Group5

Flight Management System

CSCI222 System Development

Ruixi He

Zheli Jiang

Junyan Fan

Siyuan Hou

Sandon Joubert

Contents

[**1. Business Case** 3](#_Toc417342340)

[1.1 Overview of the project 3](#_Toc417342341)

[1.2 Stakeholders 3](#_Toc417342342)

[*1.2.1 Who are the stakeholders?* 3](#_Toc417342343)

[*1.2.2 What will this system do for them?* 3](#_Toc417342344)

[1.3 Business Value 4](#_Toc417342345)

[**2. Detailed Plans** 5](#_Toc417342346)

[**3. Risk and Counter Measures** 11](#_Toc417342347)

[**4. Software Requirements Specification** 12](#_Toc417342348)

[4.1 Introduction 12](#_Toc417342349)

[*4.1.1 Purpose* 12](#_Toc417342350)

[*4.1.2 Scope* 12](#_Toc417342351)

[*4.1.3 Definitions, Acronyms, and Abbreviations* 13](#_Toc417342352)

[*4.1.4 References* 13](#_Toc417342353)

[*4.1.5 Overview* 13](#_Toc417342354)

[4.2 Overall Description 13](#_Toc417342355)

[*4.2.1 Product Perspective* 13](#_Toc417342356)

[*4.2.2 Product Functions* 16](#_Toc417342357)

[*4.2.3 User Characteristics* 16](#_Toc417342358)

[*4.2.4 Constraints* 16](#_Toc417342359)

[*4.2.5 Assumptions and Dependencies* 17](#_Toc417342360)

[4.3 Specific Requirements 17](#_Toc417342361)

[*4.3.1 Functional Requirements,* 17](#_Toc417342362)

[*4.3.2 Non-functional Requirements* 18](#_Toc417342363)

[**5. Use Cases** 18](#_Toc417342364)

[**6. Domain Model** 18](#_Toc417342365)

[**7. Meta-report** 19](#_Toc417342366)

[7.1 Tabular Summary of the Group Structure 19](#_Toc417342367)

[7.2 Group Meeting Summary 20](#_Toc417342368)

[*7.2.1 Report of Group Meeting1* 20](#_Toc417342369)

[*7.2.2 Report of Group Meeting2* 21](#_Toc417342370)

[*7.2.3 Report of Group Meeting3* 24](#_Toc417342371)

[*7.2.4 Report of Group Meeting4* 26](#_Toc417342372)

[*7.2.5 Report of Group Meeting5* 28](#_Toc417342373)

[7.3 Individual Work Diaries 28](#_Toc417342374)

[*7.3.1 Ruixi He* 28](#_Toc417342375)

[*7.3.2 Zheli Jiang:* 36](#_Toc417342376)

[*7.3.3 Junyan Fan:* 38](#_Toc417342377)

[*7.3.4 Siyuan Hou:* 38](#_Toc417342378)

[*7.3.5 Sandon Joubert:* 39](#_Toc417342379)

[7.4 Screenshot of Version Control Software (GitHub) 40](#_Toc417342380)

[8. Member Contribution Assessment 41](#_Toc417342381)

[**Appendix** 42](#_Toc417342382)

**1. Business Case**

1.1 Overview of the project

Our project is a flight management system that provides flight services of a major airline. This system is divided into several subsystems that will automate major operations of the airline. Users are divided into several user groups with different privileges.

1.2 Stakeholders

*1.2.1 Who are the stakeholders?*

System Administrator

Manager

Staff

General Public

*1.2.2 What will this system do for them?*

For administrators, our system allows them to manage personnel and they have the greatest power:

* Create user groups
* Promote staff
* Everything a manager or a staff can do

For managers, our system allows them to manage flights:

* Create flight
* Modify flight
* Order and manage services on plane

For staff, they can do modification to flights:

* Switch seats

For public (customer), different types of customer have different rights:

* Agency can have discounts
* Agency can view more specific flight information than regular customers

