EE 4IO6 - Magic Mirror

Supervisors: Dr. Sharam Shirani
Dr. Xun Li

Group L16 - April 13, 2019			
Name	Ankur Bargotra	Moshiur Howlader	Chen Song (Allen) Zhang
Student #	001306226	001316948	001350117
E-Mail	bargota@mcmaster.ca	howlam@mcmaster.ca	zhangcs@mcmaster.ca

Introduction

The motivation of our project was to design and implement a user-friendly smart mirror that is capable of notifying the user with their day-to-day informative interest. The product is targeted towards clients living in urbanized areas to provide ease of usage and quick response that is applicable to employed individuals with “busy” working hours. The modern world is full of information and this products job is to aggregate the information that matters to you with a device we all have in our homes and use every day, a mirror.

Project Goals & Planned Approach

The goal of this project was to implement a user-friendly smart mirror. The planned approach were the following. Research what technologies were available, then based on that, design the modules or components to be implemented into the overall product. Initially we had designed the system to require the below specifications. However as the project implementation began, certain technical, time, and financial limitation altered the total requirements that were actually realized and implemented.

Project Specification Details:

Requirement	**Milestone Stage**	**Status**
The system should be able to turn on with a push of a button.	Bronze	Complete
The LCD monitor and its related parts should be integrated with the Mirror, and the Wooden Frame.	Bronze	Complete
The wiring and power to the system should be put into a nice, small compact package.	Bronze	Complete
Raspberry Pi should be able to display time and date.	Bronze	Complete
Raspberry Pi should be able to display weather based on current location.	Bronze	Complete
Raspberry Pi should be able to connect manually to a smartphone using bluetooth.	Bronze	Complete
Implement Python Embedded Kivy Software/App to enable a basic operation of the smart mirror.	Bronze	Complete
Enable automatic bluetooth connection to smartphones nearby. (Polling)	Silver	Complete
Enable display of recent emails received from a binded email address.	Silver	Complete

Enable display of new text message notification as well as displaying the content.	Silver	Complete
Implement application to add “to-do-list” items on home screen of the mirror.	Silver	Unattempted
Implement voice recognition system for all silver-level milestone steps.	Silver	Unattempted
Implement suggestion feature for clothing preference depending on weather - temperature, humidity, brightness, etc.	Gold	Unattempted
Implement positive commentary system both through voice and text messages for user looking in the mirror.	Gold	Partial
Implement augmented reality system by image recognition to virtually put on clothes for user decision.	Gold	Feature Scrapped
Become your own shirt picker! The mirror will be capable of displaying apparel through augmented reality without the user physically changing on their clothes.	Platinum	Complete

Although some of the Silver and Gold features were not implemented, the actual implementation of most of these features is not difficult. More time to implement and integrate the product should allow a robust Magic Mirror. The To-Do List can be implemented using basic HTML, CSS, JavaScript, DOM, and web Cookie to store user session data. The voice recognition feature would require some time and effort to learn. The Amazon’s AWS API would be the appropriate technology to implement the feature. Even if some overhead in learning the technology, there is detailed documentation on how to use Amazon’s Automatic Speech Recognition - AWS API. As well, integrating it into a Web Application would not be very difficult. The following link: https://docs.aws.amazon.com/transcribe/latest/dg/what-is-transcribe.html, shows how to take an audio file or even a live stream audio/voice stream, and convert it in JSON formatted data to allow processing of the speech data. This JSON data can be used to implement various voice related features into the Magic Mirror’s Web UI. Also, Amazon’s documentation states a form of natural language processing technology called *diarization* or *speaker identification*, which can allow Amazon Alexa and our app to, theoretically, recognize the user’s unique voice and identify them. There is many powerful, cool features that can be implemented with this technology.

Approach Taken

Since the final product have various independent modules or components to it, our group assigned different parts of the project to each other and individually worked on them. Before starting the work, individual members in the group agreed on the modules they wish to work on and then distributed the work accordingly. Ankur worked on the hardware implementation of building the mirror, as well as the Web UI. Allen worked on the Bluetooth/Phone Sync interface. Moshiur worked on the Wardrobe App. Our group would meet up once a month to keep tab on progress of each member of the group and discuss any critical issues. At end of the project, we met up to integrate to the best of our capability the final product.

