SMART MIRROR

Matthew Caserto, Greg Eugene, Menashe Mordachai, James Timothy

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

***Abstract* — People have always looked forward to modernization. People have always looked for ways to make life simpler. The Smart Mirror is a technology that brings about a new generation of modernization. The device allows for its users to multitask more efficiently. While getting ready to tackle the day, users can check emails, see the weather pattern for the entire day and plan for rainy weather. The technology saves the users time which can be used for more pertinent tasks. What sets this device apart from its competition is ease of accessibility. Interactions with the Smart Mirror can be done through the IR touch frame or Bluetooth connection.**

I. INTRODUCTION

Our smart mirror is designed from the ground up to provide a simple yet intuitive device to optimize our morning routine and provide information with at a glance. Four main components come together to form the completed project. First is the Raspberry Pi, which runs our user-interface program and displays it to the LCD monitor. Second is the custom-made Infra-red LED frame which senses touch input on the mirror display. Third is the Atmega32u4 microcontroller which brings it all together. This device processes the input from the various sensors and feeds that information to the Raspberry Pi so that it can be viewed on the mirror. Lastly is the two-way acrylic mirror material which will act as the mirror once a LCD display is placed behind it. When text is displayed on the LCD, it shows through the mirror and appears as if text is on the mirror itself. This effect is what is used to create smart mirrors. In conjunction with the smart mirror app itself, an android app was also designed to send data to the mirror over the Raspberry Pi's Bluetooth module. Various data such as weather, contacts, and more will be sent to the mirror for viewing while the mirror will also send data back to the phone. Our apps such as the text messaging app and music control send commands to be executed using the android device.

II. SMART MIRROR UNIQUENESS

For ease of access, the Smart Mirror will be able to function using gestures which will be processed by the sensors and translated into specific actions that was required by the user. The other method of access to the mirror is using an IR-frame, which will provide a touchscreen-ability to the Smart Mirror. Bluetooth will also be added to this Smart Mirror, so that the user can use their phones to access the mirror. Mainly accessing the data specifically for that phone

III. BASIC USE CASE

A user has woken up and is getting ready to tackle the day, the user knows that the email must be checked for any information passed during late hours or early mornings. The user also knows that the weather for today must be checked to make sure that it will not rain during crucial hours when the user is outdoors. Usually the user would have to do these three activities by checking the computer for the email and weather or checking the news for the weather. This consumes quite a lot of valuable time which limits the user overall time during the day for activities. The use of a Smart Mirror allows the user to check weather and emails while brushing their teeth or combing their hair. The Smart Mirror promotes efficiency helps users to get a few extra minutes during the day so that they can spend that time more wisely.

IV. PROJECT GOALS

Our goal for this project is to build on what other students have accomplished with their smart mirrors and improve on their designs. Looking at previous smart mirrors, there is a trend with them incorporating certain features but at the expense of others which doesn't provide a good overall experience. We want to make the best Smart Mirror that has ever been shown at the University of Central Florida by incorporating a strong set of features that work well and look very polished. Features like a custom user-interface with a variety of apps and even animations throughout the Smart Mirror OS. We're hoping that by doing this it won't look like a senior design project but a fully featured product. Our mirror has a full 24" display that covers the entirety of the mirror face itself unlike some mirrors that provide a tiny screen in once section of the mirror. Having a large screen is important to us because that makes the information easier to see while also providing a fuller experience. We hope that having all the features described will help make our Smart Mirror stand out from the rest and provide an experience like no other mirror has as of yet.

V. PROJECT REQUIREMENTS

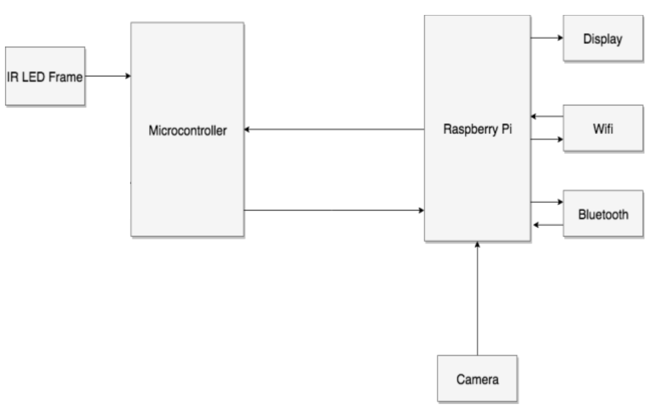
The project team has developed and documented the requirements for this project that should be required. In terms of basic features, it should have a IR touch grid, motion sensor, facial recognition, current weather data and Bluetooth connectivity. Most of the individual features have their own independent features. The IR touch grid should have less than 5 percent variation, the facial recognition should be accurate to about 90 percent, the temperature display should be relatively accurate for the area.