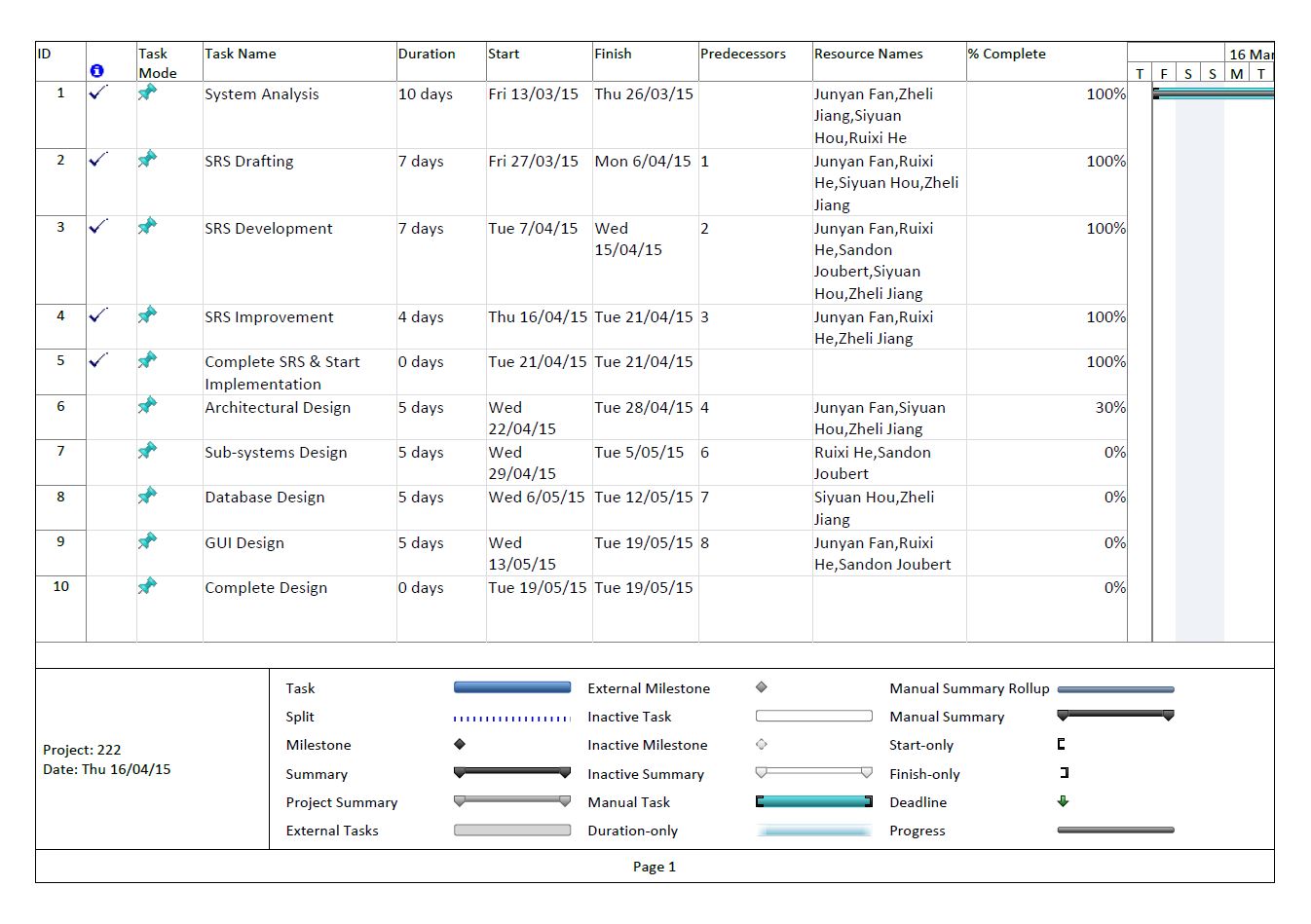
1.3 Business Value

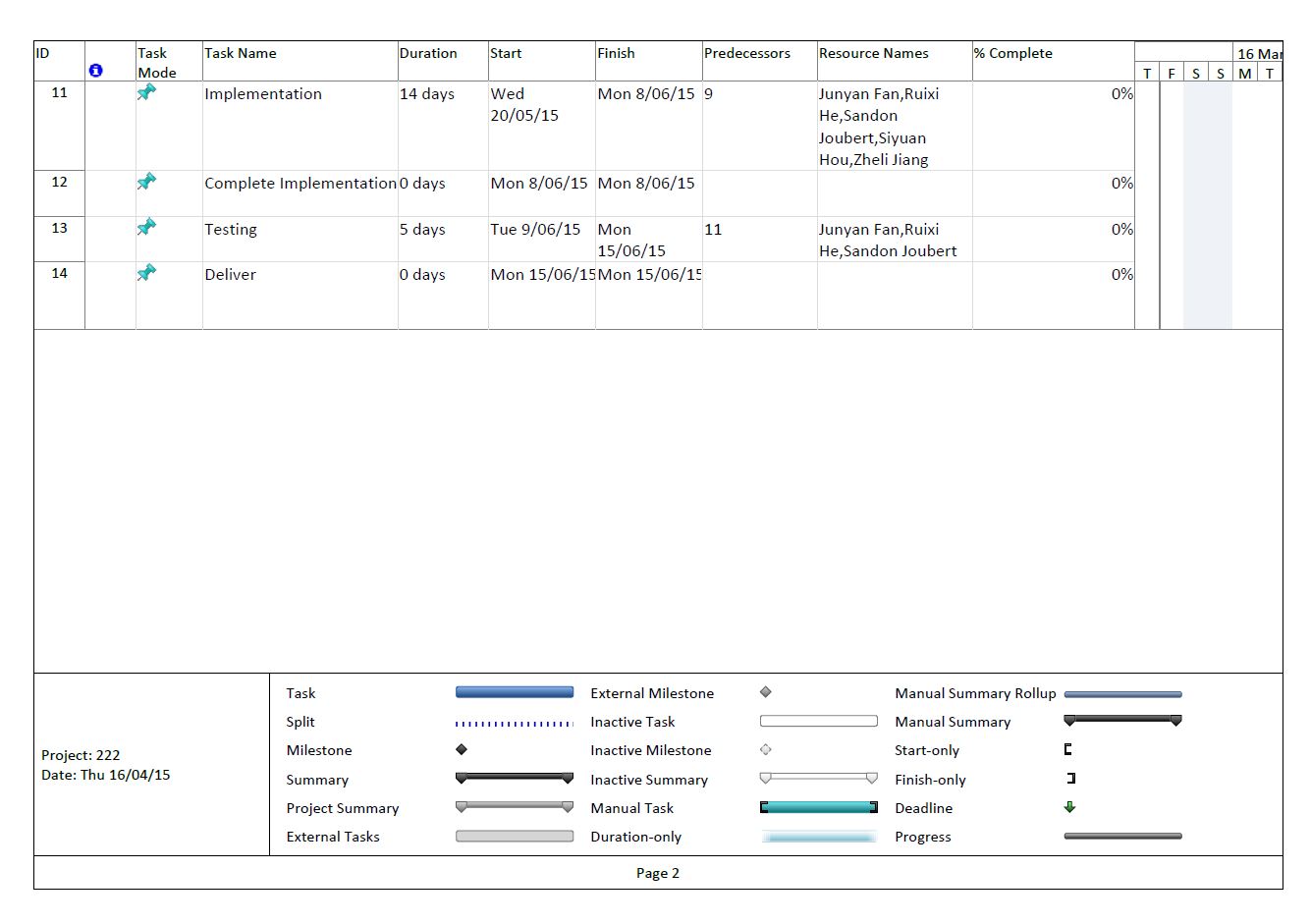
We expect the Flight Management system to reduce labor cost since the current paper-based work requires more personnel. The system is also expected to save time to earn more efficiency. Furthermore, the system should benefit from improved customer satisfaction and increased brand recognition due to its software system.

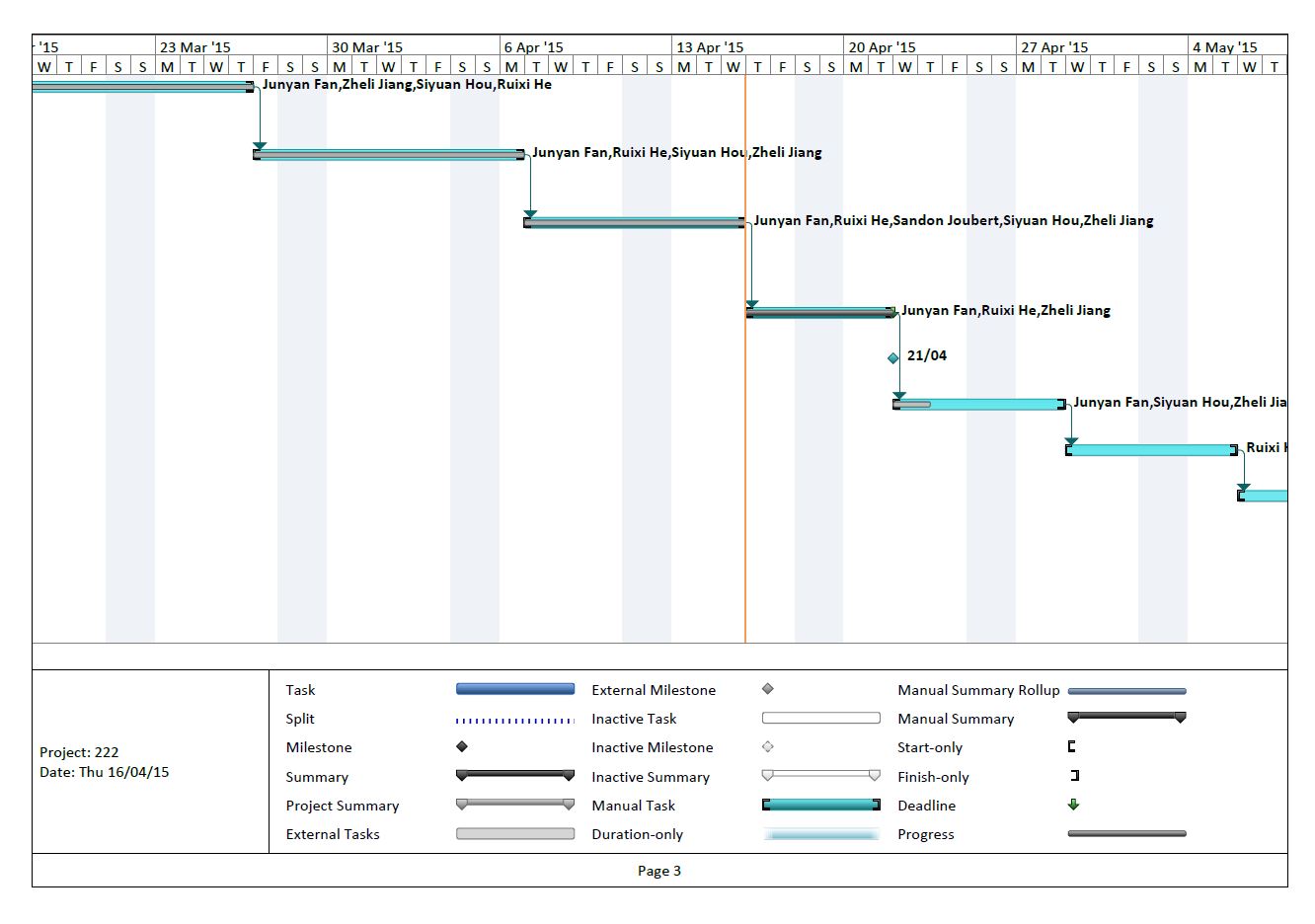
Conservative estimates of tangible value to the company include:

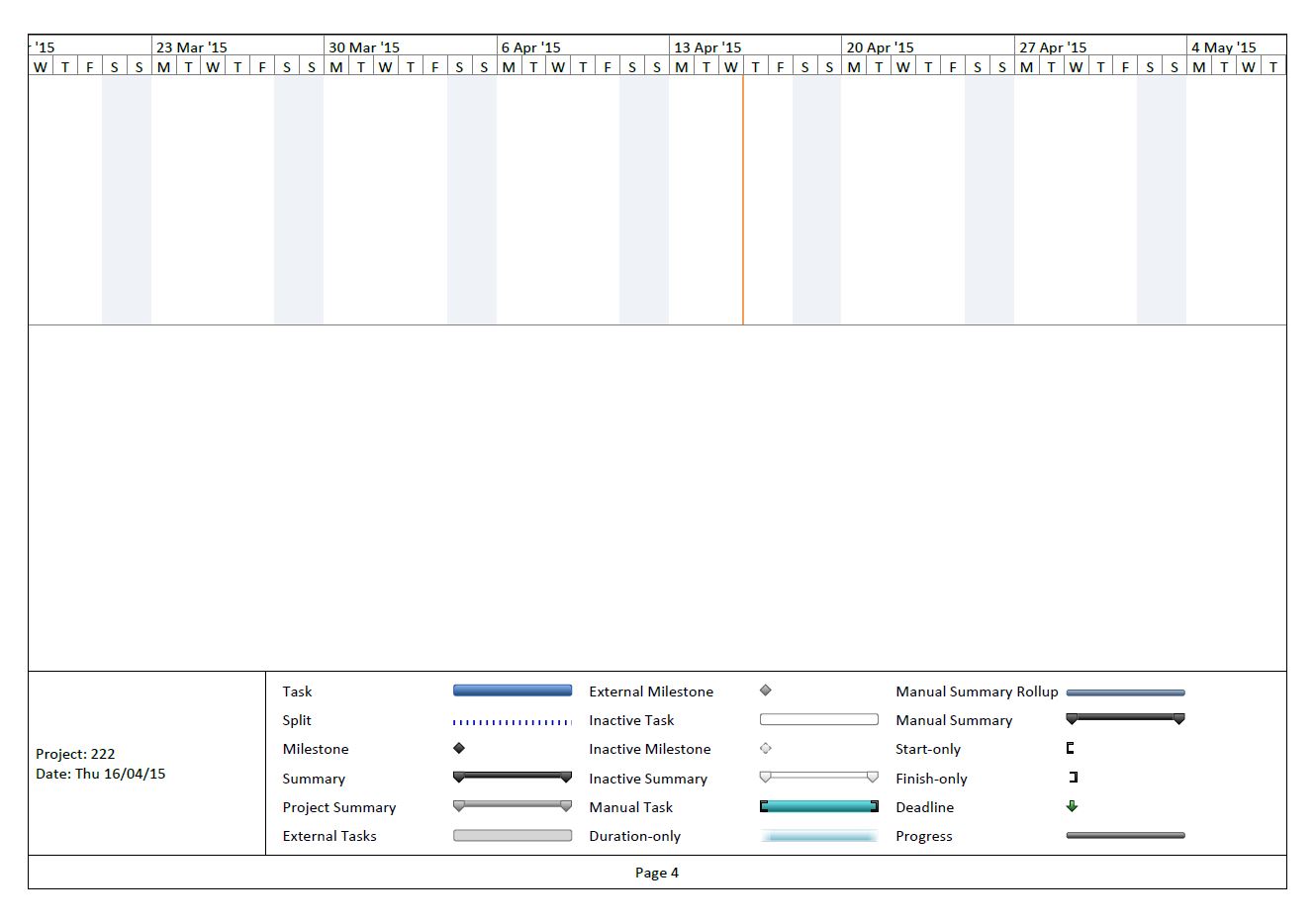
* 30% in sales from new customers
* 40% in sales from existing customers

