CSC4350/6350

Spring 2017

A stock trading software. The price is real-time and the exchange is virtual.

Virtual Stock

BATMAN

Mengyuan Zhu, Sungjae Kim, Sharon Kim,
Jakub Pietrasik, Hyeun Kang

2/28/2017

Virtual Stock

Contents

Requirements Traceability Matrix

Entry#	Para#	Requirements Traceability Matrix	Type	Use Case Name
1	1	Batman shall develop a stock trading software with a user-friendly GUI.	SW, HW	Use Case 3 Show_Main_Window
2	2	The system shall provide a window for user to register an account in Batman.	SW	Use Case 2 User_Sign_Up
3	2.1	The system shall let user know and ask the user to refill the form for registration.	SW	n/a
4	3	The system shall have a window for user to log in.	SW	Use Case 1 User_Sign_In
5	3.1	The window shall provide a user name, password input, forgot password button, and sign in and sign up buttons.	SW	n/a
6	3.2	The system shall let user into the interface if the user type the right user information	SW	n/a
7	3.3	If the user failed to log in to the system, the system shall ask the user to refill the password.	SW	n/a
8	3.4	The system shall provide a pin number for quick log in	NTH	n/a
9	4	The system shall have a portfolio interface.	SW	Use Case 4 Show_Portfolio
10	4.1	The system shall show a chart of user's balance chart of today, total balance value, breaking news of today, stocks that a user keeps and his/her shares, and a watch list.	NTH	n/a
11	4.2	The search button shall also be shown in the corner for user to search a specific stock.	NTH	n/a
12	4.3	The color of the GUI shall be green if the user's balance goes up and be red if the user's balance goes down.	NTH	n/a
13	5	The system shall provide a window to show user's account.	SW	Use Case 5 Show_Account
14	5.1	The account shall show the total balance, stocks balance, and cash balance.	NTH	n/a
15	6	The system shall give a function for user to transfer money to a bank or to the Batman app.	SW	Use Case 6 Show_Banking
6	6.1	The linked accounts shall be shown in the bottom.	NTH	n/a
17	6.2	The system shall provide a bank account for user to deposit money.	SW	n/a
18	7	The system shall provide a list of history of user's trading log.	SW	Use Case 7 Show_History
19	7.1	The system shall also show the date.	NTH	n/a

20	7.2	The system shall allow the user to see the history from/to a specific date.	NTH	n/a
21	8	The system shall have a setting interface for user to reset pin number and update user information including name, password, email, phone number and address.	SW	Use Case 8 Show_Settings
22	8.1	The system shall also have a log out button for the user to log out.	SW	n/a

Work Structure Document

Group name: **Batman**

Name	Tasks
Mengyuan Zhu	Team Coordinator Documents handler Java coder Problem Statement Requirements Traceability Matrix Dictionary A horizontal prototype of the software to be developed Use cases and Interaction Diagrams - example as per given in class Database to be used Rational Architecture
Sungjae Kim	Finalize code documentation Java coder Requirements Traceability Matrix Gannt Chart Category Interaction Diagram
Sharon Kim	User Guide Function Point Cost Analysis. Program tester Rational
Jakub Pietrasik	Rational
Hyeun Kang	Object Design

Dictionary

Portfolio: In finance, a portfolio is a collection of investments held by an investment company, hedge fund, financial institution or individual.

Broker: A person who buys or sells an investment for you in exchange for a fee (a commission). Here is Tim's favorite broker.

Dividend: this is a portion of a company's earnings that is paid to shareholders, or people that own hat company's stock, on a quarterly or annual basis. Not all company's do this.

Exchange: An exchange is a place in which different investments are traded. The most well-known in the United States are the New York Stock Exchange and the Nasdaq.

Quote: Information on a stock's latest trading price. This is sometimes delayed by 20 minutes unless you are using an actual broker trading platform.

Volume: The number of shares of stock traded during a particular time period, normally measured in average daily trading volume.

Yield: This usually refers to the measure of the return on an investment that is received from the payment of a dividend. This is determined by dividing the annual dividend amount by the price paid for the stock. If you bought stock XYZ for $40-a-share and it pays a $1.00-per-year dividend, you have a "yield" of 2.5%.

