

Ecosystem for Smart Glass Technologies (ESGT)

ECE4012 Senior Design Project

Section ET2, The Magicians
Project Advisor, Dr. Manos Tentzeris

Kairi Kozuma, Team Leader
Najee Kitchens, Web Master
Boa-Lin Lai
Jonathan Osei-Owusu
Nishant Shah

Submitted

2017 May 3

Executive Summary

The Internet of Things (IoT) device market has exploded in recent years, and is expected to reach 6.7 billion device shipments with \$1.7 trillion in value added to the global economy in 2019 [1]. To take advantage of this trend, companies have begun exploring innovative ways to add smart capabilities to everyday products. Ecosystem for Smart Glass Technologies (ESGT) is a complete hardware and software platform that brings IoT devices to the transparent realm. ESGT aims to enhance traditional glass products by displaying information and incorporating user interaction. Example data to be displayed include weather, traffic conditions, directions, personal agenda, email, and news. ESGT has two main parts: hardware technology to implement transparent displays and a software platform common to all glass devices. Application specific modules will be implemented for each glass product, leveraging the underlying software framework. Users can interact with ESGT products through gesture sensors. The prototype that showcases ESGT is a Smart Mirror. This prototype uses a traditional display with a two-way mirror; furthermore, it uses a microcontroller that connects to the local Wi-Fi connection. Hover sensors are used to receive gesture inputs from users and interact with on screen components. A smartphone application is used for setup and configuration of the Smart Mirror via Firebase, a cloud synchronization platform. The prototype has light, humidity, and temperature sensors; however, these can be replaced with biosensors (heart rate monitor, muscle contraction sensors, etc.) to provide health feedback capabilities. The initial prototype costs \$500.04 to build. Using this ecosystem, applications such as smart windows, eyeglasses, and windshields can be developed.

Table of Contents

Executive Summary	ii
1 Introduction	1
1.1 Objective	1
1.2 Motivation	1
1.3 Background	2
1.3.1 Transparent Display	2
1.3.2 Floating Touch	2
1.3.3 3D Image Processing and Gesture Recognition	3
1.3.4 Microcontroller	4
1.3.5 Wireless Connectivity	4
2 Project Description and Goals	5
3 Technical Specifications & Verification	6
4 Design Approach and Details	7
4.1 Design Approach	7
4.1.1 Hardware	7
4.1.2 Software	8
4.2 Codes and Standards	10
4.3 Constraints, Alternatives, and Tradeoffs	11
4.3.1 GUI Design Language	11
4.3.2 User Interaction Method	11
5 Schedule, Tasks, and Milestones	12
6 Final Project Demonstration	12
7 Marketing and Cost Analysis	13
7.1 Marketing Analysis	13
7.2 Cost Analysis	14
7.2.1 Cost of Components	14
7.2.2 Development Costs	14
7.2.3 Selling Price and Profit	16
8 Conclusion	17
9 Leadership Roles	17

Commented [1]: Update page numbers

Commented [KK2R1]:

9.1	Najee Kitchens	17
9.2	Kairi Kozuma	18
9.3	Boa-Lin Lai	18
9.4	Jonathan Osei-Owusu	19
9.5	Nishant Shah	19
10	References	20
	Appendix A – Task Assignment and Risk Level	24
	Appendix B – Gantt Chart	25
	Appendix C – Smart Mirror GUI	26
	Appendix D – Smart Mirror Prototype	27
	Appendix E – Links to Source Code, Media, and Documentation	28

Ecosystem for Smart Glass Technologies

1. Introduction

Ecosystem for Smart Glass Technologies (ESGT) is a complete hardware and software platform that lays the foundation for rapid development of glass IoT devices. The ESGT development team designed a Smart Mirror that displays useful user information as a prototype application of the ecosystem. The prototype cost is \$500.04.

1.1 Objective

The objective of ESGT is to design and develop a robust hardware and software ecosystem to support all glass IoT devices. The Smart Mirror prototype allows the user to interact using gestures. The gestures will be used to turn on and off the display and interact with on-screen data. In the framework, sensor values are stored locally on the device, while weather, email, and other APIs are called when needed. Information is displayed near the edges of the mirror, so that the user's own reflection remains unobstructed. A smartphone application is used for pairing and configuration.

1.2 Motivation

Although IoT has found its way into thermostats, vacuums, and refrigerators [2], it has not had a huge impact on transparent products. ESGT technology aims on providing users an intuitive and easier way to interact with glass products. All of the modules in the ESGT software platform can be tweaked through a smartphone application according to user preference. ESGT also looks at alternatives such as inkjet printing to print transparent sensor circuitry directly onto glass. This approach allows custom sensors to be directly mounted on the glass without additional wires, keeping glass products themselves transparent. It also allows rapid prototyping and deployment of any glass IoT device at a

reasonable and compact size. The Smart Mirror was chosen as the prototype for ESGT, as there are no commercial Smart Mirrors available to purchase. In addition, the prototype improves upon existing community solutions with new features: multiple user capability, configuration synchronization and backup, modular sensor input and storage, and user interaction. These features enable the ESGT platform to scale to many users and devices. Furthermore, by adding the ability to read live sensor data, the Smart Mirror can be extended to read biosensors and provide health monitoring capabilities.

1.3 **Background**

1.3.1 **Transparent Display**

The electronic display industry has done extensive research on transparent displays. The most common implementation is transparent OLED circuitry, which is constructed from electrically conductive material with low light absorption, called transparent conducting oxides (TCOs). Indium tin oxide (ITO) is the most widely used TCO because of its electrical conductivity, optical transparency, and ease with which the material can be deposited as a thin film [3].

Each pixel in a transparent OLED display is made up of four sub-pixels: red, blue, green and clear. The clear sub-pixel is responsible for transparency; furthermore, the ratio of clear to colored sub-pixels is directly proportional to resolution. A ratio above one creates less space for clear sub-pixels, resulting in more occluded displays but higher-resolution images [4][5].

