CSC 4350 Software Engineering

Magic Mirror

Fall 2017

Final Deliverable

**Group Name:** CHHAP

**Group Members:** Chris Kazenske, Aqsa Sohail, Hena Shah, Parita Malbari, Hafsah Uddin

November 19, 2017

**Table of Contents**

1. Introduction                                                                  3
2. Problem Statement  4
3. RTM  5
4. Used Cases  6
5. Category Interaction Diagram  9
6. Test Cases  14
7. Rationale  15
8. FPCA  16
9. Gantt Chart  18
10. WSD  19
11. Dictionary  21
12. Resumes  22
13. A title page consisting of the class, semester, Project title, group name, group members, date.
14. Table of contents
15. Each of the above documents that you have revised.
    * Introduction - consisting of the topic description
    * Requirements Elicitation & RTM with "Shall" statements
    * System Analysis & Design with use cases, Sequence Diagrams & Category Interaction Diagram.
    * Object Design
    * Test cases (Document)
    * Rationale for entire project design and implementation
16. Function Point Cost Analysis & COCOMO with comparisons and conclusions.
17. Include the prototype - both the printed slides and on disk
18. Project legacy
19. Updated (Final) WSD showing the group name, members and their assigned roles.
20. Updated Gantt Chart (Final) showing allocation of all tasks
21. A dictionary explaining terms and computer jargons
22. Group members resume
23. User Guide.
24. A print out of all source code

**Problem Statement**

**Magic Mirror**

The Magic Mirror shall run behind a 2-way mirror to create the feel of a Smart Mirror. It shall run on a Raspberry Pi, a very lightweight computer that runs the Raspbian operating . The code shall be written in Java, allowing it to be run on different operating systems. The Magic Mirror shall also use a gesture sensor to make it more user friendly. The gesture sensor shall recognize many gestures performed by a user in order to allow the user to customize appearance, change screens, and control many other aspects of the display. Examples of different gestures are: swipe up, swipe down, swipe right, swipe left, clockwise circles, and counterclockwise circles.

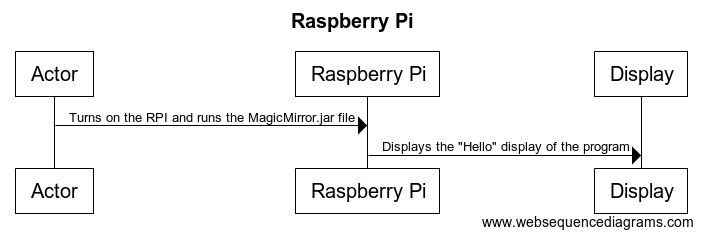
The mirror shall display many features including a clock, up to date weather, news, stocks, quotes,  and many others. The data shall come from various online API’s, which will also update the features every 5 minutes. The clock feature shall display in standard digital based on the user’s preference. The user shall also be able to change the color of the clock to his or her preference.  The preferences shall be set based on a “Preferences” tab which shall be accessible through the use of gestures. The program shall automatically locate the location of the mirror when the program runs based on IP address obtained from the network, allowing the clock to display the correct time based on the time zone.  The weather display shall display the current temperature, forecasted temperature, sunrise and sunset times. It shall also automatically display the weather based on current location, which is determined by the program at runtime. It shall also show weather conditions, like sunny, raining, cloudy, snowing, etc. The news feature shall display hot headlines from various news sources. The stocks page shall display various stock info like top daily and weekly stocks. The calendar page shall display the date at the top left.  The quotes page will display a new quote every time you swipe to it from another page. The user shall be able to choose between a daily, weekly, and monthly calendar display. Each main display (news, weather, stocks, and quotes) shall be changed using gestures. The user shall swipe left or right in front of the sensor in order to switch between these different displays. The user shall be able to use this on a day to day basis for their daily routine.

This software shall provide easy accessibility to all the apps such as weather, clock, date, news, stocks, and quotes that you have on your phone into one product, all without having to even unlock your phone. It shall allow you to view all of this information while getting ready so the user can multitask. A lot of people tend to not check the weather before they get ready so it shall allow the user to know the weather without getting fully ready and then needing to go change outfits because the user forgot to check the weather. It shall eliminate the user from wasting time especially when they are in a hurry.

