**Software Requirements Specification**

**For**

**Module 9: Accountant Delegate Task List**

**Version 1.0**

**Prepared by : 1. Siti Nur Nihlah Binti Mohd Asly 177319**

**2. Suliana Binti Said 176593**

**3. Rosninawati Binti Roslee 179196**

**4. Nurul Syahnia Binti Abdul Agis 178788**

**5. Mira Azira Binti Jiwara 177652**

**Table of Contents**

**Table of Contents……………………………………………………………………….….....2**

**1.0** **Introduction…………………………………………………………………………........3**

1.1 Problem statement …..……………..……………………………………………….....3

1.2 Objectives……………………...……………………………………………………....3

1.3 Scope……………………………………………………………………….………….4

**2.0** **Overall Description………………………………………………………………….....4-6**

2.1 User Classes and Characteristics…………………………………………….…….…..4

2.2 Operating Environment…………………………………………………………..........5

2.3 Design and Implementation Constraints………………………………………..…......5

2.4 Assumptions and Dependencies………………………………………………….........6

**3.0** **External Interface Requirements ..…………………………………...………………6-8**

3.1 User Interfaces ..……………………………………………………………….........6-7

3.2 Hardware Interfaces ..………………………………………………………….……...8

3.3 Software Interfaces……………………………………………………….…………....8

3.4 Communications Interfaces…………………………………………….………….......8

**4.0** **System Use Cases ..……………………………………………………………...…....9-15**

4.1 View All Files (UC 1) ..……………………………………………………..…….... 10

4.2 Sort Files(UC2) ..……………………………………………………………….........11

4.3 Read Task (UC3) ..………………………………………………………...................12

4.4 Update Task (UC4)……………………………………………………………...........13

4.4 View Account (UC4)…………………………………………………………............14

4.4 Open Account (UC4)……………………………………………………....................15

**5.0 Functional Requirements………………………………………………………............16**

**6.0 Non-functional Requirements………………………………………………….............17**

6.1 Performance requirement………………………………...………………………......17

6.2 Safety requirement ..……………………………………………………………........17

6.3 Security and Integrity requirement…………………………………………..….........17

6.4 Reliability requirement…………………………………………………………….....17

6.4 Usability requirement…………………………………………..………………….....17

**7.0** **Other Requirements………………………………………………………………….....18**

**8.0** **System Requirements Chart………………………………………………………........18**

**9.0 References ..…………………………………………………………………….…….....****19**

**1.0 INTRODUCTION**

**1.1 Problem Statement**

The focus of this project is to create an accounting system. Our group have been assigned to do the ‘Accountant Delegate Task List’ module. Nowadays, delegation involves so much more than just giving someone a task to do. It involves assigning accountability for a specific outcome and explaining the results that are expected to be achieved*.*How this happens is up to the person or team involved. Since most of accountant crave trust and responsibility in their work, effective delegation of tasks provides a way for them to meet their needs in order to produce good results.

The reason of creating this module is because there are a lots of problem faced by the accountant themselves. Some of the problems are, the accountant cannot see all files to be process. Not only that, they faced a problem in sorting the file based on priority. It is also hard for them to see the data input from clerk for every account. Lastly, some of them cannot open the task to enter data in that account.

The module were be done by applying the agile methodology and suitable tool.

## 1.2 Objectives.

1.2.1 To make sure the accountants can see all files to be process.

1.2.2 To help the accountants sort the file according to priority.

1.2.3 To make sure the accountant can see the data input from clerk for every account.

1.2.4 To help the accountant for opening task and enter data in it.

**1.3 Scope.**

The module is to be use in any company that have finance and accounting department in Malaysia. The target users of the module is the accountant. This project is mainly emphasized on developing a system for accounting process. It also include some other functions that can help the target users to improve their performance.

The accountant delegate task list module consist several database which include four database which includes the accountant database, file database, task database and account database.

# **2.0 OVERALL DESCRIPTION**

## 2.1 User Classes and Characteristics

This table shows user levels and their characteristics in Accountant Delegate Task List module.

|  |  |  |
| --- | --- | --- |
|  | User Classes | Characteristics |
| I. | Accountant | Accountant plays an important role because construction worker is the main user which involve in every single operation in the module.  Accountant can see all the file to be process, they can sort the file according to priority. They also can see the data input from clerk for every account. Accountant also can read and open the task to enter the data. |

## 2.2 Operating Environment

2.2.1 Software

i) Command Prompt [1]

Also known as cmd.exe or cmd (after its executable file name), is the command-line interpreter on Windows operating systems.

ii) XAMPP [2]

XAMPP cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MySQL database, and interpreters for scripts written in the PHP programming language.

iii)Sublime Text [3]

Sublime Text is a proprietary cross-platform source code editor with a Python application programming interface (API). It natively supports many programming languages and markup languages, and its functionality can be extended by users with plugins, typically community-built and maintained under free-software licenses.

