**SCHOOL BUS TRACKING AND ATTENDANCE CHECKING**

Test Plan

**By**

Mr. Thitipun Tojareonvanich 552115018

Mr. Puttipong Tadang 552115054

Department of Software Engineering

College of Arts, Media, and Technology

Chiang Mai University

**Project Advisor**

Mr. Parinya Suwansrikham

**Document History**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Document Name | Version | Status | Date | Viewable | Editable | Responsible |
| School Bus Tracking and Attendance Checking-TestPlan-V.0.1.docx | 0.1 | Reviewed | 20/7/2015 | TT, PT, PS | TT, PT | TT, PT |

\***TT** = Thitipun Tojareonvanich, \***PT** = Puttipong Tadang, \***PS** = Parinya Suwansrikham

**Document Designed by**

Mr.Thitipun Tojareonvanich

Mr.Puttipong Tadang

**Table of Contents**

[Chapter One | Introduction 1](#_Toc425296782)

[1.1 Identification 1](#_Toc425296783)

[1.2 Project Scope 1](#_Toc425296784)

[1.3 Document Overview 2](#_Toc425296785)

[1.4 Objectives 3](#_Toc425296786)

[1.5 Work Products to be Develop 3](#_Toc425296787)

[1.5.1 Deliverable 3](#_Toc425296788)

[1.6 Acronyms and Definitions 3](#_Toc425296789)

[1.6.1 Acronyms 3](#_Toc425296790)

[1.6.2 Definitions 3](#_Toc425296791)

[Chapter Two | Infrastructure 3](#_Toc425296792)

[2.1 Software Development Life Cycle 3](#_Toc425296793)

[2.2 Development Tools 3](#_Toc425296794)

[2.3 Hardware and Material Resources 3](#_Toc425296795)

[Chapter Three | Management Procedures 6](#_Toc425296796)

[3.1 Project Team Structure 6](#_Toc425296797)

[3.2 Monitoring and Controlling Mechanisms 6](#_Toc425296798)

[3.2.1 Project Meeting 6](#_Toc425296799)

[Chapter Four | Quality Planning 6](#_Toc425296800)

[4.1 Quality Standard 6](#_Toc425296801)

[4.1.1 Basic Profile Guide Processes 6](#_Toc425296802)

[4.1.2 Project Management Process 6](#_Toc425296803)

[4.1.3 Software Implementation Process 6](#_Toc425296804)

[4.2 Reviews/Responsibility 7](#_Toc425296805)

[4.3 Testing 7](#_Toc425296806)

[4.4 Quality Factors 7](#_Toc425296807)

[Chapter Five | Schedule and Milestones 7](#_Toc425296808)

[Chapter Six | Software Configuration Management 14](#_Toc425296809)

[6.1 Software Configuration Management 14](#_Toc425296810)

[6.2 Filename Format 14](#_Toc425296811)

[6.3 Change Management 14](#_Toc425296812)

[Chapter Seven | Estimated Effort and Cost 14](#_Toc425296813)

[Chapter Eight | Risk Management 14](#_Toc425296814)

[8.1 Risk Management Process 14](#_Toc425296815)

[8.2 Risk Identification and Solutions 14](#_Toc425296816)

[Chapter Nine | Reference 15](#_Toc425296817)

Chapter One | Introduction

# This test approach document describes the appropriate strategies, process, workflows and methodologies used to plan, organize, execute and manage testing of the School Bus Tracking and Attendance Checking application.

* 1. Purpose

The purpose of Test Plan Document describes the unit testing plan and the system test for the School Bus Tracking and Attendance Checking application. In the document includes of the test cases which have to test in each progresses. The unit testing covers the implemented methods. The system testing covers the user requirements. The test data is ready for each method and user requirement provided in table.

1.2 Scope

This test plan document describes the testing activities for The School Bus Tracking and Attendance Checking application to measure user requirements, system requirements and to identify the test types which will be performed.

