Smart Mirror

Georgiy Brussenskiy, Christopher Chiarella, and Vishal Nagda

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — With the increasing integration of technology into our daily lives, maintaining an efficient schedule has become both easier and more difficult. The Smart Mirror is designed to work within our schedule and not be an extra piece to it. The requirements and specifications were inspired by both technology seen in movies as well as the devices people use everyday such as PCs, tablets, and smartphones. The Smart Mirror contains six-applications that give the user the information they need to plan their day while being easy to use thanks to the touch-free UI and worry-free self-monitoring.**

***Index Terms* — Applications, Leap Motion, Smart Home, Smart Mirror**

I. Introduction

Efficiency and productivity are increasingly establishing their dominance as the keywords companies are using to market their products. The fact that their product can multitask or increase productivity better than the competition is becoming a real selling point. This is because effective time management is an essential factor in being productive. The best time management strategies involve being able to find time where there was no time before. Integration of technology into people's daily lives has made that time management possible. The use of products such as tablets, PCs, and smartphones have given people access to the tools needed to be productive.

However successful technology products have helped productivity, it has done its fair share to stifle it as well. The use of technology has become another task on everyone's daily to-do list. Technology should mold to our schedule, not the other way around. That is where the smart mirror idea originated. The smart mirror idea aimed to integrate technology seamlessly into people's lives by putting it where everyone's routine eventually collides, the bathroom. The goal of the smart mirror is to increase a user's productivity by saving them time. The smart mirror will provide a near effortless experience that allows the user to just walk up and be greeted with information they would typically need another device for. Despite the fact this information can be found on the user's other devices, it's the timesaving convenience of having this information available during the typical bathroom routines.

II. System components

The smart mirror project is a collection of purchased and designed components that work together to make this project work the way it does. Each of these components provides an essential function in the overall experience and this section will give descriptions of these pieces.

A. Leap Motion Controller

The Leap Motion contains two IR cameras and three infrared LEDs that use infrared reflections to determine hand movement and individual fingers to translate to gestures [1]. The controller is only 0.1 pounds and 3 inches long. The controller uses its cameras to translate movement and depth of every finger using a 150 degree side-to-side field of view and a 120 degree field of depth view. It is a USB peripheral and due to its small size, it can be placed within the bathroom environment discretely and unobtrusively.

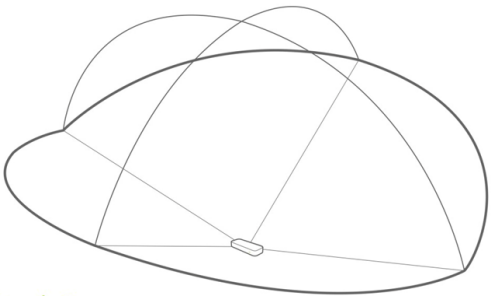

Fig. 1. Leap Motion Interaction Area

B. Temperature Sensor

The circuit compares the sensed temperature with the reference temperature in order to determine if the PC needs to be turned off. The temperature sensor selected is the TI LM35 For example, if the reference temperature is set at 40 degrees Celsius, then when the sensed temperature reaches that limit, the relay is activated. The temperature sensor is located on the same PCB as the humidity sensor for space and cost savings. Its data is used in determining when to shut down the PC under more extreme conditions. The MCU assists in sending this data.

C. Humidity Sensor

The same principle used from the temperature regulation circuit is applied to the humidity sensor circuit. The humidity sensor selected is the IST P-14. This principle involves the comparing of the sensor voltage with a reference voltage in order to determine whether or not to active a relay. For example, if the reference value is 75% humidity, then when that value is exceeded, the relay is activated.

D. Microcontroller

The microcontroller is a custom designed version of the Arduino Uno featuring the ATmega328p chip. The specifications are listed below in Table I. Our design uses the various components from the Arduino Uno but eliminates anything extra that isn't necessary for our project. This creates a more focused board to perform the needs required. The main use is to read the data returned from both the temperature and humidity sensors to be interpreted on the chip. This data will also be transferred serially to the central PC so that the user can be informed of any extreme values being read.