1.3.2 **Floating Touch**

Floating touch adds a third dimension to the touch screen, allowing users to interact by hovering over the display. This is necessary for ESGT to provide interaction capabilities without directly touching the glass surface, preventing fingerprints and smudges.

Floating touch can be implemented by innovations to traditional projected capacitive touch. A touch

event is detected by measuring changes in capacitance at an electrode. When a conductive object approaches an electrode, it interferes with the electromagnetic field around it and alters its capacitance [6]. Since the electric field extends outside of the screen, the electrode can detect objects without physical contact, provided the capacitance change is above a threshold value [7]. To obtain X and Y coordinates on a touchscreen, electrodes are arranged in a two-dimensional matrix [6].

Floating touch technology requires the same three core components as traditional capacitive touch screens: touchscreen panel, touch controller unit, and an Analog Front End (AFE). Changes in capacitance can be as little as tens of pF, so the sensing circuit must detect changes of that magnitude [8]. The AFE converts the capacitance to a voltage using a differential sensing circuit, which amplifies the signal to voltage levels ranging from 20mV to 50mV[9]. The voltage is then converted to digital data with an Analog to Digital Converter (ADC). This data is then sent to the processor, typically via I²C or SPI interface standards [9].

1.3.3 3D Image Processing and Gesture Recognition

An alternative to capacitive floating touch is 3D image processing to capture gestures. 3D image processing can be setup with a camera or 3D depth sensors. The data captured is passed to a 3D image processing software embedded in the computer [10]. For instance, the Microsoft Kinect is a commercially available gesture recognition device that has three cameras, one for the RGB (red, green, blue) colors and two for 3D depth sensors. The RGB camera helps to distinguish the user from the background, and the 3D depth sensors track the user's body [11]. This type of technology can be used to track fingers for the Smart Mirror, allowing users to control the device with simple gestures.

The main operations of gesture detection are image capture, digitization, segmentation, model fitting, motion prediction, and qualitative or quantitative conclusion. During image capture, a 3D object is projected to a 2D intensity image through a pinhole camera. Then, digitization converts the analog

signal to a digital one. The digital image is segmented into parts, simplifying its representation for processing [12]. Different model fitting and motion prediction algorithms - different motion analysis (DMA) method, the Block-Matching Algorithm (BMA), and boundary pixel decimation (BPD) - are used to extract meaningful information from the segmented image [13]. Finally, the gesture is determined to be valid or invalid by qualitative or quantitative measures. The main software program on the Smart Mirror will use the gesture to respond to user input.

1.3.4 **Microcontroller**

Microcontroller choice is important for embedded device, as it determines the performance and affects the power consumption. The Smart Mirror will use a Raspberry Pi 3 Model B as its prototype microcontroller. This is a single board computer with a 1.2Ghz 64-bit quad-core ARMv8 CPU as its main processor. The board offers built in capabilities for Wi-Fi and Bluetooth, making it an attractive option. The Pi 3 requires 2.5A at 5V, making the total power consumption 12.5W [14]. It is also cheap, available for \$35 [15]. Another option is the Raspberry Pi Pi Model A+ board, designed for low consumption by operating between 0.5W to 1W [15]. However, there is no built-in Wi-Fi capability and the processor only runs at 700Mhz on a ARMv6 CPU. This may not be enough to perform the required gesture or touch recognition processing, while driving a large display.

1.3.5 **Wireless Connectivity**

The Smart Mirror will use Wi-Fi to connect to the Internet as an IoT device. Wi-Fi refers to any type of IEEE 802.11 Wireless Local Area Network (WLAN), extending the reach of wired Local Area Networks (LANs) [16]. Since Wi-Fi ranges are typically 30 to 50 meters, the Smart Mirror can be placed far from the router and still connect to the Internet. In addition, the common 802.11n standard offers a theoretical “maximum data rate of about 540 Mbps”, which is more than adequate for sending

simple API calls and data.

A smartphone application will be used to set up the mirror. This includes Wi-Fi authentication setup and other software configurations. The application will use Bluetooth to connect to the mirror.

Bluetooth networks are called piconets, which are established dynamically and automatically as Bluetooth devices enter and leave radio proximity [17]. The smartphone will act as the master device and pair with the slave Smart Mirror device. Although Bluetooth can only achieve speeds of around 2.5Mbps and has a short range of 60m, it is sufficient for setup configuration [17].

2. Project Description and Goals

The goal of ESGT is to design a hardware and software platform for electronic glass devices. The Smart Mirror is a prototype that acts as a proof-of-concept device. A Raspberry Pi microcontroller drives the display and connects to the Internet using Wi-Fi. The software framework includes APIs common to all glass devices, with a database implementation for storing sensor values. The application specific modules of ESGT have graphical user interfaces (GUI) and interaction methods necessary for the Smart Mirror. An accompanying smartphone application is used to set up and configure software on the Smart Mirror. The target user is the general consumer market, as there are currently no Smart Mirror devices available commercially. The following lists the project goals achieved, as well as additional features desired in future work.

ESGT General Features Achieved:

- Framework of software APIs common to all glass devices
- Flexible GUI to support a variety of glass applications
- Multi-user capability, with configuration backup to the cloud
- Display weather, agenda, news, email, and calendar information

- Sensor value integration and storage to local database

Smart Mirror Features Achieved:

- Smartphone application to set up and configure the Smart Mirror
- Appearance of a regular mirror when display is off
- Interaction capability via gesture sensors

ESGT Additional Features Desired:

- Biosensor integration
- Display clear to see in all types of lighting
- Low cost, transparent sensors and displays via inkjet printing
- Additional modules (dashboard, window) with complete GUI

3. Technical Specifications & Verification

Feature	Specification	Measured Value
Mirror Glass Dimensions	400mm x 225mm x 1mm	562mm x 434mm x 3.175mm
Mirror Package Dimension	410mm x 235mm x 50mm	700mm x 540mm x 94mm
Chassis Anchor Holes	2 holes	N/A Anchor Holes Not Used
Power Supply Voltage	90-120V	120V
Display Refresh Rate	60 Hz	60 Hz
Interaction Response Time	< 100ms	< 300ms

4. Design Approach and Details

4.1 Design Approach

4.1.1 Hardware

The Smart Mirror consisted of two Raspberry Pi 3 Model Bs, one Mbed microcontroller, two Hover 2.0 gesture sensors, one phototransistor, one temperature-humidity sensor, one two-way acrylic sheet, and a wooden case and frame. Wood was chosen as the casing and framing material because of its malleability and low cost. Holes were drilled in the casing for both screws and ventilation.