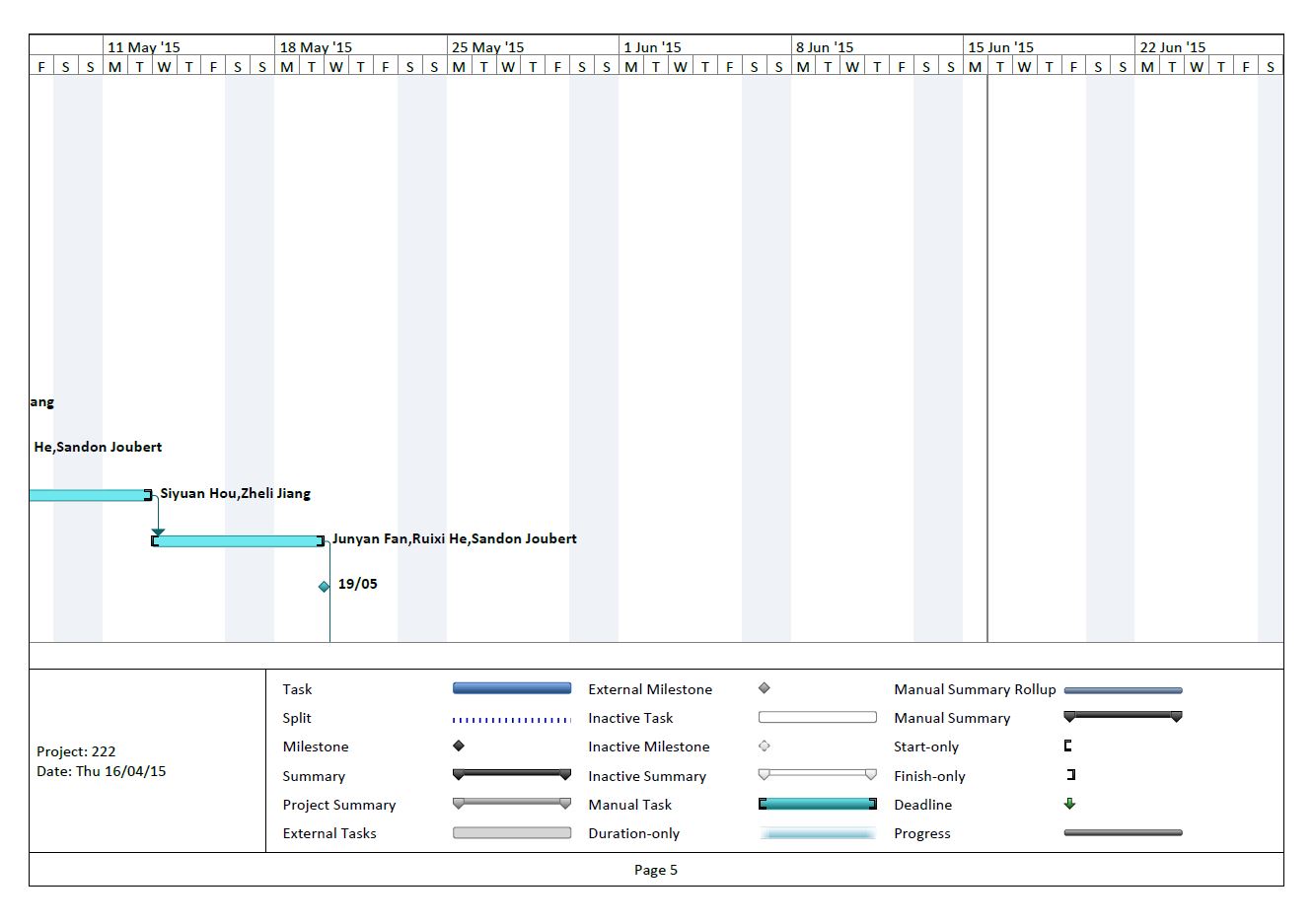
**2. Detailed Plans**

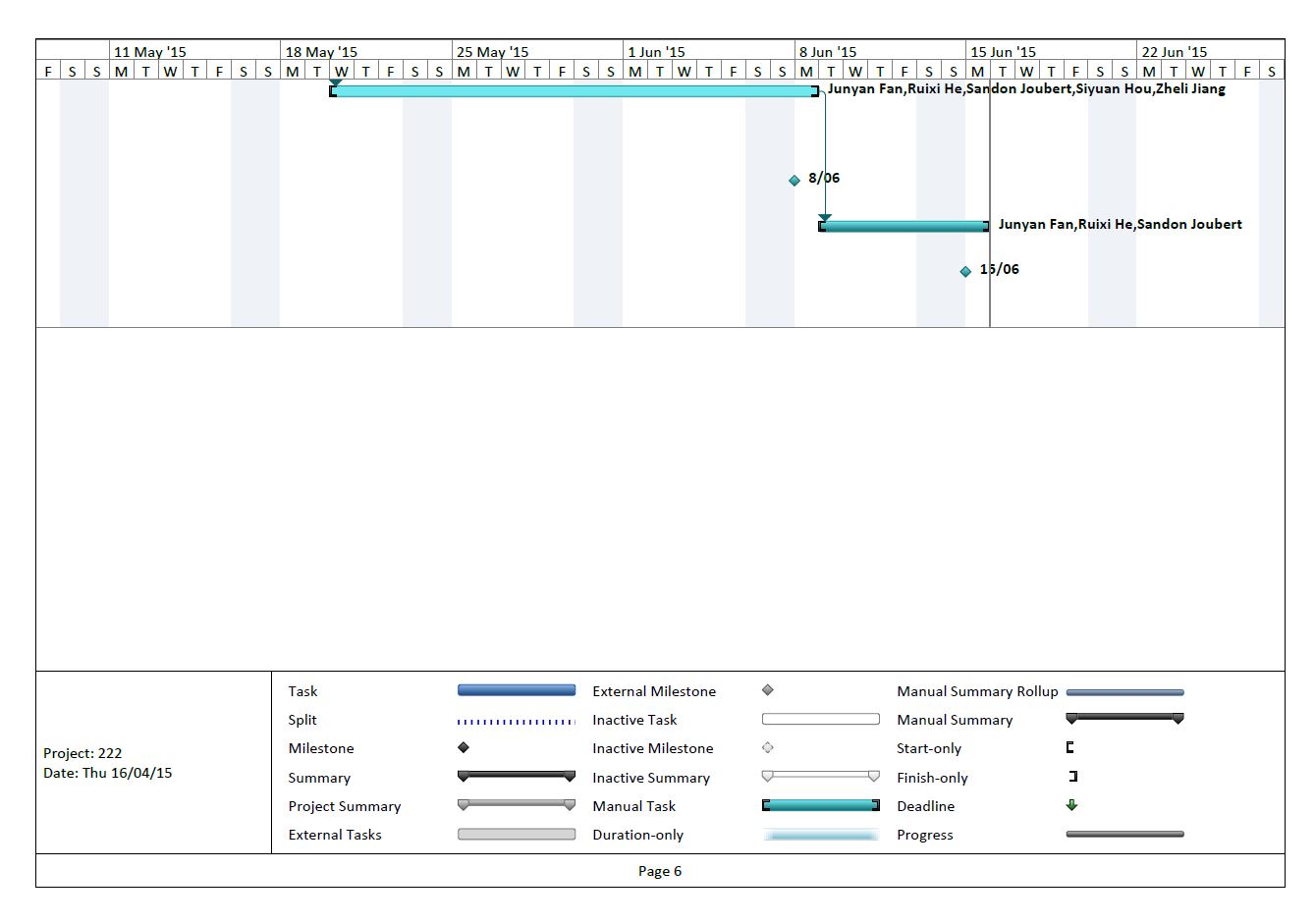












**3. Risk and Counter Measures**

|  |  |
| --- | --- |
| Risk | Counter |
| Using Java programming language for the backend of the Airline Booking System while four members are proficient in Java one has only a little experience with it. This predominately puts the programming workload onto four of the members. | We must assume this risk. Four members should still be enough with the lead programmer still doing a majority of the work. The fifth still knows enough of Java to understand the code. |
| Language barrier can be a problem in communicating. Four members first language is Chinese mandarin / Cantonese whilst the fifths is English. | We must assume this risk with main communication being in English. |
| We only have about 10 weeks to complete this Airline Booking System, with a deadline for a large project like this and all of us having other commitments including other university work and other time consuming tasks outside of university we may struggle to complete the program on time. | By meeting every week and discussing what we will do in the following week as well as keeping diaries for our individual work should hopefully keep us from falling behind. |
| Incomplete or unwritten information and plans. | Avoiding this risk is priority as it will lead to a harder development phase. Using Q & A sessions with client productively will allow up to get all the information we need. |
| Plan carrying out by the team members who have not yet worked in the same or similar projects. | This risk must be assumed as we cannot gain experience on this sort of project over night. |
| Frequent and large change of requirements. | As we continue to ask the clients our understanding of the requirements may change. As long as we can get them finished before the development phase then we will avoid this risk. |
| Unclear requirements. | Speaking to the client can help us avoid and ambiguity in the project requirements. |

**4. Software Requirements Specification**

|  |  |
| --- | --- |
| Flight Management System | Version |
| Software Requirements Specification | **Date:** |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Author |
|  | 1.0 | First SRS |  |
|  | 1.1 |  |  |
|  |  |  |  |

4.1 Introduction

*4.1.1 Purpose*

The purpose of this document is to describe the specifications on the external behaviors of a flight management system. Besides, it also documents nonfunctional requirements, design constraints and other factors necessary to provide a complete and comprehensive understanding of the system.

The intended audience includes the potential users of the system and software development team.

*4.1.2 Scope*

The software system to be produced is a web-based flight management system.

Our system aims to serve four groups of user:

|  |  |
| --- | --- |
| Roles | Purpose |
| Administrator | Maintain smooth operations of the system |
| Manager | Manage flight resources |
| Staff | Interact with manager and customer |
| General Public | Use the system to book flights |

Our system has four subsystems:

* A Reservation System that manages (e.g. add, change, and modify) all flights reservations, seat selection, ticketing, flight availability, flight details, rates and conditions.
* A Profile Subsystem that manages individual passengers and travel agency profiles.
* A Service Subsystem that manages in-flight services such as food and drinks.
* A Reporting Subsystem to generate various summary report such as: Passengers Report for a day including the occupancy rate etc., various summary Cashier Reports like total revenue for the day, or weeks etc., Monthly Booking Activity Summary, Daily Booking Activity Summary, etc.

Different user groups have different powers. For example, agency is provided with discounts. Also, agency is able to see specific flight information that a normal customer cannot see despite the fact that agency belongs to general public group.

