.

CSC 4350 Software Engineering

Fall 2017

Deliverable 5

**Group Name:** CHHAP

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

November 4,2017

**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.

Requirement Traceability Matrix (RTM)

|  |  |  |  |
| --- | --- | --- | --- |
| 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. | SW |
| 3. | 1.4 | The Magic Mirror shall also use a gesture sensor to make it more user friendly. | HW |
| 4. | 1.5 | 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. | 1.7 | It shall have a motion sensor that will detect motion in the room and display the screen when the person is in range. | HW |
| 6. | 2.1 | The mirror shall display many features including a clock, up to date weather, news, stocks, and many others. | SW |
| 7. | 2.2 | The data shall come from various online API’s. | SWC |
| 8. | 2.3 | The clock feature shall display either standard or military time and analog or digital based on the user’s preference. | SW, NTH |
| 9. | 2.6 | 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 |
| 10. | 2.7 | The weather display shall display the current temperature, forecasted temperature, sunrise and sunset times. | SW |
| 11. | 2.8 | It shall also automatically display the weather based on current location, which is determined by the program at runtime. | SW |
| 12. | 2.10 | The weather data shall be updated every 5 minutes automatically. | SWC |
| 13. | 2.11 | The mirror shall display suggested outfits based on the weather by displaying images of clothing items to wear that day. | SW, NTH |
| 14. | 2.12 | The news feature shall display hot headlines from various news sources. | SW |
| 15. | 2.15 | The user shall have the ability to change between different sources by using gestures. | SW |
| 16. | 2.16 | The stocks page shall display various stock info like top daily and weekly stocks. | SW |
| 17. | 2.17 | The calendar page shall display a Google calendar and allow the user to link their personal calendar and get info about the events stored in their calendar. | SW, NTH |
| 18. | 2.19 | Each main display (news, weather, stocks, and calendar) shall be changed using gestures. | SW |
| 19. | 2.20 | The user shall swipe left or right in front of the sensor in order to switch between these different displays. | SW |
| 20. | 2.21 | 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 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 | UC 09, Raspberry Pi |
| 2. | 1.3 | The Raspberry Pi uses Raspbian as an operating system. | SW | UC 09, Raspberry Pi |
| 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 09, Raspberry Pi |
| 6. | 1.7 | Since this is an actual device we won’t be needing a data base system to store data. | HW | UC 09, Raspberry Pi |
| 7. | 1.8 | We will also be using IP addresses to determine the location of the Magic Mirror to show the correct time and weather. | SW | UC 02, Clock  UC 04, Weather  UC 10, API |
| 8. | 2.3 | The entity objects consist of the news, weather, calendar, stocks, and clock widgets. | SW,NTH | UC 02, Clock  UC 05, News  UC 06, Stock  UC 07, Calendar |
| 9. | 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 02,clock  UC 04, Weather |
| 10. | 2.5 | In this project, the boundary objects are the mainDisplay, weatherDisplay, and clockDisplay. | NTH, SW | UC 02,clock  UC 04, Weather |
| 11. | 2.6 | Our control objects are the motion sensor, gesture sensor, and APICaller. | HW,SW | UC 01, Motion Sensor  UC 03, Gesture Sensor  UC 10, API |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entry # | Paragraph # | Requirements Traceability Matrix (RTM) | Type | Use Case Name |
| 1. | 1.1 | The pipe and filter architectural style is when the subsystems process the data received from a set of inputs, and sends the results to other subsystems via a set of outputs. | n/a | n/a |
| 2. | 1.2 | The pipe and filter architectural style we are using is through the means of receiving the data through APIs and processing that data, and having it displayed for the user on the mirror. | SW | UC 10, API |
| 3. | 1.3 | The repository architecture style is also used, because the client sends a request to the system to perform actions; this is completed when the API key is requested and that gives the information which is then parsed into the information that is desired. | HW,NTH | UC 10, API |
| 4. | 2.1 | Due to the way the magic mirror works, both of these architectures are used to retrieve the data and display it on the mirror. | HW | UC 10, API |
| 5. | 2.2 | Moreover, the pipe and filter architecture goes hand in hand with the repository architecture because of the functions they both execute. | SW | UC 10, API |
| 6. | 2.3 | The repository architecture provides us with many advantages such as providing data integrity, and backup and restore features. | SW,NTH | UC 10, API |
| 7. | 2.4 | As well as providing scalability and reusability of multiple variables as they do not have direct communication with each other. | NTH, SW | UC 02, Clock  UC 05, News  UC 06, Stock  UC 07, Calendar  UC 10, API |
| 8. | 2.5 | The pipe and filter architecture provides us with similar advantages such as reusability and concurrent execution because each variable can be implemented as separate tasks and be implemented in parallel with other variables. | SW | UC 10, API |
| 9. | 3.1 | We are also using a Raspberry Pi to run the program as it is small and lightweight. | HW | UC 09, Raspberry Pi |
| 11. | 3.2 | Raspbian is the operating system that is used by the Raspberry Pi. | HW | UC 09, Raspberry Pi |
| 10. | 3.3 | It is written in Java and it is the framework for the mirror as well as it is used to provide the APIs for the gesture sensor that will be used for user input, and the interaction that will take place to display the data to the user in a consistent fashion. | NTH, SW | UC 02, Clock  UC 05, News  UC 06, Stock  UC 07, Calendar  UC 10, API |