Raspberry Pi 3 was chosen for this Smart Mirror prototype because its built in Bluetooth, Wi-Fi modules and its excellent processing power with modern UNIX like operating system. For analog input reading unit on Smart Mirror, the Mbed (32-bit [ARM Cortex-M](#) microcontrollers) was used. For the prototype, the analog sensors integrated were the phototransistor and the temperature-humidity sensors. While the team successfully included the sensors they proposed to integrate, future iterations of this prototype would feature biosensors (i.e. muscle contraction and heart rate sensors).

The Hover was used as the main user interaction method in Smart Mirror prototype. It allows user using gesture to interact with the platform in 10 cm. distance, without touching the mirror. While the gesture sensors have been integrated successfully and provide reliable feedback, future iterations of this prototype would include more inconspicuous placement of the Hover 2.0s. The issue with hiding the Hover 2.0 is that if placed behind a thick enough material (i.e. 1/8 in. acrylic sheet), no gestures will be tracked or identified. A few alternatives include using a different gesture sensor device, using a two-way acrylic with a thickness less than 1/8 in., or reproducing the Hover 2.0 using transparent circuitry inkjet printed directly onto the display substrate.

4.1.2 Software

Appendix C shows an image of the GUI appearing on the front-end website.

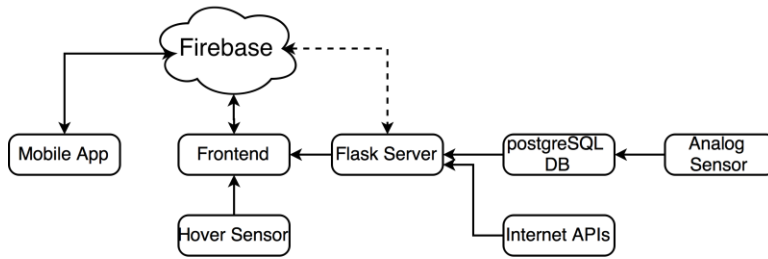


Figure 1. A diagram showing the flow of data for the Smart Mirror prototype.

For the front-end graphical user interface on the mirror screen the framework needed to be as portable as possible so it could work on a range of microcontrollers that could be used in future applications. The framework was build using VueJS [18], a javascript library for building single page applications. VueJS provides a lot of helpful abstractions to help build the UI components that would each display information to the user. Each components consisted of the HTML layout, CSS for styling the component and the Javascript code that could fetch information from the server (eg. weather, email), or get system information (eg. time, date).

The backend uses Flask to set up a REST API for the frontend to access. Flask is a Python microframework, useful for simple websites. Using REST conventions, the frontend can obtain sensor values, weather, news, and other information via HTTP GET requests to various endpoints. For instance, to obtain the light sensor values, the frontend can send a GET request to “<http://127.0.0.1/resource/light>”. The website responds with a JSON string, which can be parsed and displayed by the frontend. This decoupling of the frontend and backend allows the frontend to be replaced by any framework that can access REST APIs, enabling flexibility when leveraging the

software framework for other glass applications.

The Flask server abstracts away the difference between sensor values and other internet APIs, since the GET request format is identical for both cases. For sensor values, Flask retrieves the values stored in a PostgreSQL database. PostgreSQL is the world's leading open source SQL database. Traditionally, SQL statements are used to store and retrieve values in a table within a PostgreSQL database; however, by using the SQLAlchemy Python library, the Flask server can interface with the database using Python scripts. A separate Python script reads in sensor values from the mbed virtual serial port and inserts them into the database. This updater script uses a scheduler to read in individual sensors at custom intervals. For internet APIs, Flask simply acts as a proxy and obtains the API values with parameters on behalf of the frontend.

In order to manage the each unique user and their specific configuration for data sources, the backend uses a Firebase database. Firebase is a NoSQL database service provided by Google that allows data to be stored as key-value pairs. Each user has an object in the Firebase database, that contains their source for news, weather location, email ID and other information. The frontend uses the user configuration to request the relevant information from the backend Flask server. Firebase has a JavaScript library that allows for callbacks whenever data is changed. These callbacks allow the frontend to change as soon as a change is made rather than polling for changes at intervals.

Another software component is the android application that allows users to update their configuration. The application also performs the Google sign-in required in order to display their Gmail and Google Calendar agenda.

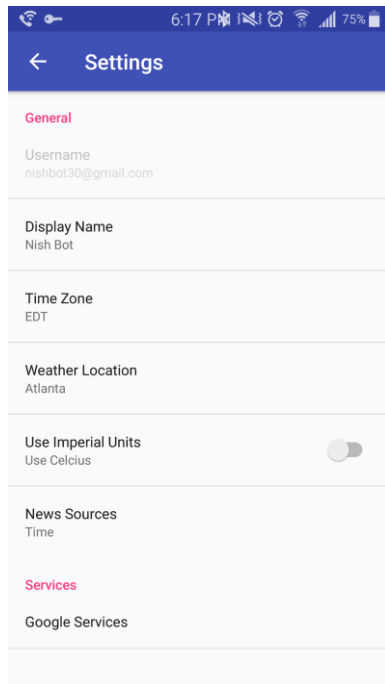


Figure 2. Screenshot of the ESGT Android application, allowing remote configuration changes.

