SECJ3323: Software Design and Architecture

System Documentation

Power File Management System

Version 3.0

Date

30/01/2022 (Sunday)

School of Computing, Faculty of Engineering

Prepared by: <Power Ranger 2.0>

Revision Page

a. **Overview**

The current version of system documentation involved description of introduction, specific requirements, system architectural design, detailed description of components, data design, user interface design, and requirements matrix. The purpose, scope, definitions, acronyms and abbreviations as well as reference are described in the introduction, while the interfaces, system features which consist of use cases, requirements, constraints, and software system attributes are explained in specific requirements. The details of the requirement are explained further in detailed description of components. Next, the data design consists of the design of the database.

b. **Target Audience**

Stakeholders of Power File Management System

-File unit staff and administrative staff of JKSNJ

c. **Project Team Members**

Team Member	Roles	Tasks	Status
Lai Ting Ying	Moderator	Monitor group progress	Completed
Kong Hao Yang	Accuracy checker/ Reporter	Check and compile the report	Completed
Tai Wen Jun	Skeptic	Give query and comment on the assignment / compile EA file	Completed
See Wen Xiang	Recorder / Scriber	Record the results of discussion	Completed

d. **Version Control History**

Version	**Primary Author(s)**	**Description of Version**	**Date Completed**
Version 1.0	Tai Wen Jun	Completed Chapter 1-7	18/12/2021
Version 2.0	Kong Hao Yang	Completed Chapter 8	14/01/2022
Version 3.0	Lai Ting Ying	Completed Chapter 9	30/01/2022

Table of Contents

1. **Introduction**

1.1 **Purpose**

This SD describes the requirements for a file and record management system that will be developed for Jabatan Kehakiman dan Syariah Negeri Johor (JKSNJ). SD is important to be completed as it will be used by the developer to describe the entire system to the stakeholder. The details of the system will be explained into multiple parts to enhance deliverability and clarity of understanding the system. The requirements in this SD are elicited from the stakeholders of JKSNJ through a workshop that is being conducted online on 11 April 2020. The intended audience for this SD are the stakeholders of JKSNJ.

1.2 **Scope**

The software product is a computerised solution for the current operation of file units in JKSNJ. The name of the system is called power file management system. The system will allow the file unit staff to open a new file and input the file title and file code in the system. The system will check whether the same file has been opened or not before in the registered file database and will create one if it does not exist inside the database. Apart from that, the system also notify the file unit staff when the file has reached the thickness of 4sm or when the file has reached 100 pages for closing purpose. The process of closing will be recorded by the system. Any administration staff who wants to borrow the file can make a request through this system. Staff will be notified whether the request is successful or not. Besides, the system will also handle the disposal of the file by filling up the form automatically when there is a request made by the file unit. Lastly, reports can be made through the system regarding the damaged file. They will then be recorded in the system and can be generated as the monthly file damage report.

The objective and goals of this system are to make the business process operation more effective than the previous system. This will enhance the efficiency of the file unit in JKSNJ. Our next goal is to help the file unit staff from making any mistakes during the process especially when classifying the file. Lastly, we want the process to be more systematic and secure while making sure everyone can access this system.

1.3 **Definitions, Acronyms and Abbreviation**

Definitions of all terms, acronyms and abbreviations used are to be defined here.

- JKSNJ - Jabatan Kehakiman dan Syariah Negeri Johor
- SD – System Documentation

1.4 **References**

1. Sommerville, I. 2016. “Software Engineering”, 10th Edition, Pearson.
2. Jabatan Kehakiman dan Syariah Negeri Johor. Portal Rasmi Jabatan Kehakiman Syariah Negeri Johor. Retrieved from https://syariah.johor.gov.my
3. Arkib Negara Malaysia. (2008, October 13). Peraturan-Peraturan Arkib Negara (Penetapan Borang-Borang bagi Pelupusan Rekod Awam). Retrieved from http://www.moha.gov.my
4. Jabatan Kehakiman dan Syariah Negeri Sembilan. (2016). Manual Penggunaan Klasifikasi Fail Jabatan Kehakiman Syariah Malaysia. Retrieved from https://jksns.ns.gov.my
5. Department of Standards Malaysian (2015). Maklumat dan pendokumenan - Pengurusan rekod - Bahagian 1: Am (ISO 15489 -1:2001, IDT). Retrieved from https://jksns.ns.gov.my
6. Jabatan Perkhidmatan Awam Malaysia (2007). Panduan Pengurusan Surat Bilangan 5 Tahun 2007. Retrieved 18 April 2021 from https://docs.jpa.gov.my/docs/pp/2007/pp052007.pdf
7. Arkib Negara Malaysia (2010). Panduan Pelupusan Rekod Awam. Retrieved 20 April 2021 from https://www.moe.gov.my/pekeliling/990-panduan-pelupusan-rekod-awam/file

1.5 **Overview**

The current version of system documentation involved description of introduction, specific requirements.

2. Specific Requirements

2.1 External Interface Requirements

2.1.1 User Interfaces

i. Login interface

The user may view the login interface before they proceed to another interface to authenticate themselves in the system. There will be a username and password text input frame which the data entered in the password text frame will be in the form of dots to protect user privacy. There will also be a submit button to submit the data user input to the server to check the validity of username and respective password in the system. If the user does not successfully login the system, the system shall not show other interface

Ii. File data interface

The user may view the search button located at the top left of the interface that is intuitive for the user to search for desired file title. The interface shall also have a create, close and dispose button that will function as create new file, close file and dispose file according to the user's need.

Iii. Borrow application interface

The administrative staff shall be able to enter the file title that intends to be borrowed and reasoned and press the submit button to be approved by file unit staff. There will also be a status button to view the status of application. Meanwhile for the file unit staff, the user shall be able to view the list of applications to be approved with the information of file title, borrower name and ID and the reason for borrowing.

Iv. User profile interface

The user might view his own name and ID in this interface and the job and position title. Besides, the user might also have the ability to change the contact and email of his profile.

2.1.2 **Hardware Interfaces**

Due to the system is web-based system, the hardware requirements are:

Website RAM: minimum 4GB,

Database server storage: 500GB,

Internet Bandwidth: 1GB/s

Website requirements: support mobile and PC with web browser.

2.1.3 **Software Interfaces**

For data management, Oracle software is used. Oracle database is an ideal database management system that helps to manage entity and its attributes data. The software itself is a free software that charges extra if it requires additional functionality from the software.

Name: Oracle

Mnemonic: Oracle

Specification number:

Version number: 1.0

Source: https://www.oracle.com/database/

For the login interface, go logr/logr is used. The software itself is an open-source package that is provided in github that is available for commercial use with credit stated for the package.

Name: Logr

Mnemonic: Logr

Specification number:

Version number: v0.4.0

Source: https://github.com/go-logr/logr

Purpose: Provide login interface of the proposed system that will authenticate the user.

2.1.4 **Communication Interfaces**

The system shall support HTTPS to ensure safe and private connection between client and server. Besides, the system shall support FTP to enable file transfer.

2.2 **System Features**

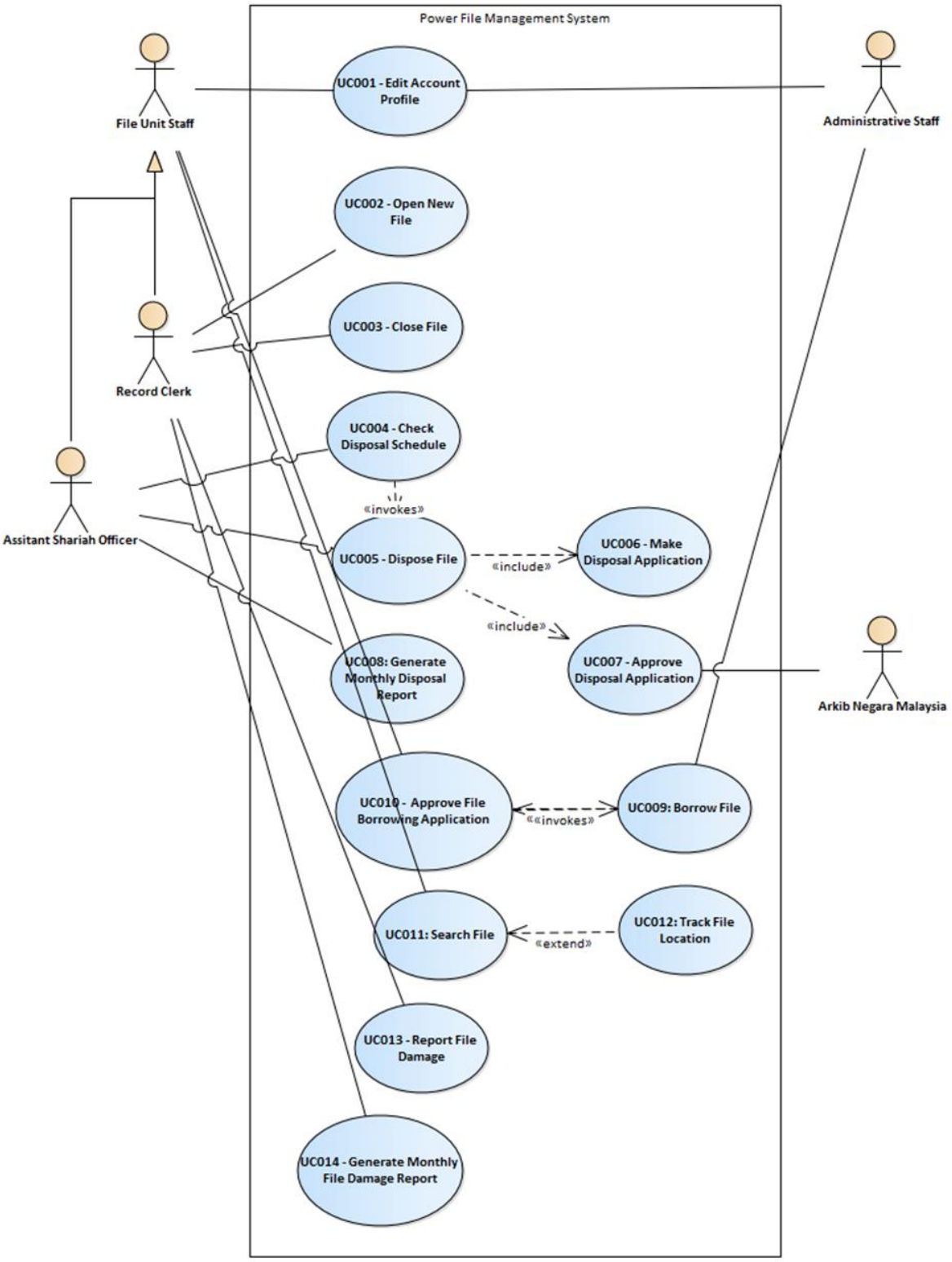


Figure 2.1: Use Case Diagram for <Power File Management System>

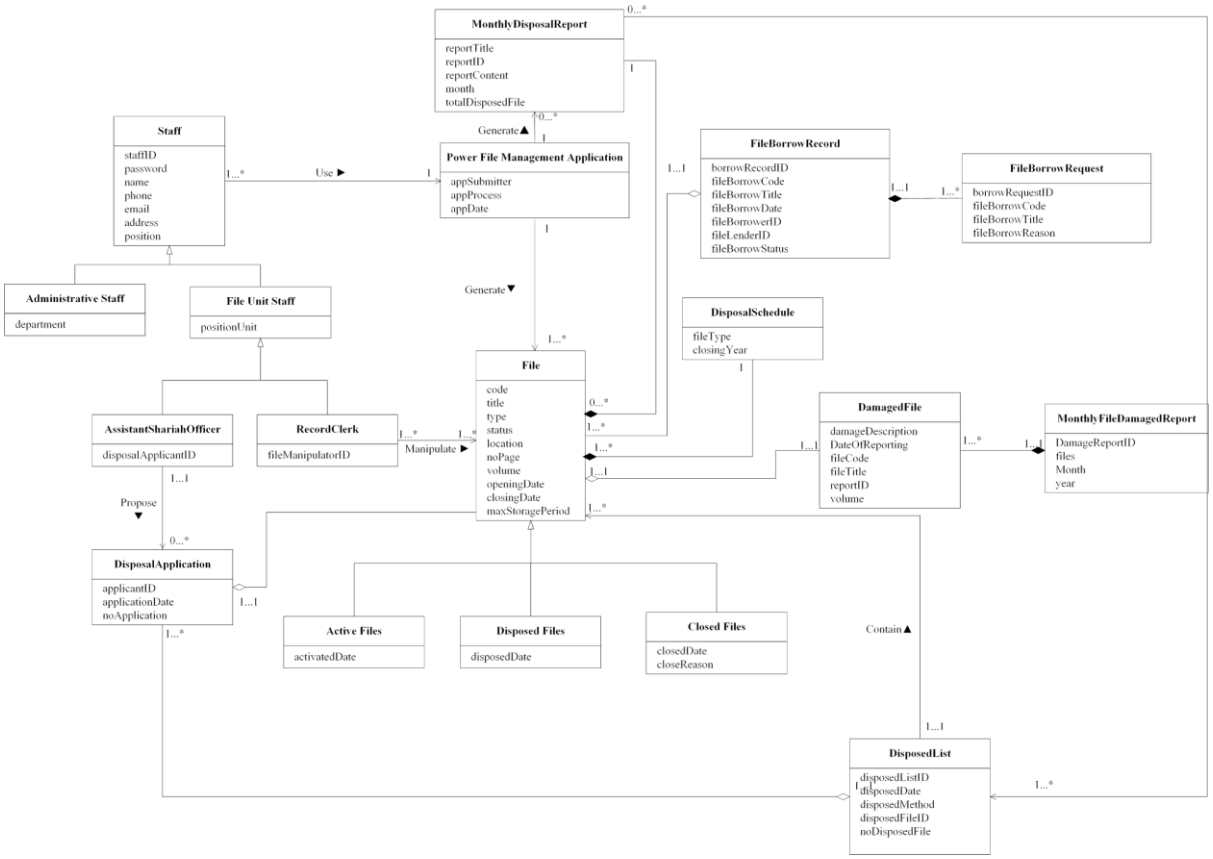


Figure 2.2: Domain Model for <Power File Management System>

The Domain Class Diagram for Power File Management System consists of 15 entities which are Staff, File Unit Staff, Administrative Staff, Assistant Shariah Officer, Record Clerk, Monthly Disposal Report, Power File Management Application, File, Active Files, Closed File, File Borrow Record, Disposal Application, Disposal List, Disposal Schedule, File Borrow Request, Damaged File and Monthly File Damaged Report. The Staff class is a superclass for the File Unit Staff and Administrative Staff subclass. The File Unit Staff is a superclass for Record Clerk and Assistant Shariah Officer subclass. The File class is a superclass for Active Files, Closed Files, and Disposed Files subclasses.

The Monthly Disposal Report class has a composition relation to the File class and association with Disposed List class. Each Monthly Disposal Report contains zero or more files. Each file is used by one monthly disposal report. Each Monthly Disposal Report refers to one or more Disposed List. Each Monthly Disposal Report refers to

one or more Disposed List. Each Disposed List can be referred by zero or more Monthly Disposal Report.

The Staff class has an association with the Power File Management Application class. Each staff member uses one file management application. Each power file management application is used by one or more staff. The Record Clerk class has an association with the File class. Each Record Clerk manipulates one or more files. Each file is manipulated by one or more Record Clerk.

The Power File Management Application class has an association with File and Monthly Disposal Report class. Each Power File Management Application generates one or more files. Each file is generated by one Power File Management Application. Each Power File Management Application generates many Monthly Disposal Reports. Each Monthly Disposal Report is generated by one Power File Management Application.

The Assistant Shariah Officer class has an association with the Disposal Application class. Each assistant shariah officer can propose zero or more disposal applications. Each disposal application is managed by one Assistant Shariah Officer. The Disposal Application class has an aggregation relationship with the Disposed List class. Each disposal application has one disposed list. Each disposed list has one or more disposal applications. The File Borrow Record class has a composition relation with the File Borrow Request class. Each file borrow request has one file borrow record. Each file borrow record has one or more file borrow requests.

The File class has a composition relationship with the Disposal Schedule class in which each file consists of one disposal schedule. Also, the File class has aggregation relationships with Disposal Application and File Borrow Record class and association relationship with Disposed List. Each disposal schedule has one or more files. Each file has one disposal application. Each disposal application has one file. Each file has one file borrow record. Each file borrow record has one or more files. Each file has a disposed list. Each disposed list contains zero or more files. File class

has aggregation relationship with the Damaged File Class. The file has many Damaged File.

The Monthly File Damaged Report has composition relationship with Damaged File class. The Damaged File is part of Monthly File Damaged Report.

2.2.1 UC001: Use Case <Edit Account Profile>

Use Case: Edit account Profile
ID: UC001
Actors: 1. Administrative staff 2. File unit staff
Precondition:
Flow of events: 1. The system display a home page menu for the Power File Management System 2. The administrative staff or file unit staff selects "Account" at the home page 3. The system displays the "Account Profile" page. 4. The administrative staff or file unit staff has two choices to edit their account profile 5. If the user choose to Edit Passowrd, AF1 is performed 6. Administrative staff or file unit staff input a new name and address 7. Administrative staff or file unit staff press save button 8. The system will update the new account profile information inside the database 9. Use case ends
Postcondition:
Alternative Flow: **AF1 Edit Password** 1. Administrative staff or file unit staff input a new password 2. Administrative staff or file unit staff press save button 3. The use case resume at step 8
Exception Flow:

2.2.2 UC002: Use Case <Open New File>

Use Case: Open New File
ID: UC002
Actors: Record Clerk
Precondition: Record clerk is logged into the system successfully
Flow of events: 1. Record clerk enters the classification code of the files that wants to be opened 2. If record clerk enter invalid file classification code, EF1 is performed. 3. The system will perform system search for existing classification code as the file that wants to be opened 4. If existing similar classificated code is not found, EF2 is performed 5. Create a new file 6. Record clerk needs to enter file code, file title and file type 7. New file opened and details updated into system 8. The use case ends
Postcondition: A new file is opened and stored in the system based on classification
Alternative Flow:
Exception Flow: **EF1 - User enters wrong classification code to be opened** 1. The system will print an error message 2. The use case ends with a failure condition **EF2 - Existing similar classificated code found** 1. System will reject the operation 2. System will print out an error message 3. The use case ends with a failure condition

2.2.3 UC003: Use Case <Close File>

Use Case: Close File
ID: UC003
Actors: Record Clerk
Precondition: Record clerk is logged into the system successfully.
Flow of events: 1. System will display the file list of active files. 2. Record Clerk checks the thickness and it reaches or exceeds 100 pages. 3. Record Clerk selects to close the file. 4. The file is closed and updated to the system. 5. Use case ends.
Postcondition: File will be closed and updated in the system.
Alternative Flow:
Exception Flow: **EF1. File thickness does not reach or exceeds 100 pages** 1. Use case continues at NF5.

2.2.4 UC004: Use Case <Check Disposal Schedule>

Use Case: Check Disposal Schedule
ID: UC004
Actors: Assistant Shariah Officer
Precondition: A valid Assistant Shariah Officer is logged on to the system.
Flow of events: 1. The system displays a web page showing the home menu for the Power File Management System. 2. The Assistant Shariah Officer selects "Check Disposal Schedule" at the home page. 3. The system displays the "Check Disposal Schedule" web page. 4. The Assistant Shariah Officer has two choices to choose for displaying the disposal schedule. 5. If the Assistant Shariah Officer selects Display by Closing Year, AF1 is performed. 6. If the Assistant Shariah Officer selects Display by File Type, AF2 is performed. 7. The system will display the file details including the name, code, opening

date, closing date, and the maximum period storage of the file.
8. The system will display a "Dispose File" choice for the file that has reached the disposal schedule.
9. If the Assistant Shariah Officer clicks on the "Dispose File" button <Dispose File>. [Invokes: UC005 - Dispose File]

Postcondition:
1. The disposal schedule is shown according to the chosen file type.
2. The disposal schedule is shown according to the chosen file closing year.

Alternative Flow:
AF1 Display By File Closing Year
1. The system displays the available closing years of the file in order to show the disposal schedule.
2. The Assistant Shariah Officer can input and choose the closing year of files displayed by the system.
3. The system displays the disposal schedule according to the chosen closing year of file.
4. The use case resumes at step 5

AF2 Display By File Type
1. The system displays the available types of files in order to show the disposal schedule.
2. The Assistant Shariah Officer can input and choose the type of files displayed by the system.
3. The system displays the disposal schedule according tot he chosen type of the file.
4. The use case resumes at step 5

Exception Flow:

2.2.5 UC005: Use Case <Dispose File>

Use Case: Dispose File

ID: UC005

Actors: Assistant Shariah Officer

Precondition: A valid Assistant Shariah Officer is logged on to the system.

Flow of events:
1. The system displays a web page showing the home menu for the Power FIle Management System.
2. The Assistant Shariah Officer selects "Dispose FIle" at the home page.
3. The system displays the "Dispose File" web page which shows the list of the files to be disposed.
4. The Assistant Shariah Officer can select the files which to be disposed.
5. <include> UC006 - Make Disposal Application

6. If the disposal application is filled incompletely and unsuccessful does not complete, the Exception E1 is performed.
7. The disposal application of the file has been sent to Arkib Negara Malaysia.
8. The system will notify the Assistant Shariah Officer and update the approval status for the file in the system as "Pending".
9. <include> UC007 - Approve Disposal Application
10. If Arkib Negara Malaysia approves the dispsal application, AF1 is performed.
11. If Arkib Negara Malaysia does not approve the disposal application, AF2 is performed.
12. The use case ends.

Postcondition:

1. The file which is approved by Arkid Negara Malaysia is successfully disposed.
2. The file which is not approved by Arkid Negara Malaysia is unsuccessfully disposed.

Alternative Flow:

AF1 Approved Disposal Application

1. The system will notify the Assistant Syariah Officer and update the approval status of disposal application "Approved".
2. The disposal details of the file are recorded in the system database.
3. The system database is updated with the latest status of the file which is "disposed".
4. The use case resumes at step 9

AF2 Reject Disposal Application

1. The system will notify the Assistant Syariah Officer and update the approval status of the disposal application "Rejected".
2. The use case resumes at step 9

Exception Flow:

E1 Incomplete and Unsuccessful Disposal Application

1. The use case ends with failure conditions.

2.2.6 UC006: Use Case <Make Disposal Application>

Use Case: Make Disposal Application
ID: UC006
Actors: Assistant Shariah Officer
Precondition: A valid Assistant Shariah Officer is logged on to the system.
Flow of events: 1. The system displays the make disposal application page. 2. The system displays a web page showing the file disposal application form. 3. The Assistant Shariah Officer fills up the disposal form completely. 4. The Assistant Shariah Officer confirms to submit the disposal form. 5. The system will have to verify the identity of the Assistant Shariah Officer before confirming to submit the disposal application form. 6. The Assistant Shariah Officer needs to enter the staffID and password again. 6.1. If **Not A Valid Identity**, Exception E1 is performed. 7. The system will send the disposal application to Arkib Negara Malaysia. 8. The system displays the information message to the Assistant Shariah Officer that the application is submitted successfully. 9. The use case ends and flow will be back to the calling case.
Postcondition: 1. The disposal application has been successfully submitted to Arkib Negara Malaysia in the system
Exception Flow: **E1: Not A Valid Identity** If in step 6 the basic flow the staff ID or password is invalid, then 1. The system displays a "Sorry, invalid staff ID or password" message. 2. The use case ends with an indication of the failure.
Postcondition: The exception message "Sorry, the invalid staff ID or password" is displayed to the Assistant Shariah Officer.

2.2.7 UC007: Use Case <Approve Disposal Application>

Use Case: Approve Disposal Application

ID: UC007

Actors: Arkib Negara Malaysia

Precondition:

1. A valid Arkib Negara Malaysia officer is logged on to the system.

Flow of events:

1. The system displays the approve disposal application page for Arkib Negara Malaysia.
2. The system displays a web page showing the disposal application details.
3. The Arkib Negara Malaysia views and checks the disposal application.
4. The Arkib Negara Malaysia decides the approval for the disposal application.
   4.1. If Not approve Disposal Application, AF1 is performed.
5. The Arkib Negara Malaysia shall approve the disposal application.
6. The system shall display a confirmation message to Arkib Negara Malaysia for confirming the approval of the disposal application.
7. The Assistant Shariah Officer selects "Yes" if confirming the approval or disapproval of the disposal application.
   7.1. If the Assistant Shariah Officer selects "Cancel" Exception E1 is performed.
8. The system database is updated with the latest approval status of the file.
9. The system sends the notification for the latest approval status of the file.
10. The use case ends and the flow will be back to the calling use case.

