CSC 4350 Software Engineering

Magic Mirror

Fall 2017

Deliverable 3

**Group Name:** CHHAP

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

October 5,2017

**Prototype**

|  |  |  |
| --- | --- | --- |
| Hello | Clock | Weather |
| Up  Left Hello Image result for hello Right  Down | 10:45PM | 10:45PM  Current Temp: 30 degrees  Forecasted Temp: 20 degrees  Sunrise: 6:00AM  Sunset: 8:00PM  Outfit suggestion:  Free Images : mountain, snow, cold, leather, warm, fur, coat ... |

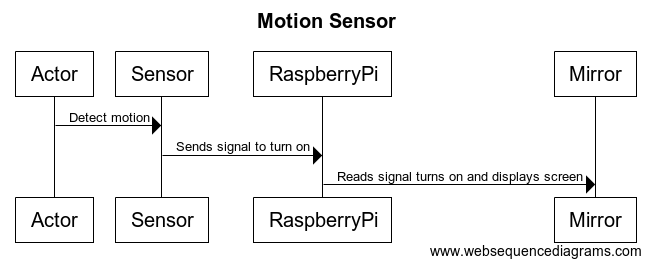
|  |  |  |
| --- | --- | --- |
| Calendar | News | Stocks |
| 10:45PM  Tasks for the day and week  Image result for google calendar | 10:45  Hot Headlines from various sources  Hot headlines based on location  Image result for ary digital news | 10:45  Top daily and weekly stocks  Image result for stocks |

**Requirement Traceability Matrix**

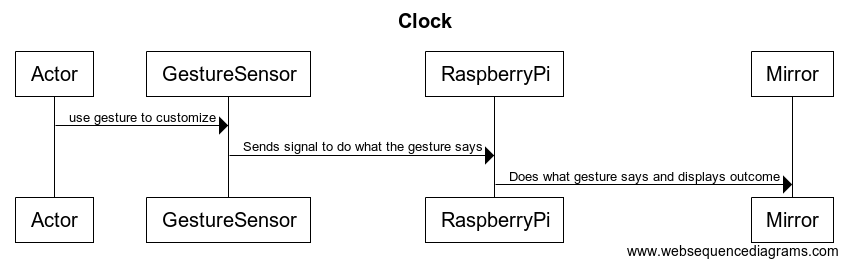
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entry # | Paragraph # | Requirements Traceability Matrix (RTM) | Type | Use Case Name |
| 1. | 1.2 | We will be using a Raspberry Pi to run our program on as it is small and lightweight. | HW, NTH | UC 08, 2 Way Mirror |
| 2. | 1.3 | The Raspberry Pi uses Raspbian as an operating system. | SW | UC 03, Gesture Sensor |
| 3. | 1.4 | We are also using a gesture sensor programmed so that we can receive user input without getting fingerprints on the mirror from a touchscreen. | HW,NTH | UC 03, Gesture Sensor |
| 4. | 1.5 | There will also be a motion sensor programmed so that the Magic Mirror turns on when it senses someone in front of it. | HW | UC 01, Motion Sensor |
| 5. | 1.6 | 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. | SW | UC 10, API |
| 6. | 1.7 | We will also be using IP addresses to determine the location of the Magic Mirror to show the correct time and weather. | SW,NTH | UC 02, Clock |
| 7. | 2.3 | The entity objects consist of the news, weather, calendar, stocks, and clock widgets. | NTH, SW | UC 02,clock |
| 8. | 2.4 | They each have their own page, but the weather and clock are displayed on top of each page for better availability. | NTH, SW | UC 04, Weather |
| 9. | 2.5 | In this project, the boundary objects are the mainDisplay, weatherDisplay, and clockDisplay. | NTH, SW | UC 04, Weather |
| 10. | 2.6 | Our control objects are the motion sensor, gesture sensor, and APICaller. | NTH, SW | UC 04, Weather |

