SECJ2203: Software Engineering

System Documentation (SD)

Mipanzu Online File Management System

Version 3.0

17/06/2021

School of Computing, Faculty of Engineering

Prepared by: <Mipanzu>

Revision Page

1. Overview

The current version of system documentation involved description of introduction and specific requirements. The purpose, scope, definitions, acronyms and abbreviations as well as reference are described in the introduction, while the interfaces, system features which consists of use cases, requirements, constraints, and software system attributes are explained in specific requirement.

2. Target Audience

Stakeholder

3. Project Team Members

List the team members in a table by stating their roles and the status for each assigned task e.g. by sections for this SD version (complete, partially complete, incomplete). If the assigned tasks are not done and have been assigned to other team members, state accordingly.

Member Name	Role	Task	Status
HUSNY MUSHARRAF BIN SHAMSUL KAMAL	Recorder	Record the results of discussion	Complete
WAN LUQMAN BIN WAN ZULLKEFLI	Moderator	Monitor group progress	Complete
TAN CHIAW TORNG	Accuracy checker & Reporter	Check and compile the assignment	Complete
SEE WEN XIANG	Skeptic	Give query and comment on the assignment	Complete

4. Version Control History

Version	Primary Author(s)	Description of Version	Date Completed
1.0	TAN CHIAW TORNG	Completed Chapter 1 and 2, Section introduction and specific requirements	05/05/2021
2.0	WAN LUQMAN BIN WAN ZULLKEFLI	Completed Chapter 3, 4, 5, 6 and 7 Section.	06/06/2021
3.0	HUSNY MUSHARRAF BIN SHAMSUL KAMAL	Completed Chapter 8 Section and Appendices	17/6/2021

Table of Contents

1 Introduction 1

1.1 Purpose 1

1.2 Scope 2

1.3 Definitions, Acronyms and Abbreviations 2

1.4 References 2

1.5 Overview 3

2 Specific Requirements 4

2.1 External Interface Requirements 4

2.1.1 User Interfaces 4

2.1.2 Hardware Interfaces 4

2.1.3 Software Interfaces 4

2.1.4 Communication Interfaces 5

2.2 System Features 6

2.2.1 UC001: Use Case <Open new file> 10

2.2.2 UC002: Use Case <Close file> 12

2.2.3 UC003: Use Case <Track file location> 14

2.2.4 UC004: Use Case <Report the loss of file> 15

2.2.5 UC005: Use Case <Dispose of file> 17

2.2.6 UC006: Use Case <Borrow file> 19

2.2.7 UC007: Use Case <View file access record> 21

2.2.8 UC008: Use Case <Detect the late return of file> 22

2.2.9 UC009: Use Case <Return file> 23

2.2.10 UC010: Use Case <Login> 24

2.3 Performance and Other Requirements 26

2.4 Design Constraints 26

2.5 Software System Attributes 26

3 System Architectural Design 27

3.1 Architectural Style and Rationale 27

3.2 Component Model 28

4 Detailed Description of Components 29

4.1 Complete Package Diagram 29

4.2 Detailed Description 29

4.2.1 P001: <Login> Subsystem 29

4.2.1.1 Class Diagram 30

4.2.1.2 Sequence Diagram 31

4.2.2 P002: <File> Subsystem 31

4.2.2.1 Class Diagram 32

4.2.2.2 Sequence Diagram 34

5 Data Design 42

5.1 Data Description 42

5.2 Data Dictionary 43

6 User Interface Design 44

6.1 Overview of User Interface 45

6.2 Screen Images 47

7 Requirements Matrix 60

8 Test Cases 61

8.1 TC001: Test Login Subsystem 61

8.1.1 TC001_01: Test Login (SD001) 61

8.2 TC002: Test File Subsystem 62

8.2.1 TC002_01: Test Open New File with the new file code (SD002) 63

8.2.2 TC002_02: Test Close File (SD003) 64

8.2.3 TC002_03: Test Track File Location (SD004) 65

8.2.4 TC002_04: Test Report the loss of file (SD005) 66

8.2.5 TC002_05: Test Dispose of File (SD006) 67

8.2.6 TC002_06: Test Borrow File (SD007) 68

8.2.7 TC002_07: Test View file access record (SD008) 70

8.2.8 TC002_08: Test Detect the late return of file (SD009) 70

8.2.9 TC002_09: Test Return File (SD010) 71

Appendices 72

Appendix A: Traceability Matrix 72

1. Introduction

1.1. Purpose

The purpose of the SD created is used in order for us to describe all the requirements elicit from the file management system in JKSNJ. For this part, it only contains SRS documentation. SRS describes what the software will do and how it will be expected to perform, and the functionality the product needs to fulfill all stakeholders needs (Kruger, 2018). We want to create and develop a new file management system that will be changed from manual system to online system. SD is best used for us to document the functional requirements of the system that need to be developed. The document will specify all the user expectations toward what the software is able to do. In the SRS, we will use a use case description along with a sequence diagram for each use case in order to get a more detailed understanding towards each process in the system. All the documents related to the file management system also will be recorded in the SD. It is also easy for us to see clearly all the issues and improvements that can be made towards each process. All the requirements gathered in the SD are elicited from the stakeholders of JKSNJ on 11 April 2020. The parties involved for this SD are JKSNJ staffs, CRU and SAO.

1.2. Scope

The software product is Mipanzu Online File Management System to replace the current manual file management system from JKSNJ. This file management system has the function of keeping track of the location of the files and records. So, when the user wants to borrow the files, they need to key in the code of the file. All file access information including the user identity, date and time of borrowing and returning and file code will be stored online therefore it can replace the physical version of “Kertas Minit”. Therefore, it is easier for SAO as they can know who borrowed the file and can keep track of the file location.

All the staff need to log in with passwords to verify their identities. Different groups of staff can access different types of files according to their security levels. When the staff return files, the system will record the date and time of return. The staff can also report the loss of file in the system. The system will automatically notify the officers

about the late return of files. The system can also be used to dispose of files, open new files and close files. When they want to dispose of files, they need the approval from ANM. Therefore, with this online version approval from ANM can be done in minutes instead of hours or even days. The system will notify the officers of the files that shall be disposed of according to JPR.

On the other hand, viewing files and store files does not need approval from ANM, instead the CRU only needs to update the file status in the system. This system brings benefits to all JKSNJ staffs, CRU and SAO where they can manage the file records and the daily procedures like borrowing, returning, disposal and classification of files more efficiently. All the file records can be backup to secure those records and prevent them from missing. Our goal is to produce an online file management system that provides one-stop services to all JKSNJ staff with an objective to offer a path for the users to have an easier work in managing the files and records while maintaining its integrity.

1.3. Definitions, Acronyms and Abbreviation

SD – System Documentation

SRS – Software Requirements Specification

JKSNJ - Jabatan Kehakiman Syariah Negeri Johor

CRU - Clerk Of Record Unit

SAO - Syariah Assistant Officer

ANM - Arkib Negara Malaysia

JPR - Jabatan Pelupusan Record

1.4. References

Athuraliya, A. (2021, April 22). *What is Sequence Diagram? Complete Guide with Examples*. Retrieved from creately: https://creately.com/blog/diagrams/sequence-diagram-tutorial/

Kruger, N. (2018, October 23). *How to Write a Software Requirements Specification (SRS Document)*. Retrieved from perforce: https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document

Malaysia, A. N. (2021, April 1). *Dasar Garis Panduan Penilaian dan Pemisahan Rekod Elektronik*. Retrieved from Arkib Negara Malaysia: http://www.arkib.gov.my/web/guest/penilaian-dan-pemisahan-rekod-elektronik

Sharma, P. (n.d.). *System Documentation: Features, Purpose and Contents*. Retrieved from YourArticleLibrary: https://www.yourarticlelibrary.com/management/mis-management/system-documentation-features-purpose-and-contents-mis/70408

1.5. Overview

The current version of SD contains the information needed to explain on the specific requirement of our software. This includes external interface and system features. For external interface, we explain about the user interface, hardware interface, software interface and communication interface. For system features, we explain about our proposed system using multiple diagrams such as use case diagram, activity diagram, domain model and state machine diagram. For each use case in the use case diagram, we provide use case description. Lastly, we elaborate on the performance and other requirements in our proposed system. In the future, SD will provide more detailed description of the software architectural design, detailed description of components, data design, user interface design and test cases. The progress of SD follows the SDLC phases from requirement analysis, design until testing phase. All phases are documented with diagrams and descriptions.

2. Specific Requirements

2.1. External Interface Requirements

2.1.1. User Interfaces

a) Logical characteristics: The GUI of the system is based on HTML, CSS, and JavaScript, where it is accessible from any device with any web browser.

b) Aspects:

1. If the user does not log in, there are limited pages can be access by the user, including the login page.
2. After login, the website should show different content for different type of user respectively (Clerks of Record Unit, JKSNJ staff, Syariah Assistant Officer)
3. On the event of website server down or error, it should show error message understandable by the users.

2.1.2. Hardware Interfaces

The Mipanzu Online File Management System is a web-based system, hence it required:

Web server RAM: 12GB

Database server: 1TB

Internet bandwidth speed: 1 GB/s

Website requirement: support mobile and PC with web browser

2.1.3. Software Interfaces

- Database Management System is used to create and maintain a database. It provides services like concurrency, backup, consistency and security of data.

Name: PHPMyAdmin (MySQL)

Mnemonic: PHPMyAdmin (MySQL)

Specification Number: N/A

Version Number: PHPMyAdmin 5.1.0

Source: phpMyAdmin

- Web host is used to host a website and manage the request and content of a website. Bluehost consists of blue host panel that manages the database, file management, email and more features.

Name: Bluehost

Mnemonic: Bluehost

Specification Number: N/A

Version Number: N/A

Source: Best Web Hosting 2021 - Domains - WordPress - Bluehost

- Operating system

Name: Window 10

Mnemonic: Window 10

Specification Number: 10240

Version Number: 2004

Source: Download Windows 10 (microsoft.com)

- Mail system

Name: Google Mail

Mnemonic: Google Mail

Specification Number: N/A

Version Number: N/A

Source: Gmail - Email from Google

Purpose: It links to the Google Mail to send email to Arkib Negara Malaysia regarding the loss of file or to get approval for file disposal.

Message content and format: Email format in Google Mail with attachment in the form of jpeg, excel or word document.

2.1.4. Communication Interfaces

Local network protocols applied are HTTPS and FTP protocol. HTTPS protocol is used to send data between web browser and server while FTP protocol allows file transfer.

2.2. System Features

The system features include several diagram that helps us to create a good to-be system in the future. There will be several diagram created and all of them are used to explain all the requirement and process needed for Mipanzu File Management System. The diagram that will be created are use case diagram to show interaction between user and all the use case in the system, activity diagram to documents the action sequences in the use case, domain model to show all the class with their attributes in the system and state machine diagram to show the state of the class when there is a trigger that made the state of the class to change if any. For every use case in the use case diagram, we will create use case description and sequence diagram to see the flow and process for every use case.

