

# Software Engineering Section 07, Sem.2 2014/2015

**SOFTWARE REQUIREMENT SPECIFICATION**

**Version 1.0**

**System’s Name:**

File Searching System

**Organization’s Name:**

Lembaga Pertubuhan Peladang Kawasan Johor Bahru Timur

**System Developer’s Name:**



**Team Members:**

1. Abdullah Azzam (A13CS4002)

2. Chan Vei Siang (A13CS0019)

3. Ikmal Faiq Albakri Bin Mustafa Albakri (A13CS0182)

4. Nur Aliya Safura Binti Mohd Ghazali (A13CS0102)

5. Nurfarrah Liana Binti Sa’adon (A13CS0120)

CONTENTS

# 1. INTRODUCTION 8

## 1.1 INTRODUCTION ON THE DOMAIN APPLICATION 8

## 1.2 ANALYSIS OF THE EXISTING SYSTEM 8

### 1.2.1 Background 8

### 1.2.2 Problem Identified 8

#### 1.2.2.1 Non-computerized system 8

#### 1.2.2.2 Not secure 9

#### 1.2.2.3 Easy to lost track of files 9

#### 1.2.2.4 Inefficient 9

## 1.3 SCOPE 9

## 1.4 GLOSSARY 9

## 1.5 REFERENCES 10

## 1.6 OVERVIEW 11

# 2. OVERALL DESCRIPTION 12

## 2.1 PRODUCT PERSPECTIVE 12

## 2.2 PRODUCT FUNCTIONAL REQUIREMENT 12

### 2.2.1 Search for File 12

### 2.2.2 Borrow File 12

### 2.2.3 Returning a File 12

### 2.2.4 Produce Report by File Category 12

### 2.2.5 Produce Report by File Type 13

### 2.2.6 Produce Report by Borrower List 13

### 2.2.7 Managing File by Adding New File 13

### 2.2.8 Managing File by Edit Existing File 13

### 2.2.9 Managing File by Removing Existing File 13

### 2.2.10 View File Borrowed History 13

### 2.2.11 Produce Checklist by Date of Edited 14

### 2.2.12 Produce Checklist by File Location 14

## 2.3 USER CHARACTERISTICS 14

### 2.3.1 Employee 14

### 2.3.2 Manager 14

### 2.3.3 File Manager 15

## 2.4 GENERAL CONTRAINTS, ASSUMPTIONS,

## DEPENDENCIES, GUIDELINES 15

### 2.4.1 General Constraints 15

#### 2.4.1.1 Browser used 15

#### 2.4.1.2 Server used 15

#### 2.4.1.3 Database used 15

### 2.4.2 Assumptions 16

#### 2.4.2.1 Working time 16

#### 2.4.2.2 Amounts of user at time 16

### 2.4.3 Dependencies 16

#### 2.4.3.1 Local server 16

#### 2.4.3.2 Backup storage 16

#### 2.4.3.3 One device 16

### 2.4.4 Guidelines 16

# 3. SPECIFIC REQUIREMENTS 17

## 3.1 SOFTWARE PRODUCT FEATURES (FUNCTIONAL REQUIREMENT) 17

### 3.1.1 Search for File Use Case 17

### 3.1.1.1 Brief Description 17

### 3.1.1.2 Flow of Events 17

### 3.1.1.2.1 Basic Flow - Search the File 17

### 3.1.1.2.2 Alternative Flow 17

#### 3.1.1.2.2.1 No file Found in the Database 17

### 3.1.1.3 Pre-conditions 18

### 3.1.1.4 Post-conditions 18

### 3.1.1.5 Sequence Diagram - Search for File 18

### 3.1.1.6 Activity Diagram - Search for File 19

### 3.1.2 Borrow File Use Case 20

### 3.1.2.1 Brief Description 20

### 3.1.2.2 Flow of Events 20

### 3.1.2.2.1 Basic Flow - Borrow File 20

### 3.1.2.2.2 Alternative Flow 20

#### 3.1.2.2.2.1 File not available 20

### 3.1.2.3 Pre-conditions 20

### 3.1.2.4 Post-conditions 21

### 3.1.2.5 Sequence Diagram - Borrow File 21

### 3.1.2.6 Activity Diagram - Borrow File 22

### 3.1.3 Returning a File Use Case 23

### 3.1.3.1 Brief Description 23

### 3.1.3.2 Flow of Events 23

### 3.1.3.2.1 Basic Flow - Returning a File 23

### 3.1.3.2.2 Alternative Flow 23

#### 3.1.3.2.2.1 Return file by "Return File" link 23

#### 3.1.3.2.2.2 The file is occupied by other personnel 24

### 3.1.3.3 Pre-conditions 24

### 3.1.3.4 Post-conditions 24

### 3.1.3.5 Sequence Diagram - Returning a File (basic flow) 24

### 3.1.3.6 Sequence Diagram - Return File by

### "Return File" Link (alt. flow) 25

### 3.1.3.7 Activity Diagram - Returning a File (basic flow) 26

### 3.1.3.8 Activity Diagram - Return File by

### "Return File" Link (alt. flow) 27

### 3.1.4 Produce Report by File Category Use Case 28

### 3.1.4.1 Brief Description 28

### 3.1.4.2 Flow of Events 28

### 3.1.4.2.1 Basic Flow - Produce Report by File Category 28

### 3.1.4.2.2 Alternative Flow 28

#### 3.1.4.2.2.1 "Produce Report" button is inactive 28

### 3.1.4.3 Pre-conditions 28

### 3.1.4.4 Post-conditions 29

### 3.1.4.5 Sequence Diagram - Produce Report by File Category 29

### 3.1.4.6 Activity Diagram - Produce Report by File Category 30

### 3.1.5 Produce Report by File Type Use Case 31

### 3.1.5.1 Brief Description 31

### 3.1.5.2 Flow of Events 31

### 3.1.5.2.1 Basic Flow - Produce Report by File Type 31

### 3.1.5.2.2 Alternative Flow 31

#### 3.1.5.2.2.1 "Produce Report" button is inactive 31

### 3.1.5.3 Pre-conditions 31

### 3.1.5.4 Post-conditions 31

### 3.1.5.5 Sequence Diagram - Produce Report by File Type 32

### 3.1.5.6 Activity Diagram - Produce Report by File Type 33

### 3.1.6 Produce Report by Borrower List Use Case 34

### 3.1.6.1 Brief Description 34

### 3.1.6.2 Flow of Events 34

### 3.1.6.2.1 Basic Flow - Produce Report by Borrower List 34

### 3.1.6.2.2 Alternative Flow 34

#### 3.1.6.2.2.1 "Produce Report" button is inactive 34

### 3.1.6.3 Pre-conditions 34

### 3.1.6.4 Post-conditions 35

### 3.1.6.5 Sequence Diagram - Produce Report by Borrower List 35

### 3.1.6.6 Activity Diagram - Produce Report by Borrower List 36

### 3.1.7 Managing File by Adding New File Use Case 37

### 3.1.7.1 Brief Description 37

### 3.1.7.2 Flow of Events 37

### 3.1.7.2.1 Basic Flow - Managing File by Adding New File 37

### 3.1.7.2.2 Alternative Flow 37

#### 3.1.7.2.2.1 The input details are insufficient 37

#### 3.1.7.2.2.2 The location is occupied by the other file 37

### 3.1.7.3 Pre-conditions 38

### 3.1.7.4 Post-conditions 38

### 3.1.7.5 Sequence Diagram - Managing File by Adding New File 38

### 3.1.7.6 Activity Diagram - Managing File by Adding New File 39

### 3.1.8 Managing File by Edit Existing File Use Case 40

### 3.1.8.1 Brief Description 40

### 3.1.8.2 Flow of Events 40

### 3.1.8.2.1 Basic Flow - Managing File by Edit Existing File 40

### 3.1.8.2.2 Alternative Flow 40

#### 3.1.8.2.2.1 The input details are insufficient 40

#### 3.1.8.2.2.2 The location is occupied by the other file 40

### 3.1.8.3 Pre-conditions 41

### 3.1.8.4 Post-conditions 41

### 3.1.8.5 Sequence Diagram - Managing File by Edit Existing File 41

### 3.1.8.6 Activity Diagram - Managing File by Edit Existing File 42

### 3.1.9 Managing File by Removing Existing File Use Case 43

### 3.1.9.1 Brief Description 43

### 3.1.9.2 Flow of Events 43

### 3.1.9.2.1 Basic Flow - Managing File by Removing Existing File 43

### 3.1.9.2.2 Alternative Flow 43

### 3.1.9.3 Pre-conditions 43

### 3.1.9.4 Post-conditions 44

### 3.1.9.5 Sequence Diagram - Managing File by Removing Existing File 44

### 3.1.9.6 Activity Diagram - Managing File by Removing Existing File 45

### 3.1.10 View File Borrowed History Use Case 46

### 3.1.10.1 Brief Description 46

### 3.1.10.2 Flow of Events 46

### 3.1.10.2.1 Basic Flow - View File Borrowed History 46

### 3.1.10.2.2 Alternative Flow 46

#### 3.1.10.2.2.1 The file selected does not has any borrowed history 46

### 3.1.10.3 Pre-conditions 46

### 3.1.10.4 Post-conditions 47

### 3.1.10.5 Sequence Diagram - View File Borrowed History 47

### 3.1.10.6 Activity Diagram - View File Borrowed History 48

### 3.1.11 Produce Checklist by Date of Edited Use Case 49

### 3.1.11.1 Brief Description 49

### 3.1.11.2 Flow of Events 49

### 3.1.11.2.1 Basic Flow -Produce Checklist by

### Date of Edited 49

### 3.1.11.2.2 Alternative Flow 49

#### 3.1.11.2.2.1 No file edited history in the database between the date intervals entered 49

### 3.1.11.3 Pre-conditions 49

### 3.1.11.4 Post-conditions 50

### 3.1.11.5 Sequence Diagram - Produce Checklist by

### Date of Edited 50

### 3.1.11.6 Activity Diagram - Produce Checklist by Date of Edited 51

### 3.1.12 Produce Checklist by File Location Use Case 52

### 3.1.12.1 Brief Description 52

### 3.1.12.2 Flow of Events 52

### 3.1.12.2.1 Basic Flow -Produce Checklist by File Location 52

### 3.1.12.2.2 Alternative Flow 52

#### 3.1.12.2.2.1 No file in the cabinet 52

### 3.1.12.3 Pre-conditions 52

### 3.1.12.4 Post-conditions 53

### 3.1.12.5 Sequence Diagram - Produce Checklist by File Location 53

### 3.1.12.6 Activity Diagram - Produce Checklist by File Location 54

## 3.2 NON-FUNCTIONAL REQUIREMENT 55

### 3.2.1 Usability 55

### 3.2.2 Security 55

### 3.2.3 Reliability 55

# 4. APPENDICES 56

**1. INTRODUCTION**

**1.1 Introduction on the Domain Application**

The **File Searching System** will feature the library domain application, as it can helps the organization to search for a physical file (paper file) from the cabinet more effectively and efficiently.

