed to be a touch screen, eliminating the need for additional peripherals such as a mouse and keyboard. A touchscreen monitor allows for intuitive navigation and interaction with the GPMS software, streamlining the user experience and reducing the overall footprint of the device.

When selecting a touchscreen monitor for the GPMS device, several factors were considered, including screen size, resolution, touch technology, compatibility, and cost. The following table compares the three main types of touchscreen monitors. [20]

Feature	**Resistive Touch Screen**	**Capacitive Touch Screen**	**Infrared Touch Screen**
Touch Technology	Pressure-sensitive; works with any object	Responds to electrical conductivity; works with fingers or capacitive stylus	Uses infrared light to detect touch; works with any object
Durability	Can withstand scratches and impacts; suitable for harsh environments	More prone to scratches and damage; not suitable for harsh environments	High durability; can withstand scratches and impacts
Accuracy	Less precise; may require calibration	High precision and responsiveness	High precision and responsiveness
Multi-touch Support	Limited or no multi-touch support	Excellent multi-touch support	Good multi-touch support
Visibility	May have reduced clarity due to additional layers	Clear display; no additional layers affecting visibility	Clear display; no additional layers affecting visibility
Cost	Generally less expensive	More expensive than resistive touch screens	Often the most expensive option

Table 3.2.6.1 *Monitor Overall Comparison*

Based on the requirements of the GPMS device, a capacitive touch screen monitor was the most suitable choice. Capacitive touch screens offer high precision, excellent multi-touch support, and a clear display, which are all essential for a user-friendly interface. Although they may be more expensive than resistive touch screens, the benefits they provide in terms of user experience and functionality justified the investment. [20]

When making the final decision, it is important to consider the specific Raspberry Pi 5 model being used and ensure that the chosen touch screen monitor is compatible with both the Pi and the GPMS software. Additionally, the screen size should be appropriate for the intended use case, balancing readability and portability.

There are many different brands and types of capacitive touch screens. The authors decided that the best size would be either 7 or 10 inches measured by its diagonal. The authors wanted to ensure the screen would not be too large, increasing the cost and making the device bulkier, but also not too small where the interface would be too hard to see. The table below compares the two size options.

Factor	7-inch Screen	10-inch Screen
Screen Size	Smaller screen size may make it more challenging to design a user-friendly interface that displays all necessary information.	Larger screen size provides more space to create a user-friendly interface that can display all required information without excessive scrolling or cluttering.
Typing Experience	Smaller touch targets for on-screen typing may lead to more user errors and a less comfortable typing experience.	Larger touch targets for on-screen typing can improve user accuracy and provide a more comfortable typing experience, reducing the likelihood of errors.
Calibration Precision	Users may encounter difficulties accurately dragging the corners of an image to the desired locations due to the smaller screen size.	The larger screen size allows for more precise dragging of image corners during calibration, potentially resulting in better calibration accuracy and a smoother user experience.
Information Display	Limited screen real estate may necessitate the use of scrolling, multiple pages, or smaller font sizes to display all information.	More screen space enables the display of more information at once, reducing the need for scrolling or multiple pages and allowing for larger, more readable font sizes.
UI Design Complexity	Designing a user interface that effectively accommodates all required elements and information may be more challenging.	The larger screen size provides more flexibility in UI design, making it easier to create an intuitive and visually appealing interface that incorporates all

		necessary components without appearing cluttered.
Portability	7-inch screens are more portable and easier to integrate into compact devices, which may be advantageous for certain use cases.	10-inch screens are less portable and may result in a larger overall device size, which could be a drawback if portability is a key concern.
Cost	7-inch screens are generally more affordable compared to larger screens.	10-inch screens are typically more expensive than 7-inch screens due to the larger display size and potentially higher resolution.

Table 3.2.6.2 *Monitor Size Comparison*

After carefully considering the advantages and disadvantages of both 7-inch and 10-inch capacitive touch screens, the authors concluded that a 10-inch monitor would be the most suitable choice for GPMS. While a 7-inch screen might offer better portability and lower costs, the benefits of a larger 10-inch display outweigh these factors in terms of user experience and functionality.

A 10-inch screen provides ample space to create a user-friendly interface that can display all necessary information without excessive scrolling or cluttering. This larger screen size allows for more precise dragging of image corners during calibration, potentially resulting in better calibration accuracy and a smoother user experience. Additionally, the increased screen real estate enables the display of more information at once, reducing the need for scrolling or multiple pages and allowing for larger, more readable font sizes.

Furthermore, the 10-inch screen offers more flexibility in UI design, making it easier to create an intuitive and visually appealing interface that incorporates all necessary components without appearing cluttered. The larger touch screen targets for on-screen typing which can also improve user accuracy and provide a more comfortable typing experience, reducing the likelihood of errors.

While a 10-inch screen may result in a slightly larger overall device size and higher costs compared to a 7-inch screen, the authors believed that the enhanced user experience and functionality justified these trade-offs. The improved precision, readability, and design flexibility offered by a 10-inch display will ultimately lead to a more effective and user-friendly GPMS device.

In conclusion, the authors determined that a 10-inch capacitive touch screen monitor was the optimal choice for the GPMS device, as it strikes the perfect balance between screen size, user experience, and functionality. This decision ensured that the device met the requirements of the intended use case while also providing a high-quality, user-friendly interface that enhances the overall effectiveness of the GPMS system. There were 3 options for 10-inch capacitive touch screens that the authors considered in the table below.

Feature	**GeeekPi 10.1” Capacitive Touchscreen**	**Waveshare 10.1inch HDMI LCD (H) with Capacitive Touch**	**Elecrow 10.1 Inch 1920x1080 HDMI TFT LCD Display with Touch Screen**
Screen Size (diagonal)	10.1 inches	10.1 inches	10.1 inches
Resolution	1024 x 600 pixels	1024 x 600 pixels	1920 x 1080 pixels
Touch Technology	Capacitive multi-touch	Capacitive multi-touch	Capacitive multi-touch
Connection Interface	HDMI for display, USB for touch	HDMI for display, USB for touch	HDMI for display, USB for touch
Compatibility	Compatible with Raspberry Pi 5	Compatible with Raspberry Pi 5	Compatible with Raspberry Pi 5
Dimensions	235mm x 142mm x 8mm	239mm x 169mm x 30mm	229mm x 149mm x 23mm
Power Supply	Powered via USB connection	Requires external power supply	Requires external power supply
Viewing Angle	178°(H) / 178°(V)	170°(H) / 170°(V)	85°(H) / 85°(V)
Price Range	\$80 - \$90	\$100 - \$120	\$120 - \$140

Table 3.2.6.3 *Selected Monitor Specifications Comparison*

After careful consideration of the three 10-inch capacitive touch screen options, the authors decided to proceed with the GeeekPi 10.1” Capacitive Touchscreen for the GPMS device. This decision was based on several key factors that aligned with the project's requirements and priorities. Firstly, the GeekPi display offers a high resolution of 1024 x 600 pixels, which ensures a clear and visually appealing user interface. Secondly, its wide viewing angles of 178° both horizontally and vertically allow for excellent visibility from various positions, making it suitable for use in different environments.

Additionally, the GeekPi display is compatible with multiple Raspberry Pi models, providing flexibility and future-proofing for the project. Most importantly, its power comes via USB, meaning it could be powered using the existing PSU also used for the Pi. Lastly, its affordable price point made it a cost-effective solution without compromising on essential features. While the other touchscreen options have their strengths, such as the Elecrow display's higher resolution and the Waveshare display's wide viewing angles, the GeeekPi 10.1” Capacitive Touchscreen offered the best balance of specifications, compatibility, and affordability, which made it the ideal choice for the GPMS device.

3.2.7 Generative AI Platforms

To effectively deploy AI models like Stable Diffusion, which demand high-quality GPU resources, the authors initially turned to Google Colaboratory due to its accessibility and cost-effectiveness. This cloud-based platform offers users a browser-based interface to develop and execute Python code. While Colab offers various plans, the authors’ underlying aim was to utilize its free option to save money. However, the authors quickly discovered that Stable Diffusion was heavily restricted on the free tier.

Subsequently, the authors opted for Colab Pro, priced at $10/month, which grants access to faster GPUs within a time limit. Despite this added benefit, the author’s efforts to run Stable Diffusion on Colab were impeded by newly identified bugs. Faced with these challenges, the authors explored alternative solutions, including procuring a more affordable GPU.

Fortunately, Advanced Micro Devices (AMD) provided the authors with access to a powerful machine directly from AMD. Recognizing the advantages of this new and improved setup, the authors decided to leverage remote access to this resource, enabling them to execute and test Stable Diffusion directly from their laptops without incurring additional expenses. The specifications of the machine are in the table below.

Component	Part Description
CPU	AMD Ryzen 9 7950X - 16-Core 4.5 GHz - Socket AM5 - 170W Processor
CPU Cooler	Noctua NH-D15 chromax.black 82.52 CFM CPU Cooler
GPU	SAPPHIRE NITRO Radeon RX 7900 XTX 24GB GDDR6 PCI Express 4.0 x16 ATX
MotherBoard	ASRock X670E Steel Legend AM5 ATX Motherboard
RAM	CORSAIR Vengeance RGB 96GB (2 x 48GB) DDR5 6000 Desktop Memory
Power Supply	Seasonic PRIME TX-1300, 1300W 80+ Titanium, Full Modular Power Supply
Case	Thermaltake Core P3 TG Pro ATX Mid Tower Case
Storage OS	Crucial T700 1 TB M.2-2280 PCIe 5.0 X4 NVME Solid State Drive
Storage APPS	SABRENT 2TB Rocket NVMe 4.0 Gen4 PCIe M.2

Table 3.2.7 *Computer Specifications*

This approach not only circumvented the limitations encountered with Google Colab but also allowed the authors to work on GPMS from any location. By leveraging existing resources, the authors were then able to streamline the workflow and focus on advancing their research without financial constraints. Most notably, the GPU used was the SAPPHIRE NITRO Radeon RX 7900 XTX 24GB GDDR6 PCI Express 4.0 x16 ATX.

3.2.8 Raspberry Pi 5 PSU Comparison

A deciding point that proved vital to GPMS, involved the authors extensively weighing the options between using a Boost Regulator on a custom PCB or using a Charger/Power Bank to power the Raspberry Pi 5. The table below compares both options accordingly. [18]

Feature	Regulator	Charger/Power Bank Comparison	AC to DC Adapter/ Wall Plug
Ease of Use	Requires design	Straightforward	Straightforward

	and assembly		
Power Regulation	Custom regulator needed, potential instability (Potential to blow)	Built-in regulation and protection	Built-in regulation and protection
Stability	Depends on design and battery quality	Highly stable and consistent	Highly stable and consistent
Risk of Damage	Higher, risk of incorrect wiring	Low, designed for power supply	Very Low
Cost	Much higher cost due to custom components, manufacturing, and turn around shipping time	Generally affordable and readily available	Most Affordable

Table 3.2.8 Regulator vs Charger/Power Bank Comparison

The first option was creating a custom PCB since it leveraged the use of the Boost Regulator. In fact, this was the initial route the authors took, where they designed a custom PCB that housed the Boost regulator capable of boosting the input voltage from double A Alkaline/Lithium batteries to the required 5V/3A input for the Raspberry Pi 5. Additionally, the regulator must be capable of handling the power requirements of the Raspberry Pi 5, which proved to be challenging over time with regards to meeting the Pi's high current draw. Battery technology, particularly Lithium batteries, may not always provide a consistent and sufficient power supply through a regulator, leading to potential power fluctuations and instability.

Moreover, when considering the first option for the Raspberry Pi 5's PSU, the authors' first thought was utilizing double A pure Lithium batteries due to their high energy density and compact size. However, this approach was halted immediately for additional reasons as there existed significant safety risks. Lithium-ion batteries, including double A pure Lithium batteries/cells, are extremely prone to overheating, which can lead to thermal runaway—a situation where the battery rapidly heats up and catches fire. This risk is heightened if the batteries are overcharged, short-circuited, or physically damaged.

Given these dangers, using double A pure Lithium batteries to power the Raspberry Pi 5 after passing voltage/current through a regulator were deemed not advisable by one of our faculty advisors. The potential for thermal runaway and the associated hazards made this option too risky. Furthermore, the pure Lithium batteries/cells are not permitted inside the Senior Design lab at the University of Central Florida. This ultimately made the decision for the authors to disregard option one and consider alternative solutions.

Option two was to use a power bank, which proved to be a more straightforward and reliable solution. Power banks are designed to provide a stable 5V output and can easily meet the 3A current requirement of the Raspberry Pi 5. They come with built-in protection circuits to ensure consistent power delivery and prevent issues such as overvoltage or short circuits. Additionally, power banks are portable, making them ideal for projects that require mobility. While battery technology has advanced, it is still challenging to achieve the same level of stability and reliability from combining double A Alkaline/Lithium batteries, a custom PCB, and a Boost Regulator compared to a commercially available power bank. Instead, using a power bank proved to be a safer and more reliable solution for powering the Raspberry Pi 5 and GPMS as a whole. However, this option proved bulky and superfluous.

However, another viable option emerged: using a 5V Wall Outlet AC Converter. This approach involves connecting the Raspberry Pi 5 directly to a standard wall outlet through an AC to DC converter that provides a stable 5V/3A output. The advantages of this method include uninterrupted power supply as long as the wall outlet is available, eliminating the concerns of battery depletion and the need for frequent recharging. Additionally, wall adapters typically offer better efficiency and longer-term reliability compared to battery-based solutions. They also remove the inherent safety risks associated with Lithium batteries, such as overheating and thermal runaway. Moreover, using an AC converter simplifies the power management system, reducing the overall complexity of the design.

In conclusion, after evaluating the options, the authors decided to utilize a 5V Wall Outlet AC Converter to power the Raspberry Pi 5 instead of relying on battery-based solutions. The AC converter provides a more stable, reliable, and safer power source, aligning better with the project's requirements and mitigating the risks associated with battery usage.

3.2.9 Raspberry Pi 5 Power Connection Type Comparison

The authors had various discussions about how to power the Raspberry Pi 5, specifically the two options which were to either use the USB-A to USB-C connection or the GPIO pins. The USB-A to USB-C connection is the most straightforward and recommended option by the Raspberry Pi manufacturers to power the Pi. This method involves using a standard USB-C power adapter that provides a stable 5V/3A supply, which is necessary for the Raspberry Pi 5 to function correctly. This method ensures that the power supply is regulated and consistent, reducing the risk of power-related issues. Additionally, using the

USB-C port is convenient as it is designed specifically for power input, which made this method a plug-and-play solution for GPMS.

On the other hand, powering the Raspberry Pi 5 through the GPIO pins involves connecting 5V-input to pins 2 or 4 and Ground to pin 6. While this method can be useful in certain scenarios, such as when space constraints prevent the use of a USB-C cable, it requires extremely careful handling to avoid directly damaging the development board. The GPIO pins are directly connected to the power rail which means that any mistake in wiring can lead to short circuits or overvoltage which would ultimately harm the Raspberry Pi 5. Moreover, the GPIO method lacks the built-in protection and regulation provided by the USB-C power adapter. The table below compares the two connection methods/options. [18]

Feature	USB-A to USB-C Connection	GPIO Pins
Ease of Use	Straightforward	Requires careful wiring and handling
Power Regulation	Built-in regulation and protection	No built-in regulation, large risk of damage
Stability	Highly stable and consistent	Depends on power source quality
Risk of Damage	Low, designed for power input	Higher, risk of short circuits or overvoltage

Table 3.2.9 Power Connection Type Comparison

Using the USB-A to USB-C connection is generally more efficient for powering the Raspberry Pi 5 because it provides a stable, regulated, and protected power supply. This method was designed specifically for the Raspberry Pi, ensuring that it receives the correct voltage and current without the risk of damage. The plug-and-play nature of the USB-C connection makes it convenient and user-friendly which actually reduces the complexity of the setup. In contrast, powering the Pi through the GPIO pins requires careful handling and lacks the built-in protections, making it a less reliable and more risky option. Therefore, the authors opted for the USB-A to USB-C connection given that it was the clear and safer choice for powering the Raspberry Pi 5 within the GPMS. [18]

3.2.10 Panseba Projector Power PSU Comparison

With regards to power, it is imperative that the authors' selected projector (Panseba) also receives enough power to operate, given that it requires 120V and operates at 60W. The first option was to power the projector by using a

Charger/Power Bank similar to the way the Raspberry Pi 5 itself would be powered. The second option would be to plug the projector into the wall. The table below compares both options. [18]

Feature	Wall Outlet	Charger/Power Bank
Power Supply Stability	Stable and Consistent	May vary, especially as battery depletes
Power Availability	Continuous, unlimited	Limited by battery capacity
Power Output	High, suitable for high-power devices	May not sustain high power for long
Usage Duration	Unlimited	Limited to battery life
Reliability	Highly reliable	Can be less reliable due to battery
Efficiency	Highly efficiency	May have inefficiencies and fluctuations
Cost	No additional cost	Additional varying cost of charger/power bank

Table 3.2.10 Panseba Projector PSU Comparison

When using a projector with the Raspberry Pi 5, the author's concluded that it was more efficient to plug the projector into a wall outlet for AC power rather than using a power bank. This is because wall outlets provide a stable and consistent power supply, which would prove to be crucial for the projector which requires a significant amount of power to operate efficiently. Projectors typically have higher power requirements compared to smaller devices like the Raspberry Pi 5, and power banks would not be able to sustain the necessary power levels for extended periods which could lead to interruptions and/or reduced performance.

Additionally, wall outlets provide continuous power without the risk of running out of battery supply which would ensure that the Panseba projector remains operational throughout the duration of GPMS without any unexpected shutdowns. Overall, plugging the projector into a wall outlet for AC power ensured a stable and reliable power supply, which was essential for the optimal performance of the projector, especially when interfacing with the Raspberry Pi 4B. [18]

3.3 Software Comparison and Selection

3.3.1 Operating System Comparison

Every computer/machine requires an Operating System to function effectively. Given that the machine that the authors were working with was brand new, selecting the appropriate OS became a crucial decision. Their familiarity with Linux prompted them to investigate whether there were compatible drivers for an AMD GPU, as Linux was their preferred choice. Below compares the OS options for the machine used in GPMS. [22]

Feature	Linux	Windows	Unix
AMD Processor Compatibility	Extensive support for AMD processors	Full support for AMD processors	Varies depending on Unix variant
Cost	Mostly free and open source	Paid licenses	Paid licenses
Ease of use	Requires command line knowledge	Requires command line knowledge	Requires command line knowledge
Customizability	Highly customizable	Limited customization	Highly customizability
Security	Generally considered secure	Secure with regular updates	Generally considered secure
Stability	Highly stable	Stable with regular updates	Highly stable
Stable Diffusion Compatibility	Highly compatible	Compatible with some setup	Varies depending on Unix variant
Required Dependencies	Easy to install dependencies	Requires manual installation of dependencies	Varies depending on Unix variant
Community Support for Stable Diffusion	Large community and resources for running Stable Diffusion	Smaller compared to linux	Limited community support
Community	Large and active	Large community	Smaller,

Support in general	community	and Microsoft support	specialized communities
Scalability	Highly scalable	Scalable with some limitations	Highly scalable
Cloud integrations	Well suited for cloud environments	Integrates with Azure cloud environments	Varies depending on Unix variant
Typical use cases	Web servers, databases, cloud computing	Active directory, exchange server, .NET applications	High performance computing, mission critical applications

Table 3.3.1 *Operating System Comparison*

After carefully analyzing the comparison chart, it became evident that Unix ranked last among the three operating system options. Its compatibility and support vary significantly depending on the specific Unix variant, making it a less suitable choice for *GMPS's* purposes. The decision ultimately came down to choosing between Windows and Linux.

While both operating systems have their strengths, Linux stood out in several key areas. Firstly, Linux is mostly free and open-source, providing a cost-effective solution. Secondly, it offers extensive compatibility with AMD processors and has a large community and resources dedicated to running Stable Diffusion. In contrast, Windows requires paid licenses and has a smaller community compared to Linux when it comes to Stable Diffusion support. Additionally, Linux's open-source nature allows for greater customization and flexibility. Considering the author's existing familiarity with Linux and the compelling features it offers, the authors confidently selected Linux as the operating system of choice for the project. [22]

3.3.2 Linux Distribution and Server Options

With confirmation of Linux, the next step was to determine which distribution, or *distro*, would best suit the project's needs. Linux distributions encompass variations of the Linux kernel bundled with additional software, serving different purposes such as general use, server hosting, software development, and IoT applications.

Given their intentions to remotely access the computer/machine to interface with Stable Diffusion, which essentially meant hosting a server, the authors focused on distros optimized for server hosting. Among the plethora of options — including Ubuntu, CentOS, Debian, and Fedora — they narrowed down their choices to both Ubuntu and CentOS, recognized for their suitability in this context. Below is a table comparing both. [23] [24]

Criterion	Ubuntu	CentOS
General Overview	User-friendly interface, designed for both general use and servers, with strong support for AI applications	Enterprise-grade stability and security, derived from RHEL, suitable for server hosting and secure environments
Intended Use	Optimized for server hosting, software development, and AI applications like Stable Diffusion	Primarily used for server hosting with an emphasis on stability and security
Driver Support	Compatible with AMD GPU, supports a wide range of hardware	Compatible with AMD GPU, prioritizes stability and may have more limited hardware support compared to Ubuntu
Update Frequency	Regular updates, including a Long-Term Support (LTS) version, though with slight delays for LTS	Provides LTS options with regular updates, but adopts new features at a slower pace than Ubuntu
Software Availability	Extensive range of available software, advantageous for diverse project requirements	More limited software range, focused on stability and security
Resource Consu-mption	Slightly higher resource consumption, which could impact performance	Generally lower resource consumption, optimizing stability and performance for server tasks
Community Support	Robust community support, beneficial for timely assistance and compatibility issues	Less extensive than Ubuntu, but still offers significant support through its association with RHEL

Table 3.3.2 *Linux & Server Comparison*

After carefully considering the various Linux distributions available, the authors concluded that Ubuntu was the most suitable choice for the GPMS project. This decision was reached after conducting a thorough comparison between Ubuntu and CentOS which are two popular distributions known for their server hosting capabilities and compatibility with AI applications like Stable Diffusion.