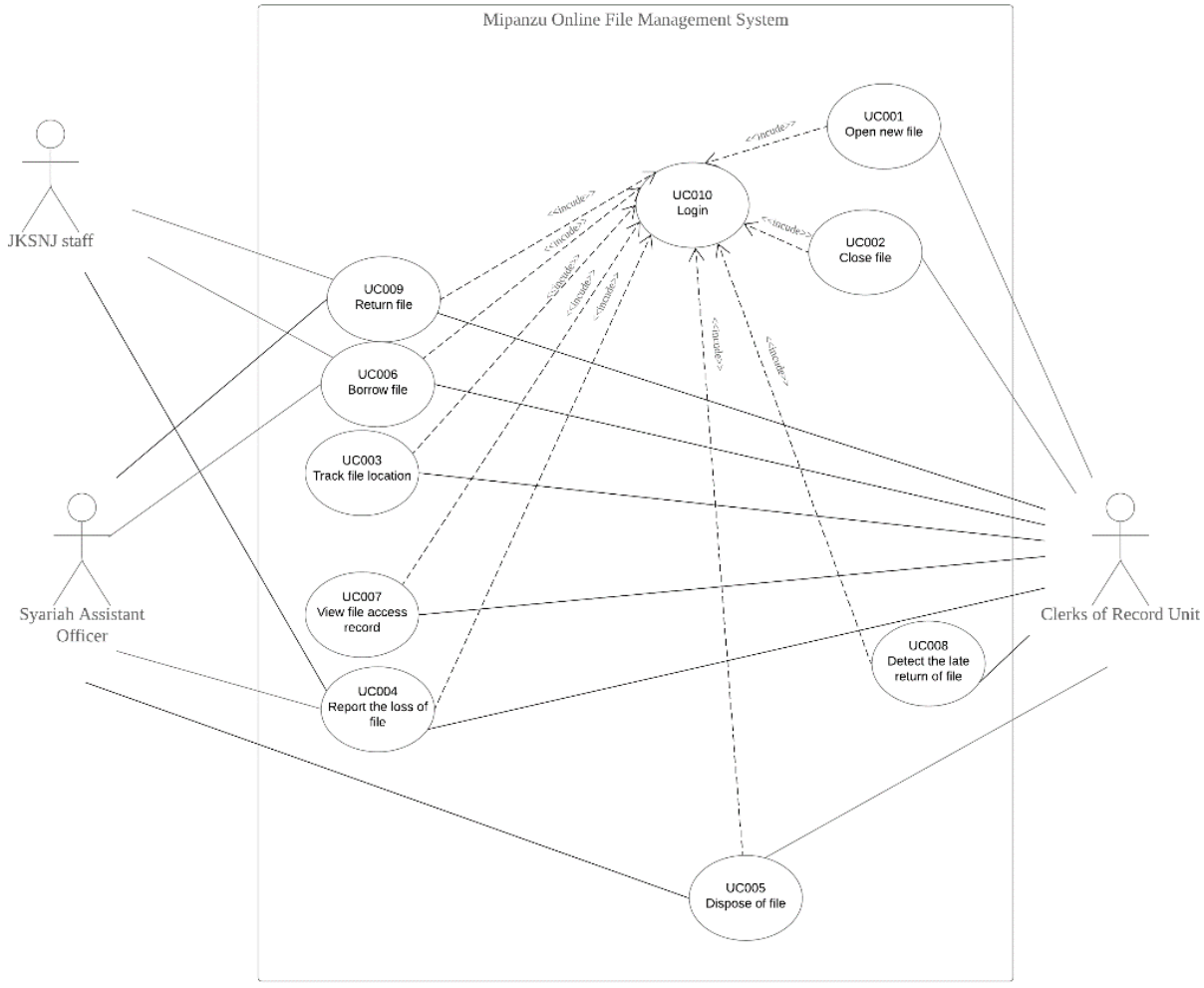

Figure 2.1: Use Case Diagram for <Mipanzu Online File Management System>

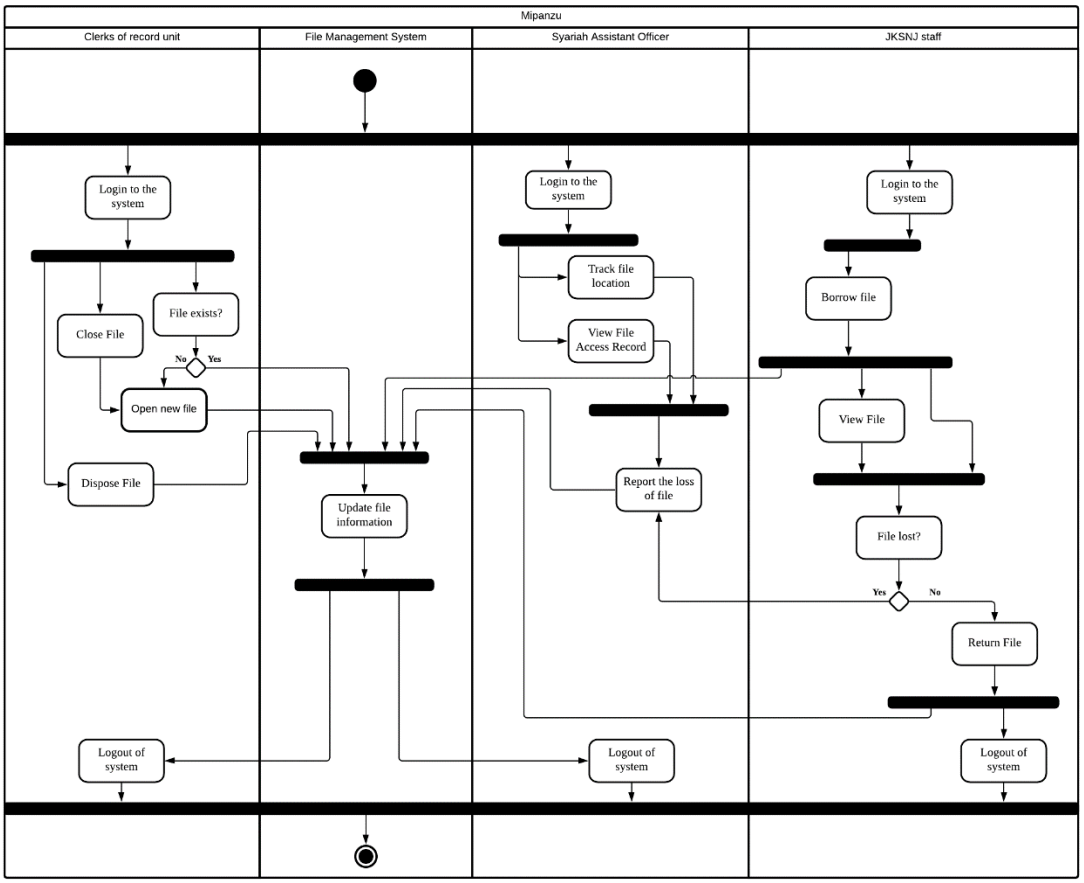

Figure 2.2: Activity Diagram for <Mipanzu Online File Management System>

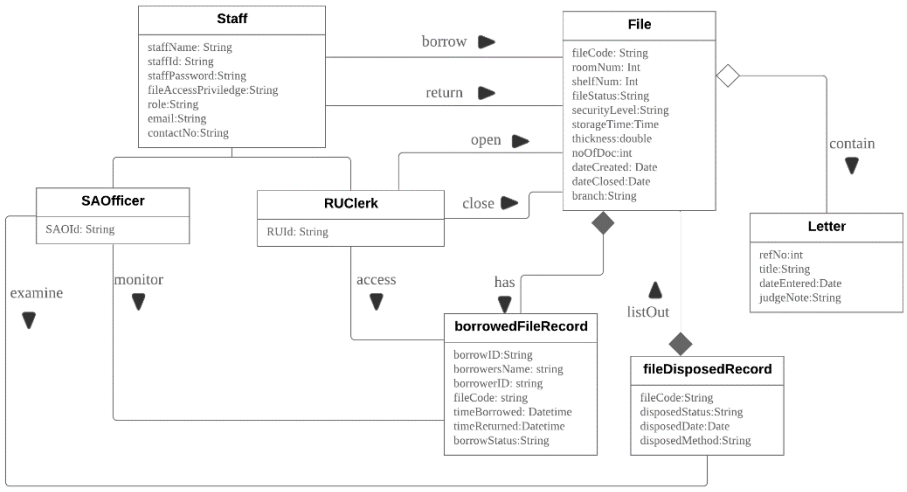

Figure 2.3: Domain Model for <Mipanzu Online File Management System>

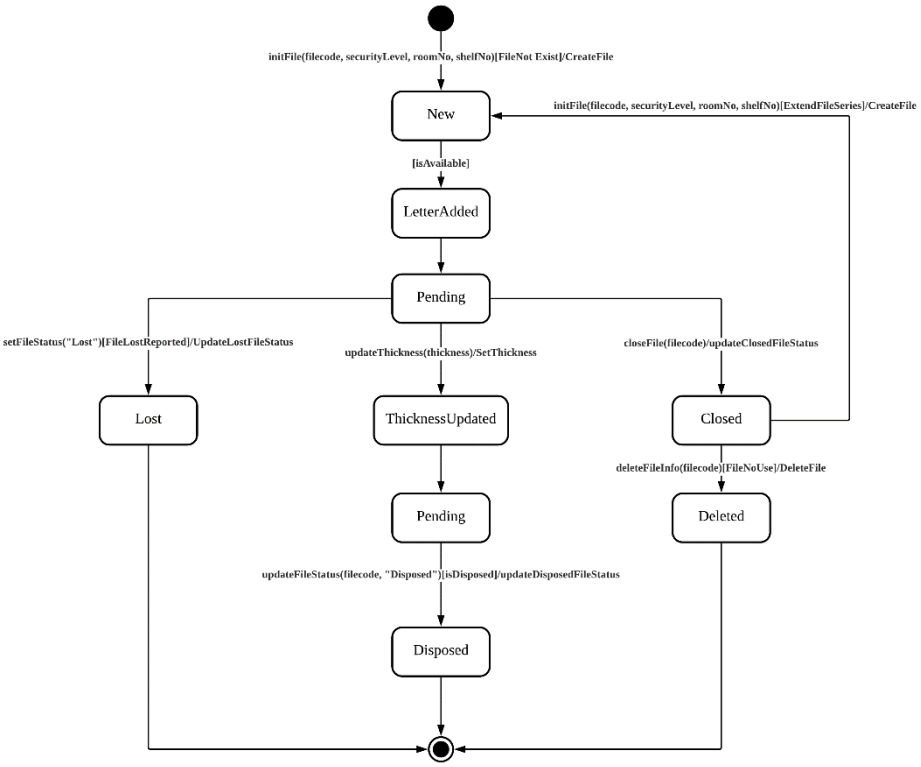

Figure 2.4: State Machine Diagram for <File Class>

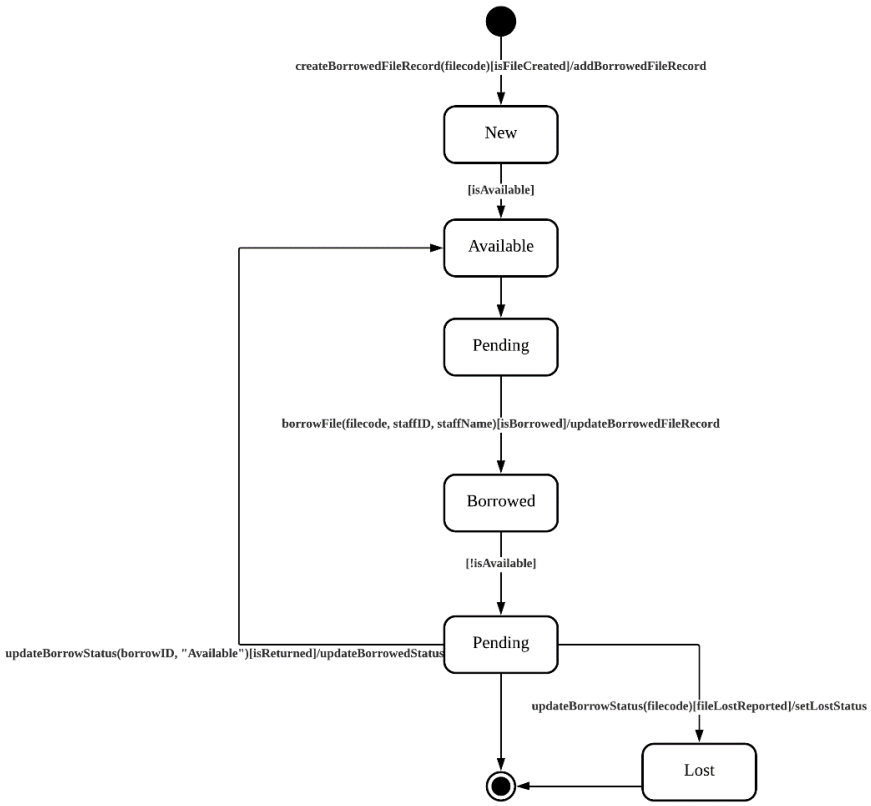

Figure 2.5: State Machine Diagram for <BorrowedFileRecord Class>

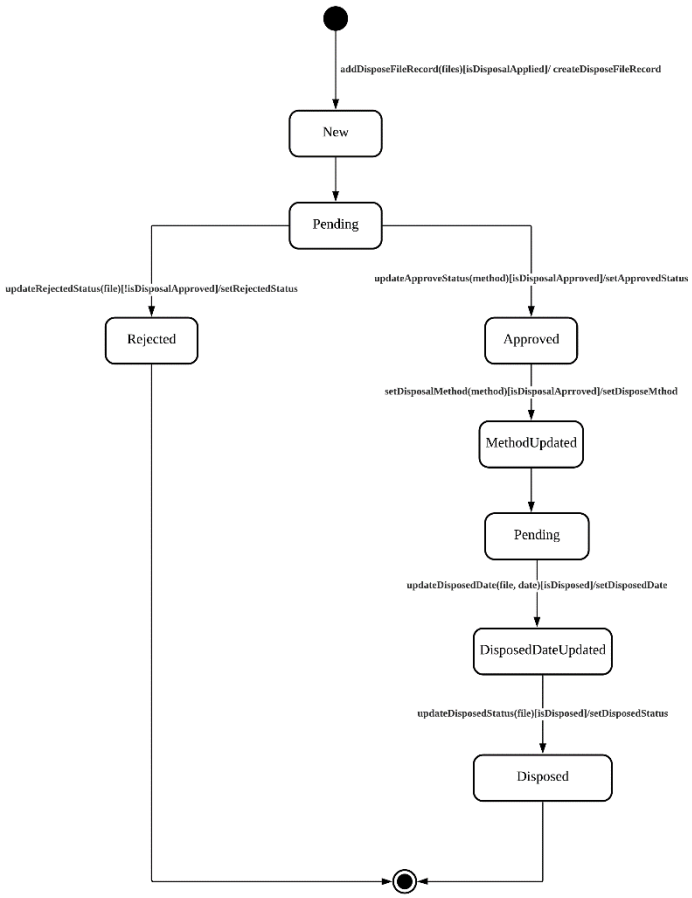

Figure 2.6: State Machine Diagram for <FileDisposedRecord Class>

2.2.1. UC001: Use Case <Open new file>

Table 2.1: Use Case Description for <Open new file>

Use case: <Open new file>
ID: UC001
Actors: Clerks of the record unit
Preconditions: 1. A valid clerk of the record unit is logged on to the system. 2. **UC010** is invoked.
Flow of events: 1. The clerk of the record unit keys in the file code of the file. 2. The system automatically checks whether the file with the file code entered exists. 3. If the file exists already, **Exception E1** is performed. 4. The system displays message File does not exist. 5. The clerk of the record unit keys in the file code, security level of file, room number and shelf number of file storage space. 6. The system registers the new file with the file info entered by the clerk of the record unit. 7. The clerk of the record unit keys in the reference number, title and judge note of a letter. 8. The system registers the letter in the letter database. 9. The system adds the letter info into the file created in the file database. 10. The system updates the file database.
E1: File exists already 1. The system displays the file info. 2. Clerk of record unit keys in the reference number of the letter. 3. The system checks whether the letter with the reference number entered exists. 4. If the letter does not exist resume step 7 5. The system display letter is already exists. 6. The use case ends.
Postconditions: 1. A new file is successfully created by the Clerks of the record unit.

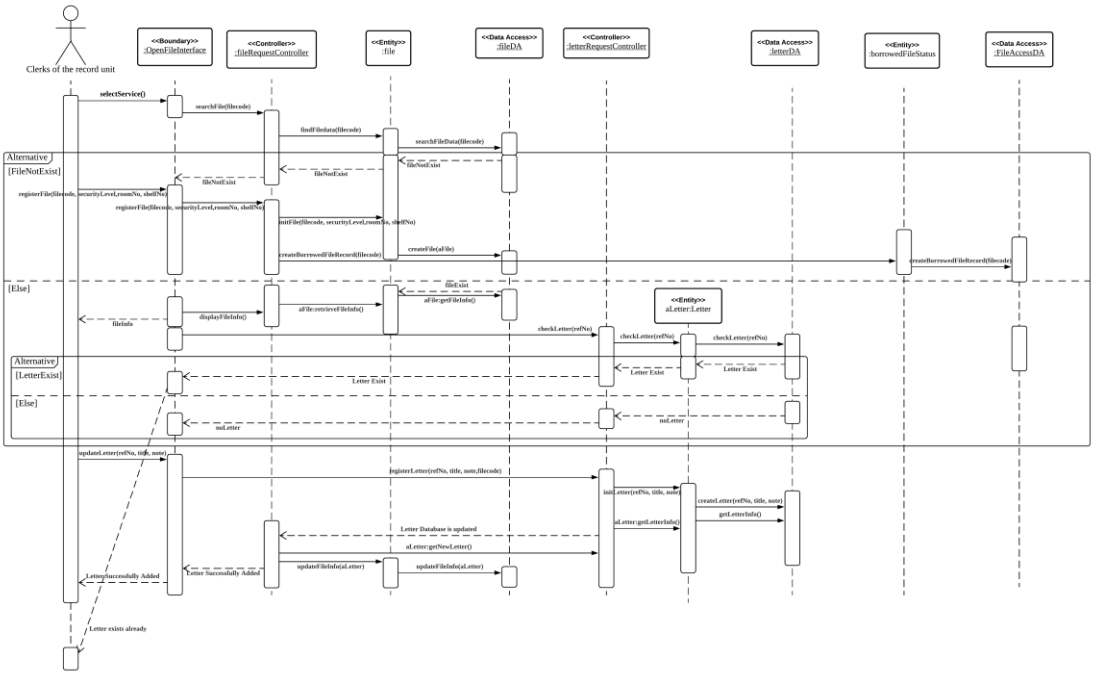

Figure 2.5: Sequence Diagram for <Open new file>

2.2.2. UC002: Use Case <Close file>

Table 2.2: Use Case Description for <Close File>