Postcondition:

1. The disposal application is approved successfully by the Arkib Negara Malaysia and approval status of the disposal application is updated successfully in the system database.

Alternative Flow:

AF1. Not Approve Disposal Application

If in step 4 of basic flow the assistant shariah officer selects “Disapprove”, then

1. The Arkib Negara Malaysia will disapprove the disposal application by selecting the disapprove button.
2. The system shall display a confirmation message to Arkib Negara Malaysia for confirming the disapproval of the disposal application.
3. The use case resumes at step 6.

Postcondition:

1. The disposal application is disapproved by Arkib Negara Malaysia and the disapproved status of the disposal application is updated successfully in the system database.

Exception Flow:

E1. Cancel Operation
If in step 6 of the basic flow Assistant Shahriah Officer selects “Cancel” for confirming the approval r disapproval of the disposal application, then

1. The system shall return to the approve disposal application page.
2. The use case resumes at step 1.

2.2.8 UC008: Use Case <Generate Monthly Disposal Report>

Use Case: Generate Monthly Disposal Report
ID: UC008
Actors: Assistant Shariah Officer
Precondition: A valid Assistant Shariah Officer logged into the system.
Flow of events: 1. The system displays a home page menu for the Power File Management System. 2. Assitant Shariah Officer selects "Generate Monthly Dispose File Report" at the home page. 3. The system displays the "Generate Monthly Dispose File Report" page. 4. The system asks Assitant Shariah Officer about the month and year of the report he or she wants to generate. 5. Assitant Shariah Officer inputs the desired month and year. 6. For each day of the month 6.1. The system will find the dispose file code, title, volume, date opened, date closed, and the date of disposal to display on the screen. 6.2. The system will calculate the total number of disposed file. 6.3. The system will display the total number of disposed file. 7. If user select print report, AF1 is performed. 8. Use case ends.
Postcondition:
Alternative Flow: **AF1 Print Report** 1. The system will print out disposed file code, title, volume, date opened, date closed, and the date of disposal made with the message of "Printed Succesfully". 2. The system will print out the total file. 3. Use case continues at NF8.

2.2.9 UC009: Use Case: <Borrow File>

Use Case: Borrow File
ID: UC009
Actors: Administrative staff
Extends: UC010 Approve File Borrow Application
Precondition: A valid Administrative staff logged in to the system
Flow of events: 1. System display main window. 2. Administrative staff choose to borrow a file & record. 3. System display borrow file window 4. Administrative staff will input file title and file borrowing reason in the input box and confirm the file borrowing button. 4.1. If the administrative staff does not enter the file borrowing reason, EF1 is performed. 4.2. If the administrative staff does not enter file borrowing title, EF2 is performed. 4.3. If the administrative staff closes the borrow file window and cancels the operation, AF1 is performed. 5. The system will store the file title to be borrowed and borrowing reason in the file borrow record database. 6. The system will invoke use case UC010 Approve file borrow application. 7. The system will retrieve fill borrow application status. 8. The administrative staff will be able to view fill borrow application status 9. The use case ends.
Postcondition: 1. Administrative staff was notified the borrowing application was approved.
Alternative Flow: **AF1. User Cancel Operation** 1.1 The system back to the main window. 1.2 The system quits the current use case.
Postcondition:
Exception Flow: **EF1. User does not enter file borrowing reason** 1.1 The system will hint the user to enter file borrowing reason before confirming the file borrowing button 1.2 The system will resume NF6 once the user has entered file borrowing reason. **EF2. User does not enter file borrow title** 2.1 The system will hint the user to enter file title before confirming the file borrowing button. 2.2 The system will resume NF6 once the user has entered the file borrow title.

Postcondition:
Administrative staff was notified the borrowing application was approved

2.2.10 UC010: Use Case: < Approve File Borrowing Application>

Use Case: Approve File Borrowing Application

ID: UC010

Actors: File unit staff

Precondition:
1. A valid file unit staff logged in to the system, invoked by use case UC009 Borrow file

Flow of events:
1. System display main window.
2. File unit staff choose to approve borrow file application.
3. System retrieve file title and file borrow reason from file borrow record database.
4. System display file title and file borrow reason to be borrowed to file unit staff.
5. File unit staff agree on file borrow reason and press the approve file borrow button.
6. If file unit staff press the exit button invoke **AF1**.
7. If file unit staff do not agree on file borrowing reason, invoke **AF2**.
8. If file unit staff do not decide on file borrow status invoke **EF1**.
9. The system will store file borrow application status in the file borrow record database.
10. Use case end.

Postcondition:
1. File borrow application is approved .

Alternative Flow:
AF.1 User Cancel Operation
1.1 The system back to main window.
1.2 The system will exit current use case.
AF.2 User disapprove file borrowing
2.1 User may press the disapprove file borrow button in the main window.
2.2 The system will return NF7.

Exception flow:
EF1. User ignore to set file borrow application status
1.1 The system will hint the user to determine the file borrow application status.
1.2 The system will return NF7 if the user determined the file borrow application status

Postcondition:
File borrow application is approved or denied

2.2.11 UC011: Use Case: <Search File>

Use Case: Search File
ID: UC011
Actors: File unit staff
Precondition: A valid file unit staff logged in to the system.
Flow of events: 1. The system displays the main window. 2. The file unit staff select the search file option. 3. The system displays a search file window and input text dialog for file title. 4. The user input file title that intended to search. 5. The system search file with the file title inputted in the file database. 6. If the **file is not found in the file database, EF1** is performed. 7. If file unit staff **press the exit button to cancel operation, AF1** is performed. 8. The system returns file location and display to file unit staff. 9. If the user want to track the file UC012 <Track File Location> 10. The use case ends.
Postcondition: File unit staff get file location
Alternative Flow: **AF1. User Press Exit Button to Cancel Operation** 1. The system back to the main window. 2. The system will exit the current use case.
Exception Flow: **EF1. File not found in File Database** 1.1 The system will return null and display files not found to file unit staff.
Postcondition: File unit staff get file not found message.

2.2.12 UC012: Use Case <Track File Location>

Use Case: Track File Location
ID: UC012
Actors: Record Clerk
Precondition: 1. Record Clerk needs to know the classification code to search for the file.
Flow of events: 1. Search for the file's location and history by searching for the classification code. 2. If the file has been borrowed 2.1. The search system will provide the contact information of the borrower and lender. 3. Else 3.1. The search system will provide the location of the file in the specific file cabinet in the file room.
Postcondition: 1. The search system will be updated if the files are borrowed.

2.2.13 UC013: Use Case <Report File Damage>

Use Case: Report File Damage
ID: UC013
Actors: Record clerk
Precondition: 1. Record clerk has logged on the system
Flow of events: 1. Record clerk selects the "Report File Damage" button at File Damage Report window. 2. The system displays "Report File Damage" screen. 3. The system will automatically generate a report ID and provide fields for user to enter the details of the damaged file. 4. Record clerk enters the file code, file title, volume, date of reporting , volume and damage description. 5. If record clerk clicks the report button with incomplete details of the damaged file, E1-Incomplete Detail is executed. 6. Else, the report of the damaged file is submitted and the system will display a successful submission message.
Postcondition: 1. Damaged File is reported successfully and stored in the report database
Exception Flow: **E1: Incomplete Detail** 1. The system displays an error message to indicate the fields that need to be filled. 2. The use case return to step 3.

2.2.14 UC014: Use Case <Generate Monthly File Damage Report>

Use Case: Generate Monthly File Damage Report
ID: UC014
Actors: Record clerk
Precondition: Report clerk has logged on the system
Flow of events: 1. Record clerk selects "Generate Monthly File Damage Report" button at File Damage Report window. 2. The system displays the "Generate Monthly File Damage Report" page. 3. Record clerk input the desired month and year for the file damage report that want to generate at the fields provided by the system. 4. For each day of the month. 4.1. The system will find the report ID, file code, file title, damage description and the date of reporting and display the result on the screen. 4.2. The system will calculate the total number of damaged files. 4.3. The system will display the total number of damaged files. 5. Use case ends
Postcondition:
Alternative Flow:
Exception Flow:

2.3 **Performance Requirements**

- The average response time for each page shall be less than 5 seconds.
- The change to the database such as opening and closing of files shall be reflected instantly in the database.
- The authorized users shall be able to access the file in real-time.
- The authorized users shall be able to update or delete the document without any connection error on the server-side.
- The system shall be able to support at least 200 concurrent users.
- The API integration with external services shall be done within 1 minute.
- The system shall be reliable with 90 per cent of reliability for each month with minimum critical failures.
- The authorized users shall be able to trace the file record across the system.

2.4 **Design Constraints**

- The process of disposal of a file should follow a standard set by Arkib Negara Malaysia.
- All file will follow the JKSM(Jabatan Kehakiman Syariah Malaysia) format of file classification.
- The software shall be designed so that both desktop and mobile can access the system.
- The system shall have built-in secure authentication security to prevent unauthorized persons from accessing the system.
- The system shall have administrative functions to assign different rights and permissions to individual users, groups or roles.
- The system user interface shall use Bahasa Melayu as their main language.

2.5 **Other Requirements**

- The system shall be available via a web interface or an appropriate common user interface.
- The system shall be flexible enough in which it is not specific to the type of operating system but versatile with any of it.
- The authorized users shall be able to use the system without any prior knowledge of the operation in the system.
- The system shall be able to be added with a new module or function with no or little modification to the existing system.

3. System Architectural Design

3.1 Architecture Style and Rationale

The architecture design pattern that is going to be used in this system is layered architecture. The reason that this pattern is chosen was due to it allows replacement of the entire layer if the interface is successfully maintained. The architecture is also easy to understand by categorizing components with the same function into the same layer. New functions and business rules are easier to be applied as long as the process and business logic is not spread throughout the code. This pattern of architecture will also allow debugging and tracing to be done faster as the code and module are grouped and instinctive to be found. There will be three layers used in this case study and project which is view layer, controller layer and data access layer, view layer will reference on the controller layer meanwhile controller layer will reference on the data access layer. All layers used in this project will be closed layers to reduce the complexity of code and system.