Ubuntu's user-friendly interface, coupled with its strong support for AI applications, made it an ideal candidate for the project's needs. Its optimization for server hosting, software development, and AI applications aligned perfectly with the goals of GPMS. Furthermore, Ubuntu's compatibility with AMD GPUs and its wide range of supported hardware ensured that the project could be implemented without facing significant driver-related issues.

One of the key factors that led to the selection of Ubuntu is its regular update cycle, which includes Long-Term Support (LTS) versions. Although there may be slight delays in the release of LTS updates, this approach would ensure that the project can benefit from the latest features and security patches while maintaining a stable and reliable environment. The extensive range of software available for Ubuntu was another significant advantage, as it allowed the project to adapt to diverse requirements and incorporate additional functionality as needed.

While CentOS is known for its enterprise-grade stability and security, its more limited software range and slower adoption of new features compared to Ubuntu made it less suitable for the GPMS project. The authors believed that the benefits of Ubuntu's robust community support outweighed the slightly higher resource consumption, as timely assistance and compatibility issues could be more easily addressed with the help of the large and active Ubuntu community.

The authors also recognized the importance of staying current with advancements in the field of AI and server hosting. Ubuntu's more frequent updates and feature additions aligned with this goal, which ensured that the GPMS project could continue to evolve and incorporate the latest developments in the field.

Therefore, after a thorough analysis of the available options, the authors determined that Ubuntu was the optimal Linux distribution for the GPMS project. Its combination of user-friendliness, strong support for AI applications, regular updates, extensive software availability, and robust community support made it the ideal choice to host the Stable Diffusion server and meet the project's diverse requirements. By selecting Ubuntu, the authors were confident that they could create a stable, secure, and feature-rich environment that would support the successful implementation and ongoing development of the GPMS project.

3.3.3 Local vs Cloud Server

Considering that a Raspberry Pi 5 is unable to run Stable Diffusion, the authors looked into server options. The purpose of a server would be to send data from the Raspberry Pi 5 to the server and back. There are two main options: a local server or a cloud server. The table below shows the comparison between the two. [25]

Features	Local	Cloud
Setup	Requires configuration of server software, wireless network, and additional setup for remote access tools (VPN and SSH)	Requires configuration of server software and remote access, but the wireless network is not applicable
Wireless connectivity	Requires setting up a wireless network for local communication, Wireless range limited by local network's coverage	Not applicable, the server is accessed over the internet
Latency	Low latency as communication occurs within the local wireless	Latency depends on internet connection speed and server location
Scalability	Limited scalability based on the AMD computer's hardware capabilities	Easily scalable
Maintenance	Requires physical access to AMD server for maintenance and updates	Requires remote server management or provider maintenance
Accessibility	Accessible within the range of local wireless network	Accessible from the internet anywhere
Cost	Initial cost of setting up AMD computer as a server	Recurring costs based on provider

Table 3.3.3 *Local vs Cloud Server*

Considering the fact that the AMD computer is on the school network, the authors believed it would require extensive paperwork to expose a port to be able to use the Cloud Server. The only option from there was to work locally accessing the server by 'sshing' into it.

More specifically, after careful consideration, the authors determined that setting up a local server would be the most suitable approach for their specific use case. The AMD computer, which will act as the server, is connected to the school network. Utilizing a cloud server would require exposing a port on the school network to enable communication between the Raspberry Pi 5 and the server. However, as mentioned before, this process would likely involve significant paperwork and administrative overhead due to the school's network policies and security considerations.

Given these constraints, the authors opted to work with a local server setup. The Raspberry Pi 5 would be to communicate with the AMD computer over the local wireless network. This local server approach simplified the setup process, reduced latency, and eliminated the need for exposing ports on the school network.

Although a local server setup had limitations in terms of scalability and accessibility compared to a cloud server, it provided a pragmatic solution given the constraints of the school network environment. The authors optimized the local server's performance by configuring the AMD computer's hardware and software stack to efficiently handle the AI image generation workload.

3.3.3 Front-End Framework

Developing a tablet application requires careful consideration of various factors, such as compatibility with the target platform (Raspberry Pi 5), ease of server communication, and seamless OpenCV integration. The team prioritized frameworks that offer straightforward OpenCV implementation, as it was crucial for the core functionality of the project to work without excessive additional effort. Other important capabilities included multi-touch support, performance, and ease of use. While the team is proficient in C++ and Python, the choice of programming language was not a limiting factor. To assist in the decision-making process, a comprehensive chart comparing various frameworks and libraries is presented below. [26] [18]

Library/ Framework	Programming Language	OpenCV Integration	Multi-touch Support	Performance	Ease of Use
PyQt / PySide	Python	Easy (Python bindings available)	Yes	Good	Moderate
Kivy	Python	Easy (Python bindings available)	Yes	High (OpenGL-based)	Easy to moderate

Electron	JavaScript (HTML/CSS)	Moderate (requires additional libraries)	Yes (with additional libraries)	Good	Easy (for web developers)
Flutter	Dart	Moderate (requires additional libraries)	Yes	High (natively compiled)	Moderate
Qt	C++	Easy (built-in support)	Yes	High	Moderate to complex

Table 3.3.4 *Library/Framework Comparison*

‘Qt’ stands out from the other frameworks due to its built-in support for OpenCV. This native integration made it highly convenient to incorporate OpenCV functionality into Qt applications without the need for additional bindings or libraries. Qt's extensive documentation and resources further facilitated the implementation process.

In the context of OpenCV integration, bindings refer to the software layer that enables communication between the programming language and the OpenCV library. Bindings allow developers to access OpenCV functions and data structures from within their chosen programming language. For example, PyQt/PySide and Kivy benefit from the availability of Python bindings for OpenCV (OpenCV-Python), which makes it easy to utilize OpenCV in Python-based applications. While Qt and Kivy share many similarities, they differ in their programming languages. Qt is built on C++, known for its performance and low-level control, while Kivy is based on Python, renowned for its simplicity and rapid development capabilities. Although Kivy excels in creating visually stunning and interactive user interfaces, Qt has a larger and more vibrant community. This extensive community support is highly advantageous, providing access help if issues arise.

After careful consideration, the authors decided to use Qt as their UI framework. The combination of Qt's native OpenCV support and extensive community resources made it the most suitable choice for their project. By leveraging Qt's strengths, the authors were then able to efficiently develop a robust and feature-rich application while also benefiting from the framework's performance, cross-platform capabilities, and the vast ecosystem of tools and libraries available within the Qt community.

Chapter 4 Standards and Design Constraints

4.1 Standards

4.1.1 HDMI

High Definition Multimedia Interface (HDMI) is a standard developed by a group of electronics manufacturers designed to facilitate high-bandwidth links among digital devices. Properly configured, HDMI supports 1080p high-definition video and up to eight channels of uncompressed audio, suitable for a 7.1 surround sound setup. HDMI seamlessly simplifies the connection process by minimizing the amount of cables needed and can also decrease the number of remote controls necessary for display use. In order for all users to fully utilize all HDMI has to offer, all components in their desired system must be compatible with HDMI. [27]

A HDMI port is one of the features the Epson projector offers. This port is used to receive and transmit audio and video signals from a variety of sources. HDMI is used to send High-Definition signals over a single cable. It can be used to transmit audio and video over a cable. The benefit of HDMI is that it can transfer a high bandwidth of data via a single cable. [27]

A common misconception about HDMI and digital signals is that they are both inherently superior to analog signals due to the absence of analog-to-digital conversion, which keeps the signal in a pure and unhindered state. However, HDMI signal transmission involves encoding through Transition Minimized Differential Signaling (TMDS) to maintain consistent signal integrity over the cable length. TMDS encoding reduces signal degradation by minimizing transitions between binary states and using a twisted pair cable setup where one cable carries the signal and the other an inverse copy. As a result, this allows the receiving device to decode and compensate for any loss. [27]

Furthermore, HDMI incorporates High-Bandwidth Digital Copy Protection (HDCP) to prevent piracy. HDCP is an authentication protocol between devices, using a “handshake” process to establish a secure connection. The “handshake” ensures that only authorized devices can decode the transmitted data, with continuous checks to maintain security. The Federal Communications Commission (FCC) mandates content protection in the U.S., requiring HDMI-compatible devices to support HDCP. [27]

One common issue that users run into is a “handshake” failure where the display system does not have HDCP support. Another issue that users run into is the limit to how long an HDMI cable can be. The HDMI standard requires a minimum length of 32 feet of cable; however, it is common for many users, including the authors, to use an HDMI cable of much less length for a variety of applications. [27]

4.1.2 Transmission Control Protocol (TCP)

As part of the authors' Senior Design project, the authors integrated Transmission Control Protocol (TCP) to establish a reliable client-server connection. This connection is crucial for relaying packets from the user device to the server, where the generative AI pipeline processes the data and then back to the projector. The decision to use TCP stemmed from its unparalleled reliability and orderliness in data transmission, which were critical for the seamless operation of the project.

TCP is extremely fundamental to Internet communications, ensuring that data packets between networked devices are delivered accurately and also in order. In the authors' project, TCP's role was indispensable. It facilitated a dependable conduit for the data exchanges between the server's AI pipeline and the user's device. Several features of TCP were particularly beneficial for the success of GPMS. The protocol's error-checking capabilities ensured that any data corrupted during transmission would be identified and retransmitted. This guaranteed the integrity of the data received by the server and, subsequently, by the projector, ensured that the AI-generated content was displayed as intended.

Moreover, TCP's sequence control is vital for maintaining the order of data packets. This was crucial for the authors because any potential mis-sequence in the AI-generated data could lead to incorrect or nonsensical visual outputs. Additionally, TCP's flow control mechanism helps prevent network congestion, which is essential for maintaining a smooth and responsive interaction between the user device and the server, to cut down on wait time during generation.

Diving deeper into the technical aspects of the Transmission Control Protocol (TCP) illuminated how its sophisticated mechanisms were instrumental for the project. One of TCP's standout features is its use of a three-way handshake to establish a connection. This process involves an exchange of 'SYN' (synchronize) and 'ACK' (acknowledge) messages between the client and server. This handshake ensured that both the sender and receiver would be ready for data transmission which would establish a reliable channel. For the authors' project, this meant that they must create a stable pathway for data to flow between the user device and their server, which was extremely crucial for processing in the generative AI pipeline.

Furthermore, TCP's congestion control algorithms are a cornerstone of its ability to manage data flow efficiently across networks. These algorithms, such as Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery, dynamically adjust the rate of data transmission based on the network's capacity to handle traffic. This adaptability prevents packet loss and minimizes latency, which was paramount for GPMS.

The authors' generative AI pipeline relied on the timely and accurate reception of data to generate and project visuals without delay or disruption. By leveraging TCP's congestion control, the authors ensured that the system remained responsive and efficient, even under varying network conditions, providing a seamless experience for all users.

Implementing TCP in the authors' architecture required careful consideration. The authors had to establish robust TCP connections that could handle the intensive data exchange required by their AI applications. This included managing TCP sessions efficiently to ensure a stable and continuous flow of information.

Despite TCP's advantages, the authors anticipated challenges, particularly in minimizing latency and maintaining real-time responsiveness. Those were critical aspects that were addressed through system design optimizations and through the leverage of TCP's capabilities to their fullest. This goal ensured that the authors' project delivered a seamless and engaging experience powered by a reliable and efficient backend communication system.

4.1.3 Secure Sockets Layer (SSL)

Integrating Secure Sockets Layer (SSL) is a fundamental step for establishing a secure HTTPS connection between the client and server within the Generative Processing Management System (GPMS). This integration is critical for safeguarding the communication channels, ensuring data integrity and confidentiality during transmission. By implementing a self-signed SSL certificate, sensitive information, such as API keys and user data, is protected from potential interception or unauthorized access, particularly when operating on UCF's network, which may be vulnerable to malicious actors. [28] [29]

SSL provides a robust encryption framework, creating a secure channel over an inherently insecure network. This ensures that all transmitted data remains encrypted, rendering it unreadable to potential eavesdroppers. API keys and other critical data are safeguarded against exposure or tampering, which is paramount for maintaining the privacy and security of the system. The encryption not only secures data in transit but also establishes a trust layer, significantly reducing the risks associated with man-in-the-middle attacks or unauthorized data access. [28] [29]

Moreover, SSL supports authentication mechanisms that validate the identity of the server through certificates, ensuring that data exchanges occur exclusively with trusted and intended servers. This prevents malicious actors from impersonating the server and compromising the system. The use of HTTPS, underpinned by SSL, allows GPMS to handle various data streams securely, ensuring the integrity and confidentiality of the AI pipeline's operations. [28] [29]

By integrating SSL, GPMS achieves a higher standard of security, which is crucial for protecting sensitive data, enabling reliable communication, and ensuring safe AI-driven processing. This robust security framework not only enhances privacy but also reinforces trust in the system's ability to operate efficiently and safely in environments that require stringent security measures. [28] [29]

4.1.4 Projector

Projectors come in a variety of shapes and sizes, but they all adhere to a few key standards. To begin with, it was necessary to discuss resolutions. Projector resolution refers to the number of pixels that can be displayed on the screen. Common resolutions include SVGA (800x600), XGA (1024x768), WXGA (1280x768), and Full HD (1920x1080). The relationship between resolution and detail is directly proportional. Higher resolutions provide sharper and more detailed images, and lower resolutions provide less sharp and less detailed images.

Firstly, there exist standards that define supported connection options like HDMI, VGA, and USB for connecting the projector to devices. Secondly, lamp life which refers to the average number of hours the projector lamp can last before needing replacement. Knowing the lamp's life helps estimate ongoing costs associated with projector usage. Thirdly, the throw ratio determines the distance required between the projector and the screen to achieve the desired image size. Understanding throw ratio is crucial for setting up the projector in a room of specific dimensions. Furthermore, projector standards also encompass ‘color gamut’, which refers to the range of colors a projector can display. This is particularly important for applications, such as GPMS, that facilitate photo projections/presentations.

Additionally, aspect ratio is crucial. The aspect ratio is the width of the projected image compared to its height. Common aspect ratios include 4:3 (standard definition) and 16:9 (widescreen). The aspect ratio of the projector should match the aspect ratio of the content users are projecting. Note: lumens encompass a critical detail to the overall standard regarding projectors. Lumens is a unit that measures the brightness of a projector. Projectors with higher lumens ratings are better suited for use in brightly lit rooms. Given just a few of the key projector standards, users can first understand these standards and then choose a projector that best meets their needs for brightness, resolution, and compatibility.

Projectors adhere to several key standards to ensure optimal performance and compatibility. Resolution, which determines the level of detail, comes in various common formats. The aspect ratio, representing the width-to-height ratio of the projected image, should match the content being displayed. Lumens measure brightness, with higher ratings suitable for brighter environments. Beyond these standards, connectivity options such as HDMI, lamp life indicating replacement frequency, throw ratio for optimal placement, and color gamut for accurate color representation are also essential considerations. By understanding and evaluating these standards, users can select projectors that best suit their specific needs and preferences. [30]

4.1.5 Camera

There are two main standards for cameras. Measurement Standards define how various aspects of camera performance are measured. They are set by organizations like the International Organization for Standardization (ISO) and ensure consistency in how camera manufacturers report things like resolution, noise, and sensitivity. Environmental Standards define how rugged a camera is in terms of dust, water, and impact resistance. They are important for understanding how well a camera will withstand different environmental conditions. After understanding these standards, the authors became equipped to compare cameras and choose the one that best met their needs.

The ISO 12233 standard defines how a camera's resolution, often megapixels (MP), is measured. It specifies using a chart with specific patterns to test the camera's ability to capture fine details. It is important to note that a higher MP does not always equate to better image quality, but it does indeed allow larger prints or cropping without losing detail.

The ISO 15739 standard defines image noise which refers to unwanted grain or speckles that appear in photos, especially and oftentimes in low-light situations. This standard defines how to measure noise levels, allowing comparison between different types of cameras. In fact, lower noise levels indicate a cleaner image.

The ISO 12232 refers to the ISO Speed Rating. This standard defines how a camera's ISO speed, which determines its sensitivity to light, is measured and reported. Higher ISO allows for capturing images in darker environments but can also introduce more noise. This standard ensures consistent reporting across camera brands.

The ISO 14524 standard, the tone curve, defines how a camera translates light captured by the sensor into brightness levels in the image file. The ISO 17850, geometric distortion, measures how straight lines appear curved in the final image due to lens imperfections. The ISO 18844, image flare, defines how a camera handles unwanted light sources appearing as streaks or halos in the image.

The IK Rating, referring to impact resistance, is an international standard (IEC 62262) that uses a two-digit code to rate a camera's resistance to physical impacts from accidental drops or bumps. Higher numbers indicate greater impact resistance. The IP rating, referring to Ingress Protection, IP ratings (IEC 60529) define a camera's resistance to dust and water ingress. The first digit indicates dust protection, and the second digit represents water resistance. For example, IPX4 signifies protection against splashing water. The NEMA Standards (designated to North America) define enclosure ratings for electrical equipment, including cameras. Similar to IP ratings, they use a code to indicate dust and water resistance levels.

In conclusion, camera standards act as a common language for understanding a camera's capabilities. Measurement standards (ISO) ensure clear communication regarding a camera's resolution, low-light performance, and other key aspects of image quality. Environmental standards (IK rating and NEMA) inform one about the camera's durability in harsh conditions. By understanding these standards, the authors were well-informed to select a camera that best suited their needs in the context of GPMS. [31]

4.2 Constraints

4.2.1 Economic

The economic constraints of the GPMS project primarily revolve around the cost of the hardware components and the available budget. The total estimated cost for the project is approximately $3,544.37, as detailed in Section 10.1. However, thanks to component donations, the out-of-pocket expenses for the authors were reduced to roughly $299.93.

Despite these donations, the authors recognize that additional funding could have enhanced the project's capabilities. For instance, if the budget allowed, the authors would have opted for an NVIDIA GPU instead of the AMD GPU currently being used. NVIDIA GPUs are widely known for their superior performance in deep learning and AI applications in comparison to AMD GPUs, which could have certainly improved the image generation quality and processing speed of the GPMS.

Furthermore, a higher budget would have enabled the authors to acquire a high-end projector capable of projecting images at greater distances with increased visibility. Such a projector would have expanded the potential applications of the GPMS, allowing for larger-scale projections in various settings.

While the current budget and donated components allowed the authors to develop a functional prototype, they acknowledge that additional financial resources could have elevated the project's performance and versatility. Careful financial planning and management was crucial to successfully complete the GPMS project within the given economic constraints while striving to maximize its capabilities.

4.2.2 Time

The GPMS project was subject to the time constraints imposed by the Senior Design joint course schedule, which presented several challenges for the project team. The entirety of the GPMS project was divided into two courses/phases: Senior Design 1 (EEL4914) and Senior Design 2 (EEL 4915L), each with its own set of deadlines and deliverables.

Senior Design 1 focused on completing all administrative tasks, such as project planning, research, and documentation. The authors set a series of deadlines (detailed in Chapter 10.2.1) to ensure steady progress throughout this phase. Key milestones include completing the 45-page document by March 10, 2024 and submitting the 90-page document by April 23, 2024. These deadlines were crucial for laying the foundation of the project and ensuring that all necessary research and planning were completed before moving on to the implementation phase.

However, the team faced additional challenges that impacted their ability to meet the deadlines. Two of the team members had limited experience with C++ programming, which was a critical skill required for developing the software components of the GPMS project. This knowledge gap did not slow down the coding process, as the team members just needed to spend extra time learning and familiarize themselves with the language and its best practices. To mitigate this challenge, the team needed to allocate additional time for training and peer support, ensuring that all members were equipped with the necessary skills to contribute effectively to the project.

Senior Design 2 was dedicated to prototyping, testing, and debugging the entire project. The authors planned to complete the prototype and testing of the entire project by October 20, 2024. The final demonstration was scheduled for November 20th, 2024. These milestones required significant effort and coordination from the team, as they involved integrating various hardware and software components, conducted rigorous testing, and addressed any issues that may arise during the process.

To meet these deadlines and effectively manage time constraints, the authors adhered to a strict schedule, regularly communicated progress, and promptly addressed any issues that arose. Effective time management and collaboration among team members was essential to ensure that the project was completed within the allotted time frame.

4.2.3 Ethics

Ethical constraints existed in the development of GPMS. Generative AI, a key component to GPMS, has transformative potential and ethical challenges posed in image generation. On the positive side, AI image generators like DALL-E, Midjourney, and Stable Diffusion enable ease of creative thought and make digital art/design accessible to a broader audience regardless of any skill level.