**Use Cases and Interaction Diagrams**

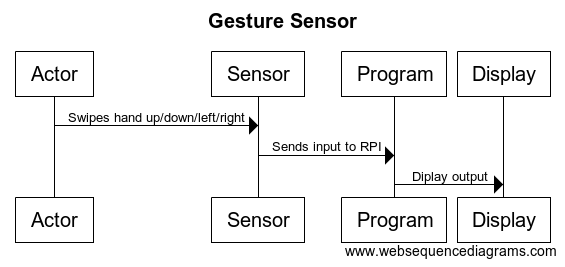
|  |  |
| --- | --- |
| Use Case ID | USC\_001 |
| Use Case Name | USC\_MotionSensor |
| Participants | User, System |
| Entry Condition | User must stand in front of the mirror. |
| Flow of Events | 1. User stands in front of the mirror 2. Motion sensor senses movement in front of the mirror. 3. System turns on. 4. System first displays Hello. 5. System displays pages. 6. User moves away from in front of the mirror to turn off. |
| Exit Condition | User must move away from in front of the mirror. |
| Quality Restraints | The motion sensor must sense motion in front of the mirror in order to turn on. When the motion sensor does not sense someone in front of the mirror it will turn off. |

****

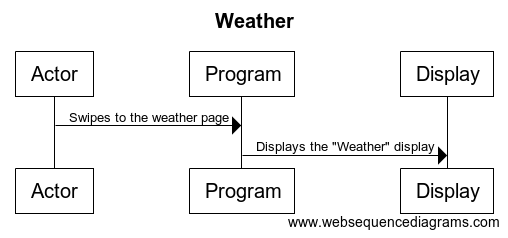
|  |  |
| --- | --- |
| Use Case ID | USC\_002 |
| Use Case Name | USC\_Clock |
| Participants | User, System |
| Entry Condition | The system must be turned on. |
| Flow of Events | 1. User stands in front of the mirror 2. System turns on and displays screen 3. The system will automatically locate the location of the mirror based on IP address from the network. 4. The system will display the correct time based on the location. 5. The clock is displayed on the top left. 6. The user can edit the clock using hand gestures. The user can change many things such as the color of the clock, analog to digital, and standard time to military time by using gestures. 7. 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. Hand gestures must also be used to change settings on the clock. |

****

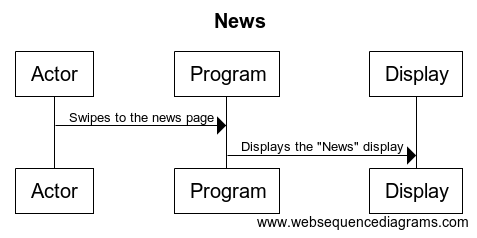
|  |  |
| --- | --- |
| Use Case ID | USC\_003 |
| Use Case Name | USC\_GestureSensor |
| Participants | User, System |
| Entry Condition | The User must stand in front of the mirror |
| Flow of Events | 1. The user gestures their hands up and down or left and right to get to the desired page. 2. The user can gesture their finger in a circular motion to turn the volume up and down. 3. 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. |

****

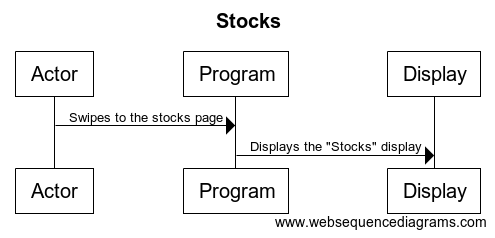
|  |  |
| --- | --- |
| Use Case ID | USC\_004 |
| Use Case Name | USC\_Weather |
| Participants | User, System |
| Entry Condition | The system must be turned on. |
| Flow of Events | **Actor:**  1. When the user turns on the Magic Mirror, the weather will be displayed in the top(left/right) corner, and the actor can use hand gestures to swipe to the next screen.  2. Along with weather, the actor can view other features (stocks, calendar, etc.)  **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. |