TABLE I
ATMEGA328P SPECIFICATIONS

Clock Speed	16 MHz
Voltage	7-12 V
EEPROM	1 KB
SRAM	2 KB

E. Webcam

The main function of the webcam for the smart mirror is to detect the presence of user in front of the mirror. Once detected, the smart mirror will "wake up" from hibernating and display the applications on the screen. The webcam also determines when the user leaves, using a two minute waiting period, to then put the smart mirror back into hibernation in order to conserve power and resources. The second function of the webcam is use of its built-in microphone that will be utilized in the voice recognition system. The webcam chosen to use for the smart mirror is the Logitech C920 HD Pro Webcam.

III. SYSTEM OVERVIEW

To get an overview and better understanding for the smart mirror project, system representations in the form of block diagrams were drawn out to be able to see the "whole picture". Since this project involves a significant amount of software, both a hardware and software block diagram were created. The block diagrams provide a top level view on how each part of the system interacts. The following sections will cover the hardware and software system overviews.

A. Hardware

The blocks in the hardware block diagram or system block diagram for this project represent an important entity. Figure 2 below shows how all the components relay through the central pc, which is the main driver for the smart mirror. The MCU is the secondary processing unit which handles the temperature data. The sensors monitor their own data and it's interpreted by the MCU. The arrows show how each block is interacting with the next. Most blocks are either an input or output in this diagram.

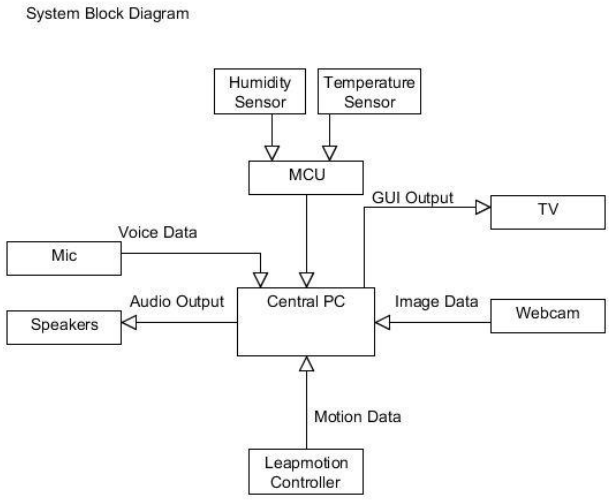

Fig. 2. Smart Mirror System Block Diagram

B. Software

The software block diagram gives the complete overview for all the software interactions for the smart mirror. Seen in Figure 3 below, the MCU will take care of all the temperature data processing that is used in the temperature regulation system. That data is then transferred to the central PC. The central PC takes care of the user gesture input, webcam image processing, voice recognition processing, and rendering the GUI to the user. The most demanding code is run on the central PC software since it has the most power.