The client requires the product because they have problems dealing with the file searching in their organization.

**1.2 Analysis of Existing System**

**1.2.1 Background**

In the current system, the users have to search the file using a log book that contains the information of the files (serial number and file’s name). The files are keep inside the drawer of the cabinet. Each drawer has a ranged serial number. Inside the drawer contains many slots that are arranged in order according to the numbers that is shown at the outside of the drawer. Each slot contains only one file.

Firstly, the users have to know the name of the file they want to search. There are serial numbers besides every file’s name in the log book. Then, the users will based on that serial number to search for the drawer of the cabinet that contain the serial number of the file. Lastly, the requested file can be found in the slot that has the same serial number with the file.

**1.2.2 Problem Identified**

These are the issues that are identified in the current system in the organization.

***1.2.2.1 Non-computerized system.***

Search the file manually, and currently depends on only one personnel to search for the requested file.

***1.2.2.2 Not secure.***

Everyone can access any files.

***1.2.2.3 Easy to lost track of the files.***

The person who borrowed the file is not recorded, so it is very hard to track the file. The probability for the missing file is high.

***1.2.2.4 Inefficient.***

Slow and take time to search for the file.

**1.3 Scope**

The new system will become a stand-alone and computerized system, as it does not need to interact with other applications and using its own database. Only registered personnel (employee, manager, file manager) can use the system.

Every personnel can search for the file location to borrow or return the file.

Manager can produce report based on the number of each file category, or file type, as well as to generate report of the current borrower of the file.

File manager can add, edit and remove file by using the system. He/she can also generated a list of checklist to help in organizing the file cabinet. Besides that, the file manager can view the borrowed history of every file to trace all the former and current borrower of the file.

**1.4 Glossary**

|  |  |
| --- | --- |
| **“File”** | The physical file that is kept in the cabinet. |
| **“Serial number”** | The unique number of each file. |
| **“Closed-file”** | The file that already reach 100 pages, but is not discarded. It is stored in the closed-file cabinet. |
| **“File type”** | The type of file in the organization, which include public file and private file.  Public file is the file that can be viewed by every personnel in the organization, while private file is the file that can be viewed by “manger” and “file manager” personnel only. |
| **“File category”** | The category of file, which include employee record, company report, and closed-file. |
| **“Employee”** | The registered low-level management personnel in the organization, which can borrow public file type. |
| **“Manager”** | The registered higher-level management personnel in the organization, which can borrow any file type, as well as produce the report from the system. |
| **“File manager”** | The registered specialized personnel in the organization that is responsible to manage this system, which can add, edit and remove the file information, as well as produce the checklist and view file borrow history. |

**1.5 References**

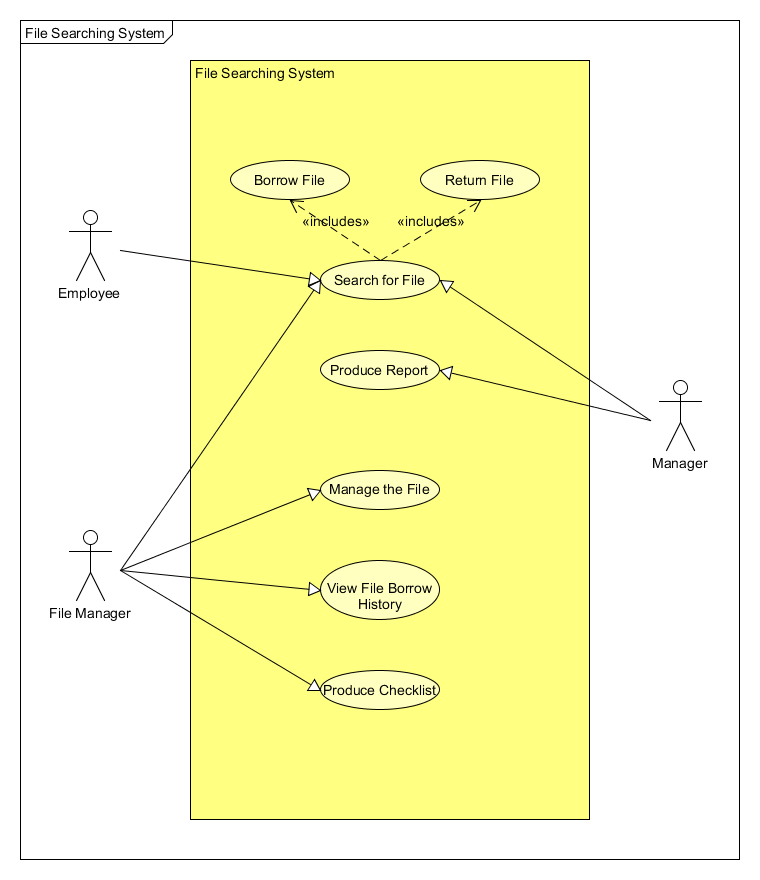
Background & roles of LPPJB*. Official Portal of Farmer’s Organization Authority*. Retrieve from http://www.lpp.gov.my/en/farmers-organization-introduction.

Gary B. Shelly & Harry J. Rosenblatt (2012). *Analysis and Design for System, Ninth Edition*. United States: Course Technology, Cengage Learning, p. 52.

Kenneth E. Kendall & Julie E. Kendall (2010). *System Analysis and Design, 8/E*. New Jersey: Prentice Hall, p. 56.

**1.6 Overview**

This is the use case diagram of the system. The employee and the manager will become the main user of the system. The system is maintained by the file manager, which basically manages the file of the system.



**2. OVERALL DESCRIPTION**

**2.1 Product Perspective**

The **File Searching System** is a stand-alone system. Records and login information are retrieved by its own database. To be able to login to the system, every personnel will use the username and password provided by the file manager.

**2.2 Product Functional Requirement**

**2.2.1 Search for File**

This use case allow the user (employee, manager, and file manager) to search for the file that is stored in the cabinet.

**2.2.2 Borrow File**

This use case allows the users (employees, file manager and manager) to borrow file by searching the file based on the file name, file id or file type.

**2.2.3 Returning a File**

This use case allows the users (employees, file manager and manager) to return file by searching the file based on the file name, file id or file type.

**2.2.4 Produce Report by File Category**

This use case allow the manager to produce a report based on the file category. There are three type of file category, which are “Employee Record”, “Company Report”, and “Closed-file”.

**2.2.5 Produce Report by File Type**

This use case allow the manager to produce a report based on the file type. There are two type of file type, which are “Private File” and “Public File”.

**2.2.6 Produce Report by Borrower List**

This use case allow the manager to produce a report based on the borrower list. The borrower(s) in the list is/are current borrower(s) of the file.

**2.2.7 Managing File by Adding New File**

This use case allow the file manager to add file information of the new file that is added into the cabinet, and help them to reduce the file location redundancy.

**2.2.8 Managing File by Edit Existing File**

This use case allow the file manager to edit the file information of the existing file that is stored in the cabinet.

**2.2.9 Managing File by Remove Existing File**

This use case allow the file manager to remove the file information of the existing file that is stored in the cabinet.

**2.2.10 View File Borrowed History**

This use case allow the file manager to view the file borrowed history of the existing file that is stored in the cabinet.

**2.2.11 Produce Checklist by Date of Edited**

This use case allow the file manager to produce a checklist based on the date he/she edited the information of the file. If the location of a file is changed, the file manager can arrange that file in the cabinet according to the checklist. It is also used to trace the file edited history.

**2.2.12 Produce Checklist by File Location**

This use case allow the file manager to produce a checklist based on the file location. This allow file manager to check the status of the file (such as available, borrowed, or exceed borrow date).

**2.3 User Characteristics**

There are three type users of the **File Searching System**.

**2.3.1 Employee**

Characteristics:

1) The registered lower-level management personnel in the organization.

2) Can only borrow and return file.

3) Can view and borrow public file type only.

**2.3.2 Manager**

Characteristics:

1) The registered higher-level management personnel in the organization.

2) Can borrow and return file, as well as produce report.

3) Can view both public and private file type.

4) Involves in decision-making process.

**2.3.3 File Manager**

1) The registered specialized personnel in the organization that is responsible to manage the system.

2) Can borrow and return file, as well as produce checklist and view borrow history.

3) Can view both public and private file type.

4) Can add, edit and remove file information from the database.

5) Responsible to organize the file in the cabinet, such as arrange the file in the cabinet, move the file to closed-file cabinet, find out who has lost a page of a file, etc.

**2.4 General Constraints, Assumptions, Dependencies, Guidelines**

**2.4.1 General Constraints**

***2.4.1.1 Browser used.***

The system can work correctly on these browsers: Google Chrome 10 and above, Firefox 5 and above.

***2.4.1.2 Server used.***

The system can work correctly on WampServer 2.0 and above, which provide local server to the system.

***2.4.1.3 Database used.***

The system can work correctly on phpMyAdmin 3.1.1 and above, which provide the database to the system.

**2.4.2 Assumptions**

***2.4.2.1 Working time.***

The File Searching System will be available 8 hours a day, 5 days a week, which is the operation hours of the organization. There shall no more than 8% down time.

***2.4.2.2 Amount of user at a time.***

The system is used by a user at a time.

**2.4.3 Dependencies**

***2.4.3.1 Local server.***

The organization does not has its own server, so the system has to depend on the WampServer install in the desktop computer.

***2.4.3.2 Backup storage.***

Seems the database can be stored in a device only, the backup process has to be done frequently. Any downtime involves in the device will probably cause the database lost in the system.

***2.4.3.3 One device.***

The organization only provide 1 desktop computer for the system.

**2.4.4 Guidelines**

**User Manual,** which contains the guidelines and overall detailed functions of the new **File Searching System** is included in the form of a document manual. This includes helps to solve the problems that will possible arise in the system.

**3. SPECIFIC REQUIREMENT**

**3.1 Software Product Features (Functional Requirement)**

**3.1.1 Search for File Use Case**

**3.1.1.1 Brief Description**

This use case allow the user (employee, manager, and file manager) to search for the file that is stored in the cabinet.

**3.1.1.2 Flow of Events**

The use case begins when the user clicks on the “Search File” link.

**3.1.1.2.1 Basic Flow – Search the File**

3.1.1.2.1.1 The user input the details of file information (such as file name, file id, and file category) in the input form of “Search File” page.