This technology enables rapid iteration of images, which in turn reduces the time and cost associated with traditional methods of image content creation. From an environmental point of view, relying on AI for image generation could potentially reduce the carbon footprint associated with the physical aspects of artistic production. The use of AI image generation fosters innovation by offering new avenues for visual media. [32]

However, significant ethical concerns surround AI image generation for a number of reasons. Since AI tools, such as ‘ChatGPT-4’, often use large datasets of images without originality metrics, the risk of copyright infringement and plagiarism was a major issue. This had potential to lead to the creation of works that infringe on the rights of original creators, posing potential legal disputes. Additionally, AI-generation images may even display biases and stereotypes based on user inputs.

This greatly emphasized the need for careful consideration of the social implications of these image-generation technologies. There is a prevalent and vital call for ethical guidelines and legal clarity to address the complex landscape of AI image generation. This is where the need for a balanced approach that respects both the innovative potential of tools and the rights/contributions of human creators. [32]

In the context of GPMS, the user input must always prohibit inappropriate and misleading images. This was because the authors desired GPMS to be a product for all audiences. With the previous measures in place, the integrity of GPMS was ensured and intact, especially for family-friendly environments.

4.2.4 Security/Network Integration

During the research phase of the project, the team explored various methods for establishing communication between the Raspberry Pi 5 and the AMD computer. The goal was to find an efficient and secure way to transfer data and execute computationally intensive tasks on the more powerful AMD machine. However, several constraints and challenges emerged throughout the process.

One of the primary hurdles encountered was the limited access to the AMD computer. The team collaborated with the DRACO Lab team who provided assistance in granting access to the machine. However, the process experienced significant delays due to unforeseen issues and administrative complexities regarding UCF IT. These hindered the team's progress and raised concerns about the feasibility of the proposed communication system, specifically the hopes of exposing a port on the network to simplify API calls.

A major constraint that came to light was the restricted accessibility of the AMD computer. The machine could only be accessed remotely through SSH while connected to the school network. This limitation posed a significant challenge, as it required team members to be physically present on campus to establish a connection. Although this issue could be mitigated by using ‘Cisco AnyConnect’, a VPN (Virtual Private Network) solution employed in other areas to access school resources remotely, it introduced additional complexity.

The most critical issue, however, stemmed from the inability to make conventional API calls to the AMD computer. In order to facilitate seamless communication and data exchange, the team initially planned to utilize API endpoints. However, this approach required opening a specific port on the school network to listen for incoming requests. Given the strict security policies and administrative regulations in place, the authors realized that obtaining permission to open a dedicated port would be a formidable challenge.

The school's IT department and administration maintain stringent control over network access and security measures. The lengthy process of acquiring SSH permission for the project served as a clear indication of the level of scrutiny and caution exercised by the institution. Consequently, the team concluded that securing approval to open a port for API communication would be highly unlikely, considering the time-consuming nature of the SSH permission process alone.

This constraint forced the authors to reevaluate their approach and explore alternative solutions for enabling communication between the Raspberry Pi 5 and the AMD computer. The inability to rely on standard API calls necessitated the development of creative workarounds and unconventional methods to facilitate data transfer and remote execution of computationally intensive tasks.

In the following sections of this documentation, the authors delved into these alternative methods and workarounds that they have explored to overcome the constraints associated with the machine. Those solutions provided a secure and efficient means of data transfer and remote execution while respecting the school's network policies and ensuring the integrity of the project. By thoroughly examining the constraints related to the AMD computer, the team gained a deeper understanding of the complexities involved in integrating external resources into their project.

Chapter 5 Comparison of ChatGPT with other Similar Platforms

5.1 Comparison Overview

In recent years, generative AI platforms like ‘ChatGPT’, ‘Microsoft's Copilot’, and ‘Gemini’ have revolutionized how individuals and organizations approach problem-solving, creativity, and learning. These platforms, through their advanced algorithms and vast databases, have the capability to assist and guide ambitious individuals, such as the authors, in the development of complex projects.

These platforms can directly impact Senior Design Engineering capstone projects across the country, including GPMS at UCF. This chapter aims to highlight and compare each of these platforms, focusing on their limitations, pros and cons, and impact on learning experiences within the context of the authors’ experience creating GPMS.

5.1.1 ChatGPT

ChatGPT is a free AI-powered natural language processing tool that directly enables human-like interactions and offers various capabilities with the chatbot. This language model can respond to inquiries and help with tasks, including writing emails, essays, and code. [33]

ChatGPT, developed by OpenAI, is widely known by technology users for its conversational abilities, making it an excellent tool for generating ideas. That being said, it is also helpful in suggesting code optimizations and alternative solutions for problems. Creating content is also a lucrative feature regarding ChatGPT's capabilities ever since its newest paid subscription model, ChatGPT-4, was released. This newest model is capable of generating AI-generated images in seconds. [33]

In the GPMS research process, the authors pondered whether to use a basic search engine to begin the research process or turn directly to AI. It was paramount to understand the differences before taking the next steps. The authors concluded that ChatGPT was designed to engage users in dialogue, whereas search engines, like Google index web pages, to deliver requested information. Each serves a distinct purpose, making them incomparable in terms of utility. [33]

However, search engines are preferred for the latest and most accurate information. That is because most ChatGPT users opt for the basic free version of ChatGPT, known as ChatGPT-3.5, which lacks internet search capabilities, relying instead on pre-existing knowledge from its training data, which may introduce inaccuracies. [33]

ChatGPT's free version only has knowledge up to 2021, limiting its ability to answer questions about events occurring after that year. On the other hand, search engineers like Google offer the most recent information. ChatGPT-4 subscribers benefit from Bing integration which gives the chatbot internet access. Moreover, the internet access enhances ChatGPT's functionality with web indexing while also maintaining its capability to process natural language queries and generate conversational responses. [33]

Given ChatGPT's ability to explain complex topics in a simplified manner, it was the most suitable option for GPMS research. Furthermore, while ChatGPT was extremely helpful, it revealed several limitations. Users may need to rephrase questions multiple times to convey their intent clearly, and the quality of responses can vary. In fact, in an attempt to facilitate a conversation regarding GPMS, this platform tended to make assumptions, leading to misinterpretations. [33]

A significant limitation in the free version of ChatGPT is the knowledge cutoff in 2021. This gap leaves the platform needing to be made aware of more recent events or developments. Additionally, the free version of ChatGPT initially did not cite sources for its information. However, this has been partially addressed by developers for ChatGPT-4 users, given that the newer version can aid in proper citation generation when needed. [33]

ChatGPT-4 was imperative and crucial in generating concise and precise tables for the GPMS technology comparison section of this document. ChatGPT-3.5, the free version, was not capable of generating tables the way the paid version did. More specifically and notably, tables were only generated via ChatGPT-4 using paragraphs that contained detailed explanations and comparisons of at least three technology parts.

The tables returned from ChatGPT-4 contained a complete and concise overview that quickly ran through a set of factors that pertained to the specific technology part at hand, unlike the tables that ChatGPT-3.5 returned. This comprehensive capability saved plenty of time from manually converting pre-existing comparison content into concise tables comparing the three technology parts. Instead of spending time on manual table generation, more time was redirected to performing adjustments and polishing the AI generated tables.

5.1.2 Copilot

Microsoft has been a key player in generative AI, which aims to produce content from user input. It has added AI capabilities to Bing and introduced an early version of the Copilot AI assistant in Windows 11, designed to boost both creativity and productivity.

Copilot, initially known as Bing Chat, is Microsoft's advanced AI chatbot integrated into its search engine and is powered by a technology more sophisticated than OpenAI's GPT-4. It was unveiled in early 2022 and later rebranded during the Microsoft Ignite event.

Copilot can assist users with various tasks, including creative writing, solving math problems, coding, and generating images. It stands out because it can provide up-to-date information from the web with links, which was extremely helpful to the authors in researching various parts of GPMS. In fact, Copilot has a feature that allows users to choose a conversation style such as "More Creative, "More Balanced," or "More Precise." These styles bring character limits. The "More Creative" and "More Precise" styles allow 4000 characters in total text input while the "More Balanced" style only allows a total of 2000 characters in total text input. [34]

Additionally, Copilot can accept both text and image inputs. This was essential to developing the ideal cover image of GPMS documentation. It is interesting to note that Microsoft's Copilot had an autocomplete feature with text inputs, which was extremely helpful in brainstorming specific adjectives to describe the desired image. A comparison between a ChatGPT image and a Copilot image resulted in utilizing the ChatGPT-generated image for simplicity purposes. [34]

Microsoft also uses the Copilot branding for its range of AI assistants tailored to specific services, like ‘Copilot for Windows’, ‘Copilot for Microsoft 365’, and more. The tool will eventually receive new features such as GPT-4 Turbo integration, enhanced search capabilities, and Code Interpreter. [34]

Copilot is accessible via its direct website to anyone with a Microsoft account, and its use cases are diverse, including functioning as a search engine, content creator, and learning tool from uploaded images. A Copilot app is available for mobile devices and also offers a premium version called ‘Copilot Pro’ which boasts expanded features for a subscription fee. [34]

Microsoft, an early investor in OpenAI, has incorporated ChatGPT into Bing, leveraging their partnership. Microsoft's Azure is OpenAI's exclusive cloud-computing provider. Initial controversies around the AI chatbot's behavior led to the implementation of chat limits.

Despite these issues, Copilot has been rated highly for its capabilities, including access to GPT-4 and internet connectivity. Copilot solved some of the significant problems ChatGPT faces, such as including knowledge of current events via internet access and providing footnotes with links to sources from the information it pulled. [34] Additionally, Microsoft offers Bing Chat Enterprise, now called Copilot, as a secure workplace chatbot solution, initially provided at no extra cost to certain Microsoft 365 account holders, with a planned future fee.

5.1.3 Gemini

Gemini, formerly known as Google Bard, is Google's conversational AI chatbot designed to rival ChatGPT. Unlike ChatGPT, which relies on information up to 2021, Gemini extracts real-time data from the web. This in itself is a massive reason why Gemini is beneficial for research. It can code, solve math problems, assist with writing, and, starting February 2024, generate images using ‘Google's Imagen 2’ model. [21]

Originally launched as Bard and based on a lightweight version of ‘LaMDA’, Gemini has seen significant upgrades. Initially, it utilized Google's Transformer neural network architecture but later transitioned to ‘PaLM 2’ for enhanced performance. In December 2023, Google introduced an advanced version of Gemini, leveraging a fine-tuned variant of Gemini Pro for English, marking it as Google's most capable Language Learning Model. [21]

Gemini is publicly available without a waitlist. It supports multimodal search, integrating Google Lens to allow users to input images alongside text for queries. This feature enables users to receive information about images they upload, such as identifying plants and even breeds of pets. [21]

Additionally, Gemini can include images in its responses for queries that benefit from visual aids. With the introduction of image generation capabilities, users can request Gemini to create images based on detailed or minimal descriptions, showcasing Google's advancements in AI-driven creativity. [21] Gemini represents Google's ambitious direction into conversational AI, offering real-time web information retrieval, multimodal searches, and advanced image generation powered by its continually evolving large language models. [21]

So far, Gemini has been an extreme aid in translating existing technology part comparisons into tables to best visualize each given part's pros and cons. Gemini aided much more substantially in generating tables derived from pre-existing paragraphs describing technology components compared to the ChatGPT-3.5 Version. This is because Gemini provided a comprehensive overview that contained sources for the information of each technology part. Compared alongside ChatGPT-4, Gemini equally matched the quality of the table generation with regard to accuracy and source provision.

5.1.4 Overall Comparison

The table below provides a comparison of various aspects of ChatGPT, Copilot, and Gemini. Each of their functionalities, features, strengths, weaknesses, and other relevant details are highlighted.

Aspect	**ChatGPT**	**Copilot**	**Gemini**
Core Functionality	Natural language processing, conversation, idea generation	AI assistant, creativity booster, productivity enhancer	Conversational AI, real-time web data extraction
Developer	OpenAI	Microsoft	Google
Internet Access	Limited (ChatGPT-3.5 lacks internet access)	Yes (provides up-to-date information with links)	Yes (extracts real-time data from the web)
Subscription Model	Free (premium version available for a subscription fee)	Free (premium version available for a subscription fee)	Free
Image Generation	Yes (Only ChatGPT-4 can generate AI-generated images)	Yes (supports both text and image inputs)	Yes (supports image generation capabilities)
Collaboration	Limited	Yes (offers Copilot for Microsoft 365)	Not specified
Multimodal Support	No	Yes (accepts both text and image inputs)	Yes (supports Google Lens for multimodal search)
Search Capabilities	Limited to pre-existing knowledge	Enhanced (access to GPT-4 and internet connectivity)	Real-time web data extraction
Integration	Standalone tool	Integrated into Bing and Microsoft ecosystem	Integrated into Google ecosystem
Table Generation	Limited (ChatGPT-3.5 lacks quality table generation	Yes (Copilot Pro offers advanced features)	Yes (supports comprehensive table generation)

	capability)		
Knowledge Cutoff	Limited to 2021	Provides up-to-date information from the web	Real-time web data extraction
Strengths	Conversational abilities, code suggestions, idea generation	Up-to-date information, internet connectivity, code assistance	Real-time web data extraction, image generation
Weaknesses	Limited internet access, outdated knowledge	Initial controversies, chat limits, subscription model	None experienced

Table 5.1.4 *ChatGPT vs Copilot vs Gemini*

The table above presents a comprehensive comparison of three leading generative AI platforms: ChatGPT, Copilot, and Gemini. Despite each of their differences, each specific platform offers unique strengths, from conversation and code suggestions to real-time web data extraction and advanced image generation.

5.2 Limitations, Pros, and Cons

ChatGPT unfortunately generates inaccurate information at times and proves to struggle with highly specialized or niche topics. In order to use ChatGPT, users must create an account and then login before accessing any of the services that ChatGPT's free and/or paid version provide.

The main pro of ChatGPT is the fact that it is highly versatile in content creation; excellent at understanding and generating human-like text. However, its main con is that it is limited real-time data access; potential for generating incorrect information.

Microsoft Copilot is limited primarily to the Microsoft ecosystem. Furthermore it may not always understand the context or the specificity of user requests outside of its integrated applications. Microsoft Copilot requires all of its users to first create an account and then login before accessing some of the services that it provides such as image generation.

The main pro of Microsoft Copilot is its deep integration with Microsoft tools; real-time assistance within familiar applications. Despite this pro, it is important to know that the main con is that it is less versatile outside of the Microsoft ecosystem; potential for dependency on specific software.

Google Gemini’s reliance on web-based information can lead to biases or inaccuracies. Furthermore, it may not provide as personalized or context-aware responses as other platforms. Gemini requires all of its users to first create an account and then login before accessing any of the services that it provides.

The main pro of Google Gemini is that it has a plethora of access to a vast range of information; innovative in generating insights based on current data. A quite noticeable drawback is that it does have potential for bias in information, which in return gives less focus on generative creativity in direct comparison to conversational understanding.

5.3 Impact on Senior Design Engineering Capstone Projects

The platforms above cover various areas that directly impacted the refinement of GPMS. Through ChatGPT, multiple ideas for the authors' Senior Design Capstone Project were generated. Moreover, ChatGPT began the authors' brainstorming process. This widely-known platform also aided in suggestions based on similar projects and current trends.

Regarding Microsoft Copilot, the authors experienced significant benefits regarding software integration by offering real-time coding assistance and debugging tips. It also provided PSU recommendations regarding using the most optimal approach to kickstart the PSU development/integration process.

Gemini aided in conducting thorough research by pulling in the latest studies, articles, and data from across the Internet, ensuring that GPMS was grounded in up-to-date information with supporting documentation when necessary. This was especially important given that it was essential to the authors that the project was in the realm of possibilities and could realistically be completed.

Although beneficial in most cases, these platforms have the potential to unintentionally impose harmful effects. These effects include, but are not limited to, overreliance, data privacy concerns, and skill atrophy. An overreliance on the AI platforms above can hinder the development of personal critical thinking and problem-solving skills among a group. Additionally, group collaboration has the potential to decrease in the presence of AI use. Fortunately, AI only improved the consistency of the authors’ collaboration.

Using the platforms for GPMS or any other project development could raise concerns regarding sensitive data privacy. This is because when sensitive or proprietary information is involved, it raises concerns on whether authors should be inputting their project-specific information into any select platform. Regarding skill deterioration, the convenience of AI assistance could lead to a decline in one's ability to perform time-consuming tasks independently, such as manual coding or original content creation. Fortunately, the authors did not run into these select problems.

5.4 Concluding Remarks

ChatGPT, Microsoft Copilot, and Google Gemini each offer unique advantages and face distinct limitations. Their impact on the authors' Senior Design Engineering capstone project, GPMS, was profound. It offered tools for innovation, efficiency, and creativity. However, it was essential for each group member to be mindful of the potential drawbacks, such as overreliance and the risk of diminishing critical thinking skills that can otherwise benefit the group.

By leveraging these platforms, each group member enhanced their learning experiences while fostering a balance between AI-assisted efficiency and the development of independent, critical Engineering skills required for the overall success of GPMS's implementation.

Chapter 6 Hardware Design

6.1 Hardware Enclosure

The hardware design of the GPMS included a 3D-printed PLA enclosure designed to sit atop the projector, secured with Velcro. This enclosure houses the touchscreen monitor, Raspberry Pi 5, and camera, with careful allowances for minor variations in printed dimensions. The touchscreen monitor is embedded seamlessly into the enclosure's top, similar to a smartphone screen, while the Raspberry Pi 5 is Velcro-mounted to the enclosure floor to prevent unintended movement. The camera was affixed to the enclosure's front face using screws and nuts, positioned parallel on the same plane and above the projector lens to maintain alignment for consistent capture and projection on the same target area.

Additionally, this enclosure concealed cables connected to the touchscreen, reducing cable clutter and enhancing user experience with the GPMS. It includes a side opening for cables from other external hardware to connect internally to the Pi.

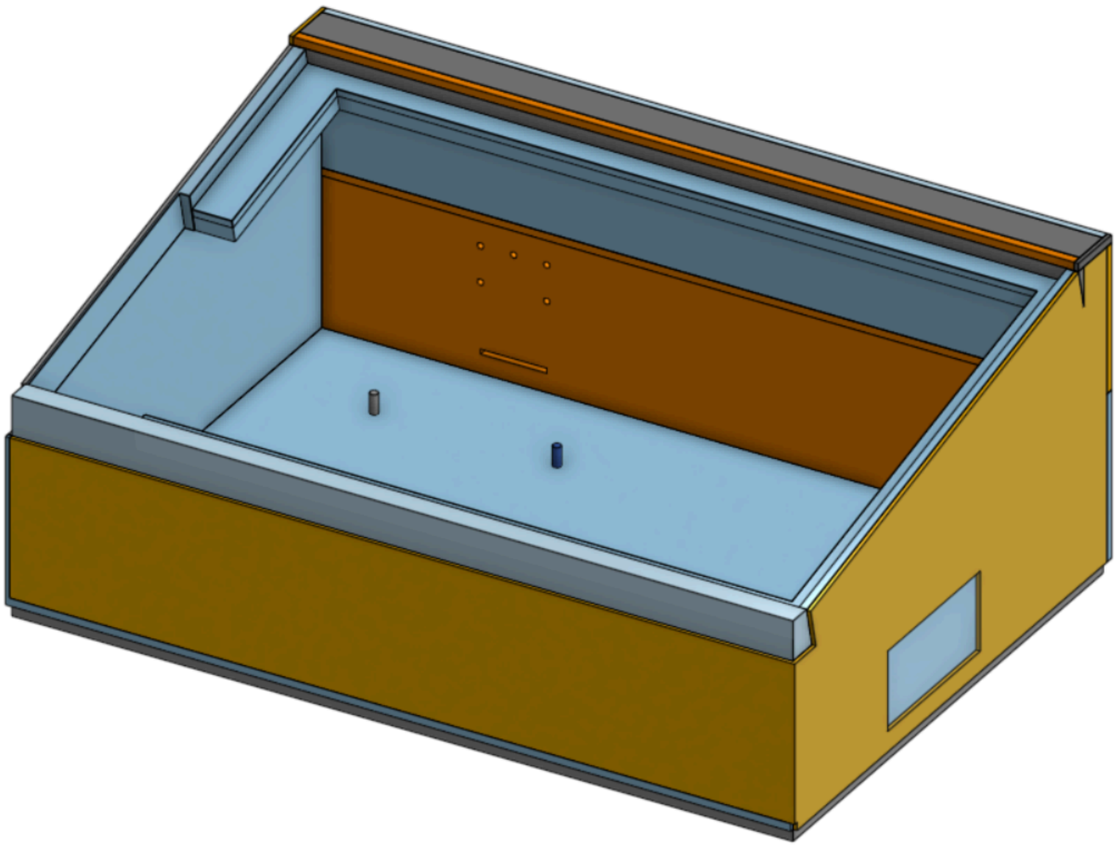


Figure 6.1 *GPMS Enclosure*

6.2 Overall Schematic

The GPMS hardware components were assembled by integrating essential, selected parts. At the core of the system was the Raspberry Pi 5 single-board computer, which served as the "brain" of GPMS. It was powered by a power bank, providing 5V at 3A through a USB-A to USB-C connection.

A touchscreen monitor communicated with the Pi via USB for data transmission and received image data from the Pi over HDMI. The camera module, connected to the Pi through the Serial Camera Interface (SCI), captured images and sent them to the Pi, receiving 3.3V at 250mA power from the Pi in return. The Pi connects via Wi-Fi to a local server using TCP/IP protocols, where it sends requests for AI-generated images. Once retrieved, the Raspberry Pi 5 transmits this image data through HDMI to a projector, which is powered by a wall outlet, displaying the AI-generated image onto the intended surface or structure.