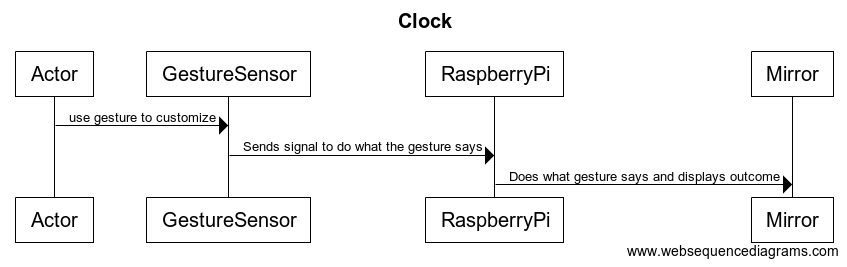
|  |  |  |  |
| --- | --- | --- | --- |
| Entry # | Paragraph # | Requirements Traceability Matrix (RTM) | Type |
| 1. | 1.1 | The Magic Mirror shall run behind a 2-way mirror to create the feel of a Smart Mirror. | HW |
| 2. | 1.2 | It shall run on a Raspberry Pi,a very lightweight computer that runs the Raspbian operating system. | SW |
| 3. | 1.3 | The code shall be written in Java, allowing it to be run on different operating systems. | HW |
| 4. | 1.4 | The gesture sensor shall recognize many gestures performed by a user in order to allow the user to customize appearance, change screens, and control other aspects of the display. | SW, HW |
| 5. | 2.1 | The mirror shall display many features including a clock and date, up to date weather, news, stocks, quotes,  and many others. | SW |
| 6. | 2.2 | The data shall come from various online API’s,which will also update the features every 5 minutes. | SWC |
| 7. | 2.3 | The clock feature shall display either standard or military time and analog or digital based on the user’s preference. | SW, NTH |
| 8. | 2.4 | The program shall automatically locate the location of the mirror when the program runs based on IP address obtained from the network, allowing the clock to display the correct time based on the time zone. | SW, NTH |
| 9. | 2.5 | The weather display shall display the current temperature, forecasted temperature, sunrise and sunset times. | SW |
| 10. | 2.6 | It shall also automatically display the weather based on current location, which is determined by the program at runtime. | SW |
| 11. | 2.7 | It shall also show weather conditions, like sunny, raining, cloudy, snowing, etc. | SWC |
| 12. | 2.8 | The news feature shall display hot headlines from various news sources, such as CNN, ESPN, and Technology. | SW, NTH |
| 13. | 2.9 | The stocks page shall display various stock info like top daily and weekly stocks. | SW |
| 14. | 2.10 | The quotes page will display a new quote every time you swipe to it from another page. | SW |
| 15. | 2.11 | Each main display (news, weather, stocks, and quotes) shall be changed using gestures. | SW |
| 16. | 2.12 | The user shall swipe left or right in front of the sensor in order to switch between these different displays. | SW |
| 17. | 3.1 | This software shall provide easy accessibility to all the apps such as weather, clock, calendar, news, and stocks that you have on your phone into one product, all without having to even unlock your phone. | SW, NTH |
| 18. | 3.2 | It shall allow you to view all of this information while getting ready so the user can multitask. | SW, NTH |
| 19. | 3.4 | It shall eliminate the user from wasting time especially when they are in a hurry. | NTH |

**Use Cases and Interaction Diagrams**

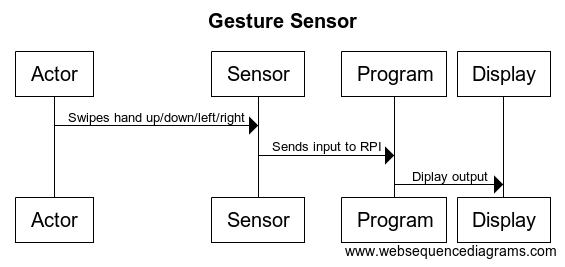
|  |  |
| --- | --- |
| Use Case ID | USC\_001 |
| Use Case Name | USC\_RaspberryPi |
| Participants | User, System |
| Entry Condition | The system must be turned on. |
| Flow of Events | **Action**   1. The user must run the Magic Mirror.jar file on the Raspberry Pi   **System**   1. The program will run without any initial user interaction and display the main display screen (as seen in the “Hello” prototype). 2. The user can begin interacting with gesture swipes. |
| Exit Condition | The Raspberry Pi loses power or the program is exited. |
| Quality Restraints | The program must run within 5 seconds. |

****

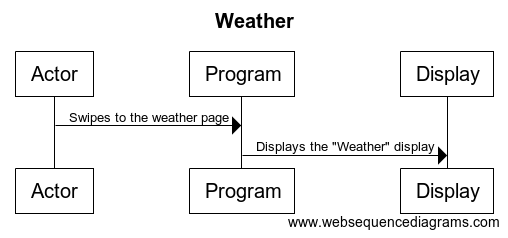
|  |  |
| --- | --- |
| Use Case ID | USC\_002 |
| Use Case Name | USC\_Clock |
| Participants | User, System |
| Entry Condition | The system must be turned on. |
| Flow of Events | **Actor**   1. User stands in front of the mirror   **System**   1. System turns on and displays screen 2. The system will automatically locate the location of the mirror based on IP address from the network. 3. The system will display the correct time based on the location. 4. The clock is displayed on the top left in digital mode 5. The time is displayed on every page while the other widgets have their own pages. |
| Exit Condition | User moves away from the mirror to turn off the system. |
| Quality Restraints | User must stand in front of the mirror to turn it on and move away to turn it off. |

