Queen’s University

Assignment:

Main Project Final Report

Course:

ELEC 490

Group Number:

36

Supervisor:

Dr. Steven Blostein

Prepared by:

Adams Liu | #: 10187602 | 14zxl@queensu.ca

Ian Chang | #: 10190262 | 14yc65@queensu.ca

Kingsley Chu | #: 10182268| 14kmc@queensu.ca

Due Date:

April 5th, 2020

Table of Contents

Table of Figures

Table of Tables

Executive Summary

As the world we live in continuously evolves and modernizes with new technologies, humans as well aim to optimize and improve their schedules and resources. However, the most important resource that no money can buy is time, and by identifying key chokepoints of data integration and acquisition in one's daily schedule, this project aims to streamline the deliverance of important information to the end-user. With the use of a Magic Mirror, this futuristic project integrates modern user-centric design with cutting edge technology focused on delivering the most informative washroom experience yet, through the application of various engineering feats, such as building a fully functional and ergonomic user interface with beneficial functions that symbiotically blends with the physical world environment, as well as a natural language processing module for user input. By integrating human-centric design with the agile software development process, this Magic Mirror leverages existing technologies such as Amazon Echo, Raspberry Pi, Electron, and Web development to bring the future to you.

Motivations and background

The term Internet of Things (IoT) describes "technologies, systems, and design principles associated with the emerging wave of Internet-connected things that are based on the physical environment" [1]. IoT is further explained as a network of uniquely identifiable objects, things, and their virtual representations in an internet-like structure [2]. To be simply put, the IoT concept is exemplified when an object has the ability to transfer data over a network without the need for human to computer interaction. IoT comprises four components, including sensing, heterogeneous access, information processing, and applications and services [2].

With the advancement of technology, people want to access information more easily. Whether it is through the television or the internet, people want to be informed of the news and the world's current affairs. Along with the growth of smartphone users, consumer adoption of IoT devices has over greatly increased over the past few years. According to the International Data Company (IDC), worldwide spending on the Internet of Things was forecast to surpass the $1 trillion mark in 2022, a double-digit annual growth rate throughout the 2017-2022 period [3]. Industries, governments, and consumers are quickly adopting IoT into daily life. Current well-developed commercial products such as smart door locks, smart plugs, and smart thermostats have been changing the way consumers are informed on the status of households, vehicles, family members, and even their own health.

Among the smart home products that have been developed, the mirror remains in a relatively primitive state. In 2006, a research paper on "information-accessing furniture" was published in Tokyo, Japan, detailing one of the early prototypes of a smart mirror, called the "Miragraphy" [4]. The system was able to display news headlines and personal picture galleries [4]. A "smart rear-view mirror" was developed in 2014 by Nissan, which consisted of a mirror with a built-in LCD screen that displayed high-resolution videos of the car's rear flanks [5]. While research has been done in the past on a smart mirror system, the adoption of smart mirrors has been low due to the high cost of production, insufficient functionality, and lack of integration between the smart mirror and other smart home devices [6].

To deal with the situation, the Smart Mirror system was developed to provide users with the convenience of accessing relevant information in addition to managing their smart home devices. The Smart Mirror enables the users to provide oral instructions, and the system will recognize the user's voice to receive instructions and respond to the users' needs. Moreover, the Smart Mirror system will be implemented on a low-cost system on chip (SOC) devices such as the Raspberry Pi.

The smart mirror system will be developed based on three objectives shown below, as major components key to functionality:

1. Use the Raspberry PI as a component in the hardware design.
2. Incorporate a voice control system that enables smart mirror navigation.
3. Enable the mirror to be user-customizable

System Design

From the beginning, we understood our project had a heavy emphasis on integration, which requires all components of our system to be working simultaneously and working with each other. These subsystems included the hardware system, the user display system, and the voice-controlled system. If one subsystem were to fail, then the entire smart mirror system would not be able to operate. For example, if the PIR sensor did not work, then the monitor would not be able to turn on, which meant that the user display and voice control subsystems would also be unusable. Our team understood that building the smart mirror system would have to be a sequential approach, and priority needed to be focused on what is most crucial to the operation at the time. For example, it would be unwise to start programming the RGB LEDs before getting the voice-controlled system operating first, as the primary purpose of the mirror is to give users a voice-controlled display. This made decision making very easy as everyone in the team agreed on what needed to be focused on. Figure 1 is a diagram that displays a flow chart of tasks and in what order they should be done in

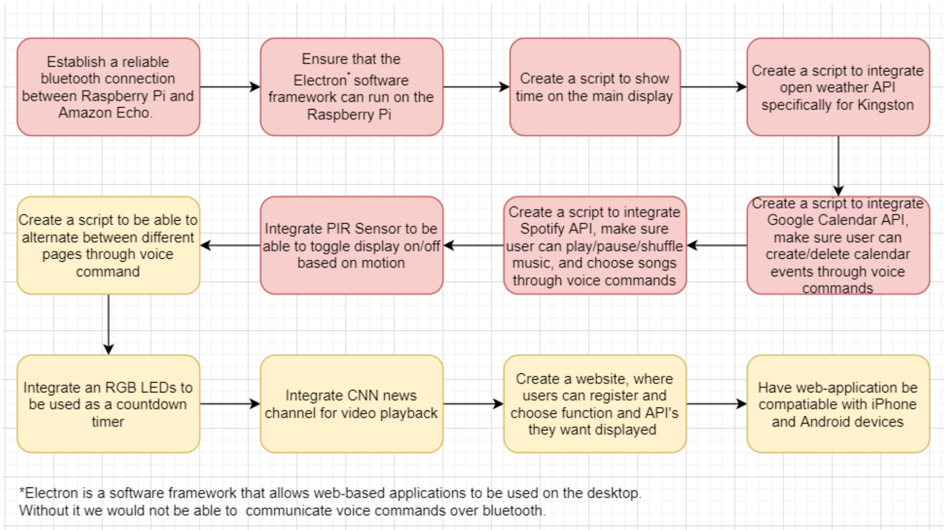


Figure 1 Flow-chart of Tasks

The red boxes highlight the requirements that our team wanted to accomplish by the end of the year, and the yellow boxes indicate stretch goals that our team would like to accomplish if time allowed. Thankfully, we were able to meet expectations and complete all crucial tasks (red) as well as accomplishing the first stretch goal (yellow).

Functional Requirements

The functional requirements of our project are structured based according to the design process as well. Table 1 shows the functions of our system as well as an example of voice commands (if required) that the mirror system can perform to trigger those functions.

Table 1 Functional Requirements with Example Voice Commands

Functional Requirement	**Example of Voice Command**
The mirror will be able to toggle on/off through voice command.	• "Alexa, turn on the mirror" • "Alexa, turn off the mirror"

The mirror will be able to accurately display the current time (Eastern Standard Time)	
The mirror will be able to connect to OpenWeatherMap API service and display daily weather for Kingston, Ontario.	
The mirror will be able to connect to Google Calendar's API service, and users will be able to add/delete/update calendar events that will display dynamically on the mirror	• "Alexa, add Event for 3 O'clock this afternoon to go pick up Sally from daycare • "Alexa, delete meeting with Kevin today" • "Alexa, update meeting with Jillian, from 7 O'clock to 7:30"
The mirror will be able to connect to Spotify's API service, and users are able to pause/play/shuffle/choose songs through voice command. The display should be able to match said command.	• "Alexa, play music" • "Alexa, stop the music" • "Alexa, play songs by Justin Bieber"
The mirror's display will be activated only if the PIR sensor detects movement within 3m of its peripheral	
When no movement is detected for 30 seconds the mirror's display will turn off	
The mirror will be able to navigate to different user pages through voice command and a given name	• "Alexa, go to Mark's page"

Constraints

Due to time constraints, we were unable to implement a customizable display where users can choose which Nodes and APIs that they would like to be displayed. Currently, the functions are pre-set in the backend, and unless users have intermediate node.js knowledge, it is very cumbersome to change the display. In the future, we hope to create a drag-and-drop web application that allows users to customize their display tailored to their needs.