3.1.1.2.1.2 Once the user clicks on the “Search” button, the system will direct the user to “Search Result” page with a table list of related file, which contain the index, file name, file id, and file category.

3.1.1.2.1.3 The user clicks on the file name link.

**3.1.1.2.2 Alternative Flow**

***3.1.1.2.2.1 No file found in the database.***

3.1.1.2.2.1.1 If the details entered do not exist in the database, the system will display an appropriate message to tell the user that no such file exist in the cabinet.

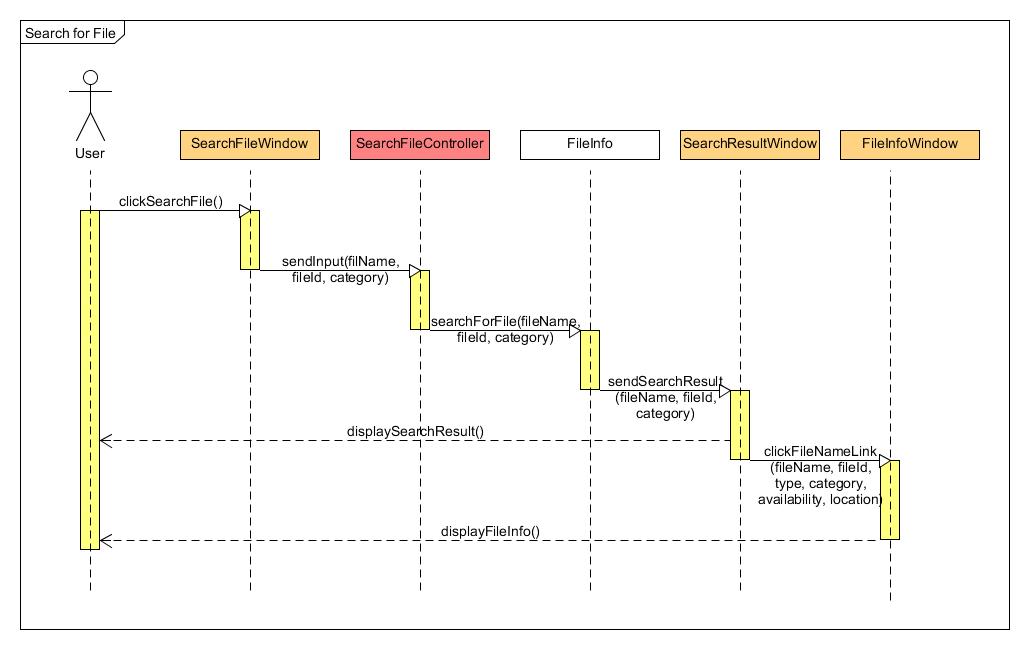
3.1.1.2.2.1.2 The user clicks on the “Back” button to return to the “Search File” page to repeat the process in **3.1.1.2.1**.

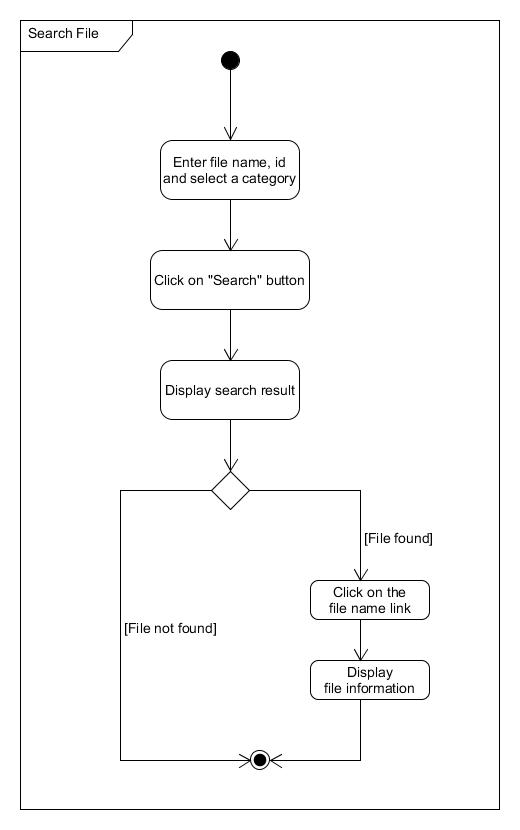
**3.1.1.3 Pre-conditions**

Before the use case begin, the user must logged in to the system.

**3.1.1.4 Post-conditions**

The system will direct the user to the “File Information” page, which display the information of the selected file.

**3.1.1.5 Sequence Diagram – Search for File**

**3.1.1.6 Activity Diagram – Search for File**

**3.1.2 Borrow File Use Case**

**3.1.2.1 Brief Description**

This use case allows the users (employees, file manager and manager) to borrow file by searching the file based on the file name, file id or file type.

**3.1.2.2 Flow of Events**

The use case begins when the user is in the “File Information” page of the selected file that need to borrow.

**3.1.2.2.1 Basic Flow – Borrow File**

3.1.2.2.1.1 After the user click on the “Borrow File” button, the system will display a message box to ask for the confirmation of the user.

3.1.2.2.1.2 The user click on “Yes” in the message box.

**3.1.2.2.2 Alternative Flow**

***3.1.2.2.2.1 File not available.***

3.1.2.2.2.1.1 If the file has already been borrowed by other user and the user clicks on the “Borrow” button, the system will display an appropriate message to tell the user that the file is not available.

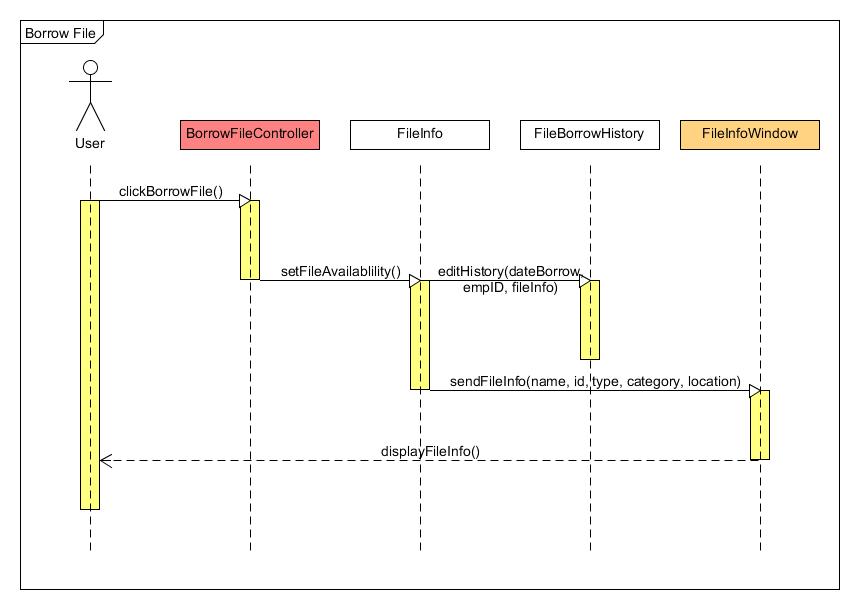
**3.1.2.3 Pre-conditions**

Before the use case begin, the user must logged in to the system, and already in the “File Information” page of the selected file, in this case is the file that user wants to borrow.

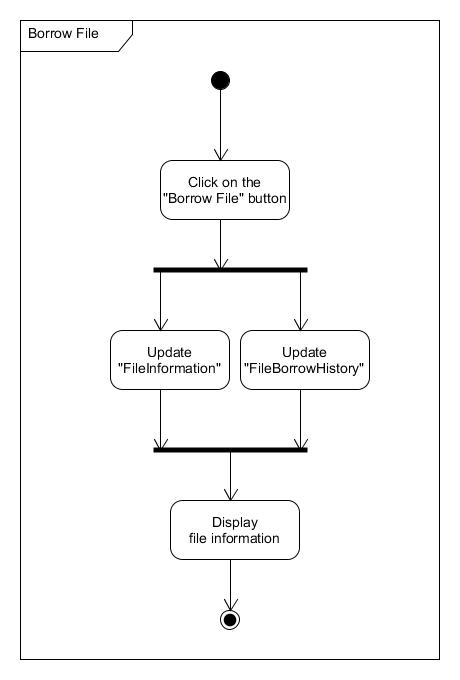
**3.1.2.4 Post-conditions**

The borrowed file will change the availability status to “Not available”, as well as update the borrower list and the file borrowed history.

**3.1.2.5 Sequence Diagram – Borrow File**

****

**3.1.2.6 Activity Diagram – Borrow File**

****

**3.1.3 Returning a File Use Case**

**3.1.3.1 Brief Description**

This use case allows the users (employees, file manager and manager) to return file by searching the file based on the file name, file id or file type.

**3.1.3.2 Flow of Events**

The use case begins when the user is in the “File Information” page of the selected file that need to return or clicks on the “Return File” link when in the home page.

**3.1.3.2.1 Basic Flow – Returning a File**

3.1.3.2.1.1 After the user click on the “Return File” button, the system will display a message box to ask for the confirmation of the user.

3.1.3.2.1.2 The user click on “Yes” in the message box.

**3.1.3.2.2 Alternative Flow**

***3.1.3.2.2.1 Return file by “Return File” link***

3.1.3.2.2.1.1 The page of “Rented File” will displays the table of file(s) that is currently rented by the user

3.1.3.2.2.1.2 After the user click on the file name link, it will be directed to “File Information” page that contains the details of the file selected.

3.1.3.2.2.1.3 The user can return the file by clicking on the “Return File” button. Then, the system will display a message box to ask for the confirmation of the user.

3.1.3.2.2.1.4 The user click on “Yes” in the message box.

***3.1.3.2.2.2 The file is occupied by other personnel.***

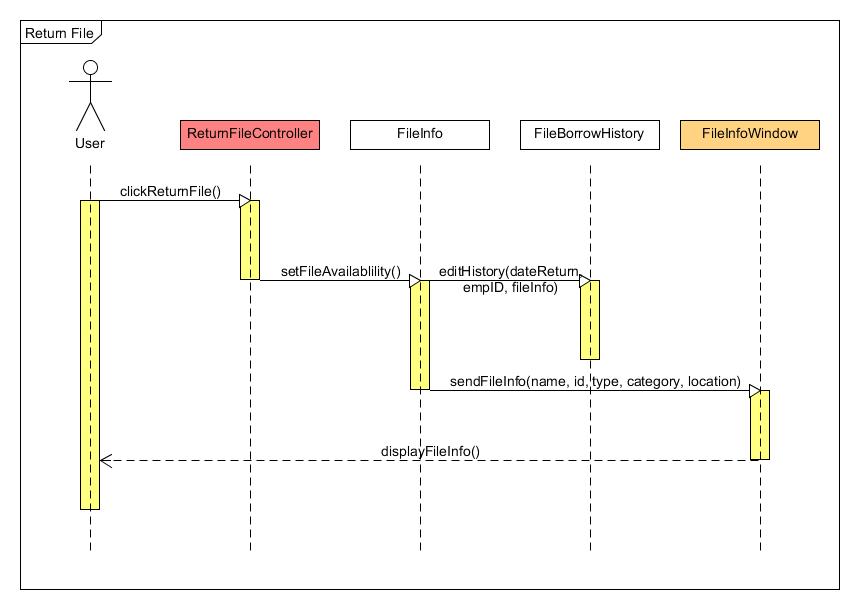
3.1.3.2.2.2.1 If the file is already borrowed by other personnel, the system will display an appropriate message to tell the user that the file is not available and cannot be returned.