Use case: <Close File>
ID: UC002
Actors: Clerk of the record unit
Preconditions: 1. A valid clerk of the record unit is logged on to the system. 2. **UC010** is invoked.
Flow of events: 1. The clerk of record unit keys in the file code of the file. 2. The system automatically checks whether the file with the file code entered exists. 3. If the file does not exist, **Exception E1** is performed. 4. The system displays file exists message. 5. The clerk of the record unit clicks on the close file button. 6. The system updates the file status to “Closed” in the file database. 7. The system automatically registers a new file with the same file info as the original closed file in the file database. 8. The system sets the subsequent file code to the new file. 9. The system displays the new file code. 10. The clerk of the record unit clicks on delete file to delete the original file info. **[A1]** 11. The system deletes the original file info from the file database.
Postconditions: 1. File is successfully closed by the clerk of the record unit
Alternative flow 1: **[A1: The clerk of the record unit doesn’t click on delete file button]** 1. The use case ends.
Postconditions: 1. File is successfully closed by the clerk of the record unit
E1: File does not exist 1. The system displays an error message. 2. The use case ends.

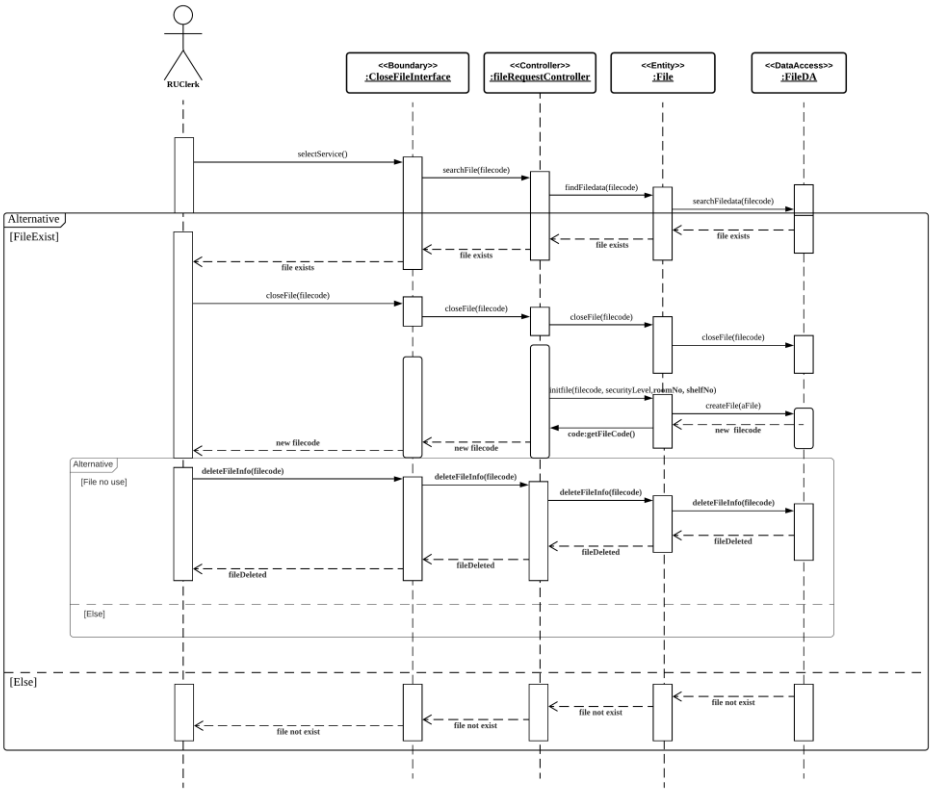

Figure 2.6: Sequence Diagram for <Close File>

2.2.3. UC003: Use Case <Track file location>

Table 2.3: Use Case Description for <Track file location>

Use case: <Track file location>
ID: UC003
Actors: Clerks of the record unit
Preconditions: 1. A valid clerk of the record unit is logged on to the system. 2. **UC010** is invoked.
Flow of events: 1. The clerk of record unit keys in the file code of the file. 2. The system automatically checks whether the file with the file code entered exists. 3. If the file does not exist, **Exception E1** is performed. 4. The system displays file exists message. 5. The clerk of the record unit clicks on the search file button. 6. The system displays the location of the file (room number and shelf number).
E1: File does not exist 1. The system displays an error message. 2. The use case ends.
Postconditions: 1. The file location is successfully tracked.