Another constraint our product has is the limited account constraint. Currently, the system only allows one Google account, and one Spotify account to be connected. This is not ideal, considering we intend the mirror to support multiple users. Having a database of users for the mirror system will be implemented in future iterations.

The final major constraint is the language constraint. Currently, the voice-controlled system only supports American English. Therefore, the mirror would not be suitable for anyone who is not is an English speaker. It also does a poor job with accents as well. Later iterations of the mirror will include more languages to support international users.

Solution Implementation

Software Specifications and Implementation

As a bulk of the project was implemented through software development, various development tools and technologies were used to accomplish the project task. This includes utilizing web development technologies integrated with the internet of things (IoT) devices. To create the software application to achieve the project task, several key packages were utilized based on simplicity, effectiveness, and overall usability. This can be illustrated in figure 2 below

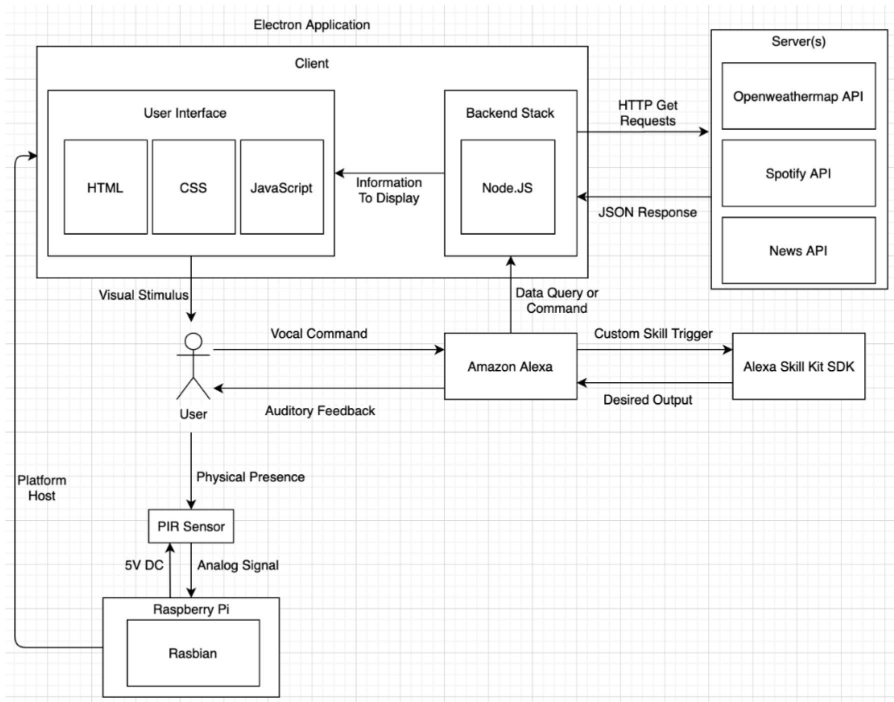


Figure 2 Level 1 data flow diagram of the magic mirror software

In addition, the use case diagram for the software system is shown below in figure 3. This outlines the possible ways for an actor or user to interact with the Magic Mirror system:

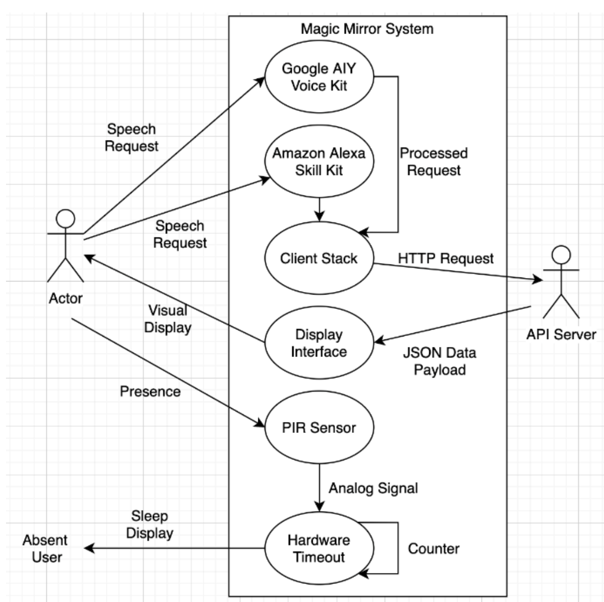


Figure 3 A use case diagram for the software of the Magic Mirror

Moreover, as the main application of the project is a visual interface designed to effectively relay information to the user, a web-based program running on Electron was chosen. This entails a front-end stack comprised of HTML, CSS, and JavaScript to build the user interface, selected for stability, maintainability, and simplicity. In addition, and monochromatic color scheme with a black background would allow white text to stand out through the one-way mirror clearly while avoiding interference with the reflection An original prototype static mockup, used as a low fidelity (Lo-Fi) prototype is shown in figure 4:

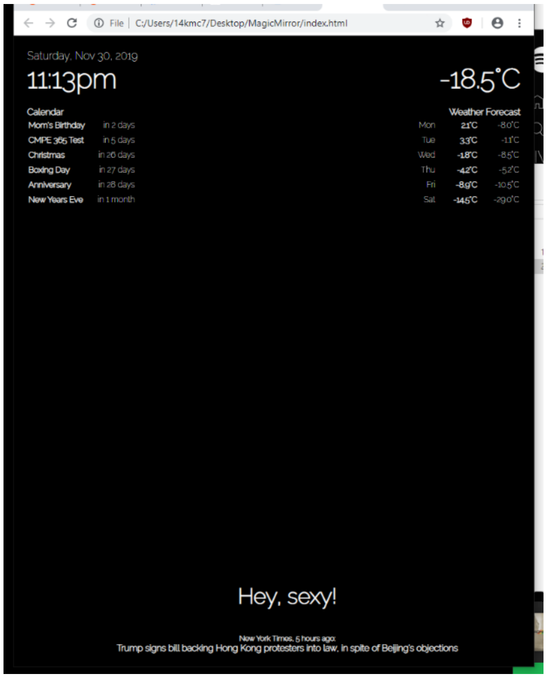


Figure 4 The orignal Lo-Fi static mockup of the user interface

Later, a high fidelity (Hi-Fi) front-end stack would utilize basic HTML to display text and icons, which are key to creating a clear and basic interface that would be relayed through the IPS display and one-way mirror. Such items would be conveyed using dedicated widgets for specific functions, which are listed below:

- Time and date, which utilizes built-in JavaScript functionality such as getDate() with options for formatting such as ISO8601, to display the current time and date as the main widget. This is stylized by CSS, which affects the visual aesthetic of the date.
- Weather widget, which is dedicated to displaying weather data alongside the current time and date. This weather widget utilizes a dedicated representative state transfer application programming interface (REST API), specifically the openweathermap.com weather API, as shown in figure 5. This utilizes a Node.js framework backend, which allows the front-end to produce usable functionality. This JavaScript library is built to allow the transference of weather data through a JavaScript Object Notation (JSON) file, which returns desired weather data such as temperature, condition, high, low, and upcoming forecast for the weather as a response to a Hypertext Transfer Protocol (HTTP). This HTTP request (in the form of GET), will request information based on the location, amount of data needed, and format of data in metric or imperial units. The response is parsed, and data is conveyed into a dedicated weather class object, with fields such pertaining to relevant information. In addition, the API will return location data along with the weather data, which can be useful in display in the time zone and location. This data is also parsed through the HTML, CSS front end stack, and stylized in a clean aesthetic for display. An example is provided in figure 6 below