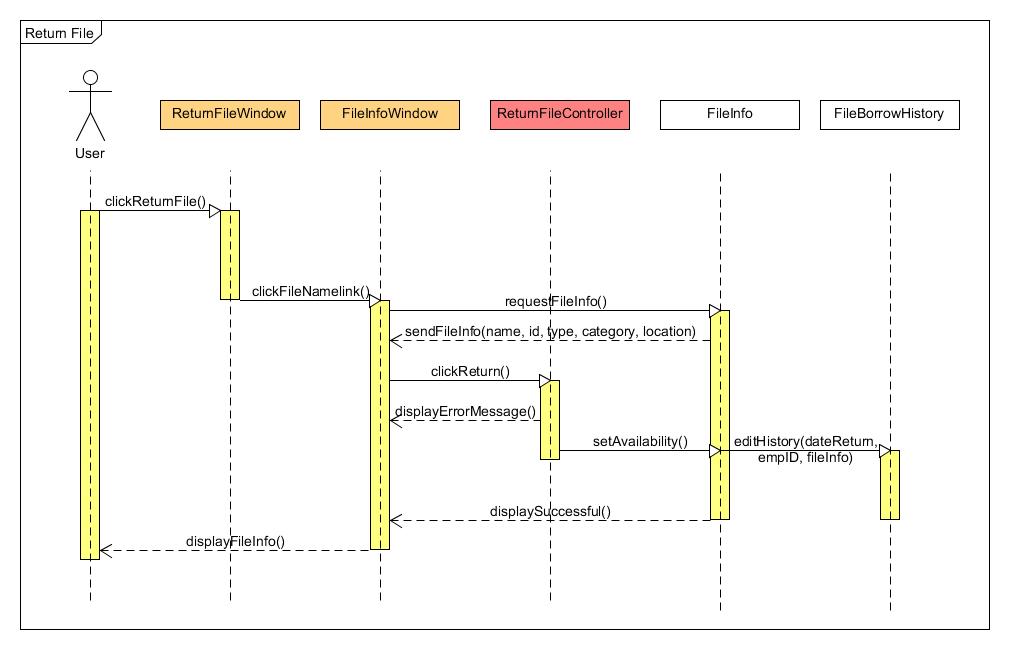
**3.1.3.3 Pre-conditions**

Before the use case begin, the user must logged in to the system, and already in the “File Information” page of the selected file, in this case is the file that user wants to borrow.

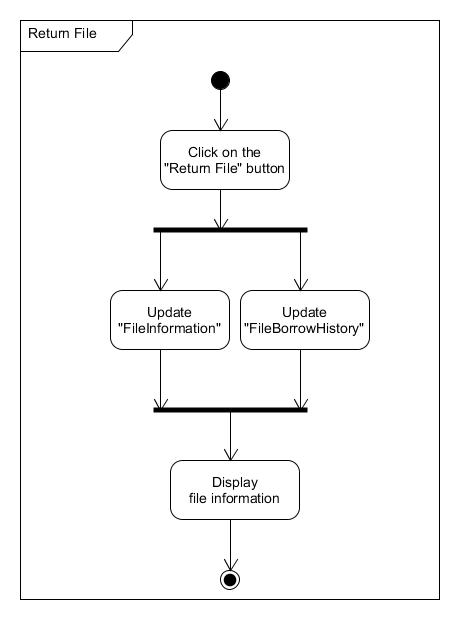
**3.1.3.4 Post-conditions**

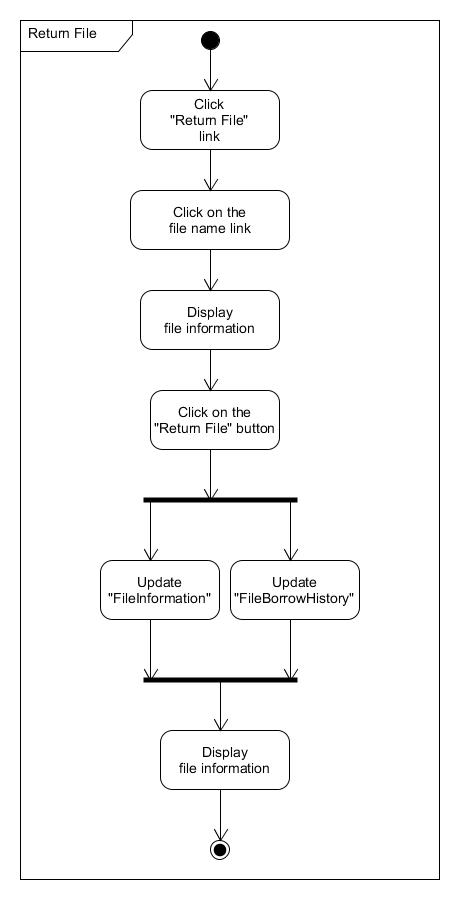
The returned file will change the availability status to “Available” ”, as well as update the borrower list and the file borrowed history.

**3.1.3.5 Sequence Diagram – Returning a File (basic flow)**

**3.1.3.6 Sequence Diagram – Return File by Return File link (alternative flow)**

**3.1.3.7 Activity Diagram – Returning a File by Search File link (basic flow)**

****

**3.1.3.8 Activity Diagram – Returning a File by Return File link (alternative flow)**

**3.1.4 Produce Report by File Category Use Case**

**3.1.4.1 Brief Description**

This use case allow the manager to produce a report based on the file category. There are three type of file category, which are “Employee Record”, “Company Report”, and “Closed-file”.

**3.1.4.2 Flow of Events**

The use case begins when the manager clicks on the “Produce Report” link.

**3.1.4.2.1 Basic Flow – Produce Report by File Category**

3.1.4.2.1.1 The manager selects “File Category” in the drop down box.

3.1.4.2.1.2 Once the manager clicks on the “Produce Report” button, a report of file category is created.

**3.1.4.2.2 Alternative Flow**

***3.1.4.2.2.1 “Produce Report” button is inactive.***

3.1.4.2.2.1.1 If the drop down box does not have any selected value, the system will not activate the “Produce Report” button.

3.1.4.2.2.1.2 The manager clicks on the drop down box to repeat the process in **3.1.4.2.1**.

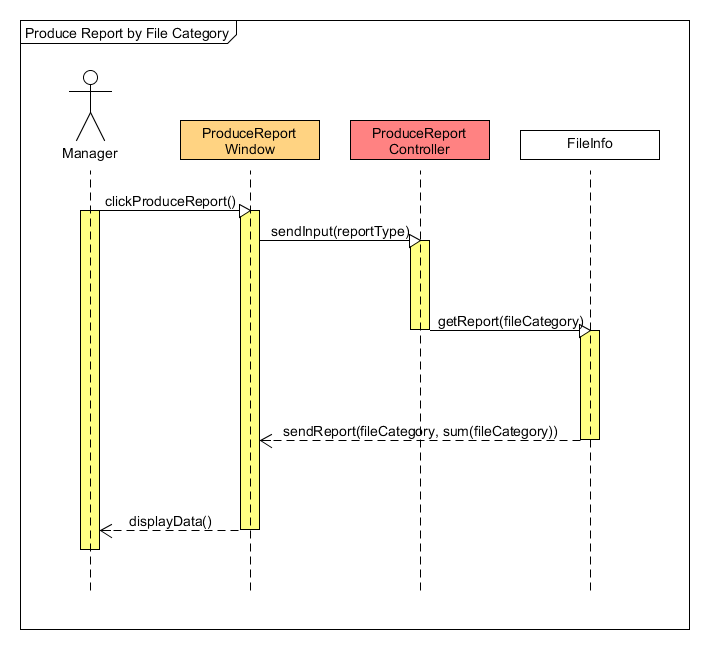
**3.1.4.3 Pre-conditions**

Before the use case begin, the manager must logged in to the system.

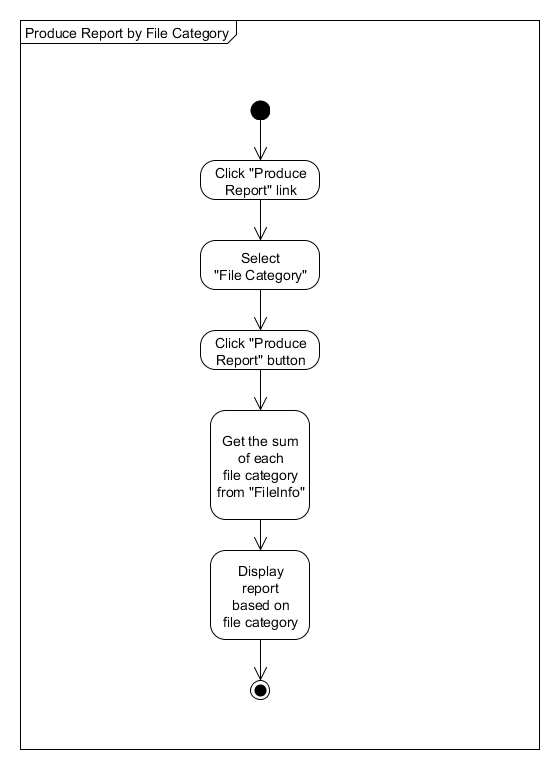
**3.1.4.4 Post-conditions**

The report based on the file category is produced, which contain the amount of file for each file category in a list of table.

**3.1.4.5 Sequence Diagram – Produce Report by File Category**



**3.1.4.6 Activity Diagram – Produce Report by File Category**



**3.1.5 Produce Report by File Type Use Case**

**3.1.5.1 Brief Description**

This use case allow the manager to produce a report based on the file type. There are two type of file type, which are “Private File” and “Public File”.

**3.1.5.2 Flow of Events**

The use case begins when the manager clicks on the “Produce Report” link.