VI. ARCHITECTURE DESIGN

The design for the major portion of this project is simply a computer screen placed behind a two-way acrylic mirror that makes it seem as the display is floating on a glassy surface. The construction of the IR frame on the other hand required some testing and thought to be put together. An equal number set of IR LEDs and photodiodes are placed across from each other mounted in a wooden frame. The IR LEDs and the photodiodes are placed inside the deep holes on the sides of the frame to limit the conic spread of the IR light thus mimicking an IR laser however much cheaper and the design will not require a heatsink to keep it cool. The straight narrow beams emitted by the IR LEDs are pointed towards a single photodiode. The photodiodes are connected to multiplexers and are coded in a way that if IR light does not reach it, it will sense that is where the user has placed his/her finger and will then access certain applications of the Smart Mirror depending on the location of the finger. The camera for facial recognition is mounted inside the wooden frame.

VII. HARDWARE SYSTEM DESIGN

There are a few main components used in the Smart Mirror to make it what it currently is. They are essential to the design and overall features of the mirror itself. The figure below is the hardware flowchart which shows how each of the components are connected.



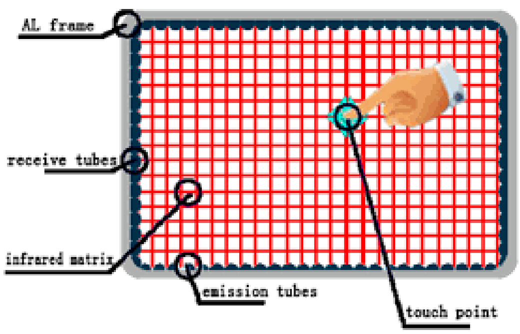
A. Microcontroller

The first step in our hardware design was choosing a suitable microcontroller to use for our Mirror. The number of microcontrollers available is astounding so the choice itself is a very difficult one to make. Various manufactures were considered from ARM, and Microchip, to Texas Instruments. The microcontrollers were first narrowed down to which we as a group were most familiar with starting with the TI MSP430. This was a device we all had experience with from our Embedded systems class that we had all taken before. While this was a great microcontroller, we weren't sold on it just yet. Our second consideration was the ATmega series of microcontrollers which can usually be found in the popular Arduino development boards. The advantage of using these is that we can then use the huge number of libraries that come with using an Arduino. Our third choice was an ARM cortex M-series microcontroller. Our only reason for considering this was that ARM chips have been becoming very popular as of late and using one would be a great learning experience. After much consideration we opted to go with an ATmega32u4 which is a chip seen in boards such as the Arduino Leonardo. We chose this because of our familiarity with the Arduino IDE and various boards of that brand.

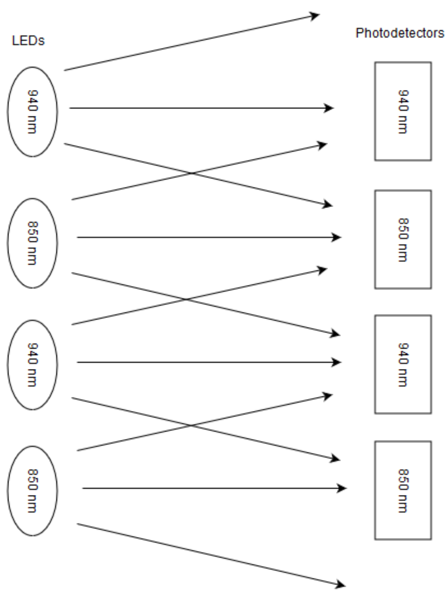
Specification	**Value**
CPU Architecture	8-Bit RISC
Operating Voltage	1.8-5.5 Volts
Clock Speed	8-20 MHz
Flash	32Kbytes
EEPROM	1Kbytes
RAM	2.5Kbytes
Communication	UART,SPI, and I^2C
Inputs/Outputs	26

B. IR Frame

Our project implements the use of an IR Frame which uses IR LEDs and photodiodes to mimic that of a resistive touch screen. Initially we had wanted to use IR laser diodes for our IR touch frame due to accuracy and lack of noise. However, while cross contamination and noise concerns may favor the LD, the LED offers a wider variety of packaging options which make it favorable. LED strips can be purchased allowing for simpler assembly. Additionally, due to the LED emitting light in a nearly omnidirectional manner, alignment becomes less of a concern. However, this results in another issue taking its place. Specifically, the highly diffracted light being emitted will result in excess noise on the adjacent detectors, which ideally would only be receiving light from the corresponding LED opposite of it. This is both the most significant advantage and disadvantage of using LEDs as our light source. The figure below shows how the IR LEDs and photodiodes form the necessary optical grid which functions as a touch screen.