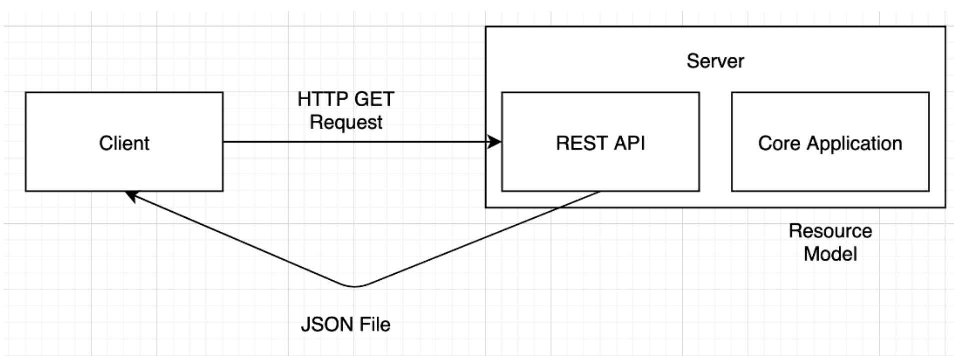


Figure 5 A simplified workflow diagram displaying the relationship between the server and client executing an API GET request

```javascript
const request = require('request');
const argv = require('yargs').argv;

let apiKey = '*****************************';
let city = argv.c || 'portland';
let url = `http://api.openweathermap.org/data/2.5/weather?q=${city}&units=imperial&appi

request(url, function (err, response, body) {
  if(err){
    console.log('error:', error);
  } else {
    let weather = JSON.parse(body)
    let message = `It's ${weather.main.temp} degrees in ${weather.name}!`;
    console.log(message);
  }
});
```

Figure 6 A code snippet of Node.JS utilizing an HTTP GET request to return a JSON containing requested weather data

- Spotify widget, which utilizes a proprietary API through a preregistered Spotify account, which will allow for music to be played back through the connected Amazon speaker. This Spotify widget displays the current track name, artist, as well as album art to allow the user to visually perceive the song. With similar functionality to the weather widget, the Spotify widget will require a similar protocol when updating and requesting information from the API through the internet, with the added addition of authentication, as shown in figure 7. Due to the monetization, legalities, and costs associated with listening to music, Spotify's paid account requires the user to be verified through OAuth2.0, which is a special authentication protocol and open standard required for access delegation, used for internet users to grant websites or applications access to information. Users will not need to deal with the OAuth authentication, as the Node.js backend code will accept the user's credentials during setup and resolve authentication for Spotify use.

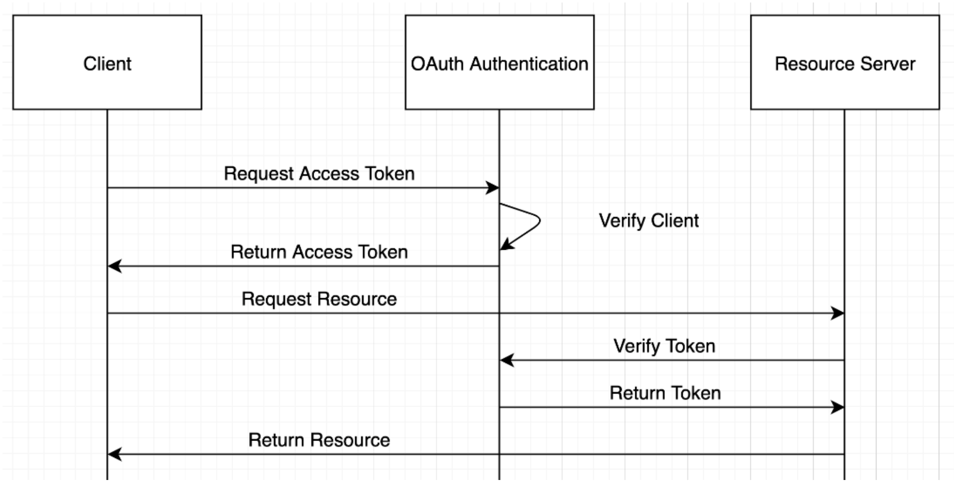


Figure 7 Diagram flowchart to illustrate the OAuth 2.0 authentication process to request a song through the Spotify API

- Another feature or widget is the quote of the day, which, unlike the other widgets, does not utilize a specific API through internet connectivity, but instead polls from a randomly generated list of quotes. These will include encouraging or entertaining phrases, which is displayed to the user through the front-end stack. By utilizing a random integer generator function in JavaScript, random(), a randomized index retrieves a specific phrase from an array or dictionary of strings, which is parsed, displayed, and stylized with HTML and CSS as shown in figure 8. Such phrases include “You miss 100% of the shots you don’t take – Wayne Gretzky” and “Hey, sexy!”. This poll will update once per day, and the widget will adjust accordingly. However, it was decided in the final iteration to omit this feature.

```
console.log(Math.floor(Math.random() * 10))
```

Figure 8 A code snippet of an RNG (random number generator) that is used to poll a preset array of quotes

- A news headlines aggregate that displays headlines like the weather data, which polls from a specific API such as News API through the form of an HTTP get request, with a JSON response.

This allows for only the news headline to be shown through the front-end stack, which is refreshed every minute.

Finally, addressing the hardware platform that the application is run upon, a cross-platform desktop program would be ideal, as it can allow developers to improve and design upon multiple iterations during the prototype testing on personal computers while taking full advantage of the Raspberry Pi's portable system. This requires a cross-platform, simple, and light framework that can utilize existing web technologies, which Electron is suited for. This reduces the need to run the program through a web browser, such as Firefox or Chrome. A sample code snippet of wrapping an index.html file through an Electron application is shown below in figure 9:

```
const { app, BrowserWindow } = require('electron')

function createWindow () {
  // Create the browser window.
  let win = new BrowserWindow({
    width: 800,
    height: 600,
    webPreferences: {
      nodeIntegration: true
    }
  })

  // and load the index.html of the app.
  win.loadFile('index.html')
}

app.whenReady().then(createWindow)
```

Figure 9 Sample JavaScript code that runs index.html as an Electron app with a dedicated window instead of a webpage

Finally, the decision to integrate an Amazon Echo module with the Magic Mirror application is due to the design of the Mirror and the limitations in interactions a user may have using this particular interface. As the user will not be able to interface with the application through haptic or physical interaction, the use of a natural language processor to receive input commands is extremely effective in solving the issue of input from the user. This will require integration of the Amazon Alexa Skills Kit. This kit is used in conjunction with the main application to process the user's vocal inputs, which is designed to flow data to the voice processor, which includes speech recognition, machine learning, natural language processing with linguistics, and text to speech for reply. This sends a request to a custom Alexa voice service set up by the developers, which returns a JSON response of the appropriate response. However, the Alexa voice service is configured and trained to receive certain custom commands, such as paging to a specific widget or updating that specific widget, which is extremely useful for specialized applications such as this project. This data flow can be seen in figure 10 and figure 11 below:

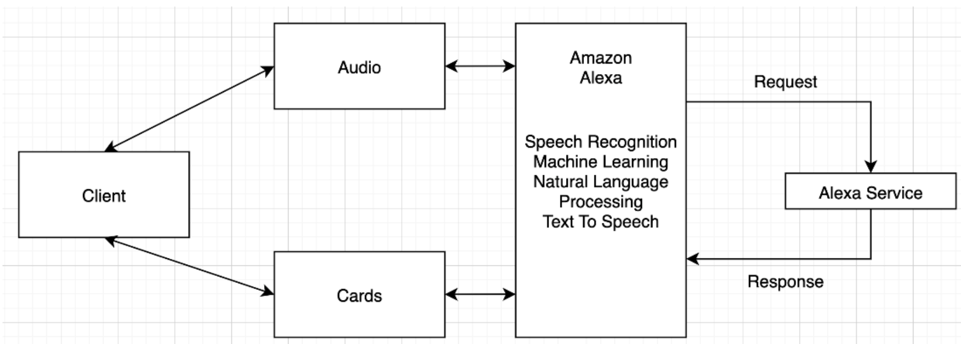


Figure 10 Data Flow diagram of the Amazon Alexa Voice Skill Kit software, showing the request and response of an Alexa Voice JSON

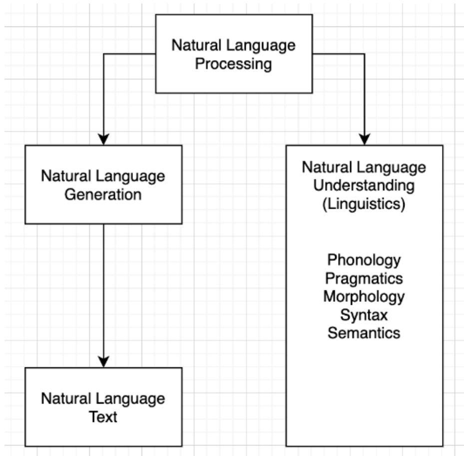


Figure 11 Natural Language Processor breakdown with various functions. Note the natural language generation module that recites certain text for auditory user feedback

Hardware Implementation

The main components for hardware implementation involved in creating the smart mirror system include a computer monitor, a Raspberry PI, a wooden frame, and a reflective window film. With those components, the developed smart mirror would be able to perform its most basic functionalities. To further expand on the basic components, additional components were used, such as the PIR sensor, Amazon Echo Dot, and the RGB LEDs. The full bill of materials for the project can be seen in table 2 below.