**3.1.5.2.1 Basic Flow – Produce Report by File Type**

3.1.5.2.1.1 The manager selects “File Type” in the drop down box.

3.1.5.2.1.2 Once the manager clicks on the “Produce Report” button, a report of file type is created.

**3.1.5.2.2 Alternative Flow**

***3.1.5.2.2.1 “Produce Report” button is inactive.***

3.1.5.2.2.1.1 If the drop down box does not have any selected value, the system will not activate the “Produce Report” button.

3.1.5.2.2.1.2 The manager clicks on the drop down box to repeat the process in **3.1.5.2.1**.

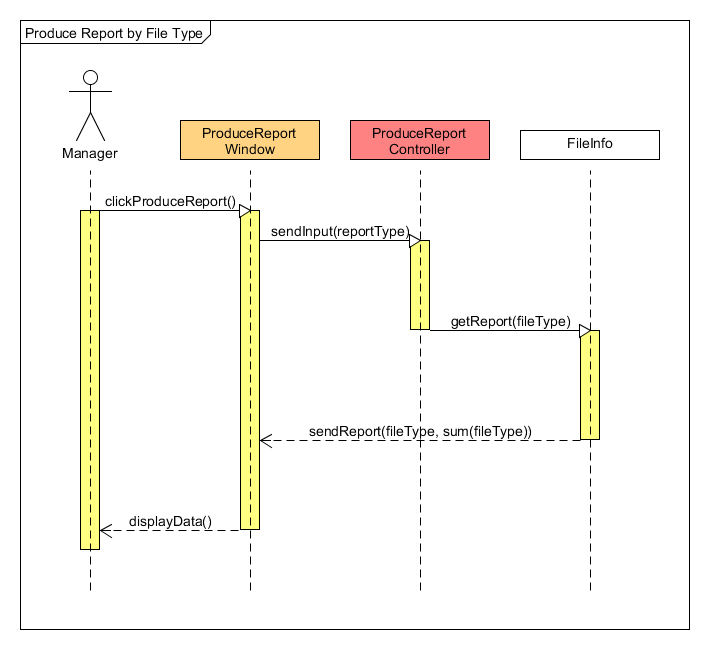
**3.1.5.3 Pre-conditions**

Before the use case begin, the manager must logged in to the system.

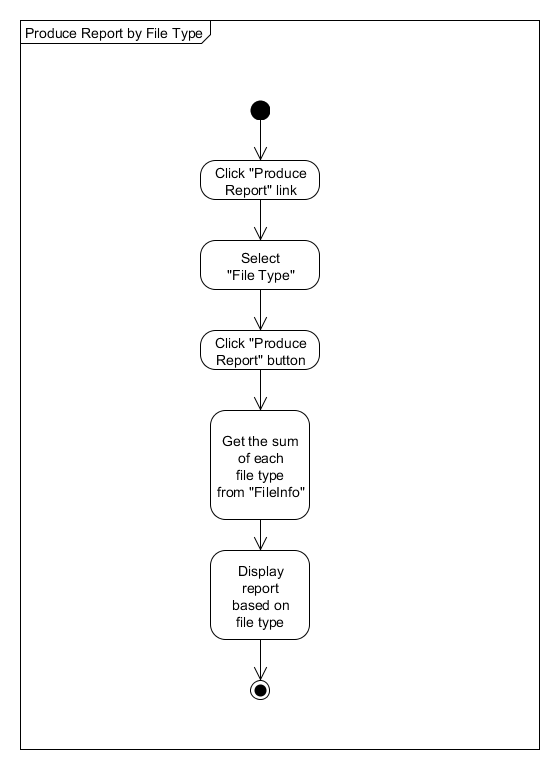
**3.1.5.4 Post-conditions**

The report based on the file type is produced, which contain the amount of file for each file type in a list of table.

**3.1.5.5 Sequence Diagram – Produce Report by File Type**



**3.1.5.6 Activity Diagram – Produce Report by File Type**

****

**3.1.6 Produce Report by Borrower List Use Case**

**3.1.6.1 Brief Description**

This use case allow the manager to produce a report based on the borrower list. The borrower(s) in the list is/are current borrower(s) of the file.

**3.1.6.2 Flow of Events**

The use case begins when the manager clicks on the “Produce Report” link.

**3.1.6.2.1 Basic Flow – Produce Report by Borrower List**

3.1.6.2.1.1 The manager selects “Borrower List” in the drop down box.

3.1.6.2.1.2 Once the manager clicks on the “Produce Report” button, a report of borrower list is created.

**3.1.6.2.2 Alternative Flow**

***3.1.6.2.2.1 “Produce Report” button is inactive.***

3.1.6.2.2.1.1 If the drop down box does not have any selected value, the system will not activate the “Produce Report” button.

3.1.6.2.2.1.2 The manager clicks on the drop down box to repeat the process in **3.1.6.2.1**.

***3.1.6.2.2.2 No current borrower in the database.***

3.1.6.2.2.2.1 If all the borrower has return the file, the system will display an appropriate message to tell the manager that there are no borrower in the list.

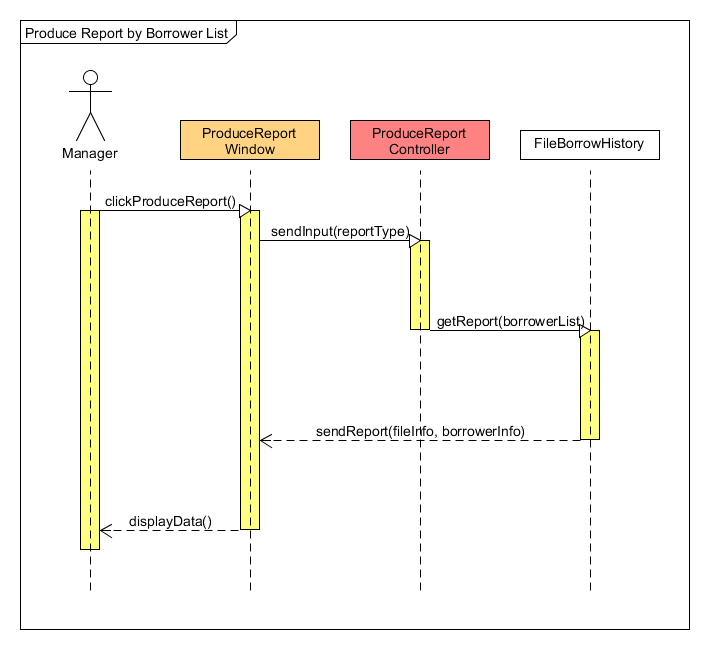
**3.1.6.3 Pre-conditions**

Before the use case begin, the manager must logged in to the system.

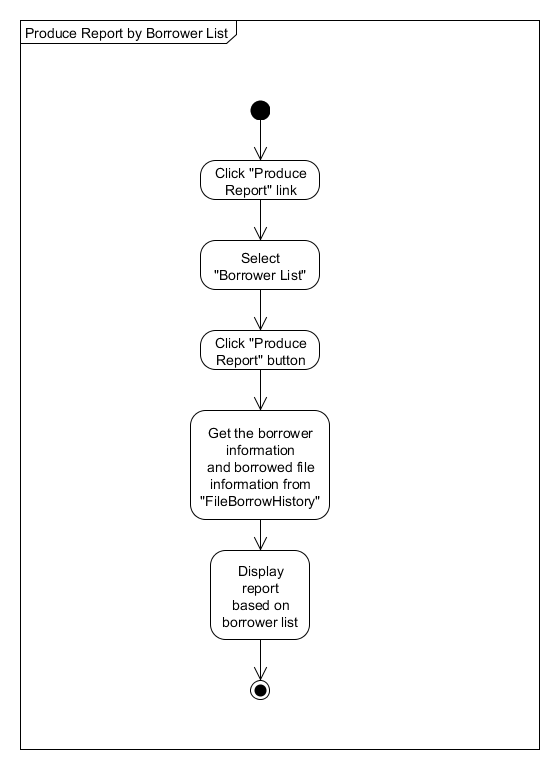
**3.1.6.4 Post-conditions**

The report based on the borrower list is produced, which contain the file name, borrower name, borrow date and return date for each borrowed file in a list of table.

**3.1.6.5 Sequence Diagram – Produce Report by Borrower List**



**3.1.6.6 Activity Diagram – Produce Report by Borrower List**

****

**3.1.7 Managing File by Adding New File Use Case**

**3.1.7.1 Brief Description**

This use case allow the file manager to add file information of the new file that is added into the cabinet, and help them to reduce the file location redundancy.

**3.1.7.2 Flow of Events**

The use case begins when the file manager clicks on the “Add New File” link.

**3.1.7.2.1 Basic Flow – Managing File by Adding New File**

3.1.7.2.1.1 The file manager input the details of file information (such as file name, file id, file type, file category, and location – cabinet number, drawer number and slot number) in the input “Add File” form.

3.1.7.2.1.2 The file manager clicks on the “Add New” button.

**3.1.7.2.2 Alternative Flow**

***3.1.7.2.2.1 The input details are insufficient.***

3.1.7.2.2.1.1 If the details input are not sufficient and incomplete, the system will display an appropriate message box to notice the file manager to complete the input form.

3.1.7.2.2.1.2 The file manager clicks on the “Ok” button to close the message box, and repeat the process in **3.1.7.2.1** to complete the detail of the new file.

***3.1.7.2.2.2 The location is occupied by the other file.***

3.1.7.2.2.2.1 If the location details of the new file is already exist in the database, the system will display an appropriate message box to notice the file manager that the location is already been occupied by other file.

3.1.7.2.2.2.2 The file manager clicks on the “Ok” button to close the message box, and repeat the process in **3.1.7.2.1** to change the location of the new file.