4. Detailed Description of Components

4.1 Complete Package Diagram

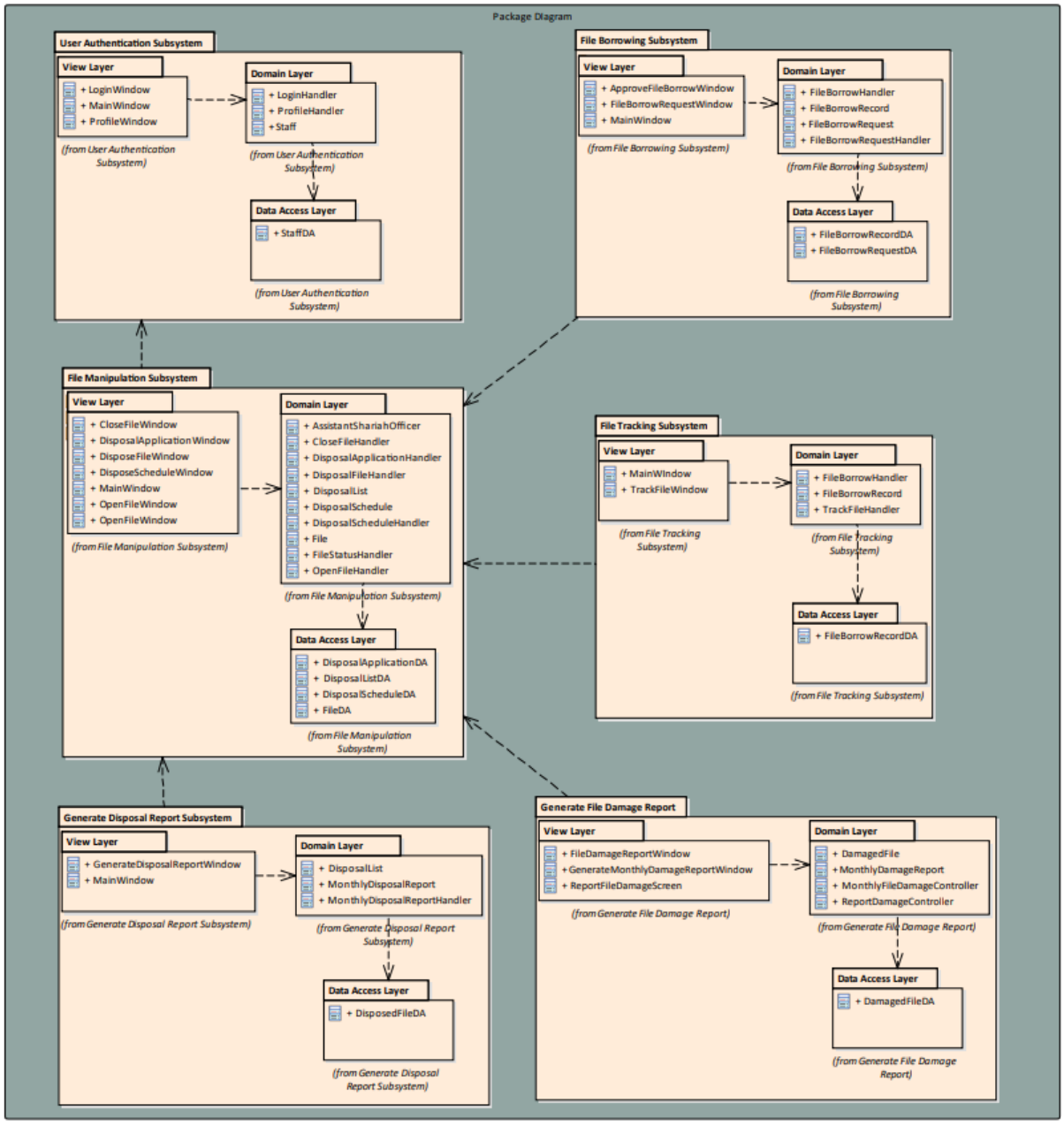


Figure 4.1: Package Diagram for <Power File Management System>

4.2 **Detailed Description**

4.2.1**:** SD001: Use Case <Open New File>

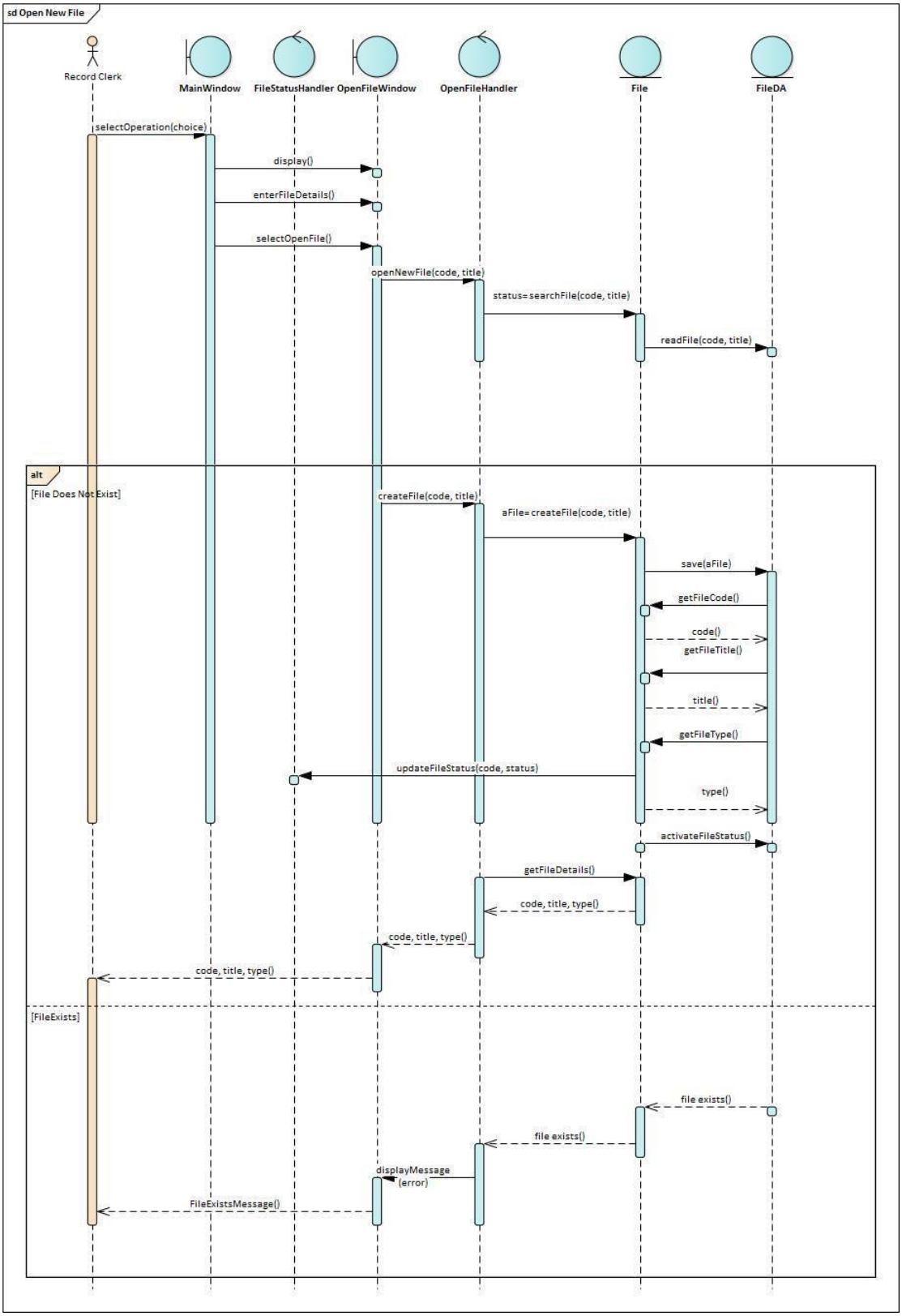


Figure 4.2.1: Sequence diagram for <Open New File> use case

4.2.2: SD002: Use Case <Close File>

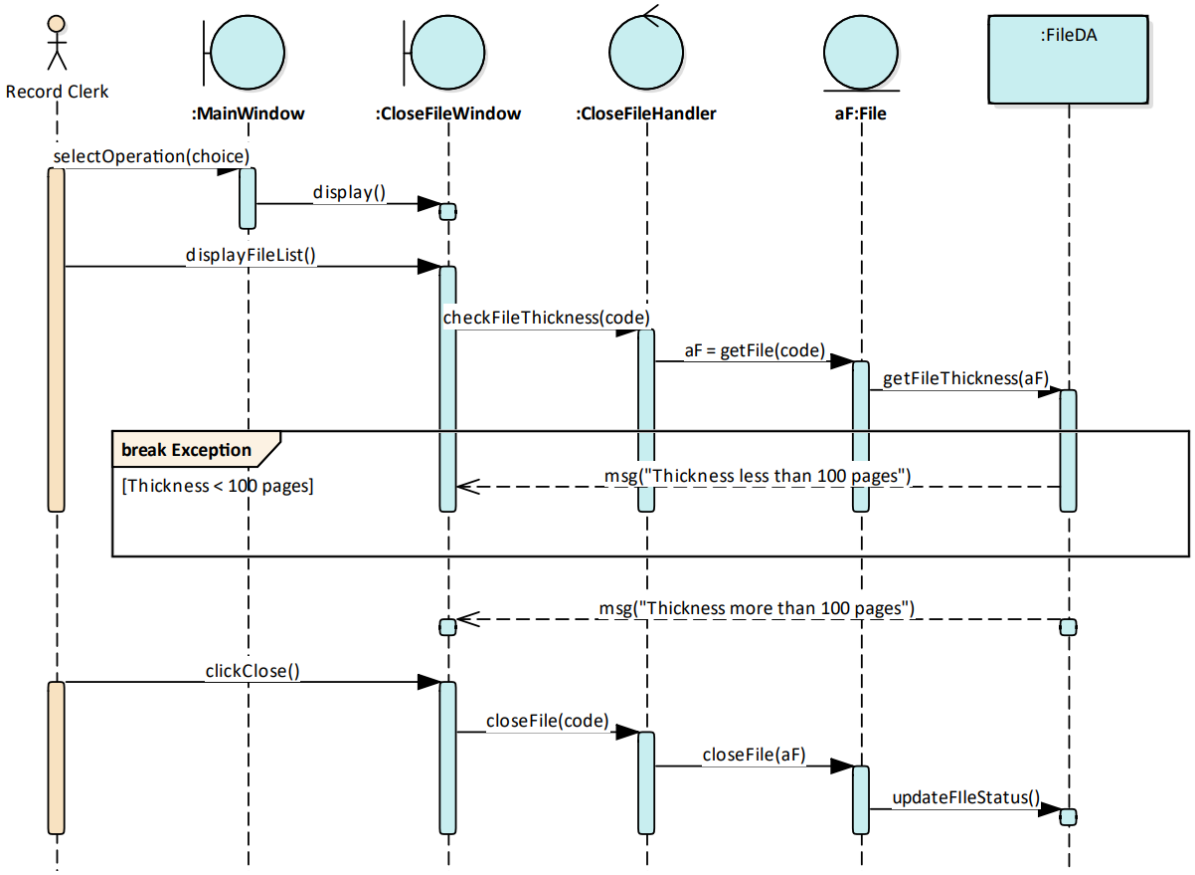


Figure 4.2.2: Sequence diagram for <Close File> use case

4.2.3: SD003: Use Case < Check Disposal Schedule >

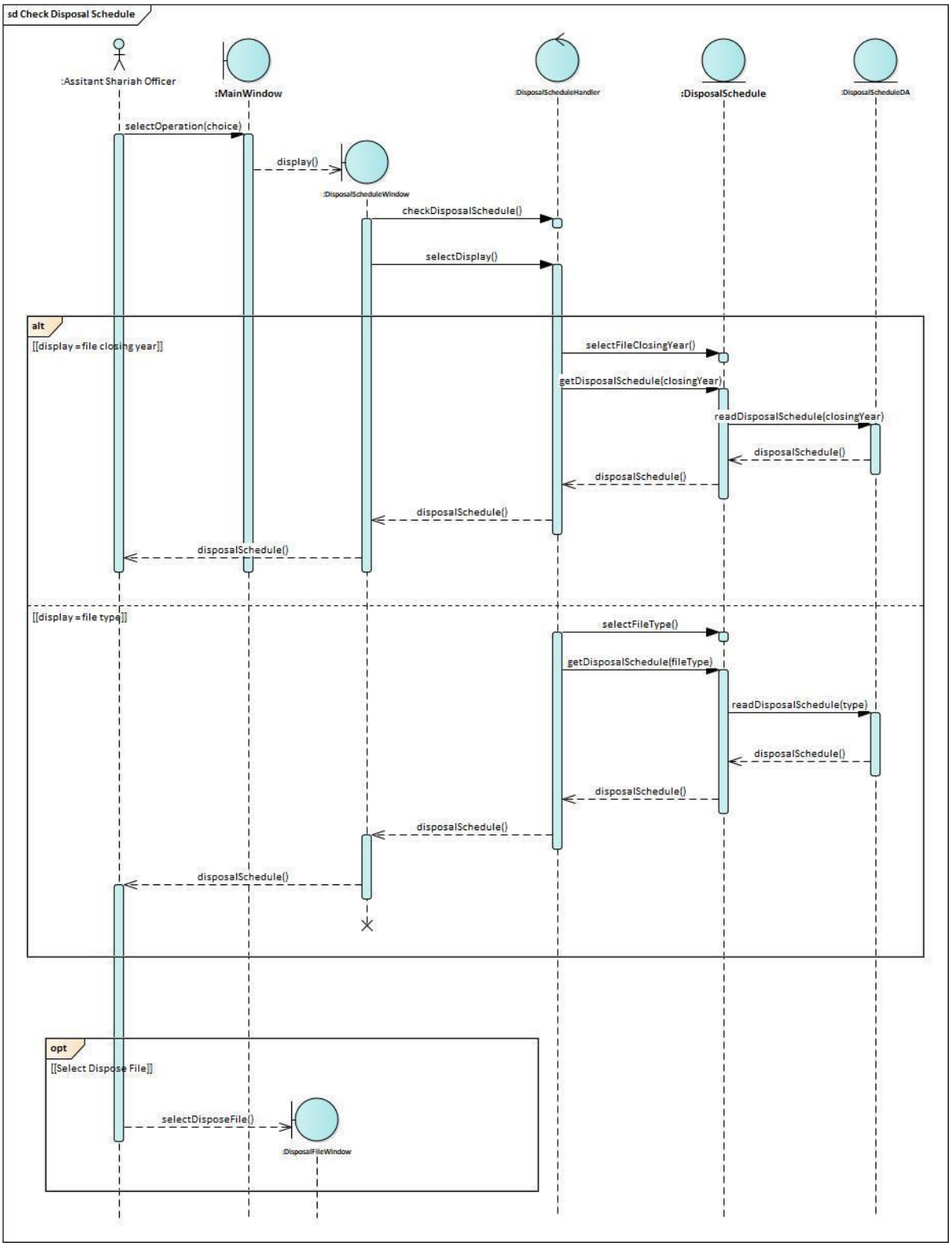


Figure 4.2.3: Sequence diagram for < Check Disposal Schedule > use case

4.2.4: SD004: Use Case <Dispose File>

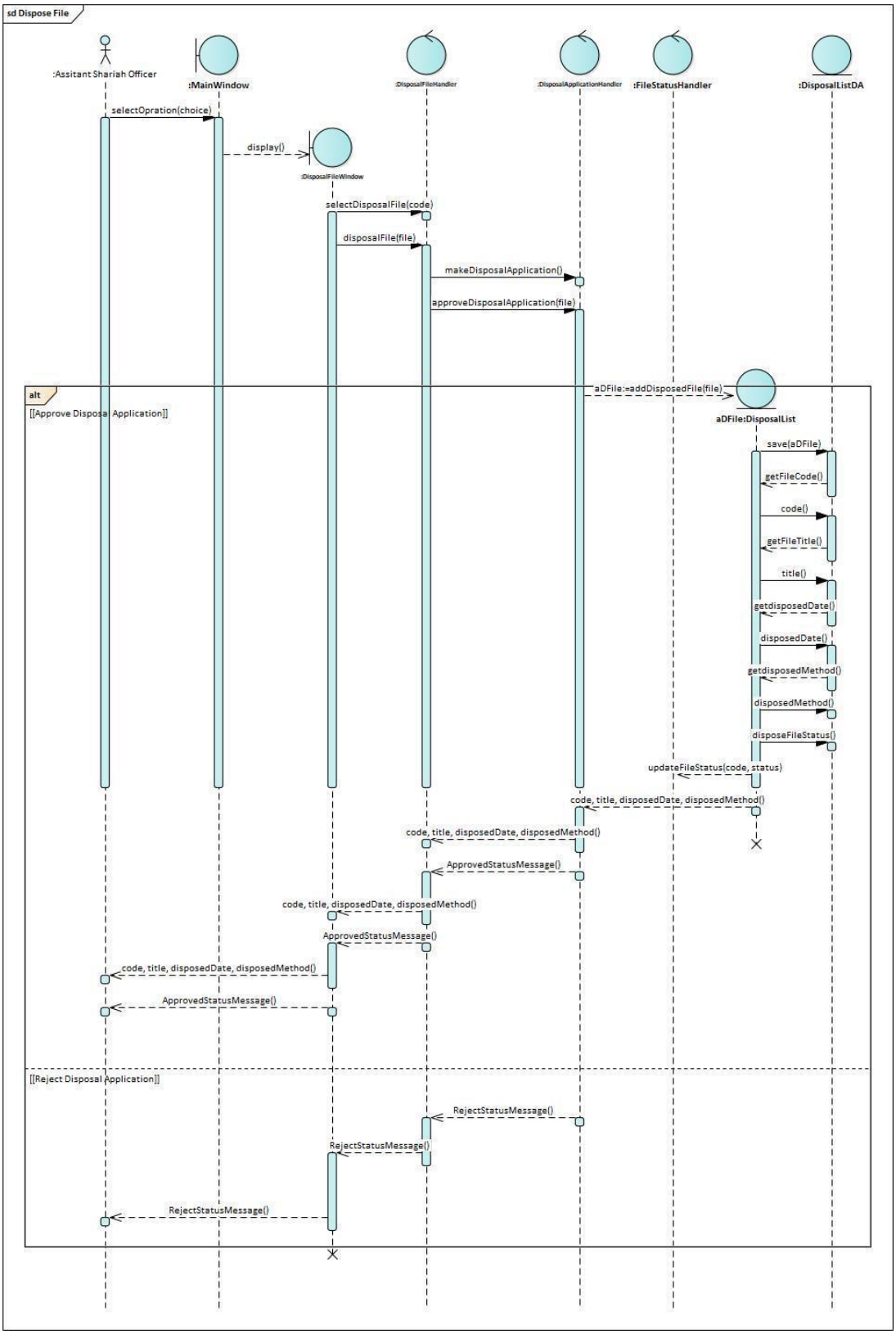


Figure 4.2.4: Sequence diagram for <Dispose File> use case

4.2.5: SD005: Use Case <Make Disposal Application>

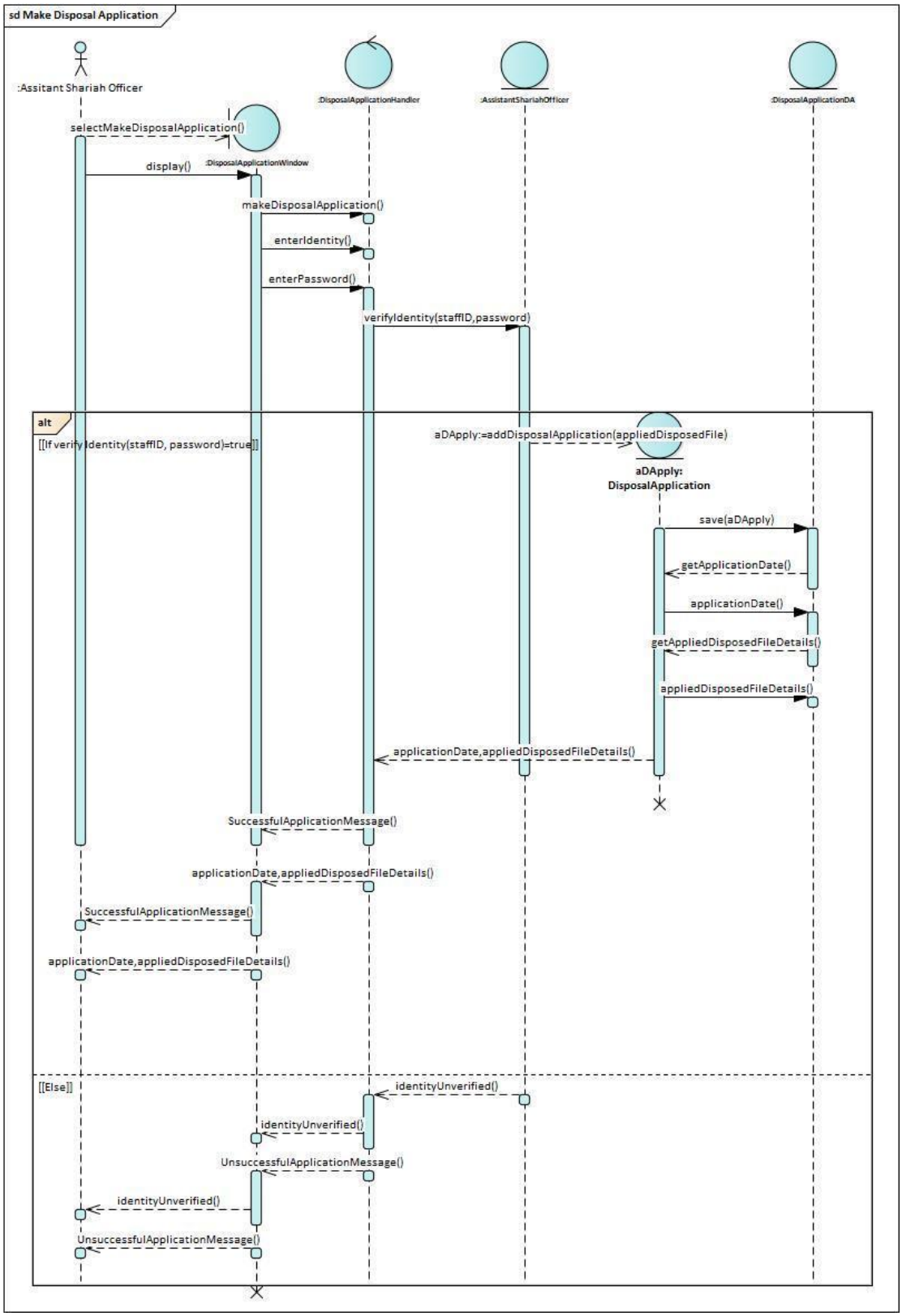


Figure 4.2.5: Sequence diagram for <Make Disposal Application> use case

4.2.6: SD006: Use Case <Approve Disposal Application>

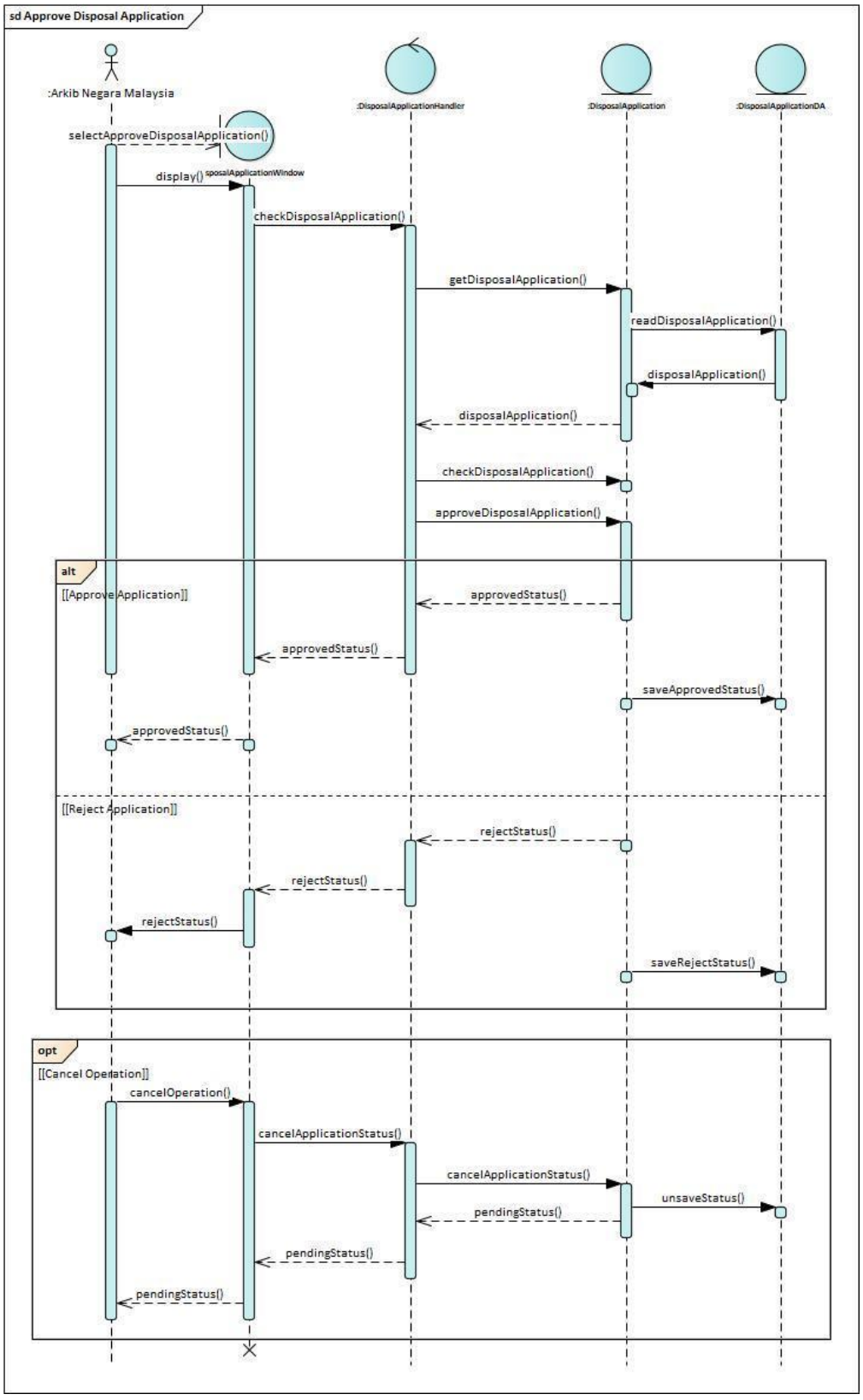


Figure 4.2.6: Sequence diagram for <Approve Disposal Application> use case

4.2.7: SD007: Use Case <Approve File Borrowing Application>

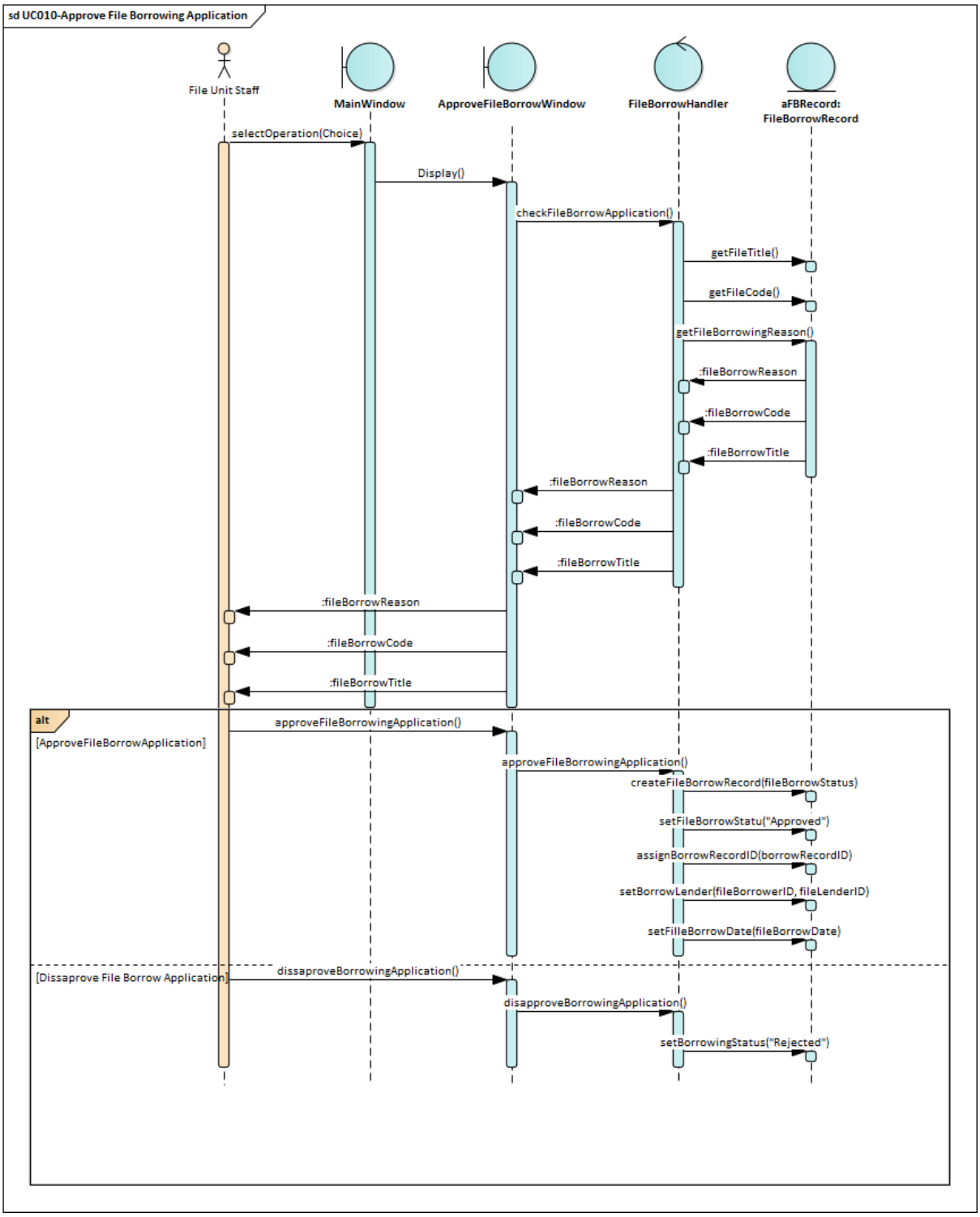


Figure 4.2.7: Sequence diagram for <Approve File Borrowing Application> use case

4.2.8: SD008: Use Case <Borrow File>

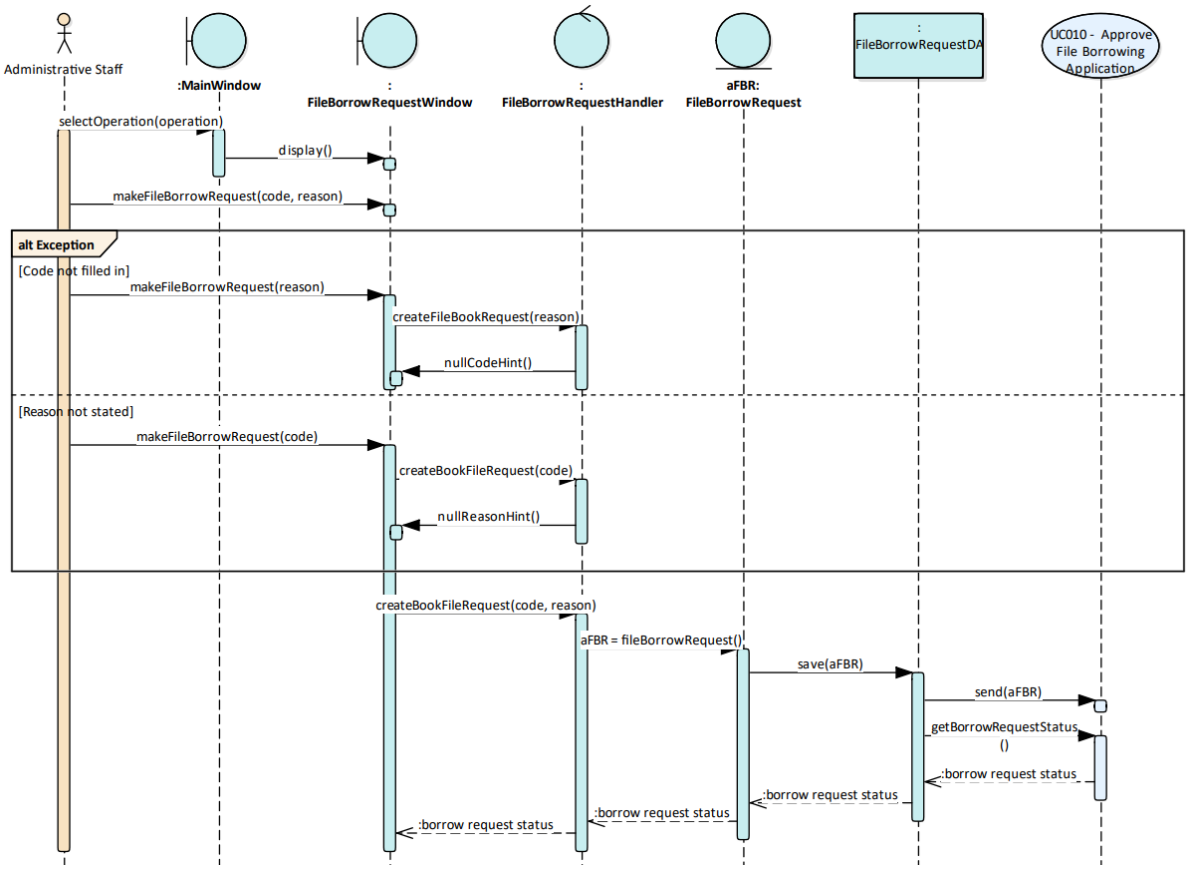


Figure 4.2.8: Sequence diagram for <Borrow File> use case

4.2.9: SD009: Use Case <Search File>

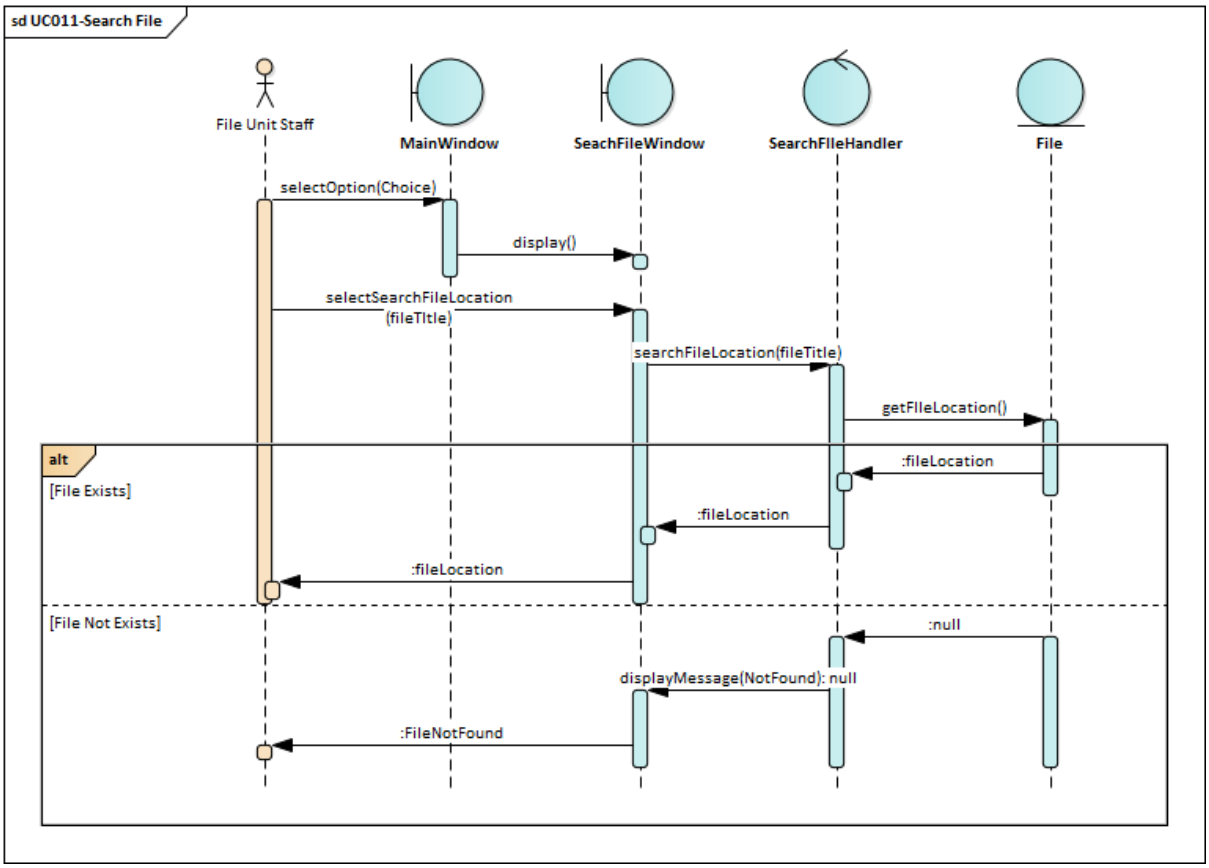


Figure 4.2.9: Sequence diagram for <Search File> use case

4.2.10: SD010: Use Case <Track File>

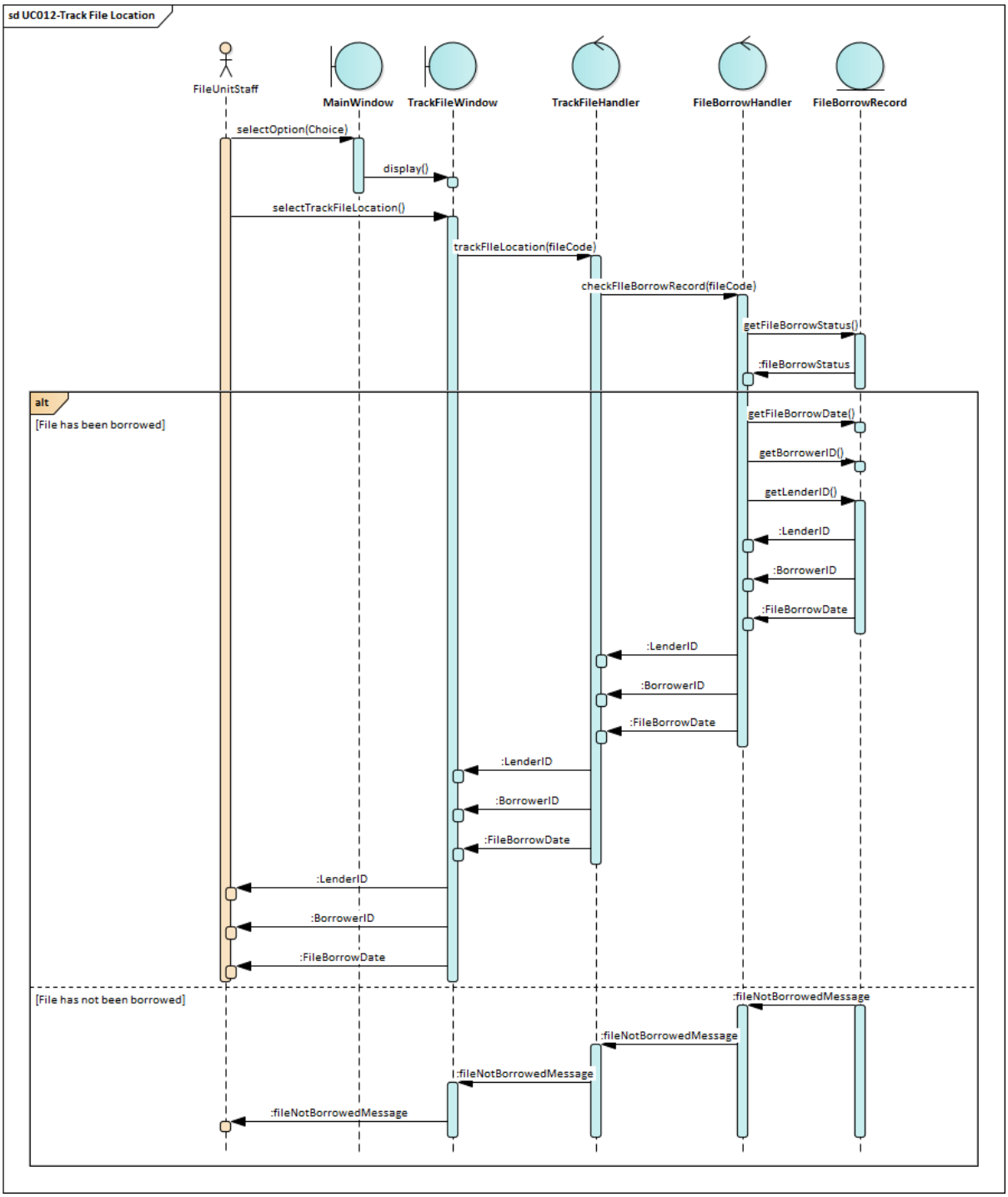


Figure 4.2.10: Sequence diagram for <Track File> use case

4.2.11: SD011: Use Case <Edit Account Profile>

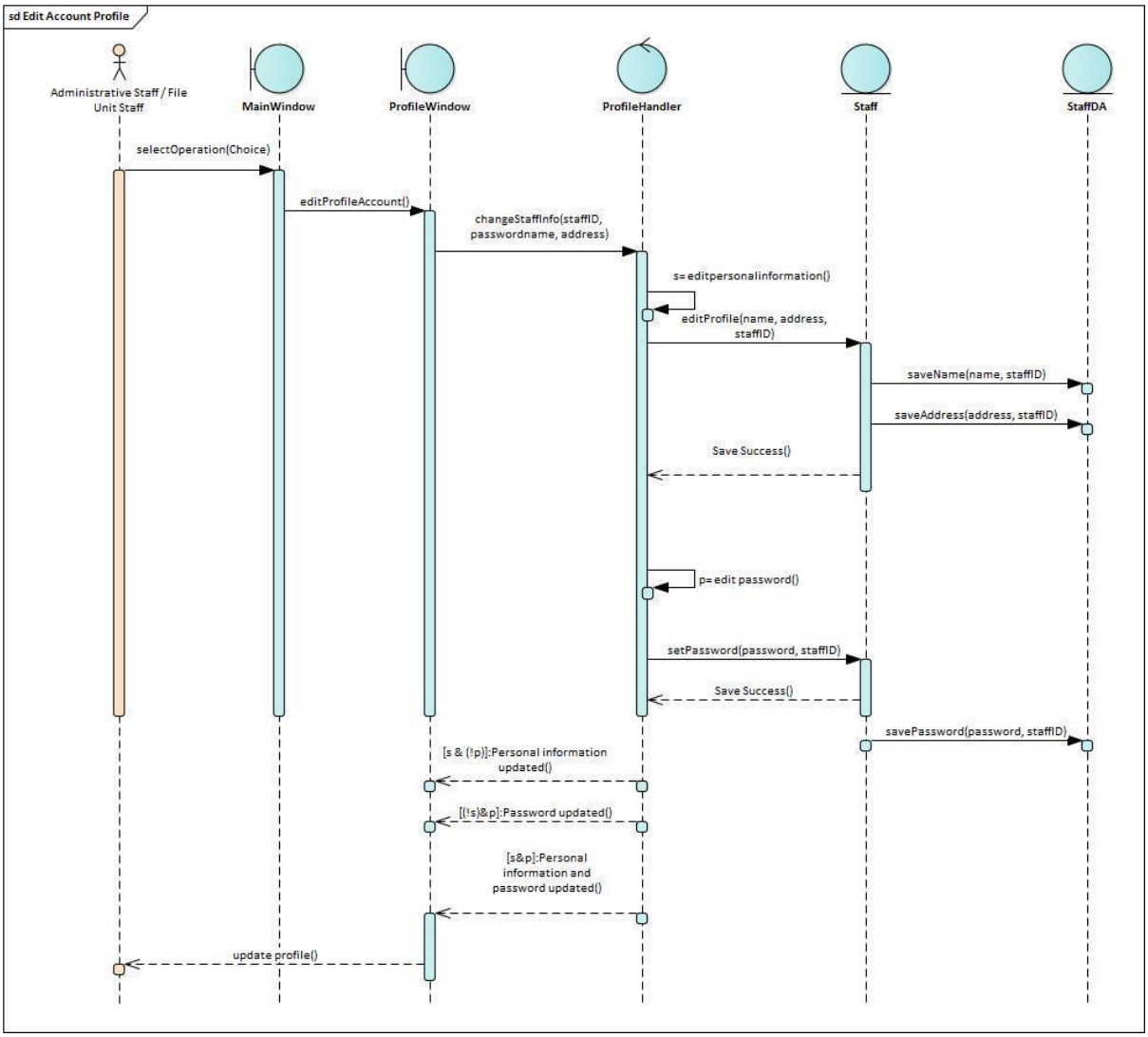


Figure 4.2.11: Sequence diagram for <Edit Account Profile> use case

4.2.12: SD012: Use Case <Generate Monthly Disposal Report>