Below are the specific details regarding the implementation:

A) Building of the Acrylic Mirror System

We purchased a slim monitor so we would not have to disassemble the monitor. Using that monitor, dry-fit everything interfaced as expected. Appropriate measurements done to fit the wooden frame perfectly onto the monitor. Appropriate Acrylic Mirror Sheet purchased. Acrylic Mirror sheet is a two way mirror sheet. It reflects back light like a regular mirror but also lets out mirror from the back. Think of an interrogation room glass. The convict cannot see you but you can see them but the glass also acts like a mirror.

B) Web UI Implementation

The web UI was implemented on http://ankurbargotra.com. The web app allowed user to create an account which allowed personalization of the app. The app allowed geographic specific weather forecasting, time, calendar, news feed from various sources, as well as customizing the layout of the UI by adjusting the widgets in the interface. Technologies used were HTML, CSS, JavaScript, and AngularJS as well as various API for getting weather and news data.

C) Bluetooth Phone Sync

The Python code to sync the smartphone with the database was implemented. The Python code on the Raspberry Pi would connect to the smartphone and received an XML file containing all the SMS and Email data. That data in the XML file would be parsed and stored in the database. From the database, the web UI can read that data and dynamically update.

D) Wardrobe App

The Python code for the Wardrobe App was implemented. The Kivy framework was used for the app. An open source Python tool called google_images_download from Github was used to implement the Google Image download feature. PIL Image library was used for the image processing. For sending emails, the standard Python email library was used.

Figure 1: Searching of T-Shirt on the Web

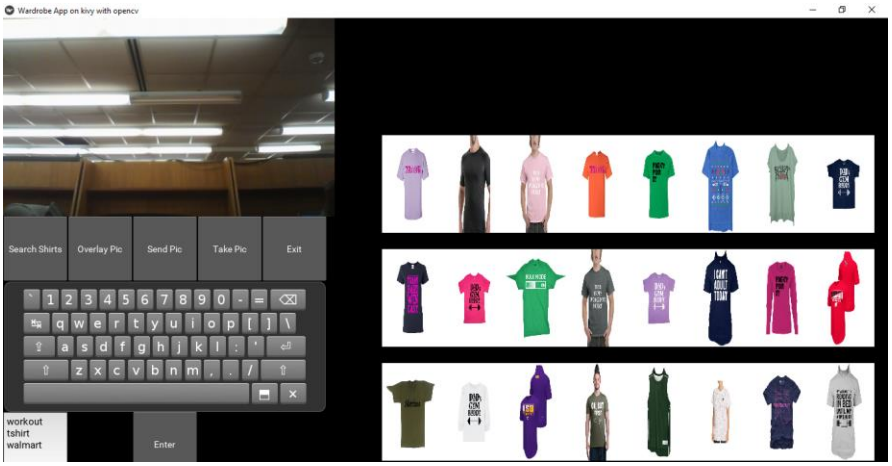
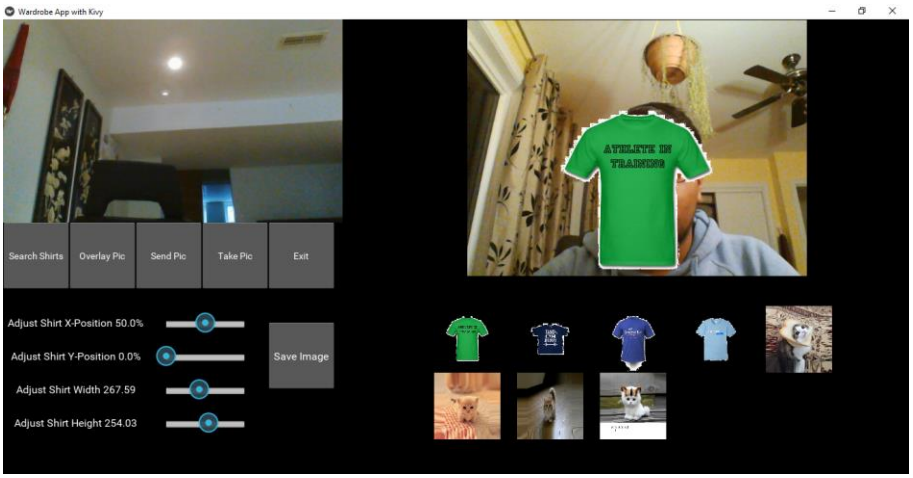