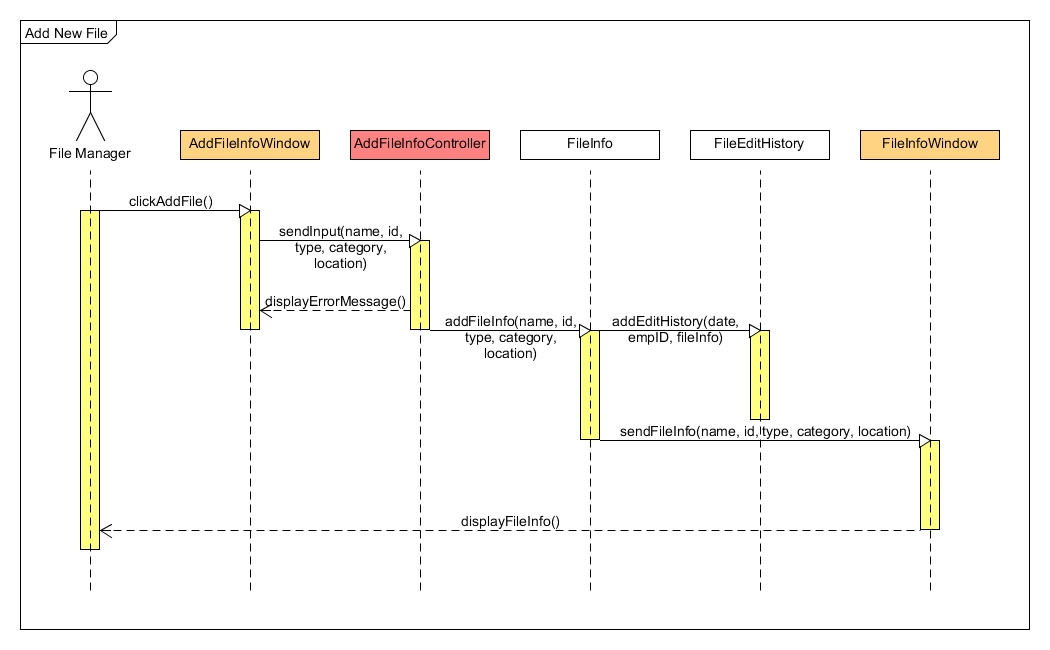
**3.1.7.3 Pre-conditions**

Before the use case begin, the file manager must logged in to the system.

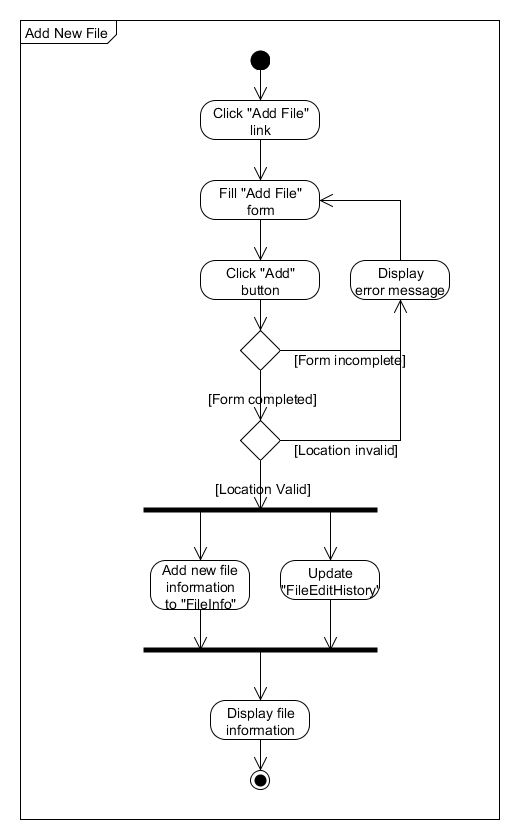
**3.1.7.4 Post-conditions**

The system will direct the file manager to the “File Information” page, which display the information of the newly created file, and the file information is added into the database.

**3.1.7.5 Sequence Diagram – Managing File by Adding New File**



**3.1.7.6 Activity Diagram – Managing File by Adding New File**



**3.1.8 Managing File by Edit Existing File Use Case**

**3.1.8.1 Brief Description**

This use case allow the file manager to edit the file information of the existing file that is stored in the cabinet.

**3.1.8.2 Flow of Events**

The use case begins when the file manager is in the “File Information” page of the selected file that need to edit.

**3.1.8.2.1 Basic Flow – Managing File by Edit Existing File**

3.1.8.2.1.1 The file manager clicks on the “Edit File” button, which directed the file manager to the “Edit File” page of the selected file.

3.1.8.2.1.2 The file manager changes the details of file information.

3.1.8.2.1.3 Then, the file manager clicks “Ok” button to finish the edit process.

**3.1.8.2.2 Alternative Flow**

***3.1.8.2.2.1 The input details are insufficient.***

3.1.8.2.2.1.1 If the details input are not sufficient and incomplete, the system will display an appropriate message box to notice the file manager to complete the input form.

3.1.8.2.2.1.2 The file manager clicks on the “Ok” button to close the message box, and repeat the process in **3.1.8.2.2** to complete the details of the file.

***3.1.8.2.2.2 The location is occupied by the other file.***

3.1.8.2.2.2.1 If the new location details of the edited file is already exist in the database, the system will display an appropriate message box to notice the file manager that the location is already been occupied by other file.

3.1.8.2.2.2.2 The file manager clicks on the “Ok” button to close the message box, and repeat the process in **3.1.8.2.2** to change the location of the edited file.

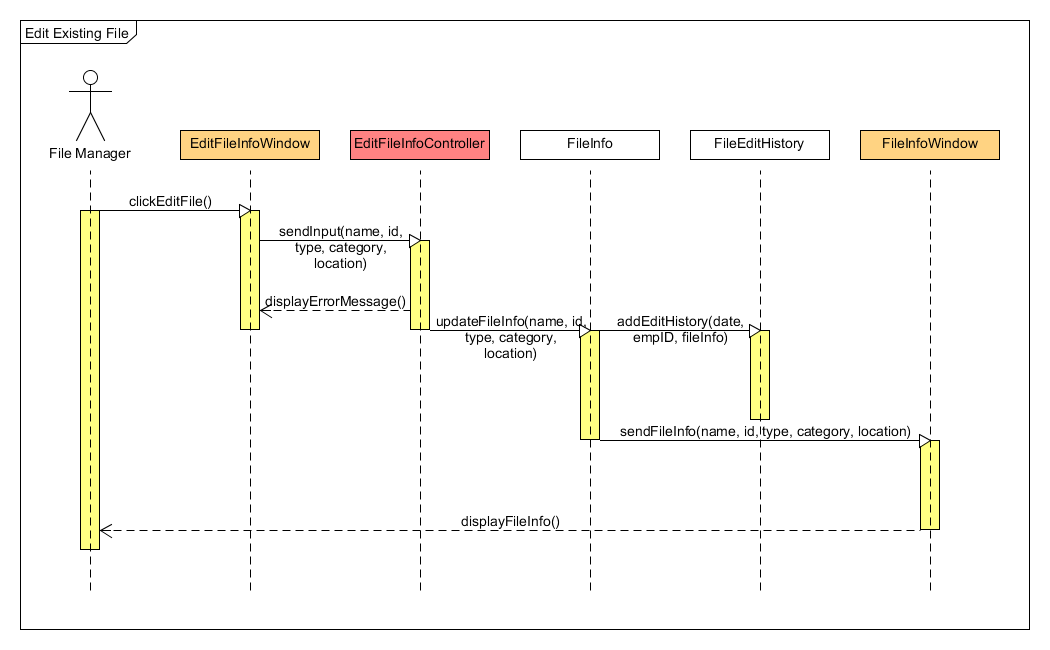
**3.1.8.3 Pre-conditions**

Before the use case begin, the file manager must logged in to the system, and already in the “File Information” page of the selected file, in this case is the file that the file manager wants to edit.

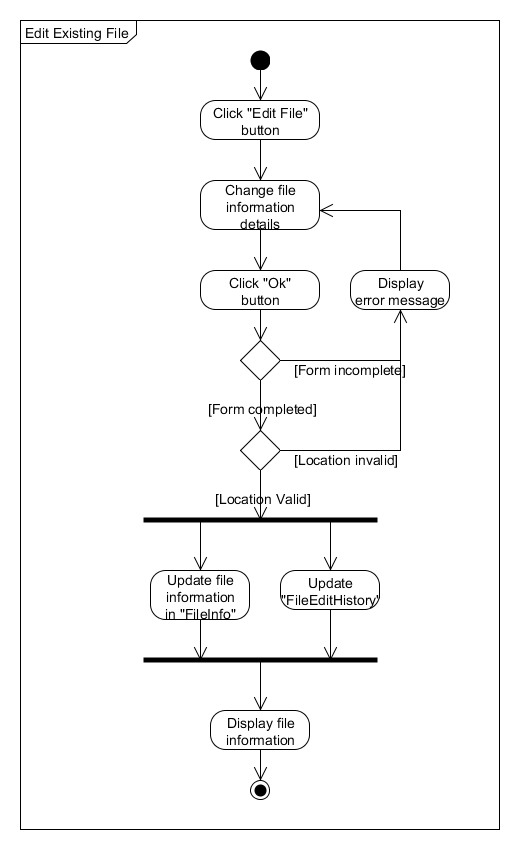
**3.1.8.4 Post-conditions**

The system will direct the file manager to the “File Information” page, which display the information of the newly edited file, and the updated file information, as well as the file edited history are updated and are added respectively into the database.

**3.1.8.5 Sequence Diagram – Managing File by Edit Existing File**



**3.1.8.6 Activity Diagram – Managing File by Edit Existing File**



**3.1.9 Managing File by Remove Existing File Use Case**

**3.1.9.1 Brief Description**

This use case allow the file manager to remove the file information of the existing file that is stored in the cabinet.

**3.1.9.2 Flow of Events**

The use case begins when the file manager is in the “File Information” page of the selected file that need to remove.

**3.1.9.2.1 Basic Flow – Managing File by Remove Existing File**

3.1.9.2.1.1 The file manager clicks on the “Remove File” button.

3.1.9.2.1.2 A message box will be output to ask for the conformation of the file manager.

3.1.9.2.1.3 Then, the file manager clicks “Yes” button to finish the removal process.

**3.1.9.2.2 Alternative Flow**

There are no abnormal event. If it exists, such as can’t remove the file information, it is considered a bug, and need to consult the software developer.

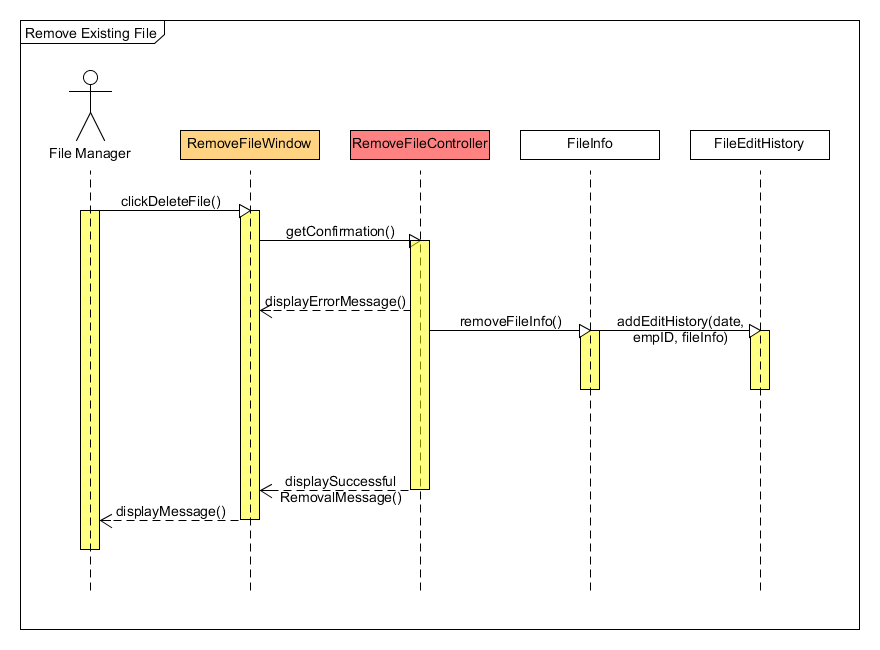
**3.1.9.3 Pre-conditions**

Before the use case begin, the file manager must logged in to the system, and already in the “File Information” page of the selected file, in this case is the file that the file manager wants to remove.