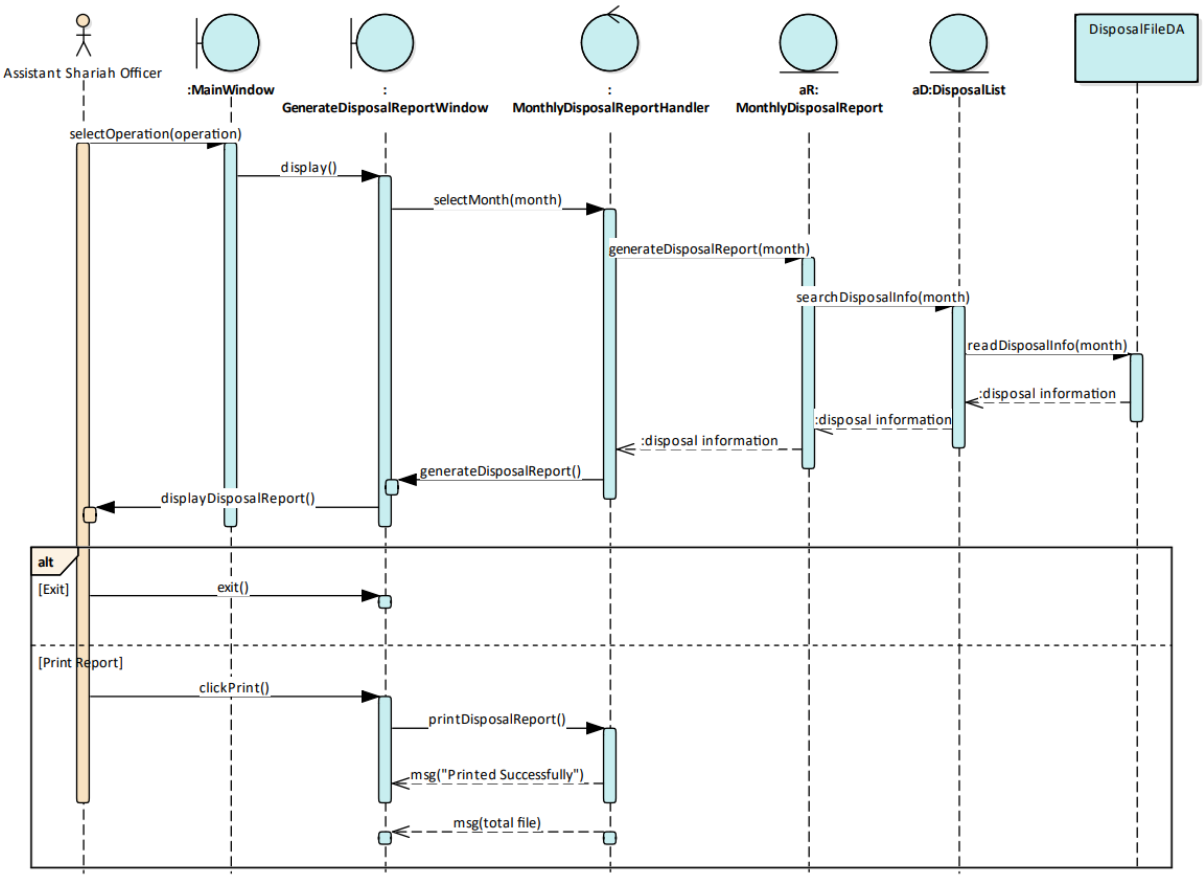


Figure 4.2.12: Sequence diagram for <Generate Monthly Disposal Report> use case

4.2.13: SD013: Use Case <Report File Damage>

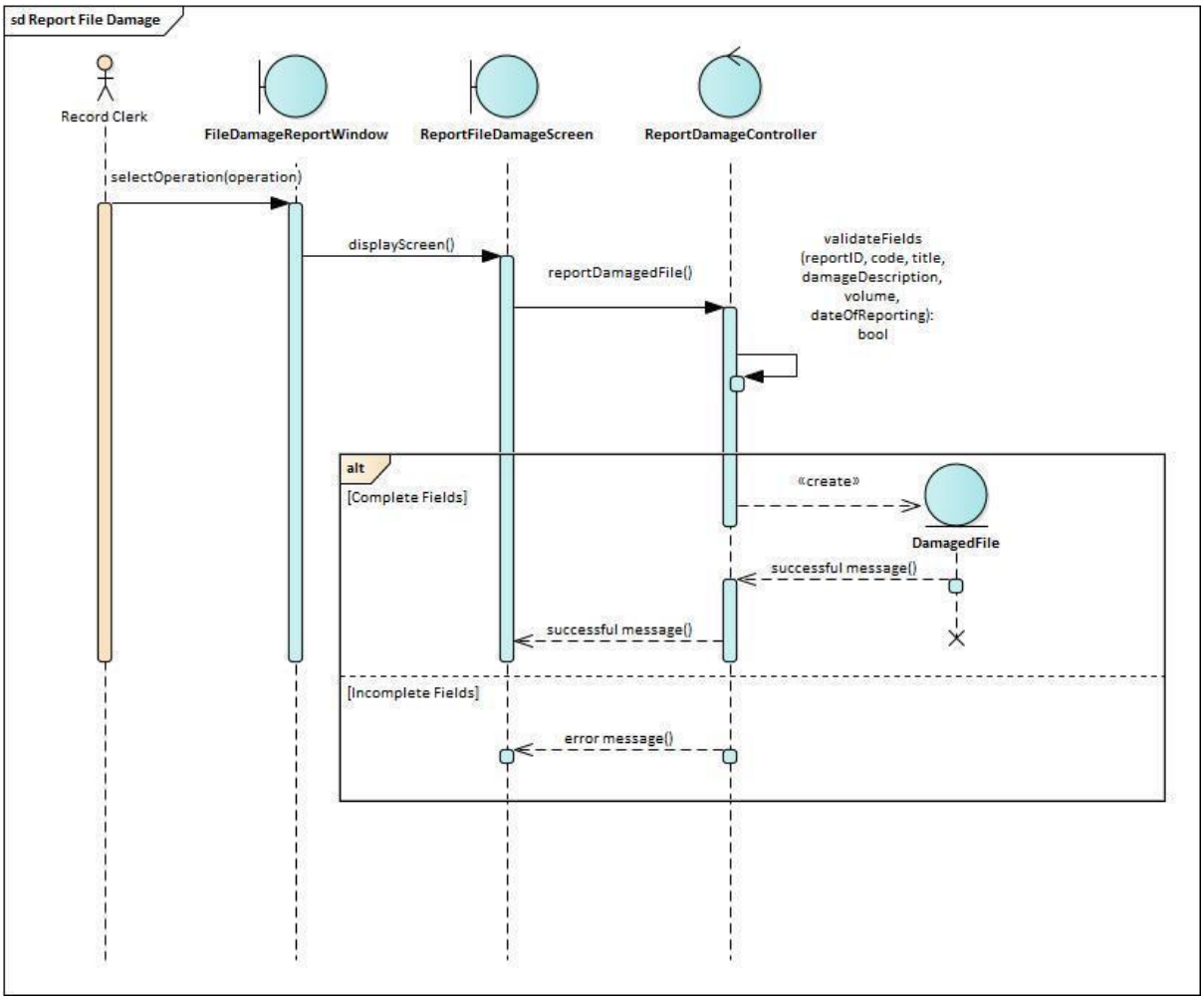


Figure 4.2.13: Sequence diagram for <Report File Damage> use case

4.2.14: SD014: Use Case <Generate Monthly File Damage Report>

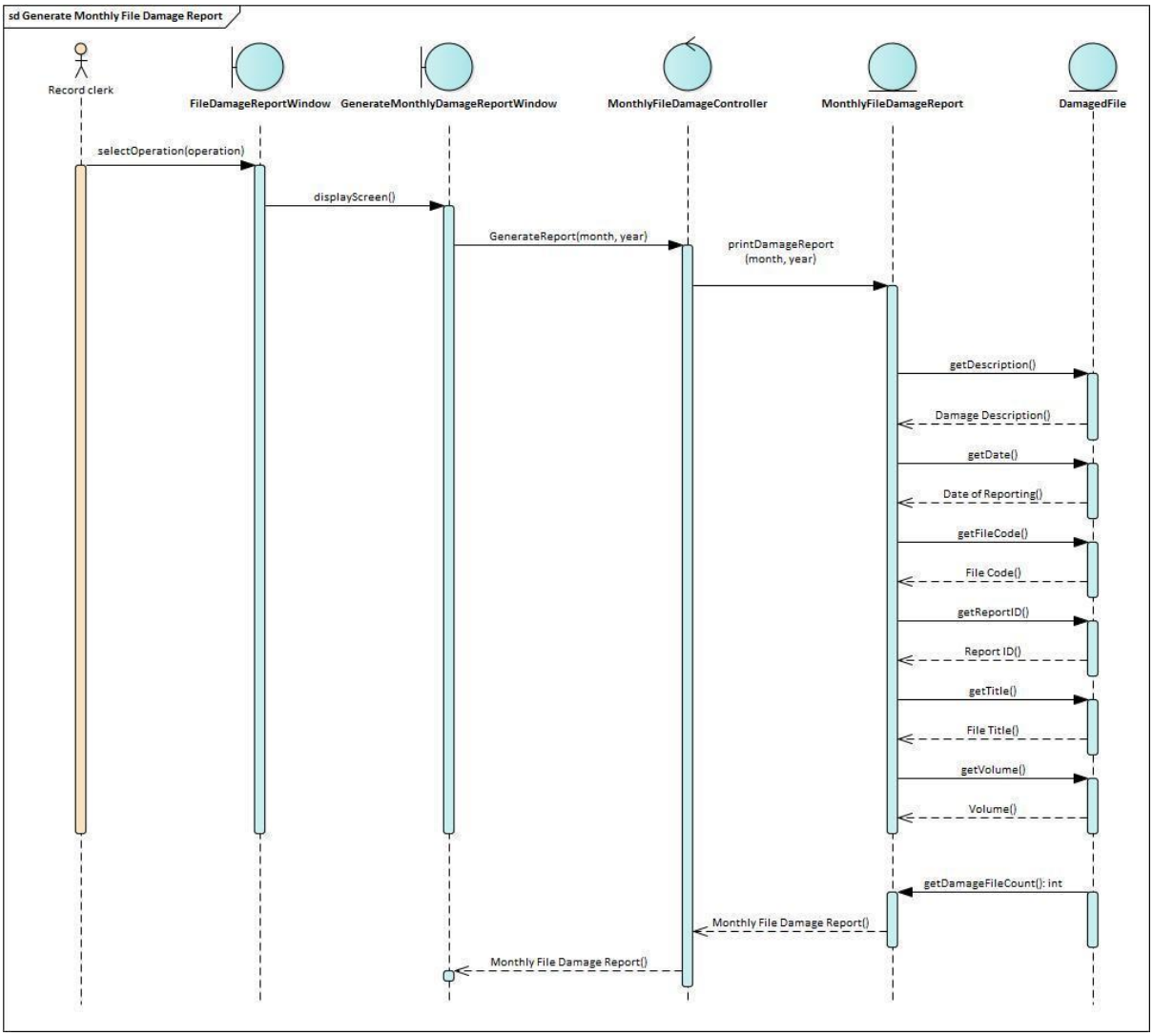
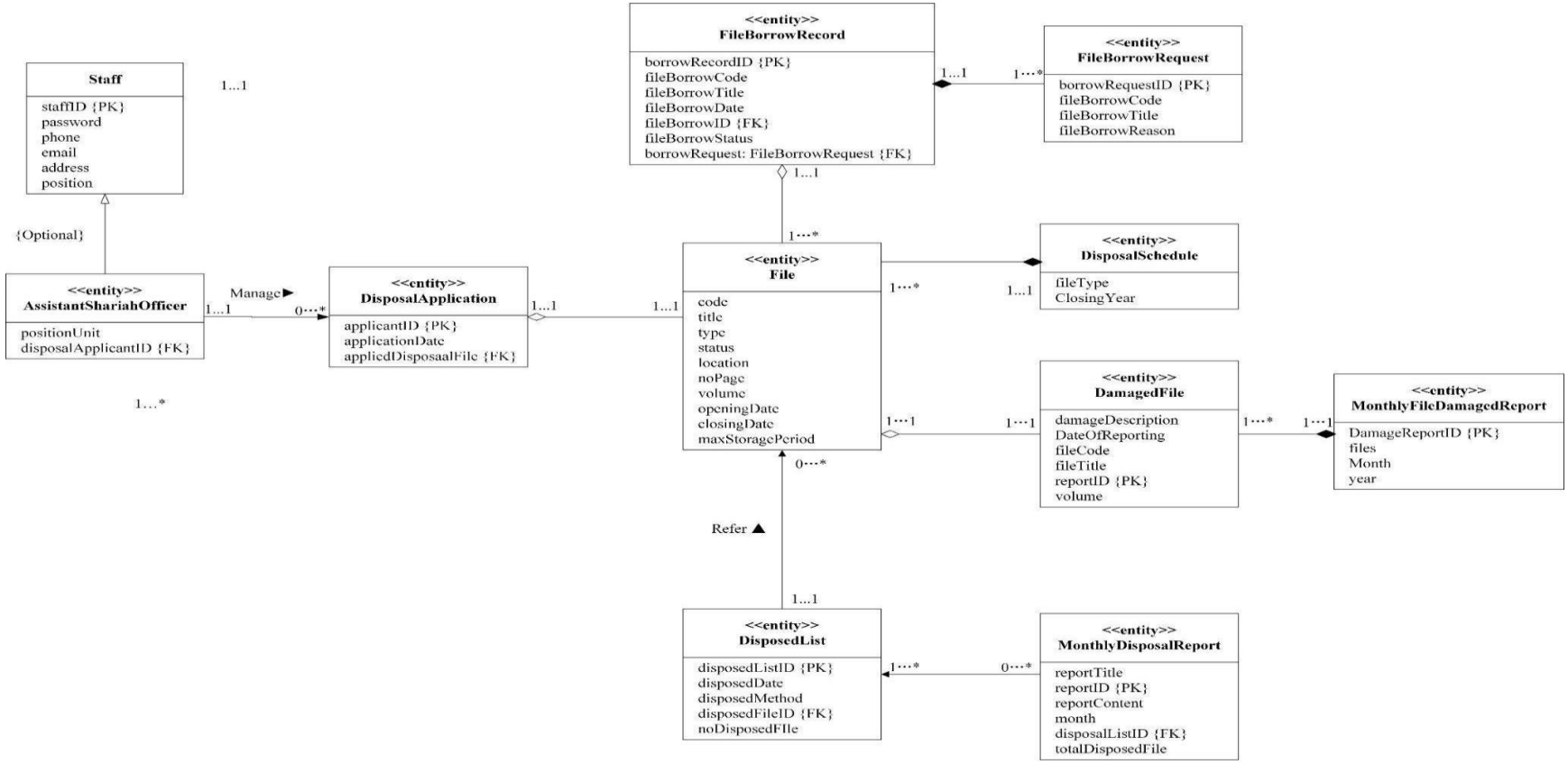


Figure 4.2.14: Sequence diagram for <Generate Monthly File Damage Report> use cas

5. Data Design

5.1 **Data Description**



The major data or systems entities are stored into a relational database named as Power File Management System database, processed and organized into 11 entities as listed in Table 5.1.

Table 5.1: Description of Entities in the Database

No.	Entity Name	Description
1.	Staff	An entity that represents the user as staff in the system is a superclass of file unit staff and administrative staff.
2.	Assistant Shariah Officer	An entity that represents the user as assistant shariah officer in the system.
3.	File	An entity that represents the file that will be manipulated throughout various business processes such as opening, closing and disposal.
4.	Disposal Schedule	An entity that represents the disposal schedule of the file.
5.	Disposal Application	An entity that represents the disposal application of the file.
6.	Monthly Disposal Report	An entity that represents the disposal monthly report that will be created for documentation of the monthly disposed files.
7.	FileBorrow Record	An entity that stores the information of file borrowing record.
8.	File Borrow Request	An entity that stores the information of file borrowing request.
9.	DisposalList	An entity that represents the list of disposed file information.
10.	DamagedFile	An entity that stores the information of damaged file.
11.	MonthlyFileDamagedReport	An entity that represents the damage file monthly report will be created for documentation of the monthly damaged files.