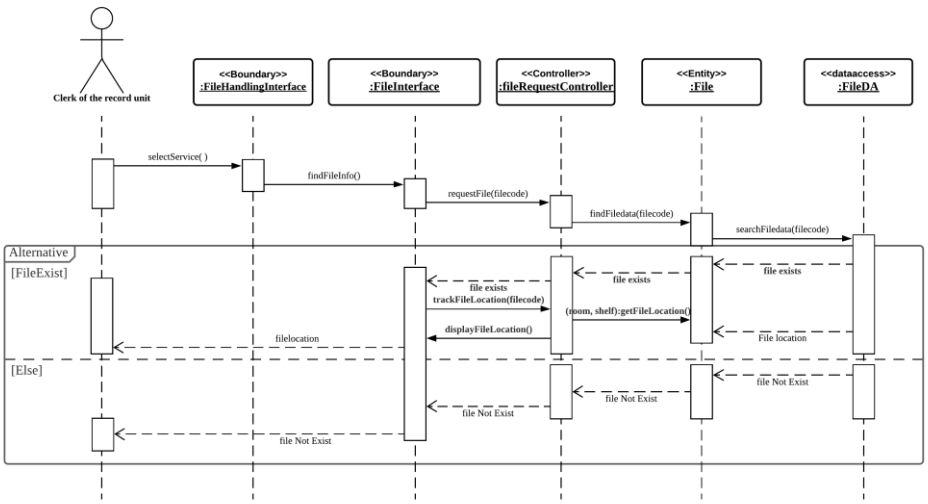

Figure 2.7: Sequence Diagram for <Track file location>

2.2.4. UC004: Use Case <Report the loss of file>

Table 2.4: Use Case Description for <Report the loss of file>

Use case: <Report the loss of file>
ID: UC004
Actors: Clerk of record unit Syariah Assistant Officer JKSNJ staff
Preconditions: 1. A valid JKSNJ staff is logged on to the system. 2. **UC010** is invoked.
Flow of events: 1. The JKSNJ staff keys in the file code. 2. The system automatically checks whether the borrow record with the staff's ID and the file code entered exists. 3. If the borrow record does not exist, **Exception E1** is performed. 4. The system displays borrow record found message. 5. The JKSNJ staff clicks on the report file lost button. 6. The system updates the borrow status as lost in the file access database. 7. The system updates the file status as lost in the file database. 8. The system sends notification to Syariah Assistant Officer and the clerk of Record Unit about the file lost with the file code and the info of staff who lost the file. 9. The system notifies all clerks of the record unit of that branch regarding the identity of the staff who lost file and the file code of the lost file.
E1: Borrow Record does not exist 1. The system displays an error message. 2. The use case ends.
Postconditions: The loss of file is successfully reported.

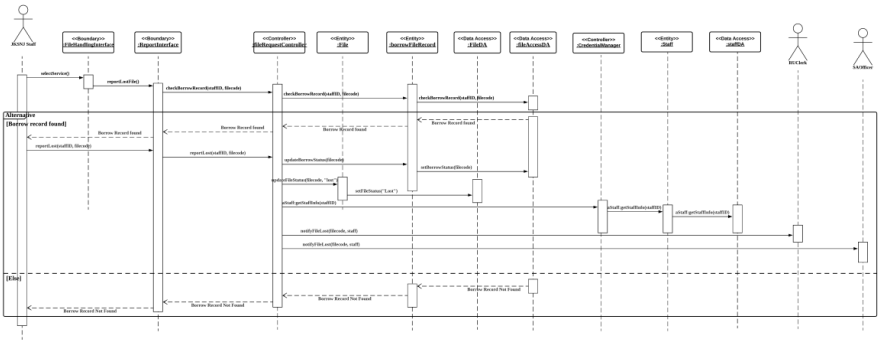

Figure 2.8: Sequence Diagram for <Report the loss of file>

2.2.5. UC005: Use Case <Dispose of file>

Table 2.5: Use Case Description for <Dispose of file>

Use case: <Dispose of file>
ID: UC005
Actors: 1. Clerks of the record unit 2. Syariah Assistant Officer 3. Arkib Negara Malaysia
Preconditions: 1. **UC010** is invoked.
Flow of events: 1. The clerk of record unit clicks on “Outdated Files” to display and load the files that exceed the maximum storage time. 2. The clerk of the record unit enters the file codes of other damaged files to be disposed of and clicks on Search. 3. The system checks whether the file code exists in the system. 4. If the file code does not exist, **A1** is performed. 5. The system displays a “File code is found” message. 6. The clerk of record unit clicks the add button to add file to the list of files to be disposed of. 7. The clerk of the record unit clicks on Submit to send the file dispose list through the system to Syariah Assistant Officer. 8. Syariah Assistant Officer updates the thickness of files from the file dispose list on the dispose file page. 9. Syariah Assistant Officer clicks on “Email” which links to the Outlook mailbox to apply for the file disposal with Arkib Negara Malaysia. 10. The system checks whether the thickness of all files is entered. 11. If the thickness of files is not fully filled in, **A2** is performed. 12. Syariah Assistant Officer clicks on Add to Disposal Record to insert the files from file disposal list to disposal record in the system. 13. Syariah Assistant Officer receives email of file disposal list with approval status and disposal method from Arkib Negara Malaysia. 14. If the application of file disposal is rejected, **Exception E1** is performed. 15. Syariah Assistant Officer updates the file dispose status as “Approved”. 16. Syariah Assistant Officer updates the file disposal method. 17. The clerk of the record unit views the list of files with approved file disposed status. 18. The clerk of the record unit updates the file disposal date. 19. The clerk of record unit updates the file dispose status as “Disposed”.
A1: The file code entered is not found 1. The system displays an error message 2. Step 7 resumes. **A2: The thickness of all files is not fully filled in** 1. The system displays an error message. 2. Step 8 resumes
E1: The application of file disposal is rejected **1.** Syariah Assistant Officer updates the file dispose status as “Rejected”. **2.** The use case ends.
Postconditions: 1. File is successfully disposed of.

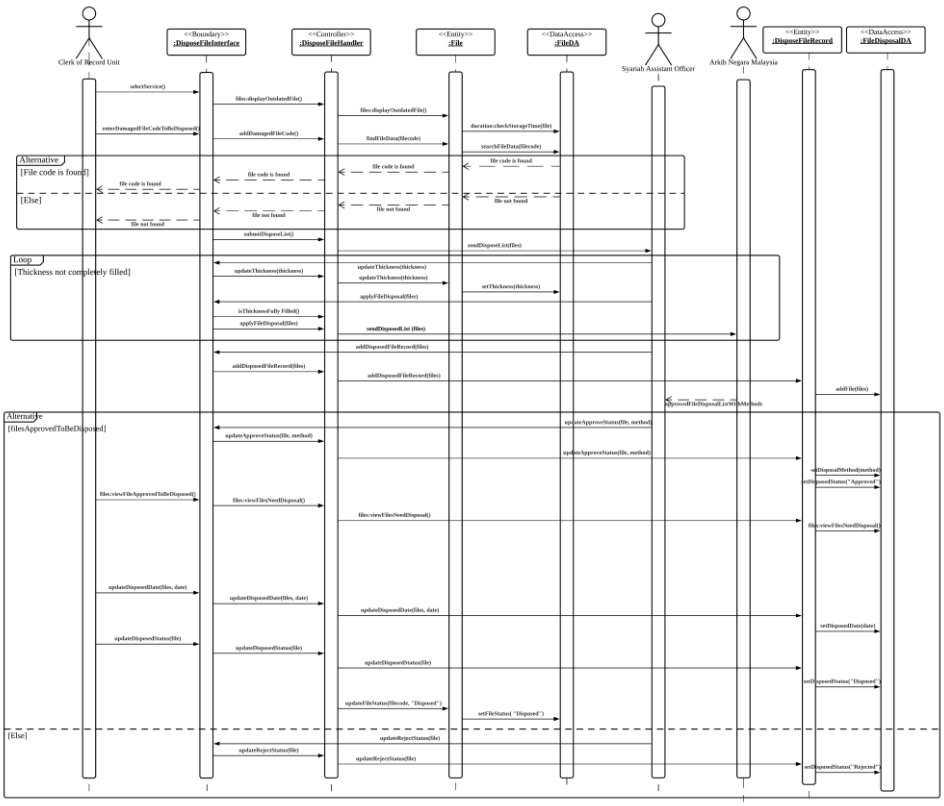

Figure 2.9: Sequence Diagram for <Dispose of file>

2.2.6. UC006: Use Case <Borrow file>

Table 2.6: Use Case Description for <Borrow file>

Use case: Borrow file
ID: UC006
Actors: JKSNJ staff Clerk of the record unit
Preconditions: 1. A valid JKSNJ staff is logged on to the system. 2. **UC010** is invoked.
Flow of events: 1. The JKSNJ staff keys in the file code. 2. The system automatically checks whether the file with the file code entered exists. 3. If the file does not exist, **Exception E1** is performed. 4. The system automatically checks whether the file with the file code entered is still available (not borrowed). 5. If the file is not available, **Exception E2** is performed. 6. The system automatically checks whether the staff has the privilege to borrow the file. 7. If the staff does not have the privilege to borrow the file, **Exception E3** is performed. 8. The system creates a borrow ID. 9. The system updates the file access database. 10. The system shows the borrow ID to the staff on the borrow file interface. 11. The clerk of the record unit keys in the borrow ID. 12. The system displays the borrow record.
E1: File does not exist 1. The system displays File not found message. 2. The use case ends. **E2: File is not available** 1. The system displays the last borrower's name and staff ID. 2. The system displays File is not available message. 3. The use case ends. **E3: The staff does not have the privilege to borrow the file** 1. The system displays message No right to borrow the file. 2. The use case ends.
Postcondition: File is successfully borrowed by the JKSNJ staff.