1.3 Objectives

The quality objective of The GPS Travel route on mobile is to ensure that the system delivered:

- Meet or exceeded users’ requirements

- The functionality delivered by the development team is as specified by Software Requirement Specification (SRS) and Software Design Document (SDD)

- Functions are consistently and reliability.

- Does not impact other systems or the existing technology environment.

1.4 Dependency Document

This document; which is Software Test Plan Document; is depended on the following documents:

- Software Project Management Plan version 3.0

- Software Requirement Specification version 3.0

- Software Design document version 3.0

- Test record version 3.0

- Traceability record version 3.0

Chapter Two | System Testing

**Test Case 1: Unregistered user can register on the application. (URS-01, URS-02, URS-03)**

***Description***

This test case is a system testing from user requirement number 1, 2, 3.

Unregistered user can register on application by entering information which is username, e-mail and password. (Username valid type is “a-z”, “A-Z”, “0-9” and “+ \_ - %.”) (E-mail valid type is “@”and “.”) (Password valid type is “a-z”, “A-Z”, “0-9” and “+ \_ %.”)

***Perquisite or test input***

Unregistered user’s username, e-mail and password.

***Test script***

1. Unregister user open application and goes to login form.

2. Unregister user fills his/her username, email and password.

3. Unregister user press “Register” button.

4. System check for authentication.

5. User’s information adds into the database and registration complete.

***Test Case***

|  |  |  |  |
| --- | --- | --- | --- |
| Test No. | Description | Input | Expect output |
| 1 | Fill all required  registration  information form,  which are Email and Password. | E-mail=t.t\_t-t@hotmail.com Password = art2533 | Unregistered user  has registered and  his/her information  stored in the  database. |
| 2 | Fill E-mail that already  exists in the database. | E-mail= **t.t\_tt@**  **hotmail.com** Password = abc123 | Warning message  shows and  unregistered user  not register yet. |
| 3 | Fill some required  registration  information form,  which are username  and password but do  not fill e-mail. | E-mail=  Password = abc123 | Warning message  shows and  unregistered user  not register yet. |
| 4 | Fill some required  registration  information form,  which are username  and e-mail but do not  fill password. | Email=  iloveyou@hotmail.com  Password = | Warning message  shows and  unregistered user  not register yet. |
| 5 | Not fill any registration  information. | Null | Warning message  shows and  unregistered user  not register yet. |
| 6 | There should be the  case that application  cannot connect to  internet | E-mail=t.t\_t-t@hotmail.com Password = art2533 | Warning message  shows and  unregistered user  not register yet. |
| 7 | Fill password less than  6 characters | E-mail=t.t\_t-t@hotmail.com Password = 1234 | Warning message  shows and  unregistered user  not register yet. |

**Test Case 2: Registered users can login to application*.* (URS-04, URS-06, URS-08)**

***Description***

This test case is a system testing from user requirement number 4, 6, 8.

Registered users can login to application by authentication user system that check registered user’s information from the database. (E-mail valid type is “@”and “.”) (Password valid type is “a-z”, “A-Z”, “0-9” and “+ \_ %.”)

***Perquisite or test input***

Registered user’s e-mail and password.