The photodiodes take in the IR light and in such a way functions that if the constant stream of IR light stops hitting the photodiodes it would click on that section of the screen. The picture below shows how the photodiodes are receiving the IR light from the IR LEDs. Each photodiode was connected to a connector and a multiplexer input and used poling to determine which photodiode has not taken any more IR light input.



C. Gesture Sensor

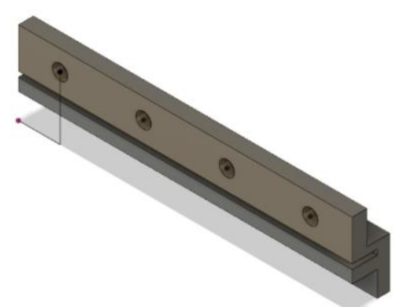
The gesture sensor will input user hand motions such as up, down, left or right to open or close the Smart Mirror application. This is an added component which serves to make the user interface more innovative.

D. Power supply

For powering up the entire Smart Mirror we have opted to use an old computer charger and a DC jack splitter which puts the necessary power to the monitor, and the IR frame. The third splitter was required for powering up the Raspberry Pi which in turn powers up the PCB through the USB port. For powering the Raspberry Pi, a step-down switching regulator was used to bring the voltage to the required 5 volts.

E. Housing

One of the uniqueness of this device is the use of an IR touch frame as a makeshift touch screen device. The use of this IR touch frame meant that the overall housing of the Smart Mirror had to be built around it to make it optimal. Due to this design choice, the frame of the mirror will have added complexity compared to other smart mirrors. This results in extra care being necessary when designing the frame of the smart mirror, because this mirror can have potential misalignment issues in addition to an increase in required internal space to route the extra wires.



Cutouts as shown above are placed between the surface of the glass and the outer frame of the mirror with the cutout being perpendicular to both so that the light from the LEDs flow across the surface of the glass toward the corresponding photodiode. This methodology for installing the LEDs and photodiodes serves two purposes. The first purpose is to make alignment of the LEDs and photodiodes as simple and automated as possible. By using a laser cutter to create the outer dimensions as well as the spacing and diameter of the circular cutouts we will prevent potential issues with alignment. The second purpose is to provide adequate space within the frame of the mirror to route the wires from the LEDs, photodiodes and other electronic components.

IX. SOFTWARE SYSTEM DESIGN

The Smart Mirror is a very software heavy project an depends on that software being good to function as intended.. There are three components of software heavy sections in this device. These components are the Bluetooth, the user interface and the facial recognition. The entirety of the Smart Mirror application is coded in the Java programming language to make testing easier and it was also what we were most familiar with. Java can be compiled on our coding machine and then executed on the Raspberry Pi with no issue and that made the software testing process much easier.

In the beginning the goal was to use the Java FX graphics library for building the user-interface but that did not work on the Raspberry Pi and so we had to resort to using Java's older graphics library, Swing. Swing made the design a little bit challenging just because it's fairly dated and doesn't have as much of the functionality as Java FX. Simple things such as animations for moving visual devices don't even exist in Swing so we had to create our own custom animation classes that could create the look and feel of a professional app.

The last piece of software of note was the android app which is also coded in Java. The android app essentially connects to the Mirror and intercepts commands and responds to them so it's very simple. I used the android developer website's example's for Bluetooth RFComm to get the connection to the pi working and after many hours of trouble shooting, we had a connection. Once a Bluetooth connection is made, the app goes into a mode where it waits for commands from the Pi. The response to these commands did need a lot of research and coding to complete though. I had to figure out how to retrieve calendar events from the native calendar, get contacts and send a text message, and also retrieving the weather from an API using OpenWeatherMap. Learning all of these things took some time but eventually it all worked out.