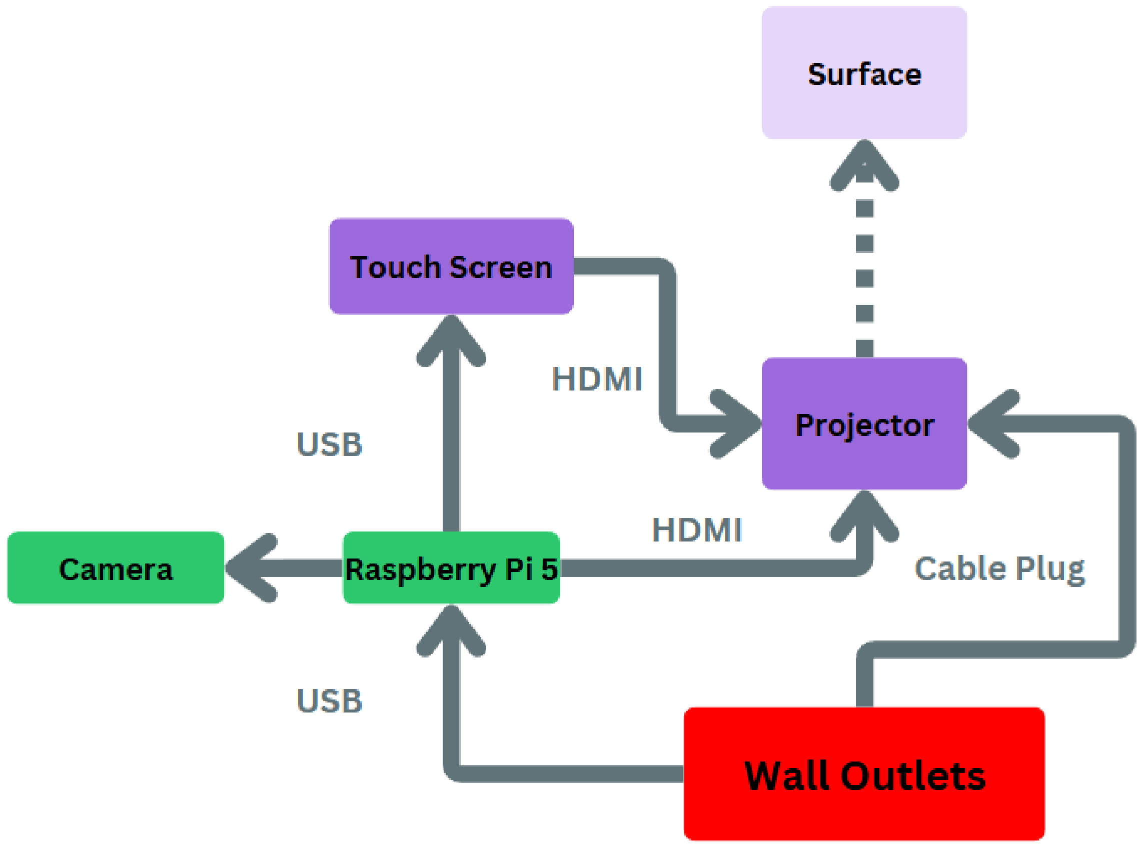


Figure 6.2 *Overall Schematic for System*

Chapter 7 Software Design

7.1 Software Design Overview

Due to the nature of GPMS and its number of off-the-shelf hardware features, cohesive software design was paramount to providing the users with a tailored experience that aligned with the authors' vision. As demonstrated in Section 2.5.3, *Software Diagram*, there were 2 major platforms that needed to be designed and developed: the GPMS device and the generative server stack.

In this chapter, the authors discussed each of these design platforms in detail, highlighting the flow of data both internally and between platforms, as well as the interaction between the user and these unique systems.

7.2 Compute on Device

Within the framework of this system, the Raspberry Pi 5 served as the central processing unit, ensuring the seamless integration of hardware inputs and user directives. Its role began at the very onset of the device's activation, handling the initialization that calibrates the camera to correct for lens distortion and align the projection parameters with the user’s environmental context. This ensured that the images produced were tailored to the physical constraints of the projection surface. The Pi's computation was also leveraged in providing a user-friendly interface through a 10" LCD capacitive touch display, allowing users to input their creative prompts and adjust projection settings in real time.

As the device transitioned to its main operational phase, the Raspberry Pi 5 maintained its pivotal role by managing the “main loop” - a continuous cycle of user interaction, data processing, and projection adjustments. It handled the intricacies of network connectivity, packaged user inputs and sensor data, and dispatched them to an external server for processing.

The primary tools/languages used in developing this platform were OpenCV for its extensive open-source library of computer vision processes, Qt for its native front-end framework, and C++/Python to integrate these tools into a cohesive, full-stack device.

7.2.1 Camera Calibration

The initialization process was a critical sequence that ensured the Raspberry Pi 5, in conjunction with the OV5640 camera, was correctly calibrated for projecting images. As the device powered up, it immediately engaged in a pre-calibration routine to counteract the typical distortions produced by the camera's lens. This step was automated, using the camera's specifications to apply a real-time correction that flattened and aligned the image edges with the projector's target surface.

In the camera calibration process using OpenCV, the focus was on correcting lens distortion to enhance the accuracy of image projection. This involved compensating for both radial and tangential distortions. Radial distortion, which creates a "barrel" or "fish-eye" effect, was addressed using a formula that recalculates pixel positions based on the distortion coefficients. Tangential distortion was corrected through a similar set of formulas, adjusting for the misalignment of the lens with the imaging plane. Both types of distortions are represented in OpenCV by a matrix, which contains five coefficients, $[k_1, k_2, k_3, p_1, p_2]$, demonstrated by the equations below.

$x_{distorted} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$	$x_{distorted} = x + [2p_1 xy + p_2(r^2 + 2x^2)]$
$y_{distorted} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$	$y_{distorted} = y + [p_1(r^2 + 2y^2) + 2p_2 xy]$
Figure 7.2.1.1	**Figure 7.2.1.2**
Equations for Radial Factors	*Equations for Tangential Distortion*

Table 7.2.1.1 *Distortion Equations* [36]

In addition to distortion correction, the calibration process included determining the camera matrix that translates between camera coordinates and real-world units. The camera matrix accounts for the camera's focal lengths and the optical centers expressed in pixel coordinates. For reliable calibration, OpenCV can utilize various types of patterns, such as chessboard or circle grids, captured in multiple snapshots from different angles. These snapshots allow OpenCV to formulate a set of equations that are solved to determine the camera matrix and distortion coefficients.

Once the parameters are obtained, functions like 'cv::undistort' can actually be used to correct the captured images, significantly improving the quality of the projected image even for low-cost and low-quality cameras. [36] Lastly, it's important to note that these 5 coefficients can be pre-computed using the exact camera from GPMS and saved to the device for later use, allowing this tuning to take place instantly without any user interaction.

The importance of this process can be seen below in the previously discussed work of Addison Sandvik, who provided images showcasing the importance of lens correction. Note the roundness of the edges towards the horizontal and vertical bounds of the image.

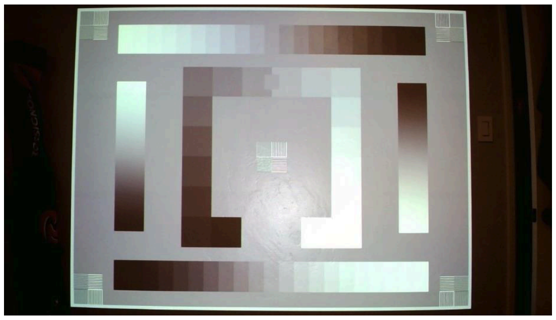
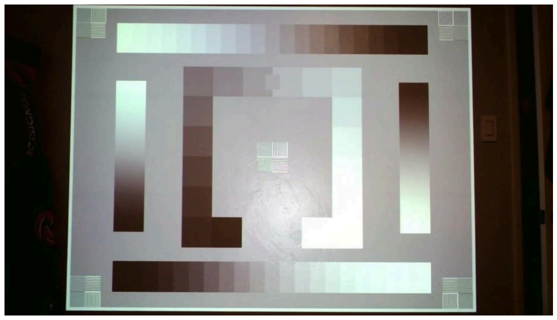
 **Figure 7.2.1.3** *Without Lens Correction* [4]	 **Figure 7.2.1.4** *With Lens Correction* [4]

Table 7.2.1.2 *Comparison Using Lens Distortion* [4]

Following this automated correction, the system transitioned to a user-interactive mode. It projected a calibration image onto the surface, which was then used to select the 4 outer corners of the display (i.e. the corner points that define the boundaries of the projection area). The 10" LCD capacitive screen was used to provide a tactile interface for the user. Users were then able to directly interact with the calibration image by dragging and adjusting corners to achieve an exact overlay of the projected image with the physical boundaries. Next, using a transform, the image capture was cropped/shifted using these points to account for unevenness in the edges. This careful alignment was what ensured that subsequent images were projected with precision, fitting neatly within the desired area. From here, edge detection was completed to show the aligned vertices.

7.2.2 Threshold Calibration

Another essential aspect of initialization was defining the threshold values for the edge detection process. These could not be automatically set without complex computer vision algorithms due to any number of environmental variables that GPMS might be introduced to (i.e., lighting, diverse colors, patterns, etc.); therefore, it was preferable for the user to set these values manually using a simple slider to decide what looks best.

These thresholds acted as filters determining what constituted an edge, thereby defining the critical contours of the object/shape. Lower threshold values may have resulted in the detection of faint or less pronounced edges, potentially introducing noise and unwanted details. Conversely, excessively high threshold values could lead to the omission of significant edges, rendering the object unrecognizable. The fine line between these extremes was where the optimal edge definition was achieved, ensuring that the object is accurately captured without superfluous data.

Below is an example using an early prototype of the project to show the differences between various threshold values and the importance of the user fine-tuning these values manually. The image used to highlight these differences is *"How to Draw Pusheen with Donut"*. [37]

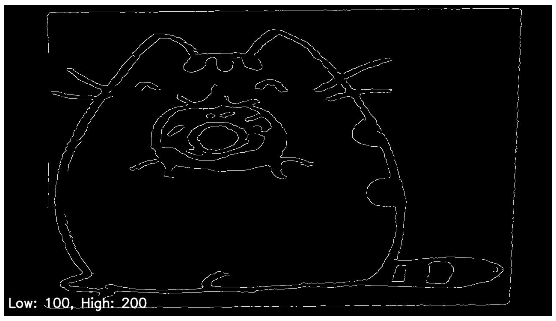
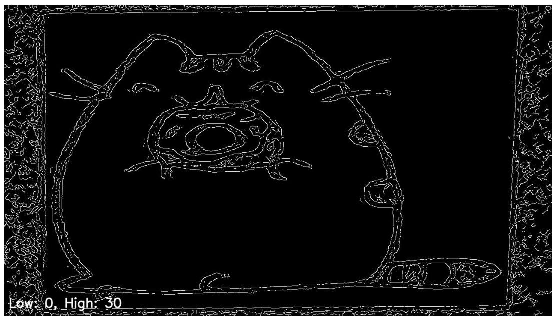
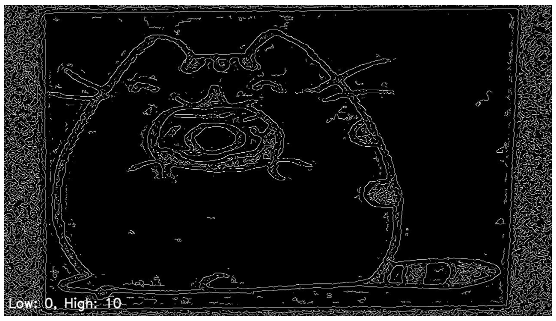
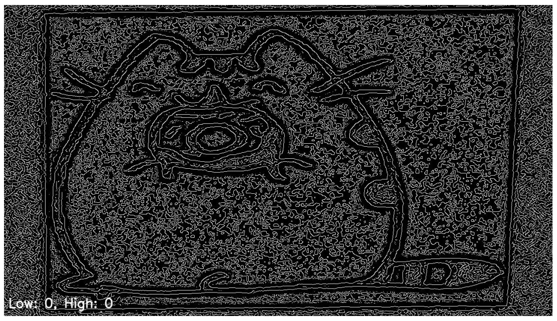
 **Figure 7.2.2.1** *Thresholding - Low: 100, High: 200*	 **Figure 7.2.2.2** *Thresholding - Low: 0, High: 30*
 **Figure 7.2.2.3** *Thresholding - Low: 0, High: 10*	 **Figure 7.2.2.4** *Thresholding - Low: 0, High: 0*

Table 7.2.2 *Thresholding Comparison*

In the first image in the top left of the table, with a low threshold of 100 and a high of 200, moderate edge detection was observed, with key boundaries of the subject discerned, yet some finer details were potentially lost. This setting might be ideal for scenarios where the primary goal would be to detect prominent features while avoiding excessive noise that could complicate the analysis.

The subsequent images, beginning in the top right of the table, depicted the increasing impact of lowering both the high and low threshold values, with the last one featuring the most extreme case where both thresholds were set to 0. The progressive reduction in thresholds resulted in a more comprehensive capture of edges, including more subtle variations in the image. However, this sensitivity also includes undesirable noise, highlighting the critical nature of striking a balance between detail and clarity.

7.2.3 Main Loop

Once initialization concluded, the main loop began. The 10" LCD acted as the interactive gateway for users to input their creative directives. The system was designed for straightforward engagement: users can type/select prompts and choose from various styles for how the image should look.

The most important aspect of this step was that the UI would be clean and responsive. The design of the UI/UX is discussed further in section 7.4. After entering a prompt and selecting a style, the user would be presented with a choice of 2 images matching their creative aspirations. They are then able to choose any one of these, save them for later display, or regenerate a new set of 2 images. From there, after they have found an image they think is suitable, they will be able to then display it via the projector on the target surface once they select the desired image. Finally, when they want to move the projector or adjust it for any reason, they will also be able to re-enter the initialization process at any point to re-calibrate the frame.

7.2.4 Network Connectivity

Next, the camera data, alongside the user's input and selected threshold values, were bundled and transmitted to a non-local server for processing using an SSH connection powered by TCP to generate the final image before projection on the surface. This process was then handled by a Python script that established the SSH connection, handled the requisite SCP commands, and was then able to trigger the server's Python script remotely.

This connection utilized a non-traditional method to connect the components responsible for the data transfer/image request through the available IP address on the University network. The data packet was made up of 2 critical aspects, which were added to a folder and zipped before transferring via SCP to the server (a protocol established through SSH).

Those files were made up of the following information:

info.json
- Threshold Lo: uint8_t
- Threshold Hi: uint8_t
- Prompt: string
- Style: string
- Image Size (4 bytes): uint32_t

Image.png
- The image of the scene that can be used to map ControlNet

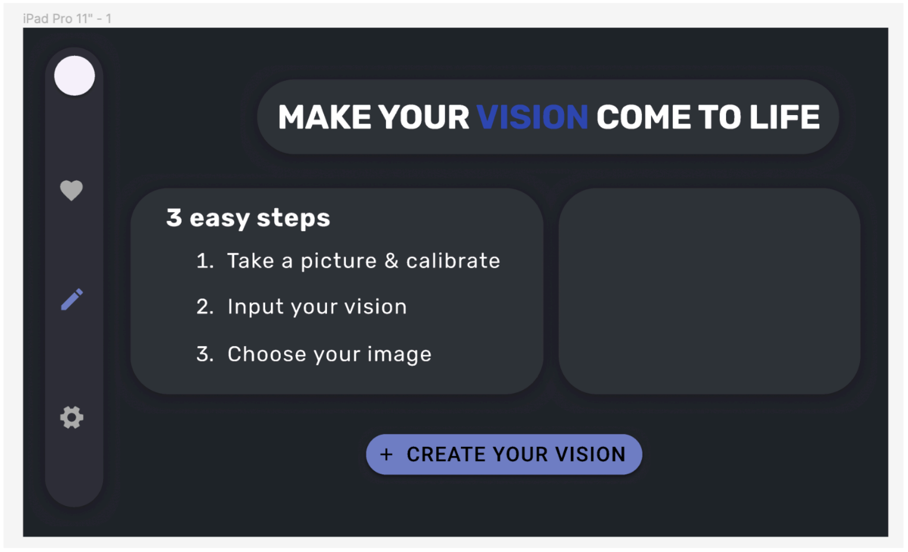
From there, the script utilized SCP to be able to access the newly created files and have them downloaded onto the Raspberry Pi 5 and then transferred to the touch screen monitor for selection and display via projector.

It is important to note the context in which this design has been engineered. The machine being used exists in a secure UCF Research Lab (DRACO) on campus, as AMD has provided the authors access to it. The computer has only been made available remotely through SSH, and there is an unlikely chance of UCF allowing a port to be made available to create a legitimate REST API. If this project were to become its own legitimate venture outside of Academia, this network architecture would be changed to a more normalized networking standard, likely using some sort of formal API process on an available port. These constraints are further discussed in Section 4.2.4, *Constraints in Security/Network Integration*.

7.3 User Interface

The user interface of the Graphical Projection Mapping System (GPMS) was designed to provide a seamless and intuitive experience for users, enabling them to generate and display images on physical structures with ease. The authors conducted extensive research on application design principles to create an interface that is both user-friendly and visually appealing, focusing on simplicity, clarity, and consistency.

The main objectives of the GPMS user interface were to guide users through the process of capturing an image, adjusting edge detection settings, inputting a text prompt, selecting a generated image, and aligning that image with the physical structure. By minimizing visual clutter and providing interactive controls, the interface aimed to enhance the user experience and support the core functionalities of the application.



3 easy steps

1. Take a picture & calibrate
2. Input your vision
3. Choose your image

Figure 7.3.1 *Home Page*

The workflow of the GPMS application begins on the home page (Figure 7.3.1), where users can initiate the image capture process. After clicking the "Submit" button, users are directed to the picture page (Figure 7.3.2), where they can approve the captured image or choose to retake the picture if needed. This flexibility allows users to ensure that the captured image accurately represents the desired structure.

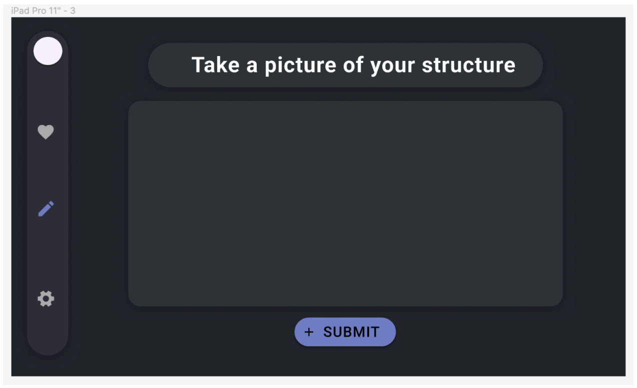


Figure 7.3.2 *Picture Page*

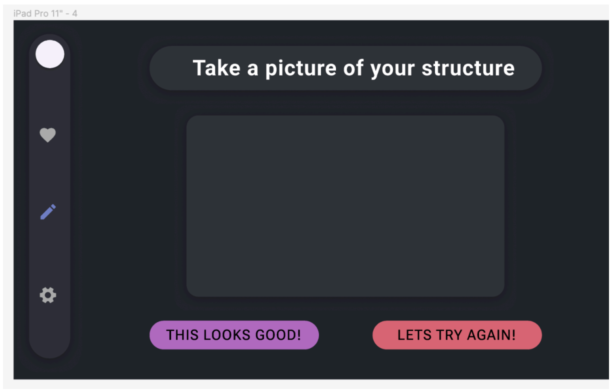


Figure 7.3.3 *Picture Approval Page*

Once the image is approved, the application displays an edge-detected version of the image on the sensitivity threshold page (Figure 7.3.4). Here, users can interact with two sliders to adjust the amount of edges included in the image, providing control over the level of edge outline detail. This feature enables users to fine-tune the edge detection process to suit their specific needs and preferences.

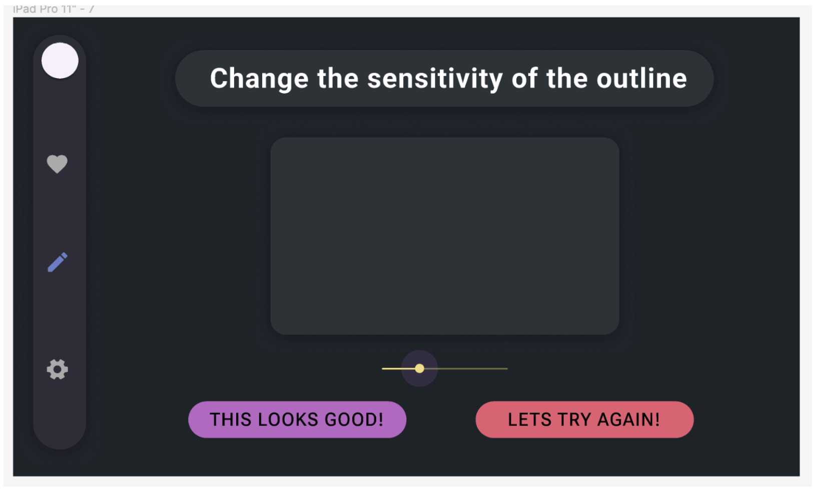


Figure 7.3.4 *Sensitivity Threshold Page*

After finalizing the edge detection settings, users proceed to the input prompt page (Figure 7.3.5), where they can enter a text prompt to guide the image generation process. The application then utilizes both the user-provided prompt and the captured structure to generate a set of images, which are presented on the generated image selection page (Figure 7.3.1). Users can browse through these images, save their favorites, and select the image they wish to display on the physical structure.