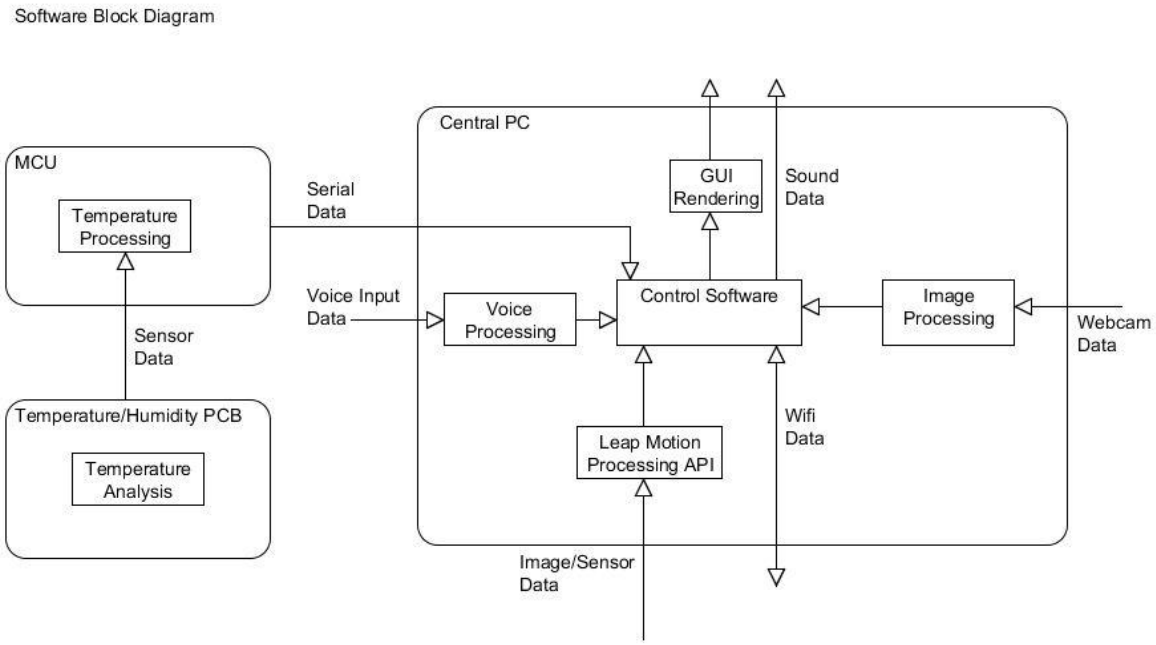

Fig. 3. Smart Mirror Software Block Diagram

IV. HARDWARE DETAIL

This section focuses on both the primary purchased and designed hardware aspects of the smart mirror. Such pieces include the control PC, the custom microcontroller, and the sensors that make up the temperature regulation system.

A. Control PC

The control computer is the main central processing part of the smart mirror project. All the data and control signals run through the control computer. The control computer is essentially a compact PC. It is running the Windows 8.1 operating system where the designed Windows 8 application controls the mirror and provides the GUI to the screen. The control computer is made up of a mini ITX Intel motherboard that has the modern standard of USB 3.0 ports, audio ports, and video output including HDMI. The graphics are rendered using integrated graphics on the Intel Haswell i3 processor. The motherboard also houses 4GB of DDR3 RAM as well as a 64GB solid-state drive. Finally, a 380 watt power supply powers the computer. A mini ITX computer case contains all the parts mentioned and provides appropriate air cooling to the electronics. All these parts will be contained within the designed temperature regulation system to provide maximum protection against temperature and humidity damage to the electronics. The control PC provides extra fans for protection against the temperature and humidity.

B. Microcontroller

As described earlier, the smart mirror is utilizing a custom designed version of the Arduino Uno featuring the ATmega328p chip. Only the components of the Arduino Uno that were necessary to the functionality needed were preserved in the final design. The key functionality needed was use of the ATmega328p for manipulating the temperature and humidity data. Other functions needed also were the use of an analog pin to connect to the PCB as well as the USB and UART components in order to have communication with the central PC. The final design from the Eagle file can be seen below in Figure 4.