2.2.2 Hardware

i) Laptop

Laptop is a small, portable personal computer. Easy to carry that make laptop as the hardware use to develop DGD.

## 2.3 Design and Implementation Constraints

There are several constraints in the Accountant Delegate Task List module. Firstly, this module only can support up to 200GB. The application only supports 1 language which is English. Thus those who are not native English speakers will find it hard to interact with the system interface.

## 2.4 Assumptions and Dependencies

This module might face problem when each module need to be integrated since this project is a distributed project. It is hard to integrate the code since each module were done by different team and some of the version of Laravel are not compatible to other version.

# **3.0 EXTERNAL INTERFACE REQUIREMENTS**

**3.1 Application Interface**

3.1.1 File Page

## 

## Figure 1. shows the File List Page

This page basically handles two main function which are view all files and sort the file base on its priority which can be prioritize by High, Medium and Low.

## 3.1.2 Task Page

## 

## Figure 2. shows the Task List Page

This page basically handles two main function which are read tasks and update the task. To update the task, the accountant need to click the update link.

## 3.1.3 Account Page

## 

## Figure 3. shows the Account List Page

This page basically handles two main function which are view the account and open the account. We can detect the chart of account (CoA) and the account type from this page.

## 3.2 Hardware Interfaces

## None

## 3.3 Software Interfaces

## None

## 3.4 Communications Interfaces

## None

# **4.0 SYSTEM USE CASE**



Figure 4. The use case of Accountant Delegate Task List module.

This is the use case for Accountant Delegate Task List module. The actor involve is the Accountant. The Accountant can view all files and also sorts the files base on it priority. Accountant also can read and update the task that they have been assigned. The accountant also can view the account after that open the account.

# **4.1 View all files (UC1)**

4.1.1 UC1

4.1.2 Objective

To allow the accountant to view all files that are assign to them.

4.1.3 Priority

High

4.1.4 Source

File

4.1.5 Actor

Accountant

4.1.6 Flow of Events

|  |
| --- |
| Use Case : View all files. |
| Actor : Account |
| 1. The use case begins when accountant get the files that is assigned to them. 2. The accountant can view all files that have been assigned to them |

4.1.7 Includes

None

4.1.8 Preconditions

None

4.1.8 Post conditions

The information is displayed.

4.1.9 Notes/Issues

None

# **4.2 Sort files (UC2)**

4.2.1 UC2

4.2.2 Objective

To help the accountant to sort all the file based on the file’s priority.

4.2.3 Priority

High

4.2.4 Source

File

4.2.5 Actor

Accountant

4.2.6 Flow of Events

|  |
| --- |
| Use Case : Sort file |
| Actor : Accountant |
| 1. The use case begins when the accountant can view the files. 2. The accountant can sort the file based on the priority either it is high, medium or low. |

4.2.7 Includes

None

4.2.8 Preconditions

None

4.2.9 Post conditions

The information is displayed and save in the system.

4.2.10 Notes/Issues

None

# **4.3 Read task (UC3)**

4.3.1 UC3

4.3.2 Objective

To ease the accountant to open the task given to them.

4.3.3 Priority

High

4.3.4 Source

Task

4.3.5 Actor

Accountant

4.3.6 Flow of Events

|  |
| --- |
| Use Case : Read task |
| Actor : Accountant |
| 1. The use case begins when the accountant was assigned task to them. 2. The system will able the accountant to open/read the task. |

4.3.7 Includes

None

4.3.8 Preconditions

None

4.3.9 Post conditions

The information is displayed

4.3.10 Notes/Issues

None

# **4.4 Update task (UC4)**

4.4.1 UC2

4.4.2 Objective

To help the accountant to update the task assigned to them

4.4.3 Priority

High

4.4.4 Source

Task

4.4.5 Actor

Accountant

4.4.6 Flow of Events

|  |
| --- |
| Use Case : Update task |
| Actor : Accountant |
| 1. The use case begins when the accountant read the task and realise something to be updated. 2. The accountant made some new update to the task and save it. |

4.4.7 Includes

None

4.4.8 Preconditions

None

4.4.9 Post conditions

The information is displayed and save in the system.