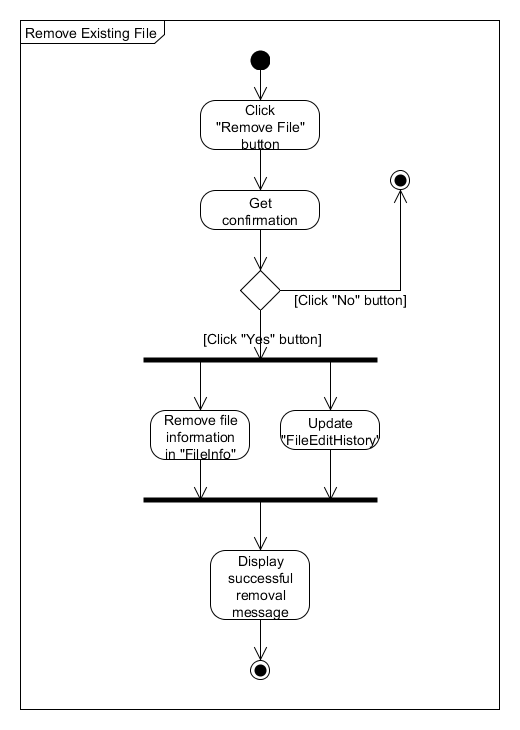
**3.1.9.4 Post-conditions**

The system will output a message box which tell the file manager that the removal process is successful and direct the file manager to the “Home” page. The information of the selected file will be removed and the file edited history is updated in the database.

**3.1.9.5 Sequence Diagram – Managing File by Remove Existing File**



**3.1.9.6 Activity Diagram – Managing File by Remove Existing File**



**3.1.10 View File Borrowed History Use Case**

**3.1.10.1 Brief Description**

This use case allow the file manager to view the file borrowed history of the existing file that is stored in the cabinet.

**3.1.10.2 Flow of Events**

The use case begins when the file manager is in the “File Information” page of the selected file that need to view its history.

**3.1.10.2.1 Basic Flow – View File Borrowed History**

3.1.10.2.1.1 The file manager clicks on the “View Borrowed History” button.

**3.1.10.2.2 Alternative Flow**

***3.1.10.2.2.1 The file selected does not has any borrowed history.***

3.1.10.2.2.1.1 If the file stored in the cabinet is not touched by any employees at all, the system will display an appropriate message to tell the file manager that the file does not have any borrowed history.

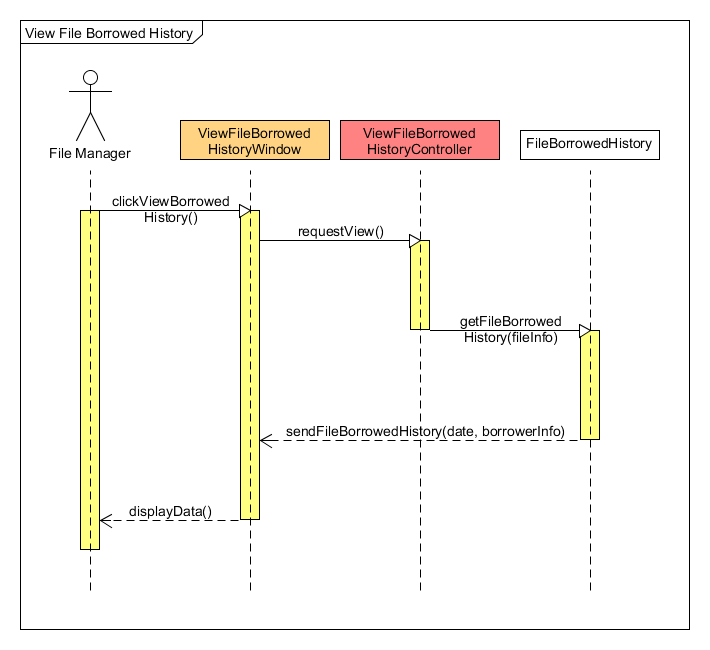
**3.1.10.3 Pre-conditions**

Before the use case begin, the file manager must logged in to the system, and already in the “File Information” page of the selected file, in this case is the file that the file manager wants to view its history.

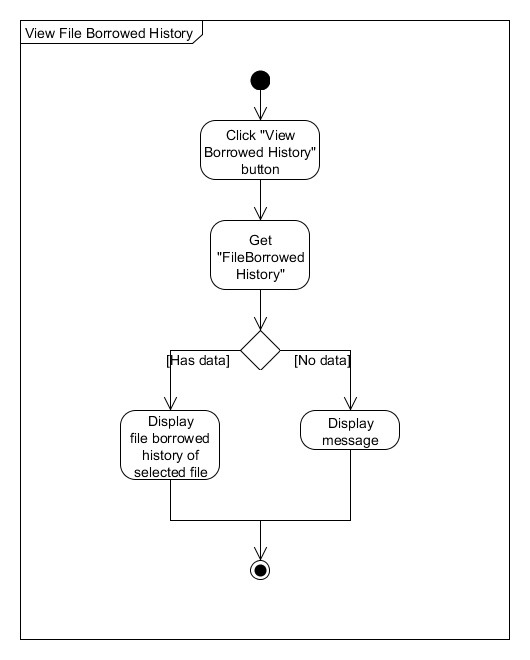
**3.1.10.4 Post-conditions**

The system will direct the file manager to the “File Borrowed History” page, which display the information in the form of table that contains the borrower name of the selected file, and the date of borrowed by each borrower.

**3.1.10.5 Sequence Diagram – View File Borrowed History**



**3.1.10.6 Activity Diagram – View File Borrowed History**



**3.1.11 Produce Checklist by Date of Edited Use Case**

**3.1.11.1 Brief Description**

This use case allow the file manager to produce a checklist based on the date he/she edited the information of the file. If the location of a file is changed, the file manager can arrange that file in the cabinet according to the checklist. It is also used to trace the file edited history.

**3.1.11.2 Flow of Events**

The use case begins when the file manager clicks on the “Produce Checklist” link.

**3.1.11.2.1 Basic Flow – Produce Checklist by Date of Edited**

3.1.11.2.1.1 The file manager selects “Date Edit” in the drop down box.

3.1.11.2.1.2 The file manager choose the date of file edited in the date boxes “Date from” and “Date to”.

3.1.11.2.1.3 Once the file manager clicks on the “Produce Checklist” button, a checklist of file(s) between the date intervals is created.

**3.1.11.2.2 Alternative Flow**

***3.1.11.2.2.1 No file edited history in the database between the date intervals entered.***

3.1.11.2.2.1.1 If the date intervals entered does not have the record or the history, the system will display an appropriate message to the file manager.

3.1.11.2.2.1.2 The file manager clicks on the “Back” button to repeat the process in **3.1.11.2.1**.

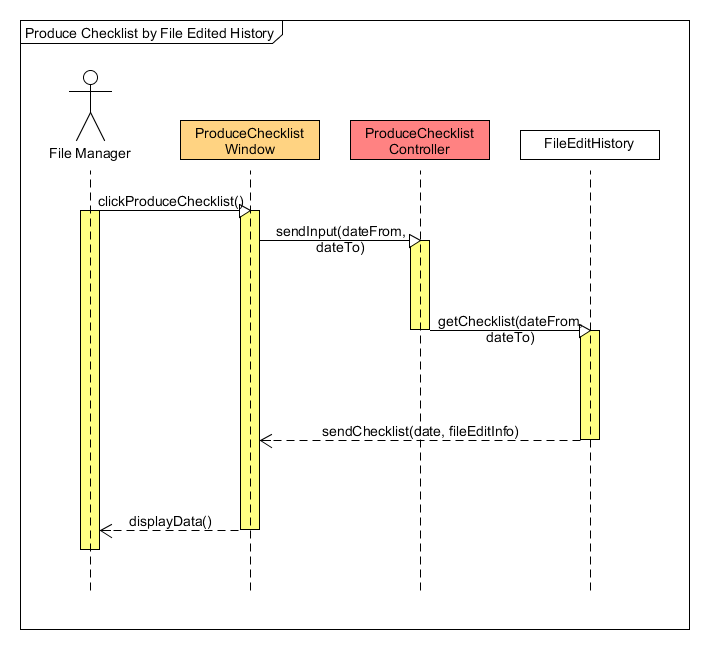
**3.1.11.3 Pre-conditions**

Before the use case begin, the file manager must logged in to the system.

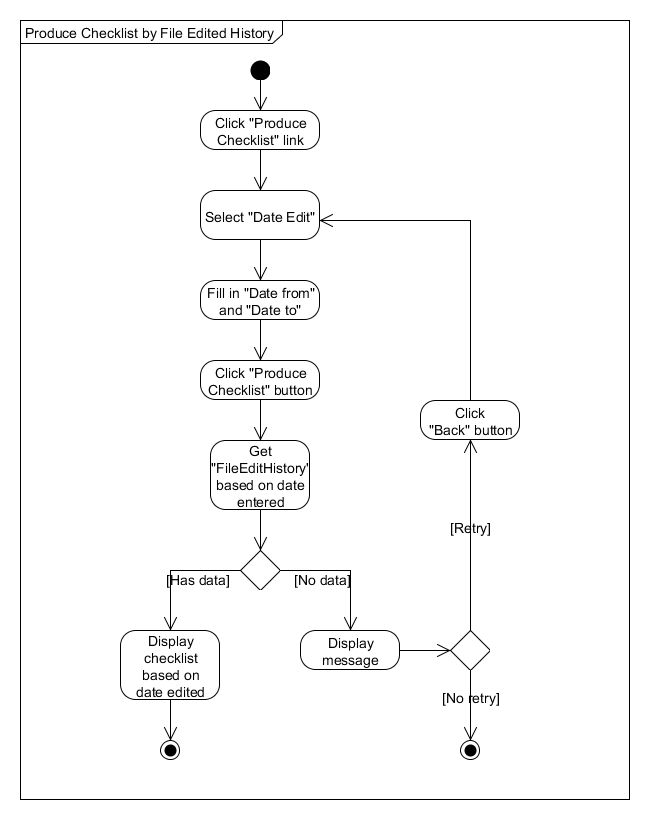
**3.1.11.4 Post-conditions**

The checklist based on the date of edited is produced.