****

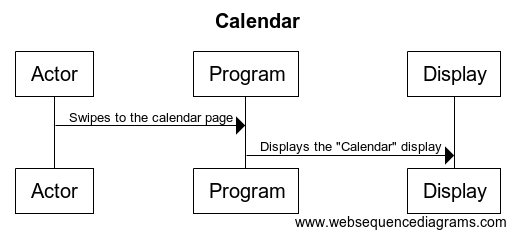
|  |  |
| --- | --- |
| Use Case ID | USC\_005 |
| Use Case Name | USC\_News |
| Participants | User, System |
| Entry Condition | The system must be turned on. |
| Flow of Events | **Actor:**   1. The news will be displayed on the screen and the top 5 news headlines will be available for the user. 2. User can use hand gestures to swipe to the next set of news headlines, such as going from CNN to ESPN.   **System:**   1. The mirror will display the home screen. 2. The system will get the news through various APIs and the top 5 news headlines will be displayed. 3. The mirror will pick up hand gestures through means of a sensor gesture and the mirror will change to a different news source. |
| 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 without the user having to wait more than 15 seconds. The code and the software should run immediately to the point where the user should not have to wait at all. |

****

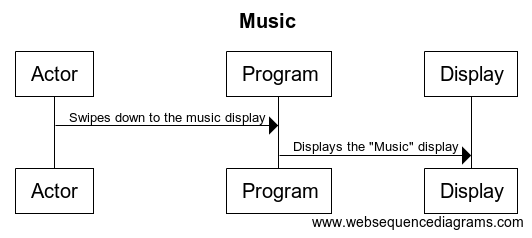
|  |  |
| --- | --- |
| Use Case ID | USC\_006 |
| Use Case Name | USC\_Stocks |
| Participants | User, System |
| Entry Condition | The system must be turned on. |
| Flow of Events | **Actor:**   1. The actor will approach the mirror so the mirror “turns on” through the means of sensor detection. 2. The stocks will be displayed on the screen and the top 5 stocks will be available for the user. 3. User can use hand gestures to swipe to the next set of stock sources.   **System:**   1. The mirror will “turn on” and the basic home screen will be displayed. 2. The system will get the stocks through various APIs and the top 5 stocks will be displayed. 3. Swipe gestures will be used to change screens. |
| Exit Condition | The stocks will be displayed. There will be various stock sources that are used and the user will be able to switch from sources using hand gestures through means of a sensor gesture. |
| Quality Restraints | The stock prices should be displayed from the various sources without the user having to wait more than 15 seconds. The code and the software should run immediately to the point where the user should not have to wait at all. |

****

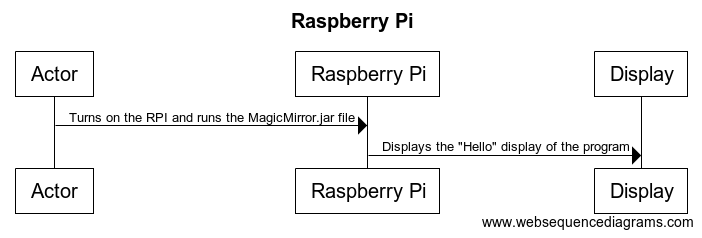
|  |  |
| --- | --- |
| Use Case ID | USC\_007 |
| Use Case Name | USC\_Calendar |
| Participants | User, System |
| Entry Condition | The system must be turned on. |
| Flow of Events | **Actor:**  1. When the user turns on the Magic Mirror, Calendar will be displayed on the next screen and the actor Shall use hand gestures to swipe to the next screen.  2. Along with weather, the actor can view other features (stocks, calendar, etc.) on each screen.  **System:**  1. On the basic home screen 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 | Actor swipes left/right. |
| Quality Restraints | Constant HDMI connection to computer (Raspberry Pie).  Constant power supply. |