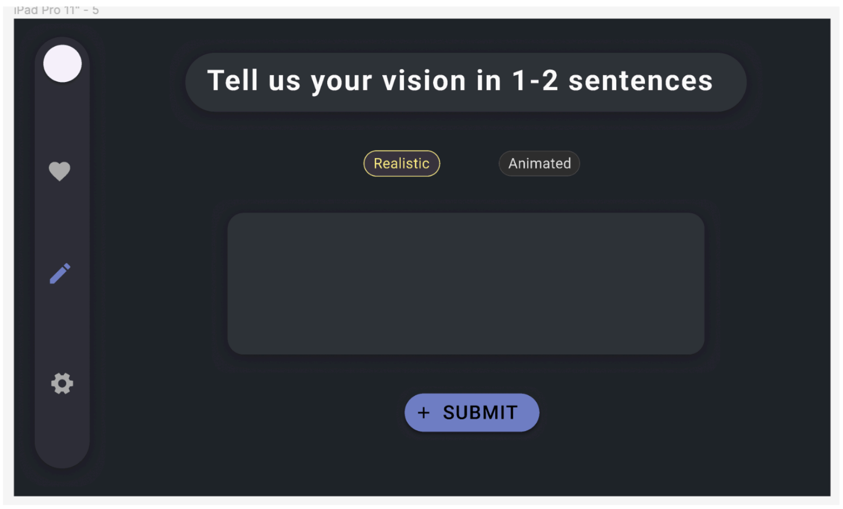


Figure 7.3.5 *Input Prompt Page*

To ensure accurate projection of the selected image onto the structure, the application provides an intuitive cropping-like function. Users can drag the corners of the image to their corresponding locations on the physical structure, allowing for precise positioning and scaling. This feature is essential for achieving a seamless and visually coherent projection, aligning the edges of the image with the real-world structure.

Throughout the development process, the authors prioritized design principles such as simplicity, clarity, and consistency. The user interface features a dark theme, which not only reduces visual clutter but also enhances the viewing experience in low-light environments. The minimalist design approach ensures that users can focus on the core functionalities of the application without unnecessary distractions.

In addition to its core features, the GPMS user interface incorporated personalization options, such as the ability to save and "favorite" generated images. This allows users to curate their own collection of artworks and easily access their preferred images for future use.

The GPMS user interface offers a compelling and user-friendly experience, guiding users through the process of generating and displaying images on physical structures. By combining intuitive controls, interactive features, and a visually appealing design, the interface supports the core functionalities of the application while prioritizing simplicity and clarity.

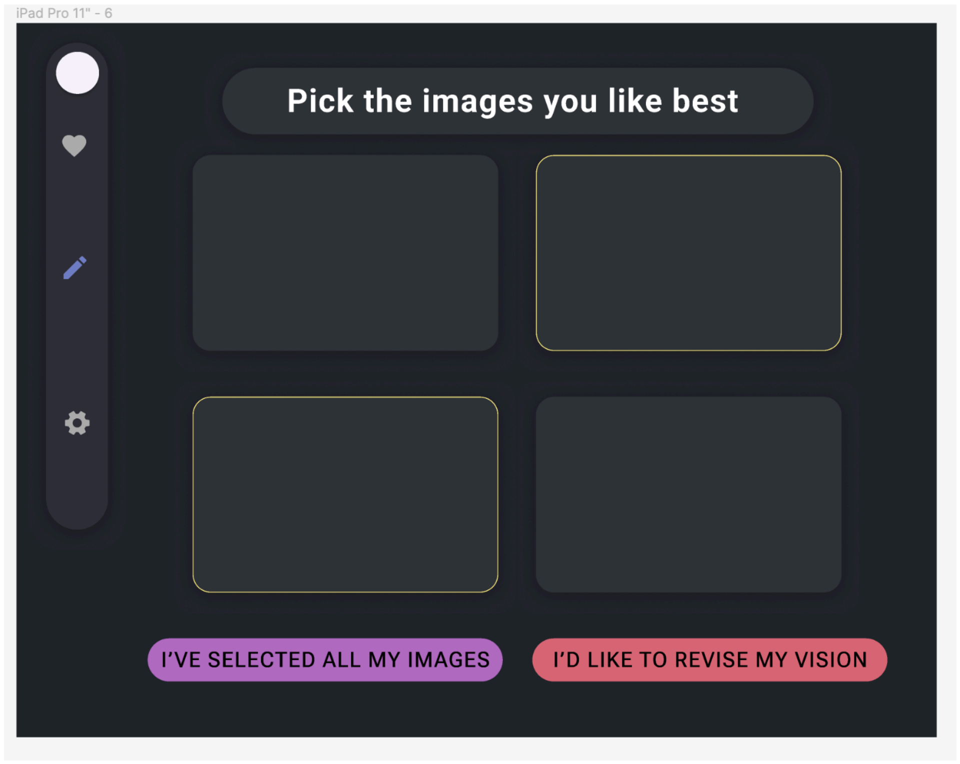

iPad Pro 11" - 6
Pick the images you like best
I'VE SELECTED ALL MY IMAGES
I'D LIKE TO REVISE MY VISION

Figure 7.3.6 *Generated Image Selection Page*

The user interface of GPMS was meticulously prototyped using Figma, a powerful wireframing and design collaboration tool. By leveraging Figma's capabilities, the team ensured that all members had a clear understanding of the application's flow, visual aesthetics, and user experience. This collaborative approach allowed for seamless communication and alignment among the team, which fostered a unified vision for the project.

During the initial stages of the design process, the team considered implementing a vibrant and cheerful color scheme to create an engaging and lively user interface. However, a crucial insight from the authors prompted a significant shift in the design direction. It was astutely pointed out that GPMS would likely be used primarily in dimly lit or dark settings. In these environments, a bright and colorful interface could potentially interfere with the immersive experience and distract users from the core functionality of the application.

Taking this valuable feedback into account, the team made a strategic decision to adopt a darker themed user interface. By employing a color palette dominated by darker shades and muted tones, the interface would seamlessly blend into the ambient lighting conditions, minimizing visual distractions and allowing users to focus on the primary tasks at hand. This design choice not only enhanced the overall user experience but also demonstrated the team's adaptability and willingness to incorporate insights that align with the specific context and requirements of the application.

The decision to pivot towards a darker themed UI exemplified the authors' commitment to creating an interface that was not only visually appealing but also functionally optimized for the intended use case. By carefully considering the environmental factors and user needs, the team crafted a user interface that strikes a harmonious balance between aesthetics and practicality, ultimately enhancing the usability and effectiveness of the GPMS application.

7.4 Generative Design Stack

This section focuses on the image generation process that creates images based on the structure's edges and the user's prompt. The process takes input from the user interface, which is passed through a script (as mentioned in Section 7.2.4) and then to a program responsible for generating the images.

The image generation process involves several key components, including prompt expansion, edge detection pre-processing, ControlNets, and the Stable Diffusion model. [38] These components work together to generate images that accurately reflect the user's prompt and incorporate the structure's edge information.

First, the user's prompt undergoes prompt expansion, a technique that enhances the accuracy and detail of the generated images. Next, the structure's edge information is obtained through the image taken by the user through the user interface. The image is passed through a specific edge detection preprocessor to extract the relevant edge features.

The processed edge image is then fed into a ControlNet, a neural network that allows for more user control over the image generation process. The ControlNet outputs a set of conditioning vectors/maps that encode the desired modifications to the Stable Diffusion model. The Stable Diffusion model is a deep learning model trained to generate images based on input prompts and conditioning information. It uses the conditioning vectors or maps from the ControlNet and the expanded prompt to guide the image generation process, iteratively refining the output image. The following subsections will delve into each component of the image generation process in more detail.

7.4.1 Prompt Expansion

Prompt expansion is a technique that plays a crucial role in generating more detailed and vivid images when working with Text-to-Image models. It addresses the common challenge of users providing insufficient information in their input prompts, which can result in less descriptive and less accurate generated images. By employing prompt expansion, the system can enhance the user's input prompt by adding more relevant details and context.

During the research phase, it became apparent to the authors that while there are numerous models available for Text-to-Image generation, finding a suitable model specifically designed for prompt expansion proved to be more challenging. The authors encountered difficulties in identifying a free and readily available model that was tailored for Text-to-Image generation. It was crucial to select a prompt expansion model that is compatible with Text-to-Image tasks, since using a model intended for other purposes may potentially degrade the quality of the generated images.

After extensive research, the authors discovered that one viable approach is to develop a custom prompt expansion model. While the Stable Diffusion model itself incorporates a built-in prompt expansion mechanism, user feedback suggests that employing an additional prompt expansion step can further improve the quality and coherence of the generated images. [38] However, before investing resources into creating a custom prompt expansion model, the authors needed to assess the performance of GPMS without prompt expansion. This evaluation would prove to help them determine whether prompt expansion was indeed necessary for achieving the desired results.

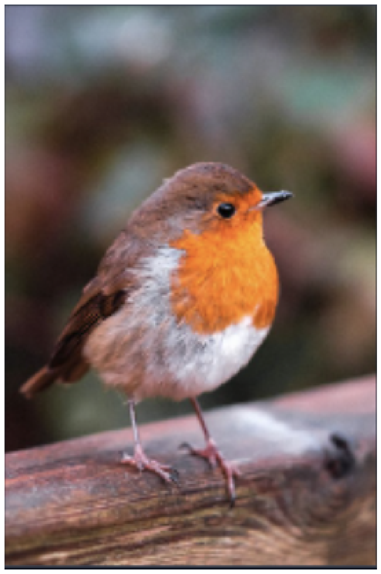
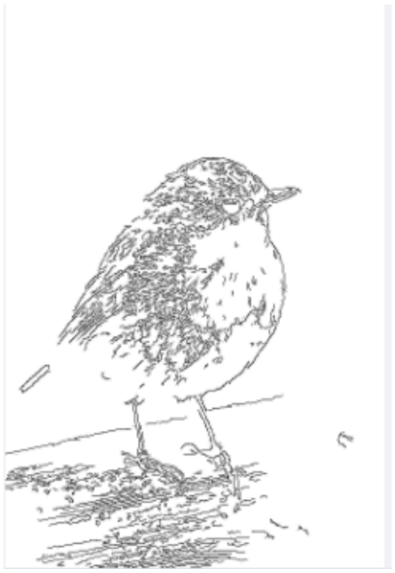
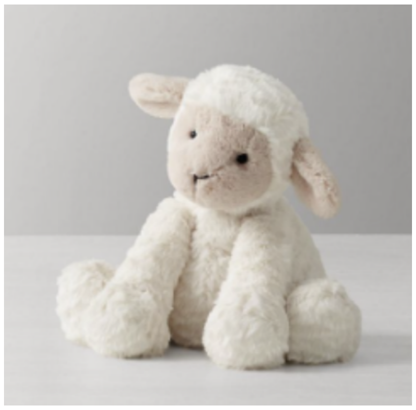
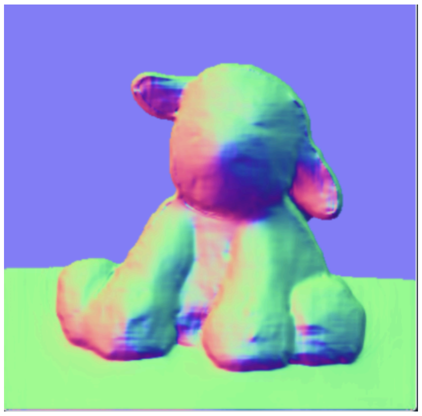
If the assessment indicated that prompt expansion would significantly enhance the system's output, or if time permits, the authors would further investigate the process of developing their own prompt expansion model. This may involve exploring techniques such as fine-tuning pre-trained language models, leveraging domain-specific datasets, or employing few-shot learning approaches. Alternatively, the authors planned to continue their search for a suitable prompt expansion model that was specifically designed for Text-to-Image generation. This involved collaborating with the research community, exploring open-source repositories, and considering commercial solutions that aligned with the project's requirements and constraints.

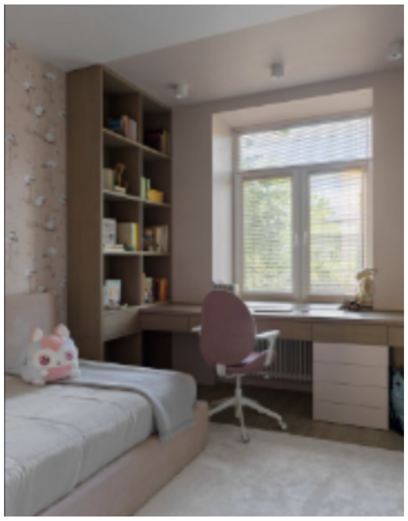
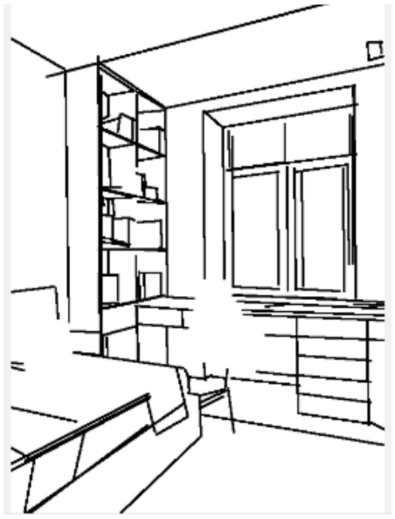
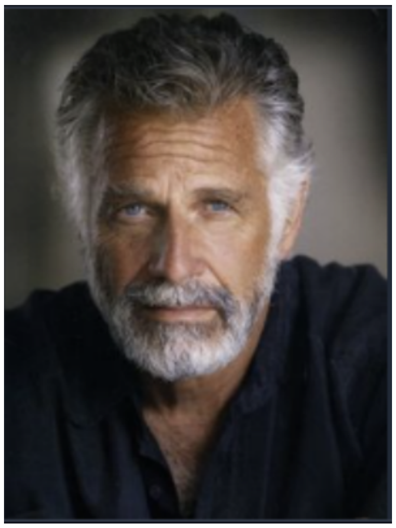
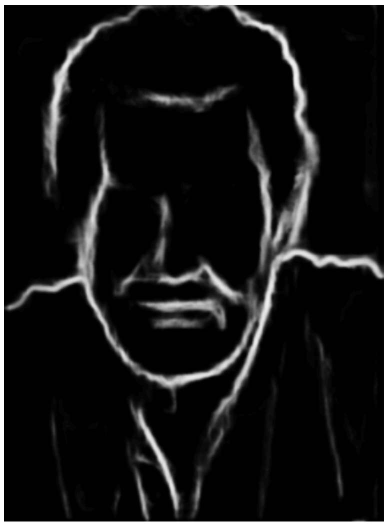
Incorporating an effective prompt expansion module into the GPMS pipeline has the potential to greatly improve the quality, relevance, and creativity of the generated images. By providing more comprehensive and detailed prompts, the system can better capture the user's intent and generate images that are more visually appealing and semantically meaningful.

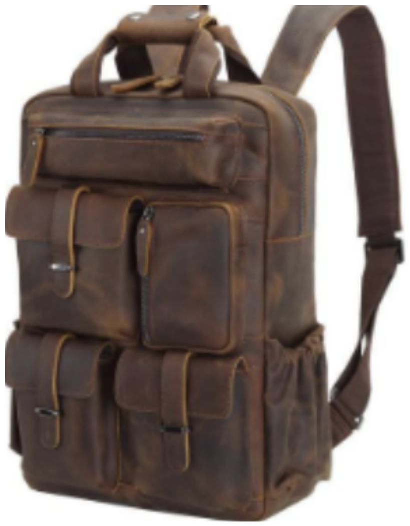
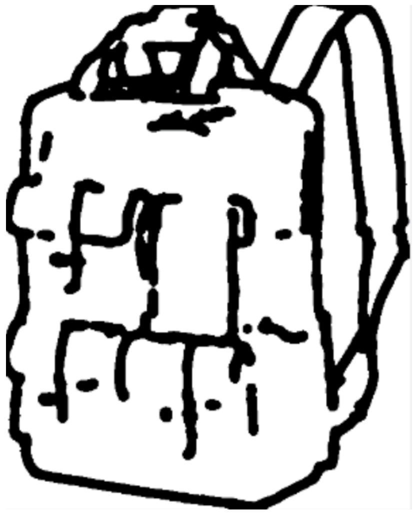
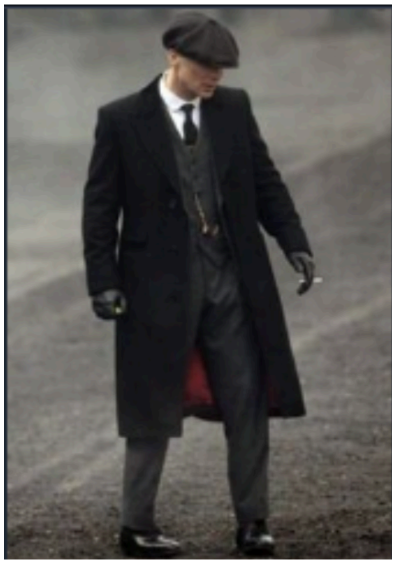
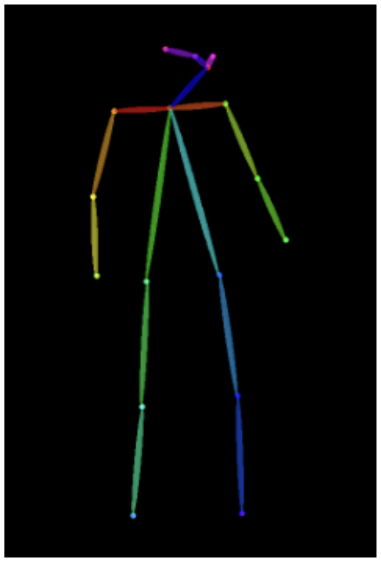
7.4.2 Edge Detection Pre-Processing

As discussed in the introduction of this section, pre-processing must occur on the image prior to being inputted into the ControlNet. This pre-processing step involves applying specific edge detection techniques to the image and returning a new image that highlights the relevant edge features. Edge detection is a crucial component in the Stable Diffusion with ControlNets pipeline, as it provides the necessary conditioning information for the ControlNet to guide the image generation process.

There are several types of edge detection techniques available for Stable Diffusion with ControlNets. In this project, we have access to eight different edge detection models: canny, depth, HED boundary, scribble, semantic segmentation, normal, Openpose, and M-LSD. Each of these models has its own unique characteristics and outputs, as shown in the table below. The example images are sourced from the GitHub repository that contains the respective models.

Edge Detector	Input Image	Output Image
Canny	 **Figure 7.4.1.1** *Pre-canny*	 **Figure 7.4.1.2** *Post-canny*
Normal	 **Figure 7.4.1.3** *Pre-Normal*	 **Figure 7.4.1.4** *Post-Normal*

M-LSD	 **Figure 7.4.1.5** *Pre-M-LSD*	 **Figure 7.4.1.6** *Post-M-LSD*
HED	 **Figure 7.4.1.7** *Pre-HED*	 **Figure 7.4.1.8** *Post-HED*

Scribbles	 **Figure 7.4.1.9** *Pre-Scribbles*	 **Figure 7.4.1.10** *Post-Scribbles*
Openpose	 **Figure 7.4.1.11** *Pre-Openpose*	 **Figure 7.4.1.12** *Post-Openpose*

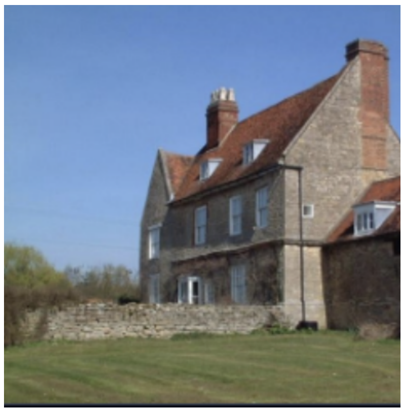
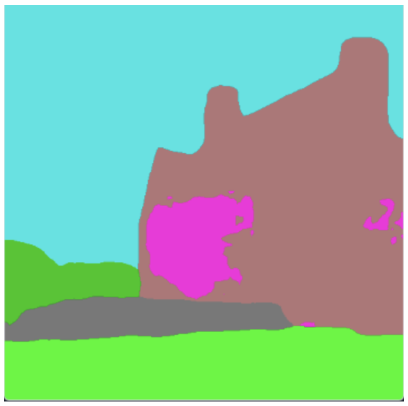
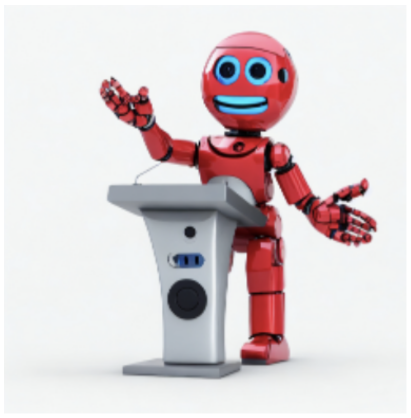
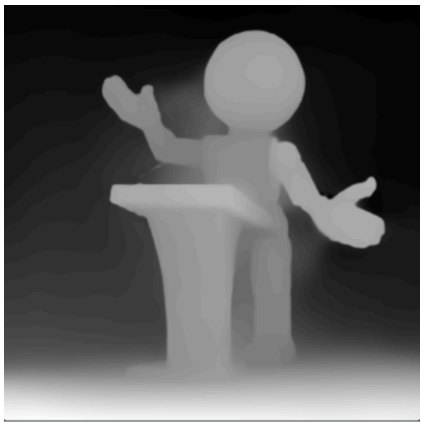
Semantic Segmentation	 **Figure 7.4.1.13** *Pre-Semantic Segmentation*	 **Figure 7.4.1.14** *Post-Semantic Segmentation*
Depth	 **Figure 7.4.1.15** *Pre-Depth*	 **Figure 7.4.1.16** *Post-Depth*

Table 7.4.2 *Edge Detection Models* [7] [39]

The choice of edge detection model depended on the specific requirements and characteristics of the project. In the context of this project, where the projector was expected to be positioned approximately 6 feet away from the structure, certain models were more suitable than others.