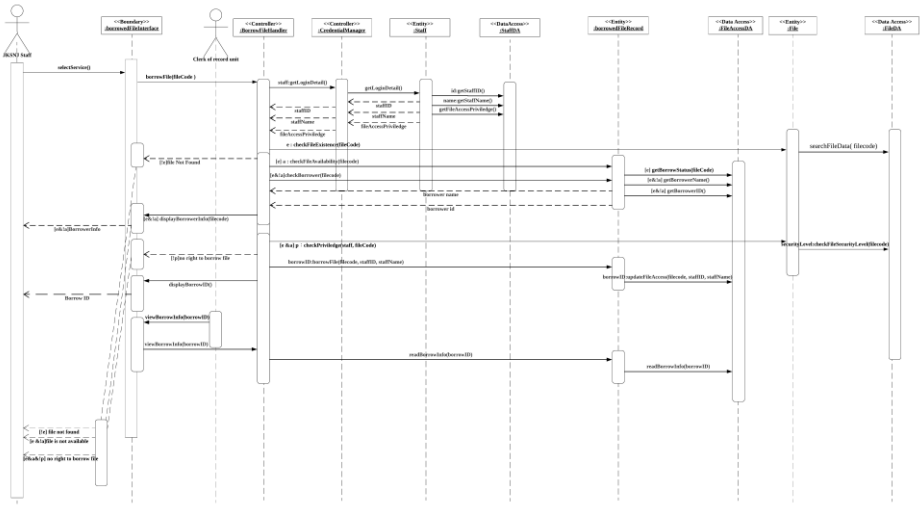

Figure 2.10: Sequence Diagram for <Borrow file>

2.2.7. UC007: Use Case <View file access record>

Table 2.7: Use Case Description for <View file access record>

Use case: <View file access record>
ID: UC007
Actors: Clerk of the record unit
Preconditions: 1. A valid clerk of the record unit is logged on to the system. 2. **UC010** is invoked.
Flow of events: 1. The clerk of record unit keys in the file code of the file. 2. The system automatically checks whether the file with the file code entered exists. 3. If the file does not exist, **Exception E1** is performed. 4. The system displays the file access record.
E1: File does not exist 1. The system displays an error message. 2. The use case ends.
Postconditions: 1. File access record is successfully viewed by Clerk of the record unit

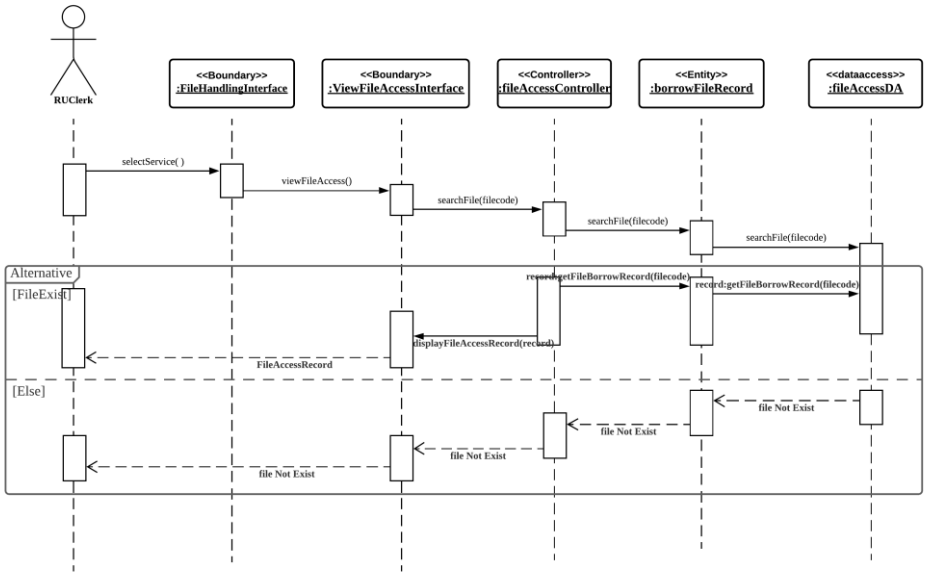

Figure 2.11: Sequence Diagram for <View file access record>

2.2.8. UC008: Use Case <Detect the late return of file>

Table 2.8: Use Case Description for <Detect the late return of file>

Use case: <Detect the late return of file>
ID: UC008
Actors: JKSNJ staff Clerk of the record unit
Preconditions: 1. A clerk of the record unit is logged on to the system. 2. **UC010** is invoked.
Flow of events: 1. The clerk of the record unit clicks on the View File Late Return button. 2. The system displays the list of files that exceeds the return deadline with their file access record. 3. The system notifies the last borrower of the files in the late return file list.
Postconditions: 1. The list of files that exceed the deadline of return is successfully viewed. 2. The return file notification is successfully sent to the last borrowers.

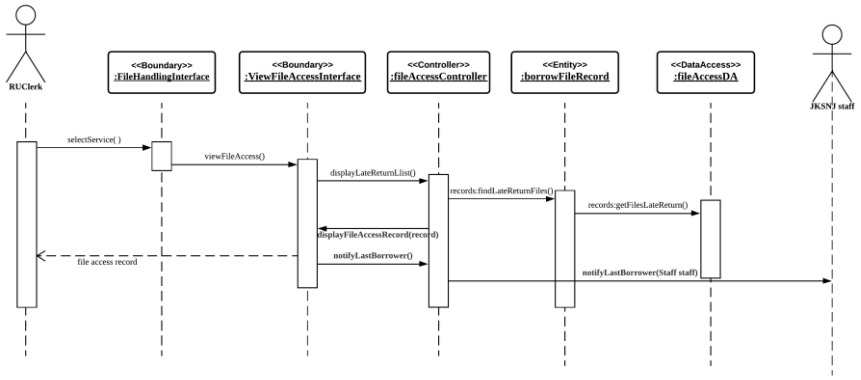

Figure 2.8: Sequence Diagram for <Detect the late return of file>

2.2.9. UC009: Use Case <Return file>

Table 2.9: Use Case Description for <Return file>

Use case: <Return file>
ID: UC009
Actors: JKSNJ staff Clerk of the record unit
Preconditions: 1. A JKSNJ staff is taking the file that they borrowed to the counter of JKSNJ. 2. A valid clerk of the record unit is logged on to the system. 3. **UC010** is invoked.
Flow of events: 1. The clerk of the record unit keys in the borrow ID. 2. The system automatically checks whether the borrow record with the borrow ID entered exists. 3. If the borrow record does not exist, **Exception E1** is performed. 4. The system displays the file borrow info. 5. The clerk of the record unit clicks on the Return File button. 6. The system updates the borrow status in the file access database.
E1: Borrow Record does not exist 1. The system displays an error message. 2. The use case ends.
Postconditions: 1. File is successfully returned from the JKSNJ staff

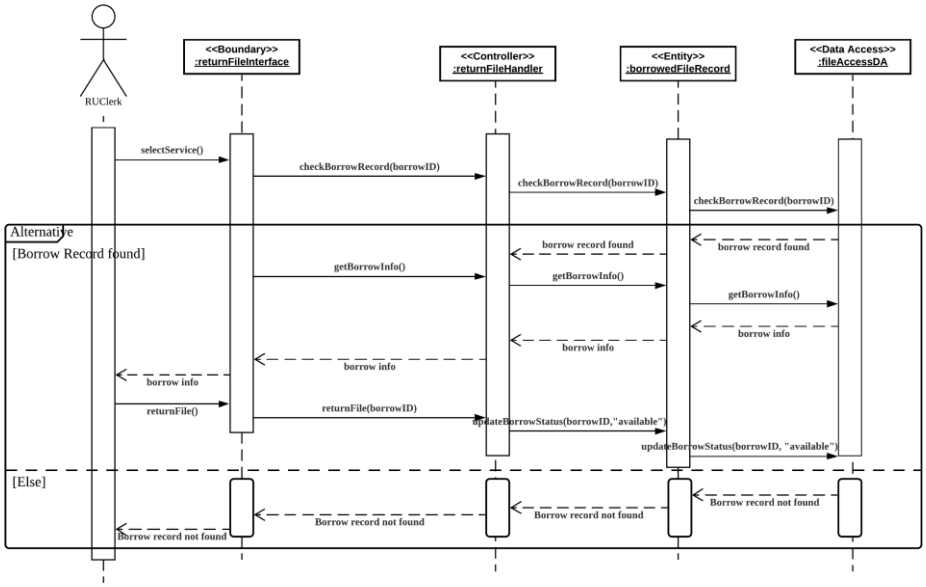

Figure 2.9: Sequence Diagram for <Return File>

2.2.10. UC010: Use Case <Login>

Table 2.10: Use Case Description for <Login>

Use case: <Login>
ID: UC010
Actors: JKSNJ staff
Preconditions: JKSNJ staff enters the website of Mipanzu System.
Flow of events: 1. JKSNJ staff enters the staff id and password. 2. The system checks whether the staff ID and password matches a staff record in the staff database. 3. If the staff is not found, **Exception 1** is performed. 4. The system checks the role of staff. 5. The system displays the home interface of general staff. **[A1] [A2]**
Postconditions: 1. The JKSNJ staff is successfully logged in.
Alternative flow 1: **[A1: The role of staff is Syariah Assistant Officer]** 1. The system displays the home interface of Syariah Assistant Officer.
Postconditions: 1. The JKSNJ staff is successfully logged in.
Alternative flow 2: **[A2: The role of staff is clerk of record unit]** 1. The system displays the home interface of the clerk of the record unit.
Postconditions: 1. The JKSNJ staff is successfully logged in.
E1: The system cannot find the staff info with the staff id and password entered 1. The system displays message Staff Not Found. 2. The use case ends.

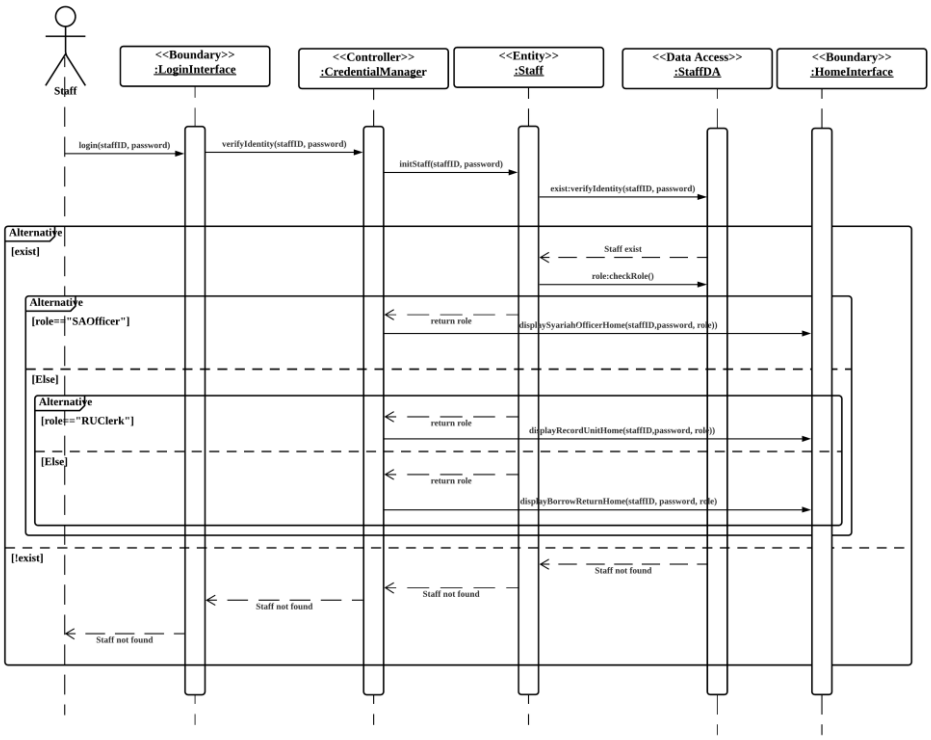

Figure 2.10: Sequence Diagram for <Login>

2.3. Performance and Other Requirements

- Average response time per interface must be less than 3 seconds.
- The system shall be able to support at least 100 concurrent users.
- Any changes made in the database should be updated instantly, this is for other users to not have a conflicted action being done in the system.
- The system shall be able to run on any desktop or mobile devices that has internet connection.
- The system shall be able to verify the login information of the user in between 5 seconds.
- The system shall be able to recover loss data in case of a sudden power outage or flood.
- The system shall be able to connect to the internet.
- The system shall be able to maintain all the data despite having a software update.
- The system shall be able to operate 24/7.

2.4. Design Constraints

1. The file and record management system shall be based on Malaysian Standards for Record Management.
2. Any page displayed shall not contain any copyright law.
3. The system shall not display any content of the file.
4. The file is classified based on the code classification and security level of the file.
5. The file can only be accessed by those who have authorities.
6. The system shall only store the file records that are in JKSNJ.

2.5. Software System Attributes

a) The system interfaces shall be easy to use by the user in any age range.
b) The system should prompt an output that can help the user to make transaction without making any input error.
c) The system shall provide an option to choose a language between Bahasa Malaysia and English.
d) The system shall use a universal language that can be understand by the user.
e) The user only needs less than 5 hours of training and surfing through the system to be able to fully understand the system.