*4.1.3 Definitions, Acronyms, and Abbreviations*

Refer to Appendix.

*4.1.4 References*

* **Rational Unified Process**, SRS template (upedu\_srs.doc), COSC2151 Final Year Software Engineering Project , RMIT International University Vietnam, 2004
* Sample - SoftwareRequirementsSpecification.pdf

*4.1.5 Overview*

The rest of this Software Requirements Specification is divided into two main sections:

* The Overall Description (section 4.2) describes the general factors that affect the system and its requirements.
* The Specific Requirements (section 4.3) contains all software requirements that the system must meet in order to satisfy the needs of customer.

4.2 Overall Description

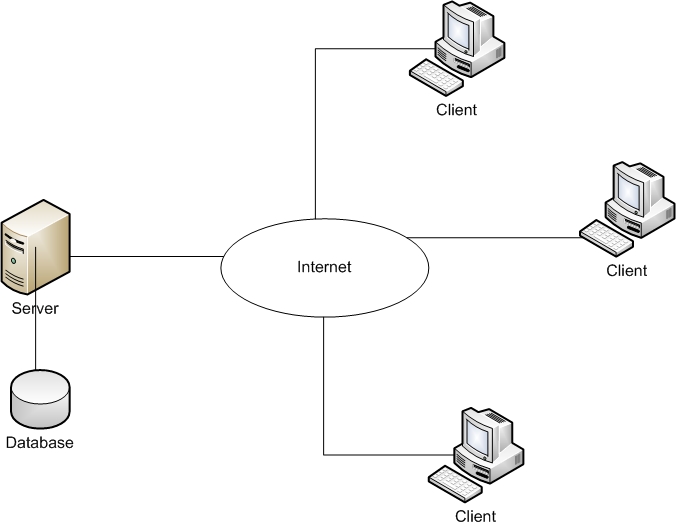
*4.2.1 Product Perspective*

Currently, some of the activities in some flight companies are carried out and managed manually, which makes the process of booking flights difficult to perform and requires an incredible amount of time and efforts.

Therefore, building this system appears necessary and essential. Flight Management System aims to be a perfect tool for achieving goals such as managing flights.

*4.2.1.1 System Interfaces*

The Flight Management System is a web-based application that allows users to interact with the system via internet or intranet.



The clients can simultaneously log into the system from any personal computer that has internet services, and then the clients can interact with the system to do reservation, view profiles, check flight information, etc.

The server takes and processes the clients' request and then retrieves the database and return the result to the clients.

The database is used to store the entire user's profiles, flight information, in-flight services.

*4.2.1.2 User Interfaces*

The user interface provided by the system must be GUI and must be accessible through any web-browser. Depending on the different privileges of the user, they will have slightly different interfaces and the access to slightly different functionalities of the system. For instance, admin has the highest privilege, which means he/she has the right to access all the functionalities provided by the system.

The database should be powered by MySQL server and performed indirectly through the GUIs provided by the system.

*4.2.1.3 Hardware Interfaces*

All components must be able to execute on any computer, which runs on any OS operation system with Java Running Environment.

*4.2.1.4 Software Interfaces*

4.2.1.4.1 User Interface

The user interacts with the system through web browser.

The system supports any web browser.

*4.2.1.5 Communication Interfaces*

The client machines must communicate with the Web Server over TCP/IP connection.

We might have the Web Server and the Database Server are located on different servers.

*4.2.1.6 Memory Constraints*

The client machine must be able to operate within 128MB minimum (including memory for browser)

The Web Server and the Database Server must be able to operate within 512MB minimum.

*4.2.1.7 Operations*

The Flight Management System needs to be easy to use for all users. For customers and agencies, we designed the GUI of this system based on the common sense, which means when the first time the customers or agencies use the system, they will know what 's this button does, and what the next screen will be. And the admin, managers and staff will also be very easy to get used to the functions of the system.

The server installation and maintenance should also be simple for the admin, and we will provide instruction material that teaches the admin how to install and maintain the system.

Backup and recovery must be specified in case of the network failure, database failure, out of power etc.

*4.2.2 Product Functions*

For administrators, our system allows them to manage personnel and they have the greatest power:

* Create user groups
* Promote staff
* Everything a manager or a staff can do

For managers, our system allows them to manage flights:

* Create flight
* Modify flight
* Order and manage services on plane

For staff, they can do modification to flights:

* Switch seats

For public (customer), different types of customer have different rights:

* Agency can have discounts
* Agency can view more specific flight information than regular customers

All functionalities of the system are built on the needs of any flight management system so that this system can make all activities and tasks carried out by users easier and more convenient.

*4.2.3 User Characteristics*

The users of Flight Management System include system administrators, managers, staff and general public.

* Administrators have strong knowledge on networks and web applications to be able to install and maintain the system.
* Managers have good knowledge on web applications and flight information.
* Staff has solid understanding of flights and good interaction skills.
* General public are people who have enough understanding on the use of Internet to use the system.

*4.2.4 Constraints*

The system should strictly obey and satisfy the following constraints:

* Authentication security: the system should enforce user authentication security
* Access control: the system must provide appropriate access right and user interface to each type of user.
* Backup and recovery: the backup and recovery of all the system’s database must be easy to perform to prevent databases from corruption and loss risks
* Integrity control: since the system consists of many databases that are correlated with each other, integrity among these databases must be strictly maintained

*4.2.5 Assumptions and Dependencies*

The following assumptions and dependencies for the system are stated:

* All potential users must have access to Internet.
* All potential users must have a valid email address.

4.3 Specific Requirements

Each requirement (either functional or non-functional requirements) of Flight Management System is ranked based on its level of priority:

* Critical: highest priority level, these requirements are the core functionalities of Flight Management System and must be firstly implemented
* Essential: second priority level, these requirements are the important functionalities of Flight Management System and should be implemented when all the Critical requirements have been finished
* Desirable: medium priority level, these requirements are the necessary functionalities of Flight Management System and should be covered when Critical and Essential requirements have implemented
* Optional: lowest priority level, these requirements are the enhanced functionalities of Flight Management System and should be considered only when all Critical, Essential and Desirable requirements are completed

*4.3.1 Functional Requirements,*

*4.3.1.1 Reservation Subsystem*

This section includes all the functionalities that Flight Management System provides to System Admin, Flight Manager, Staff, and Customer

*4.3.1.2 Profile Subsystem*

*4.3.1.3 Service Subsystem*

*4.3.1.4 Reporting Subsystem*

*4.3.2 Non-functional Requirements*

**5. Use Cases**

**6. Domain Model**

**7. Meta-report**

7.1 Tabular Summary of the Group Structure

|  |  |  |
| --- | --- | --- |
| Group Members | Roles | Artifacts Delivered |
| Ruixi He | Project Manager  Business-Process Analyst  Requirement Specifier  System Analyst | * Distribute work * Make project plan * Finish business case * Create GitHub repository * Provide functional requirements of Staff * Provide use cases of Staff and detail description * Record meeting agendas * Finish individual work diary |
| Siyuan Hou | Lead Programmer  Software Engineer | * Class diagram Architectural design * Individual work diary |
| Zheli Jiang | System Analyst  Business Designer  Requirement Specifier | * Detailed plan * Gantt Chart * Meeting agendas * Admin-related use cases * Admin-related use case descriptions * Admin-related functional requirements * Individual work diary |
| Junyan Fan | System Analyst  Business Designer  Requirement Specifier | * Manager use cases * Manager use case descriptions * Manager functional requirements * Individual work diary |
| Sandon Joubert | System Analyst  Business Designer  Requirement Specifier | * Risk and counter measures * General public use cases * General public use case descriptions * General public functional requirements * Individual work diary |

7.2 Group Meeting Summary

*7.2.1 Report of Group Meeting1*

Date: 13/03/2015

Time: 2:30pm - 4:30pm

Meeting called by Attendance: Ruixi He - Project Manager,

Zheli Jiang,

Junyan Fan,

Siyuan Hou

Place: #204 Library

Preparation:

-None

Meeting Agendas:

-Introduction: The group members introduced themselves to each other and shared thoughts about on this project.