5.2 Data Dictionary

5.2.1 Entity: <Staff>

Attribute Name	Type	Description
staffID	varchar(20)	Primary key which is the staff ID of the staff
name	varchar(50)	Name of the staff
password	varchar(50)	Password of the staff
address	varchar(50)	Address of the staff
email	varchar(50)	Email of the staff
phone	varchar(50)	Phone of the staff
position	varchar(50)	Position of the staff

5.2.2 Entity: <Assistant Shariah Officer>

Attribute Name	Type	Description
staffID	varchar(20)	Staff ID of the assistant shariah officer
name	varchar(50)	Name of the assistant shariah officer
password	varchar(50)	Password of the assistant shariah officer
address	varchar(50)	Address of the assistant shariah officer

email	varchar(30)	Email of the assistant shariah officer
phone	varchar(50)	Phone of the assistant shariah officer
position	varchar(50)	Position of the assistant shariah officer
positionUnit	varchar(50)	Specific position of the assistant shariah officer in the file unit
disposalApplicantID	varchar(30)	Specific ID for the assistant shariah officer as disposal applicant

5.2.3 Entity: <File>

Attribute Name	**Type**	**Description**
code	varchar(50)	Primary key which stores the classification code of the file as file ID
title	varchar(50)	Title of the file
location	varchar(50)	Location of the file
type	varchar(50)	Type of the file
status	varchar(30)	Status of the file whether it is active, closed or disposed

noPage	int	Number of pages of the file
volume	int	Volume number of the file
openingDate	datetime	Opening date of the file
closingDate	datetime	Closing date of the file
maxStoragePeriod	int	Maximum storage period for the years of the file can be stored

5.2.4 Entity: <DisposalSchedule>

Attribute Name	Type	Description
fileType	varchar(50)	Foreign key which represents the type of the file
closingYear	int	Foreign key which represents the closing year of the file

5.2.5 Entity: <DisposalApplication>

Attribute Name	Type	Description
applicationID	varchar(50)	Primary key which represents the disposal application ID
applicationDate	varchar(50)	Date of the disposal application for the file

noApplication	int	Number of disposal application in the system
appliedDisposedFile	File	Foreign key which represents the file for the disposal application

5.2.6 Entity: <MonthlyDisposalReport>

Attribute Name	Type	Description
reportID	varchar(50)	Primary key which is the report ID of the month for the monthly disposal report
reportTitle	varchar(50)	Report title of the month for the monthly disposal report
reportContent	varchar(50)	Report content of the month for the monthly disposal report
month	int	Month of the monthly disposal report
disposalList	DisposalList	Foreign key which represents the disposal list information in file.
totalDisposedFile	int	Total disposed file of the monthly disposal report for that particular month

5.2.7 Entity: <File Borrow Record>

Attribute Name	Type	Description
borrowRecordID	varchar(50)	File borrow record ID that will act as primary key of this entity
fileBorrowerID	varchar(50)	Administrative staff ID that borrow the file
fileBorrowCode	varchar(50)	Code of the borrowed file
fileLenderID	varchar(50)	Foreign key acts as file unit staff ID that lend or manage the file borrowing application
fileBorrowStatus	boolean	File borrowing status
fileBorrowTitle	varchar(50)	Title of the file borrowed
fileBorrowDate	varchar(50)	Date of file being borrowed
fileReturnDate	varchar(50)	Date of file will be returned
borrowRequest	FileBorrowRequest	Foreign key which represents the borrow request of the file

5.2.8 Entity: <File Borrow Request>

Attribute Name	Type	Description
borrowRequestID	varchar(50)	Primary key which is the ID of the borrow request
fileBorrowCode	varchar(50)	Code of the file requested
fileBorrowTitle	varchar(50)	Title of the file requested
fileBorrowReason	varchar(50)	Reason of the file borrowing

5.2.9 Entity: <DisposalList>

Attribute Name	Type	Description
disposalListIID	varchar(50)	Primary key which represents the ID of the disposal list
disposedDate	datetime	Disposed date of the disposed file
disposedMethod	varchar(20)	Disposal method of the disposed file
noDisposedFile	int	Number of disposed file contains in the disposal list
disposedFileID	varchar(30)	Foreign key which represents the disposed File information in File.

5.2.10 Entity: <DamagedFile>

Attribute Name	Type	Description
reportID	varchar(50)	Primary key is the ID of the damaged File.
damageDescription	varchar(1000)	Damage description of the damaged File.
DateOfReporting	datetime	Date of reporting of the damaged file.
fileCode	varchar(50)	Code of the damaged file.
fileTitle	varchar(50)	Title of the damaged file.
volume	int	Volume number of the file.

5.2.11 Entity: <MonthlyFileDamagedReport>

Attribute Name	Type	Description
DamageReportID	varchar(50)	Primary key is the ID of the File damaged report.
files	varchar(50)	Damaged file information.
Month	int	Month of printing monthly file damaged report.
year	int	Year of printing monthly file damaged report.

6. **User Interface Design**

6.1 **Overview of User Interface**

The user shall enter the login page to continue to other functionality of the system. In the main index, the user shall view the function of search file, generate report, borrow file, dispose file, open file, close file, approve file borrowing application, check dispose schedule, report file damage and generate file damage report.

If the user would like to search for file, he will be requested to input the file classification code and view the detail of file. If the user would like to generate report, he will be requested to input month and year of report intended to be generate and view the report detail of mentioned input. If the user would like to borrow file, he will be requested to input the file classification code and view the detail of file intended to be borrowed. If the user would like to dispose file, he would be request to input file details in disposal application for disposal. If the user would like to open file, he will be requested to input the category of file and then the details of the mentioned file. If the user would like to close file, he will view the list of file allowed to be closed and allow closing it. If the user would like to approve file borrowing application, he will view the list of file borrowing application and approve it. If the user would like to check dispose schedule, he will be requested to choose between view according to year or view according to file type and requested to enter year to view if view according to year or requested to enter file type if view according to file type is chosen and then view the list of file disposal schedule. If the user would like to report file damage, he will be required to input file classification code and report the file damage. If the user would like to generate file damage report, he will be requested to enter the month and year to view and view the list of file damage.

6.2 **Screen Images**

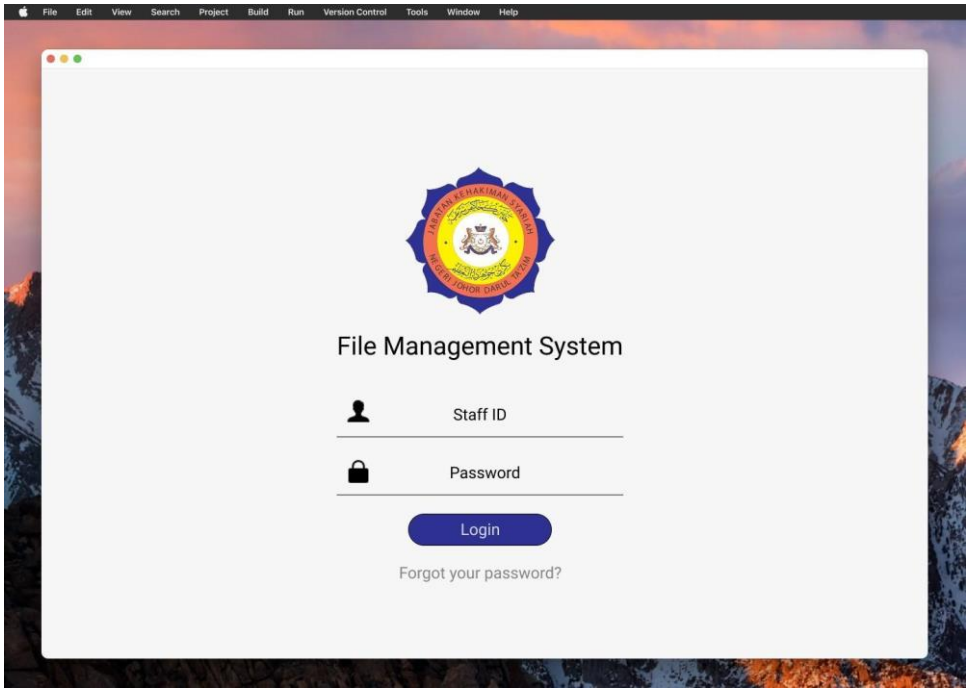


Figure 6.2.1: Interface for Login

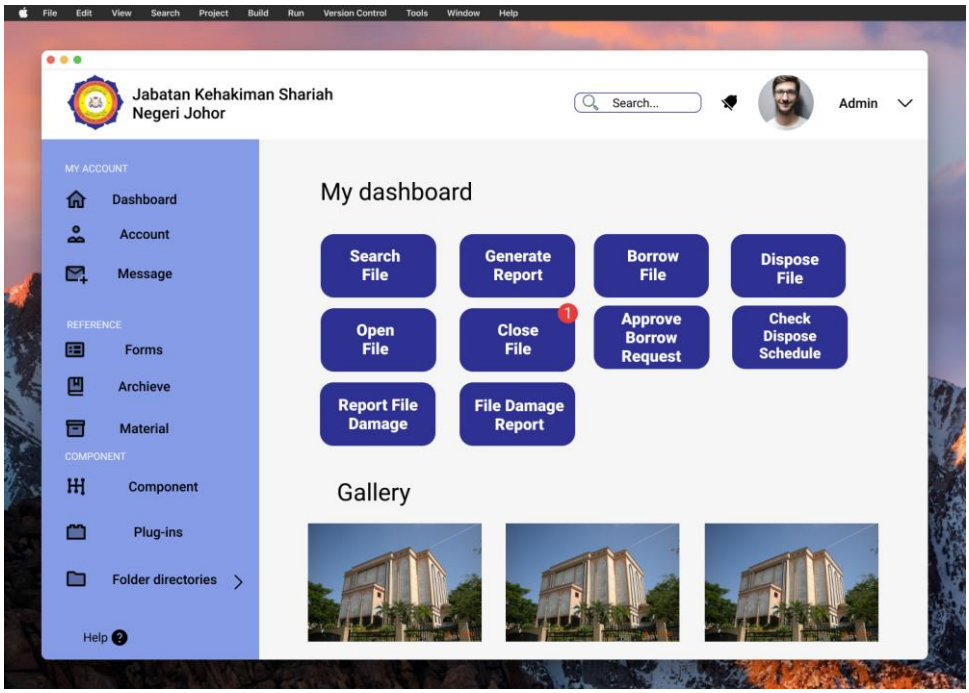


Figure 6.2.2: Interface for User Homepage

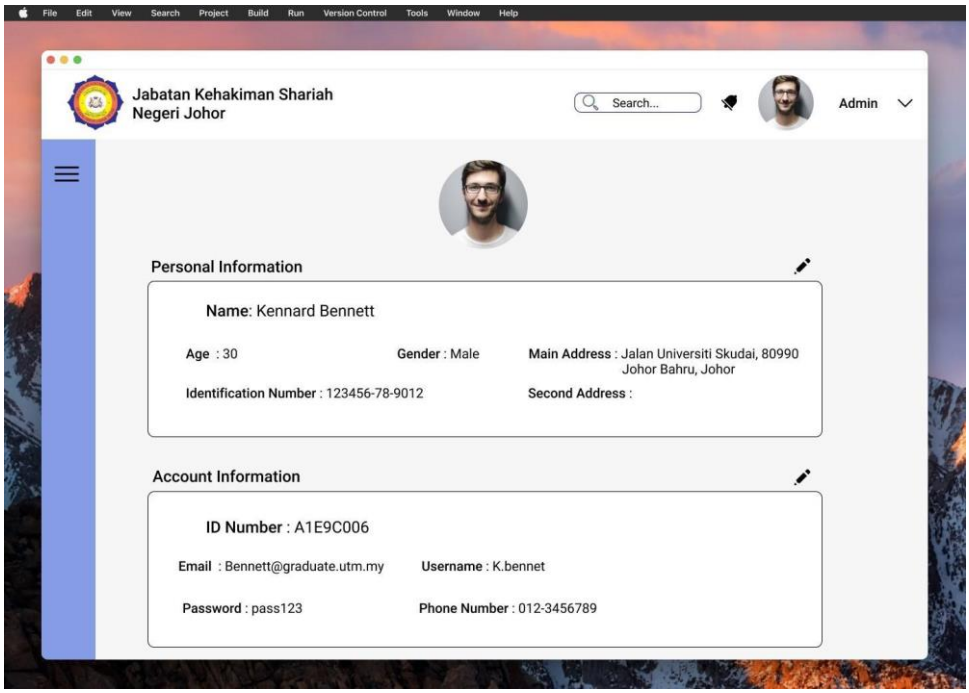


Figure 6.2.3: Interface for Edit profile Function

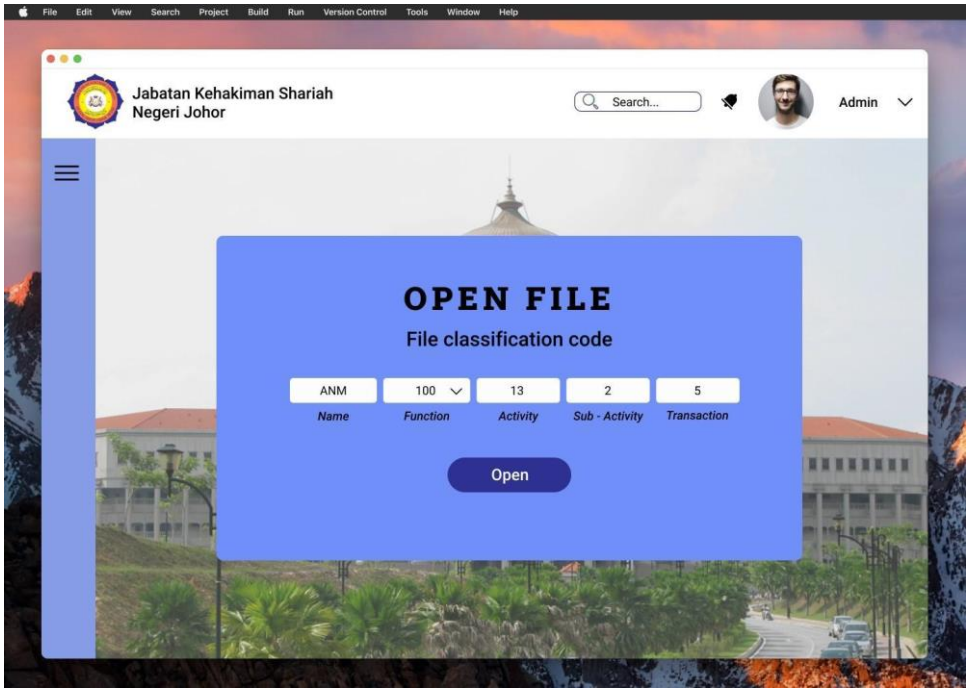


Figure 6.2.4: Interface for Open File

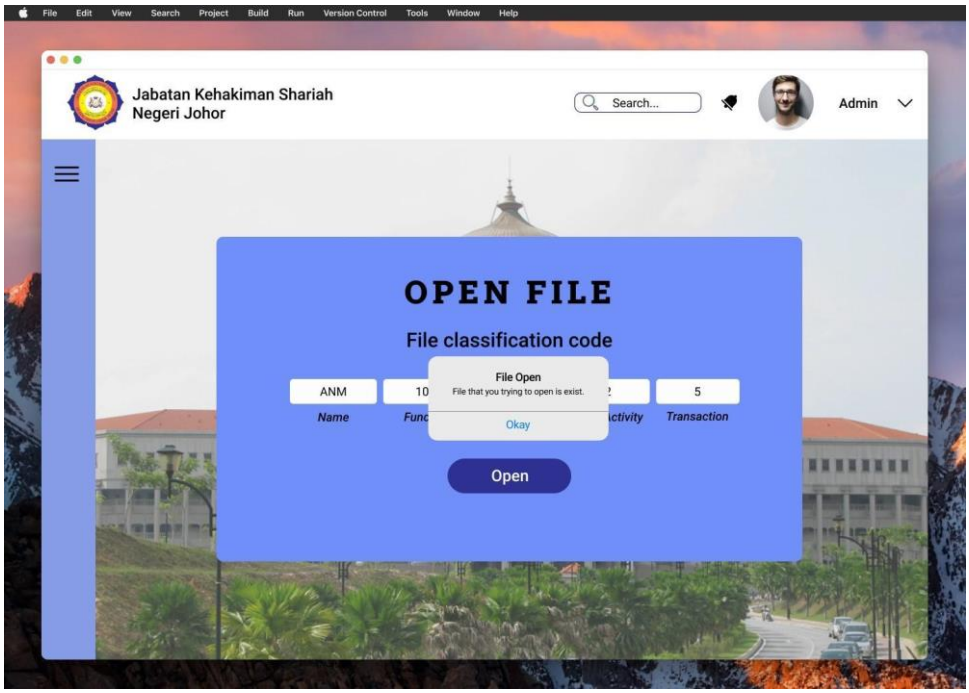


Figure 6.2.5: Interface for open file If the file exist

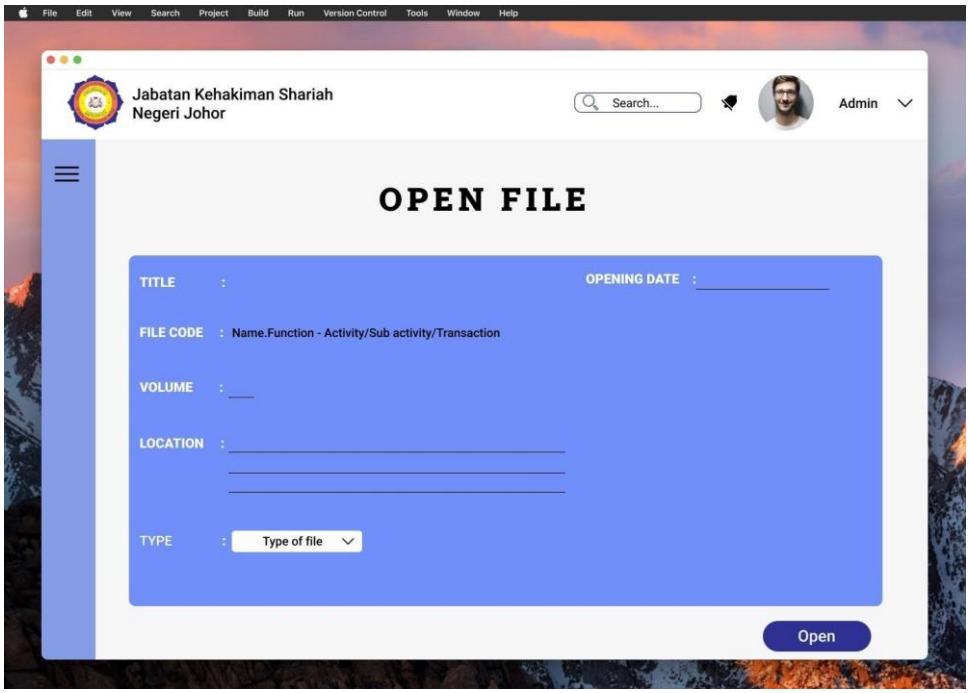


Figure 6.2.6: Interface for open file If the file doesn't exist

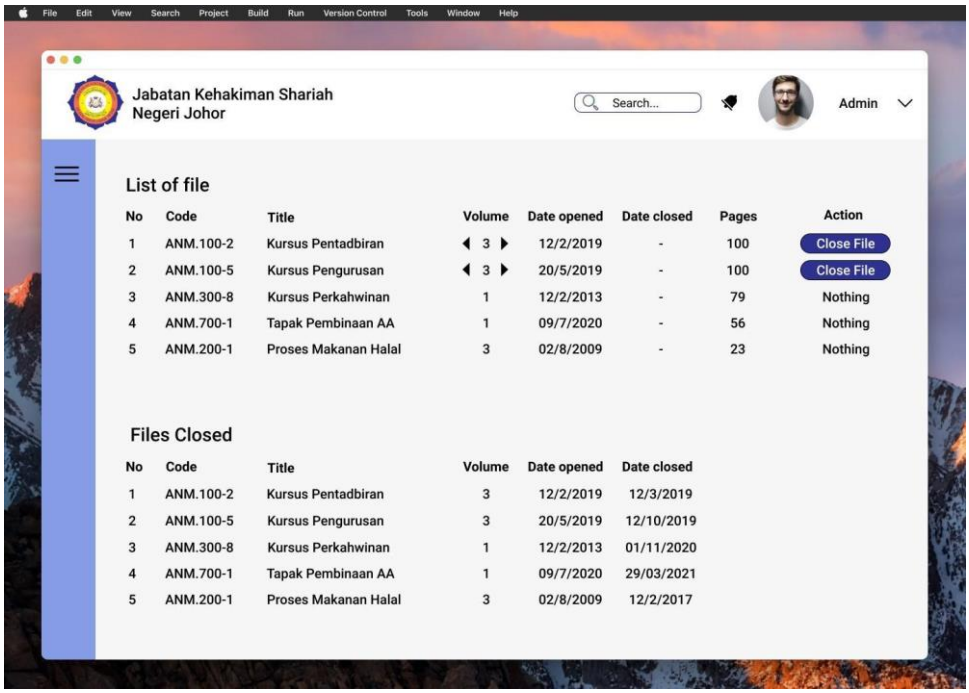


Figure 6.2.7: Interface for Close File

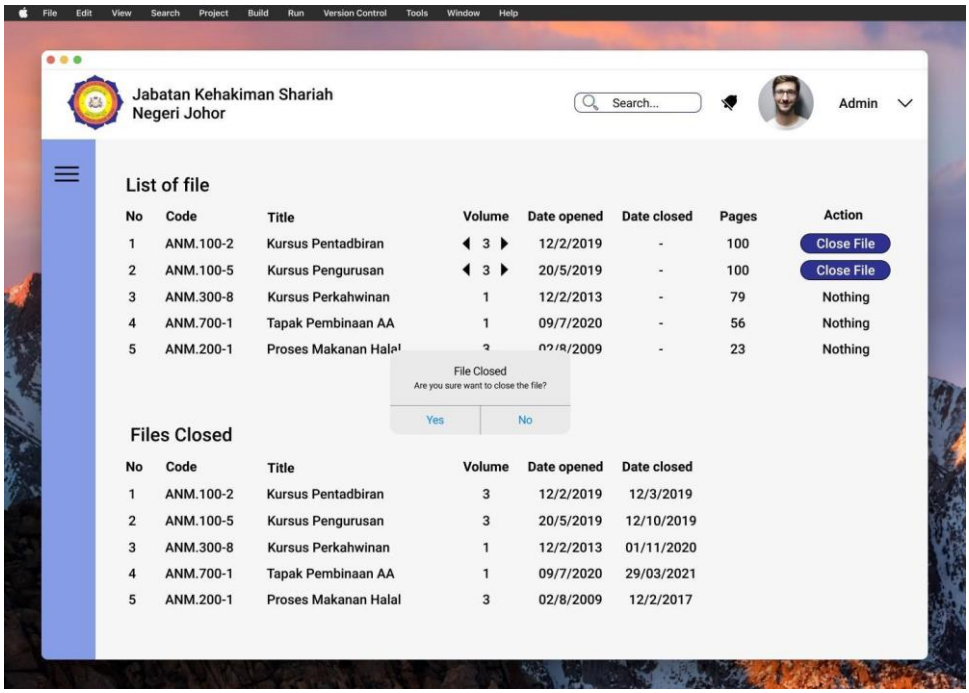


Figure 6.2.8: Interface for Close File if the user select close file

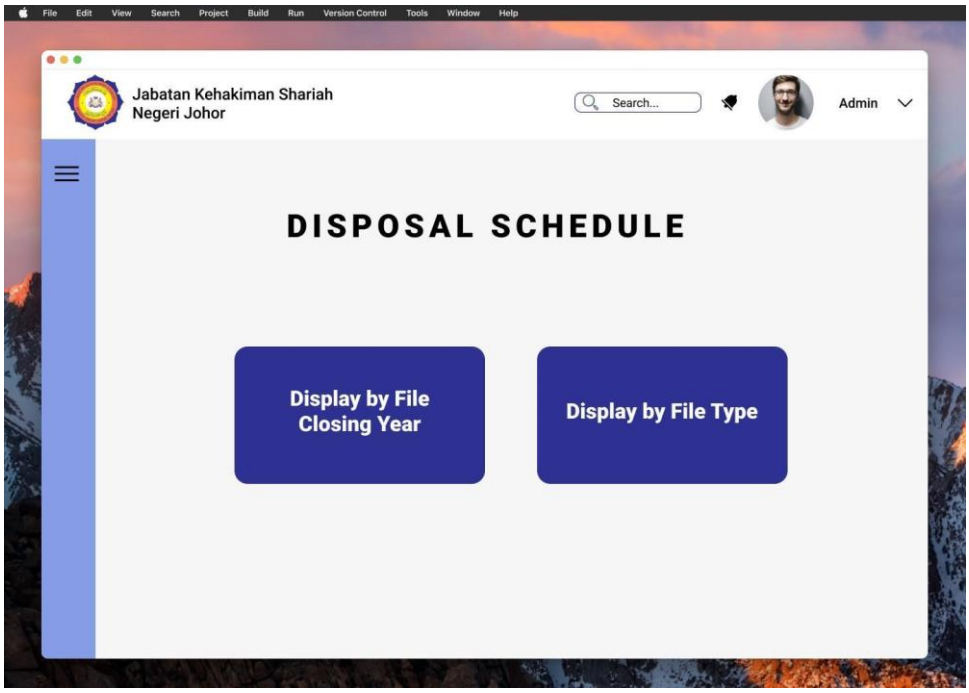


Figure 6.2.9: Interface for Check Schedule Record

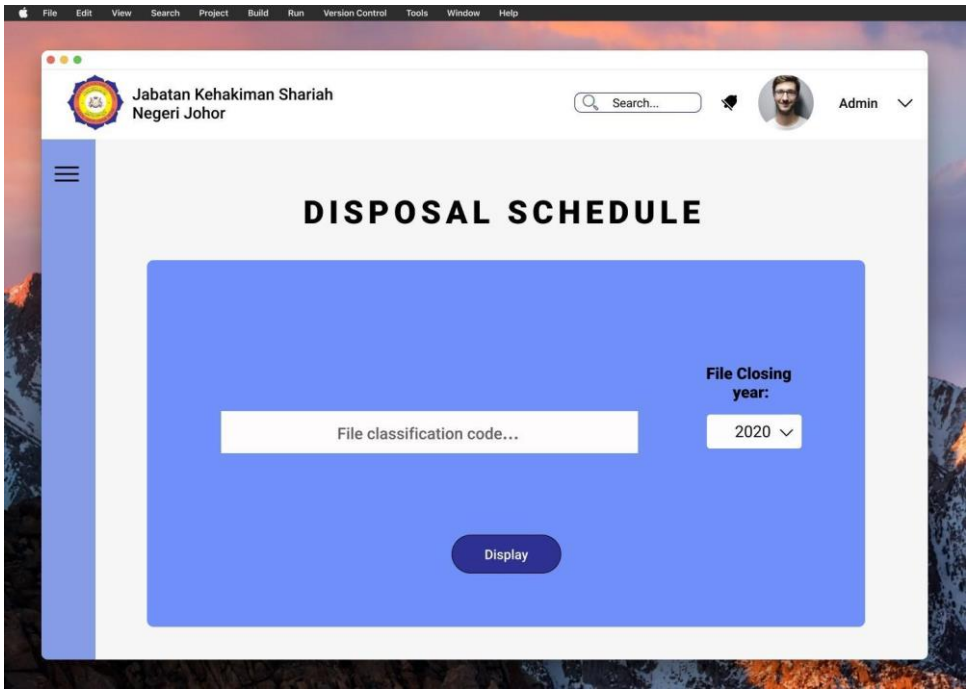


Figure 6.2.10: Interface for Check Schedule Record if user select display by year