***Test script***

1. Registered user open application then system brings the user to login form.

2. Registered user enters his/her e-mail and password.

3. Registered user press “Login” button.

4. System check for authentication.

5. Registered user logged in to the application.

***Test Case***

|  |  |  |  |
| --- | --- | --- | --- |
| Test No. | Description | Input | Expect output |
| 1 | Test with correct  Registered user’s e-mail  and correct password. | E-mail=t.t\_t-t@hotmail.com Password = art2533 | Unregistered user  has registered and  his/her information  stored in the database. |
| 2 | Test with correct  Registered user’s e-mail  but wrong password. | E-mail= **t.t\_tt@**  **hotmail.com** Password = abc123 | Warning message shows  and registered user  cannot login to the application. |
| 3 | Test with wrong  Registered user’s e-mail  and wrong password. | E-mail=  Password = abc123 | Warning message shows  and registered user  cannot login to the application. |
| 4 | Test with invalid type of  Registered user’s e-mail  and wrong password. | Email=  iloveyou@hotmail.com  Password = | Warning message shows  and registered user  cannot login to the  application. |
| 5 | Test with filling E-mail  but do not fill password. | Null | Warning message shows  and registered user  cannot login to the  application. |
| 6 | Test with filling  password but do not fill e-mail. | E-mail=t.t\_t-t@hotmail.com Password = art2533 | Warning message shows  and registered user  cannot login to the  application. |
| 7 | Test with not fill  anything. | Null | Warning message shows  and registered user  cannot login to the  application. |
| 8 | There should be the  case that application  cannot connect to  internet | E-mail=t.t\_t-t@hotmail.com Password = art2533 | Warning message shows  and registered user  cannot login to the  application. |

**Test Case 3:** **Registered users can logout to application*.* (URS-05, URS-07, URS-09)**

***Description***

This test case is a system testing from user requirement number 5, 7, 8.

Registered users can logout from application any time they want.

***Perquisite or test input***

Registered user already login to the application.

Logout button.

***Test script***

1. Registered user press “Logout” button.

2. Registered user logout from application

***Test Case***

|  |  |  |  |
| --- | --- | --- | --- |
| Test No. | Description | Input | Expect output |
| 1 | Registered user press logout button for logging out from application. | Logout button. | Registered user logout from application and system bring registered user to login page. |

**Test Case 4:** **School bus driver can scan QR code for children attendance. (URS-16)**

***Description***

This test case is a system testing from user requirement number 16.

Registered user as school bus driver can scan the children QR code by built in camera.

***Perquisite or test input***

Registered already login to the application as school bus driver user.

Scan QR code button.

***Test script***

1. Press Scan QR code button.

2. System open built-in camera.

3. School bus driver scan the children QR code.

4. System identify the QR code

5. Record the data (time that QR code was scan) to database.

6. System send the message their parents.

***Test Case***

|  |  |  |  |
| --- | --- | --- | --- |
| Test No. | Description | Input | Expect output |
| 1 | Registered user as school bus driver press on “Scan QR code” button. | Scan QR code button. | Children QR code was scan and record to the database. Then message to their parents |

**Test Case 5:** **Parents can checking their children's attendance. (URS-14)**

***Description***

This test case is a system testing from user requirement number 14.

Registered user as parents can check their children’s attendance.

***Perquisite or test input***

Registered user already login to the application as parents user.

Attendance button.

***Test script***

1. Press on Attendance button.

2. System show their children status.

***Test Case***

|  |  |  |  |
| --- | --- | --- | --- |
| Test No. | Description | Input | Expect output |
| 1 | Registered user as parents press on Attendance button. | Attendance button. | Their children’s status have been shown. |

**Test Case 6:** **Children can receive the message when the bus are nearby. (URS-15)**

***Description***

This test case is a system testing from user requirement number 15.

Registered user as children can receive the message when the bus are nearby.

***Perquisite or test input***

Registered user already login to the application as children user.

***Test script***

1. The bus come nearby the children position.
2. Children can receive the message.

***Test Case***

|  |  |  |  |
| --- | --- | --- | --- |
| Test No. | Description | Input | Expect output |
| 1 | Registered user as children can receive the message when the bus come nearby. | The bus nearby. | The message send to the children user when the bus nearby. |

**Test Case 7:** **Parents can receive the message when their children arrive the school or home. (URS-18)**

***Description***

This test case is a system testing from user requirement number 18.

Registered user as parents can receive the message when their children arrive school or home.

***Perquisite or test input***

Registered user already login to the application as parents user.