****

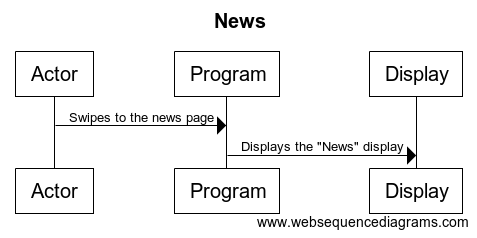
|  |  |
| --- | --- |
| Use Case ID | USC\_003 |
| Use Case Name | USC\_GestureSensor |
| Participants | User, System |
| Entry Condition | The User must stand in front of the mirror |
| Flow of Events | **Action**   1. The user gestures their hands up and down or left and right to get to the desired page. 2. The user can gesture their finger in a circular motion to turn the volume up and down.   **System**   1. The system recognizes the gestures and displays the direction the screen is moving in. |
| Exit Condition | They must move away from the mirror to exit the system. |
| Quality Restraints | The sensor must swipe within a second. |

****

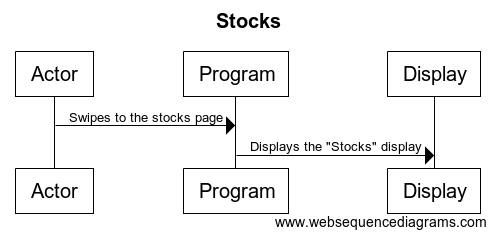
|  |  |
| --- | --- |
| Use Case ID | USC\_004 |
| Use Case Name | USC\_Weather |
| Participants | User, System |
| Entry Condition | The system must be turned on. |
| Flow of Events | **Actor:**  1.     When the user turns on the Magic Mirror, the weather will be displayed in the top(left/right) corner, and the actor can use hand gestures to swipe to the next screen.  **System:**  1.     On the basic home screen, certain features such as time, date, and weather will be displayed.  2.     The mirror will pick up hand gestures through means of a sensor gesture and the mirror will swipe to a different feature. |
| Exit Condition | Magic mirror is turned off. |
| Quality Restraints | Constant HDMI connection to computer (Raspberry Pie).  Constant power supply. |

****

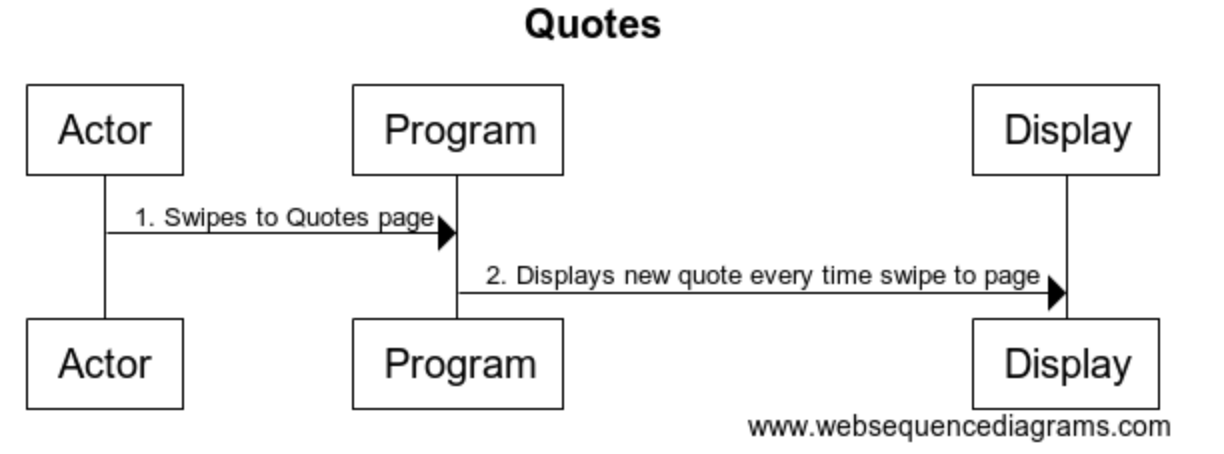
|  |  |
| --- | --- |
| Use Case ID | USC\_005 |
| Use Case Name | USC\_News |
| Participants | User, System |
| Entry Condition | The system must be turned on. |
| Flow of Events | **Actor:**   1. The news will be displayed on the screen and the top 5 news headlines will be available for the user.   **System:**   1. The system will get the news through various APIs and the top 5 news headlines will be displayed on page. |
| Exit Condition | The news will be displayed. There will be various news sources that are used and the user will be able to switch from sources using hand gestures. |
| Quality Restraints | The news headlines should be displayed from the various sources. The code and the software should run immediately, no waiting |