A. Bluetooth

Bluetooth will be implemented using the Bluetooth module built onto the Raspberry Pi 3 that we're using for our Smart Mirror. Getting the Bluetooth working on the Pi was a very difficult task that included installing many libraries on the Pi and also compiling an Open Source Bluetooth stack for Java called Blue Cove. Blue Cove allows us to communicate with the bus that the Bluetooth module is connected to and send commands to it for sending and receiving data. Blue Cove is a very old library and so it was not built for the ARM processor on the Pi nor did it support Bluetooth low-energy. Getting it running on the Pi involved compiling the source files on the Raspberry Pi itself which is a lengthy process. Once that was done though these libraries could be included with the execution file and it worked from there. Many problems were encountered setting this up and if I could do it again, I would opt for a Bluetooth module off the Pi instead of using the one built into it. A lot of these issues could have been avoided but that also would have increased the cost of the project.

B. User Interface

Since we will be using a semi-reflective material for our smart mirror. Our mirror itself will only physical work as a mirror as long as there isn't a light behind it. It is extremely important to make that background for the system is black or dark. We get the best contrast when the information display is white against a black background.

When we first started the project, we decided that the interface would allow for limited user interaction with the mirror itself. This is because we want the mirror interactive and still function as one and no one wants an excess of smudges on their mirror. The finalized decision was to use the IR touch frame for accessing the required sections of the user interface. We wanted to ensure that the device could still be used as a mirror without having the view impeded by excessive information. We wanted to make use of the outer sections of the mirror so that content can be viewed while at the same time leaving enough room for people to see themselves. With all of this in mind, the following components are what have been included in the UI.

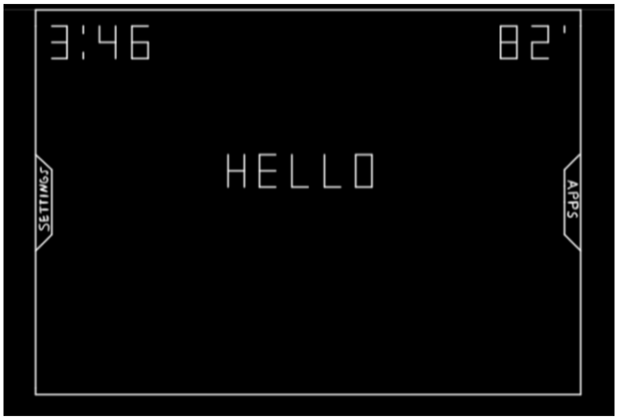
- **Weather**
  We will be showing them the forecast based on their location

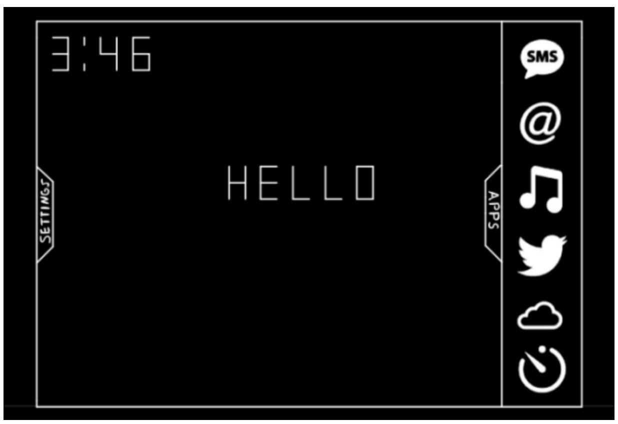
- **Clock/Calendar**
  We will be integrating the internal clock and a calendar system

- **Plans on the calendar**
  Uses and important events placed in the smart phone to display onto the device

- **Music controller**
  Uses the user chosen music app (in our case iTunes library) and can play/stop songs and play previous or next songs in the library

We have opted to use Java to write the code for the UI because it is simpler and aesthetically we also wanted a simplistic UI that could be used by regular people. Also the device is still technically a mirror and the simplistic UI allows for less clutter which in turn still allows for the user reflection to show in a large portion of the mirror itself. The figures below show the initial UI design, which does not differ too greatly from the finalized design.



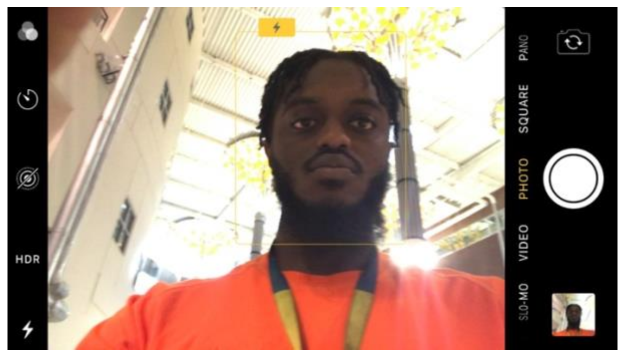


C. Biometric Verification

In researching technologies for the smart mirror, we looked at all types of products that implemented some form of facial recognition. From that researched what components would be necessary to build a functioning biometric system.