JDK: The Java Development Kit (JDK) is an implementation of either one of the Java Platform, Standard Edition; Java Platform, Enterprise Edition or Java Platform, Micro Edition platforms released by Oracle Corporation in the form of a binary product aimed at Java developers on Solaris, Linux, Mac OS X or Windows.

GUI: The graphical user interface is a type of user interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation, instead of text-based user interfaces, typed command labels or text navigation.

HTTP: The Hypertext Transfer Protocol (HTTP) is an application protocol for distributed, collaborative, and hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web.

XML: In computing, Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. The W3C's XML 1.0 Specification and several other related specifications-all of them free open standards—define XML.

Rational

Object Rational

Stock.java

Stores information pertaining to stock attributes. Includes name, price, shares, open, TodayHigh, TodayLow, YearHigh, YearLow, Volume, Marketcap, PARatio, and DivyYield values.

StockDayData

Manages retrieval of the stock data from Yahoo and contains method for getting the URL source.

StockListWrapper

Wrapper for Stock List that has a getter and a setter for the person object.

User

Stores information about the user including attributes such as name, address, social security number, and date of birth.

DateUtil

Finds and formats the date from dateString.

AccountController

Provides the elements of the visual interface for the Account screen. This includes labels for total, stock and cash, as well as an OK and a withdraw button.

BankingController

Provides the elements of the visual interface for the Banking screen. This includes buttons for transferring between the bank and VS, automatic depositing, and linking accounts.

HistoryController

Provides the elements of the visual interface for the History screen. This includes a combo box for the date range and a table of the history.

PortfolioController

Provides the elements of the visual interface for the Portfolio screen. This includes elements such as a bar chart, buttons for managing time intervals for day/week/month, and allows for the setting of person data.

RootLayoutController

Provides the elements of the visual interface for the Root Layout screen. Handles the Sign In, Sign Up and Showing of portfolio, banking, account and history screens.

SettingController

Provides the elements of the visual interface for the Setting screen. This includes text fields for account name and password, email, phone, address as well as the buttons for updating the content with that entered as well as one for logging out.

SignInController

Provides the elements of the visual interface for the Sign In screen. Includes buttons for signing in, registering and text fields to allow for the entering of a username and a password.

SignUpController

Provides the elements of the visual interface for the Sign Up screen. Includes labels for the name, password, account name, address, routing number, account number, initial deposit, phone and email contact information. Also includes a button to confirm and one to cancel.

StatisticsController

Provides the elements of the visual interface for the Statistics screen. Includes a bar chart and category axis of stock information. Allows for the viewing of and setting of Person Data.

StockDayController

Provides the elements of the visual interface for the Stock Day screen. Allows for the setting of stock quote data as well as the controller for Stock Day.

StockOverviewController

Provides the elements of the visual interface for the Stock Overview screen. Manages overview elements including stock name, stock price, shares, open, today high, today low, year high, year low, volume, average volume, market cap, PERatio, and div field. Shows person details.

StockPurchaseController

Provides the elements of the visual interface for the Stock Purchase screen. Displays the elements of stock name, shares, and the dialog box that appear when making a stock purchase.

Architecture Rational

Virtual Stock software is in the Model/View/Controller (MVC) architecture style. Subsystems are classified into three different type. It has these reasons: Simultaneous Development - Multiple developers can work simultaneously on the model, controller and views. High Cohesion - MVC enables logical grouping of related actions on a controller together. The views for a specific model are also grouped together. Low Coupling - The very nature of the MVC framework is such that there is low coupling among models, views or controllers. Ease of modification - Because of the separation of responsibilities, future development or modification is easier. Multiple views for a model - Models can have multiple views.

Use cases Rational

Use Case 1: User_Sign_In
User accounts personalize Virtual Stock usage so that a User can reuse many Virtual Stock features customized to the User's needs. These User accounts are also necessary to implement the Virtual Stock protocol function. With those aspects in mind, this Use Case ensures that a User has a Virtual Stock account before using Virtual Stock's main functions.

Use Case 2: User_Sign_Up
This Use Case's rationale directly follows from the rationale given in UC1.

Use Case 3: Show_Main_Window
By showing main window, users have access to see several other windows. Such as stock name, buying/selling stocks, access to portfolios, personal account, banking account, history, settings, about, and exiting the program.

Use Case 4: Show_Portfolio
By showing portfolio, users have access to see days, weeks, months, and years worth of stocks.