***Test script***

1. Their children arrive school or home.
2. Parents can receive the message.

***Test Case***

|  |  |  |  |
| --- | --- | --- | --- |
| Test No. | Description | Input | Expect output |
| 1 | Registered user as parents can receive the message when their children arrive school or home. | The bus arrive school or home. | The message send to parent user when the bus arrive school or home. |

**Test Case 8:** **Parents can cancel the schedule. (URS-10)**

***Description***

This test case is a system testing from user requirement number 10.

Registered user as parents can cancel the school bus schedule.

***Perquisite or test input***

Registered user already login to the application as parents user.

“Cancel schedule” button.

***Test script***

1. Press on the “Cancel schedule” button.

2. System provide confirm pop-up

3. Press on “Yes” button.

4. System cancel the school bus schedule.

5. System show Parents interface.

***Test Case***

|  |  |  |  |
| --- | --- | --- | --- |
| Test No. | Description | Input | Expect output |
| 1 | Registered user as parent press on the “Cancel schedule” button. | “Cancel schedule” button. | The schedule was cancel from school bus driver schedule. |

**Test Case 10:** **School bus driver can send an extra case message. (URS-19)**

***Description***

This test case is a system testing from user requirement number 19.

Registered user as school bus driver can send an extra case message to the parents user.

***Perquisite or test input***

Registered user already login to the application as school bus user.

“Extra case” button.

***Test script***

1. Press on “Extra case” button.

2. System show type of the case interface to user.

3. Press on the wanted case.

4. System show the confirm pop-up.

5. Press on “Yes” button.

6. System send selected case as message to parent.

***Test Case***

|  |  |  |  |
| --- | --- | --- | --- |
| Test No. | Description | Input | Expect output |
| 1 | Registered user as school bus driver press on the “Extra case” button. | “Extra case” button. | Send the selected case as message to parent user. |

**Test Case 11:** **Registered user can view their own route. (URS-11, URS-12, URS-13)**

***Description***

This test case is a system testing from user requirement number 11, 12, 13.

Registered user can view their own route via google map api on application.

***Perquisite or test input***

Registered user already login to the application as school bus user, parents user, children user.

“Tracking” button.

***Test script***

2. Press on “Tracking” button.

3. System show the route via google map api.

***Test Case***

|  |  |  |  |
| --- | --- | --- | --- |
| Test No. | Description | Input | Expect output |
| 1 | Registered user as school bus user, parents user, children user press on the “Tracking” button. | “Tracking” button. | Google api can provide the route. |

**Test Case 12:** **School bus driver can receive alert message from the system when they drive over the speeding. (URS-17)**

***Description***

This test case is a system testing from user requirement number 17.

Registered user as school bus driver can receive alert message when the speed are more than the speeding.

***Perquisite or test input***

Registered user already login to the application as school bus user.

***Test script***

1. School bus driver drive the bus more than speed limit.

2. System send alert message to school bus driver.

3. School bus driver receive the alert message.

***Test Case***

|  |  |  |  |
| --- | --- | --- | --- |
| Test No. | Description | Input | Expect output |
| 1 | School bus driver drive the bus more than speeding. | School bus driver drive more than the speeding | School bus driver receive speeding message. |

**Test Case 13:** **School bus driver can turn on the application. (URS-20)**

***Description***

This test case is a system testing from user requirement number 20.

Registered user as school bus driver can turn on the application.

***Perquisite or test input***

Registered user already login to the application as school bus user.

***Test script***

1. Press on the “Start” button.

2. Start all of activity in application.

***Test Case***

|  |  |  |  |
| --- | --- | --- | --- |
| Test No. | Description | Input | Expect output |
| 1 | School bus driver press on the “Start” button to start all of activity in this application. | “Start” button. | Start all of activity in application. |

**Test Case 14:** **School bus driver can turn off the application. (URS-21)**

***Description***

This test case is a system testing from user requirement number 21.

Registered user as school bus driver can turn off the application.