3. System Architectural Design

3.1. Architecture Style and Rationale

The layered architectural style that we choose is **layered** architecture. There are several reasons why our group decided to choose this architectural style. This architecture helps us to organize our system into a set of layers such that each layer will provide their own function and services. Other than that, it is also easier for us to see the development of our system as each sub-system will support the incremental development of each layer. Layered architecture also made it easier for our system to use multi-platform implementation such as system application and website. Lastly, if there are some improvements and changes that can be made in the system, layered architecture also allows the replacement of each layer as long as the interface of the system is maintained. As our system needs to maintain a large number of operations, processes and databases, this layered architecture is really helpful as we can see all the things that need to be included in our system in each layer and how they connect with each other. It is also easier for our system to make any changes without having to change the interface in the system.

3.2. Component Model

There are 2 subsystems in Mipanzu Online File Management System which are login subsystem and file subsystem. Login system is responsible for verifying the identity of users during login and manage staff database where it allows modification and access to the staff information. On the other hand, File subsystem handles the operation of all the main functions related to file record provided by the system including borrrow file, return file, open file, close file, dispose of file, report file lost, check late return of file, track file location and view file access. File subsystem manages the entry, update, deletion, and access to the file record, dispose file record and borrow file record. Both subsystems interact when File subsystem requires staff information to perform its functions. Most interactions occur among components within the same subsystem which gives rise to a system with high cohesion and low coupling.

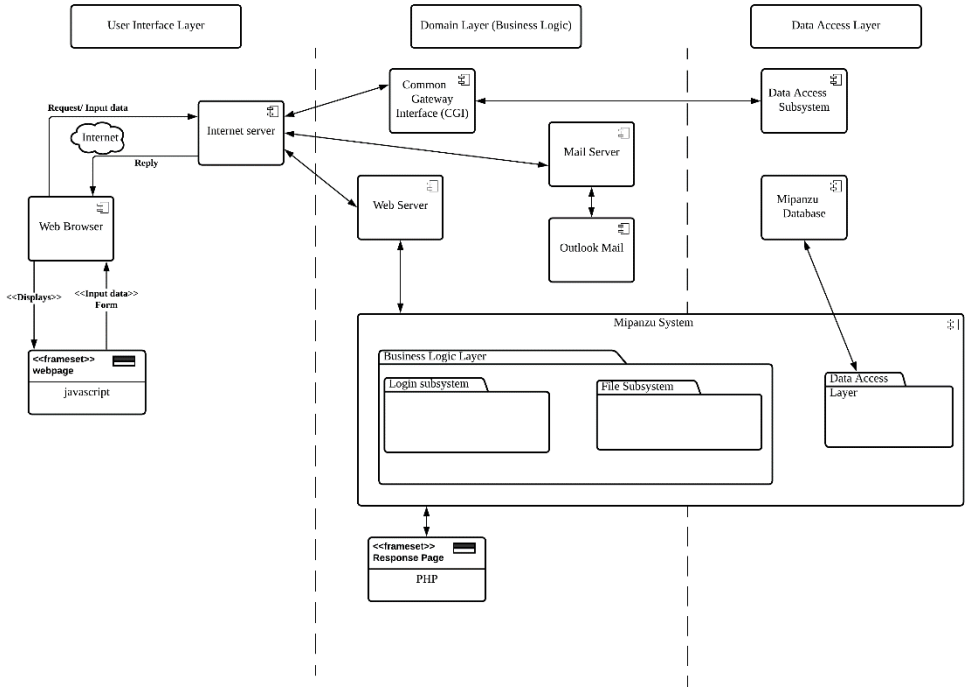

Figure 3.1: Architecture Diagram for <Mipanzu Online File Management System>

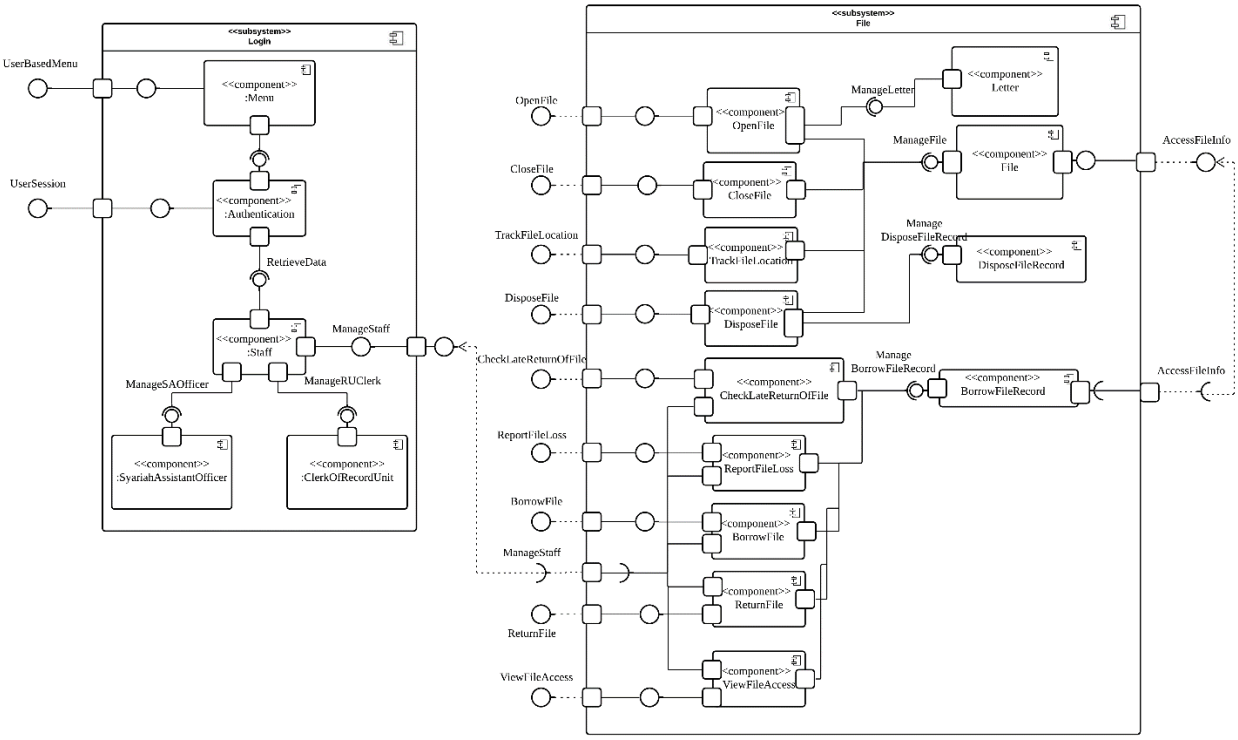

Figure 3.2: Component Diagram of <Mipanzu Online File Management System>

4. Detailed Description of Components

4.1. Complete Package Diagram

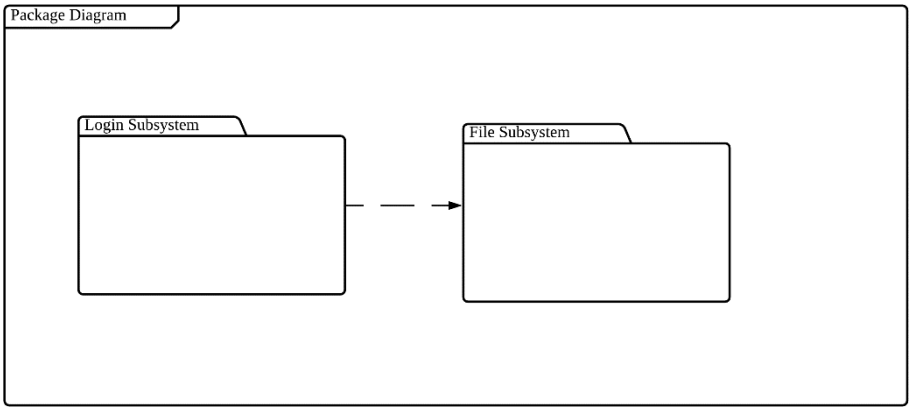

Figure 4.1: Package Diagram for <Name of the System>

4.2. Detailed Description

4.2.1. P001: Login Subsystem

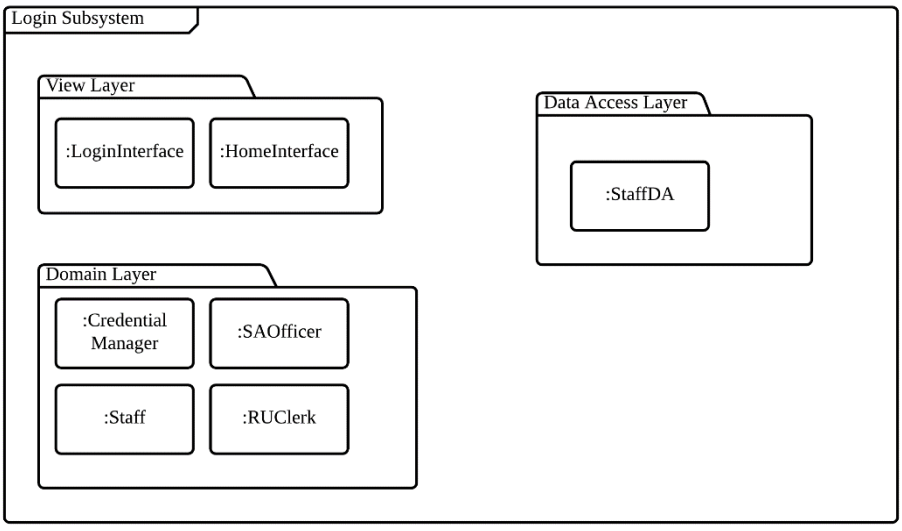

Figure 4.2: Package Diagram for <Login> Subsystem

4.2.1.1. Class Diagram

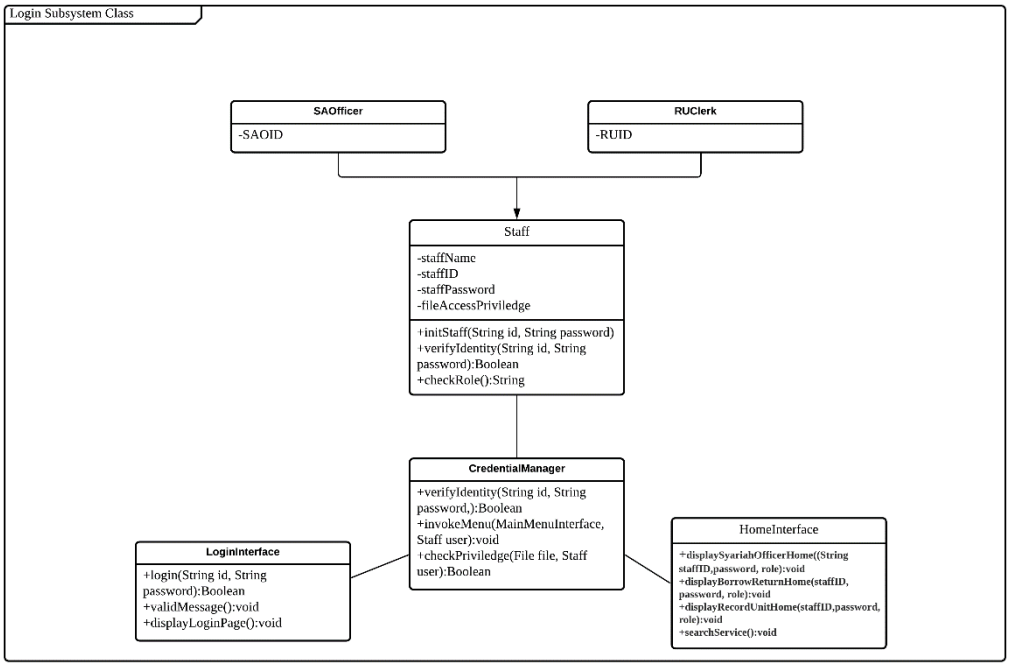

Figure 4.3: Class Diagram for <Login> Subsystem

Entity Name	Staff
Method Name	verifyIdentity
Input	Staff ID, Password
Output	boolean
Algorithm	(a) Start (b) Validate the staff information (c) If staff ID or password is wrong then return false. (d) If correct then return true. (e) End