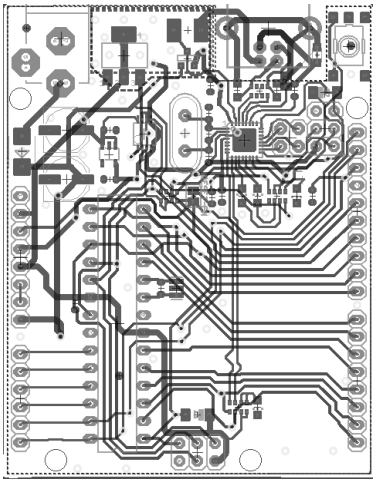

Fig. 4. Custom MCU Eagle Layout

C. Temperature Regulation System

The temperature regulation system involves two components: the MCU, and the temperature and humidity PCB. The system has been designed to be within +/- 1 °C accuracy for the temperature and +/- 1.5% accuracy for the humidity. Comparators are used to compare two voltages, determined by the reference voltage and the voltage that is produced in relation to the sensor. The temperature sensor used in the system is the TI LM35 due to its fast response time as well as being highly accurate [2]. The humidity sensor used is the IST P-14 which allows for high humidity stability coupled with low linearity error and fast response time [3]. The temperature and humidity PCB is wired to the MCU. The MCU reads the sensors data in order to then transmit that data to the central PC in order for a warning to be admitted to the user if any temperature or humidity level is reached that is unsafe for the computer hardware. If the levels are extreme, the system shuts down. The temperature and humidity circuit from Multisim design software can be seen in Figure 5 below. The same circuit represented in the final Eagle layout configuration is in Figure 6 below.

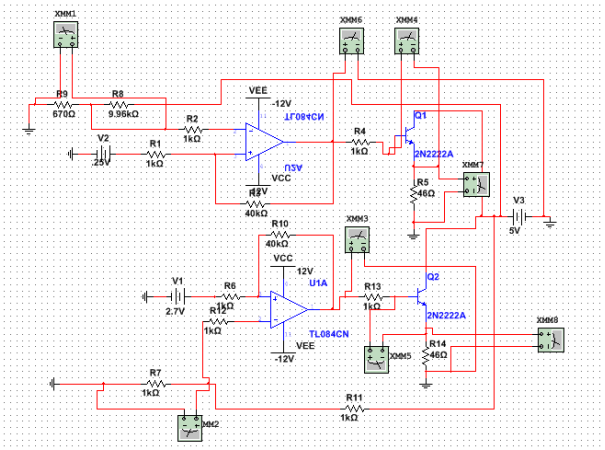

Fig. 5. Temperature/Humidity Circuit in MultiSim

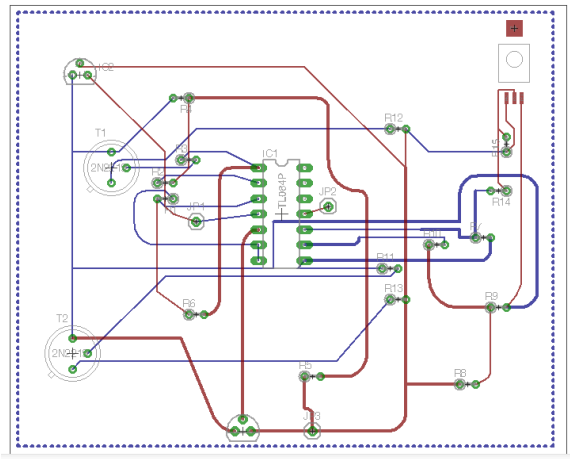

Fig. 6. Temperature/Humidity Eagle Layout

V. SOFTWARE DETAIL

This section lays out the details of the software that keeps the smart mirror running. Such topics include the software environment used to create the smart mirror, the types of tasks handled by the central PC, and how the UI works in relation to the user interaction.

A. Software Architecture

The IDE chosen to be used for the smart mirror project is Microsoft Visual Studio 2013. For this project, the primary language was visual C++ coupled with XAML for the visual user interface. XAML is a "declarative markup language" which essentially makes for creating a UI faster and easier [4]. The GUI was all designed and implemented using XAML. Having access to the Windows API allowed for easy integration of the various peripherals including the webcam, Leap Motion, and MCU. In addition, other APIs and SDKs were utilized. The AT&T Speech API was used to translate the recorded voice commands to text to be used to perform the actions specified. The Toodledo API provided instructions on how to access the user's task list and add to-do list items programmatically. The Twitter API, similarly, allowed the smart mirror to gain access to the user's twitter account, import recent tweets, and post to their timeline on their behalf. Finally, the Leap Motion SDK allowed for quick integration of the leap motion controller and interpretation of any gestures performed by the user.

B. Application Task Processing

The processing of all these data was divided into two task processing sections. These sections are the foreground UI thread and the background asynchronous processing tasks.