****

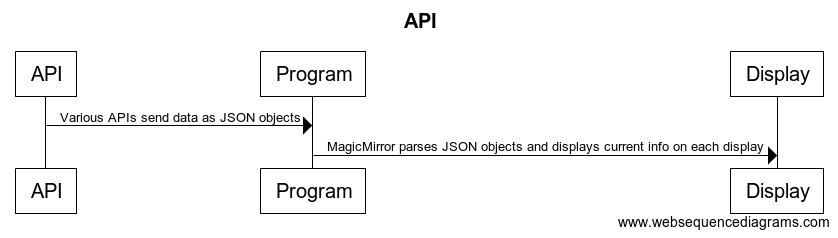
|  |  |
| --- | --- |
| Use Case ID | USC\_006 |
| Use Case Name | USC\_Stocks |
| Participants | User, System |
| Entry Condition | The system must be turned on. |
| Flow of Events | **Actor:**   1. The actor will approach the mirror so the mirror “turns on” through the means of sensor detection. 2. The stocks will be displayed on the screen and the top 5 stocks will be available for the user.   **System:**   1. The mirror will “turn on” and the basic home screen will be displayed. 2. The system will get the stocks through the API’s and the top 5 stocks will be displayed. 3. The system will allow the user to view open, close, up, and down. |
| Exit Condition | The stocks will be displayed. There will be various stock sources that are used and the user will be able to view on panel. |
| Quality Restraints | The stock prices should be displayed from the various sources without the user having to wait more than 15 seconds. |

****

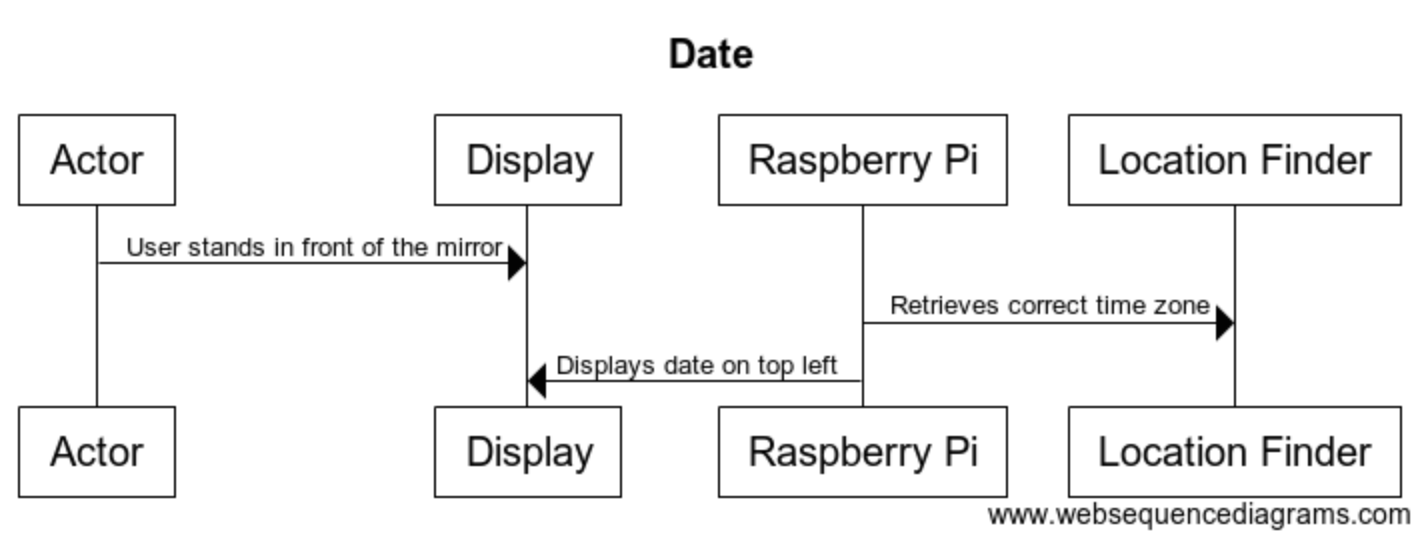
|  |  |
| --- | --- |
| Use Case ID | USC\_007 |
| Use Case Name | USC\_Quotes |
| Participants | User, System |
| Entry Condition | The system must be turned on. |
| Flow of Events | **Actor:**  1.     When the user turns on the Magic Mirror, the user can swipe either left or right to see the quotes panel.  2.     New quote will appear every time the user swipe to the Quotes panel.  **System:**  2.     The system will recognize the gesture to the Quotes and page and display a new Quote. |
| Exit Condition | Actor swipes left/right. |
| Quality Restraints | Constant HDMI connection to computer (Raspberry Pie).  Constant power supply. |