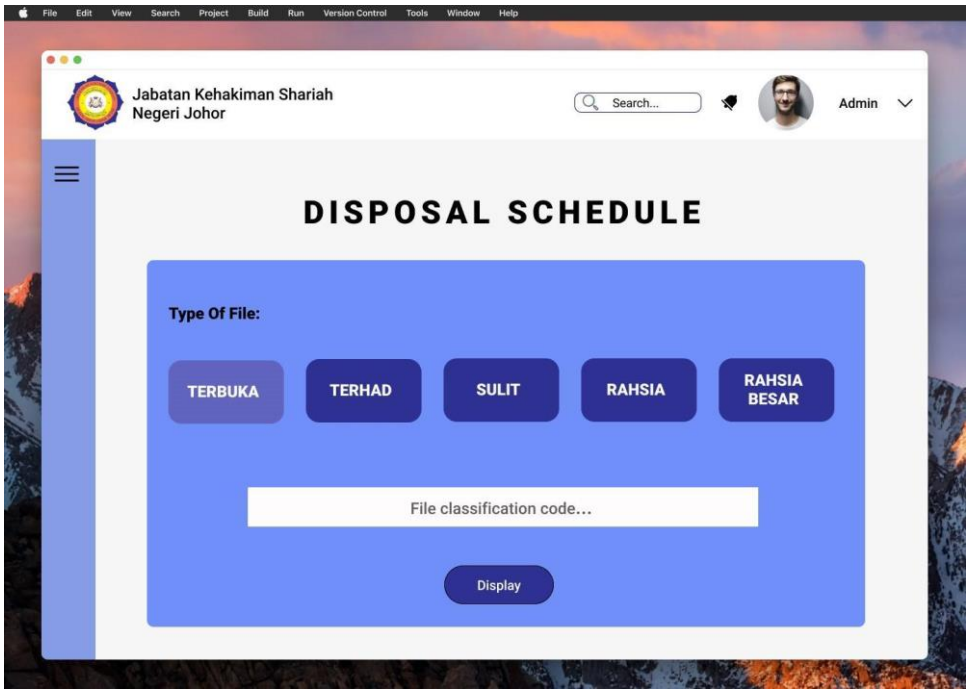


Figure 6.2.11: Interface for Check Schedule Record if user select display by file type