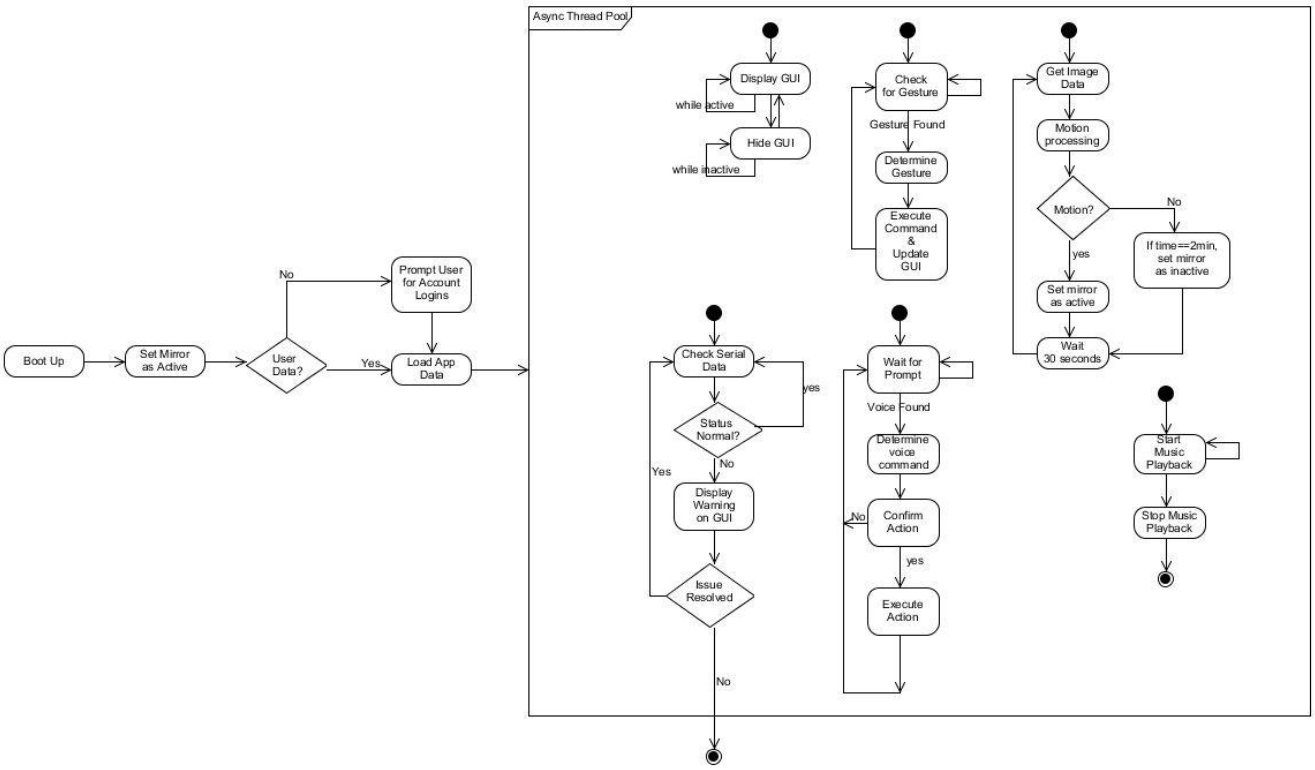

Fig. 7. Software State Diagram

The foreground task involves managing the GUI elements with the logic elements of the background tasks. The asynchronous aspect allows the user to continue using the mirror without the GUI freezing up. The multi-thread aspect suggested allows for a closer to parallel execute of each of the code than a linear programming technique would allow. This almost parallel-type style allows for the Leap Motion data, along with the voice processing and other processes, to be handled as soon as data is delivered. This is represented in the software state diagram in Figure 7 above.

C. Application Task Algorithms

There are three significant tasks that warrant analysis of how they are performed. These three include interpreting the leap motion data, the motion detection algorithm, and the speech recognition logic. First, the leap motion data is collected through a listener object setup by use of the Leap SDK. Through this the current frame of data is interpreted. Using the frame data, the number of hands and fingers are determined as well as their movement vector information. In addition, the SDK contains default gestures that the code is already setup to look for and recognize. These gestures include swipes, circles, key presses, and screen taps. A dispatch timer is setup to check the frame data for the leap motion every tenth of a second. Then based on what gesture or action is performed by the user, the code calls the appropriate function to execute that command.