**Software Architecture**

The pipe and filter architectural style is when the subsystems process the data received from a set of inputs, and sends the results to other subsystems via a set of outputs. The pipe and filter architectural style we are using is through the means of receiving the data through APIs and processing that data, and having it displayed for the user on the mirror. The repository architecture style is also used, because the client sends a request to the system to perform actions; this is completed when the API key is requested and that gives the information which is then parsed into the information that is desired.

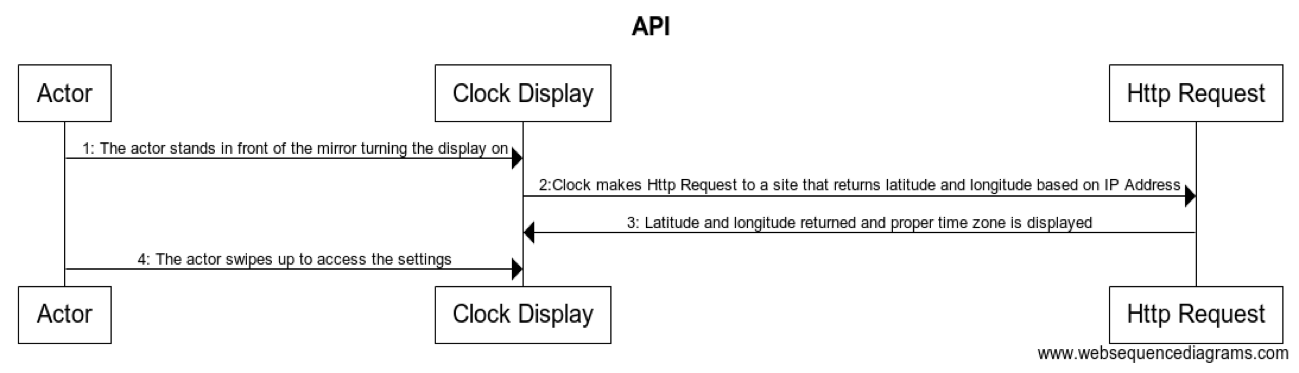
Due to the way the magic mirror works, both of these architectures are used to retrieve the data and display it on the mirror. Moreover, the pipe and filter architecture goes hand in hand with the repository architecture because of the functions they both execute. The repository architecture provides us with many advantages such as providing data integrity, and backup and restore features. As well as providing scalability and reusability of multiple variables as they do not have direct communication with each other. The pipe and filter architecture provides us with similar advantages such as reusability and concurrent execution because each variable can be implemented as separate tasks and be implemented in parallel with other variables.

We are also using a Raspberry Pi to run the program as it is small and lightweight. Raspbian is the operating system that is used by the Raspberry Pi. It is written in Java and it is the framework for the mirror as well as it is used to provide the APIs for the gesture sensor that will be used for user input, and the interaction that will take place to display the data to the user in a consistent fashion.

**Category Interaction Diagrams**

Chris Kazenske CID:

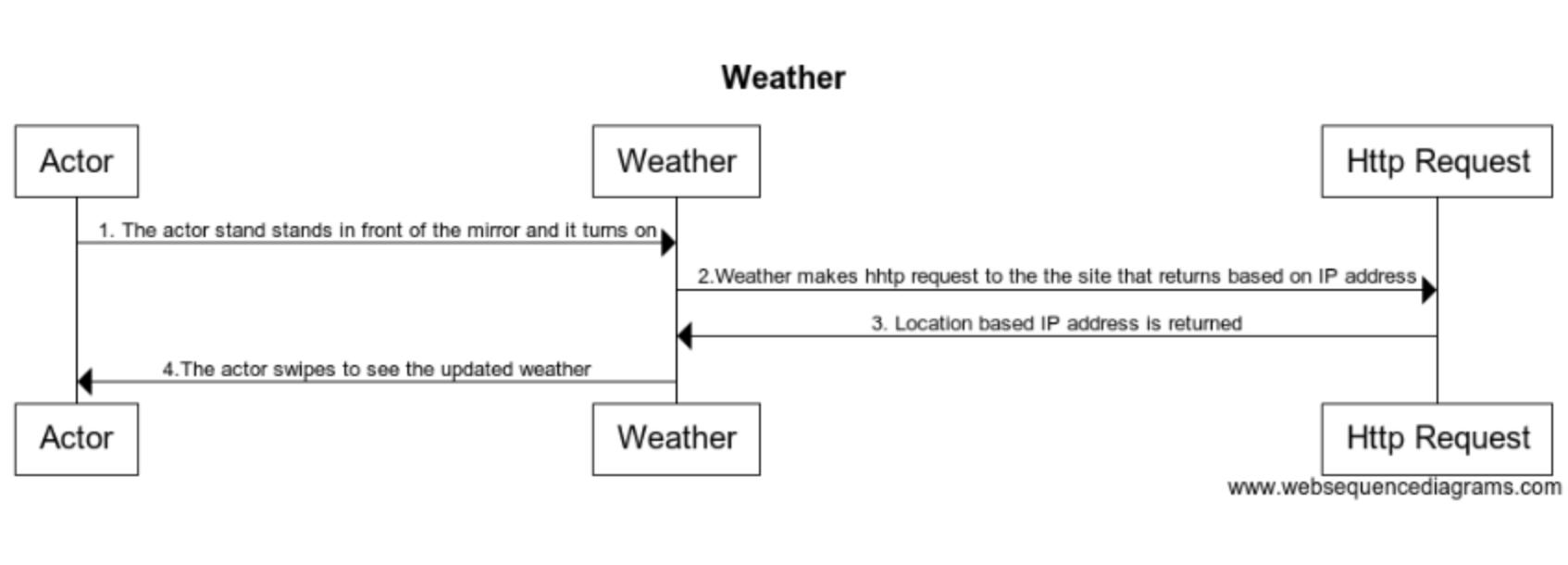
USC\_002\_CLOCK\_CID



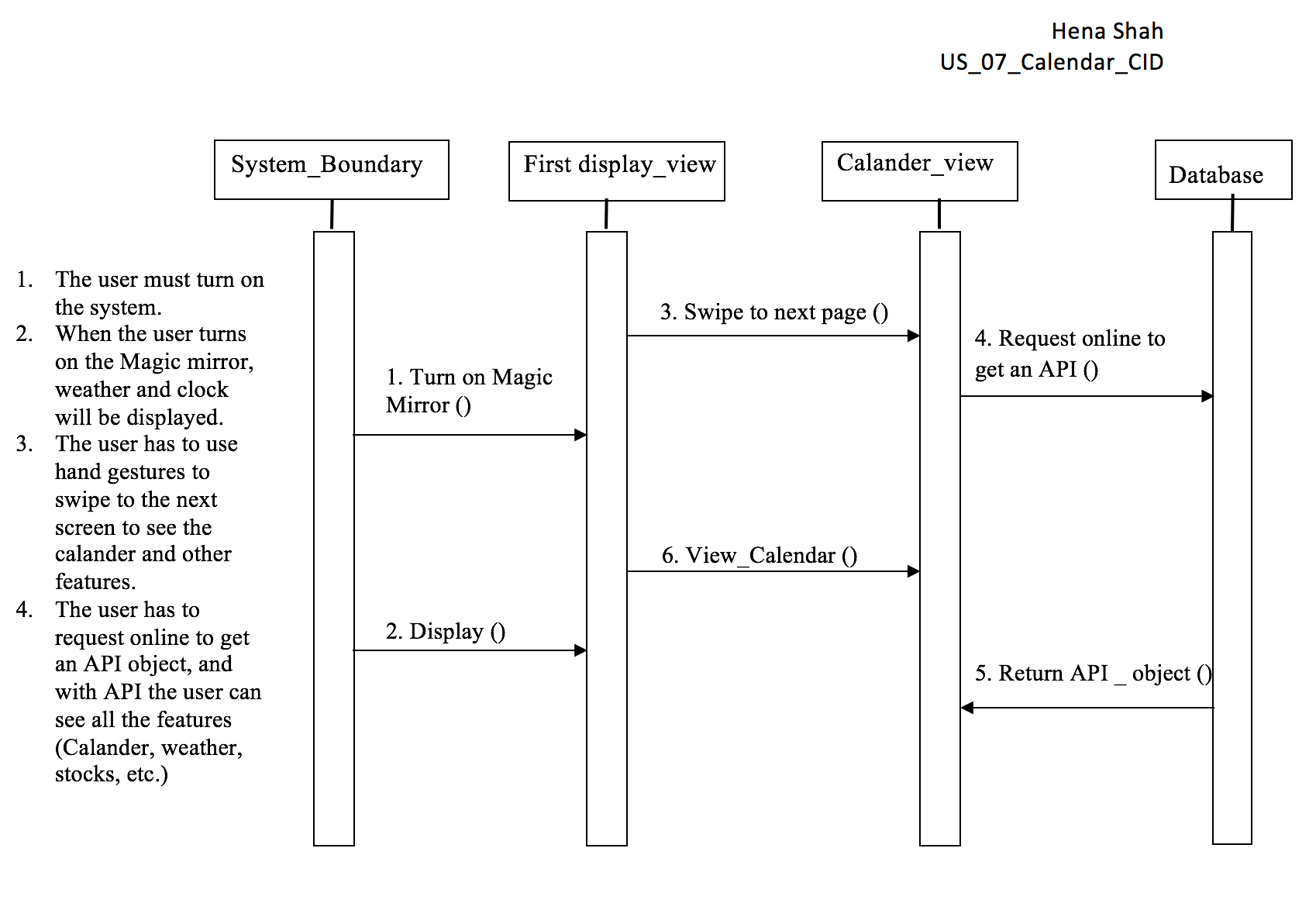
|  |
| --- |
| 1. The actor stands in front of the mirror turning the display on.  2. The system will automatically locate the location of the mirror based on IP address from the network.  3. The system will display the correct time based on the location.  4. The user can edit the clock using hand gestures. The user can |