4.2 **Codes and Standards**

Since the mirror will have wireless connectivity to communicate with the Internet as well as the mobile device of the user, the Wi-Fi and Bluetooth modules need to be Federal Communications Commission (FCC) certified for use in the United States. The Wi-Fi also needs to follow the IEEE 802.11 protocol [19].

The microcontroller and the transparent display circuitry will be connected using an HDMI cable, so the ports of both need to be the correct shape and size with 19 pins [20].

The microcontroller has a USB port to connect a USB cable to the gesture sensor. The Raspberry Pi 3

Model B fits the requirements of the design, since it has both an HDMI and USB port [15].

The United States outlet voltage standard is 120V; thus, this limits the amount of power that can be supplied to both the microcontroller and the display [21].

Using two plastic anchors would have a failure weight of 160 lbs, which limits the weight of the product materials [22]. Two different types of wood were used in the creation of the Smart Mirror. The frame was constructed from 1 in. x 6 in. x 8 ft. stained and beveled brown “barn” wood [23], while the casing was of 1 in. x 6 in. x 8 ft. “common” wood [24].

4.3 **Constraints, Alternatives, and Tradeoffs**

4.3.1 **GUI Design Language**

Another method that could be used to create the GUI is to use native C/C++ libraries such as Qt, which are faster and more memory efficient compared to web technologies and do not require a browser application to run. The advantages of using HTML, CSS, and JavaScript is that they are platform independent and can be easily ported to run on different kinds of hardware. Another advantage is that since Python and Ruby are Object Oriented Programming languages, it is easier to build and extensible framework as well as the modules for that framework, which lowers the barrier for developers who want to make their own custom GUI components.

4.3.2 **User Interaction Method**

In the first prototype, the initial plan was to use a touch capable display to receive user interaction via a capacitive touch screen. However, the touch screen could not be activated behind the acrylic mirror. Therefore, two Hover sensors detecting gestures via micro-capacitance readings were used. Using these gesture sensors had the advantage of preventing mirror smudging.

5. **Schedule, Tasks, and Milestones**

Appendix A contains a table of major tasks and their respective owners. Please see section 9.

Leadership Roles for more information regarding overall contribution of each team member. Appendix B contains a Gantt chart, showing the project timeline for all tasks and milestones.

Designing and implementing the transparent display technology could not be accomplished within the allotted time, as this is a nascent field with a variety of options and difficulty in manufacturing.

Although the plan was to finish the first hardware prototype before spring break, most of the construction was done the week after spring break. This was mainly due to delays in acquiring components.

6. **Final Project Demonstration**

The prototype demonstration was done at McCamish Pavilion during the Capstone Expo. The demonstration consisted of the following:

1. The user logged into the smartphone application with a Google account.
2. The user paired the smartphone application to the mirror using the unique identifier string.
3. The user configured personalized information, such as weather location, news source, email, and calendar login information via the smartphone application.
4. Finally, the user performed gestures to interact with the Smart Mirror through the two Hover sensors attached to the frame.

7. **Marketing and Cost Analysis**

7.1 Marketing Analysis

The intended market for the Smart Mirror prototype consists of consumers who wish to gain more functionality from their wall-mounted mirrors. Because transparent display technology is a concept still being researched and developed, no commercially available Smart Mirrors exist on the market. Corning, Inc. Intel, Inc. and MemoryMi are collaborating to create the MemoryMirror to allow shoppers to “try on” new clothing without entering the dressing room. While the MemoryMirror claims to be a digital mirror, it is actually a translucent glass that displays images captured by a mounted camera. To avoid latency and security concerns associated with placing a camera in a home, the Smart Mirror featured featured a two-way mirror instead.

In addition, the ESGT platform provides for synchronization and backup of user configuration across all devices through Firebase. There is also an accompanying smartphone application to configure devices, so that users do not need to manually edit or install files to programmatically customize information such as location or API keys.

7.2 Cost Analysis

7.2.1 Cost of Components

An approximate of the cost of the Smart Mirror prototype is \$500.04. Successful construction of this prototype serves as a proof-of-concept for a final version of the product and glass technologies ecosystem.

Table 2. Smart Mirror Prototype Equipment Costs

Component	Cost
-----------	------

Raspberry Pi 3 Model B	\$35.00 [25]
Raspberry Pi Power Supply	\$8.99 [26]
Two-Way Mirror	\$72.00 [27]
Touchscreen LCD Monitor	\$289.95 [28]
HDMI Cable	\$4.17 [29]
Hover 2.0 Sensor (2x)	2x \$29.00 [30]
Phototransistor [BL1]	\$0.95 [31]
Temperature-Humidity Sensor [BL2]	\$5.00 [32]
1 in. x 6 in. x 8 ft. Barn Wood Brown Shiplap Pine Wood	\$12.33 [23]
1 in. x 6 in. x 8 ft. Common Board	\$8.07 [24]
Wood Screws	\$5.58 [33]
Total	\$500.04

7.2.2 Development Costs

Development hours invested per engineer is itemized in Table 3. Five engineers will work to develop the Smart Mirror prototype. Besides group meetings, assembly was the most time consuming task, as anticipated in the previous semester. Assembly accounts for the building of the frame, case, sensor wiring and integration, among other tasks.

Task	Hours
-------------	--------------

Group Meetings	100
Report Preparation	8
Poster	4
Expo Preparation	1
Research	15
Fabrication	15
Assembly	30
Testing	25
Total	198

Assuming fringe benefits are 34% of labor and overhead is 120% of labor and equipment, Table 3 shows the total development costs associated with one Smart Mirror prototype. At 198 labor hours and an assumed salary of \$58,000 per engineer, the resultant labor expenses amount to \$5,520 per engineer. Therefore, the labor expenses for all five engineers is \$27,600.