4.2.1.2. Sequence Diagram

a) SD001: Sequence diagram for Login

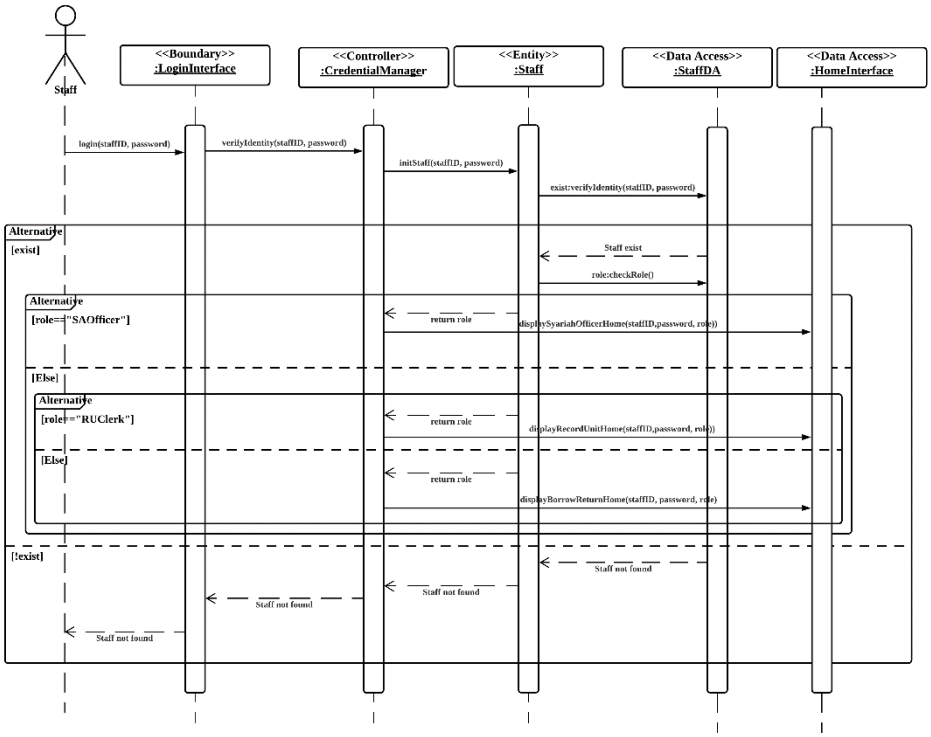

Figure 4.4: Sequence Diagram for <Login>

4.2.2. P002: File Subsystem

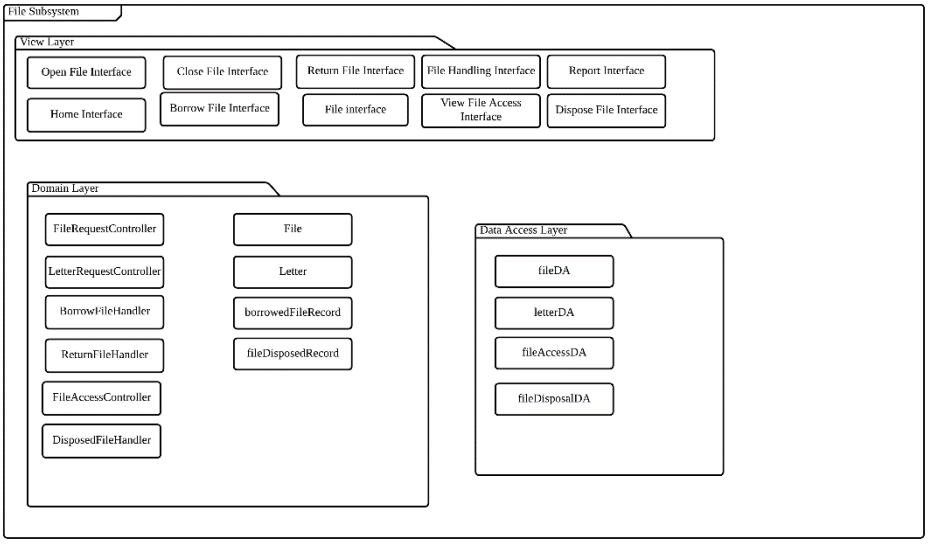

Figure 4.5: Package Diagram for <File> Subsystem

4.2.2.1. Class Diagram

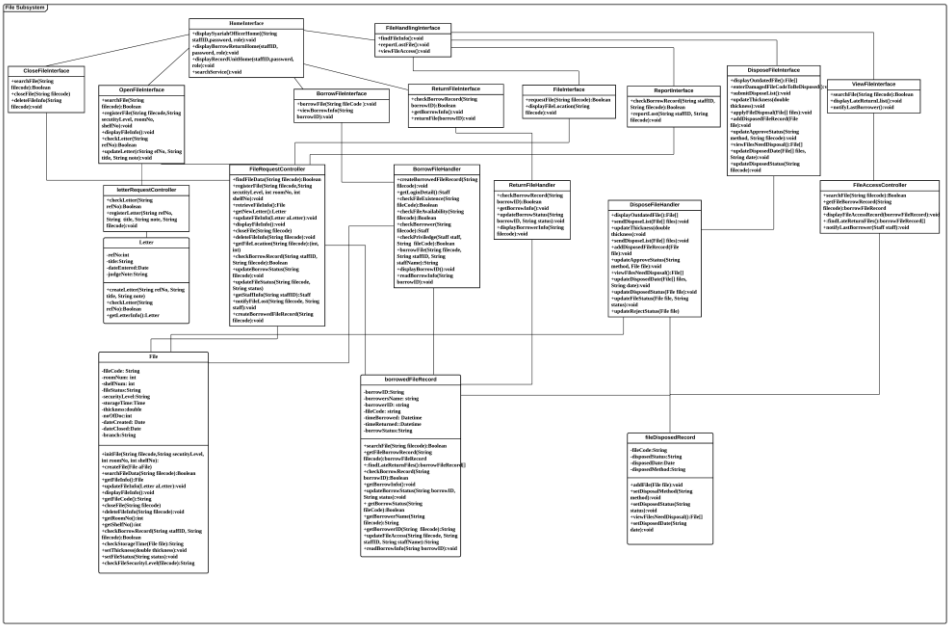

Figure 4.3: Class Diagram for <File> Subsystem

Entity Name	OpenFileInterface
Method Name	registerFile
Input	Filecode, security level
Output	-
Algorithm	1. Start 2. Get the file code from clerks of the record unit 3. Get the security level of the file 4. End

Entity Name	borrowedFileInterface
Method Name	borrowFile
Input	fileCode
Output	BorrowID, BorrowerInfo
Algorithm	1. Start 2. Get the file code from JKSNJ Staff 3. Send the file code to the Borrow File Handler 4. End

Entity Name	DisposeFileHandler
Method Name	updateApproveStatus
Input	Method, file
Output	-
Algorithm	1. Start 2. Use the file instance to call the setDisposalMethod with method passed as argument. 3. Use the file instance to call the setDisposedStatus and pass “Approved” as argument. 4. End

