CSC 4350 Software Engineering

Fall 2017

Deliverable 4

**Group Name:** CHHAP

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

October 21,2017

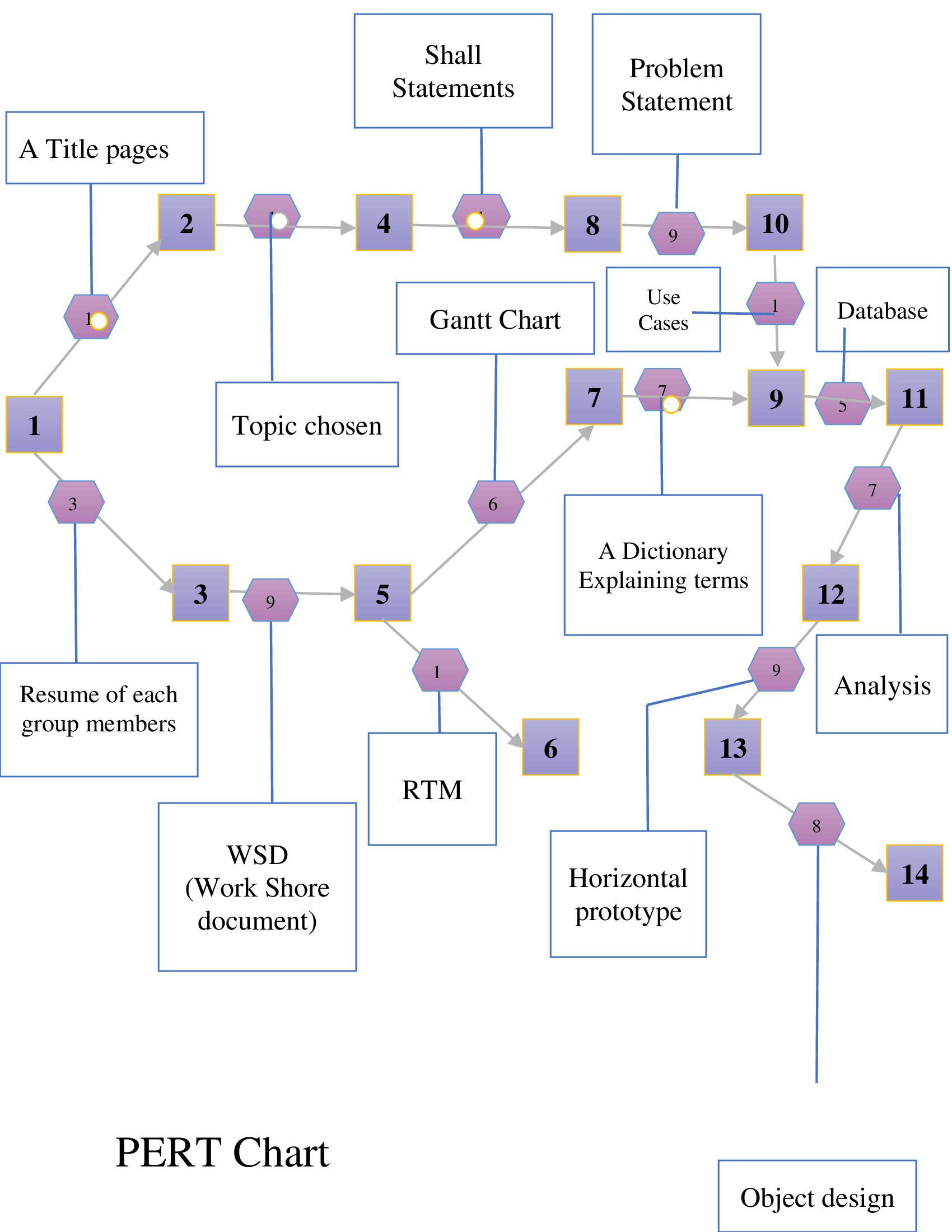
**Rationale**

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**

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

****

**Gantt Chart**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task | Description | Duration | Start | End |
| Task 1 | A Title Page | 0 | 10/7/2017 | 10/7/2017 |
| Task 2 | RTM- the first five columns and all other updates | 4 | 10/7/2017 | 10/11/2017 |
| Task 3 | Cost Analysis | 2 | 10/8/2017 | 10/10/2017 |
| Task 4 | Class Interface | 1 | 10/9/2017 | 10/10/2017 |
| Task 5 | Workshare Document | 8 | 10/7/2017 | 10/15/2017 |
| Task 6 | Pert Chart | 4 | 10/17/2017 | 10/21/2017 |
| Task 7 | Dictionary explaining | 1 | 10/7/2017 | 10/8/2017 |
| Task 8 | Rational | 10 | 10/11/2017 | 10/21/2017 |

**Workshare Document**

**Phase 4:**

|  |  |  |
| --- | --- | --- |
| Task | Assigned To | Due Date |
| RTM | Aqsa Sohail | 10/20/2017 |
| Gantt Chart | Hena Shah | 10/19/2017 |
| Pert Chart | Hena Shah | 10/19/2017 |
| Dictionary | Chris Kazenske | 10/19/2017 |
| Rational | Hafsah Uddin | 10/20/2017 |
| Function Point Cost Analysis | Hafsah Uddin | 10/20/2017 |
| Class Interface | Parita Malbari | 10/20/2017 |

**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 |

**Class Interface**

**Boundary Objects:**

MainDisplay

Attribute:

Clock: String

Weather: String

News: String

Stocks: String

Calendar: String

Method:

getClock()

getWeather()

getNews()

getStocks()

getCalendar()

WeatherDisplay:

Attribute:

CurrTemp: int

HiTemp: int

LowTemp: int

Method:

getCurrTemp()

getHiTemp()

LoTemp()

ClockDisplay:

Attribute:

Hour: Int

Minute: int

Method:

getHour()

getMinute()

**Entity Objects:**

News:

Attribute:

CNNSource: String

SportSource: String

TechSource: String

EntSource: String

Method:

getCNNSource()

getSportSource()

TechSource()

EntSource()

Weather:

Attribute:

CurrTemp: int

HiTemp: int

LowTemp: int

Method:

getCurrTemp()

getHiTemp()

LoTemp()

Calendar:

Attribute:

Days: int

Month: int

Year: int

DaysOfWeek: String

Method:

getDays()

getYear()

getMonth()

getDaysOfWeek()

Stocks:

Attribute:

Name: String

High: Double

Low: Double

PercentChange: Double

Method:

getName()

getHigh()

getLow()

getPercentChange()

Clock:

Attribute:

Hour: Int

Minute: int

Method:

getHour()

getMinute()

**Control Objects:**

Gesture Sensor:

Attribute:

SwipeUp: Boolean

SwipeDown: Boolean

SwipeLeft: Boolean

SwipeRight: Boolean

Method:

getSwipeUp()

getSwipeDown()

getSwipeLeft()

getSwipeRight()

Motion Sensor:

Attribute:

SwipeUp: Boolean

SwipeDown: Boolean

SwipeLeft: Boolean

SwipeRight: Boolean

Method:

getSwipeUp()

getSwipeDown()

getSwipeLeft()

getSwipeRight()

APICaller:

Attribute:

API: String

Method:

getAPI()

**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