-We spent 15 minutes reading specification and we discussed about what the system should look like (implementation) and what should our report includes (documentation).

-Implementation: We are to develop a flight management system that has several subsystems that we need to implement respectively. This Flight Management System includes four major subsystems:

* + Reservation Subsystem that can add, change and modify all flight reservations, seat selection, ticketing, flight availability, flight details, rates and conditions.
  + Profile Subsystem that manages individual passengers and travel agency profiles.
  + Service Subsystem that manages in-flight services.
  + Reporting Subsystem to generate various summary reports.

-Documentation: The documentation includes two major parts. This is the first time we encounter this sort of documentation so we are unfamiliar with the structure of the report. Hence we decided to have some research on SRS.

-We drew a user group architecture to show the client.

-In order to fully understand the objectives of this assignment, we clarified some questions as listed below:

* 1. Can we build the system based on website and java?
  2. What kind of users to use this system?
  3. What's this system for, a company or the platform?
  4. Reservation phase: Does customer have to login to do reservation?
  5. What's different between normal customer and agency, do we need to think about how the agency uses this system?
  6. Service: What's the service system for, for customer or the management?
  7. Reporting: What kind of data should we provide different users? (Different data?)

-Risks:

-Initially the group had only four members. Furthermore, none of the members speak English as first language. So documentation appeared challenging at first.

-One of our members had no experience coding Java, which could be a potential risk.

-Roles distribution:

-Ruixi He was assigned project manager because he did this subject twice so he was very experienced. Also he is very talented in leading.

-Siyuan Hou was assigned lead programmer because he had most experience in coding (JAVA) and he had a good understanding of this system structure.

-Zheli Jiang was assigned documentation manager.

-Junyan Fan was assigned design manager.

Work to do:

-Do some research on SRS.

-Consult the clients.

*7.2.2 Report of Group Meeting2*

Date: 20/03/2015

Time: 2:30pm - 4:30pm

Meeting called by Attendance: Ruixi He - Project Manager,

Zheli Jiang,

Junyan Fan,

Siyuan Hou

Place: #204 Library

Preparation:

-Research on SRS.

-Answers to questions:

1.Can we build the system base on website and java?

Yes.

2.What kind of users to use this system?

User who use the system are: General (Public), Administrator (super user they can do everything), Staff (they help users do things), Managers (a flight manager and aa service manager, they set prices and stuff)

3.What's this system for, a company or the platform?

Just one airline.

4.Reservation phase: Does customer have to login to do reservation?

Customer does not have to log in can have separate member and customer.

5.What's different between normal customer and agency, do we need to think about how the agency use this system?

Difference between agent and customer, travel agents get discount for customer.

6.Service: What's the service system for, for customer or the management?

When a customer books flight they can choose their food and drink in booking process.

7.Reporting: What kind of data should we provide different users? (different data?)

Various reports would like to see travel agent reports like how many people they get through them.

Meeting Agendas:

-A new member joined our team who speaks English as first language.

-We decided to use RUP as our development model.

-Sort out the stakeholders to elicit requirements.

-We improved our user group architecture after consultation with the client.

-General public

-Customer

-Normal Customer

-Member

-Agency

-Administrator

-Staff

-Manager

-Flight Manager

-Service Manager

Work to do:

-Zheli Jiang: Admin requirements

-Ruixi He: Staff requirements

-Junyan Fan: Manager requirements

-Sandon Joubert: General public requirements

-Siyuan Hou: System Analysis

*7.2.3 Report of Group Meeting3*

Date: 27/03/2015

Time: 2:30pm - 4:30pm

Meeting called by Attendance: Ruixi He - Project Manager,

Zheli Jiang,

Junyan Fan,

Siyuan Hou,

Sandon Joubert

Place: #204 Library

Preparation:

-Zheli Jiang: Admin requirements

-Ruixi He: Staff requirements

-Junyan Fan: Manager requirements

-Sandon Joubert: General public requirements

-Siyuan Hou: System Analysis

Meeting Agendas:

-This is the first time Sandon joined our meeting so we updated him for our group progress.

-Gather and discuss the requirements.

1.Admin:  
1. Grant different access levels(admin, staff, manager, general public)  
2. Promote staff to manager  
3. Enable/Disable accounts  
4. View all profiles and flight info  
5. Change flight details(departure time,etc.)

2.Manager:  
Flight Manager:  
1. Create a flight(dep/arr, time, plane type, price, flight number, etc.)  
2. Change flight details(departure time,etc.).  
Service Manager:  
1. Order meals

3.Staff:  
1. Switch seats  
2. Search for users involved in this flight and filter the results  
Search for flight info(Staff can see more specific info than customers)

4.General Public:  
Customer:  
Non-member:  
1. Enter details during first purchase and automatically become registered  
Member:  
1. Discount  
Agency:  
1. Further discount (than members)  
2. View profiles of users who bought tickets through this agency  
3. Search for flight info(pretty much the same as staff)  
4. Save user profiles(list)

Report:(Reports of numbered items will be generated)  
Admin:  
1. Financing;  
2. Staff;  
3. Customer;  
4. Sales volume.

Manager:  
Flight Manager:  
1. Flight (those under his/her management)  
Service Manager:  
1. Meal

Staff:  
1. Flight and meal

General Public:  
Customer:  
1. Purchase history  
Agency:  
1. User profiles who bought tickets through this agency

-Siyuan Hou provided a sketch of class diagram.

Work to do:

-Develop the requirements.

-Improve class diagram.

-Create use cases.

*7.2.4 Report of Group Meeting4*

Date: 03/04/2015

Time: 2:30pm - 4:30pm

Meeting called by Attendance: Ruixi He - Project Manager,

Zheli Jiang,

Junyan Fan,

Siyuan Hou,

Sandon Joubert

Place: #204 Library

Preparation:

-All use cases and class diagram.

-All improved functional requirements,

Meeting Agendas:

-Discuss the use cases.

-We created scenarios which include both successful and unsuccessful situations according to our individual actors and use cases.

-Ruixi He had another assignment due so he failed to complete his use cases. Therefore he had to finish his use cases during the meeting.

-We decided to use GitHub as our version control software. (Before we only used Facebook to contact and share files.) We all registered and shared our files on GitHub.

-Siyuan Hou proposed an idea that we can have an interface with access level passed in as an argument to identify user groups and which functions they have access to.

[GRAPH]

Work to do:

-Ruixi He: Business Case

-Zheli Jiang: Detailed Plan

-Junyan Fan: Use case development

-Siyuan Hou: Class diagram improvement

-Sandon Joubert: Risk and counter measure

*7.2.5 Report of Group Meeting5*

Date: 17/04/2015

Time: 2:30pm - 4:30pm

Meeting called by Attendance: Ruixi He - Project Manager,

Zheli Jiang,

Junyan Fan,

Siyuan Hou,

Sandon Joubert

Place: #204 Library

Preparation:

-Ruixi He: Business Case

-Zheli Jiang: Detailed Plan

-Junyan Fan: Use case development

-Siyuan Hou: Class diagram improvement

-Sandon Joubert: Risk and counter measure

Meeting Agendas:

-We combined the use cases and requirements.