****

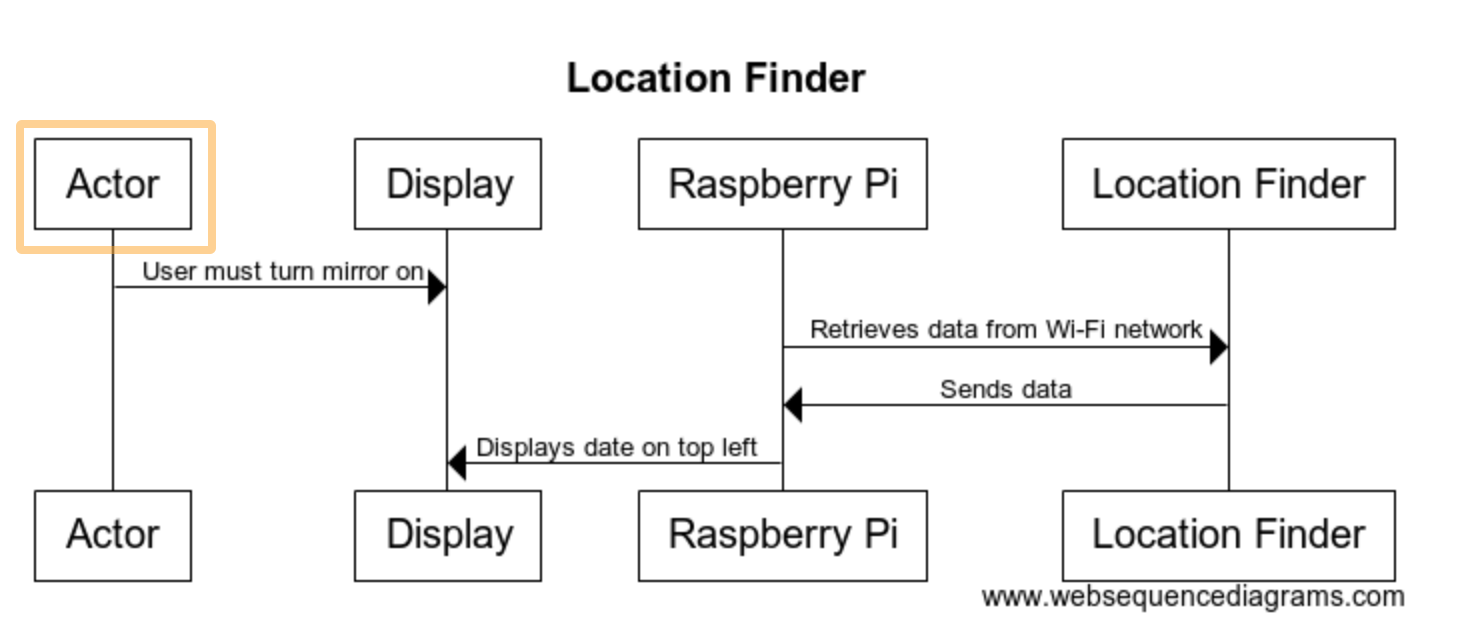
|  |  |
| --- | --- |
| Use Case ID | USC\_008 |
| Use Case Name | USC\_API’s |
| Participants | User, System |
| Entry Condition | The system must be turned on. |
| Flow of Events | 1. When user the stands in front of the mirror, the system is turned on. 2. Then the system connects to API’s to get the informations to user. 3. When it connects the user can look at the all their apps, and look at the most updated things. 4. The API’s update every 5 minutes. |
| Exit Condition | The user moves away from the mirror, and the system will turn off. |
| Quality Restraints | The system must be connected to wifi. |

****

|  |  |
| --- | --- |
| Use Case ID | USC\_009 |
| Use Case Name | USC\_Date |
| Participants | User, System |
| Entry Condition | The system must be turned on. |
| Flow of Events | **Action**   1. User stands in front of the mirror   **System**   1. The date is displayed on the top left. 2. The date is displayed on every page while the other widgets have their own pages. |
| Exit Condition | User moves away from the mirror to turn off the system. |
| Quality Restraints | User must stand in front of the mirror to turn it on and move away to turn it off. Hand gestures must also be used to change settings on the clock. |

****

|  |  |
| --- | --- |
| Use Case ID | USC\_010 |
| Use Case Name | USC\_Location finder |
| Participants | User, System |
| Entry Condition | The system must be turned on. |
| Flow of Events | **Action**   1. When user the stands in front of the mirror to use the mirror.   **System**   1. The system will automatically locate the location of the mirror based on IP address from the Wi-Fi network. 2. The system will display the correct time based on the location. |
| Exit Condition | User moves away from the mirror to turn off the system. |
| Quality Restraints | The system must be connected to wifi. |

****

**Test Cases**

Chris Kazenske

Type: Unit Testing and Integration Testing (Big Bang testing)

Tested: 2 times

|  |  |
| --- | --- |
| Test-Case Identifier | Quote |
| Test-case Location | Quote.java |
| Feature to be tested | If a new quote appears every time it is swiped to. |
| Feature Pass/Fail Criteria | Passes if new quote appears. |
| Means of control | Through a set of arrays that contain the quotes that prints them out. |
| Data | Gets the data from arrays. |
| Test Procedure | The test is started when the user stands swipes to the Quotes page |
| Special requirements | Swipe to page for a new quote. |

Chris Kazenske

Type: Unit Testing and Integration Testing (Bottom - Up testing)