Based on the distance and the nature of the structure, models like semantic segmentation and depth were not the most effective. These models tended to perform better when capturing details at further distances. On the other hand, edge detection models such as canny, M-LSD, and HED appeared to be the most promising candidates for GPMS.

The type of edge detection model used at this step has a significant effect on the output of the ControlNet and therefore the final generated image. Different edge detection models capture different types of edge information and provide varying levels of detail, which determines how ControlNet regulates the Stable Diffusion through the image generation process. Therefore, edge characteristics and the level of detail are important characteristics to take note of when choosing a model.

Canny edge detection is a widely used technique that identifies edges by looking for the local maxima of the gradient of the image. It applies noise reduction, non-maximum suppression, and hysteresis thresholding to produce clean and accurate edge maps. Canny edge detection is known for its ability to detect fine details and produce sharp edges. Canny edge detection models focus on capturing fine-grained edges and produce thin, precise edge maps. This can lead to generated images with sharp and well-defined edges, which can enable the ControlNet to guide the Stable Diffusion model towards generating images with more precise and detailed features. [40]

M-LSD (Multiscale Line Segment Detector) is a line segment detection algorithm that identifies meaningful line segments in an image. It is particularly effective in capturing long and straight edges, making it suitable for detecting the overall structure and boundaries of objects. This can result in generated images with more pronounced and coherent structural elements. Because this model focuses on higher-level edge information, it may result in generated images with more abstract or simplified representations of the structure. [41]

HED (Holistically-Nested Edge Detection) is a deep learning-based edge detection model that leverages a fully convolutional neural network architecture. It is trained to predict edges at multiple scales and combines the predictions to produce a final edge map. HED is known for its ability to capture both fine-grained and high-level edge information. This can lead to generated images with a balance of local and global edge coherence, which means the image generated is more precise and with detailed features, similar to the canny edge detection model. [42]

The selection of the appropriate edge detection model was based on empirical evaluation and testing with multiple structures and lighting conditions of the project. The authors conducted experiments and assessed the performance of each model, considering factors such as the accuracy of edge detection, the level of detail captured, and the overall quality of the generated images. Through these evaluations, we determined which edge detection model or models are best suited for our specific use case. It is possible that a single model may emerge as the most effective choice, and/or the authors may find that different models excel in different scenarios.

Furthermore, we explored the possibility of providing users with the option to choose their preferred edge detection model. This would allow for greater flexibility and customization, enabling users to select the model that aligns with their specific requirements and/or artistic preferences. By offering a choice of edge detection models, GPMS caters to a wider range of user needs and enhances the overall user experience.

Ultimately, the decision on which edge detection models to incorporate into GPMS would be made based on a combination of performance evaluations and the specific goals of the project. The pre-processed edge image, obtained from the selected model, served as the conditioning input for the ControlNet, enabling it to guide the Stable Diffusion model towards generating images that align with the desired structure and edge features.

7.4.3 Stable Diffusion with ControlNet

This part of the process ran concurrently with the edge detection pre-processing. As explained in detail in Section 2.2.4 Stable Diffusion with ControlNet, ControlNet is essentially a copy of the Stable Diffusion model that functions as a HyperNetwork. It injects weights from ControlNet into the Stable Diffusion model, allowing information from the pre-processed image to be incorporated into the final output.

In this project, the authors utilized the Stable Diffusion with ControlNet Pipeline from the Diffusers library provided by Hugging Face. The pipeline employs the same ControlNet models created by the original ControlNet developers. These models include both the Stable Diffusion model and ControlNet, along with the specific edge detection model used. It is important to note that ControlNet is also tailored to the particular edge detection model employed.

By using the pipeline, the authors could easily import the appropriate model and input all the necessary user information, which is transmitted through the zip file as described in Section 7.2.4. The pipeline handles the integration of the pre-processed edge image with the ControlNet and Stable Diffusion models, generating the final output image. The generated image is then sent back to the Raspberry Pi using the same communication mechanism outlined in Section 7.2.4.

The Stable Diffusion with ControlNet Pipeline simplified the implementation process by providing a high-level interface for utilizing ControlNet models. It abstracts away the complexities of integrating ControlNet with Stable Diffusion, allowing complete focus on providing the necessary inputs and handling the generated output. This streamlined approach enhanced the efficiency and maintainability of the image generation pipeline.

7.4.4 The Generate Program

The entire image generation process was encapsulated within a single program, which was executed by the script described in Section 7.2.4. This program followed the sequence of steps outlined in Section 7.3, with all the required variables being passed through the script. The program begins by applying a prompt expansion model to ensure that the user input prompt is as accurate and detailed as possible. This step aimed to enhance the quality and relevance of the generated images by providing a more comprehensive and specific prompt.

Next, the program determines which edge detection model to use based on the results of our testing and evaluation, as explained in Section 7.3.2. Depending on the authors findings, the edge detection model may be pre-selected by the authors or chosen by the users through the user interface. Once the appropriate edge detection model is determined, the program proceeds to pre-process the input image using the selected model. This step involves applying the specific edge detection technique to extract the relevant edge features from the image. With the pre-processed edge image available, the program will then invoke the corresponding Stable Diffusion with ControlNet model, which is specifically associated with the chosen edge detection model. The ControlNet model then guides the image generation process by conditioning the Stable Diffusion model based on the edge information provided.

The Stable Diffusion with ControlNet model generates the final output image's based on the user's prompt and the pre-processed edge image. These generated images are then returned to the Raspberry Pi 5 and displayed on the user interface via touch-screen monitor for the user to view and interact with. By encapsulating the entire image generation process within a single program, a cohesive and efficient workflow was ensured. The program handles the prompt expansion, edge detection model selection, pre-processing, and integration with the Stable Diffusion and ControlNet models. This modular approach allowed for easy maintenance, debugging, and future enhancements to the image generation pipeline.

Overall, the Generate Program served as the central component that orchestrates the various steps involved in generating images based on user prompts and edge detection techniques. It leveraged the power of Stable Diffusion with ControlNet to produce visually compelling and structurally coherent images that aligned with the user's intentions.

Chapter 8 System Fabrication/Prototype Construction

8.1 Final GPMS Integration, Testing, & Results

To test the integration of hardware and software, festive storylines were used to evaluate GPMS in a realistic setting. There were three critical metrics that the authors focused on—elapsed time, alignment accuracy, and AI generation time. These were key for evaluating system performance and efficiency. For elapsed time, the authors set a goal for GPMS to complete setup and image projection within five minutes. Testing involved ten trials, timed with a stopwatch from initial setup to final image projection. Each trial included steps such as projector setup, calibration, threshold adjustment, prompt entry, and image selection. Excluding AI generation, times ranged from 1 minute 13 seconds to 1 minute 37 seconds, with an average of 1 minute 24 seconds. Including AI generation (which added about 3 minutes), the mean total time was 4 minutes 24.5 seconds (264.5 seconds), well within the target.

Target	Elapsed Time (s)
< 5 min (300s)	73
< 5 min (300s)	92
< 5 min (300s)	87
< 5 min (300s)	89
< 5 min (300s)	80
< 5 min (300s)	97
< 5 min (300s)	78
< 5 min (300s)	90
< 5 min (300s)	82
< 5 min (300s)	77
Mean without Generation	**84.5**
Standard Deviation	**7.64852927**
AI Image Generation	**180**
Mean with Generation	**264.5**

Table 8.1.1 *Elapsed Time Results*

The alignment accuracy test assessed the precision of image projection on the target surface. Across ten trials, alignment accuracy was measured at four points on the projection surface. The average accuracy was 1.1725 mm with a standard deviation of 0.6175 mm, significantly better than the 8 mm target, demonstrating consistent precision for high-quality projection.

Target (mm)	Average Accuracy (mm)
< 8	1.75
< 8	1.75
< 8	2.25
< 8	2.75
< 8	2
< 8	1.25
< 8	0.75
< 8	1.5
< 8	2.25
< 8	1
Mean	**1.725**
Standard Deviation	**0.6175**

Table 8.1.2 *Alignment Accuracy Results*

Measurement 1 (mm)	Measurement 2 (mm)	Measurement 3 (mm)	Measurement 4 (mm)
3	1	2	1
2	0	5	0
4	0	3	2
1	2	3	5
2	2	1	3

1	0	4	0
0	1	0	2
3	0	1	2
4	0	0	5
0	3	0	1

Table 8.1.3 *Four Points - Measurement Results*

For AI generation time, GPMS aimed for under 1.5 minutes per image. In ten tests, generation times ranged from 42.6 to 43.4 seconds, averaging 42.9 seconds with a standard deviation of 0.23 seconds—well within the target, indicating stable performance. Further optimization of overall delivery time, including image transmission back to the user, was necessary to enhance efficiency, precision, and speed, thus improving user experience and reliability.

Target	AI Generation Time
< 1.5/image	42.6
< 1.5/image	43
< 1.5/image	42.8
< 1.5/image	43
< 1.5/image	43.1
< 1.5/image	42.7
< 1.5/image	43.4
< 1.5/image	42.8
< 1.5/image	42.7
< 1.5/image	42.9
Mean	**42.9**
Standard Deviation	**0.2357022604**

Table 8.1.4 *AI Generation Time Results*

Chapter 9 System Testing and Integration

9.1 Early Software Testing

The authors initially developed an early-stage image calibration and image generation prototype. On a Raspberry Pi 4B model, the authors developed a basic image calibration implementation using OpenCV in Python. This was an imperative step the authors took since the ultimate goal was to use a Raspberry Pi 5 for the final version of GPMS. This initial prototype allowed the user to select the corners that the projector was able to reach, effectively calibrating the image to be superimposed over what the camera was viewing. From there, Canny Edge detection was done on the camera view and displayed through the projector, highlighting the edges and vertices seen via the edge detection algorithm. This process is more thoroughly described in Section 9.2, which discusses integration. However, the frame still needed to be adjusted for barrel distortion in the camera lens and automated with a better visualization of the process.

Next, concerning image generation, the authors developed an initial pipeline utilizing ControlNet + Stable Diffusion XL with the open-source GUI Automatic 1111. [5] [7] The table below, in the leftmost cell, is an example of an original existing image. This particular existing image was used due to its explicit edges and cartoonish style. In the middle cell of the table below is an example of an image generated using this new pipeline, using the prompt "*Cat eating a donut.*" Finally, on the rightmost cell of the table below is an animated GIF showing the tightness and accuracy of the edges in the original photo from the leftmost cell of the table. This accuracy would prove to be vital in aligning the images with existing features.

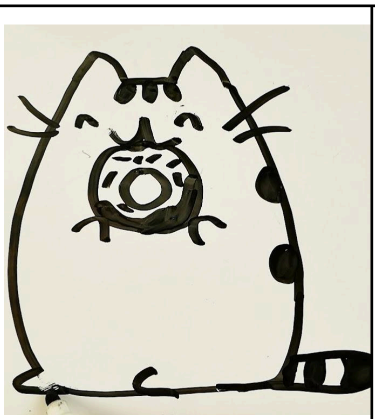
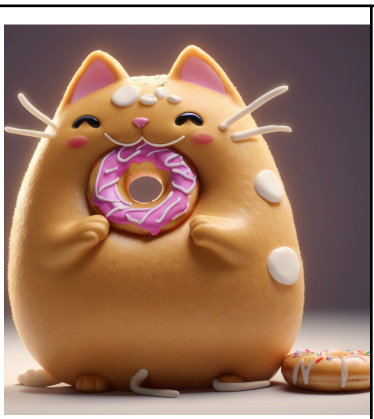
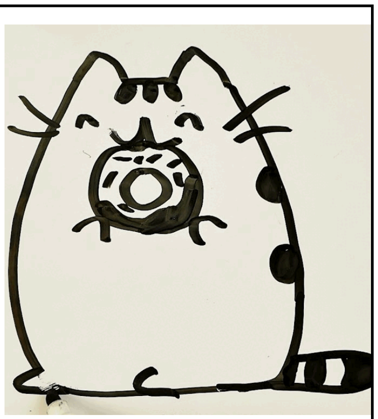
 **Figure 9.1.1** *Original Image [37]*	 **Figure 9.1.2** *Image Generated w/ SDXL*	 **Figure 9.1.3** *Animated GIF to Display Similarities*

Table 9.1 *Reprinted from "How to Draw Pusheen with Donut"* [37]

This process needed further exploration to enable ease of use for the final user. Much of this will come from prompt engineering algorithms, whether it be a LLM or a custom database that the authors can design to find keywords in user prompts and expand on what they believe the user is seeking to generate in their desired image. Finally, collective work should also be done to detect poorly generated images and regenerate them before they are returned to the user which is another critical step in automating the process.

9.2 Integration Testing

In terms of integration, the authors worked to put together a number of aspects that are key to representing the project as a whole. So far, this integration included many of the physical aspects of the GPMS and their connections-specifically the projector, touch screen, Raspberry Pi 4B, and camera, as shown below. This prototype mirrored how the final design will be integrated/operated.

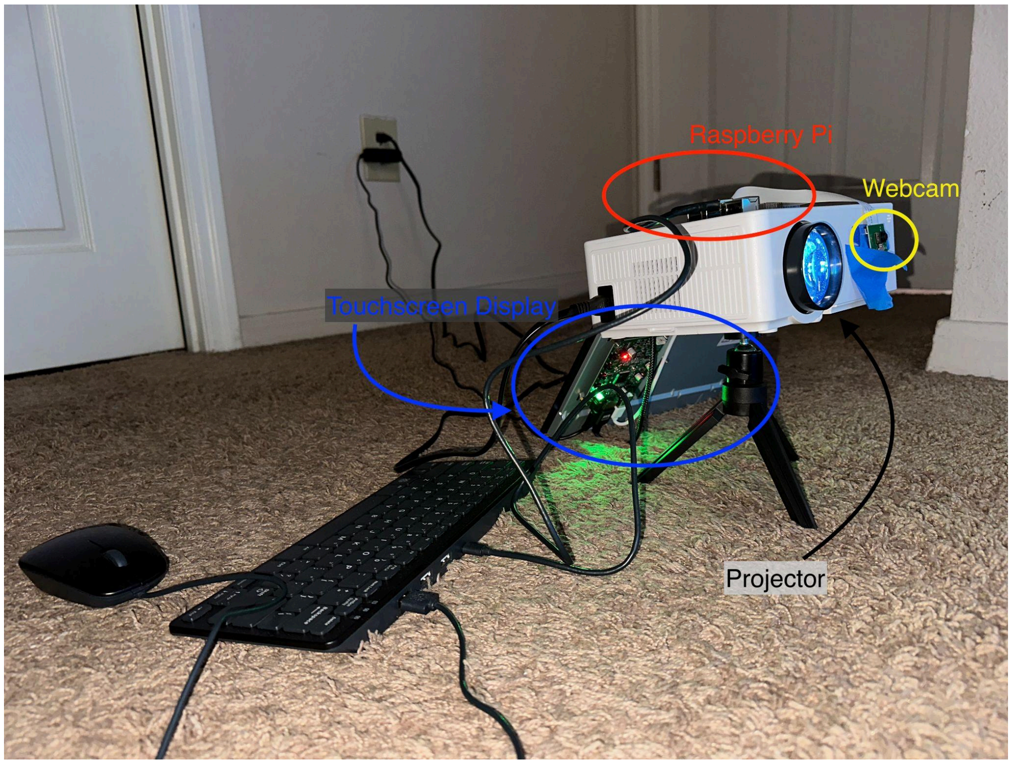


Figure 9.2.1 *Annotated Integration Image*

Once connected and set up, the authors loaded their Python code onto the Raspberry Pi 4B, and after extensive debugging, tuning, and development, the application started. During this time, a whiteboard was placed as a backdrop for the projected scene, and images/words were drawn.

The prototype software, as seen in Appendix D, was built using Python to control OpenCV, which worked well for this initial prototype; however, the Qt framework would prove to be critical to managing windows and displays better, as this initial prototype had an initialization sequence where the user must first drag one window onto the other display before continuing as to circumvent a number of issues with OpenCV integration on the Pi's Operating System.

Once initialized, though, a white screen was shown on the projector to allow for better calibration by the user with a camera stream from the front webcam displayed on the touchscreen. In this "reset" stage, the user selects the outer four corners of the projected scene, as shown below.

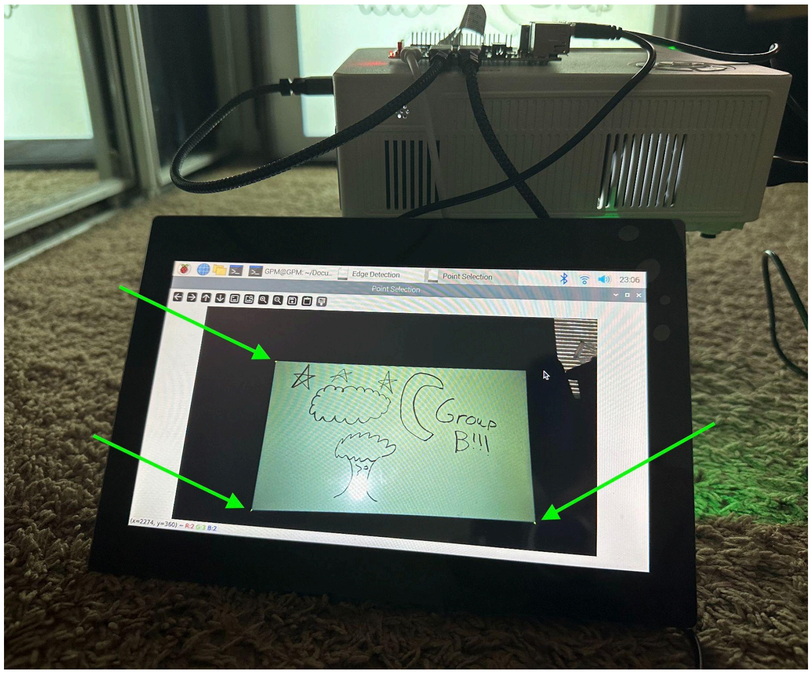


Figure 9.2.2 *Prototype Image Calibration*

Once selected, the device captured a still frame of the scene using the front webcam, and performed edge detection on it, and traced the outline of the scene, as shown below.

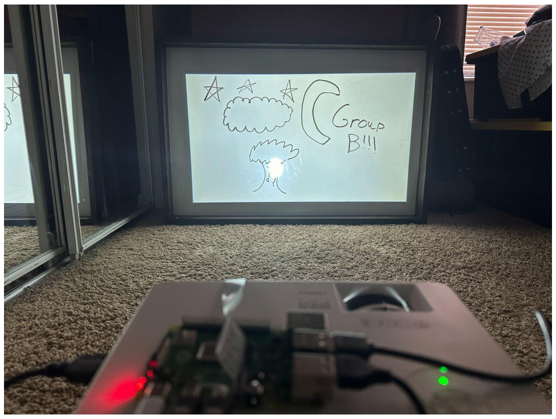
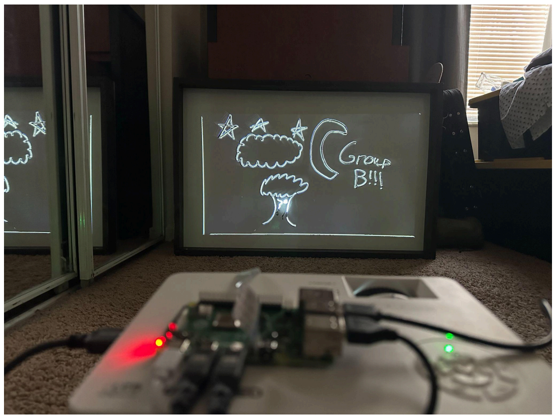
 **Figure 9.2.3** *Initial Drawing*	 **Figure 9.2.4** *Projected Outline*

Table 9.2 *Prototype Image Alignment Performance*

From here, the threshold can then be adjusted using the ‘O’/’P’ and ‘K’/’L’ keys to decrease and increase the low and high thresholds, respectively. Additionally, the user can press ‘R’ to reset the 4 edge points and ‘U’ to update the threshold. When updating the projection, the projector flashed white for ~3 seconds to allow time for the webcam exposure to compensate before grabbing a still frame to perform the edge detection algorithm. Finally, this image was displayed again as before.

This prototype effectively proved the core process for aligning the edges of the projection against surface features. However, this implementation also provided valuable insight into key details for future versions. For example, the fine white edges on the left and bottom edges of the projection were the result of not being able to precisely select the corner due to the size of the image on the screen. Because of this, the edge detection algorithm detected the edge of the projection screen instead of the actual features. Therefore, moving forward, the better plan proved to be to allow the user to zoom in on the image to select the corners more precisely. Additionally, the bright shine of the projector on the semi-reflective whiteboard made detecting the black marker difficult - not totally highlighting the marks in the shine. This was easily solved by using an additional camera such that neither glare would be in the same spot, allowing software to effectively overwrite the issue and create a complete image without glare.