Use Case 5: Show_Account
By showing account, users have easy access to their money and funds which they can withdraw from their virtual stock to their bank.

Use Case 6: Show_Banking
By showing bank account, the user has easy access to deposit money at a specific time to a bank or virtual stock by routing number and account number.

Use Case 7: Show_History
By showing history, users can see specific dates of stocks and their prices of when they are bought or sold

Gantt Chart

Virtual Stock

Batman

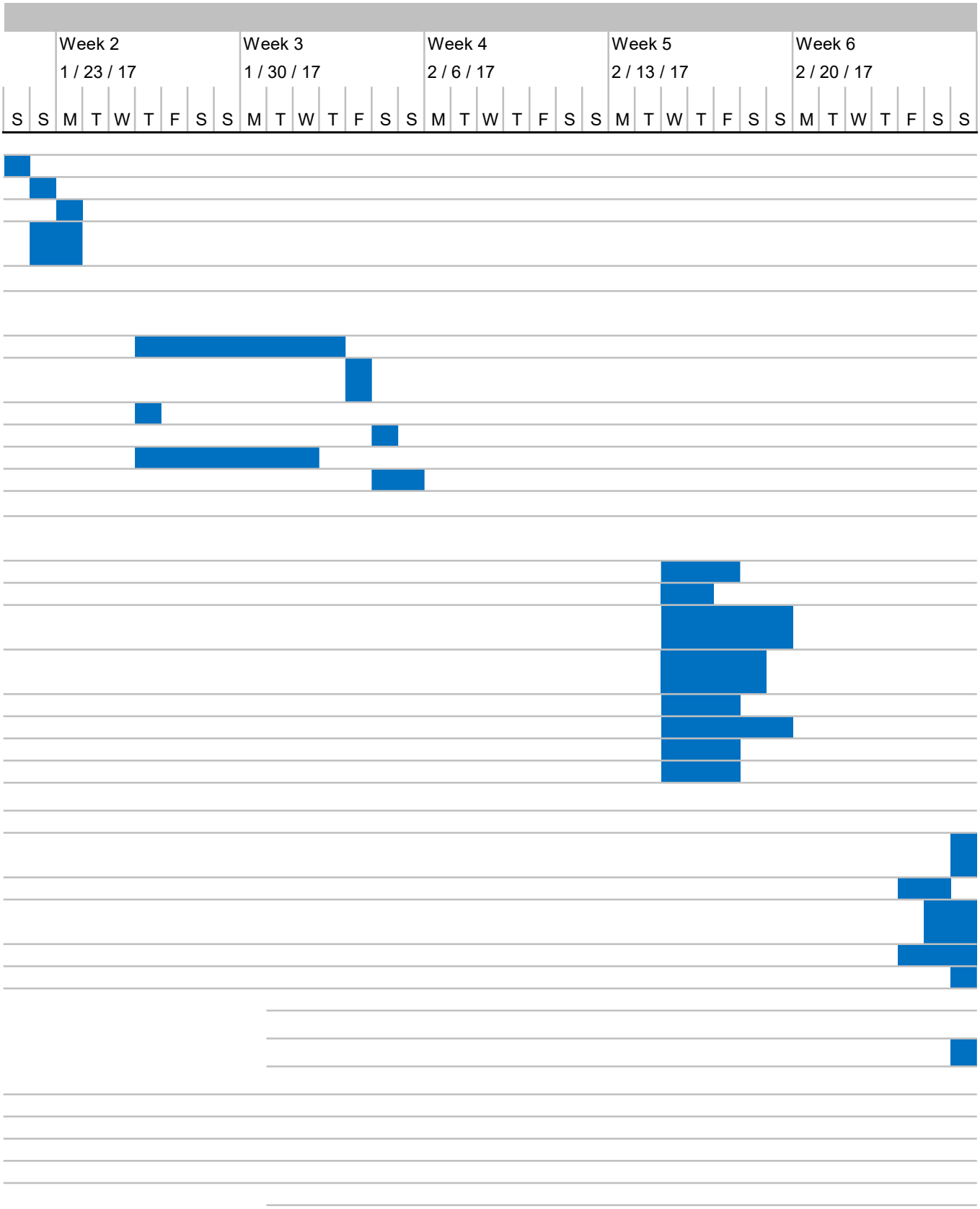
Project Lead: Mengyuan Zhu

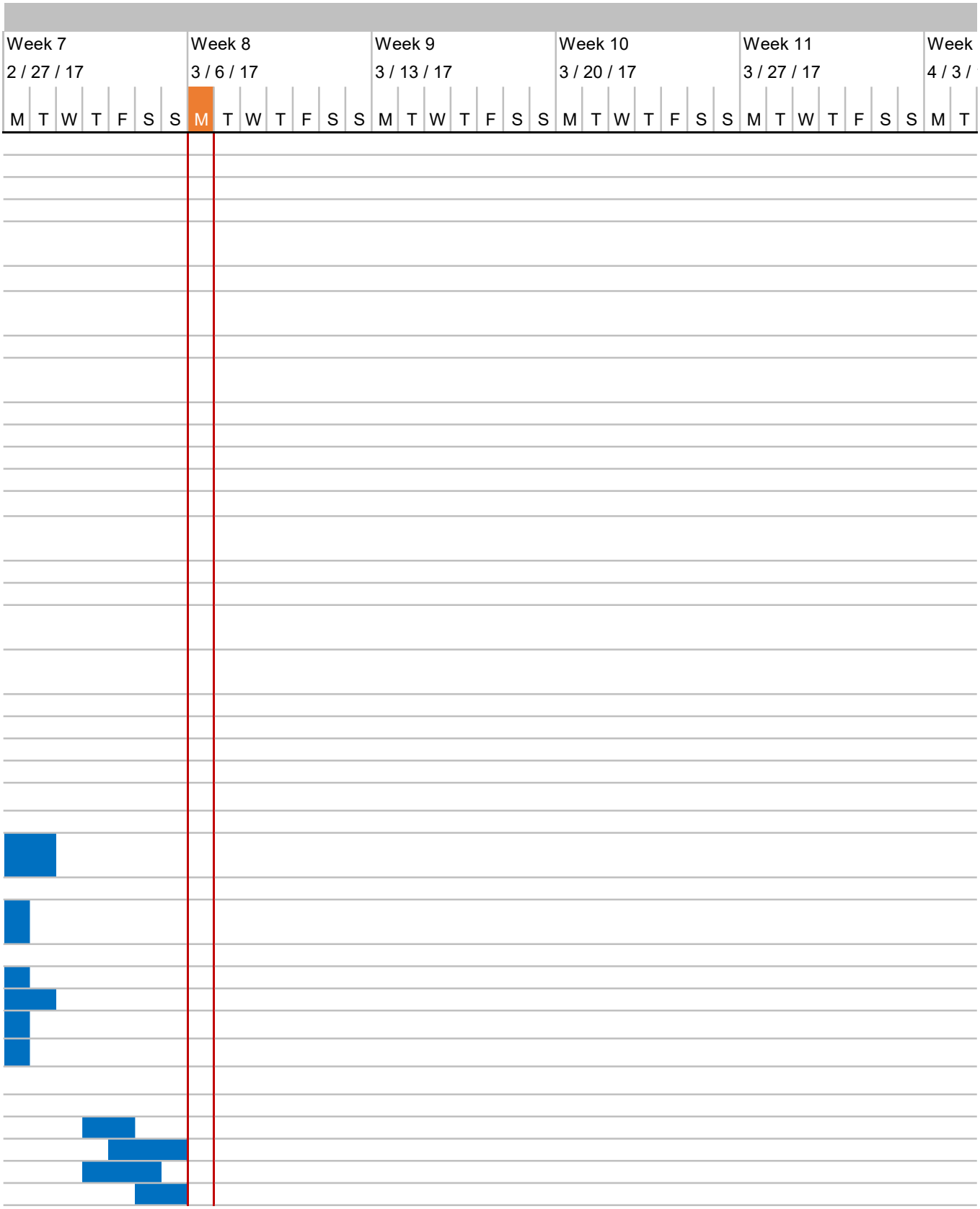
Project Start Date: 1/21/2017 (Saturday)