Tested: 2 times

|  |  |
| --- | --- |
| Test-Case Identifier | HoverSensor |
| Test-case Location | HoverSensor.java |
| Feature to be tested | The sensor swipe without touching the mirror |
| Feature Pass/Fail Criteria | Passes if user gestures hand right,left,up, or down and it switches to next page |
| Means of control | When the program runs, it runs a python script and it returns output from sensor output is the redirected to java |
| Data | The data is read from output that is sent from the python script which is redirected to Java. |
| Test Procedure | The test is started when the user stands in front of the mirror and gestures hand to move to next page |
| Special requirements | Stand in front of mirror and wave hand in close proximity of the mirror. |

Aqsa Sohail

Type: Unit Testing and Integration Testing (Big Bang testing)

Tested: 3 times

|  |  |
| --- | --- |
| Test-Case Identifier | Raspberry Pi |
| Test Location | display.java |
| Feature to be tested | Raspberry Pi turns on when plugged in.. |
| Feature Pass/Fail Criteria | If the test passes the raspberry pi displays the home screen on the 2-way mirror. |
| Means of Control | Automatically runs the program when it is booted up for the first time. |
| Data | The raspberry pi turns on when it is plugged in. |
| Test Procedure | The test starts when the user plugs in the raspberry pi to an electrical outlet and turns on. |
| Special Requirements | It needs a power outlet close enough to plug in. |

Aqsa Sohail

Type: White Box Test

Tested: 3 times

|  |  |
| --- | --- |
| Test-Case Identifier | Validate Clock |
| Test-case Location | ClockLabel.java |
| Feature to be tested | Displaying the magic mirror when we stand in front of it |
| Feature Pass/Fail Criteria | Passes if mirror is turned on, and the clock is displayed at the left corner at all times throughout all the pages |
| Means of control | When the mirror awakes, because it is connected to Wi-Fi, it recognizes the IP Address from the location and gets the longitude and latitude and gives the correct time. |
| Data | The clock data it read from clock API, which is a http request. |
| Test Procedure | The test is started when the user stands in front of the mirror and home screen is displayed with the correct time |
| Special requirements | Just stand in front of mirror |

Hafsah Uddin

Type:Unit Testing and Integration Testing (Big Bang testing)

Tested: 3 times

|  |  |
| --- | --- |
| Test-Case Identifier | 2-way mirror |
| Test Location | Display.java |
| Feature to be tested | 2- way mirror displays the icons and the user is also able to use it as a mirror |
| Feature Pass/Fail Criteria | If the test passes the 2-way mirror displays the icons and also allows the user to use it as an actual mirror. |
| Means of Control | The display layout is done by a grid layout using JPanels. |
| Data | The 2-way mirror reads the data from the raspberry pi and displays. |
| Test Procedure | The test starts when the user stands in front of the mirror and the user is able to see himself and the icons. . |
| Special Requirements | The user has to stand close enough to the mirror, they cannot be in front of the mirror but on the other side of the room. |

Hafsah Uddin

Type: White box and Integration Testing (Bottom-up testing)

Tested: 3 times

|  |  |
| --- | --- |
| Test-Case Identifier | API |
| Test Location | Weather.java, new.java, stocks.java, clock.java, Quotes.java |
| Feature to be tested | If the API’s of all the icons work and are fetching data. |
| Feature Pass/Fail Criteria | If the test passes the icons will update every five minutes with new data. |
| Means of Control | The API’s do a http request and send the data to the Json objects. |
| Data | After 5 minutes the program reaches out to the API and collects the data. |
| Test Procedure | The test starts first when the program starts and then after 5 minutes when the new data is displayed. |
| Special Requirements | The user has to stand close enough to the mirror, they cannot be in front of the mirror but on the other side of the room. |

Parita Malbari

Type: White Box Test

Tested: 3 times

|  |  |
| --- | --- |
| Test-case identifier | Validate Weather |
| Test location | WeatherData.java |
| Feature to be tested | Weather |
| Feature Pass/Fail Criteria | The test passes if the weather is displayed in the corner of the mirror at all times regardless of switching to another screen. |
| Means of control | The parsing of the JSon Object is done through the  private void parse(JsonObject obj) method. This allows for the information about the weather to be displayed in the correct format. |
| Data | Weather is read from the API and is then parsed to get the input onto the mirror. |
| Test Procedure | The test is started when the mirror turns on and the home screen is displayed with the weather being present. |
| Special requirements | The user has to stand close enough to the mirror, they cannot be in front of the mirror but on the other side of the room. |

Parita Malbari

Type: Black Box Test

Tested: 3 times

|  |  |
| --- | --- |
| Test-case identifier | Validate News |
| Test location | NewsData.java |
| Feature to be tested | News |
| Feature Pass/Fail Criteria | The test passes if the news is displayed when the user uses a hand gesture and switches onto that screen. |
| Means of control | The parsing of the JSon Object is done through the  private void parse(JsonObject obj) method. This allows for the information about the news to be displayed in the correct format. |
| Data | News is read from the API and is then parsed to get the input onto the mirror. |
| Test Procedure | The test is started when the user swipes their hand and brings up the news panel. |
| Special requirements | The user has to stand close enough to the mirror so the sensor can detect the hand movement to switch the panels. |