***Perquisite or test input***

Registered user already login to the application as school bus user.

***Test script***

1. Press on the “Stop” button.

2. Start all of activity in application.

***Test Case***

|  |  |  |  |
| --- | --- | --- | --- |
| Test No. | Description | Input | Expect output |
| 1 | School bus driver press on the “Stop” button to stop all of activity in this application. | “Stop” button. | Stop all of activity in application. |

Chapter Three | Management Procedures

3.1 Project Team Structure

3.2 Monitoring and Controlling Mechanisms

3.2.1 Project Meeting

Chapter Four | Quality Planning

4.1 Quality Standard

• ISO 29110 for Very Small Entity (VSE)

ISO 29110 is a software processes and guidelines for very small entities. A very small entity is mean an enterprise, organization, department or project having up to 25 people. The guide are based on subsets of appropriate standards elements, referred to as VSE profiles. The purpose of a VSE profile is to define a subset of ISO/IEC standards relevant to the VSE.

4.1.1 Basic Profile Guide Processes

4.1.2 Project Management Process

The purpose of the project management process is to establish and carry out in a systematic way the tasks of the software implementation project, which allows complying with the project’s objectives in the expected quality, time and cost.

Selected processes

4.1.2.1 Project planning process

4.1.2.2 Project plan execution process

4.1.2.3 Project assessment and control process

4.1.2.4 Project closure process

4.1.3 Software Implementation Process

The purpose of the software implementation process is the systematic performance of the analysis, design, construction, integration and tests activities for new or modified software products according to the specified requirement.

Selected processes

4.1.3.1 Software implementation initiation process

4.1.3.2 Software requirement analysis process

4.1.3.3 Software architectural and detailed design process

4.1.3.4 Software construction process

4.1.3.5 Software integration and test process

4.1.3.6 Software delivery process

4.2 Reviews/Responsibility

4.3 Testing

4.4 Quality Factors

URS

Chapter Five | Schedule and Milestones

4.1 Schedule and Milestones

**Member System**

* Feature#1: Registration system
* Feature#2: Login system

**Checking attendance system**

* Feature#3: QR code reader
* Feature#4: QR generator
* Feature#5: Checking attendance system

**Notifying System**

* Feature#6: Parent notifying message system
* Feature#7: Cancelling the school bus ride system

**Tracking System**

* Feature#8: Tracking system
* Feature#9: Calculating approximate arrival time system

**Speed Control Alert System**

* Feature#10: Send alert message system

4.2 Schedule Plan

* **Proposal phase**

Create proposal document.

* **Progress I**

Create Development Plan, Quality plan, Software requirement specification, Software design document and some part of Test document. Start creates member system and checking attendance system.

* **Progress II**

Create notifying system and tracking system, overall of the system should be higher than 65%. And Test document.

* **Progress III**

Create speed control alert system and integrate all features. Overall of the system should be complete or nearly. And Test document.

* **Final progress**

Integrate and review all document. Make sure all system and document are complete.