The motion detection algorithm is what allows the mirror to determine if the user is present or not. The algorithm initially takes a reference image through an asynchronous image capture by the webcam. Next, either after 5 seconds if the mirror is inactive or 30 seconds if the mirror is active, another image is captured. These two images are transformed into two pixel arrays in which they are then compared one-by-one. If the pixel values are greater than the threshold, then a counter is incremented. Once all the pixels are checked, the number of pixels that qualified to have significant change is counted. This number is compared against a second threshold where the code then determines whether motion has been detected or not. The algorithm must conclude that there is no motion at least 4 times, or 2 minutes, before it turns the mirror screen to inactive. This algorithm can be seen in Figure 8 below.

The final task algorithm is in regards to the voice recognition. On boot-up, the webcam microphone is initialized in order to be prepared to record audio. When the user performs the action to begin voice control, an asynchronous function call is performed to start recording

audio. A dispatch timer allows for 5 seconds of recording before the audio file is saved as a single channel, WAV file. Next, this audio stream is embedded into a HTTP Post request to the AT&T speech server, which has been already authenticated. The server returns a JSON object containing the text of the transcoded audio file. This text is then searched for key action words that will allow the code flow to call the right function related to the command the user is requesting.

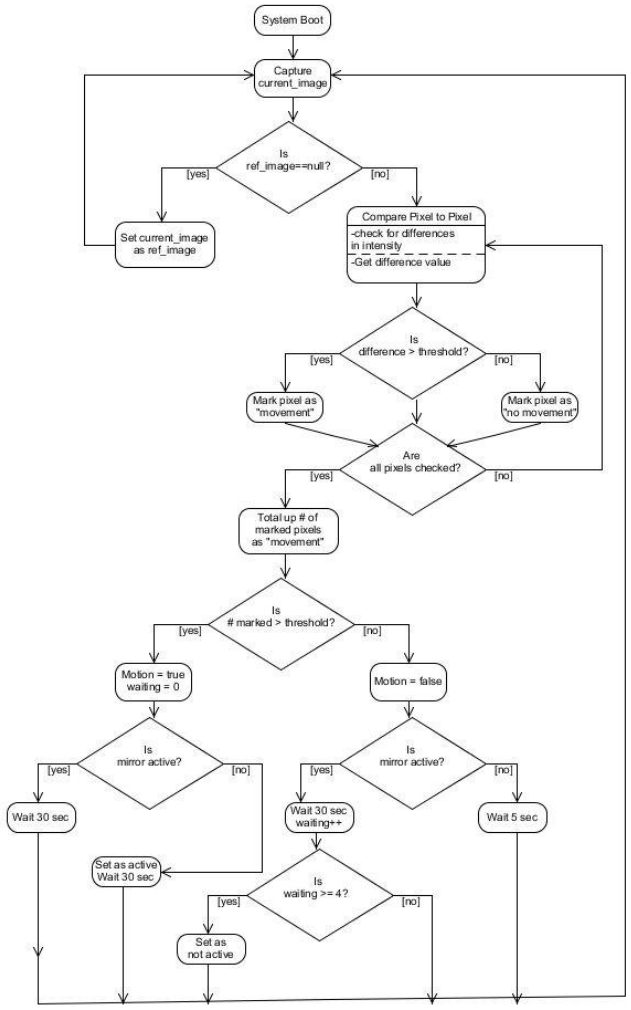

Fig. 8. Motion Detection State Diagram

D. User Interface and Applications

The user interface and GUI design are specifically designed with the user in mind. The state diagram of how the GUI reacts and works with the Leap Motion gestures can be seen in Figure 9 below. Here the system first boots up and the first item to display on the screen is the time. Next, the apps open to normal mode, the default boot up state. From here, depending on whether the gesture is a swipe or tap will determine what the GUI does next. The applications have three states: minimized, normal, and expanded. Each varies in terms of how much information is displayed on the screen. The correct swipe on the left-side of the screen will change those apps into the minimized state, leaving the right side in normal mode. The appropriate swipe on the right side will change those apps into a minimized state. The states are circular so the user can make either one side or the other normal to return to having all the apps in normal mode. When all are in normal mode, a tap on the app will put it into expanded mode. A consecutive tap on the app will return it to normal mode.