Hena Shah

Type: Black Box Test

Tested: 3 times

|  |  |
| --- | --- |
| Test-case identifier | Validate Stocks |
| Test location | StocksData.java |
| Feature to be tested | Stocks |
| Feature Pass/Fail Criteria | The test passes if the stocks are displayed when the user uses a hand gesture and switches onto that screen. |
| Means of control | The parsing of the JSon Object is done through the  private void parse(JsonObject obj) method. This allows for the information about the stocks to be displayed in the correct format. |
| Data | Stocks is read from the API and is then parsed to get the input onto the mirror. |
| Test Procedure | The test is started when the user swipes their hand and brings up the stocks panel. |
| Special requirements | The user has to stand close enough to the mirror so the sensor can detect the hand movement to switch the panels. |

Hena Shah

Type: Black Box Test

Tested: 3 times

|  |  |
| --- | --- |
| Test-case identifier | Validate Date |
| Test location | Location Date.java |
| Feature to be tested | Whether the  date correct to the time zone the user is in. |
| Feature Pass/Fail Criteria | The test passes if the date is correct and it appears. |
| Means of control | When the mirror is turned, because it is connected to Wi-Fi, it recognizes the IP Address from the location and gets the longitude and latitude and gives the correct date. |
| Data | The clock data it read from clock API, which is a http request. |
| Test Procedure | The test is started when the user stands in front of the mirror and home screen is displayed with the correct time. |
| Special requirements | Just stand in front of mirror |

**Rationale**

When given the assignment we each had our own ideas as to what we wanted to do. Each group member came up with an idea and we voted on which one we wanted to create. We came up with ideas such as: a t-shirt designing software/ website, hotel reservation website, airplane seat reservation website, a game, and a magic mirror. After careful deliberation we decided we wanted to create a magic mirror because since technology is advancing household products can also become technologically inclined. Another reason for choosing this project is that many people were creating websites and apps but we wanted to create an actual product. We saw a few examples of smart mirrors online and were amazed. We wanted to know how it was created and if we could replicate it with additional features.

The several use cases we picked are all crucial to the successful operation of our Magic Mirror design. We will use various API’s that will act on the system without the use of an external actor. The program will automatically run an API request for each API every 5 minutes that will not require the presence or input of the user. We chose to use a Raspberry Pi to run our program on because it is a very small, lightweight computer that has a lot of programmer support for various hardware devices. We are using a gesture sensor because we needed a way to receive user input without a keyboard and mouse or a touchscreen. We don’t want to use a touchscreen because a crucial part of this product is mirror. If our display was touchscreen, the use would have fingerprints all over the mirror. Our addition of the music use case was to add an extra element of enjoyment to the user. Now the user can get ready for work while gathering up to date information about their day and listen to music while they do it.

There are many software architecture components in this project. We will be using a Raspberry Pi to run our program on as it is small and lightweight. The Raspberry Pi uses Raspbian as an operating system. We are also using a gesture sensor programmed so that we can receive user input without getting fingerprints on the mirror from a touchscreen. There will also be a motion sensor programmed so that the Magic Mirror turns on when it senses someone in front of it. The program will be programmed mostly in Java. Since this is an actual device we won’t be needing a data base system to store data. We will also be using IP addresses to determine the location of the Magic Mirror to show the correct time and weather.

There are three different types of objects in this project. We have entity objects, boundary objects, and control objects. The entity objects consist of the news, weather, calendar, stocks, and clock widgets. They each have their own page, but the weather and clock are displayed on top of each page for better availability. In this project the boundary objects are the mainDisplay, weatherDisplay, and clockDisplay. Our control objects are the motion sensor, gesture sensor, and APICaller.

The project will be tested using various test cases. The 2-way mirror shall use Unit Testing and Integration Testing (Big Bang testing) type of test case. The test case shall test whether the 2-way mirror displays the icons and allows the user to use it as an actual mirror. The HoverSensor test case shall test whether you can change the page of the display without touching the screen only using gestures. It shall be tested using the Unit Test and Integration Test (Big Bang test). The RaspberryPi test case shall use Unit Testing and Integration Testing (Big Bang test). The test case shall test whether the RaspberryPi wakes up from sleeping mode when the user is detected in front of the mirror. The Validate Clock test case shall test whether the clock displays on when the display turns on. It also tests whether the clock displays on every page. It shall be tested using the White Box test. API test case shall test whether the program is receiving the data from the APIs. It shall be tested using the White Box test and Integration test (Bottom-up test). The Validate Weather test case shall test whether the weather is displayed on all pages. It shall be tested using the White Box test. Another test, the Validate News test case tests whether the news page is displayed. It shall be tested using the Black Box Test. Validate Stocks test case shall test whether the stocks are displayed when the user uses a gesture to switch onto the screen that displays the stocks. It shall be tested using the Black Box Test. The Validate Calendar test case shall test whether the calendar is displayed on the correct page. It shall be tested using the Black Block Test. The test cases helped us understand where the mistakes and how we can improve our project.