****

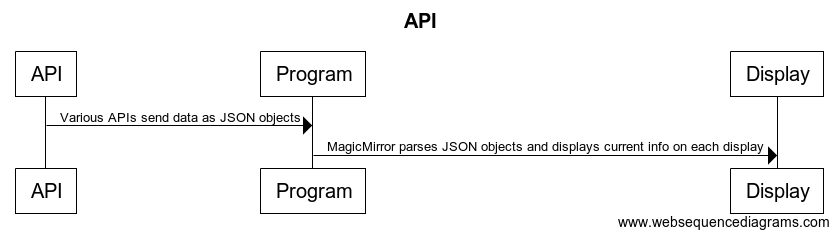
|  |  |
| --- | --- |
| Use Case ID | USC\_008 |
| Use Case Name | USC\_Music |
| Participants | User, System |
| Entry Condition | The system must be turned on. |
| Flow of Events | 1. The user swipes down while the display is on any screen.  2. The system displays music options: playlists, volume control, play/pause button, and next/previous track.  3. The user swipes in the direction the options display/  4. The system executes the command given. |
| Exit Condition | The user swipes up to exit the music display. |
| Quality Restraints | The music command must be executed within one second of the user issuing the command. |

****

|  |  |
| --- | --- |
| Use Case ID | USC\_009 |
| Use Case Name | USC\_RaspberryPi |
| Participants | User, System |
| Entry Condition | The system must be turned on. |
| Flow of Events | 1. The user must run the Magic Mirror.jar file on the Raspberry Pi  2. The program will run without any initial user interaction and display the main display screen (as seen in the “Hello” prototype).  3. 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\_010 |
| 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. |

****

**Database to be used**

Google API (for Calendar)

Darksky.net API (for Weather)

Newsapi.org (for News)

Quandl API (for Stocks)

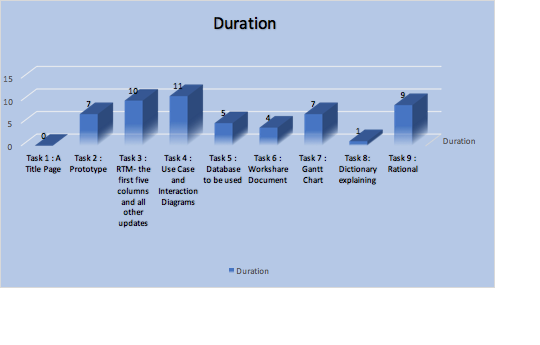
**Workshare Document**

**Phase 3:**

|  |  |  |
| --- | --- | --- |
| Task | Assigned to | Due Date |
| RTM | Aqsa Sohail | 10/6/2017 |
| Gantt Chart | Hena Shah | 10/7/2017 |
| Dictionary | Parita Malbari | 10/6/2017 |
| Rational | Chris Kazenske | 10/7/2017 |
| Use Case 1&2, Interaction Diagrams 1&2 | Hafsah Uddin | 10/3/2017 |
| Use Case 3&4, Interaction Diagrams 3&4 | Aqsa Sohail | 10/3/2017 |
| Use Case 5&6, Interaction Diagrams 5&6 | Hena Shah | 10/3/2017 |
| Use Case 7&8, Interaction Diagrams 7&8 | Parita Malbari | 10/3/2017 |
| Use Case 9&10, Interaction Diagrams 9&10 | Chris Kazenske | 10/3/2017 |

**Gantt Chart**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task** | **Description** | **Duration** | **Start** | **End** |
| **Task 1** | **A Title Page** | **0** | **09/25/2017** | **09/25/2017** |
| **Task 2** | **Prototype** | **7** | **09/26/2017** | **10/3/2017** |
| **Task 3** | **RTM- the first five columns and all other updates** | **10** | **09/27/2017** | **10/7/2017** |
| **Task 4** | **Use Case and Interaction Diagrams** | **11** | **09/25/2017** | **10/6/2017** |
| **Task 5** | **Database to be used** | **5** | **09/29/2017** | **10/4/2017** |
| **Task 6** | **Workshare Document** | **4** | **10/2/2017** | **10/6/2017** |
| **Task 7** | **Gantt Chart** | **7** | **09/30/2017** | **10/7/2017** |
| **Task 8** | **Dictionary explaining** | **1** | **09/26/2017** | **09/27/2017** |
| **Task 9** | **Rational** | **9** | **09/27/2017** | **10/06/2017** |

****

**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

**Rationale**

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.