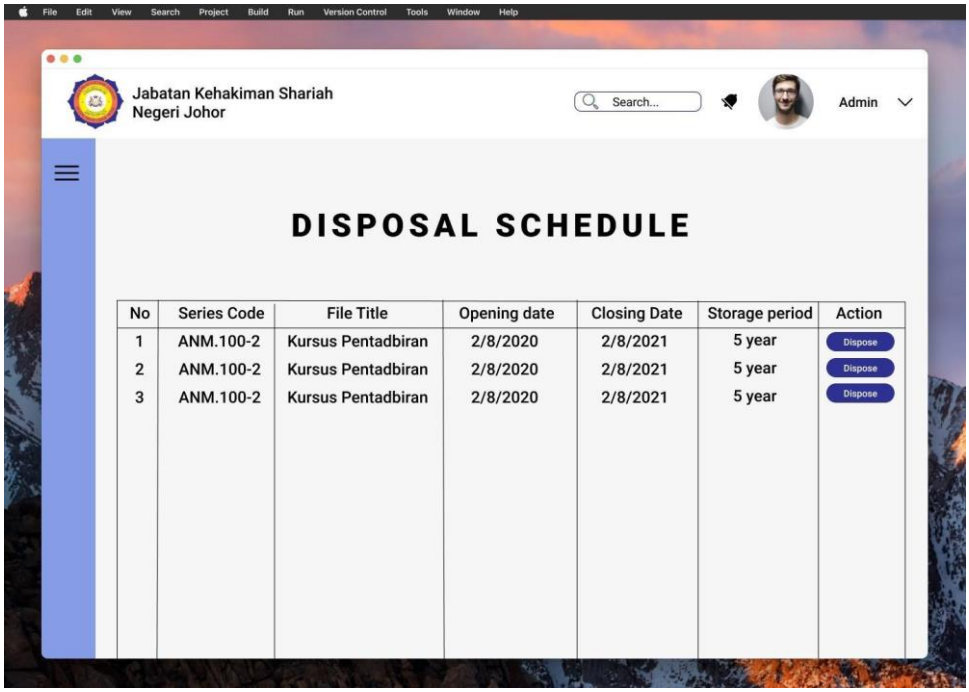


Figure 6.2.12: Interface for Check Schedule Record after enter file code

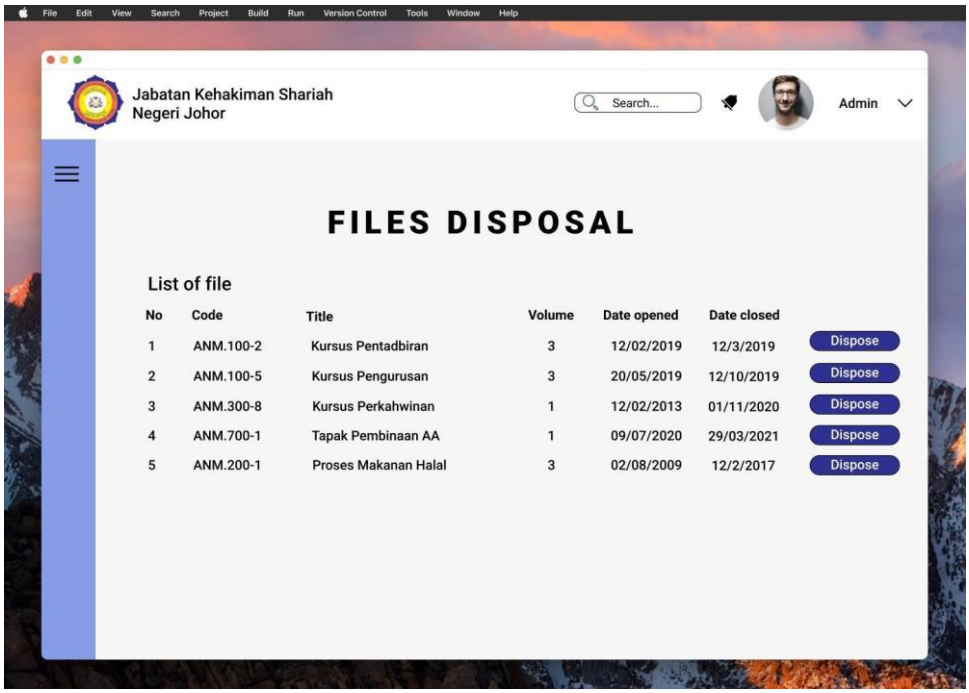


Figure 6.2.13: Interface for Dispose File

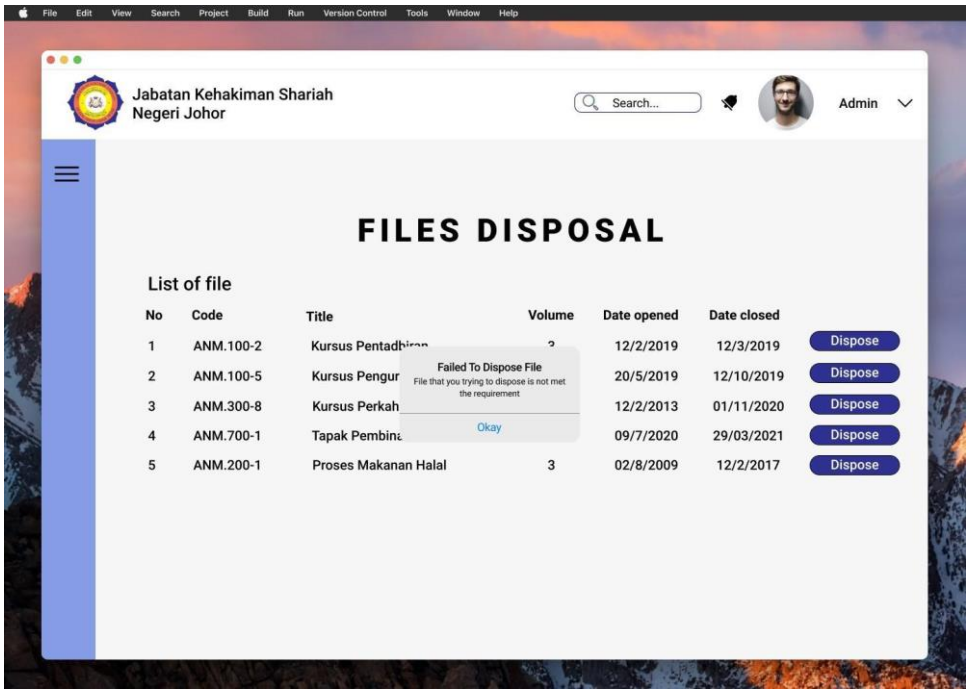


Figure 6.2.14: Interface for Dispose File if the file cannot be dispose

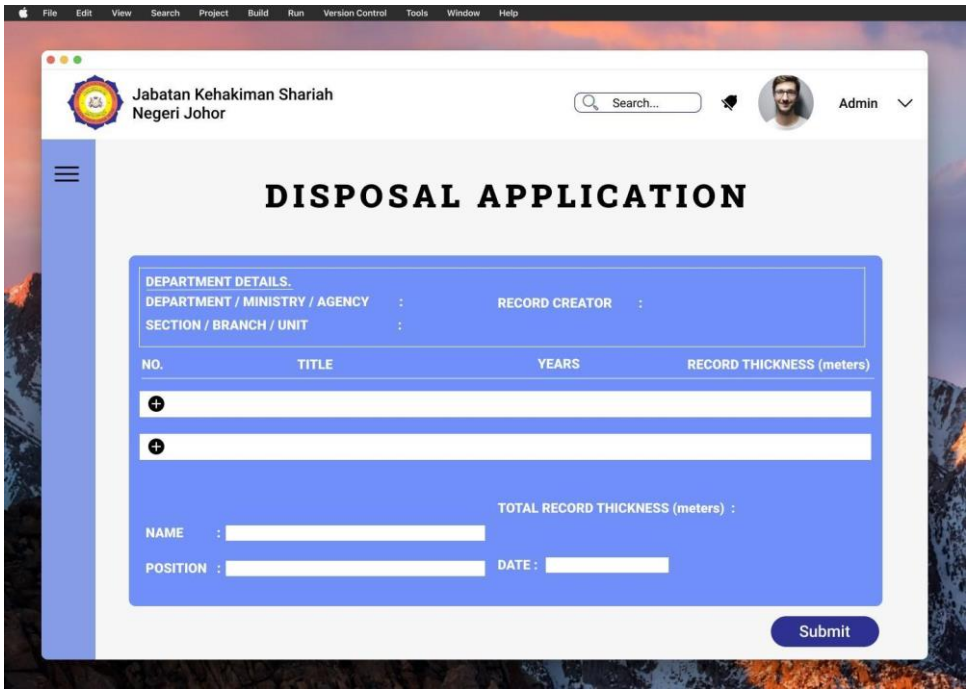


Figure 6.2.15: Interface for Disposal Application

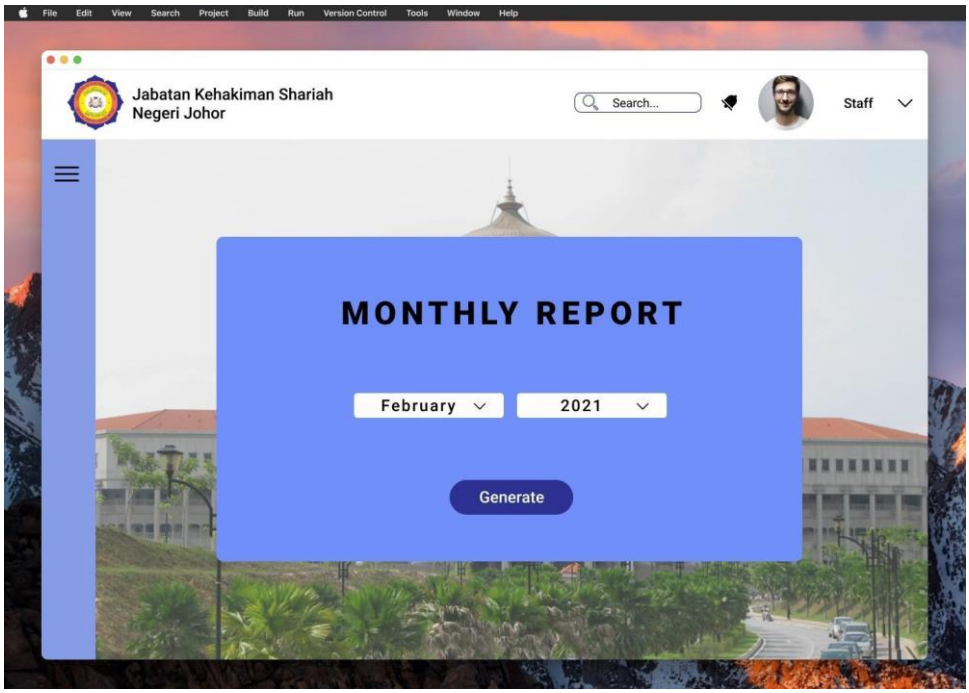


Figure 6.2.16: Interface for Generate Monthly Report

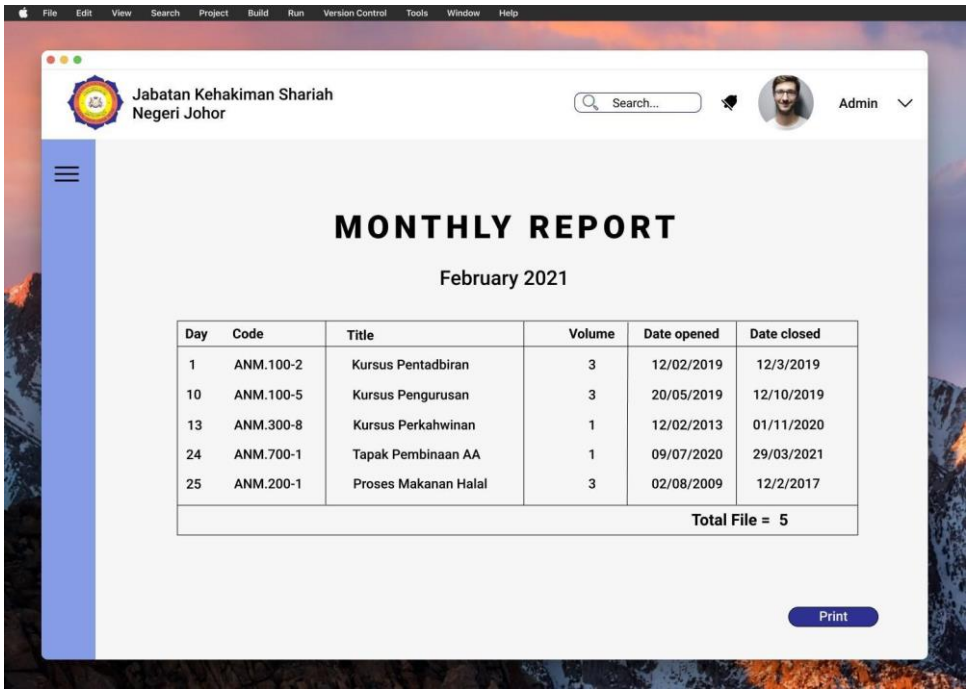


Figure 6.2.17: Interface for Generate Monthly Report 2

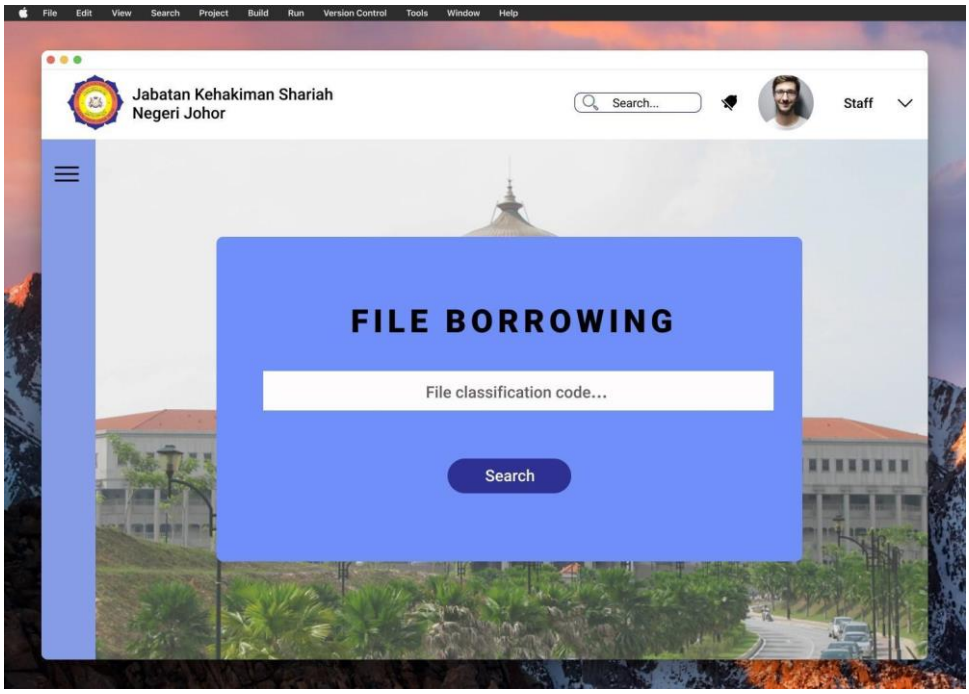


Figure 6.2.18: Interface for Borrow File

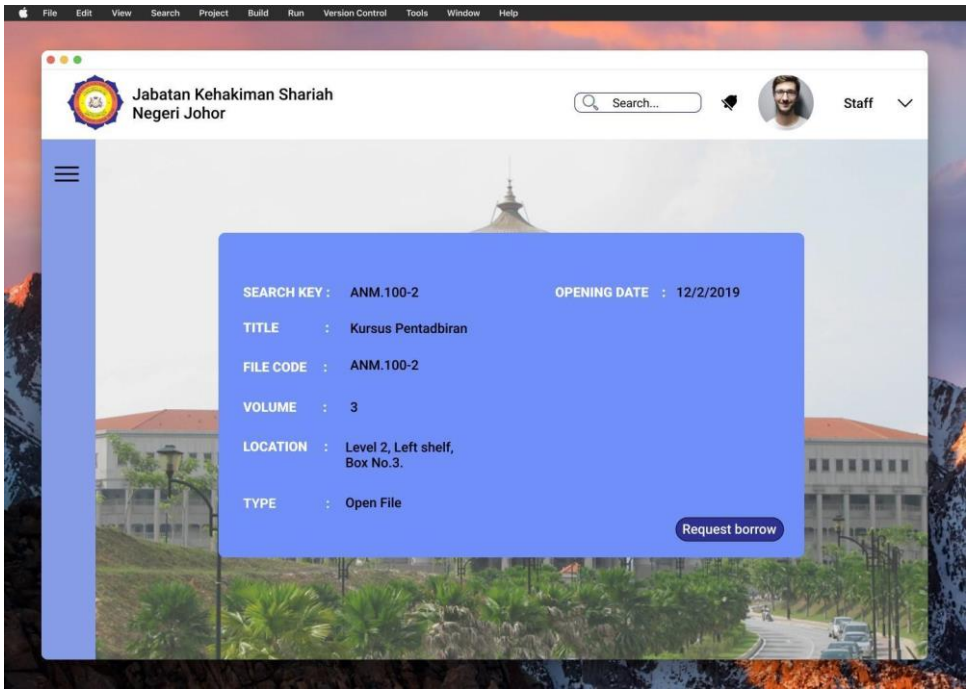


Figure 6.2.19: Interface for Borrow File 2

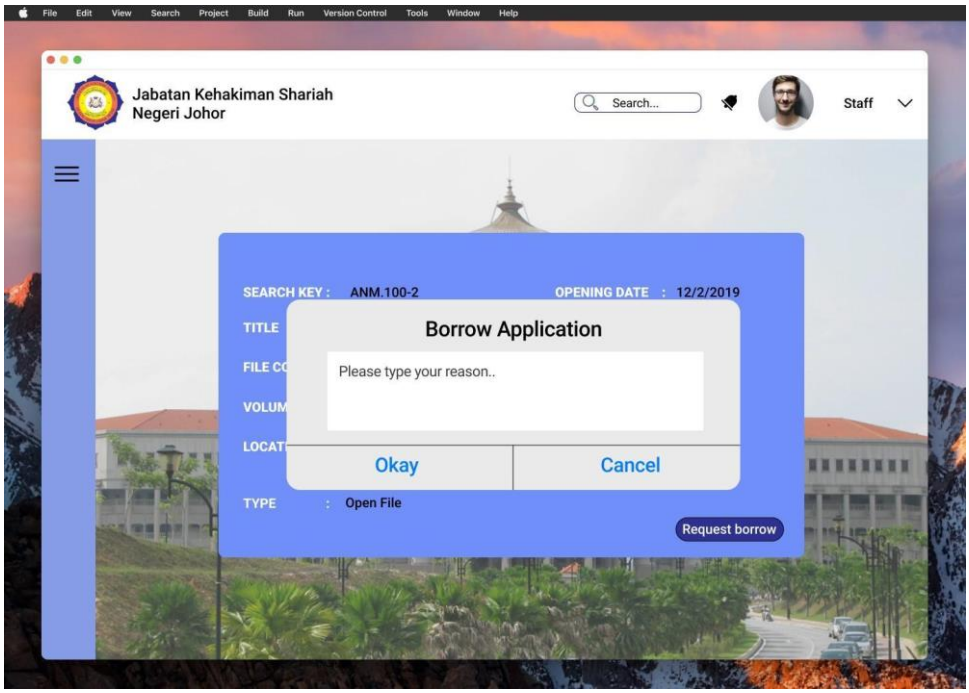


Figure 6.2.20: Interface for Borrow File 3

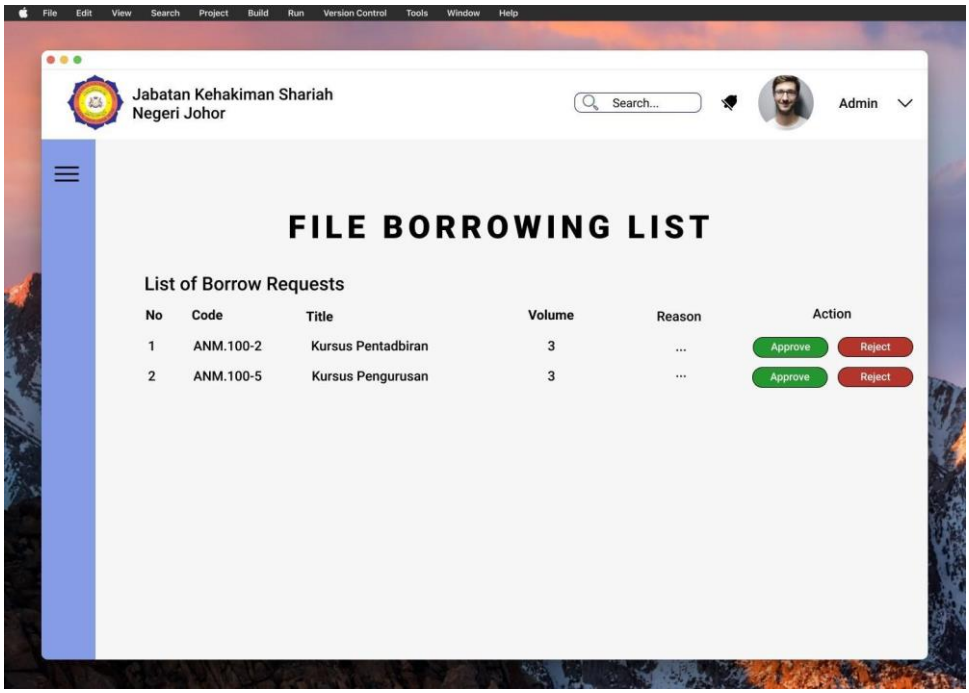


Figure 6.2.21: Interface for Approve Borrow Application

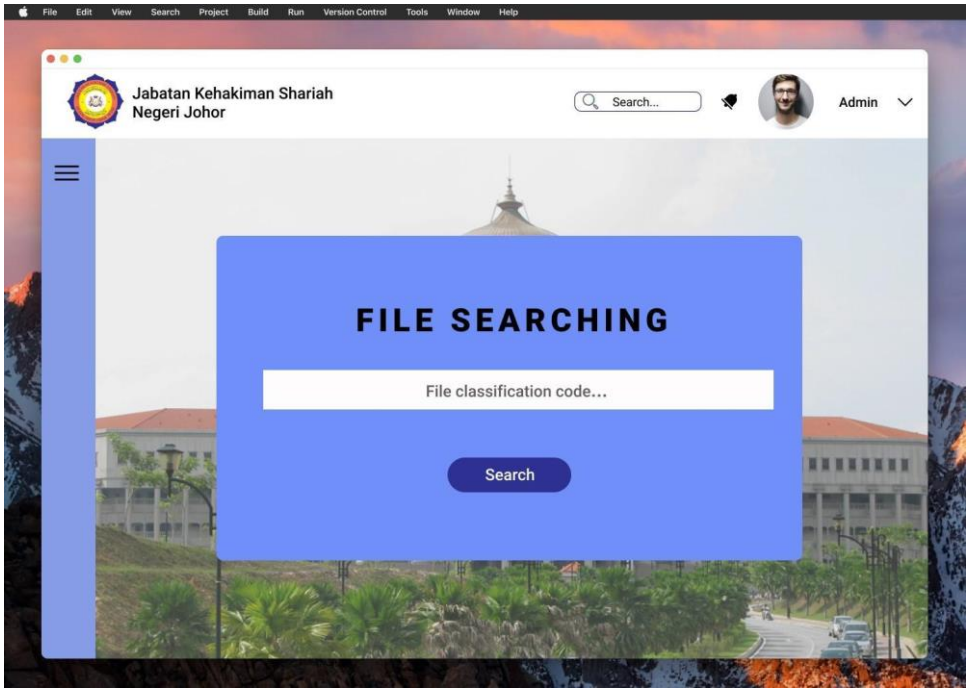


Figure 6.2.22: Interface for Search File

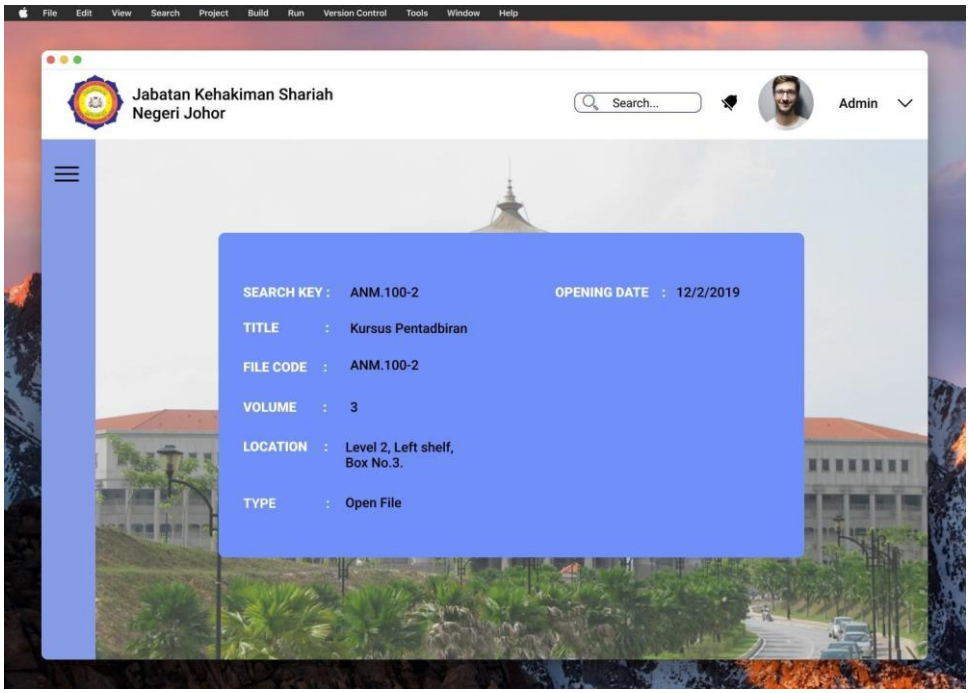


Figure 6.2.23: Interface for Search File 2

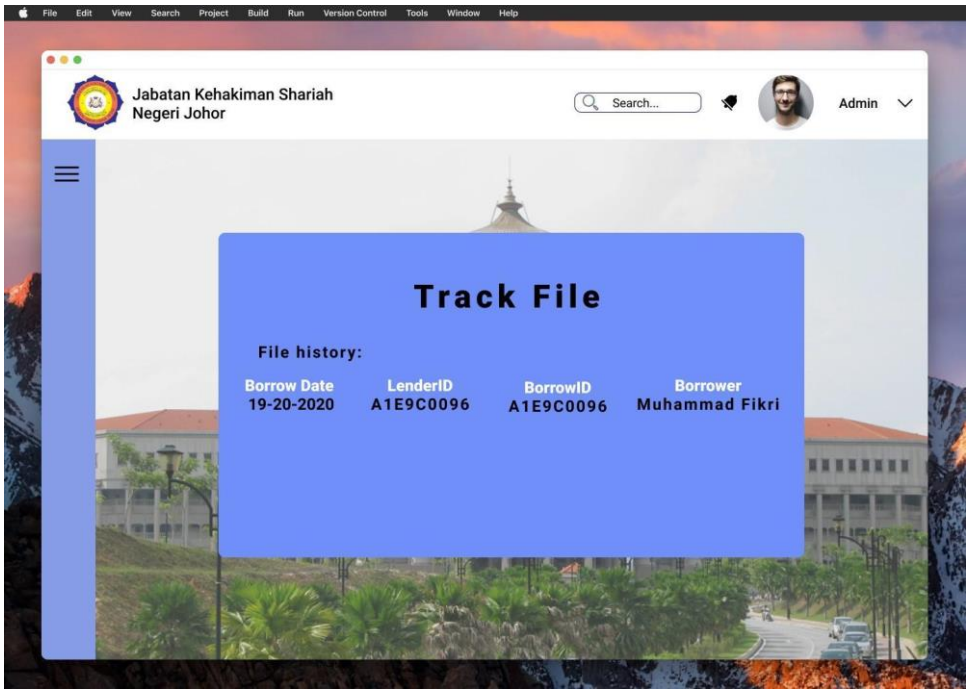


Figure 6.2.24: Interface for Tracking File

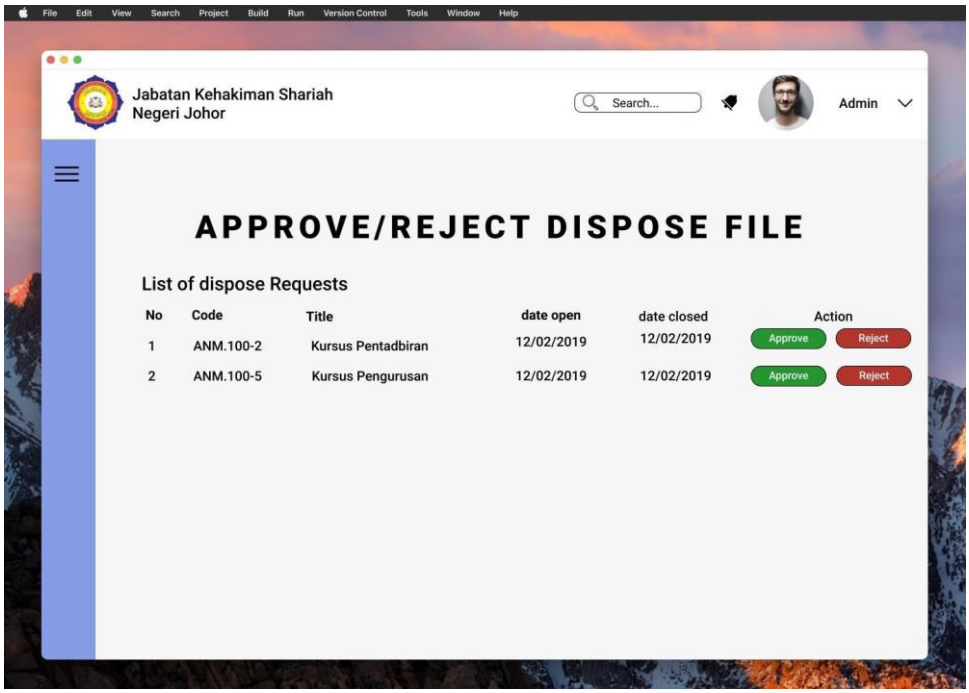


Figure 6.2.25: Interface for Approve/reject dispose File

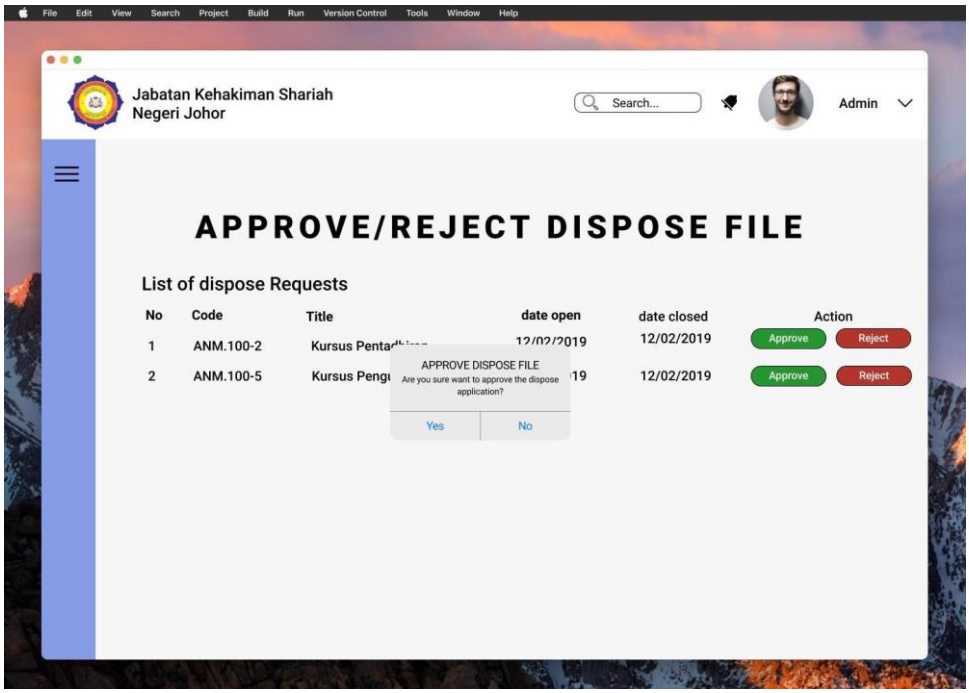


Figure 6.2.26: Interface for Approve/reject dispose File

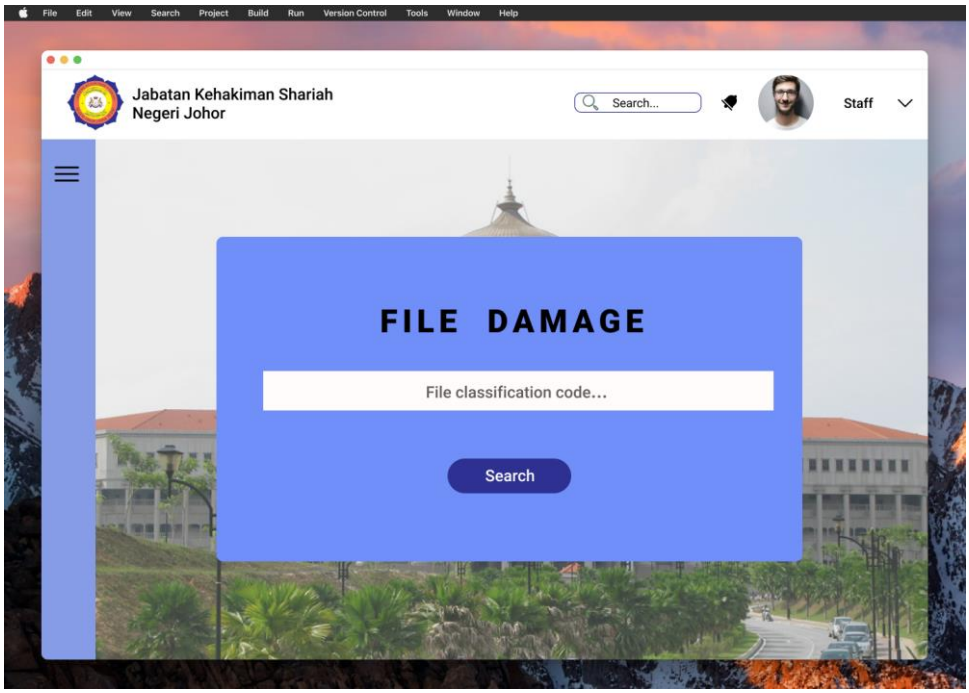


Figure 6.2.27: Interface for Report File Damage

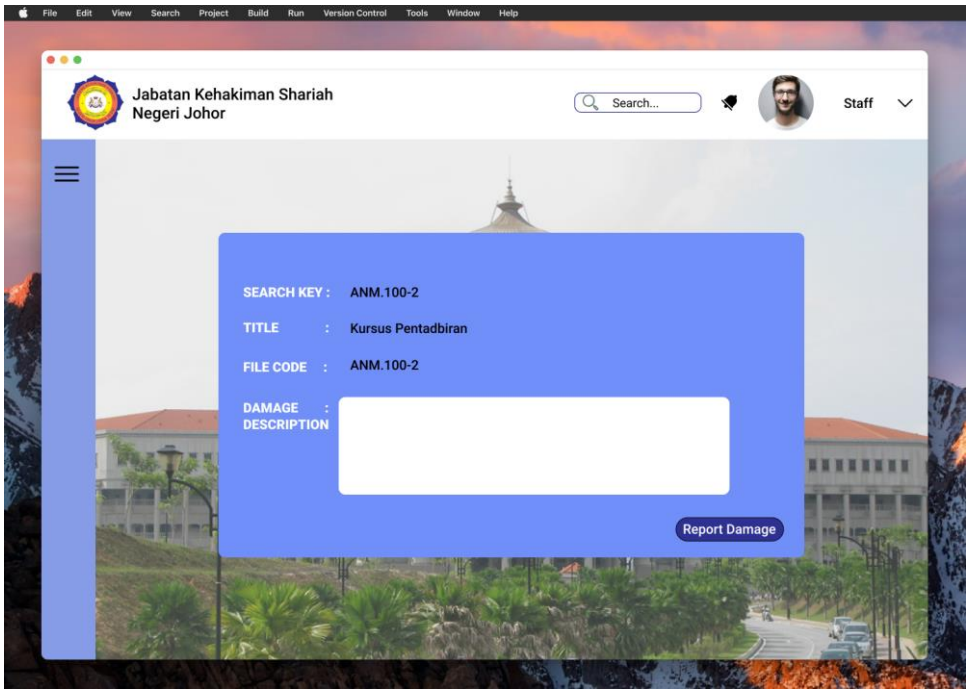


Figure 6.2.28: Interface 2 for Report File Damage

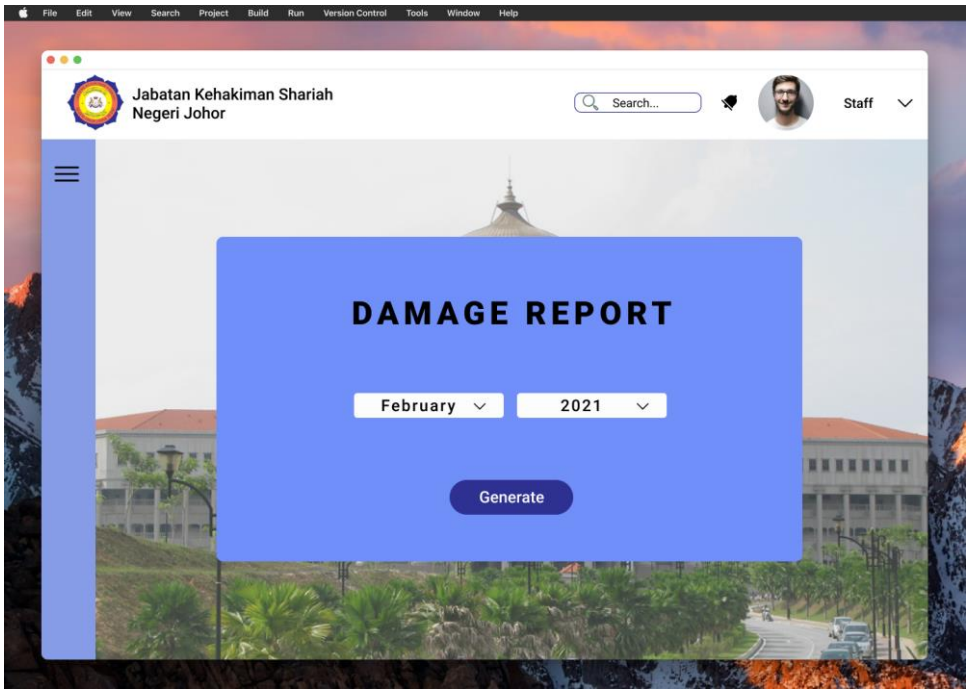
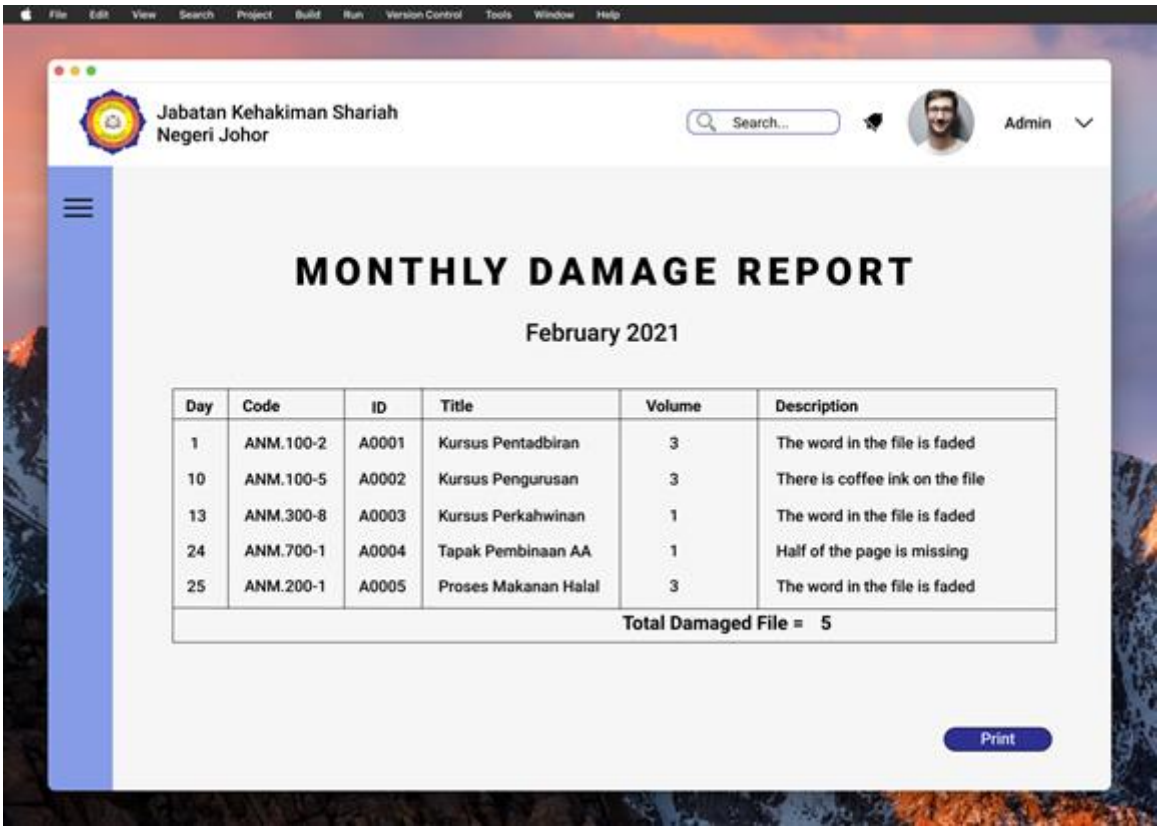


Figure 6.2.29: Interface for Generate Monthly File Damage Report



Day	Code	ID	Title	Volume	Description
1	ANM.100-2	A0001	Kursus Pentadbiran	3	The word in the file is faded
10	ANM.100-5	A0002	Kursus Pengurusan	3	There is coffee ink on the file
13	ANM.300-8	A0003	Kursus Perkahwinan	1	The word in the file is faded
24	ANM.700-1	A0004	Tapak Pembinaan AA	1	Half of the page is missing
25	ANM.200-1	A0005	Proses Makanan Halal	3	The word in the file is faded
				Total Damaged File =	5

Figure 6.2.30: Interface 2 for Generate Monthly File Damage Report

7. **Requirements Matrix**

The sequence diagrams for each use case vs. corresponding classes are as below.