Hafsah Uddin: CID

USC\_004\_WEATHER\_CID

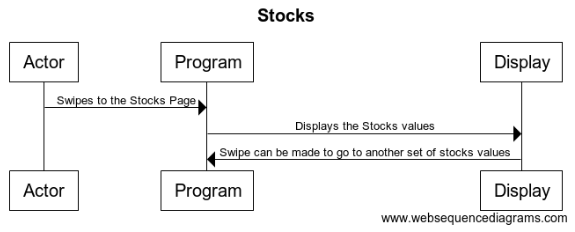
****

|  |
| --- |
| 1. The actor stand stands in front of the mirror and it turns on  2.Weather makes http request to the the site that returns based on IP address  3. Location based IP address is returned  4.The actor swipes to see the updated weather |



Aqsa Sohail: CID

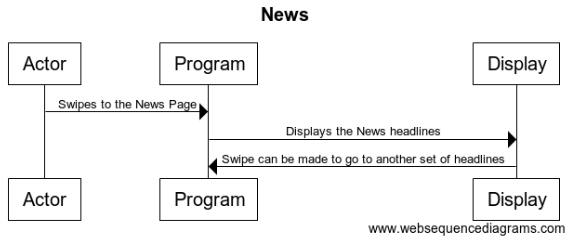
USC\_005\_Stocks\_CID



1. Swipe is made by the Actor to display the Stocks page where the program sends a HTTP request and from there the stock are pulled and updated.
2. The program then displays the stock values on the mirror.
3. The Actor can then swipe again to display another set of stocks values that are either high, low, open, or closed.

Parita Malbari : CID

USC\_006\_News\_CID

****

1. Swipe is made by the Actor to display the News page.
2. The program then displays the News headlines on the mirror.
3. The Actor can then swipe again to display another set of News headlines.

**Workshare Document**

**Phase 5:**

|  |  |  |
| --- | --- | --- |
| Task | Assigned To | Due Date |
| RTM | Aqsa Sohail | 11/3/2017 |
| Gantt Chart | Hena Shah | 11/1/2017 |
| Software Architecture | Parita Malbari | 11/1/2017 |
| Dictionary | Chris Kazenske | 11/2/2017 |
| Rational | Hafsah Uddin | 11/2/2017 |
| Category Interaction Diagram | Hafsah Uddin, Aqsa Sohail, Chris Kazense, Parita Malbari, Hena Shah | 10/20/2017 |

**Gantt Chart**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task** | **Description** | **Duration** | **Start** | **End** |
| Task 1 | A Title Page | 0 | 10/23/2017 | 10/23/2017 |
| Task 2 | Rationale | 10 | 10/25/2017 | 11/04/2017 |
| Task 3 | RTM- the first five columns and all other updates | 7 | 10/24/2017 | 10/31/2017 |
| Task 4 | Software Architecture | 5 | 10/25/2017 | 10/30/2017 |
| Task 5 | A category Interaction Diagram | 6 | 10/27/2017 | 11/02/2017 |
| Task 6 | WSD (Workshare document) | 3 | 10/24/2017 | 10/26/2017 |
| Task 7 | Gantt Chart | 2 | 11/02/2017 | 11/04/2017 |
| Task 8 | Dictionary explaining | 1 | 10/30/2017 | 10/31/2017 |

**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
* **Hover:** gesture sensor used for the Magic Mirror project.
* **Dark Sky API:** the weather API used.
* **Quandl:** The stock API used.