Table 2 Bill of Materials for the Project

Item	**Price**
21' Monitor (19.6 x 8.3 x 15.11")	$129.03
Window Film (36" by 15ft)	$50.69
PIR Sensor	$3.44
RGB Programmable LEDs	$13.99
Amazon Echo Dot	Already owned
Raspberry Pi	Already owned
Shadowbox Frame	Already owned
Subtotal	$197.15
HST	$25.63
Total	$222.78

As seen in the table, the main spending for the project was the 21' monitor, reflective window film, the PIR sensor, and the RGB LEDs. The other components, such as the Raspberry Pi, the wooden frame, and the Echo Dot, were already owned by the group members. The final amount was $222.78 spent out of the total $400 budget that was given to the group.

The main initial steps for hardware development involved in assembling the frame for the smart mirror. For the frame of the mirror to be set up, the computer monitor must be placed within a wooden frame with the front opening of the frame covered with a piece of glass. The back of the wood frame would also include an opening allowing the installation of the Raspberry PI unit and creating space for the power supply wires to exit the frame. The general CAD designs for the wooden frame was created in Google Sketchup and can be seen in figure 12 and figure 13 below. The designs seen below is labeled with general dimensions of the wooden frame and the monitor used. Figure 12 shows the front view of the frame showing the wooden frame opening where a glass panel would be added to allow monitor display to show. A reflective film would be added on top of the glass panel to create a one-way mirror effect, making the surface of the glass reflective while still displaying the contents of the monitor. Figure 13 below shows the back view of the mirror, where a Raspberry PI unit can be seen attached to the back of the monitor. In addition, two wooden pieces can be seen; one in the top left corner, and the other in the bottom right corner of the monitor, which are used to secure the monitor into the wooden frame.

While conventional methods required for the wooden frame to be custom made to fit the dimensions of the computer monitor, a group member was able to obtain a premade 20' by 20' shadowbox frame that fits the dimensions of the 21' monitor purchased, allowing the group to skip the process of creating custom fame and the wooden pieces to needed to secure the monitor. The shadowbox frame that was used for came with a clear plastic panel which also substituted the glass panel needed in the original design of the frame. To complete the assembly of the mirror, the group placed the computer monitor within the frame and applied the reflective film on top of the plastic panel. In the back of the frame, openings were left for the power supply cables of the monitor and the Raspberry PI unit to exit.

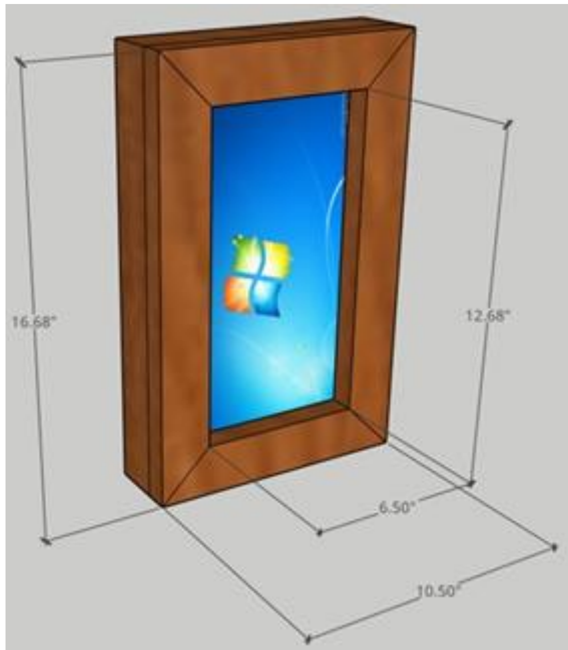


Figure 12 The front view of the frame

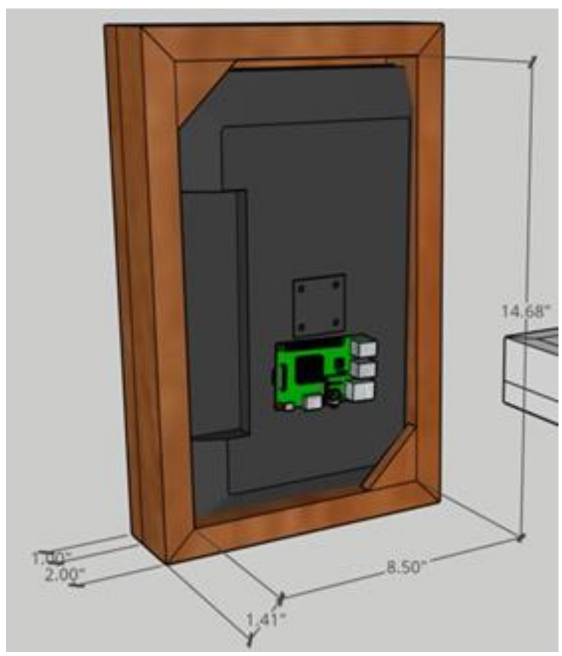


Figure 13 The back view of the frame

One of the drawbacks of using the prebuilt shadowbox frame was that the monitor's display did not take up the entirety of the frame's openings. Although the monitor did indeed fit into the frame securely, approximately half of the wooden frame's opening is left uncovered by the monitor. Therefore, to prevent outside light from entering the frame from the back of the frame, a black cardboard cutout was used to cover the residual space in the back of the frame. The black background within the wooden frame enhanced the reflectivity of the mirror surface.

On the back of the monitor, the Raspberry PI unit was attached, connecting the monitor's HDMI input and the PIR sensor's signal input. Figure 14 below shows the wiring of the PIR sensor on the Raspberry Pi. As seen in the figure, wires were connected to the 3.3V power supply, ground, and signal input on the Pi's unit. The PIR sensor exits from the back of the frame and is placed on the top surface of the frame, facing the front. This placement of the PIR sensor is to enable the sensor to detect movement in front of the smart mirror and trigger the turn off signal if no movement is detected.

The Amazon Echo Dot was used in the hardware setup of the smart mirror to enable a microphone input into the smart mirror system. The Echo Dot is built-in with an Amazon Alexa voice assistant which was used to enable voice commands for the smart mirror interface. The hardware setup for the Echo Dot unit was separate from the wooden frame and only a required wired connection to a nearby power outlet. The Amazon Echo Dot would communicate with the Raspberry PI unit through a private Wi-Fi network.

The RGB LEDs were purchased by the group with plans to attach the LEDs on the perimeter of the wooden frame to enable a lighting effect when the user used different features of the smart mirror system, such as a countdown timer. However, due to time constraints in the development of the interactive software modules for the smart mirror, the group decided not to implement the LEDs as part of the hardware design of the smart mirror.