Development Component	Cost
Equipment for Smart Mirror	\$500
Labor	\$27,600
Fringe Benefits, % of Labor	\$9,384
Subtotal	\$37,459
Overhead, % of Equipment, Labor, & Fringe Benefits	\$44,951
Total	\$82,436

7.2.3 Selling Price and Profit

The production of 4,000 Smart Mirror units employed with transparent display and sensor circuitry will be sustained over five years (800 units per year). When buying in bulk, the Raspberry Pi 3 can be purchased at a discount for \$34.50 per unit [25]. Buying in bulk yields a total equipment cost of \$64 per unit. Advertising the product presents a 7% sales expense of the final selling price, which is \$28. A group of workers will be employed at an hourly rate of \$25 to assemble and test the final products. At a market price of \$400, the expected total revenue is \$1,600,000. This reflects a \$134 profit per unit sold, and a \$800,000 profit over the five year production period. The selling price and expected profit per unit is displayed in Table 4.

Table 5. Selling Price and Profit Per Unit (Based on Production of 4,000 units)

Expense or Income	Dollar Amount
Equipment cost	\$64
Assembly Labor	\$10
Testing Labor	\$10
Total Labor	\$20
Fringe Benefits, % of Labor	\$6
Overhead, % of Equipment, Labor & Fringe Benefits	\$107
Subtotal, Input Costs	\$217
Sales Expense	\$28
Amortized Development Costs	\$21
Subtotal, All Costs	\$266
Profit	\$134
<i>Selling Price</i>	<i>\$400</i>

8. Conclusion

The current prototype is complete with a Raspberry Pi microcontroller driving the display, running the frontend GUI and backend server, and reading the sensor values.

The display used for the final prototype was an LCD screen with a capacitive touch screen. There were two major issues with the display: it could not detect touches behind the acrylic mirror, and the display still emitted a backlight when the screen was black. A better alternative would have been an OLED display without a touch screen. OLED displays can completely turn off each pixel, preventing the backlight issue. In addition, foregoing the touch screen would have allowed the team to purchase a much larger display to cover more of the mirror's area.

In addition, the team should have ordered the parts earlier in the semester. Although the team ordered them a week before spring break, they arrived at the beginning of April. To allow ample time to test different hardware configurations, the team should have ordered them by end of January.

9. Leadership Roles

9.1 Najee Kitchens

Najee volunteered as the group's webmaster. Before the server space was available to post the team's webpage, she created and edited the website using Github. On the website, she included a description of the ESGT project, links to deliverables, important completion milestones, and a short biography of each member.

Along with be webmaster, Najee worked on the software frontend team with Nishant. She worked on designing two components: the weather component and the clock component. In the weather component, she used weather icons to correspond to the weather of that time.

Najee also played a major part in helping put together the front frame of the mirror. In the Invention Studio, she use the power saw to cut 45 degree angles in the wood for the corners of the frame.

9.2 Kairi Kozuma

Kairi acted as the team leader and documentation coordinator. He was responsible for drafting the initial proposal, summary, and final report. He also made sure each member was assigned a section to complete in the proposal and final report.

Kairi's primary technical role was as a backend software developer lead. He was in charge of designing the PostgreSQL database tables and REST API that the frontend team accesses for information. To aid frontend team development, he set up EC2 instances in AWS that provided constant access to mock data, using the APIs agreed upon by both the frontend and backend team. He also worked with the hardware team, writing an updater script that read sensor values from the Mbed virtual serial port.

In addition, Kairi worked as the primary mobile application developer, integrating the Firebase real-time database into the backbone of the application and providing hooks for the frontend team.

9.3 Boa-Lin Lai

Boa-Lin's main role was to design and print the transparent circuitry for showing the proof of concept for the Ecosystem for Smart Glass Technologies. He also helped design, debug, and test the hardware features for the team.

Boa-Lin worked closely with Bijan in ATHENA lab on Transparent Circuitry. He designed the layout of the circuit with Eagle CAD, and perform the final testing for the circuit. The final circuitry had a 3 pin connector and 2 surface mounted resistors. This transparent pull-up resistors circuitry was used to show the concept of future approach on Smart Glass product. It minimized the wires used and kept the glass product transparent.

9.4 Jonathan Osei-Owusu

Jonathan's responsibilities as sensor integration lead included purchasing, programming, and validating the functionality of all analog sensors used in the Smart Mirror prototype. He also ensured reliable and swift serial communication between the Mbed slave and the Raspberry Pi 3 master.

Additionally, Jonathan purchased wood, glue, and other materials to facilitate the creation of the Smart Mirror's casing and frame. He worked closely with the software team to ensure that the frame's design did not conflict with the front-end website design and user experience. Further responsibilities included designing a layout for each ventilation hole and analog sensor on the mirror's frame, meanwhile rationing building supplies to keep costs low.

Further, Jonathan was responsible for creating and directing the team's video demo of the Smart Mirror.

9.5 Nishant Shah

Nishant's main role was to implement the frontend graphical user interface for the mirror, which involved creating standalone components that would either integrate into a backend datasource or rely on system information. These components were then arranged into the final user interface taking into account design and usability. Nishant also worked on adding support for gesture sensor inputs to the frontend using the hover sensors [30].