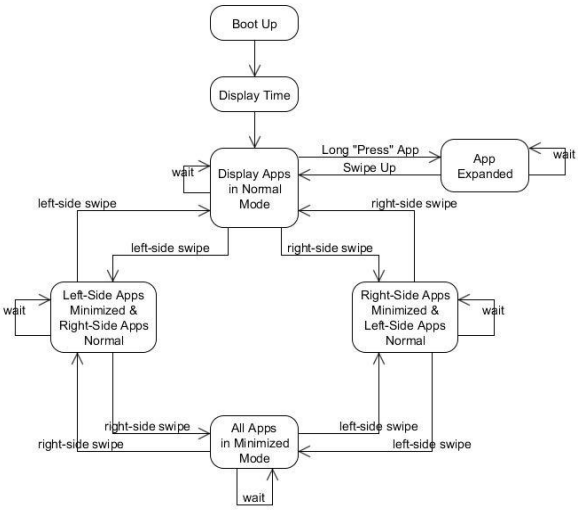

Fig. 9. GUI State Diagram

The design of the control software for the user focuses on six applications. These applications include weather, music, twitter, to-do list, calendar, and news. To briefly describe each, the weather provides the local weather for the day and up to the next two days while in expanded mode. The music app plays any music stored locally on the smart mirror hard-drive and is controlled through voice recognition. The Twitter app links with the user's Twitter account to display the most recent tweets from their timeline and allows the user to post to their tweeter feed with use of voice control as well. The to-do list syncs with popular iOS application, Toodledo, and also allows the user to add items to their list via voice recognition. The calendar app takes in the user's iCal file, from such sources as Google's calendar, and displays up to three days of events from their calendar. Finally, the news app

pulls in the top stores via Bing News and allows the user to preview the article as well while the app is in expanded mode.

E. User Interface Gesture Controls

The gesture controls are a combination of swipes, pointing, and circles through the use of the leap motion controller. Since the swipe gesture is the simplest to perform, they are the majority of the controls. The user is essentially swiping away and swiping back the applications between their normal and minimized state. The swipes focus on controlling what the user sees and wants to see. The user points and waits when they want to access the expanded state then swipes vertically to exit that state. The more complicated gesture, the circle, is used to either refresh the application data or begin voice input. The refresh applications circle gesture involves two rotations in the clockwise direction. More often or not, the applications will update correctly and promptly enough that the gesture won't be needed. There is always a possibility that the communication with the server misbehaviors so the option to manually refresh is a good option to have to avoid user frustration. As for the voice, the circle gesture involves two rotations in the counter-clockwise direction. A collection of all the gestures can be seen in Table II below.

TABLE II
SUPPORTED GESTURES

Left Swipe (Left of Screen Center)	Changes Apps on left side from normal to minimized mode
Right Swipe (Left of Screen Center)	Changes Apps on left side from minimized to normal mode
Left Swipe (Right of Screen Center)	Changes Apps on right side from minimized to normal mode
Right Swipe (Right of Screen Center)	Changes Apps on right side from normal to minimized mode
Vertical Swipe	Exits App Expanded mode
Long-Point	Enters App Expanded mode
Clockwise Circle	Refreshes All App Data
Counter-Clockwise Circle	Activates Voice Recognition

VI. HOUSING

The housing is what encases all the components of the smart mirror. Cost and material were the main factors in the decision making process. The material had to be humidity resistant and keep the cost of the project within limits. The original concept design of the frame can be seen below in figure 10.