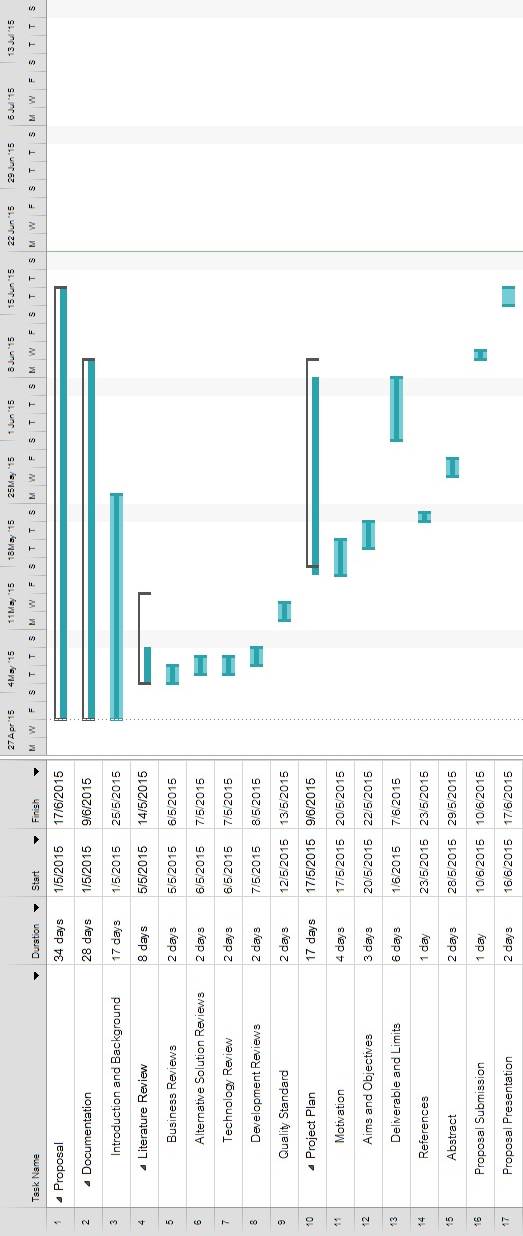


Figure Proposal phase

Figure Proposal phase

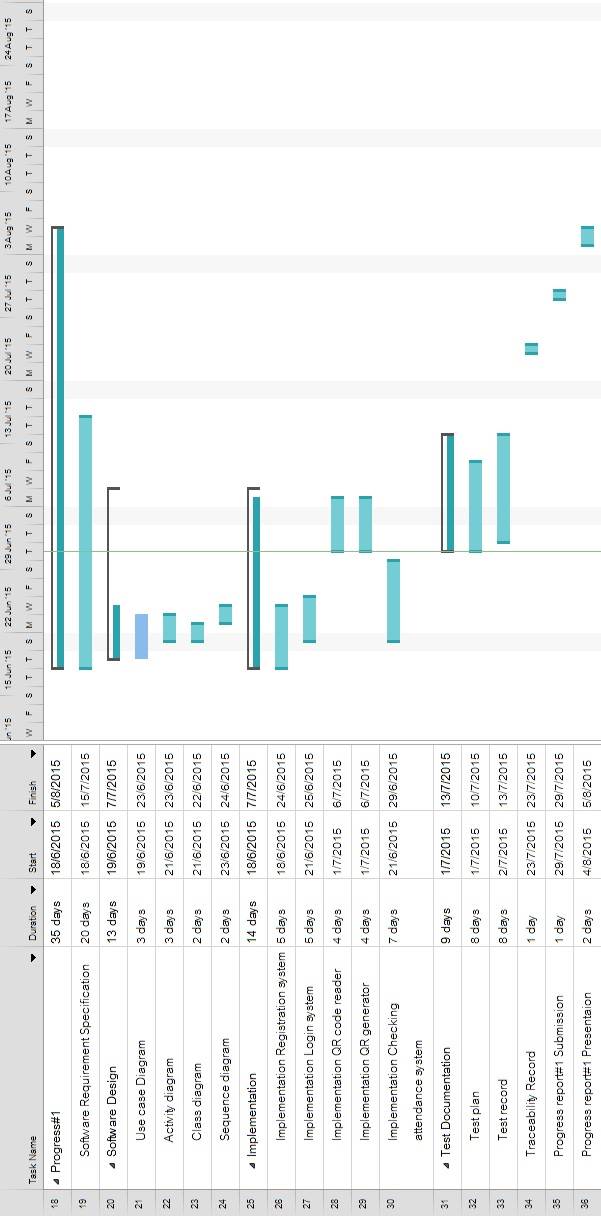


Figure Progress I

Figure Progress I

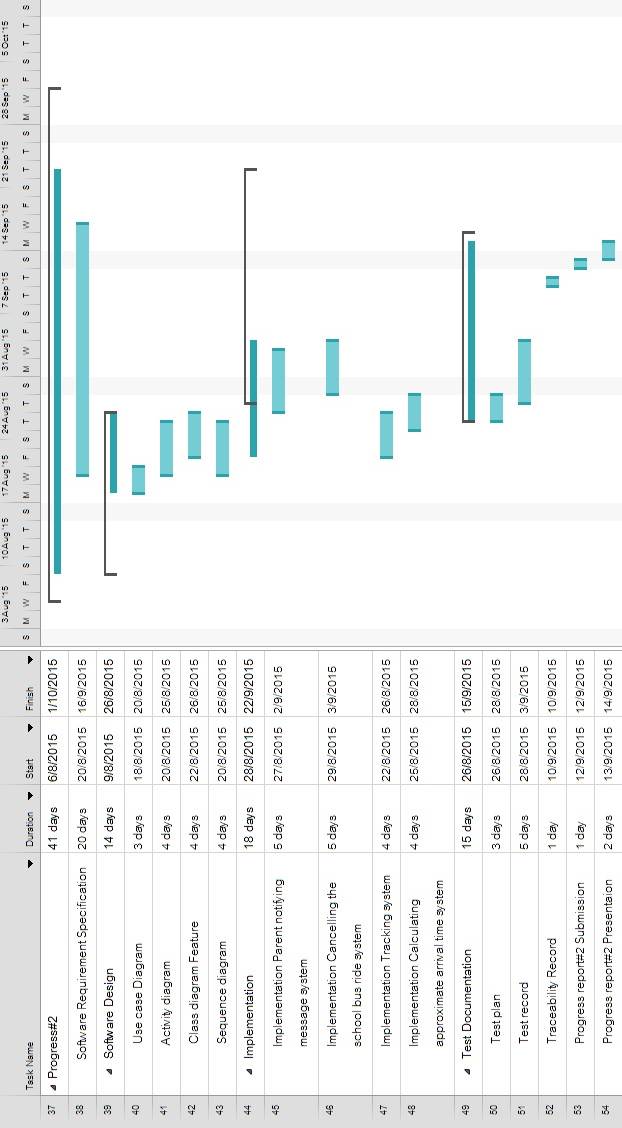


Figure Progress II

Figure Progress II



Figure Progress III

Figure Progress III

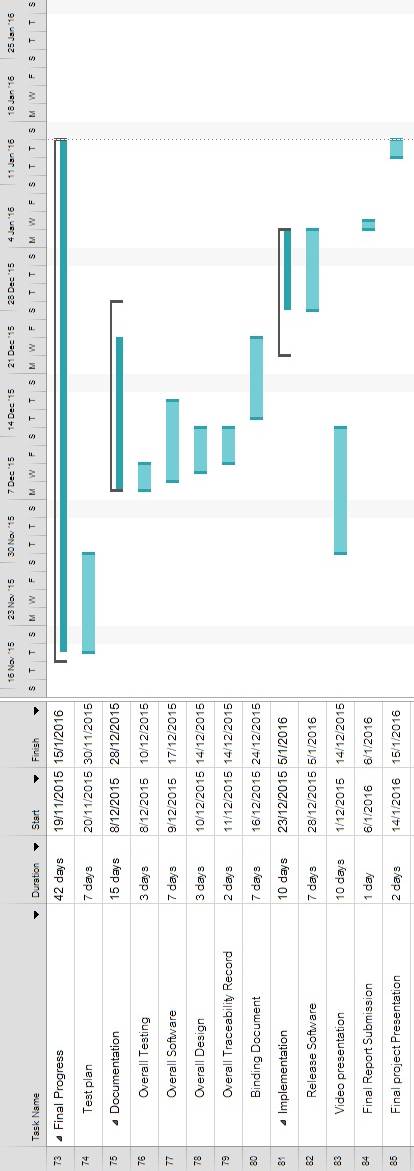


Figure Final Progress

Figure Final Progress

Chapter Six | Software Configuration Management

6.1 Software Configuration Management

Software Configuration Management can coordinate the software management, which can minimize the confusion in the development. It is a set of activities designed to control changes by identifying the parts of the development that is likely to be changed, establishing relationships among them, defining mechanisms for managing different versions of them, controlling the changes imposed, and auditing and reporting on the changes made. In a word, Software Configuration Management is a methodology to control and manage a software development with its configuration. It can determine what to change and who to be responsible for the change when something goes wrong.