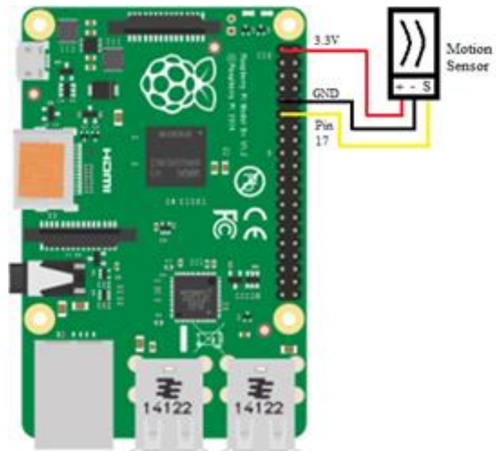


Figure 14 Wiring of the PIR sensor to the Raspberry PI

Testing, Evaluation, Verification, and Validation

The main methods of testing used in the project were technical input testing and user testing. Once the base features of the smart mirror were developed, the technical input testing was performed on the system. To conduct the technical testing of the smart mirror, each functional requirement decided by the group was set up as individual test cases, as seen in table 3 below, where various user inputs were used to test the system outputs. The objective of the test case is then compared with the actual system results, where changes were made to the system if the test case fails.

Table 3 List of Test Cases for Technical Testing

Test Case	**Objective**	**Test Results**
Toggle Mirror on/off with voice command.	The mirror turns on/off when the user says voice command, “Alexa, turn on the mirror” or “Alexa, turn off the mirror.”	Success. The mirror was able to turn on and off correctly, whenever the user issues voice correct voice command.
The time module displays the current time.	The time module accurately displays time in a set time zone.	Success. After the configuration of a specific region, the module

		accurately displays the time for the specified region.
Weather Module displays current weather.	The weather module connects to the OpenWeatherMap API and displays the correct weather for the configured location.	Success. The weather module was able to accurately display the weather of the configured location.
The calendar module displays user scheduled events.	Calendar module connects to the Google Calendar API, allows users to add/delete/update calendar events, and dynamically updates calendar events.	Success. The calendar module was able to connect to the Google Calendar API and dynamically change and display user's calendar events.
Spotify Module plays user-specified music.	The module is able to connect to Spotify API and enable users to pause/play/shuffle/select music on command. The module displays the music that is currently playing.	Success. Spotify module was able to connect to the Spotify API and correctly play the user-specified music and change on user command. The module accurately displays player for music played.
PIR sensor toggles mirror display after detecting movement.	PIR will toggle the monitor's display on/off if the sensor detects movement within 3m of its range.	Failure. During initial testing, the PIR sensor did toggle the monitor on and off accurately when the user stood in front of the mirror.

Mirror user interface switches to different user pages.	When a user uses voice command “Alexa, switch to ____’s page,” the mirror UI will navigate to the user’s page.	Success. Mirror was able to switch to the correct user page when after the user’s specified voice command.

As seen in table 3 above, during initial testing, most of the technical test cases were successful in achieving the test case objective. The PIR sensor during initial testing did not consistently toggle the mirror on or off when it detected user movement. When the user stood within the sensor range, the PIR sensor would sometimes toggle randomly and turn off the mirror display. In other trial runs of the PIR testing, the PIR sensor would seem to turn itself on when there is nothing within the sensor range. After multiple test runs of the PIR sensor, it was found that there was a 30-second delay set on the PIR sensor settings. After reducing the sensor delay to less than 30 seconds, the PIR reacted to the user movement more accurately, with the screen toggle going off shortly after sensor detection.

For the method of user testing for the smart mirror, a survey with specific and open-ended questions was sent to people to help understand their familiarity of the smart mirror system and to gain insight on their interests of the system. The process of user testing helps the group validate whether there is a need for a smart mirror system and if there are modules that users would like to see added onto the system. The survey was given to eight people within the age group of 18-60, with the majority of people in the age group of 18-24. The people surveyed were asked to try to answer the question in the survey to the best of their abilities. The survey questions and answer summaries can be seen in the appendix below.

Analyzing the survey results reveal that there is great interest in a smart mirror system in a household setting. Survey results show that 62.5% of people use at least 2 mirrors personally, and 50% of people spend 1-5 minutes in front of the mirror in the morning. When asked what the most stressful activities in the morning are, 62.5% of people answered along the lines of commuting and catching the bus. When asked what features the users would like to see in a smart mirror, 25% answered a module on bus schedules or weather. When asked if users would be comfortable with an infrared sensor in front of the mirror to sense for user movement, all users answered that they are comfortable. When asked how likely they would want a smart mirror in their home, given a scale from 1-5, with 1 being very unlikely and 5 being very likely, the majority of people answered above a 3 and with 50% of people answering a 5 for very likely.

Related Work

The group surveyed other related work on related to interactive mirrors and the smart mirror system. In 2006, researchers in Tokyo, Japan, published a paper on “information-accessing furniture,” one of which was an interactive user mirror called the “Miragraphy” [4]. The system design was a mirror that had an embedded module behind it containing a small computer and projector that superimposed the projected image through a diffusion film outward [4]. The Miragraphy was able to display weather and news headlines in its display. In October 2016, a project similar to the group’s smart mirror system was created in Edmonton Alberta under the name “Smart Reflect” [7]. The Smart Reflect system also uses a similar design to our group with the use of a Raspberry PI as the core of the system and a mirror with an embedded monitor as its source of display [7]. The user interface was a lightweight process and but lacked in user customizability and voice control [7]. In November 2016, a smart mirror project was created at Ulm University, Germany, called the “FitMirror” [8]. The FitMirror system created with a display behind a mirror frame, in addition to adding components such as the Wii Balance board and the Microsoft Kinect [8]. The main purpose of the FitMirror was not for entertainment but for tracking the user’s health [8].

In comparison to the other related research on the smart mirrors, the group’s smart mirror system was found to be simpler, more lightweight, and user-customizable. As tested to be compatible with the Raspberry PI, the smart mirror system demonstrated that it is computationally lightweight. The group’s smart mirror system integrates voice control through the use of the Amazon Echo Dot, enabling the users to control the smart mirror hands-free. Moreover, the group’s smart mirror system is set apart from the other existing work through the ability for multiple users to customize their own display and further extend module functionality.

Stakeholder Needs

As this project was able to simulate the full lifecycle of the engineering product development, various stakeholders are required to be considered for each stage for the project to meet and exceed all requirements. This will involve several stages and considerations from the initial planning stages outlined in ELEC 390 Electrical and Computer Engineering Design, where the engineering and development cycles were initiated. Several stakeholders typical of a life cycle stage would include acquirer, panel or potential users, marketing division, research, and development department, standardization body, supplier, verification and validation team, production system, certification or regulatory authorities, design engineers, and integration team as outlined in table 4 below:

Table 4 Stakeholder indentification based on life cycle stage

Life Cycle Stage	Example of Related Stakeholders
Engineering	Acquirer, a panel of potential users, marketing division, research, and development department, standardization body, suppliers, verification and validation team, production system, regulator/certification authorities, etc.
Development	Acquirer, suppliers (technical domains for components realization), design engineers, integration team, etc.
Transfer for Production or for Use	Quality control, production system, operators, etc.
Logistics and Maintenance	Supply chain, support services, trainers, etc.
Operation	Normal users, unexpected users, etc.

Disposal	Operators, certifying bodies, etc.

However, the later life cycle stages, such as logistics and maintenance, operations, and disposal, are less applicable, and a greater stakeholder weight is placed on the engineering and development life cycle phases. This will include the consideration of the Faculty of Engineering as the validation and verification team, supplier, and development department, the engineering design and integration teams, and ultimately the end-user as the key stakeholders.

Faculty of Engineering: As the governing body and overseer of the project, working closely with the supervisor and teaching requires the core development team to consider particular constraints and guidelines relating to the safety, privacy, codes, standards, manufacturability, ethics, and costs of the project, with significant emphasis on the standards and codes. This stakeholder will require the development team to adhere to certain protocols and standards when developing the core application, such as the thoroughness of the project reports, clarifications on intentions, and consistent updates on the progress of development. This was achieved as expectations were met and exceeded throughout the development period, with all deadlines being met and project reports being successfully delivered. Moreover, as the supplier of several physical parts required for the project to function, especially the IPS monitor, the design team would have to consider budget restraints and the costs of acquiring the necessary technical components. This guideline and stakeholder need to have the budget under a total sum of $400.00 was met and exceeded, as the total costs were $222.78.