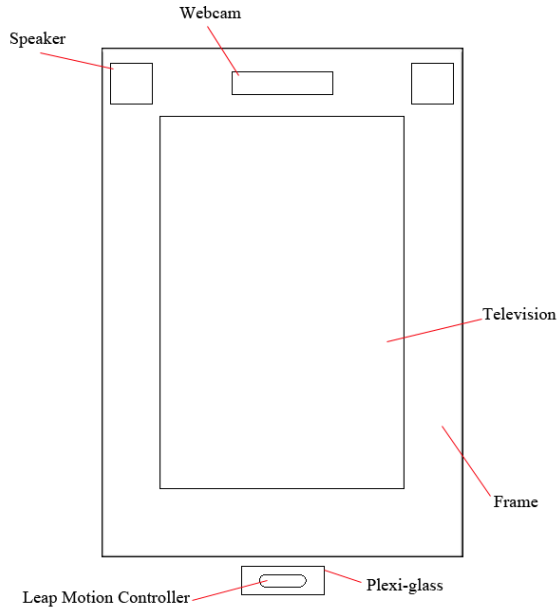

Fig.10. Housing Design Concept

The main component of the housing is the wooden frame that holds the television, the webcam, two-way mirror, and speakers. This frame holds the television and the webcam in place while having an insert for the two-way mirror, keeping it stationary as well. The basic outline of the "Smart Mirror" starts with the reflective glass facing the user. Behind the glass is the television that displays all the applications of the smart mirror. When the TV is off, just the mirror can be observed, but when any of the apps are activated, they can be seen. This allows for use of the mirror as a mirror at the same time the user is interacting with the applications. Connected to the television is the central PC which has connections to the webcam, speakers, leap motion and MCU. The central PC is stored separately from the housing. In addition to this, the webcam is placed on the top of the mirror frame inside the encasing. There are also two speaker vents in the top corners of the frame border for sound to travel to the user.

VII. Conclusion

This documentation was about the smart mirror project. This project stemmed from the need for better time management and productively along with the inspiration of new, developing technologies now available. The smart mirror idea was created to give instant access to information in a convenient and time-saving environment, the bathroom. All other aspects of the mirror's design developed from these ideas and inspirations. The smart mirror comprised of multiple hardware and software components coming together to make a working system. Each component added another element to better the user experience, including but not limited to, the automatic sleep and wake system, the touch-free interaction, the temperature system, and the easy voice control. All of these in the end the smart mirror could not only be a solid prototype but an example of what the smart mirror idea can become in the future.

Acknowledgement

The authors wish to acknowledge the assistance and support of Dr. Samuel Richie for his direction and encouragement to design the project to the best of our abilities. Also, we sincerely thank the project sponsor, Central Florida Impatient Medicine, for without them this project wouldn't have been possible.

The engineers

Georgiy Brussenskiy is a senior at the University of Central Florida and will be receiving his Bachelors of Science in Electrical Engineering in May 2014. After graduation, George plans to purse graduate studies in Electrical Engineering with an emphasis on microelectronics.

Christopher Chiarella is a 23-year old graduating Computer Engineering student. Chris has accepted a position with Texas Instruments in Houston, TX, as a software engineer with a focus on embedded software development.

Vishal Nagda is a 23-year old graduating May 2014 with a Bachelor's Degree in Electrical Engineering. Vishal aspires to pursue a career in sales engineering.

References

[1] Leap Motion, Inc. (2013) Retrieved September 2013, World Wide Web: https://www.leapmotion.com/product

[2] Texas Instruments, Inc. (2013) LM35 Precision Centigrade Temperature Sensors Datasheet. Retrieved October 2013, World Wide Web: http://www.ti.com/lit/ds/symlink/lm35.pdf

[3] Innovative Sensor Technology. (2013) Retrieved October 2013, World Wide Web: http://www.ist-usadivision.com/sensors/humidity/

[4] Microsoft Corporation. (2014) Retrieved September 2013, World Wide Web: http://msdn.microsoft.com/en-us/library/ms752059(v=vs.110).aspx