Facial recognition software applications are biometric based algorithms designed to identify the unique features of an individual by comparing segmented facial patterns to verify the identity of the person. By using deep learning algorithms, the software can analyze live capture features or use stored digital imprints to verify the identity of an individual by measuring their overall facial structure. For our particular system it was necessary to utilize a biometric technique of non-contact nature to limit physical contact with product to prevent residue and preserve visibility. Being one of the least intrusive and fastest forms of biometric technology, facial recognition as form of authentication has become one of the most viable options for security purposes. The feature measurements that are gathered by the algorithm are given a numerical code and stored in a database and then compared to other collected datasets of faces whenever a person stands in front of the imaging device. In today's market there are many consumer electronic companies that offer some form of facial recognition for both authentication and identification.

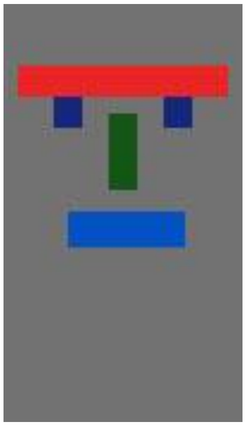
In 2009 Apple released its iPhoto application with both facial detection and recognition using the facial detection algorithm to determine if an objects features were at first consistent with those of face, and then facial recognition to distinguish people based on their unique features.



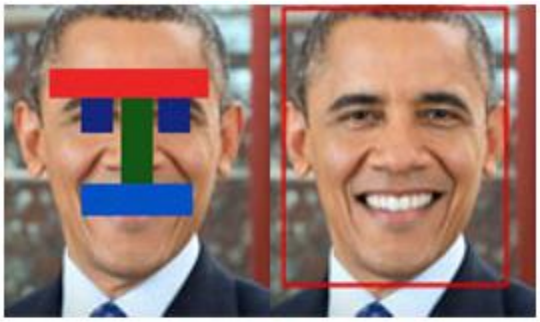
In 2017 the iPhone X was released with a facial recognition system for unlocking devices using a 3D mapped facial imprint stored in a local part of the processor.

D. Facial Recognition

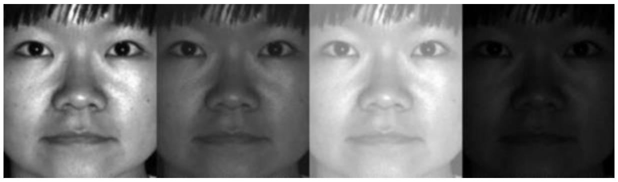
The purpose of the facial recognition software was to instantaneously identify individuals from either the use of video or digital image capture device for accessing our application. Using specific algorithms to analyze particular features on an individual's face we were able to accomplish unique feature identification using the relative position of points on a face or size and shape of an individual's features. Unlike fingerprinting and voice recognition, facial recognition software yields nearly instant results because subject consent is not required. Our facial recognition software was intended to be primarily used as a protective measure and for verifying personnel activities computer access. An advantage of the live capture is that the appropriate action can be taken whenever the biometric data is gathered. Like other biometrics solutions, face recognition technology processes and identifies the distinct characteristics of an individual that can be used for identification or authentication. Often times using some form of digital or video/live capture device. Since face recognition algorithms have the ability to identify faces in images/videos the software can compute individual features, and then validate them based on stored datasets within in a user database. There are many live capture methods used in consumer facing devices. For example, there are some ATM's that use face recognition to verify that the individual making the transaction at the machine is the individual whose information matches the card being used at the machine. The individual has to look in the direction of the live capture device; images are captured for verification and transaction continues otherwise, the transaction is not cancelled. Face recognition software requires a live capture device to use identification instead of passwords to secure user devices such phones, computers or and laptops. For example, whenever individuals sit in front of their computer, the camera uses the image which the software then processes the face to identify the person in front of the device. If the individual in front of the computer is a registered user, then he will be automatically logged on to the computer. The main difference between detection and recognition is that with former the only thing we need to determine is whether or not there is an actual face in the image or live feed in front of the camera. With face recognition the goal is to determine the face and verify whether it grants access to the user. In our implementation we used pattern matching to determine if the image had features consistent with a particular face…this is detection.