-We has a discussion on meta-report.

Work to do:

-Finish SRS.

7.3 Individual Work Diaries

*7.3.1 Ruixi He*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Week | Start Date | Date Planed to Finish | Tasks | Description | Completed Date |
| 2 | 13/03/2015 | 13/03/2015 | Read project specification | We spent 15 minutes to go through the project specification. | 13/03/2015 |
|  | 13/03/2015 | 13/03/2015 | Discussion about the project | We discussed and wrote down what kind of system should we build, and what should the report have. Then we got that we need to build a flight management system which include four major subsystem: Reservation Subsystem (manages all fight reservations), Profile Subsystem (manages users profile), Service Subsystem (manages in-flight service) and Reporting Subsystem (generate summary reports). | 13/03/2015 |
|  | 13/03/2015 | 13/03/2015 | Prepare questions to ask client | After we need to prepare some questions to ask the client to elicit the requirements of the system. | 13/03/2015 |
|  | 13/03/2015 | 13/03/2015 | Define risks | We defined the potential risks not only about the creation of the system, but also our team. | 13/03/2015 |
|  | 13/03/2015 | 13/03/2015 | Assign roles to each group member | I was at the role as a leader at this time because I have more experience, Siyuan Hou was assigned lead programmer because he had most experience in coding (JAVA) and he had a good understanding of this system structure,  Zheli Jiang was assigned documentation manager,and  Junyan Fan was assigned design manager. | 13/03/2015 |
|  | 13/03/2015 | 20/03/2015 | Research about SRS | From the meeting till to next meeting, we need to do some research and understand what is SRS and how to make a good SRS | 15/03/2015 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Week | Start Date | Date Planed to Finish | Tasks | Description | Completed Date |
| 3 | 20/03/2015 | 20/03/2015 | Discussion about the team structure | Because we got a new group member, Sandon, from this week lab, so we need to reassign the work again. And Sandon had thing to do, so he did not attend this week’s meeting. | 20/03/2015 |
|  | 20/03/2015 | 20/03/2015 | SRS | We talk about what we got from the research, and create the format of the SRS. | 20/03/2015 |
|  | 20/03/2015 | 20/03/2015 | Discuss about the answers we got from last client meeting | From the answers we got from last client meeting, we decide to build this system based on website and Java as background. We also correct the misunderstands that we had when we go through the specification, this is very helpful to the analyse of the requirements. | 20/03/2015 |
|  | 20/03/2015 | 20/03/2015 | Define stakeholders | We defined the stakeholders and potential of the system, and we got four major users group: Admin, Manager, Staff and General Public. | 20/03/2015 |
|  | 20/03/2015 | 27/03/2015 | Assign work to do before next meeting | We need to create the first draft of the requirements:  Zheli Jiang: Admin requirements,  Ruixi He: Staff requirements,  Junyan Fan: Manager requirements,  Sandon Joubert: General public requirements,  Siyuan Hou: System Analysis | 27/03/2015 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Week | Start Date | Date Planed to Finish | Tasks | Description | Completed Date |
| 4 | 27/03/2015 | 27/03/2015 | Review before meeting | This is the first time Sandon joined in the group meeting, so we need to tell him the whole progress of project and what we had done till now. | 27/03/2015 |
|  | 27/03/2015 | 27/03/2015 | Analyse the requirements | Each group member presented their part of the requirements, and we need to find whether there are faults or missing. | 27/03/2015 |
|  | 27/03/2015 | 27/03/2015 | System structure | Our lead programmer showed us the system structure by the present the Domain Model. | 27/03/2015 |
|  | 27/03/2015 | 03/04/2015 | Assign work to do before next meeting | We need to continue the development of requirements, and also Siyuan Hou still needs to do more works to improve the class diagram.  And the rest members need to make the draft of the use cases. And also prepare the questions to ask the client on next lab. | 03/04/2015 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Week | Start Date | Date Planed to Finish | Tasks | Description | Completed Date |
| 5 | 03/04/2015 | 03/04/2015 | Review before meeting | From the meeting of the client on last lab, we showed them the draft of use cases that we got, and the client pointed out the question we had, like the non-member cannot book the flight until he/she register an account in the system, and the use cases of service manager should be more specific. | 03/04/2015 |
|  | 03/04/2015 | 03/04/2015 | Make scenarios of the use cases | Each group member present two scenarios which include both successful and unsuccessful situations. | 03/04/2015 |
|  | 03/04/2015 | 03/04/2015 | Version control system (GitHub) | We prefer to use GitHub as our version control system, I created a repository (CSCI222) and ask each members to share their file on the repository. | 03/04/2015 |
|  | 03/04/2015 | 03/04/2015 | System Structure | Siyuan Hou proposed an idea that we can have an interface with access level passed in as an argument to identify user groups and which functions they have access to. | 03/04/2015 |
|  | 03/04/2015 | 03/04/2015 | Assign work to do before next meeting | Next week should be the Easter break, which means we need to do more work during the break, so each members should keep developing the functional requirements and also finish the report, and here is the plan:  -Ruixi He: Business Case  -Zheli Jiang: Detailed Plan  -Junyan Fan: Use case development  -Siyuan Hou: Class diagram improvement    -Sandon Joubert: Risk and counter measure | 17/04/2015 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Week | Start Date | Date Planed to Finish | Tasks | Description | Completed Date |
| 6 | 17/04/2015 | 17/04/2015 | Review before meeting | Each member showed their work together. | 17/04/2015 |
|  | 17/04/2015 | 17/04/2015 | Combine works together | Combine all the requirements and use cases together, and find whether there are any requirements or use cases missing. | 17/04/2015 |
|  | 17/04/2015 | 17/04/2015 | Meta-report | Create the roles of the group, and combine each meeting record to form the summary of the meeting report, each member can finish their work diaries base on the meeting report. | 17/04/2015 |
|  | 17/04/2015 | 17/04/2015 | First draft of project report | Group each thing to form the first draft of report, and we need to do more work to complete the final report before the due day. | 21/04/2015 |

*7.3.2 Zheli Jiang:*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| week | Start Date | Date Planned to Finish | Tasks | Complete Date |
| 2 | 13/03/2015 | 14/03/2015 |  Sort out basic functional requirements. | 14/03/2015 |
|  | 14/03/2015 | 16/03/2015 |  Draw up a user group structure. | 16/03/2015 |
|  | 16/03/2015 | 18/03/2015 |  Clarify some questions related to specification. | 18/03/2015 |
|  | 18/03/2015 | 19/03/2015 |  Record meeting agendas. | 19/03/2015 |
|  | 19/03/2015 | 20/03/2015 |  Research on SRS. | 20/03/2015 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| week | Start Date | Date Planned to Finish | Tasks | Complete Date |
| 3 | 20/03/2015 | 22/03/2015 |  Improve the user group structure. | 22/03/2015 |
|  | 22/03/2015 | 25/03/2015 |  Work out administrator requirements | 25/03/2015 |
|  | 25/03/2015 | 25/03/2015 |  Consult with client. | 25/03/2015 |
|  | 25/03/2015 | 27/03/2015 |  Work out who our stakeholders could be. | 27/03/2015 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| week | Start Date | Date Planned to Finish | Tasks | Complete Date |
| 4 | 27/03/2015 | 27/03/2015 |  Discussed requirements with members.   Gather the requirements. | 27/03/2015 |
|  | 27/03/2015 | 31/03/2015 |  Improve class diagram. | 31/03/2015 |
|  | 31/03/2015 | 03/04/2015 |  Create administrator use cases. | 03/04/2015 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| week | Start Date | Date Planned to Finish | Tasks | Complete Date |
| 5 | 10/04/2015 | 10/04/2015 |  Do some research on GitHub and decide to use it as version control software. | 10/04/2015 |
|  | 10/04/2015 | 11/04/2015 |  Upload files to GitHub. | 11/04/2015 |
|  | 11/04/2015 | 17/04/2015 |  Complete business case. | 17/04/2015 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| week | Start Date | Date Planned to Finish | Tasks | Complete Date |
| 6 | 17/04/2015 | 18/04/2015 |  Finish meta-report. | 18/04/2015 |
|  | 18/04/2015 | 19/04/2015 |  Finish individual work-diary. | 19/04/2015 |
|  | 19/04/2015 | 20/04/2015 |  Tidy Up SRS. | 20/04/2015 |