Figure 2: Overlaying of T-Shirt on Selfie Image (note the app in the final mirror was vertical)



E) Final Integration of the Magic Mirror

The Magic Mirror was required to be in a demo-able state. In order to do so, the Magic Mirror needed to be standing upright. A wooden board with 4 wooden plank was used to stabilize the Mirror upright. Various drills to hold the wooden plank was placed to hold the Mirror. Finally two screws from the bottom of the wooden board was drilled to stabilize the mirror. Note if this product were to be an actual marketable product, the manufacturing of the system would need to be light enough to be able to be placed on the wall. The Raspberry Pi, the Pi Cam, and all the cords were nicely tucked in the back of the mirror. The Pi Cam was placed on an angle at the top of the mirror.

Figure 3: The Final Magic Mirror Hardware

Critical Problems Solved

1) Dead LCD

After our initial hardware dry-fit assembly of all components in the frame the monitor we had stopped working. It is assumed this happened because, one, the LCD monitor we were using was from 2010 and, two, because in order to fit in the back of the frame we had to disassemble the monitor frame which may have cause internal damage.

To resolve this issue we purchased a slim LCD monitor of the same screen dimensions. We purchased a slim monitor so we would not have to disassemble the monitor and upon dry-fit everything interfaced as expected.

2) Securing LCD + Mirror

Once we put the acrylic mirror and monitor at the back of the frame we noticed an issue where they had degrees of freedom to move in the x, y and z directions. To stop movement in the x and y direction we added pieces of wood along the side of the back of the frame as well as rounded brackets to keep the monitor and frame securely in place. For the z-axis, after completion of the project when all hardware is residing at the back of the frame, we will add long horizontal pieces of wood and screw them in to the back frame to prevent movement along that axis.

3) Porting Wardrobe App onto Raspberry Pi

Once the final wardrobe app was finished with all the functionalities, it had to be ported to the Raspberry Pi for the monitor size of 1920x1080. Because the monitor was to be viewed in portrait, all the dimensions were required to be fined tuned for the app. As well some optimization of the usage of the Camera module was required. Since the Camera module is draining in terms of its use of resource, it was necessary that only for functionalities such as taking a photo or load page to utilize the video webcam. All other functionalities were required to be stripped off of this video functionality on the page.

4) Incompatibility of Webcam on Raspberry Pi

Originally we had purchased a Logitech Webcam but after trial and error, it was found that there were no clean interface to code or use the Logitech Webcam. Hence an official Raspberry Pi camera was purchased.

5) Challenges involved with interfacing and translating Phone Data

An appropriate scheme for transporting the SMS and Email data from the phone to the Raspberry Pi was required. XML was the file format to send the phone data from the phone to the database by Bluetooth. From there Raspberry Pi can access the database.

Conclusion

The main modules or component were implemented to a satisfactory degree. Unfortunately, integration to make it into a seamlessly fluid product was not achieved. The Web UI, the Phone Sync, and the Wardrobe app worked. The Web UI have the core functionalities of showing weather, time, news from various news channels, customized compliment generator, and a nice sleek interface. The software technologies supported touch screen although the hardware did not support touch inputs. This is due to the cost of the touch-enabled Acrylic Mirror Sheet. The phone sync with Bluetooth properly and accurately stored SMS, and Email data to the database. This database was easily accessible by Web, however due to unforeseen time constraints, interfacing of the database to the web UI was not possible. Finally, the Wardrobe App was implemented with Python Kivy and the major functionalities of Searching for Shirts on the web, taking picture, overlaying and previewing the shirt, and sharing the preview shirt overlay image by email was successfully implemented. The performance of the app was great on a computer but significantly slow on the raspberry pi. More time and research as well as development time to full flesh out the product is needed to bring it to an industrial/market ready product. The main functionalities of the product was implemented correctly however.

Future Plans/Possibilities

Once a reasonable manufacturing cost is achieved, and enough useful technologies such as IoT, face recognition, voice recognition using Amazon AWS Transcribe service, a central hub for information and automation control can be implemented and packed into a convenient easy to use package, this product can definitely be found in everyone’s homes. Especially if a big name software corporation can extend the wardrobe app to take full advantage of all the advancement in computer visions and machine learning, e-commerce platforms can even be implemented and interfaced with Amazon’s clothing store for example.