When the system is trained it would be able to should be able to recognize the person in the photo as the Barack Obama.



We used a FisherFaces based algorithm This particular algorithm is an improved version Eigenfaces. Our algorithm inspects the entire training dataset at one time and finds principal components from all of the training images combined. By doing this, it doesn't place emphasis on any feature that might distinguish one image from another. Rather, it focuses on the features that are representative of all the faces in the training dataset in its entirety. The algorithm also uses light as a part of its calculation, it uses this difference as an aspect of face recognition. These variances that are extracted represents just a singles individual's features.



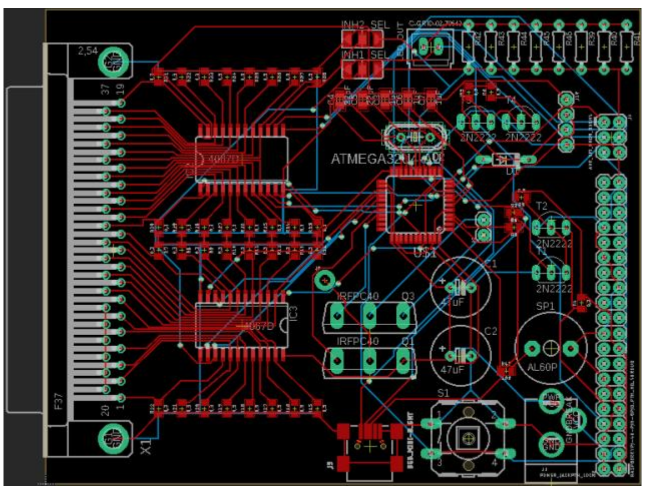
By tuning the algorithm to extract useful features from the face of each individual separately rather than extracting features from all them combined, regardless of the changes in illumination in any image it will not affect the other individuals feature extraction process. FisherFaces face recognizer algorithm extracts principal components that differentiate one person from another. In that sense, individual features do not become more dominant over the others.

Unresolved Raspberry Pi Multithreading issues

In order to incorporate both our C++ and Java programs we needed to make use of the Java Native Interface(JNI). Java provides the native keyword that's used to indicate that the method implementation will be provided by a native code. When making a native executable program, we can choose to use static or shared libs. Using shared libraries for the JNI is highly important as we can't mix bytecode and natively compiled code into the same binary file. Therefore, our shared lib keeps the native code separately within its *.so/.dll/.dylib* file (depending on which Operating System we're using) instead of being part of our classes. Because we are linking the native code with a dynamic shared library which then creates a unique set of threads that utilizes rendering the raspberry pi has no native way of handling this. It seems that OS allows rendering only in main thread, which as of the latest stable release is an issue that has not yet been resolved.

X. FINAL BOARD DESIGN

The Smart Mirror integrated a simple two-layer PCB which were tightly packed to integrate all the necessary components required. This was one of the most important components in the Smart Mirror because it essentially acts as the brain of the whole device after everything was put together. The final product took a few trials and errors to reach its current optimal look and design. The figure below shows how the Eagle rendition of the finished board.



The Client

The Smart Mirror was created with the intentions of bettering the lives of everyday people. The mirror allows for the users to multitask and save ample times every day which can be allocated to doing better things. The device is also makes for a great addition for the tech savvy who enjoys using technological devices for everything. Overall this device can prove beneficial to most users since the user interface is not too difficult to use.

The Smart Mirror Engineers

Matthew Caserto is a Electrical Engineering student graduating in Fall of 2018. He has worked at Nelson Engineering Co. for two semesters and enjoys Embedded Systems development from software to hardware.

Greg Eugene is a Computer Engineering student graduating in Fall of 2018. He has worked for the Siemens Digital Grid. He has an interest in Embedded Systems and Hardware Description. He is currently deciding between multiple offers of employment and will be attending Graduate School in the near future.

Menashe Mordachai is an Electrical Engineering and Photonics Science & Engineering student graduating in the Fall of 2018. He has an interest in RF and microwave circuits. He currently works at Lockheed Martin through the College Work and Experience Program and has been offered a full-time position here upon graduation.

James Timothy is an Electrical Engineering student graduating in the Fall of 2018. James aspires to pursue a career in electrical engineering.

Acknowledgements

We acknowledge the assistance, support, and guidance of Dr. Samuel Richie and Dr. Lei Wei. They have aided greatly in giving out ideas and answering all questions for both the actual engineering as well as assistance in the management of the Smart Mirror team.