The Development Team, as another stakeholder, will require the elicitation of several stakeholder needs and requirements, as these guidelines also contribute to shaping the development of the overall project. As the main design engineers and integration team, several key design choices were made with the considerations of development, such as the most effective way to implement a solution for the application. This was executed and determined through the consideration of the following design criteria:

- Clear and effective design: By utilizing standard web design technologies such as HTML, CSS, Node.JS, and Electron, this provides a flexible design with prebuilt standards for designing the user interface, as opposed to creating and specifying items in a grid system similar to a Python or C++ graphical user interface (GUI) development design. This allows the team to scale back reasonably on development and coding timeframes and allows the team to focus on testing and integration.
- Reasonably simple and standard development: Moreover, utilizing web design technologies in conjunction with Electron allows for a simple translation from the constraints of a web page to a fully standalone application. This was chosen as a simpler, familiar, and standardized alternative to creating a barebones GUI utilizing other technologies, allowing for easier debugging as well as compartmentalizing and mobilizing the simple application.
- Achievable functionality requirement was achieved by the integration team during the engineering and development phase, which is critical in allowing the project to function up to the standards outlined. With the integration of various widgets along with the Echo voice functionality, the development team was able to achieve this stakeholder requirement through the rigorous development process through various testing procedures as well as coding sprints.

Finally, the most important stakeholder is the end-user, which the guidelines and requirements will influence the design of the project to a higher caliber. This user stakeholder process was conducted through a user-centered design (UCD) iterative process. This is an optimistic approach that attempts to invent new solutions from the perspective of human users and provides answers to individual needs. By understanding the end-user and their requirements, the team is able to tailor specific design functionalities to achieve these requirements, which involve requirement discovery, requirement classification, requirement prioritization, and requirement specification, which is done through an iterative and agile process, which involves constant feedback and improvement. By following the steps below, the team is able to identify and fulfill the end-user stakeholder needs:

- Understanding the context of use: By creating specific use cases and using mockup demos, a specific target environment is identified by the possible brainstorming issues a user may have without the assistance or use of the product. This allows for certain constraints to be visualized and illustrated from the perspective of the user, as the Magic Mirror can be applicable in certain locations. This may cause constraints on the end-user, such as limited input functionality, use in a wet environment, or the delayed or misjudged reaction of a user due to lighting, sleepiness, or time of day. Ultimately by identifying the context of use, the team is able to identify slips or mistakes of user error in handling and interacting with the product in a specific context.
- Specify user requirements through elicitation: Moreover, once the context has been identified and understood, the user requirements are further clarified through elicitation, which will involve conversing with the user or illustrating the required design implementations from the perspective of a user. This put the development team as the end-user, with additional consultation from other users in trial tests, such as housemates, classmates, or peers. This allowed the team to specify and prioritize these formal requirements, such as the inability to interact physically with the Mirror, the inability to use a mobile device in the washroom with occupied appendages, and the lack of information available to the user when completing other unrelated tasks.
- Design solutions: Once these user requirements have been prioritized and specified, the team will consider various solutions to implement, such as the inclusion of an internet of things (IoT) device, Amazon Echo input with the user through natural language processing. By testing out various designs for the user interface such as python GUI, or web development technologies, an effective and clean design was achieved to relay information to the user in an optimal way. In addition, this solution requires a functional back end stack that will load the relayed information to the user, while integrating with the Amazon Echo module. Overall, this design was considered through the design process, which is built upon the various user requirements that have been elicited from the previous steps of UCD.

- Evaluate against requirement: This final step involves the feedback from potential end users, alongside the supervisor and evaluator of the project that would place themselves from the end user perspective. By receiving feedback on the functional measures implemented that address the user requirements and guidelines, the team was able to narrow down the scope of identified issues, such as PIR timeout, or voice accuracy of the input commands. This final step of the user centric design principle is extremely effective in achieving the stakeholder needs of the end user, with direct feedback on the features.

Finally, several other miscellaneous stakeholder needs have been considered with the overall design of the product; however, these stakeholders are mostly out of scope and potentially irrelevant. Regardless, the team still considers the possibility of future implementations that will include these, which are listed below:

- Safety is considered in the overall design of the mirror. However, as a static application based on software, physical safety is limited to the structural integrity of the physical mirror and ensuring it has been properly mounted to avoid falling on to the end-user, which fails the user stakeholder requirement of not being killed or physically maimed by the product.
- The privacy of the product is also considered, especially physical. As this IoT device concerns the use of an internet-connected peripheral in a vulnerable context such as the washroom, the removal of a camera or visual input is highly recommended and implemented in the final solution. This prevents any software attacks from accessing visual data on unknowing users. By establishing this privacy stakeholder, the product was able to exceed this requirement.
- Manufacturability is also considered alongside cost in the event of scaling to larger production. By evaluating the overall cost and capabilities required to effectively produce a large quantity of the product, the manufacturing methods and cost are accounted for to determine the margin of profit,

as well as any special skills required. However, as the design is simple, no special tools or skills are required.

- Lastly, ethics as a stakeholder is considered briefly, as we have not identified any ethical issues with visualizing weather data in the washroom.

Compliance with specifications

As stated in the blueprint, the project is mainly a software integration project. Therefore, the hardware components are mediums in which our software operates on. No hardware will be created from scratch. Instead, they will be utilized as a part of the smart mirror system.

The only hardware that was tested for the smart mirror system was the PIR sensor. We wanted to test if the PIR sensor can detect movement within a 3m distance and if it can detect no movement for greater than 30 seconds. Table 5 shows the results.

Table 5 Hardware specification of the PIR sensor

	Specification	Target Value	Tolerance	Achieved Value
1	Detect movement within a 3m distance	3m	30cm	~3m
2	Detect no movement for greater than 30 seconds	30 seconds	10 seconds	~32 seconds

Throughout the project, there have been many changes to the functional requirement. Some were added because we did not initially anticipate having it until later stages, and one was added because it was meant to be a stretch goal, but we managed to achieve it in our given timeframe.

Table 6 Software Specifications Table

1	**Functional requirements**	**Specification met?**
1.1	The mirror will be able to connect to OpenWeatherMap API service and display daily weather for Kingston, Ontario.	Yes
1.2	The mirror will be able to connect to OpenWeatherMap API service location, such as city, province, and country	Yes
1.3	The mirror will be able to accurately display current time and date (Eastern Standard Time)	Yes
1.4	The mirror will be able to connect to Google Calendar's API service, and users will be able to add/delete/update calendar events that will display dynamically on the mirror	Yes
1.5	The mirror will be able to connect to Spotify's API service, and users are able to pause/play/shuffle/choose songs through voice command. The display should be able to match said command.	Yes
1.6	The mirror's display will be activated only if the PIR sensor detects movement within 3m of its prereferral	Yes
1.7	When no movement is detected for 30 seconds the mirror's display will turn off	Yes
1.8	The mirror will be able to navigate to different user pages through voice command and a given name	Yes
2	**Interface requirements**	
	The interface from the device to the user:	Yes

	Display information using icons and text, using Raleway or Quicksand (font types). Other icons can be created in photoshop with a flat monochromatic design in white Respond to the user with information on: • Music • Weather • Time • Calendar	
2.2	The interface from the user to the device: Sync between the user device and display to show music, calendar, weather, time Request device with information on: • Music • Weather • Time • Calendar	Yes
3	**Performance requirements**	
3.1	Able to pull data from an API using HTTP get requests over Wi-Fi	Yes
3.2	Updates the display portlets within a reasonable time (once per day)	Yes
3.3	Updates the clock once per minute	Yes
3.4	Recognizes and processes voice commands within the Amazon's Alexa API specifications	Yes

Conclusions and recommendations

Further Research and Development

Currently, our project is in the proof of concept stage, with a lot of potential to expand further than the current scope. Many improvements we wanted were previously stated as the stretch goals in the system design section.

One of our improvement was to add RGB LED light strips surrounding the frame of the mirror. We understand that it is very easy for people to lose track of time when showering or getting ready in the morning. Therefore, by adding LED lights that progressively change color over time, we hope it can serve as a visual cue indicator of how long the person has been in the washroom.