4.4.10 Notes/Issues

None

# **4.5 View account (UC5)**

4.5.1 UC2

4.5.2 Objective

To ease the accountant to view the their account.

4.5.3 Priority

High

4.5.4 Source

Account

4.5.5 Actor

Accountant

4.5.6 Flow of Events

|  |
| --- |
| Use Case : View account. |
| Actor : Accountant. |
| 1. The use case begins when the accountant have assigned account. 2. The accountant can view information about the account. |

4.5.7 Includes

None

4.5.8 Preconditions

None

4.5.9 Post conditions

The information is displayed

4.5.10 Notes/Issues

None

# **4.6 Open account (UC6)**

4.6.1 UC2

4.6.2 Objective

To ease the accountant to open the new account.

4.6.3 Priority

High

4.6.4 Source

Account

4.6.5 Actor

Accountant

4.6.6 Flow of Events

|  |
| --- |
| Use Case : Open account. |
| Actor : Accountant. |
| 1. The use case begins when the accountant need to open up the new account. 2. The accountant opens a new account and save it to the system. |

4.6.7 Includes

None

4.6.8 Preconditions

None

4.6.9 Post conditions

The information is save in the system.

4.6.10 Notes/Issues

None

**5.0 FUNCTIONAL REQUIREMENTS**

5.1 The accountant shall be able to view all files.

5.2 The accountant shall be able to sort files.

5.3 The accountant shall be able to read the tasks.

5.4 The accountant shall be able to update the tasks through update link.

5.5 The accountant shall be able to view the account.

5.6 The accountant shall be able to open the account.

5.7 The task page shall include task list.

5.8 The task information shall include :

5.8.1 Task Title

5.8.2 Task Description

5.8.3 Status

5.8.4 Date

5.9 The file page shall include file list.

5.10 The file information shall include:

5.10.1 File Name

5.10.2 Priority

5.10.3 Size

5.10.4 Type

5.10.5 Date

5.11 The account information shall include:

5.11.1 Code

5.11.2 Name

5.11.3 Type

**6.0 NON-FUNCTIONAL REQUIREMENTS**

**6.1 Performance requirements**

The system should be able to response, display and update faster the information that have been inserted by the accountant . The system should be able to display every information and interfaces simultaneously.

**6.2 Safety requirement.**

The system should be able to cover the loss data after viruses attack to the hardware implemented with the system. The system also should be able to detect false information inserted by user other than the accountant itself so that it cannot change any future information*.*

* 1. **Security and Integrity requirement.**

The system should be secure and must use encryption to protect the databases and can be changed only by the system administrator. The system also should be able to provide specific identification accessing to the system before users can use if fully, for example; only accountant that are able to view, update, read and open the file, tasks and the account information. No further access by unauthorized people.

**6.4 Reliability requirement.**

The system reliability where it shall be able to use during work time which on office hours and shall always provide real time information about the accountant file, task and account information.

**6.5 Usability requirement.**

The system shall provide an easy to use graphic user interface, a user friendly system and the interface should be easily navigable.

**7.0 OTHER REQUIREMENTS**

None

**8.0 SYSTEM REQUIREMENT CHART**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Priority** | **Type** | **Source** | **Contained in Use Case(s)** | **Description** |
| UC1 | High | Functional | File | View all files | The system allow the accountant to view all files assigned. |
| UC2 | High | Functional | File | Sort Files | The system allow the accountant to sort file based on the priority. |
| UC3 | High | Functional | Task | Read task | The system allow the accountant to read or open the task assigned. |
| UC4 | High | Functional | Task | Update task | The system allow the accountant to change and update the task assigned. |
| UC5 | High | Functional | Account | View account | The system allow the accountant to view the account. |
| UC6 | High | Functional | Account | Open account | The system allow the accountant to open up the account. |

**9.0 REFERENCE**

[1] H. (n.d.). What is the Command Prompt in Windows? Retrieved January 12, 2017, from <https://www.lifewire.com/command-prompt-2625840>

[2] What is xampp? (n.d.). Retrieved January 12, 2017, from <http://www.frihost.com/forums/vt-96620.html>

[3] Using Sublime Text as your IDE. (n.d.). Retrieved January 12, 2017, from <https://www.chromium.org/developers/sublime-text>