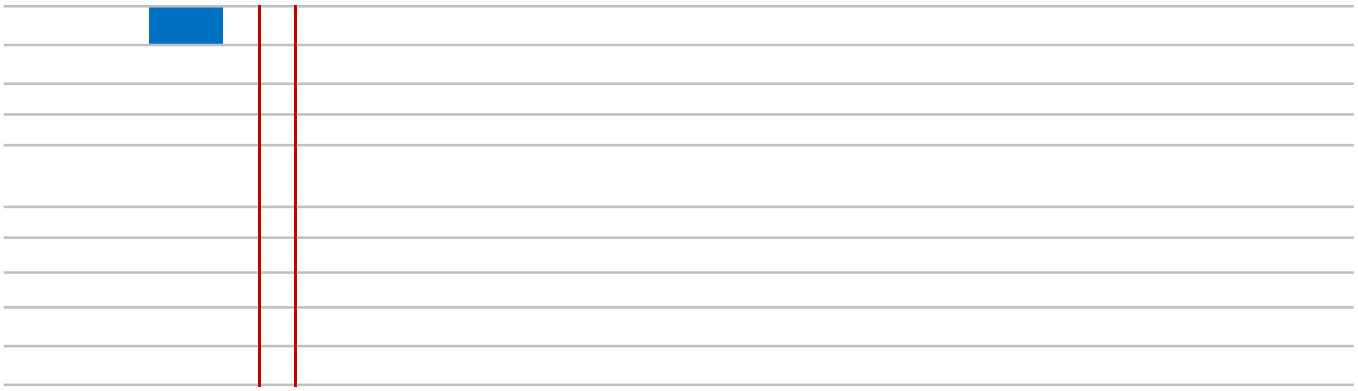
Display Week: 1

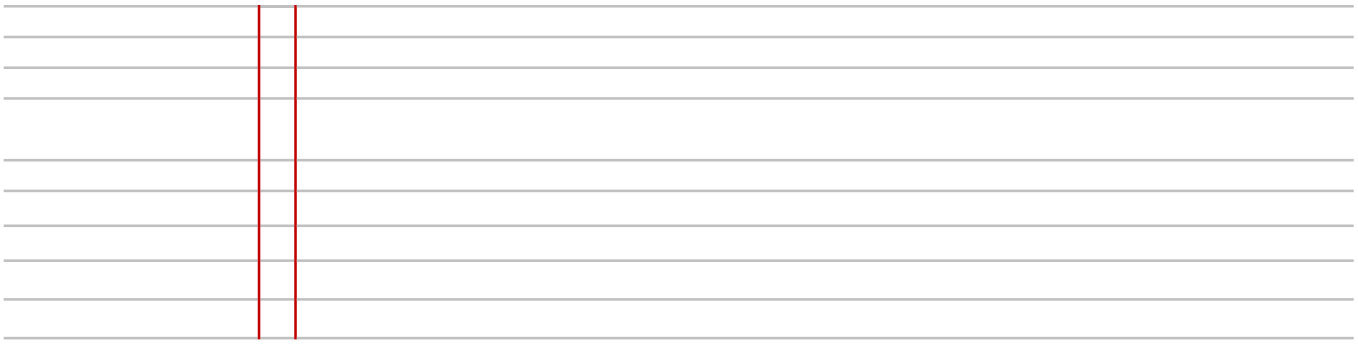
WBS	Task	Lead	Start	End	Cal. Days	% Done	Work Days	Week 1 1 / 16 / 17 M	T	W	T	F
1	**Introduction**											
1.1	Title Page	Mengyuan	Sat 1/21/17	Sat 1/21/17	1	100%	1					
1.2	Resumes	Mengyuan	Sun 1/22/17	Sun 1/22/17	1	100%	1					
1.3	Approved topic	Mengyuan	Mon 1/23/17	Mon 1/23/17	1	100%	1					
1.4	Work Strucutere Document	Mengyuan	Sun 1/22/17	Mon 1/23/17	2	100%	1					
2	**Requirements Elicitation**											
2.1	Problem statement	Mengyaun	Thu 1/26/17	Thu 2/02/17	1	100%	6					
2.2	Requirements Traceabillity Matrix	Mengyaun	Fri 2/03/17	Fri 2/03/17	1	100%	1					
2.3	Gantt Chart	Sungjae	Thu 1/26/17	Thu 1/26/17	1	100%	1					
2.4	Dictionary	Mengyaun	Sat 2/04/17	Sat 2/04/17	1	100%	1					
2.5	Topic rational	Sharon	Thu 1/26/17	Wed 2/01/17	1	100%	1					
2.6	Updated WSD	Mengyaun	Sat 2/04/17	Sun 2/05/17	1	100%	1					
3	**System Analysis & Design**											
3.1	Horizontal prototype	Mengyaun	Wed 2/15/17	Fri 2/17/17	1	100%	3					
3.2	RTM Update	Sungjae	Wed 2/15/17	Thu 2/16/17	1	100%	2					
3.3	Use cases & Interaction Diagram	Mengyaun	Wed 2/15/17	Sun 2/19/17	1	100%	3					
3.4	Function Point Cost Analysis	Sharon	Wed 2/15/17	Sat 2/18/17	1	100%	3					
3.5	Updated WSD	Mengyaun	Wed 2/15/17	Fri 2/17/17	1	100%	3					
3.6	Updated Gantt Chart	Sungjae	Wed 2/15/17	Sun 2/19/17	1	100%	3					
3.7	Dictionary	Hyeun	Wed 2/15/17	Fri 2/17/17	1	100%	3					
3.8	Use cases rational	Sharon	Wed 2/15/17	Fri 2/17/17	1	100%	3					
4	**Object Design**											
4.1	Software architecture used	Hyeun	Sun 2/26/17	Tue 2/28/17	2	100%	2					
4.2	RTM Update	Mengyaun	Fri 2/24/17	Sat 2/25/17	1	100%	1					
4.3	Category Interaction Diagram	Sungjae	Sat 2/25/17	Mon 2/27/17	2	100%	1					
4.4	Design of objects	Jakub	Fri 2/24/17	Sun 2/26/17	2	100%	1					
4.5	Updated WSD	Mengyaun	Sun 2/26/17	Mon 2/27/17	1	100%	1					
4.6	Updated Gantt Chart	Sungjae	Mon 2/27/17	Tue 2/28/17	1	100%	2					
4.7	Dictionary	Mengyaun	Mon 2/27/17	Mon 2/27/17	1	100%	1					
4.8	Object design rational	Sharon	Sun 2/26/17	Mon 2/27/17	2	100%	1					
5	**Rationale**											
5.1	Merge rationale	Sharon	Thu 3/02/17	Fri 3/03/17	1	100%	1					
5.2	RTM Update	Jakub	Fri 3/03/17	Sun 3/05/17	1	100%	2					
5.3	Updated WSD	Mengyaun	Thu 3/02/17	Sat 3/04/17	1	100%	2					
5.4	Updated Gantt Chart	Sungjae	Sat 3/04/17	Sun 3/05/17	1	100%	1					

