CSC 4350 Software Engineering

Magic Mirror

Fall 2017

Deliverable 6

**Group Name:** CHHAP

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

November 15,2017

**Workshare Document**

**Phase 6:**

|  |  |  |
| --- | --- | --- |
| Task | Assigned To | Due Date |
| RTM | Aqsa Sohail | 11/14/2017 |
| Gantt Chart | Hena Shah | 11/13/2017 |
| Dictionary | Chris Kazenske | 11/14/2017 |
| Rational | Hafsah Uddin | 11/13/2017 |
| Test Cases | Hafsah Uddin, Aqsa Sohail, Chris Kazense, Parita Malbari, Hena Shah | 11/12/2017 |

**Gantt Chart**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task** | **Description** | **Duration** | **Start** | **End** |
| Task 1 | A Title Page | 0 | 11/05/2017 | 11/05/2017 |
| Task 2 | RTM- add an extra column and write the TEST CASE NAME | 3 | 11/05/2017 | 11/08/2017 |
| Task 3 | Test cases used to test your software | 7 | 11/06/2017 | 11/13/2017 |
| Task 4 | WSD (Workshare document) | 5 | 10/07/2017 | 11/12/2017 |
| Task 5 | Gantt Chart | 3 | 11/12/2017 | 11/15/2017 |
| Task 6 | Dictionary explaining | 3 | 11/09/2017 | 11/12/2017 |
| Task 7 | Rationale for test cases | 9 | 11/06/2017 | 11/15/2017 |

**Use Cases**

1. Motion Sensor (Chris)
2. Clock (Aqsa)
3. Gesture Sensor (Chris)
4. News(Parita)
5. Stock (Hena)
6. Weather (Parita)
7. Calendar (Hena)
8. 2 Way Mirror (Hafsah)
9. Raspberry PI (Aqsa)
10. API (Hafsah)

**Test Cases**

Chris Kazenske

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

Tested: 2 times

|  |  |
| --- | --- |
| Test-Case Identifier | MotionSensor |
| Test-case Location | MotionSensor.java |
| Feature to be tested | Displaying the magic mirror when we stand in front of it |
| Feature Pass/Fail Criteria | Passes if turned on if the mirror is turned on as soon as User stands in front |
| Means of control | Through the pins and raspberry pi, if it senses then it will set pin state to high and that is how it would read the state of the pin |
| Data | The motion sensor get the data from pins of the raspberry pi. |
| Test Procedure | The test is started when the user stands in front of the mirror |
| Special requirements | Just stand in front of mirror |

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 wakes up from sleeping mode when the motion sensor senses someone in front of the mirror. |
| 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 motion sensor sends a trigger to the raspberry pi to wake up from sleep mode. |
| Test Procedure | The test starts when the user stands in front of the mirror and the motion sensor senses the user which causes the raspberry pi to awaken. |
| 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. |

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, calander.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 Calendar |
| Test location | CalendarData.java |
| Feature to be tested | Calendar |
| Feature Pass/Fail Criteria | The test passes if the calendar 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 Calendar to be displayed in the correct format. |
| Data | Calendar 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 Calendar 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. |

**Rationale**

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. MotionSensor test case shall test whether the magic mirror turns on when the user stands in front of it. It shall be tested using the Unit Test and Integration Test (Big Bang test). 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.

**Requirements Traceability Matrix (RTM)**

|  |  |  |  |
| --- | --- | --- | --- |
| Entry # | Para/Line # | RTM | Test - Case Type |
| 1 | 1.2 | The 2-way mirror shall use Unit Testing and Integration Testing (Big Bang testing) type of test case. | 2 Way Mirror - Unit Testing and Integration Testing (Big Bang testing) |
| 2 | 1.3 | The test case shall test whether the 2-way mirror displays the icons and allows the user to use it as an actual mirror. | 2 Way Mirror - Unit Testing and Integration Testing (Big Bang testing) |
| 3 | 1.4 | MotionSensor test case shall test whether the magic mirror turns on when the user stands in front of it. | Motion Sensor - Unit Testing and Integration Testing (Big Bang testing) |
| 4 | 1.5 | It shall be tested using the Unit Test and Integration Test (Big Bang test). | Motion Sensor - Unit Testing and Integration Testing (Big Bang testing) |
| 5 | 1.6 | The HoverSensor test case shall test whether you can change the page of the display without touching the screen only using gestures. | Hover Sensor - Unit Testing and Integration Testing (Big Bang testing) |
| 6 | 1.7 | It shall be tested using the Unit Test and Integration Test (Big Bang test) | Hover Sensor - Unit Testing and Integration Testing (Big Bang testing) |
| 7 | 1.8 | The RaspberryPi test case shall use Unit Testing and Integration Testing (Big Bang test). | Raspberry Pi - Unit Testing and Integration Testing (Big Bang testing) |
| 8 | 1.9 | The test case shall test whether the RaspberryPi wakes up from sleeping mode when the user is detected in front of the mirror. | Raspberry Pi - Unit Testing and Integration Testing (Big Bang testing) |
| 9 | 1.10 | The Validate Clock test case shall test whether the clock displays on when the display turns on. | Validate Clock - White Box |
| 10 | 1.11 | It also shall test whether the clock displays on every page. | Validate Clock - White Box |
| 11 | 1.12 | It shall be tested using the White Box test. | Validate Clock - White Box |
| 12 | 1.13 | API test case shall test whether the program is receiving the data from the APIs. | API - White Box test and Integration test (Bottom-up test) |
| 13 | 1.14 | It shall be tested using the White Box test and Integration test (Bottom-up test). | API - White Box test and Integration test (Bottom-up test) |
| 14 | 1.15 | The Validate Weather test case shall test whether the weather is displayed on all pages. | Validate Weather - White Box |
| 15 | 1.16 | It shall be tested using the White Box test. | Validate Weather - White Box |
| 16 | 1.17 | Another test, the Validate  News test case tests whether the news page is displayed. | Validate News - Black Box |
| 17 | 1.18 | It shall be tested using the Black Box Test. | Validate News - Black Box |
| 18 | 1.19 | 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. | Validate Stock - Black Box |
| 19 | 1.20 | It shall be tested using the Black Box Test. | Validate Stock |
| 20 | 1.21 | The Validate Calendar test case shall test whether the calendar is displayed on the correct page. | Validate Calendar - Black Box |
| 21 | 1.22 | It shall be tested using the Black Block Test. | Validate Calendar - Black Box |

**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.