Another improvement would be to add more functions or APIs to the current pool. Right now, the user is very limited in terms of what functions they have. More research and development for this project can allow functions or services that users typically see on their smartphones, such as cellular services, video playback, stocks, traffic reports, or email.

By providing some more of these features to the users, the mirror can be more suitable for a wider range of audiences.

Main Technical Lessons

The main technical lesson to be learned from this project is that software system integration is very messy. And having knowledge and background is specific software domains is very useful to produce an effective working product. For example, in our project, we used two environments the Amazon Skills Kit developer tool and the Electron software framework (which is open source). Having communication between these 2 separate platforms is an interesting challenge because any updates or changes from one of the platforms can cause compatibility issues since Amazon and Electron operate as separate entities. Therefore, reading documentation and understanding which versions are compatible with each other is crucial to a working system. However, as there is a growing market for voice-controlled smart home devices and projects, software engineers of Amazon will continue to develop wrappers or frameworks that have compatibility between different systems.

Future Potential for the Project

From our research on the internet to the overwhelmingly positive feedback we got from students during the capstone showcase, we believe that there is a market of people who would like to have a smart mirror in their homes. The only problem is the product is not deemed as an essential good but more so a luxury good to improve a person's quality of life. A good comparison to a product like this would be smartwatches.

However, with the increasing market of smart home assistants and IoT platforms, it is not unlikely a smart mirror type system will reach the market in the distant future. An article from MarketLand estimates that as of the 2nd quarter of 2019, there are approximately 76 million units of smart speakers are sold in the U.S, with 70% of the market share dominated by Amazon alone. (Shown in Figure 15) [9] The objective for these corporations is that they first need to integrate voice-assistant devices into people's lives, and once that is integrated into, it is easier for them to push a product that can work with their smart home device. Therefore, the potential for a product like this is very strong.

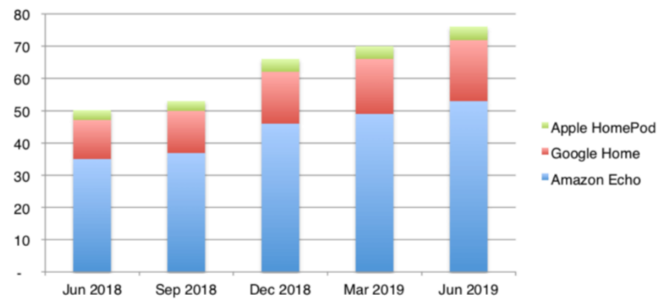


Figure 15 U.S smart speaker market share (millions)

Pricing

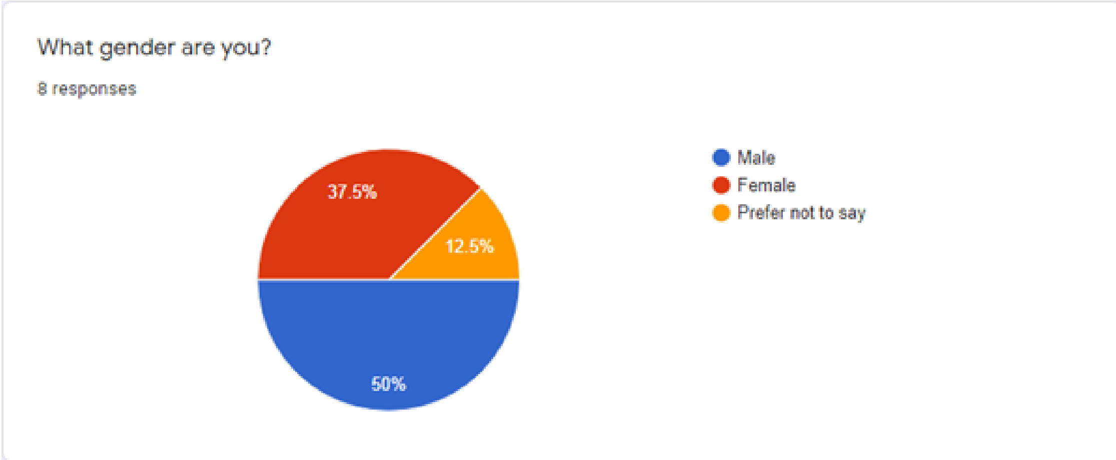
Overall, with the correct cost analysis, business model, and go to market strategy, this project would have a potential in the market as a commercialized product for general consumer use. However, this will consider several key factors first, such as manufacturing constraints and costs. As the construction of the solution implemented consists of a basic frame utilizing a one-way mirror over an LCD monitor, this product does not require any proprietary or specialized manufacturing methods, which in turn will lower manufacturing tooling and training costs. This lean development principle applied throughout the product keeps the product lightweight and modular, which is a significant market advantage. As outlined in table 2, the overall estimated cost will amount to $222.78 CAD, with the main components listed as Raspberry PI, LCD Screen, and one-way mirror, which are easily manufactured and sourced from various suppliers. However, considering usual consumer-friendly prices while also factoring a minor percentage of revenue towards research and development, a suggested retail price of $299.99 CAD would be suggested. However, without additional factors considered, such as marketing, deployment, various scale points, and other business costs, an estimated cost for a manufactured version can vary.

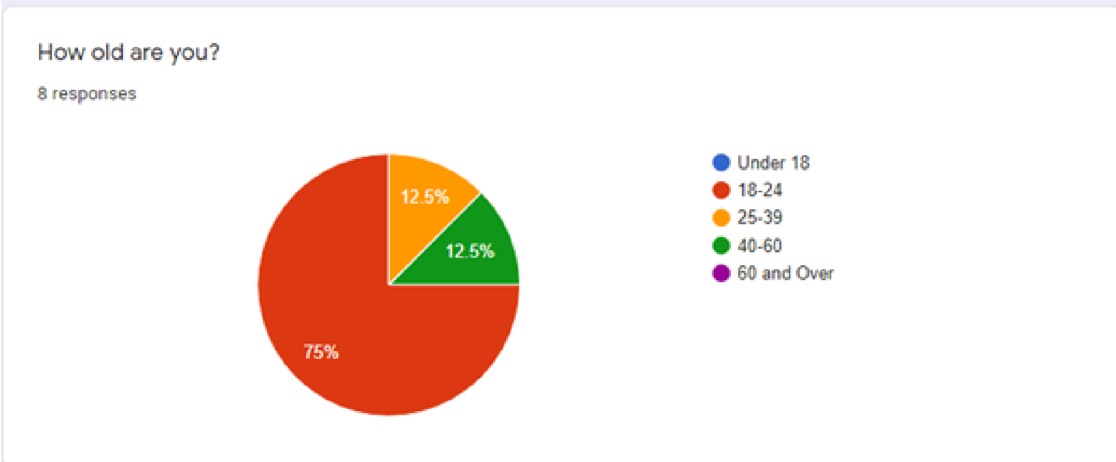
Group Effort table

Table 7 Group Effort Table

Name	**Overall effort expended (%)**
Adams	100
Ian	100
Kingsley	100

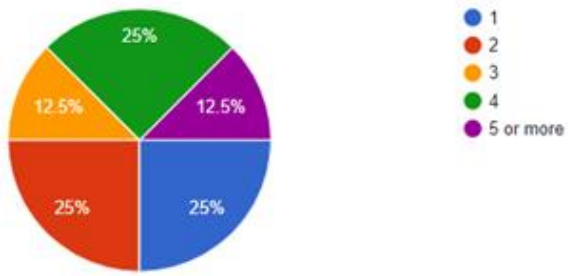
Appendix - Survey Results





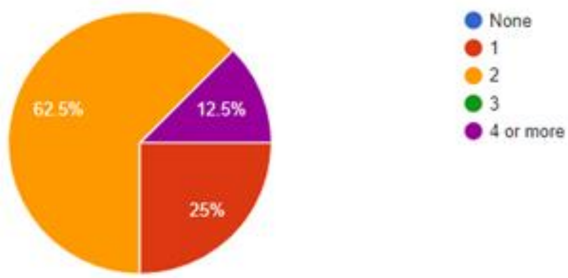
How many people are in your household?