**3.1.11.5 Sequence Diagram – Produce Checklist by Date of Edited**



**3.1.11.6 Activity Diagram – Produce Checklist by Date of Edited**



**3.1.12 Produce Checklist by File Location Use Case**

**3.1.12.1 Brief Description**

This use case allow the file manager to produce a checklist based on the file location. This allow file manager to check the status of the file (such as available, borrowed, or exceed borrow date).

**3.1.12.2 Flow of Events**

The use case begins when the file manager clicks on the “Produce Checklist” link.

**3.1.12.2.1 Basic Flow – Produce Checklist by File Location**

3.1.12.2.1.1 The file manager selects “File Location” in the drop down box.

3.1.12.2.1.2 The file manager choose the file cabinet number in the “Cabinet” drop down box.

3.1.12.2.1.3 Once the file manager clicks on the “Produce Checklist” button, a checklist of file(s) in the cabinet is created.

**3.1.12.2.2 Alternative Flow**

***3.1.12.2.2.1 No file in the cabinet.***

3.1.12.2.2.1.1 If the cabinet does not have any files, the system will display an appropriate message to the file manager.

3.1.12.2.2.1.2 The file manager clicks on the “Back” button to repeat the process in **3.1.12.2.1**.

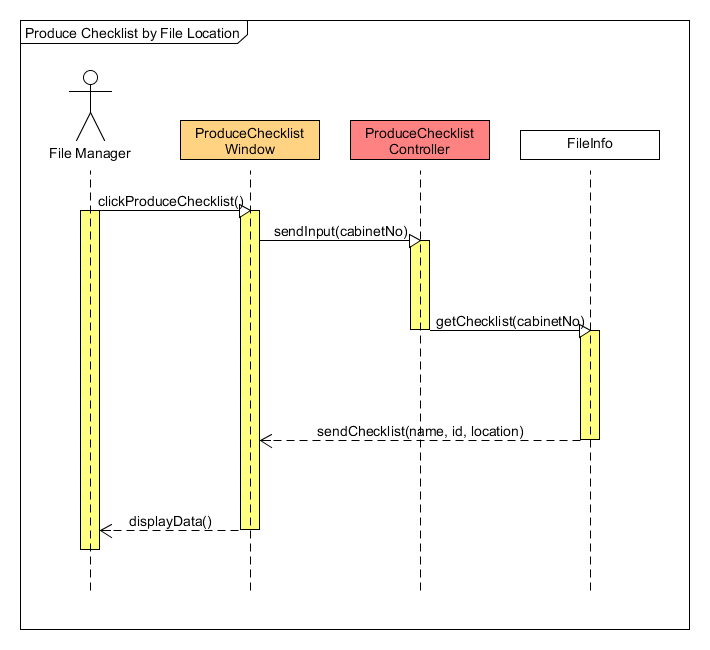
**3.1.12.3 Pre-conditions**

Before the use case begin, the file manager must logged in to the system.

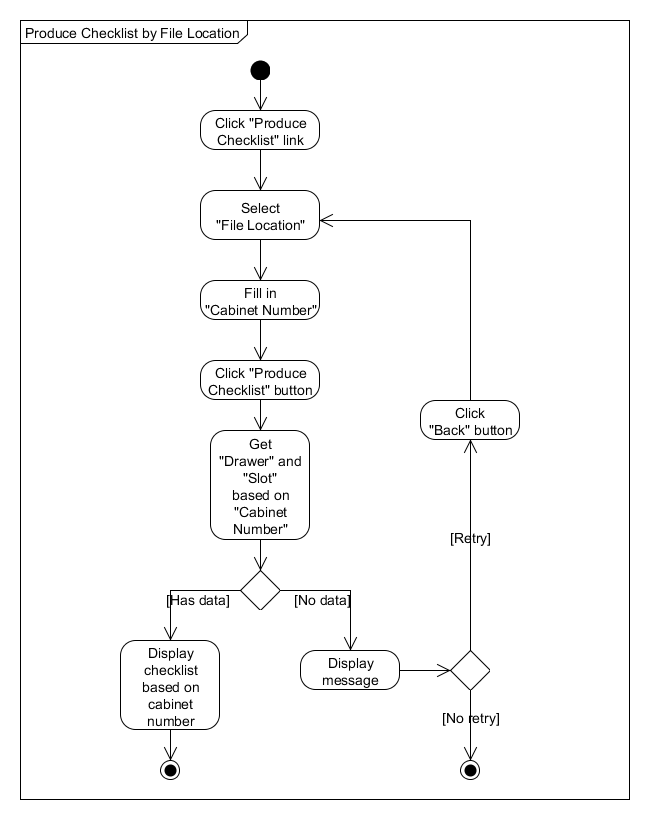
**3.1.12.4 Post-conditions**

The checklist based on the file location is produced.

**3.1.12.5 Sequence Diagram – Produce Checklist by File Location**



**3.1.12.6 Activity Diagram – Produce Checklist by File Location**



**3.2 Non-Functional Requirement**

**3.2.1 Usability**

The system should be easy to use by the employees and should be organized in such a way that the user error is minimized.

**3.2.2 Security**

User need to login to use this system. Some files are also protected from certain employees, which can be accessed by higher-level management.

**3.2.3 Reliability**

The system should be able to search for the requested file accurately, display the availability of the file as well as the borrower of the file, if the file is not available in the cabinet.

**4. APPENDICES**

The figures below are the example page of the log book used by the LPPKJBT to search the file manually. The list contain the name of file as well as the serial number of the file.

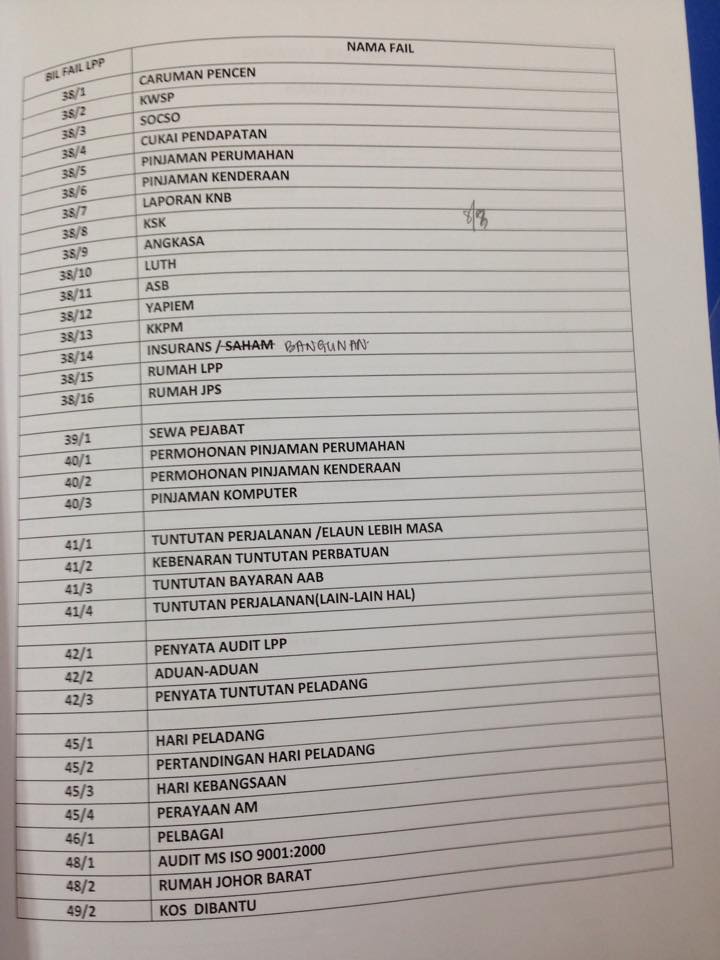


Figure 1 shows the one of the file log page that is categorized as company report.

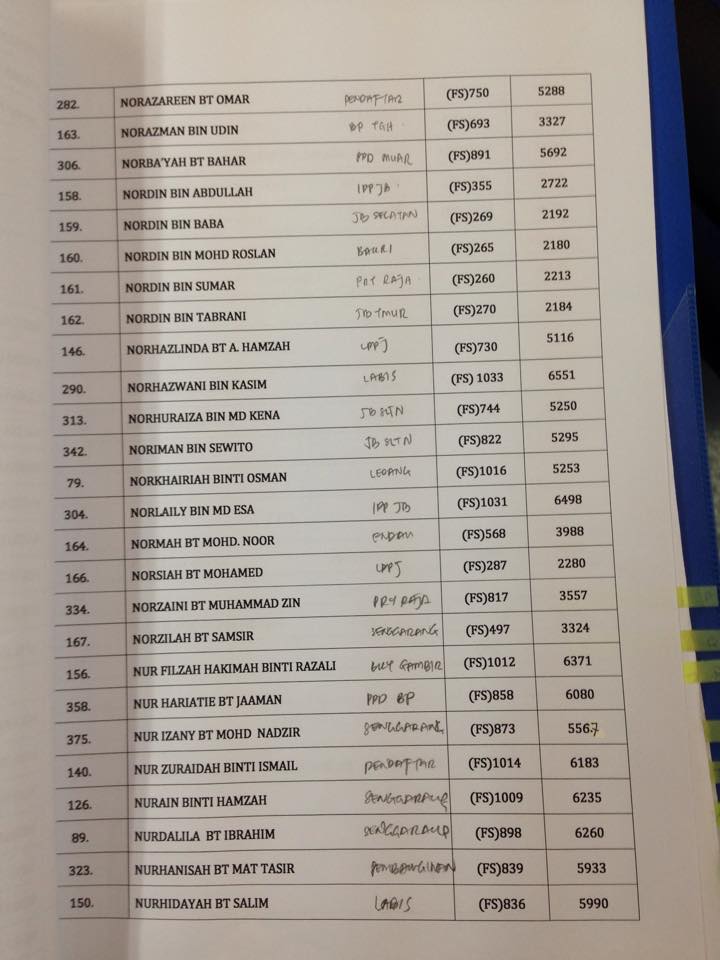


Figure 2 shows one of the file log page that is categorized as employee record.