10. References

- [1] J. Greenough, "The 'Internet of Things' will be the world's most massive device market and save companies billions of dollars", *Business Insider*, Feb. 18, 2015. [Online]. Available: <http://www.businessinsider.com/the-internet-of-things-market-growth-and-trends-2015-2>. [Accessed Nov. 20, 2016].
- [2] R. Henderson, "Best smarthome and IoT gadgets of CES 2016", *Pocket Lint*, Jan. 11, 2016. [Online]. Available: <http://www.pocket-lint.com/news/136306-best-smarthome-and-iot-gadgets-of-ces-2016-samsung-lg-and-more>. [Accessed Nov. 14, 2016].
- [3] A. Stadler, "Transparent conducting Oxides—An up-to-date overview," *Materials*, vol. 5, no. 12, pp. 661–683, Apr. 2012.
- [4] K. Kiyokawa, Y. Kurata, and H. Ohno, "An optical see-through display for mutual occlusion with a real-time stereovision system," *Proc. IEEE*, vol. 25, no. 5, pp. 765–779, Oct. 2001.
- [5] Planar Systems Inc., "Transparent OLED Displays", *Planar Systems Inc.*, n.d. [Online]. Available: <http://www.planar.com/innovations/transparent-oled/>. [Accessed Nov. 20, 2016].
- [6] G. Barrett and R. Omote, "Projected-Capacitive Touch Technology," *Information Display*, vol. 32, no. 3, p.16, March, 2010. [Online serial]. Available: http://www.walkermobile.com/March_2010_ID_Projected_Capacitive.pdf. [Accessed Nov. 20, 2016].
- [7] Cypress Semiconductor Corp., "TrueTouch® Multi-Touch All-Points Touchscreen Controller," CYTMA568 datasheet, May 2016.
- [8] G. Walker. Part1: Fundamentals of Projected-Capacitive Touch Technology. Presented at SID Display Week '14. [Online]. Available: http://www.walkermobile.com/Touch_Technologies_Tutorial_Latest_Version.pdf. [Accessed Nov. 20, 2016].

- [9] S. H. Bae and Y. S. Lee, "Implementation of a High-Performance Touch Controller and Differential Sensing Circuit," *International Journal of Computer and Information Technology*, vol. 3, no. 6, p. 1177, November, 2014. [Online serial]. Available: <http://www.ijcit.com/archives/volume3/issue6/Paper030602.pdf>. [Accessed Nov. 20, 2016].
- [10] Technavio, "Gesture Recognition, Coming to a Smartphone Near You", *Infiniti Research Limited*, Jun. 30, 2015. [Online]. Available: <http://www.technavio.com/blog/gesture-recognition-coming-to-a-smartphone-near-you>. [Accessed Nov. 20, 2016].
- [11] Microsoft, "Kinect Sensor for Xbox 360 Components", *Microsoft*, n.d. [Online]. Available: <https://support.xbox.com/en-US/xbox-360/accessories/kinect-sensor-components>. [Accessed Nov. 20, 2016].
- [12] R. Boyle, V. Hlavac, and M. Sonka, *Image Processing, Analysis, and Machine Vision*, 4th ed., Stamford, CT: Cengage Learning, 2014. [E-book]. Available: <https://goo.gl/yv3m0r>. [Accessed Nov. 20, 2016].
- [13] Y. Lin, H. Tang, and T. Wu, "Real-time Object Image Tracking Based on Block-Matching Algorithm", *University of Wisconsin-Madison*, 2006. [Online]. Available: <https://homepages.cae.wisc.edu/~ece734/project/s06/lintangwuReport.pdf>. [Accessed Nov. 20, 2016].
- [14] S. Hymel, "Single Board Computer Benchmarks - learn.sparkfun.com.", *learn.sparkfun.com*. [Online]. Available: <https://learn.sparkfun.com/tutorials/single-board-computer-benchmarks>. [Accessed Nov. 20, 2016].
- [15] Raspberry Pi Foundation, "Raspberry Pi FAQs - Frequently Asked Questions," *Raspberry Pi Foundation*, n.d. [Online]. Available: <https://www.raspberrypi.org/help/faqs/>. [Accessed Nov. 20, 2016].

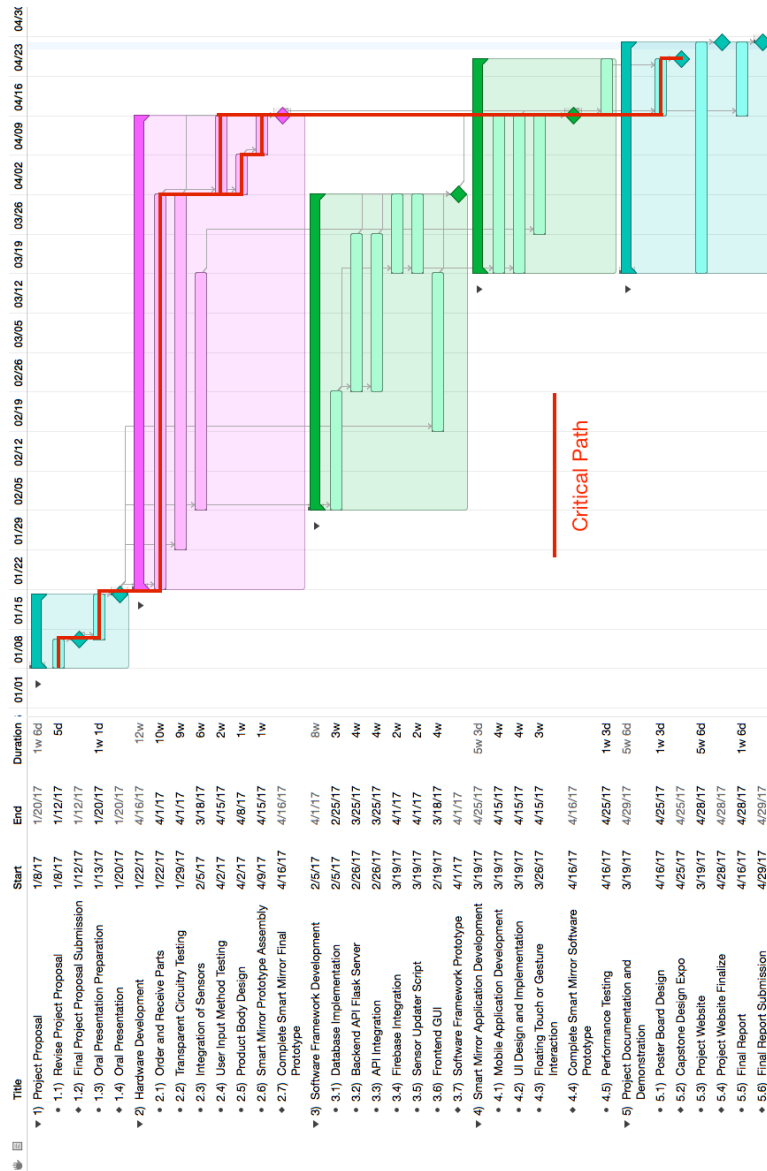
- [16] V. Varma, “Wireless Fidelity—WiFi”, *IEEE Emerging Technology Portal*, 2012. [Online serial]. Available: <https://www.ieee.org/about/technologies/emerging/wifi.pdf>. [Accessed Nov. 20, 2016].
- [17] Bluetooth SIG, Inc., “What is Bluetooth?”, *Bluetooth SIG, Inc.*, 2016. [Online]. Available: <https://www.bluetooth.com/what-is-bluetooth-technology/bluetooth-technology-basics>. [Accessed Nov. 20, 2016].
- [18] E. You, “The Progressive JavaScript Framework”, *Vuejs.org*, n.d. [Online]. Available: <https://vuejs.org>. [Accessed Apr. 29, 2017].
- [19] P. Harp, “Sharp IR Sensor Displayed Through Virtual COM Port”, *ARM Ltd.*, n.d. [Online]. Available: <https://developer.mbed.org/users/pharp6/notebook/sharp-ir-sensor-displayed-through-virtual-com-port/>. [Accessed Apr. 28, 2017].
- [20] HDMI Licensing Administrator, Inc., “Inside an HDMI cable”, *HDMI Licensing Administrator, Inc.*, n.d. [Online]. Available: <http://www.hdmi.org/installers/insidehdmicable.aspx>. [Accessed Nov. 20, 2016].
- [21] World Standards, “Complete list: Plug, socket & voltage by country,” *World Standards*, 2017. [Online]. Available: <http://www.worldstandards.eu/electricity/plug-voltage-by-country/>. [Accessed Nov. 21, 2016].
- [22] B. Erickson, “Testing Wall Anchors and Picture Hangers”, *Today's Homeowner*, 2016. [Online]. Available: <http://www.todayshomeowner.com/testing-wall-anchors-and-picture-hangers/>. [Accessed Nov. 20, 2016].
- [23] The Home Depot, “1 in. x 6 in. x 6 ft. Barn Wood Dark Brown Shiplap Pine Board (6-Pack)”, *The Home Depot Product Authority, LLC*, n.d. [Online]. Available: <http://www.homedepot.com/p/1-in-x-6-in-x-6-ft-Barn-Wood-Dark-Brown-Shiplap-Pine-Board-6-Pack-263242/300198780>. [Accessed Apr. 29, 2017].