5.5	Dictionary	Hyeun	Fri 3/03/17	Sat 3/04/17	1	100%	1
6	**Test Document & Code**						
6.1	RTM Update	Jakub			1	0%	0
6.2	Source code & Executable code				1	0%	0
6.3	Test cases				1	0%	0
6.4	Updated WSD	Mengyaun			1	0%	0
6.5	Updated Gantt Chart	Sungjae			1	0%	0
6.6	Dictionary	Hyeun			1	0%	0
6.7	Test cases rational	Sharon			1	0%	0
6.8	COCOMO						
7	**Final Report**						
7.1	Table of Contents				1	0%	0
7.2	Full documentation				1	0%	0
7.3	Function Point Cost Analysis & COCOMO				1	0%	0
7.4	Prototype				1	0%	0
7.5	Project legacy				1	0%	0
7.6	Final WSD	Mengyaun			1	0%	0
7.7	Final Gantt Chart	Sungjae			1	0%	0
7.8	Dictionary	Hyeun			1	0%	0
7.9	Resumes				1	0%	0
7.10.1	User Guide				1	0%	0
7.10.2	Email from GIT				1	0%	0


Week 2
1 / 23 / 17
Week 3
1 / 30 / 17
Week 4
2 / 6 / 17
Week 5
2 / 13 / 17
Week 6
2 / 20 / 17
S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S


Week 7
2 / 27 / 17
Week 8
3 / 6 / 17
Week 9
3 / 13 / 17
Week 10
3 / 20 / 17
Week 11
3 / 27 / 17
Week
4 / 3 /
M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T
M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T





12					Week 13						
17					4 / 10 / 17						
W	T	F	S	S	M	T	W	T	F	S	S