6.2 Filename Format

6.3 Change Management

Change Management manages all the changes in the software development of the project. All the change requests will be recorded in the Change Request Document. We use the strategy for change management as following steps:

1. Analyzing the change.

2. Designing the change plan.

3. Requesting for the change.

4. Approving the change request by project advisor.

5. Implementing the change as the approved change request.

6.4 Software Configuration Item Table

Chapter Seven | Estimated Effort and Cost

Most cost will come from learning materials and the hard copy documents. Because for this project, we develop our application with open source tools. So the most cost will be spent on buying some learning textbooks and printing the documents.

Chapter Eight | Risk Management

8.1 Risk Management Process

## QQ截图20140510113842

Figure 9 : Risk Management Process

**First step**, Identify project and business risks that have the potential of affecting the project and documenting the risk’s characteristics.

**Second step**, Identify and assess the probability and impact of the risks.

**Third step**, Come up with plans that will minimize or avoid threats and maximize opportunities.

**a**. Accepting the risk

**b**. Avoid the risk

**c**. Contingency plans

**d**. Transfer the risk

**e**. Mitigate the risk

8.2 Risk Identification and Solutions

**Three criterion:**

**L-**Low, **N**-Normal, **H**-High

**Technology failure**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Risk ID** | **Headline** | **Description** | **Priority Level** | **Impact** | **Likelihood**  **of Event** | **Mitigation Strategy** |
| 1 | Server fail | Some factors lead to the situation where the server cannot provide accurate data. | N | Users may not be satisfied with the application. | Certainty | Exception Handling. Displaying proper messages for the users. |

**Human failure**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Risk ID** | **Headline** | **Description** | **Priority Level** | **Impact** | **Likelihood**  **of Event** | **Mitigation Strategy** |
| 2 | Lack of responsibility of team’s member | A member of the team cannot finish her task on time, or cannot meet the requirement of the project. | H | The project cannot be delivered in time. | Somewhat likely | Report every team member’s work at project meetings, and discuss what should they finish before next meeting. |
| 3 | Lack of skills | A member of the team does not have experience and skill to do the project. To find/create certain functions, developers take a longer time since they need to gather knowledge first. | H | It may lead to the situation where the project cannot finish in time. | Certainty | Practice and training |
| 4 | A team member gets sick or has an accident | A team member may get sick or has an accident, and cannot work on the project. | L | It may lead to the situation where the project cannot finish in time. | Somewhat likely | · Adjust the plans and schedules.  · The absent member should assign works to the other member who can continue work on the project |

**Process failure**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Risk ID** | **Headline** | **Description** | **Priority Level** | **Impact** | **Likelihood**  **of Event** | **Mitigation Strategy** |
| 5 | Project schedule changes. | Some factors may force the project schedule to change. | H | Influence progress | Certainty | Create comprehensive project timeline and adjust the project schedule according to the requirements. |
| 6 | The system faces reliability or stability | Users cannot use the application or cannot get complete information because of unexpected external factors. | N | Users may not be satisfied with the application. | Somewhat likely | Manage the application and maintain the system timely. |
| 7 | Requirements might change. | New ideas from stakeholders make the project requirements change. | H | Affect the process of the application development. | Certainty | · Before developing each function, understand the needs as much as possible. · Meet and discuss with other stakeholders. |

Chapter Nine | Reference