- [24] The Home Depot, “1 in. x 6 in. x 8 ft. Common Board”, *The Home Depot Product Authority, LLC*, n.d. [Online]. Available: <http://www.homedepot.com/p/1-in-x-6-in-x-8-ft-Common-Board-914770/202074483>. [Accessed Apr. 29, 2017].
- [25] MCM Electronics, “Raspberry Pi™ 3 Model B 1GB Project Board”, *MCM Electronics*, n.d. [Online]. Available: <https://goo.gl/ThFwjJ>. [Accessed Apr. 29, 2017].
- [26] MCM Electronics, “Official Raspberry Pi™ Foundation 5V 2.5A Power Supply”, *MCM Electronics*, n.d. [Online]. Available: <https://goo.gl/5bMRGz>. [Accessed Apr. 29, 2017].
- [27] TAP Plastics, “2-Way Mirrored Acrylic Sheets”, *TAP Plastics*, n.d. [Online]. Available: http://www.tapplastics.com/product/plastics/cut_to_size_plastic/two_way_mirrored_acrylic/558. [Accessed Apr. 29, 2017].
- [28] Newegg, Inc., “Dell P2314T Black 23" IPS 8ms (GTG) Touchscreen LCD/LED Monitor, 270 cd/m2 8000000:1, USB 3.0 Port with 10 Touch Points Function with Smooth Operation”, *Newegg, Inc.*, n.d. [Online]. Available: <https://goo.gl/5RCi8K>. [Accessed Apr. 29, 2017].
- [29] Walmart, “Highs High Speed HDMI Cable with Ethernet, 6'”, *Walmart Stores, Inc.*, n.d. [Online]. Available: <https://goo.gl/5YDkiW>. [Accessed Apr. 29, 2017].
- [30] Hover Labs, “Hover 2.0”, *Hover Labs*, n.d. [Online]. Available: <http://www.hoverlabs.co/shop/hover-20>. [Accessed Apr. 29, 2017].
- [31] Adafruit, “Photo Transistor Light Sensor”, *Adafruit*, n.d. [Online]. Available: <https://www.adafruit.com/product/2831>. [Accessed Apr. 29, 2017].
- [32] Adafruit, “DHT11 basic temperature-humidity sensor + extras”, *Adafruit*, n.d. [Online]. Available: <https://www.adafruit.com/product/386>. [Accessed Apr. 29, 2017].
- [33] The Home Depot, “#12 2-1/2 in. Phillips Flat-Head Wood Screws (25-Pack)”, *The Home Depot Product Authority, LLC*, n.d. [Online]. Available: <http://www.homedepot.com/p/Everbilt-12-2-1-2-in-Phillips-Flat-Head-Wood-Screws-25-Pack-21252/202102322>. [Accessed Apr. 29, 2017].