Chapter 10 Administrative Content
10.1 Budget and Financing

The prices provided are based on online research and manufacturer quotes and are, therefore, estimates. These costs are subject to change for anyone who aims to replicate the authors' processes and perform slight differences in implementation of the final product. GPMS had no corporate sponsors and the group covered the majority of associated expenses. Additionally, all the authors agreed to evenly split the total cost (after donations) upon completing Senior Design 2 (EEL 4915L).

ITEM	QUANTITY	PRICE ESTIMATE (USD)
Raspberry Pi 4B	1	60.79
Camera	1	5.00
Monitor	1	89.99
Projector	1	63.89
AMD Machine*	1	3244.44
TOTAL (ESTIMATE)	**N/A**	**~ 3,544.37**
AFTER DONATIONS	**N/A**	**~ 219.67**

Table 10.1 *Overall Project BOM*

10.2 Project Milestones

The dates outlined in the table provided served as crucial milestones for the group, marking hard deadlines that all the authors adhered to in order to manage the progress effectively. The authors established weekly meetings to address individual concerns and collectively ensure that they maintain successful momentum in advancing the overall design objectives.

During the initial phase of Senior Design 1 (EEL 4914C), the primary focus was on fulfilling administrative responsibilities. This included meeting deadlines for outlining all project details and specifications meticulously. Once these foundational tasks were successfully accomplished, the authors shifted their focus entirely toward the development phase of the project. This concentrated effort was aimed at ensuring substantial progress by the end of Senior Design 1, setting a solid foundation for the subsequent phase, Senior Design 2.

10.2.1 Senior Design 1

All project details were completed and finalized by the end of Senior Design 1. The authors made it a priority to not make any further adjustments past the project planning phase for GPMS. Below is a table that comprehensively outlined the entire roadmap of Senior Design 1. This table strategically broke down the roadmap into stages with respective milestones and hard deadlines.

Stage	**Milestone**	**Deadline**
1	Recruit Group Members	1/11/24
2	Individual/Collaborative Research	1/15/24
3	Reach Project Decision	1/24/24
4	Submission of Divide & Conquer Document	2/2/24
5	Divide & Conquer Feedback Meeting	2/5/24
6	Completion of 45 Page Document	3/10/24
7	Finalize BOM	3/15/24
8	Completion of User Interface	3/20/24
9	Submission of 45 Page Document	3/29/24
10	45 Page Feedback Meeting	3/30/24
11	Begin Implementing CV Algorithms	4/15/24
12	Submission of 90 Page Document	4/19/24
13	SD1 Reviewer Committee Check In	4/22/24

Table 10.2.1 *Senior Design 1 Timeline*

10.2.2 Senior Design 2

With the majority of all the project details completed by the end of Senior Design 1, the authors set out to solely use the allotted time for Senior Design 2 to prototype, test, and debug GPMS in order to complete the project well before the final deadline. Below is a table that outlines the entire roadmap of Senior Design 2 broken down into stages with respective milestones and hard deadlines.

Stage	Milestone	Deadline
1	Design Hardware Enclosure	10/10/24
2	Complete Hardware Enclosure	10/20/24
3	Overall CV Software Stack	11/12/24
4	Finishing Touches to UI Design	11/13/24
5	Test Entire Integrated Project	11/15/24
6	Reviewer Committee Demonstration	11/20/24

Table 10.2.2 *Senior Design 2 Timeline*

10.3 Table of Work Distributions

Effective work distribution was crucial for the success of the development of GPMS and ensured that tasks were completed efficiently and on schedule. In the authors' project, which involved various aspects such as PSU/hardware integration, computer vision software stack, UI interface design, prototype design, generative AI pipeline, and overall design report, assigning responsibilities to the appropriate team members was paramount. This division of labor ensured that technical expertise was utilized effectively, with primary responsibility resting on the team member best suited for the task.

This distribution of responsibilities allows each team member to play to their strengths, maximizing productivity and expertise in their designated areas. Ultimately, by carefully assigning tasks to primary and secondary persons based on their skills and capabilities, the authors ensured that efficient progress across all facets of the project, culminated in a successful outcome. The table below visualizes a distribution of work tasks between all the authors with respective primary and secondary persons.

Stage	Component Owner	Secondary Owner(s)
PSU/Hardware Integration	Francisco Soriano	Declan Carter
Computer Vision Software Stack	Declan Carter	Victoria Moreno/Francisco Soriano
UI Interface Design	Victoria Moreno	Francisco Soriano/Declan Carter
Prototype Design	Declan Carter	Victoria Moreno/Francisco Soriano
Generative AI Pipeline	Victoria Moreno	Declan Carter/Francisco Soriano
Design Report/Administrative Tasks	Francisco Soriano	Declan Carter

Table 10.3 *Work Distribution*

Note: Beyond the fundamental stages necessary to the development and completion of GPMS, the website, which contains all documents related to the project, must be maintained. Therefore, Francisco Soriano, in addition to creating the website in HTML and CSS, took on the responsibility to maintain the website throughout the duration of both Senior Design 1 and 2.

10.4 Project Management Strategy

Given that Jira is a widely used project management software in industry, primarily designed for agile teams, the authors decided upon using this software throughout the duration of Senior Design 1 and 2. Additionally, since Jira is scalable to work for small groups to large groups, the authors determined that their three person team would benefit from all that Jira has to offer. Jira's usefulness to the team stemmed from several stand out features regarding the following information listed in the table below. [43]

Stand Out Features	Description
Task Tracking	Jira allows the authors to create and track tasks, issues, bugs, and user stories

	throughout the project lifecycle. This is extremely helpful as it organizes work and ensures that no task is left unaddressed.
Agile Methodology Support	Jira provides visual boards for tracking progress to manage projects effectively.
Collaboration	Jira promotes collaboration among team members by providing features such as commenting, mentioning, and assigning tasks, which in turn fosters communication and transparency amongst the authors.
Reporting and Insights	Jira offers reporting and dashboard features that provide the authors insight into project progress, team performance, and bottlenecks to identify significant areas for improvement.
Remote Work Support	Jira's cloud-based version has allowed the authors to access and manage their tasks from anywhere on or off campus. It has allowed the fostering of collaboration and productivity regardless of physical location.

Table 10.4 *Project Management Strategy*

Jira's versatility, flexibility, and robust feature set made it an invaluable tool for project management for the authors due to several key, stand out, reasons. Moreover, Jira fosters collaboration among team members by offering features such as commenting, mentioning, and task assignments, which enhance communication and transparency among the authors.

Furthermore, Jira's cloud-based version enabled remote work, allowing the authors to access and manage tasks from anywhere, whether on or off campus. This flexibility not only enhanced collaboration but also boosted productivity, regardless of the authors' physical locations. Overall, Jira's comprehensive suite of features made it an indispensable asset for authors to manage the project efficiently.

10.5 Technical Milestones

The Gantt Chart in the image below, provided through Jira Management Software, is a high-level monthly overview of the various technical milestones stretching from January to December in quarter intervals.

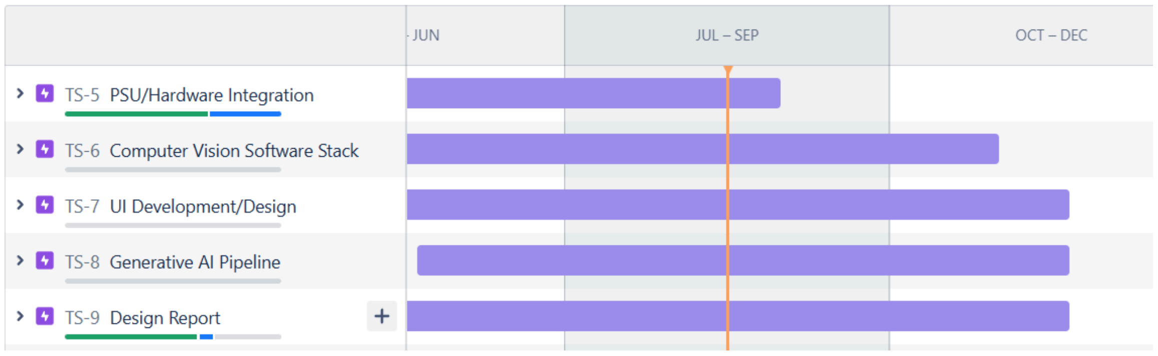

JUN
JUL – SEP
OCT – DEC
TS-5 PSU/Hardware Integration
TS-6 Computer Vision Software Stack
TS-7 UI Development/Design
TS-8 Generative AI Pipeline
TS-9 Design Report

Figure 10.5 *Jira Overview*

The following list depicts a high-level view of what significant, overarching tasks/stages were needed to contribute to the successful completion of GPMS. The main overarching tasks/stages are listed as follows:

- PSU/Hardware Integration
- Computer Vision Software Stack
- UI Development/Design
- Generative AI Pipeline
- Design Report

More specifically, objectives in software and hardware integration for Senior Design 2 were split into Software and Hardware. In terms of planned Software Development, the general outline is as follows:

- **User Interface (UI) Development**: Design a user-friendly interface that allows easy interaction with the system and aligns with the overall design plans discussed in Section 7.3. The UI will enable users to seamlessly enter custom prompts and view generated images, providing an intuitive platform for both novice and experienced users.
- **Networking Infrastructure**: To facilitate robust communication between the user interface and server, develop a networking infrastructure as discussed in Section 7.2.4. This setup will handle requests and responses efficiently, ensuring smooth data flow and system responsiveness.
- **Python Server Development**: Create a dedicated Python server to manage the generative design stack, which will reside on the server as discussed in Section 7.4. This server will handle requests and handle the automation of image generation as it is discussed in the next objective.
- **Automation of Image Generation**: In the initial prototype, each image was generated with the authors supervision, so automating the image generation process is essential. By streamlining operations from prompt input to image output, the aim is to minimize user wait times and optimize system performance.
- **Exploration of Prompt Engineering**: To enhance the relevance and quality of generated images, prompt engineering techniques need to be

further explored with respect to the automation methods previously discussed. These methods will analyze user inputs to extract and expand upon key themes, potentially integrating a Large Language Model (LLM) to refine the system’s interpretative capabilities.

With respect to Hardware plans:

- **Physical Enclosure Design**: A robust physical enclosure will be designed to house the touchscreen monitor, Raspberry Pi 5, and camera. This enclosure will protect the hardware from environmental factors and facilitate ease of transport and setup.
- **Sensor Vision Stack Development**: Further development of the sensor vision stack is planned. By enhancing the capabilities of these sensors, the aim is to improve the accuracy and quality of image calibration and alignment, which is crucial for precise image overlay on varied surfaces.

The authors approached all of the tasks/stages together as they progressed towards the complete integrated system of GPMS.

Chapter 11 Conclusion

Throughout the duration of Senior Design 1, the authors learned first hand the importance of planning strategically and collectively to design a Generative Projection Mapping System, GPMS, capable of immersing all audiences through projection mapping.

The authors were ecstatic to continue on an ambitious project aimed at creating GPMS, which seeked to revolutionize storytelling environments through dynamic image projection onto various structures. Their primary goal was to continue utilizing advanced computer vision techniques to intelligently detect and project images onto unique features such as pillars, windows, or drawings. GPMS integrated a responsive touchscreen monitor to allow users to input prompts which initiated the generation of images that seamlessly aligned with the structure's features.

In their pursuit of enhancing user customization, the authors aimed to implement sensitivity and threshold controls to increase user control over image representation which empowered all users to intricately adjust the outline of the structure according to their preferences. These enhancements ensured a personalized experience to all users.

Furthermore, the authors enabled users to highlight specific structural features through an intuitive interface, allowing for a personalized visual experience. They also aimed to streamline the user experience by integrating an automatic calibration process into the device, minimizing user input and ensuring optimal image display without unnecessary complexity. The authors met weekly for the entirety of the GPMS development journey.

The authors met the set deadlines to ensure that there was ample time in accomplishing the project adequately, including all the necessary and time consuming debugging. In conclusion, the authors' project pushed the boundaries of immersive storytelling environments while providing users with unprecedented control and customization options throughout the completion of Senior Design 2.

Appendix

Appendix A - Pledge of Originality

We, the members of Senior Design Group B of the University of Central Florida, hereby pledge and affirm that our final report, submitted in partial fulfillment of the requirements for the EEL 4914C and EEL 4915L courses, represents our own original work.

All ideas and findings presented herein are the result of our collective effort and intellectual endeavor. Where we have consulted the work of others, it is duly acknowledged within the body and Appendix-B of the report, ensuring that credit is given where it is due.

This document is a product of our collaboration as a group. Each team member has contributed significantly to the conceptualization, research, analysis, and writing processes. We have openly shared knowledge and worked together to overcome challenges, ensuring a comprehensive and cohesive final product. We have adhered strictly to the ethical guidelines provided by our Faculty Mentors and the University of Central Florida's Academic Honor Code. We have made every effort to document our methodologies, data sources, and analysis techniques transparently, allowing for the reproducibility and scrutiny of our work.

By signing this pledge, we commit to the truthfulness of the above statements.

[Signed]

Francisco Soriano Declan Carter Victoria Moreno

Appendix B - References

[1] OpenAI. “ChatGPT.” Chat.openai.com, OpenAI, 30 Nov. 2022, chat.openai.com/.

[2] EVERY Disney World and Disneyland Castle TRANSFORMATION in HISTORY! | the Disney Food Blog. 26 May 2020, www.disneyfoodblog.com/2020/05/26/every-disney-world-and-disneyland-castle-transformation-in-history/. Accessed 30 Jan. 2024.

[3] Sylt, Christian. “The Magic behind the Scenes of Universal’s Harry Potter Show.” Forbes, www.forbes.com/sites/csylt/2020/03/07/the-magic-behind-the-scenes-of-universals-harry-potter-show/. Accessed 30 Jan. 2024.

[4] Sandvik, A., & Zhang, J. (2023). Dynamic Projection Mapping A Senior Project presented to. https://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?article=1377&context=cpesp

[5] Luma Box. (n.d.). Projection Mapping. Luma Box. https://www.lumabox.com/

[6] Chauhan, V. K., Zhou, J., Lu, P., Molaei, S., & Clifton, D. A. (2023, August 10). *A brief review of Hypernetworks in deep learning*. arXiv.org. https://arxiv.org/abs/2306.06955#:~:text=Hypernetworks%2C%20or%20hypernets%20in%20short,sharing%2C%20and%20model%20compression%20etc.

[7] Zhang, L., Rao, A., & Agrawala, M. (2023, November 26). *Adding conditional control to text-to-image diffusion models*. arXiv.org. https://arxiv.org/abs/2302.05543 https://github.com/lllyasviel/ControlNet

[8] Chauhan, V. K., Zhou, J., Lu, P., Molaei, S., & Clifton, D. A. (2023, August 10). *A brief review of Hypernetworks in deep learning*. arXiv.org. https://arxiv.org/abs/2306.06955#:~:text=Hypernetworks%2C%20or%20hypernets%20in%20short,sharing%2C%20and%20model%20compression%20etc.

[9] MCL_Admin. (2021, April 6). FPGA vs. Microcontroller | What is the Difference? Mcl. https://www.mclpcb.com/blog/fpga-vs-microcontroller/

[10] Gregersen, C. (2023, September 6). *A Complete Guide to Development Boards*. Nabto. https://www.nabto.com/development-board-guide/

[11] *How to choose a microcontroller for your task: the pros and cons of various MCUs*. (2022, May 4). 2Smart. https://2smart.com/docs-resources/articles/how-to-choose-a-microcontroller-for-your-task-the-pros-and-cons-of-various-mcus

[12] Raspberry Pi Documentation - Raspberry Pi Hardware. (n.d.). Www.raspberrypi.com. https://www.raspberrypi.com/documentation/computers/raspberry-pi.html

[13] ESP32-WROOM-32 Datasheet. (n.d.). https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf

[14] MSP-EXP430FR6989 | Buy TI Parts | TI.com. (n.d.). Www.ti.com. Retrieved February 17, 2024, from

https://www.ti.com/product/MSP-EXP430FR6989/part-details/MSP-EXP430FR6989#:~:text=Features%20for%20the%20MSP-EXP430FR6989%201%20MSP%20ULP%20FRAM-based

[15] Amazon.com. (n.d.). Projector, Video Projector Multimedia Compatible Smartphone. Retrieved April 11, 2024, from https://www.amazon.com/Projector-Video-Projector-Multimedia-Compatible-Smartphone/dp/B07MTCMHZX/

[16] Amazon.com. (n.d.). Projector, Bluetooth 100Screen Compatible Smartphone. Retrieved April 11, 2024, from https://www.amazon.com/Projector-Bluetooth-100Screen-Compatible-Smartphone/dp/B0B28G5Y4R/

[17] Amazon.com. (n.d.). Bluetooth Projector Supported Portable Compatible. Retrieved April 11, 2024, from https://www.amazon.com/Bluetooth-Projector-Supported-Portable-Compatible/dp/B0BGXKQ4MP/

[18] Microsoft. (2024). Microsoft Copilot: Your everyday AI companion. Copilot.microsoft.com. https://copilot.microsoft.com/

[19] Kumar, P. (2021, November 11). *MIPI camera vs USB camera – a detailed comparison*. E-Con Systems. https://www.e-consystems.com/blog/camera/technology/mipi-camera-vs-usb-camera-a-detailed-comparison

[20] Davis, C. (2023, July 21). *4 touch panel types - explained*. ViewSonic Library. https://www.viewsonic.com/library/business/touch-panel-types-explained/

[21] What is Google's Gemini AI tool (formerly Bard)? Everything you need to know. (n.d.). ZDNET. https://www.zdnet.com/article/what-is-googles-gemini-ai-tool-formerly-bard-everything-you-need-to-know/

[22] Ashtari, H. (2023, March 13). *Unix vs. Linux vs. windows: How they compare: Spiceworks - Spiceworks*. Spiceworks Inc. https://www.spiceworks.com/tech/tech-101/articles/unix-linux-windows-comparison/

[23] G., D. (2023, December 1). *Centos vs ubuntu - which one to choose for your web server*. Hostinger Tutorials. https://www.hostinger.com/tutorials/centos-vs-ubuntu

[24] Dillon, L. (2020, July 15). *CentOS vs. ubuntu: Feature comparison and use cases*. OpenLogic by Perforce. https://www.openlogic.com/blog/centos-vs-ubuntu

[25] Bonuccelli, G. (2022, February 15). *Cloud vs server: Learn the key differences and benefits*. Server and Cloud Blog. https://www.parallels.com/blogs/ras/cloud-vs-server/

[26] Wassim. (2023, September 13). *Comparing desktop application development frameworks: Electron, flutter, Tauri, react native, and...* Medium. https://medium.com/@maxel333/comparing-desktop-application-development-frameworks-electron-flutter-tauri-react-native-and-fd2712765377

[27] How HDMI Works. (2007, October 8). HowStuffWorks. https://electronics.howstuffworks.com/hdmi.htm#:~:text=HDMI%20uses%20transition%20minimized%20differential%20signaling%20%28TMDS%29%20to

[28] Cloudflare. (2021). What is SSL (Secure Sockets Layer)? | Cloudflare. Cloudflare. https://www.cloudflare.com/learning/ssl/what-is-ssl/

[29] Bhakhra, S. (2019, June 10). Secure Socket Layer (SSL) - GeeksforGeeks. GeeksforGeeks. https://www.geeksforgeeks.org/secure-socket-layer-ssl/

[30] *Projector Specs Explained – Your Ultimate Projector Buyer's Guide! (2024)*. (2023, September 12). https://theaterdiy.com/projector-specs-explained-ultimate-projector-buyers-guide/

[31] *Camera Measurement Standards*. (n.d.). Www.imaging.org. Retrieved April 10, 2024, from https://www.imaging.org/IST/IST/Standards/Camera_Measurement_Standards.aspx

[32] Roller, J. (2023, December 27). Ethical Pros and Cons of AI Image Generation. IEEE Computer Society. https://www.computer.org/publications/tech-news/community-voices/ethics-of-ai-image-generation

[33] Ortiz, S. (2023, September 15). What is ChatGPT and why does it matter? Here's everything you need to know. ZDNET; ZDNET. https://www.zdnet.com/article/what-is-chatgpt-and-why-does-it-matter-heres-everything-you-need-to-know/

[34] What is Copilot (formerly Bing Chat)? Here's everything you need to know. (n.d.). ZDNET. https://www.zdnet.com/article/what-is-copilot-formerly-bing-chat-heres-everything-you-need-to-know/

[35] *Raspberry Pi Documentation - Raspberry Pi Hardware*. (n.d.). Www.raspberrypi.com. https://www.raspberrypi.com/documentation/computers/raspberry-pi.html

[36]"OpenCV: Camera Calibration with OpenCV." *Docs.opencv.org*, docs.opencv.org/3.4/d4/d94/tutorial_camera_calibration.html.

[37] How to Draw Pusheen with Donut 🍩 Easy Drawing on a Whiteboard. (2019, November 8). www.youtube.com. https://www.youtube.com/watch?v=zxxJWL2TZEA

[38] Datta, S., Ku, A., Ramachandran, D., & Anderson, P. (2023, December 27). *Prompt expansion for adaptive text-to-image generation*. arXiv.org. https://arxiv.org/abs/2312.16720

[39] Illyasviel. (2023, May 16). ControlNet 1.1. GitHub.