Table 7.1: Description of Entities in the Database

	Assistant Shariah Officer	File	FileBorrow Record	FileBorrowReques	MonthlyDisposalReport	Staff	DisposalSchedule	DisposalApplication	DisposalList	DamagedFile	MonthlyFileDamagedReport
UC2, SD001		X				X					
UC3, SD002		X				X					
UC4, SD003	X						X				
UC5, SD004	X								X		
UC6, SD005	X							X			
UC7, SD006						X		X			
UC10,SD007						X					
UC9, SD008			X	X		X					
UC11, SD009		X				X					
UC12, SD010		X				X					
UC1, SD011						X					
UC8, SD012	X				X				X		
UC13, SD013						X				X	
UC14, SD014						X					X

8. Design Patterns

Facade

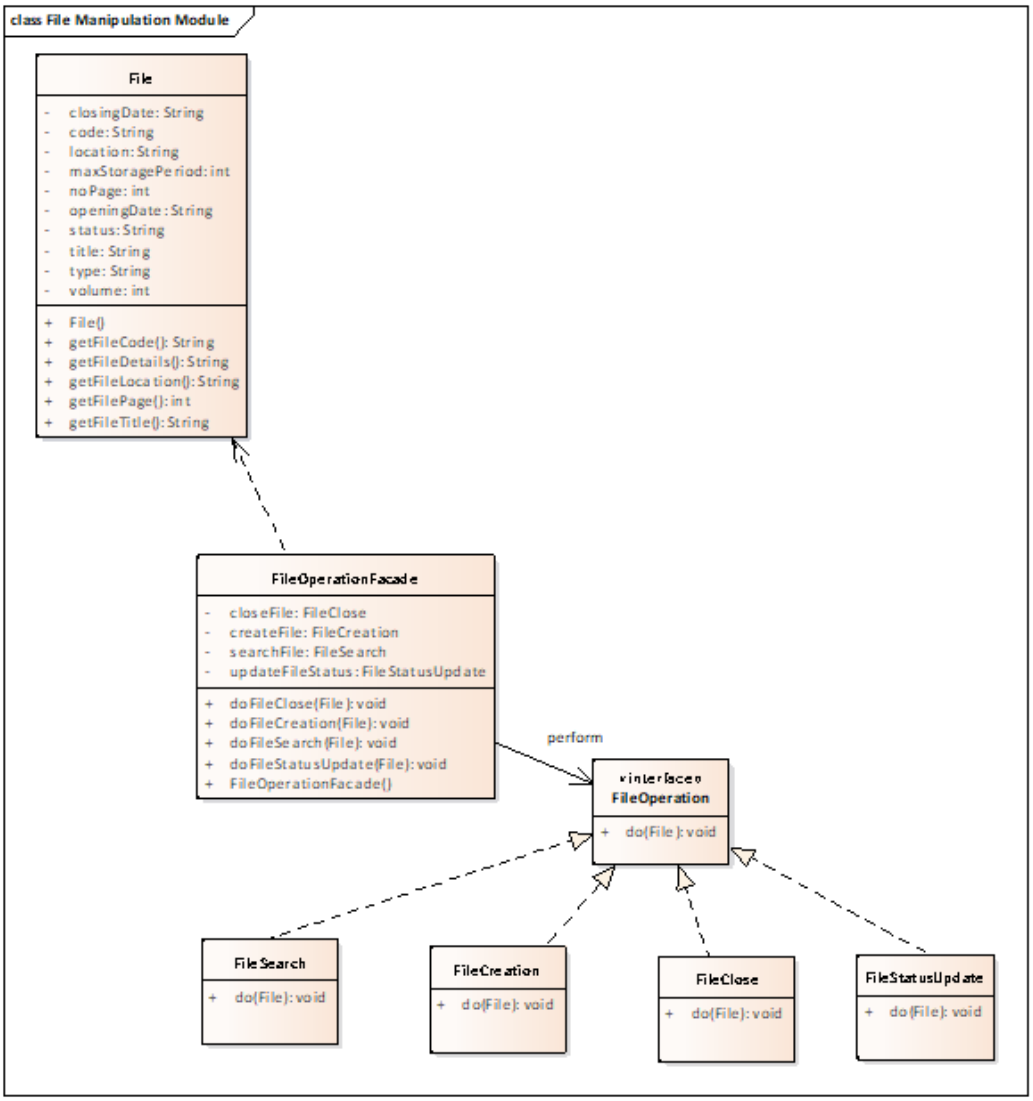


Figure 8.1: Class diagram for facade design pattern

Facade design pattern can hide the complexities of the system and provide an interface to the client using which the client can access the system. In our case, referring to Figure 8.1, we implement the facade design pattern to hide the complexity of the file operations such as operations for file searching, file creation, closing file and update file status under file class and also provide an interface which is FileOperation to the users so that they can access these file operations.The classes of FileSearch, FileCreation,FileClose and FileStatusUpdate implement the interface of FileOperation. And, the class of FileOperationFacade can use the method of the classes that implement interface FileOperation via its methods which are doFileClose, doFileCreation, doFileSearch, and doFileStatusUpdate.

Iterator

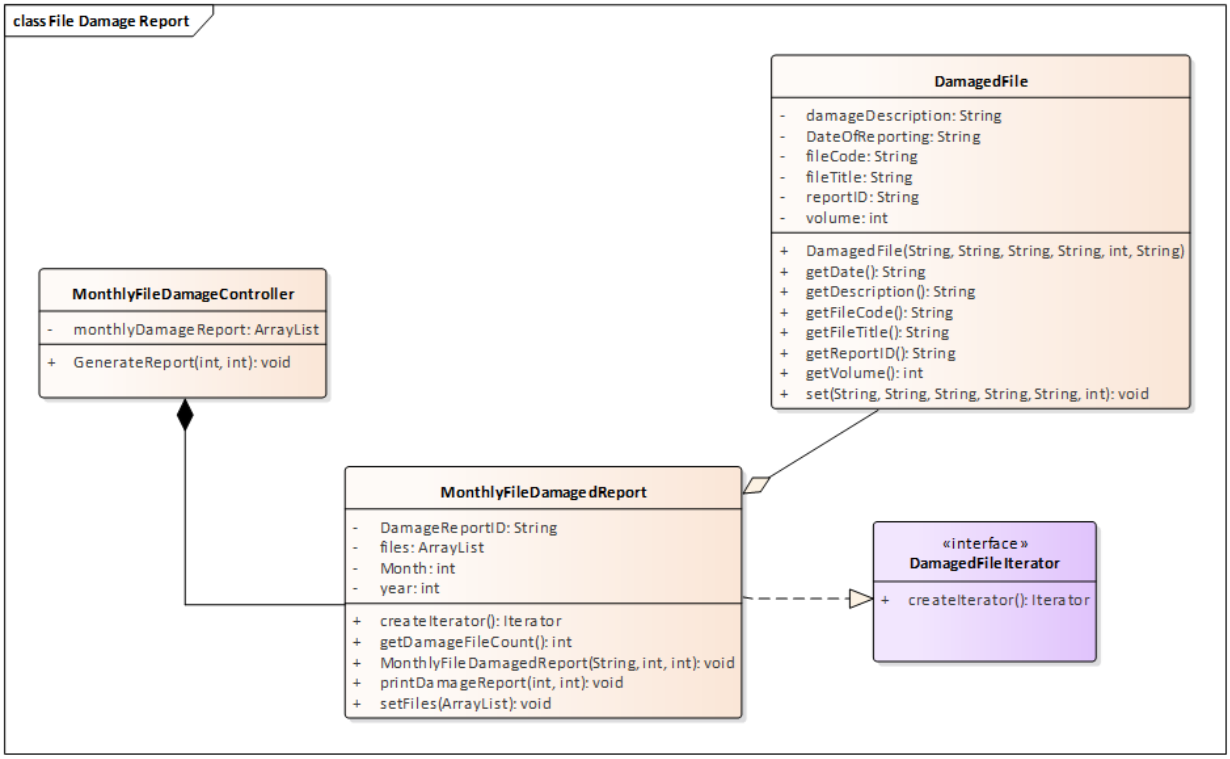


Figure 8.2: Class Diagram for Iterator Design Pattern

Iterator design pattern is an easy to access the elements of an aggregate object. An iterator itself is an object that helps to loop through the collections such as ArrayList and more than one iterator object can be created for different collection types. In our case, based on Figure 8.2, the relationship between the MonthFileDamageReport and DamagedFile class is an aggregation, where MonthlyFileDamageReport class implements the iterator interface - DamageFileIterator and it consists of an ArrayList as a collection of the objects from DamagedFile class. An iterator object is created for the ArrayList and the collection will then be looped through using hasNext() and next(), which are the methods of an iterator, for printing the details of the damaged file as a report.

9. Construction Design

9.1 Flow-based Design

9.1.1 Activity Diagram

9.1.1.1 AD013: Activity Diagram <Report File Damage>

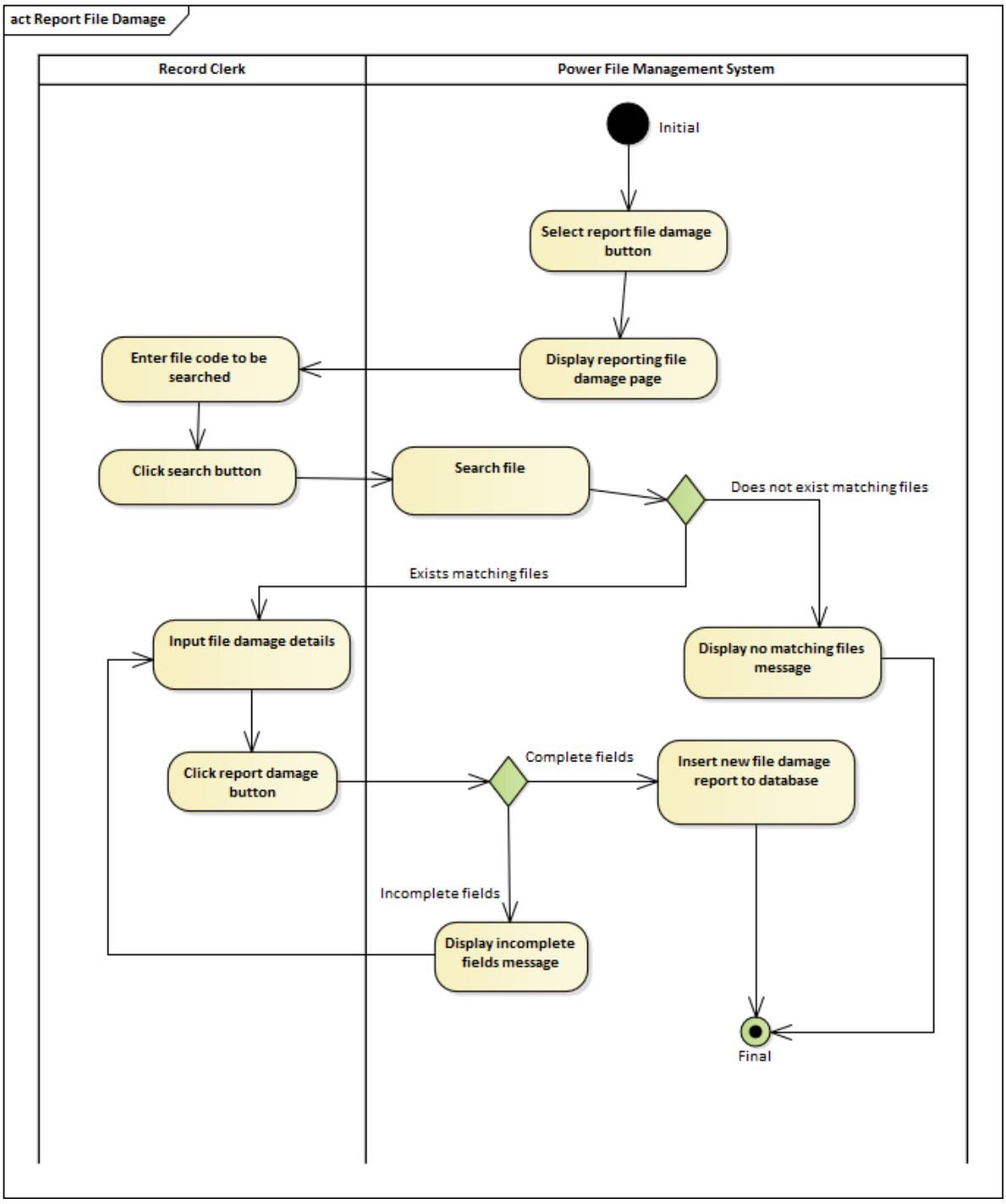


Figure 9.1.1.1: Activity diagram for <Report File Damage>

9.1.1.2 AD014: Activity Diagram <Generate Monthly File Damage Report>

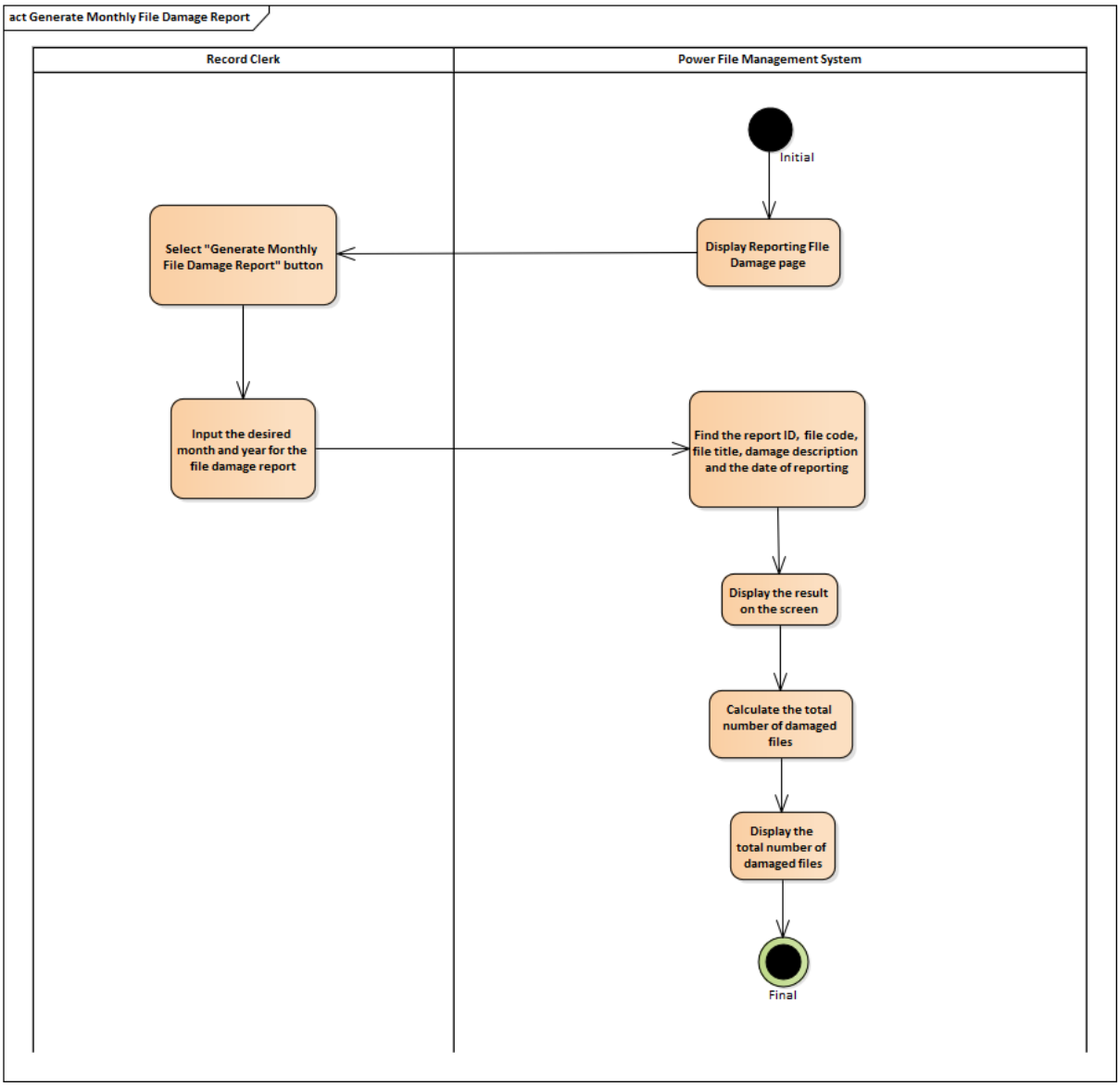


Figure 9.1.1.2: Activity diagram for <Generate Monthly File Damage Report>

9.1.2 Flow Chart

9.1.2.1 FC013: Flow Chart <Report File Damage>

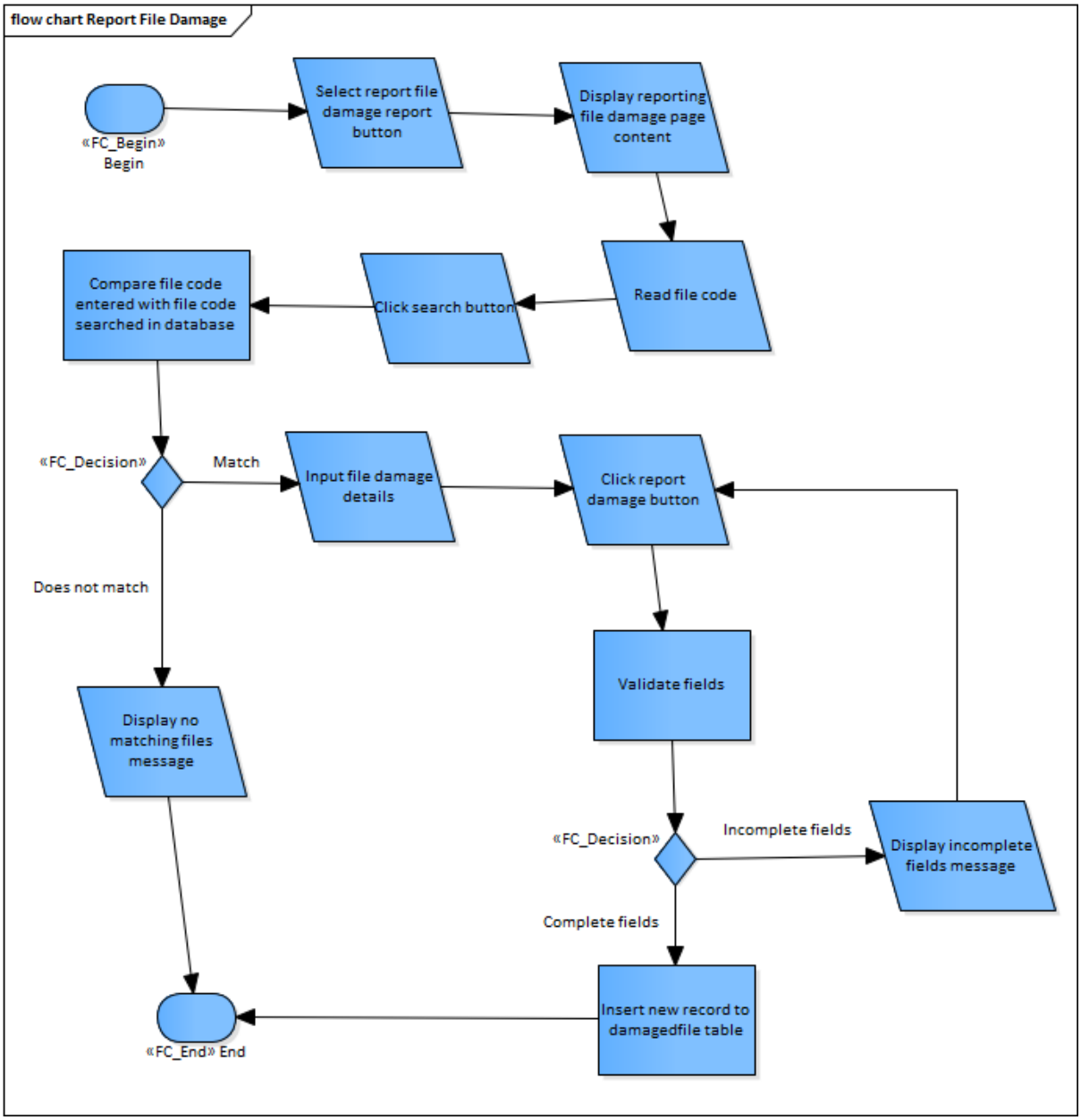


Figure 9.1.2.1: Flow chart for <Report File Damage>

9.1.2.2 FC014: Flow Chart <Generate Monthly File Damage Report>

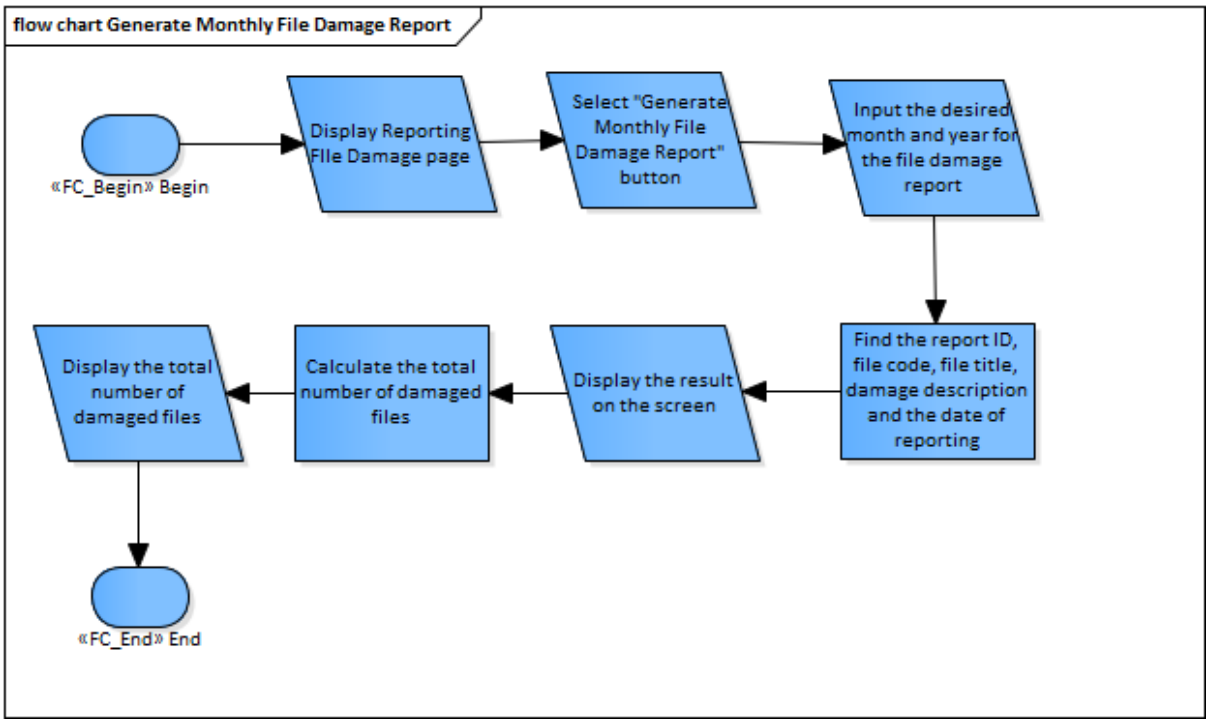


Figure 9.1.2.2: Flowchart for <Generate Monthly File Damage Report>

9.2 **State-based Design**

9.2.1 SMD009: State machine diagram <Borrow Filet>

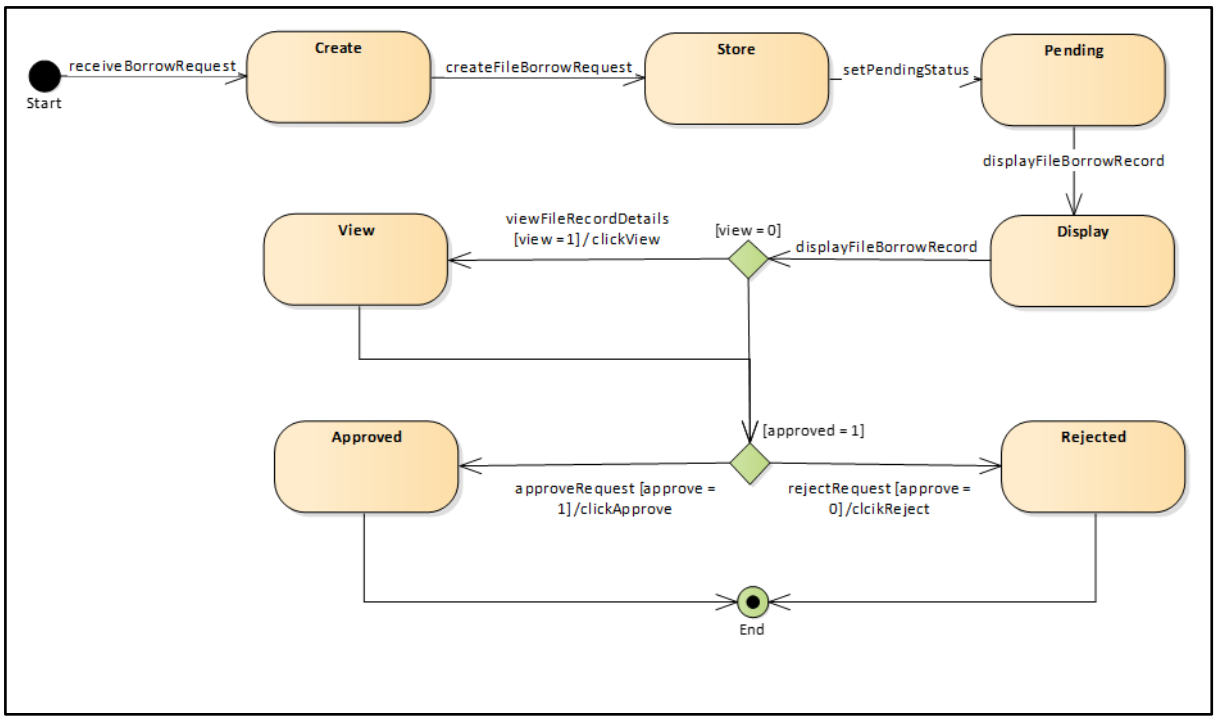


Figure 9.2: State machine diagram

9.3 **Table-based Design**

9.3.1 TDD009: Table-based design diagram <Borrow File>

Condition and action	1	2	3	4	5	6
User request for borrowing	T	T	T	T	T	T
User eligible for borrowing	F	T	T	T	T	T
User inputted in textarea	F	F	T	T	T	T
User input correct format of file classification code	F	F	F	T	T	T
File classification code exist	F	F	F	F	T	T
File to borrow exist	F	F	F	F	F	T
inform user are not eligible for borrowing	X					
prompt user to input file classfication code		X				
prompt user the correct format of file classification code			X			
inform user file classification code not found				X		
inform user file already borrowed					X	
Display file detail information						X

Figure 9.3: Table-based design diagram