Appendix A – Task Assignment and Risk Level

Task Name	Task Lead	Difficulty	Risk Level
Proposal, Documentation, and Presentation	All	Low	Low
Project Proposal	All	Low	Low
Oral Presentation	All	Low	Low
Project Website	NK	Low	Low
Final Report	All	Low	Low
Demonstration	All	Medium	Medium
Capstone Design Expo Preparation	All	Medium	Low
Hardware Development	BL, JO	High	High
Order and Receive Parts	BL, JO	Low	Medium
User Input Method Testing	BL, NS	Medium	High
Setup Mbed Microcontroller as Slave Device	BL, JO	Medium	Low
Product Body Design	All	Medium	Medium
Final Hardware Assembly	All	High	High
Transparent Circuitry Sample Testing	BL	High	High
Sensor Conversion of Signal to Meaningful Values	JO	Low	Medium
Software Framework Development	KK, NK, NS	Medium	Low
Database Implementation	KK	Low	Low
API Integration	KK, NS	Low	Low
Performance Optimization	NS	Medium	Medium
Firebase Integration	KK	Low	Medium
Sensor Updater and Insertion into Database	KK	Low	Low
Frontend GUI	NK, NS	Medium	Low
Backend API Flask Server	KK	Medium	Low
Smart Mirror Application Development	All	Low	Low
UI Design and Implementation	NK, NS	Medium	Low
Hover Gesture Interaction	NS	Medium	High
Mobile Application and Setup Software	KK, NS	Medium	Low
Graphics Optimization	NS	Medium	Low
Contingency Planning	All	Medium	Medium
LCD Display with Two-Way Mirror	All	Low	Medium

Appendix B – Gantt Chart

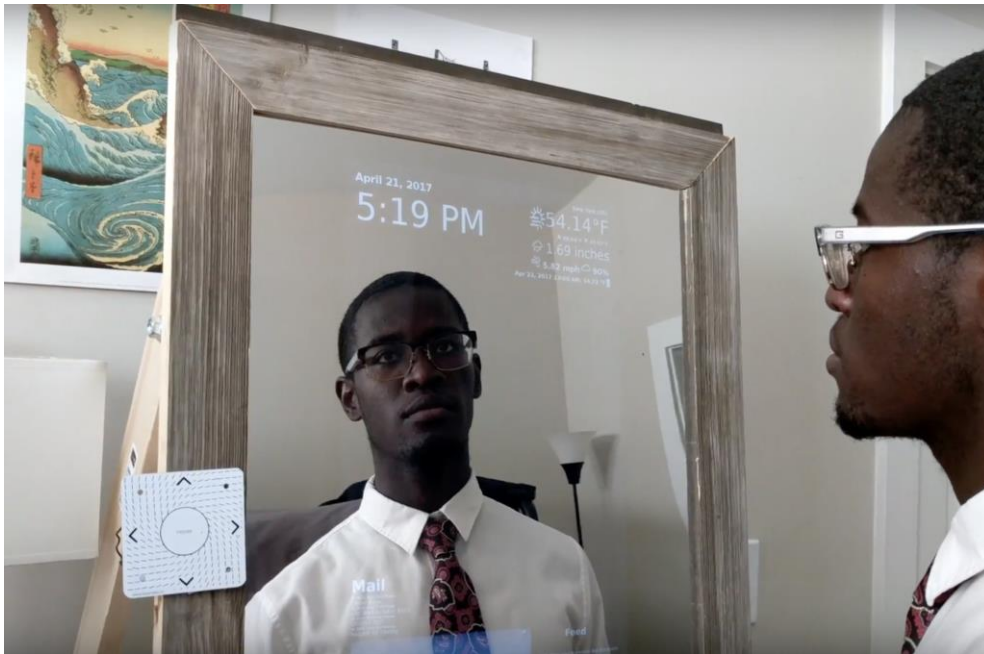


Appendix C – Smart Mirror GUI

The Smart Mirror GUI displays the following information:

- Date:** April 23, 2017
- Time:** 12:51 AM
- Location:** New York (US)
- Weather:** Cloudy, 48.9°F. High: 59.5°F, Low: 45.43°F. Rain: 0.54 inches. Wind: 5.19 mph. Humidity: 75%.
- Light Level:** 1537.834 lux, accompanied by a line graph showing light intensity fluctuations.
- Mail:**
 - As: 04-21, From: ESGT Project, Dinner Show
 - As: 04-21, From: ESGT Project, Just Want to Get in Touch
 - As: 03-05, From: Kairi Kozuma, URGENT TEST
 - As: 03-05, From: Kairi Kozuma, Request for Meeting
- Agenda:**
 - 04-27: Meeting with Team
 - 04-28: All Day Party
 - 05-02: Another Fun Event
 - 05-03: Dinner with Friends
 - 05-09: My Birthday
 - 05-12: Birthday Party
 - 05-14: Flower Shopping for Mother's Day
- Feed:**
 - Video: "This is Tokyo Tokyo City Time lapse" (Tokyo skyline at night).
 - Image: "March for Science Demonstrators Say They're the Real Patriots" (Protesters holding signs).

Appendix D – Smart Mirror Prototype



Appendix E – Links to Source Code, Media, and Documentation

The Magicians (ECE 4012L3A)

Project Proposal PDF	http://ece4012y2017.ece.gatech.edu/spring/sd17sET2/ESGT_Project_Proposal.pdf
Project Proposal Docx	http://ece4012y2017.ece.gatech.edu/spring/sd17sET2/ESGT_Project_Proposal.docx
Project Presentation PDF	http://ece4012y2017.ece.gatech.edu/spring/sd17sET2/ESGT%20Presentation.pdf
Project Presentation PPTX	http://ece4012y2017.ece.gatech.edu/spring/sd17sET2/ESGT%20Presentation.pptx
Project Summary PDF	http://ece4012y2017.ece.gatech.edu/spring/sd17sET2/ESGT_final_project_summary.pdf
Project Summary Docx	http://ece4012y2017.ece.gatech.edu/spring/sd17sET2/ESGT_final_project_summary.docx
Poster PDF	http://ece4012y2017.ece.gatech.edu/spring/sd17sET2/images/esgt_poster.pdf
Poster PPTX	http://ece4012y2017.ece.gatech.edu/spring/sd17sET2/images/esgt_poster.pptx
Promotional Video	https://youtu.be/tykuLydx-dc
Display Video	https://youtu.be/Uh05ecy40KM
Frontend Source Code	http://ece4012y2017.ece.gatech.edu/spring/sd17sET2/images/frontend-master.zip
Framework Source Code	http://ece4012y2017.ece.gatech.edu/spring/sd17sET2/images/hardware-framework-master.zip