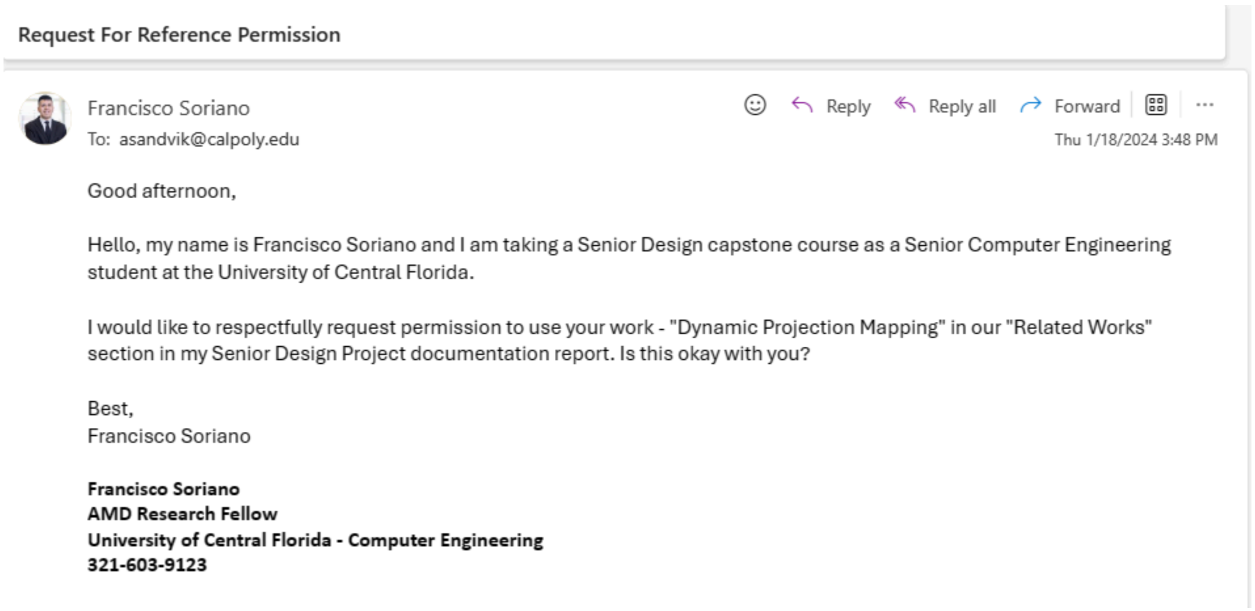
[40] *Canny edge detection*. OpenCV. (n.d.). https://docs.opencv.org/4.5.2/da/d22/tutorial_py_canny.html

[41] *CV::Linesegmentdetector class reference*. OpenCV. (n.d.-b). https://docs.opencv.org/4.5.2/db/d73/classcv_1_1LineSegmentDetector.html

[42] Xie, S., & Tu, Z. (2015, October 4). *Holistically-nested edge detection*. arXiv.org. https://arxiv.org/abs/1504.06375

[43] jira about. (n.d.). Bing. Retrieved March 10, 2024, from https://www.bing.com/search?q=jira+about&qs=n&form=QBRE&sp=-1&pq=jira+about&sc=11-10&sk=&cvid=5CB8D38767B8473DB9907FE67C7F3538&ghsh=0&ghacc=0&ghpl=

Appendix C - Copyright Permissions


Request For Reference Permission

Francisco Soriano
To: asandvik@calpoly.edu

Reply Reply all Forward

Thu 1/18/2024 3:48 PM

Good afternoon,

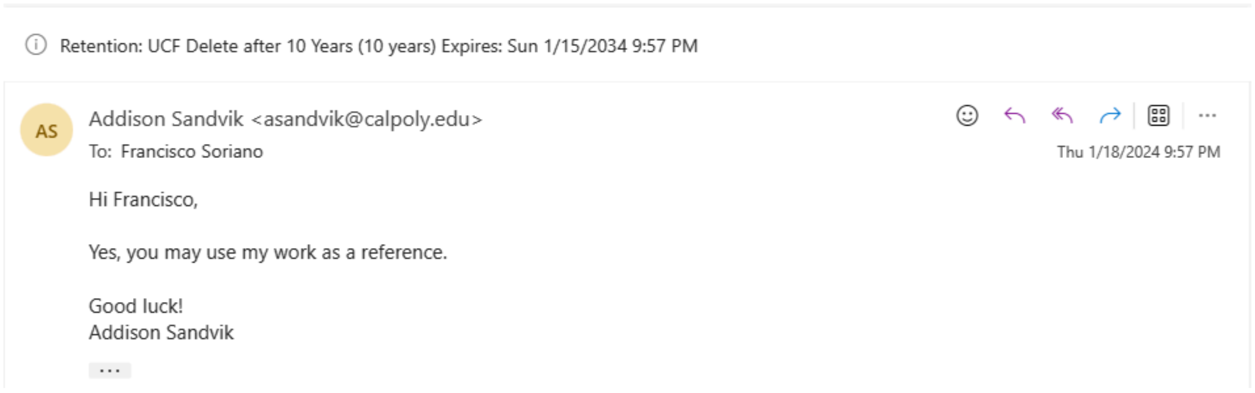
Hello, my name is Francisco Soriano and I am taking a Senior Design capstone course as a Senior Computer Engineering student at the University of Central Florida.

I would like to respectfully request permission to use your work - "Dynamic Projection Mapping" in our "Related Works" section in my Senior Design Project documentation report. Is this okay with you?

Best,
Francisco Soriano

Francisco Soriano
AMD Research Fellow
University of Central Florida - Computer Engineering
321-603-9123

Figure C.1 *Copyright Request Email Screenshot*


Retention: UCF Delete after 10 Years (10 years) Expires: Sun 1/15/2034 9:57 PM

AS Addison Sandvik <asandvik@calpoly.edu>
To: Francisco Soriano

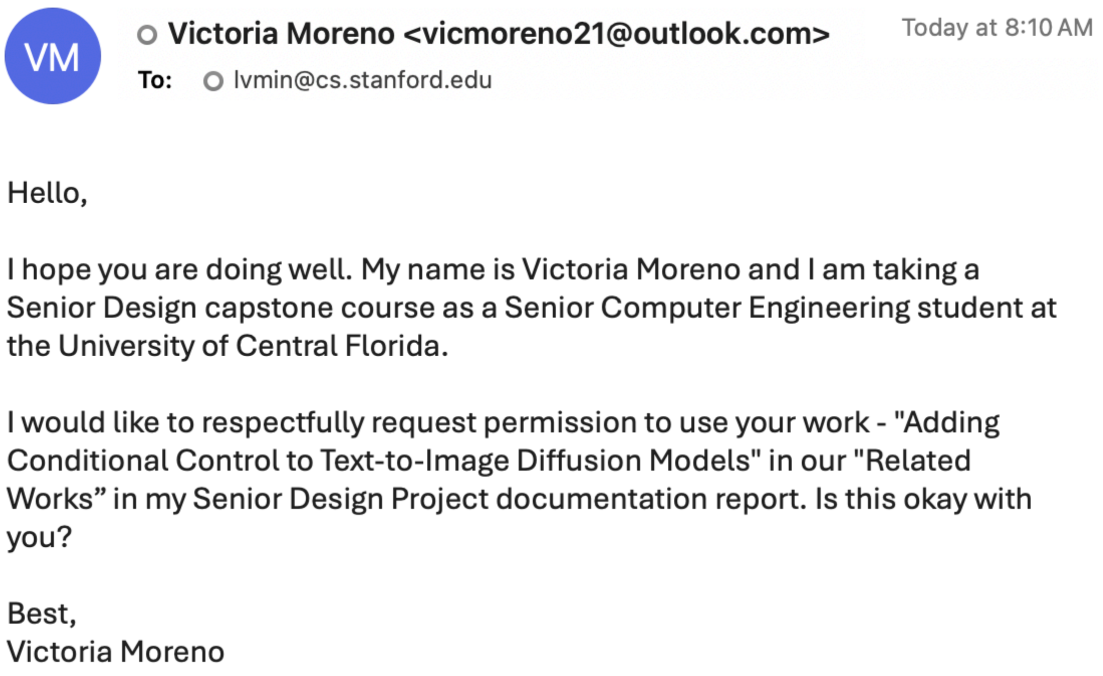
Thu 1/18/2024 9:57 PM

Hi Francisco,

Yes, you may use my work as a reference.

Good luck!
Addison Sandvik

Figure C.2 *Copyright Response Email Screenshot*



VM

Victoria Moreno <vicmoreno21@outlook.com> Today at 8:10 AM

To: lvmin@cs.stanford.edu

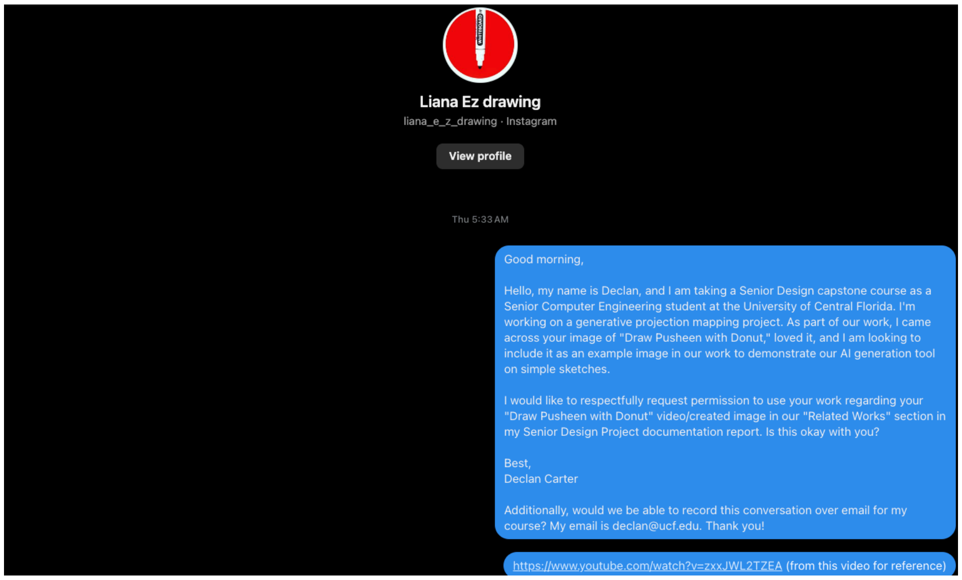
Hello,

I hope you are doing well. My name is Victoria Moreno and I am taking a Senior Design capstone course as a Senior Computer Engineering student at the University of Central Florida.

I would like to respectfully request permission to use your work - "Adding Conditional Control to Text-to-Image Diffusion Models" in our "Related Works" in my Senior Design Project documentation report. Is this okay with you?

Best,
Victoria Moreno

Figure C.3 *Copyright Request Email Screenshot*



Good morning,

Hello, my name is Declan, and I am taking a Senior Design capstone course as a Senior Computer Engineering student at the University of Central Florida. I'm working on a generative projection mapping project. As part of our work, I came across your image of "Draw Pusheen with Donut," loved it, and I am looking to include it as an example image in our work to demonstrate our AI generation tool on simple sketches.

I would like to respectfully request permission to use your work regarding your "Draw Pusheen with Donut" video/created image in our "Related Works" section in my Senior Design Project documentation report. Is this okay with you?

Best,
Declan Carter

Additionally, would we be able to record this conversation over email for my course? My email is declan@ucf.edu. Thank you!

https://www.youtube.com/watch?v=zxxJWL2TZEA (from this video for reference)

Figure C.4 *Copyright Request Screenshot*
(Email Could Not Be Located)

Appendix D - Code

```
# Declan Carter; UCF '24
# Senior Design 1; GPMS Prototype
# Description:
#       This script configures and uses Picamera2 for capturing video streams and applies real-time edge detection.
#       It features a dual-window setup for point selection and edge detection visualization.
#       Users can interactively select corner points on a white screen which will then be used to warp the perspective
#       of the captured frames. The script also allows for dynamic adjustment of edge detection thresholds via keyboard inputs.
# Usage:
#       1. Run the script in an environment that supports cv2 and Picamera2.
#       2. Drag the 'Point Selection' window to the desired screen as instructed by the console.
#       3. Select points on the 'Point Selection' window by clicking the left mouse button.
#       4. Press 'u' to capture the current frame and use it for edge detection.
#       5. Adjust the hi and lo edge detection thresholds using 'o'/'p' and 'k'/'l' keys respectively.
#       6. Press 'q' to quit the script.
#       7. Additional functions include resetting selected points ('r'), applying a watermark ('c'), and resetting thresholds to default ('d').

import cv2
from picamera2 import Picamera2, Preview
import numpy as np
import datetime
import time

# Initialize Picamera2
picam2 = Picamera2()

# Capture the Resolution
width = picam2.sensor_resolution[0]
height = int(width * (9/16))  # The display will be 16:9

# Set Image Resolution
picam2.preview_configuration.main.size = (width, height)
picam2.preview_configuration.main.format = "RGB888"
picam2.preview_configuration.align()
picam2.configure("preview")
picam2.start()

# Create Window 2
cv2.namedWindow('Edge Detection', cv2.WINDOW_NORMAL)
cv2.setWindowProperty('Edge Detection', cv2.WND_PROP_FULLSCREEN, cv2.WINDOW_FULLSCREEN)

# Create Window 1, wait for user to drag it to display 2
white_screen = np.full((600, 800, 3), 255, dtype=np.uint8)
cv2.namedWindow('Point Selection', cv2.WINDOW_NORMAL)
```

```
cv2.imshow('Point Selection', white_screen)
# Display instruction
print("Please drag the window to the desired screen and press any key to
continue...")

cv2.waitKey(0)  # Wait for user input

# Default edge detection thresholds
low_threshold = 100
high_threshold = 200
default_low_threshold = low_threshold
default_high_threshold = high_threshold

# Default corner points and state
default_pts_webcam = np.float32([[0, 0], [0, 720], [1280, 720], [1280,
0]])
new_pts = []
reset_mode = True
matrix = None

# Mouse callback function
def select_point(event, x, y, flags, param):
   global new_pts
   if event == cv2.EVENT_LBUTTONDOWN and len(new_pts) < 4:
       new_pts.append([x, y])

# Set mouse callback
cv2.setMouseCallback('Point Selection', select_point)

# Function to reset the corner points
def reset_points():
   global new_pts, reset_mode
   reset_mode = True
   new_pts = []
   cv2.setMouseCallback('Point Selection', select_point)

# Function to calculate the perspective transformation matrix
def get_matrix():
    pts_projection = np.float32([[0, 0], [0, 720], [1280, 720], [1280,
0]])
               return    cv2.getPerspectiveTransform(np.float32(new_pts),
pts_projection)

# Function to apply Canny edge detection
def apply_edge_detection(frame, low, high):
   gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
   blurred = cv2.GaussianBlur(gray, (5, 5), 0)
   edges = cv2.Canny(blurred, low, high)
   return edges

# Function to add watermark to the frame
def add_watermark(frame, text):
   font = cv2.FONT_HERSHEY_SIMPLEX
     cv2.putText(frame, text, (10, 700), font, 1, (255, 255, 255), 2,
cv2.LINE_AA)
   return frame
```

```
# Function to capture and save frame with timestamp
def capture_frame_with_timestamp(frame):
     print("Attempting to save the frame:", type(frame), frame.shape if
isinstance(frame, np.ndarray) else "Not a frame")
    if isinstance(frame, np.ndarray) and frame.size != 0:
        timestamp = datetime.datetime.now().strftime("%Y%m%d_%H%M%S")
        filename = f'captured_frame_{timestamp}.jpg'
        cv2.imwrite(filename, frame)
        print(f"Frame captured and saved as {filename}.")
    else:
        print("Error: No valid frame to save.")

def capture_frame():
    frame = picam2.capture_array()
    if frame is None:
        print("Failed to capture video.")
        return None

    # Flip the frame for both point selection and edge detection
    frame = cv2.flip(frame, 0)
    frame = cv2.flip(frame, 1)

    return frame



while True:
    frame_ps = capture_frame()

    if reset_mode:
        if new_pts != []:
            for pt in new_pts:
                cv2.circle(frame_ps, tuple(pt), 5, (0, 255, 0), -1)
        if len(new_pts) == 4:
            matrix = get_matrix()
             still_frame = frame_ps.copy()  # Capture still frame for edge
detection
            reset_mode = False
        frame_ed = np.full((600, 800, 3), 255, dtype=np.uint8)
    else:
        if matrix is not None and still_frame is not None:
             transformed = cv2.warpPerspective(still_frame, matrix, (1280,
720))
              frame_ed = apply_edge_detection(transformed, low_threshold,
high_threshold)
        else:
            frame_ed = np.full((600, 800, 3), 255, dtype=np.uint8)

    cv2.imshow('Point Selection', frame_ps)
    cv2.imshow('Edge Detection', frame_ed)

    key = cv2.waitKey(1) & 0xFF
    if key == ord('q'):
        break
    elif key == ord('o'):
        high_threshold = max(high_threshold - 10, 0)
    elif key == ord('p'):
        high_threshold = min(high_threshold + 10, 255)
    elif key == ord('k'):
        low_threshold = max(low_threshold - 10, 0)
    elif key == ord('l'):
        low_threshold = min(low_threshold + 10, 255)
    elif key == ord('r'):
        reset_points()  # Reset the points
          frame_ed = np.full((600, 800, 3), 255, dtype=np.uint8)   # Edge
detection shows white
    elif key == ord('u'): # Update Frame being used for edge detection
        # Flash White Frame to Capture Image
        frame_ed = np.full((600, 800, 3), 255, dtype=np.uint8)
        cv2.imshow('Edge Detection', frame_ed)
        cv2.waitKey(3000)  # Wait for 3 seconds with GUI responsive
        still_frame = capture_frame()
    elif key == ord('c'):
        watermark_text = f'Low: {low_threshold}, High: {high_threshold}'
                   frame_with_watermark   =   add_watermark(frame_ps.copy(),
watermark_text)
        capture_frame_with_timestamp(frame_with_watermark)
        print("Frame captured with watermark.")
    elif key == ord('d'):
        low_threshold = default_low_threshold
        high_threshold = default_high_threshold
        print("Thresholds reset to default values.")

# Release the webcam and destroy all OpenCV windows
picam2.stop()
cv2.destroyAllWindows()
```

Appendix E - Figure

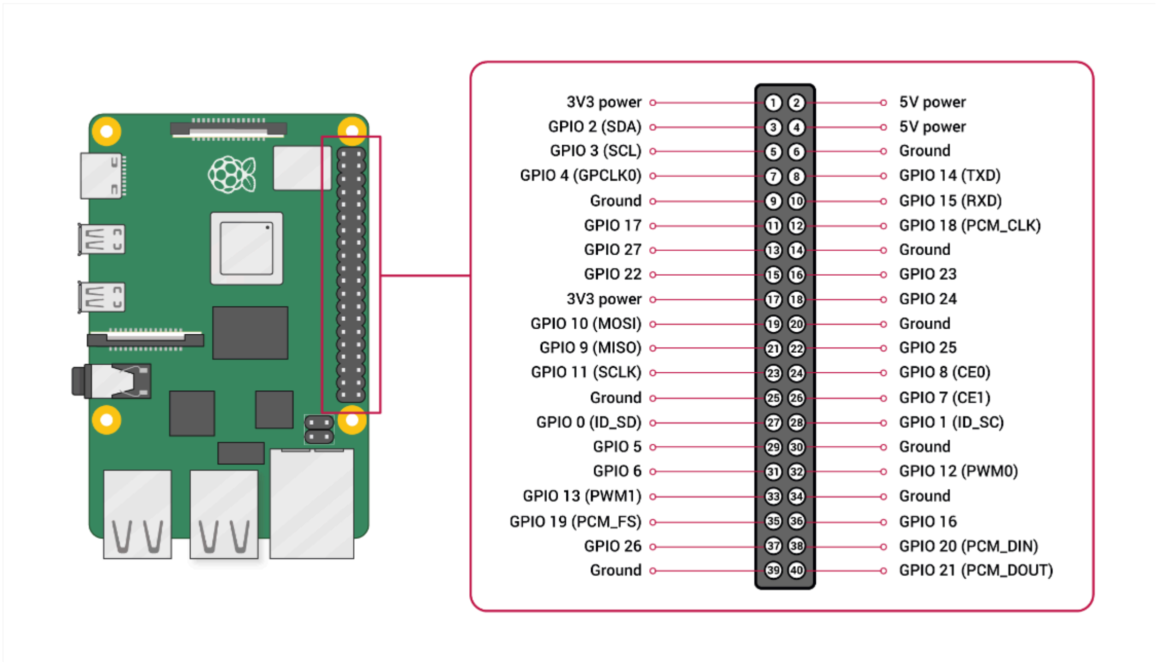


Figure E.1 *Raspberry Pi 4B GPIO Layout* [35]

Appendix F - Tables

The BOM table below is specifically for the AMD machine. AMD generously lended the authors compute time on the AMD machine.

ITEM	**QUANTITY**	**PRICE ESTIMATE (USD)**
CPU*	1	516.54
CPU Cooler*	1	128.64
GPU*	1	1,049.99
Motherboard*	1	259.99
RAM*	1	349.99
Power Supply*	1	489.32
Case*	1	139.99
Storage OS*	1	159.99
Storage APPS*	1	149.99

Table F.1 *AMD Machine BOM*