**Prototype**

|  |  |  |
| --- | --- | --- |
| Home Screen | News Page | Stocks Page |
| 10:45PM HI LO  Current temp Sunset/Sunrise    Hello | 10:45 PM HI Lo  Current temp Sunset/Sunrise  5 top News stories | 10:45 PM HI LO  Current temp Sunset/Sunrise  Stocks |
| Weather page | Quote page |
| 10:45 HI LO  Current temp Sunset/Sunrise    Weekly Weather | 10:45 HI LO  Current temp Sunset/Sunrise  Quote |

**Dictionary**

* **Raspberry pi:** light weight computer
* **Raspbian:** operating system
* **Java:** most universal coding language
* **Motion Sensor:** detects motion
* **Gesture Sensor:** detects gestures/ hand motions
* **LCD monitor(Liquid Crystal Display):** connects to a computer and shows the display
* **Sleep mode/power saving mode:** when a device or parts of a device are turned off until they are needed again
* **API(Application Programming Interface):** a set of subroutines and tools to build a application software.
* **IP address (Internet Protocol):** a string of numbers that is different for each computer and identifies each computer in order to communicate over a network
* **RPI:** short for Raspberry Pi
* **Hover:** gesture sensor used for the Magic Mirror project.
* **Dark Sky API:** the weather API used.
* **Quandl:** The stock API used.

**User Guide**

1. Introduction
   1. Purpose
      1. The Smart Mirror is a product that allows the allows the user to look at themselves and catch up on News, Weather, Stocks, and some daily Motivation.
   2. Who Will Use It?
      1. Clients
   3. How to use it?
      1. The user will turn on the

**Function Point Cost Analysis**

**URL:**[**https://docs.google.com/spreadsheets/d/13WvHcL0TW1ZYej95\_Xp9yHSYpTyqTDF-LzTu9nqzcG8/edit#gid=0**](https://docs.google.com/spreadsheets/d/13WvHcL0TW1ZYej95_Xp9yHSYpTyqTDF-LzTu9nqzcG8/edit#gid=0)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Measurement Parameters** |  |  |  |  |  |  |  |  | **FPC- Rating Estimate(VAF)** |  |
|  | Count |  | Simple | Average | Complex |  | Total |  | **Category** | **Rating** |
| Number of User Inputs | 10 | X | 3 | 4 | 6 |  | 40 | 1 | Does the system require reliable backup and recovery? | 0 |
| Number of User Outputs | 7 | X | 4 | 5 | 7 |  | 35 | 2 | Are data communications required? | 5 |
| Number of User Inqueries | 0 | X | 3 | 4 | 6 |  | 0 | 3 | Are there distributed processing functions? | 2 |
| Number of Internal Files | 5 | X | 7 | 10 | 15 |  | 50 | 4 | Is performance critical? | 5 |
| Number of External Interfaces of Files | 0 | X | 5 | 7 | 10 |  | 0 | 5 | Will the system run in a existing, heavliy utilized operational environment? | 0 |
|  |  |  |  |  |  | Grand Total (FP) | 125 | 6 | Does the system require on-line data entry? | 3 |
|  |  |  |  |  |  |  |  | 7 | Does the on-line data entry require the input transaction to be built over multiple screen operations? | 3 |
| **UAP** | 125 |  |  |  |  |  |  | 8 | Are the master files updated on-line? | 0 |
| **VAF** | 0.94 |  |  |  |  |  |  | 9 | Are the inputs, outputs, files or inquires complex? | 4 |
| **FPC** | 117.5 |  |  |  |  |  |  | 10 | Is the internal processing complex? | 4 |
| **$ per FPC** | 300 |  |  |  |  |  |  | 11 | Is the code designed to be reusable? | 5 |
| **$ Total** | 58750 |  |  |  |  |  |  | 12 | Are conversion and installation included in the design? | 0 |
| **$ per team member** | 11,750 |  |  |  |  |  |  | 13 | Is the system designed for multiple installations in different organizations? | 0 |
| **Hourly wage(per person)** | 25.89 |  |  |  |  |  |  | 14 | Is the application designed to facilitate change and ease of use by the user? | 0 |
| **Median hourly wage for programmers** | 38.24 |  |  |  |  |  |  |  | **Total sum of all category ratings** | 29 |

**Workshare Document**

These were everyone’s primary roles during the project. However, we assisted each other and we all did parts of the programming.

1. Chris Kazenske: Project Manager/Primary Programmer
2. Hafsah Uddin: Technical writer/Coordinator
3. Aqsa Sohail: Design/ Technical writer
4. Hena Shah: Tester/Documentation
5. Parita Malbari: Tester/Documentation