*7.3.3 Junyan Fan:*

*7.3.4 Siyuan Hou:*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| week | Start Date | Date Planned to Finish | Tasks | Complete Date |
| 2 | 13/03/2015 | 14/03/2015 | discuss basic system purpose | 14/03/2015 |
|  | 14/03/2015 | 16/03/2015 | decide role of group member | 16/03/2015 |
|  | 16/03/2015 | 18/03/2015 | discuss unknown problem | 18/03/2015 |
|  | 18/03/2015 | 19/03/2015 | record meeting content. | 19/03/2015 |
|  | 19/03/2015 | 20/03/2015 | did some research about java and C++ | 20/03/2015 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| week | Start Date | Date Planned to Finish | Tasks | Complete Date |
| 3 | 20/03/2015 | 22/03/2015 | decide system language | 22/03/2015 |
|  | 22/03/2015 | 25/03/2015 | discuss administrator requirements | 25/03/2015 |
|  | 25/03/2015 | 25/03/2015 | clear special requirements with client. | 25/03/2015 |
|  | 25/03/2015 | 27/03/2015 | research features about java | 27/03/2015 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| week | Start Date | Date Planned to Finish | Tasks | Complete Date |
| 4 | 27/03/2015 | 27/03/2015 | implement system structure | 27/03/2015 |
|  | 27/03/2015 | 31/03/2015 | implement class diagram. | 31/03/2015 |
|  | 31/03/2015 | 03/04/2015 | discuss sub-system, and get new idea about system implement | 03/04/2015 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| week | Start Date | Date Planned to Finish | Tasks | Complete Date |
| 5 | 10/04/2015 | 10/04/2015 | create GitHub | 10/04/2015 |
|  | 10/04/2015 | 11/04/2015 | improve system structure | 11/04/2015 |
|  | 11/04/2015 | 17/04/2015 | improve class diagram | 17/04/2015 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| week | Start Date | Date Planned to Finish | Tasks | Complete Date |
| 6 | 17/04/2015 | 18/04/2015 | research about JSP | 18/04/2015 |
|  | 18/04/2015 | 19/04/2015 | discuss report issues | 19/04/2015 |
|  | 19/04/2015 | 20/04/2015 | complete individual work-diary. | 20/04/2015 |

*7.3.5 Sandon Joubert:*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Week | Start Time | Planned End Time | Task | Description | Actual End Time |
| 3 | 20/03/2015 | 27/03/2015 | Analyse General Public Requirements | Did draft use case for the General Public, as well as corresponding use case description and srs | 24/03/2015 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Week | Start Time | Planned End Time | Task | Description | Actual End Time |
| 4 | I was away for a weeding for most of this week |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Week | Start Time | Planned End Time | Task | Description | Actual End Time |
| 5 | 03/04/2015 | 03/04/2015 | Complete Requirements | Finalised used case diagrams, description and srs | 03/04/2015 |
|  | 03/04/2015 | 03/04/2015 | Analyse Risk Management | Analysed and finalised possible risks and countermeasures | 06/04/2015 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Week | Start Time | Planned End Time | Task | Description | Actual End Time |
| 6 | 10/04/2015 | 17/04/2015 | Individual Work Diary | Bring my individual work diary up to standard and organise it to be near ready for hand in | 12/04/2015 |
|  | 10/04/2015 | 23/04/2015 | Class Description | Complete the class description | 22/04/2015 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Week | Start Time | Planned End Time | Task | Description | Actual End Time |
| 7 | 17/04/2015 | 23/04/2015 | Documentation | Help out with any final documentation to be handed in at the end of the week | 23/04/2015 |

7.4 Screenshot of Version Control Software (GitHub)

8. Member Contribution Assessment

|  |  |  |
| --- | --- | --- |
| Name/Student number | Contribution Rank | Signature |
| Zheli Jiang/ | Contributed |  |
| Junyan Fan/ | Contributed |  |
| Siyuan Hou/ | Contributed |  |
| Sandon Joubert/ | Contributed |  |
| Ruixi He/4174458 | Contributed |  |

**Appendix**

**Definitions**

**actor**

Someone or something outside the system or business that interacts with the system or business.

**class**

A description of a set of objects that have the same responsibilities, relationships, operations, attributes, and semantics.

**domain**

An area of knowledge or activity characterized by a family of related systems.

**management**

A supporting discipline in the software engineering process whose purpose is to plan and manage the development project.

**object**

An entity with a well-defined boundary and identity that encapsulates state and behaviour. State is represented by attributes and relationships, and behaviour is represented by operations and methods. An object is an instance of a class.

**operation**

A service that can be requested from an object to affect behaviour.

**requirement**

A description of a condition or capability of a system; either derived directly from user needs or stated in a contract, standard, specification, or other formally imposed document.

**scenario**

A described use-case instance or a subset of a use case.

**use case**

A sequence of actions a system performs that yields an observable result of value to a particular actor. A use-case class contains all main, alternate, and exception flows of events related to producing the observable result of value. Technically, a use case is a class whose instances are scenarios.

**vision**

The user's or customer's view of the product to be developed.

**Acronyms**

SRS - Software Requirement Specification

Admin – Project Management System Administrator

GUI - Graphical User Interface

GB - Gigabytes

|  |  |
| --- | --- |
| Customer report | Generate a Customer report |
| Flight report | Generate a Flight report |
| Service report | Generate a Service report |
| Agency report | Generate an Agency report |
| Company report | Generate a Company report |
|  |  |
| Customer info | Get customer info, such as name, credit card, reservation… |
| Flight info | Get flight information, such as destination, source airport... |
| Flight details | Get more information about flight, such as seat be reserved, left ticket in each class… |
| Agency info | Get agency information, such as agency name, member in this agency, ticket reserved by this agency… |
| Service info | Get service information, such as service company provided… |
| Company info | Get company information, such as airplane number, airport can be arrived… |
| Create flight | Create a new airline. |
| Edit flight | Change the information of a flight. |
| Delete flight | Delete a exist flight. |
| Add airport | Add a new airport could be arrived. |
| Edit airport | Change the information of an airport. |
| Delete airport | Delete a exist airport. |
| Add airplane | Add a new aircraft. |
| Edit airplane | Change the information of an aircraft. |
| Delete airplane | Delete a exist aircraft. |
| Enable account | Enable account |
| Disable account | Disable account |
| Promote | Promote a stuff to manager |
|  |  |
| Register | Gather Non-member’s information; create a member account which could make reservation. |
| Ticket reservation | Book a ticket |
| Service reservation | Add new service to a ticket |
| Change seat | Select of reselect a seat if available. |
| Chang service reservation | Delete or change exist service. |
| Switch seat | Stuff can switch passages’ seat after them select a seat. |
| Chang flight | Rescheduling a ticket |
|  |  |
| Add service | Create a new service, make it selectable |
| Change service | Change exist service’s name, price, availability… |
| Delete service | Delete a exist service. |