8 responses



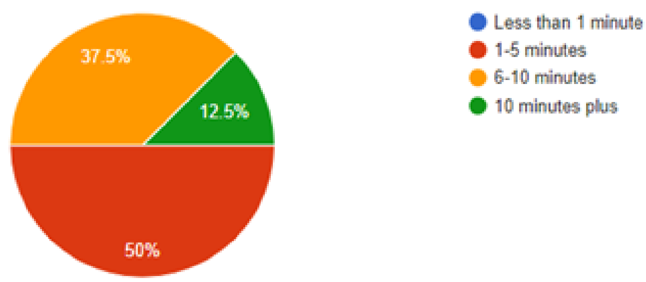
How many mirrors exist in your household that you personally use?

8 responses



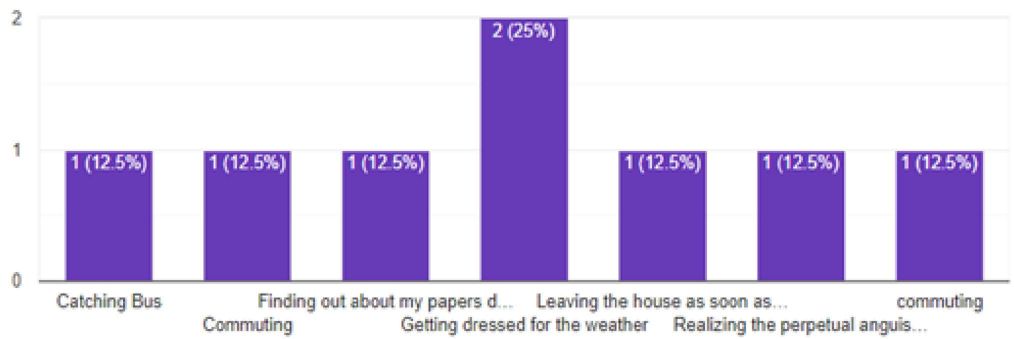
How much time do you spend in front of a mirror in the morning?

8 responses



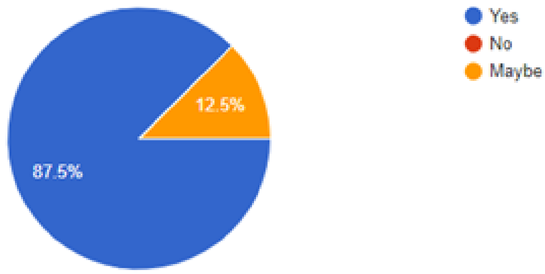
What are your most stressful activities in the morning?

8 responses



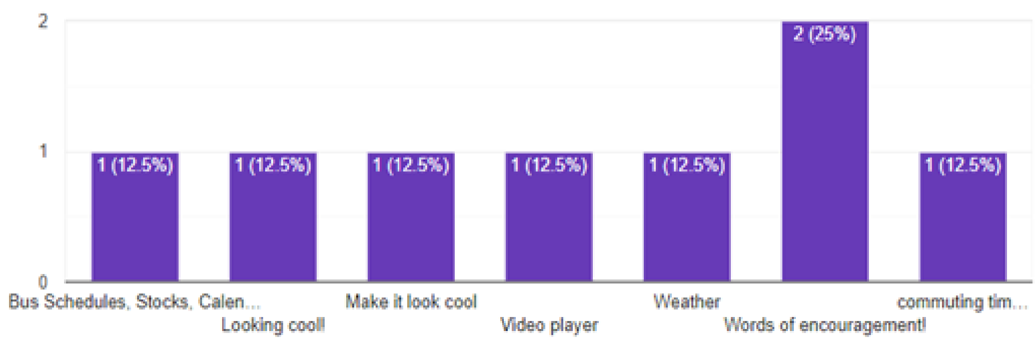
Would you like the content to be different depending on where the smart mirror is located? (e.g. news feed in washroom mirror, reminders in hallway mirror)

8 responses



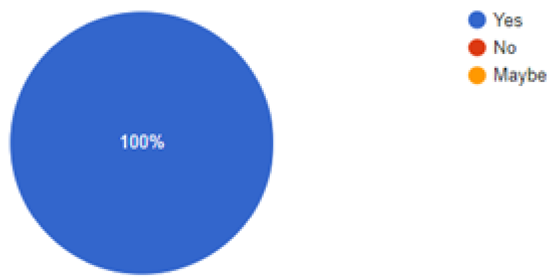
What features would you like to have in your smart mirror?

8 responses



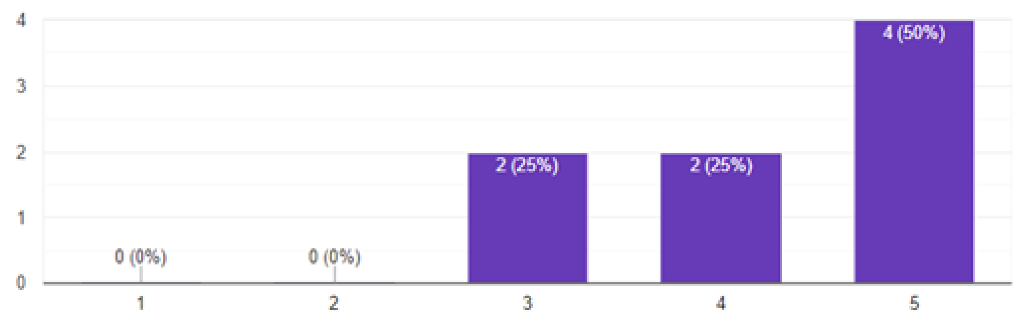
Would you be comfortable with an IR (infrared) sensor in your mirror which would be used for sensing when a person is near?

8 responses



How likely is it that you would want a smart mirror in your home?

8 responses



References

[1] From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence,” *ProtoView*, vol. 1, no. 44. Ringgold Inc, Beaverton, 12-Nov-2014.

[2] M. Chen, J. Wan, and F. Li, “Machine-to-machine communications: architectures, standards and applications.(Report),” KSII Transactions on Internet and Information Systems, vol. 6, no. 2, pp. 480–497, Feb. 2012.

[3] “IDC Forecasts Worldwide Spending on the Internet of Things to Reach $745 Billion in 2019, Led by the Manufacturing, Consumer, Transportation, and Utilities Sectors,” IDC. [Online]. Available: https://www.idc.com/getdoc.jsp?containerId=prUS44596319. [Accessed: 06-Apr-2020].

[4] Hiroko Sukeda, Youichi Horry, Yukinobu Maruyama, and Takeshi Hoshino, “Information-accessing furniture to make our everyday lives more comfortable,” IEEE Transactions on Consumer Electronics, vol. 52, no. 1, pp. 173–178, Feb. 2006.

[5] Dave Leggett, “JAPAN: Nissan develops ‘smart rearview mirror,’” Feb. 2014.

[6] K. Jin, X. Deng, Z. Huang, and S. Chen, “Design of the Smart Mirror Based on Raspberry PI,” in 2018 2nd IEEE Advanced Information Management,Communicates,Electronic and Automation Control Conference (IMCEC), 2018, pp. 1919–1923.

[7] D. Besserer, J. Bäurle, A. Nikic, F. Honold, F. Schüssel, and M. Weber, “Fitmirror: a smart mirror for positive affect in everyday user morning routines,” in Proceedings of the Workshop on multimodal analyses enabling artificial agents in human-machine interaction, 2016, pp. 48–55.

[8] D. Gold, D. Sollinger, and Indratmo, “SmartReflect: A modular smart mirror application platform,” in 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2016, pp. 1–7.

[9] G. Sterling, "Alexa devices maintain 70% market share in U.S. according to survey," Marketing Land, 09 08 2019. [Online]. Available: https://marketingland.com/alexa-devices-maintain-70-market-share-in-u-s-according-to-survey-265180. [Accessed 04 05 2020].