4.2.2.2. Sequence Diagram

a) **SD002**: Sequence diagram for Open new File

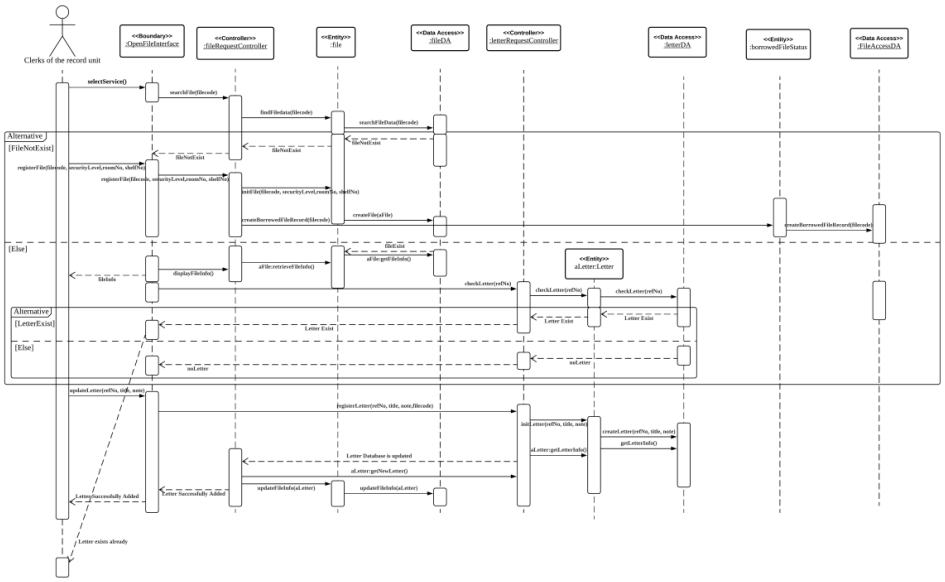

Figure 4.7: Sequence Diagram for <Open new File>

b) **SD003**: Sequence diagram for Close File

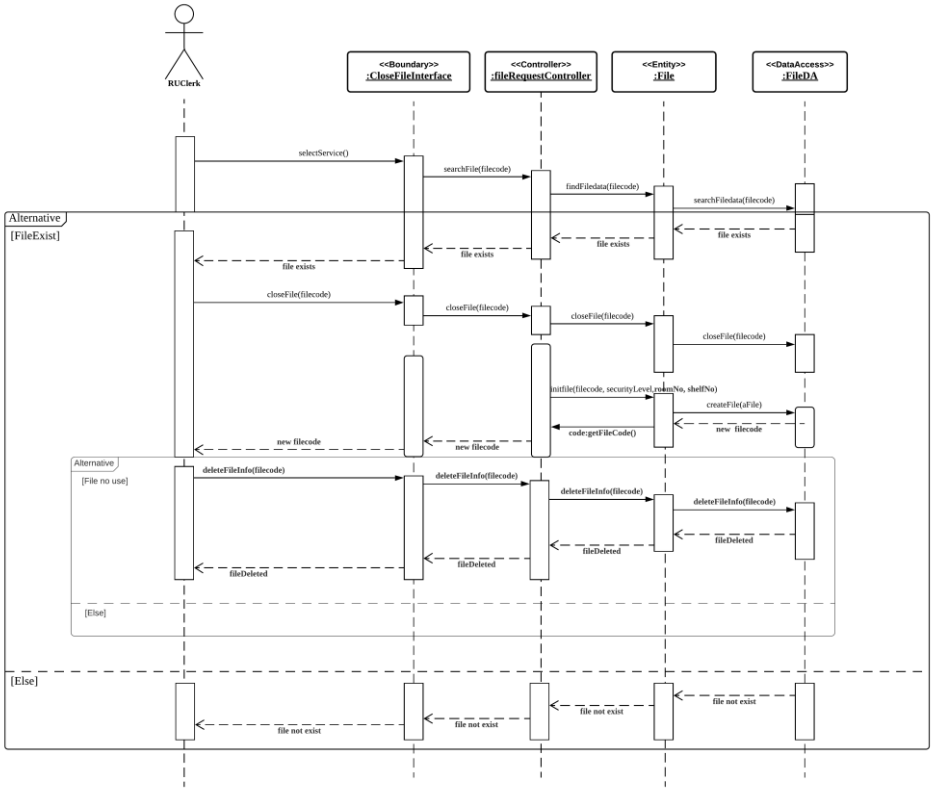

Figure 4.8: Sequence Diagram for <Close File>

c) **SD004**: Sequence diagram for Track File Location

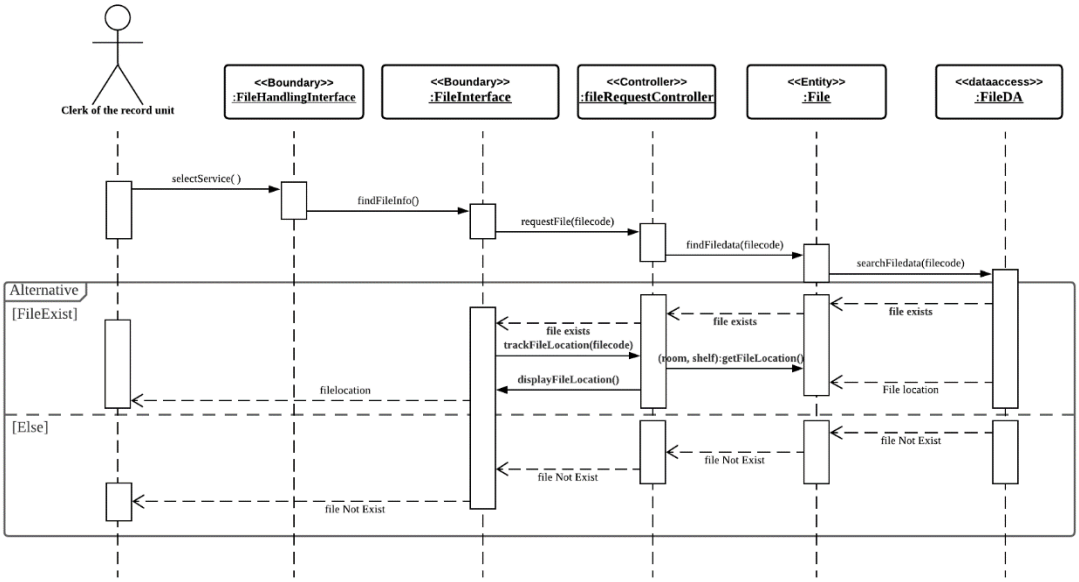

Figure 4.9: Sequence Diagram for <Track File Location>

d) **SD005**: Sequence diagram for Report the Loss of File

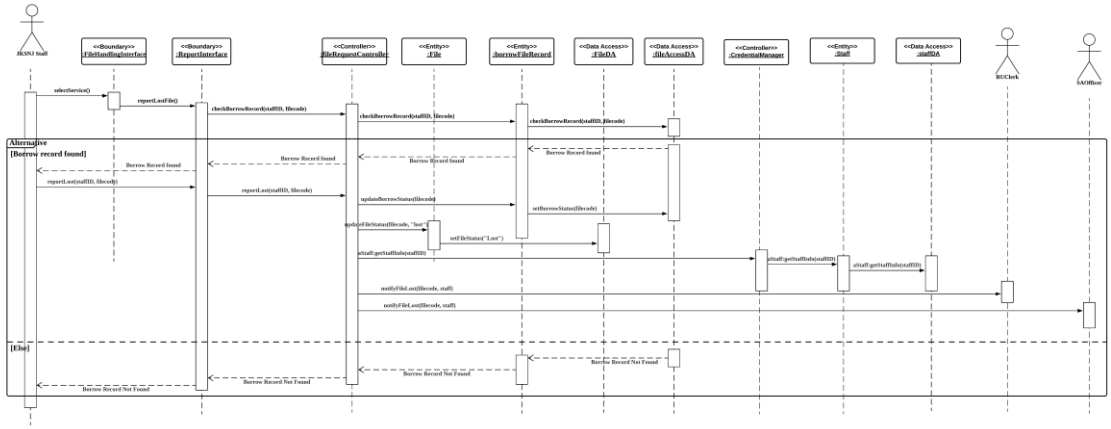

Figure 4.10: Sequence Diagram for <Report the Loss of File>

e) **SD006**: Sequence diagram for Dispose of File

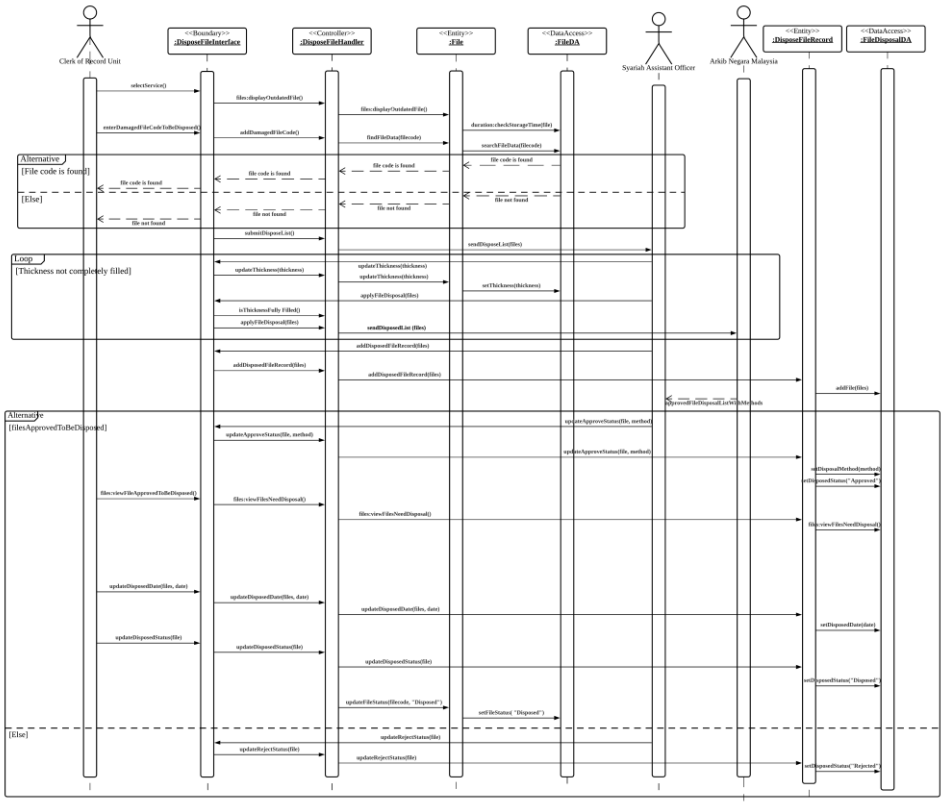

Figure 4.11: Sequence Diagram for <Dispose of File>

f) **SD007**: Sequence diagram for Borrow File

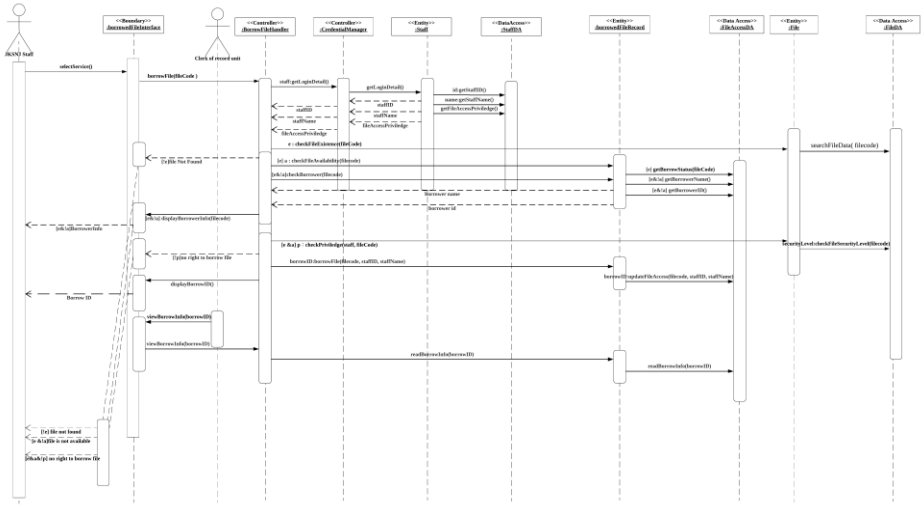

Figure 4.12: Sequence Diagram for <Borrow File>

g) **SD008**: Sequence diagram for View File access Record

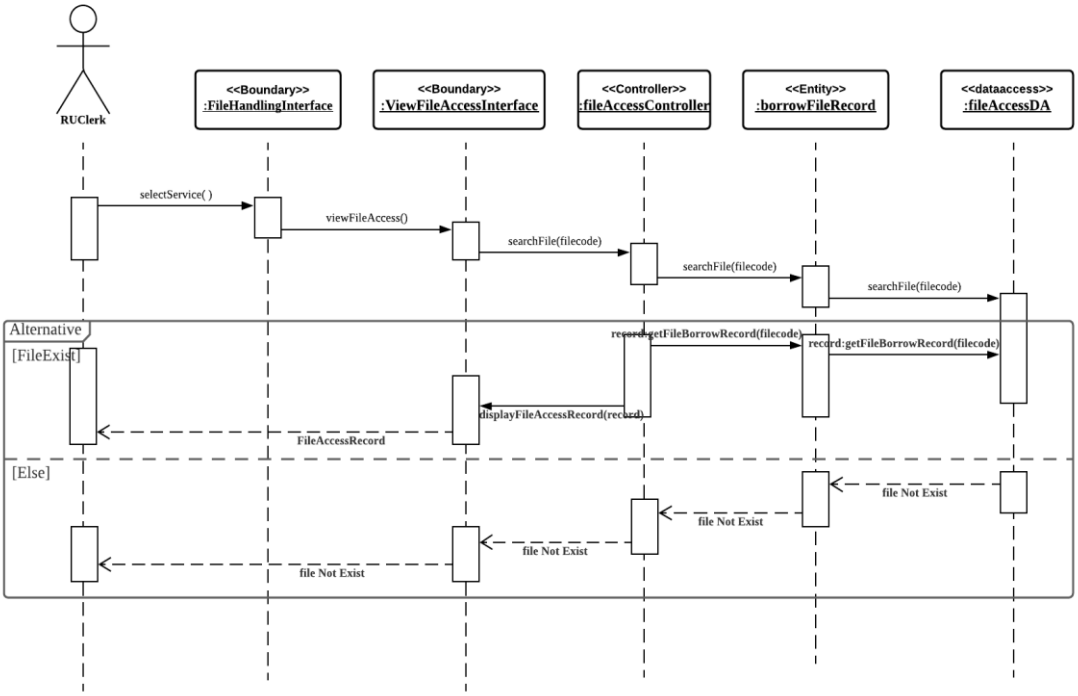

Figure 4.13: Sequence Diagram for <View File access Record>

h) **SD009**: Sequence diagram for Detect the late return of File

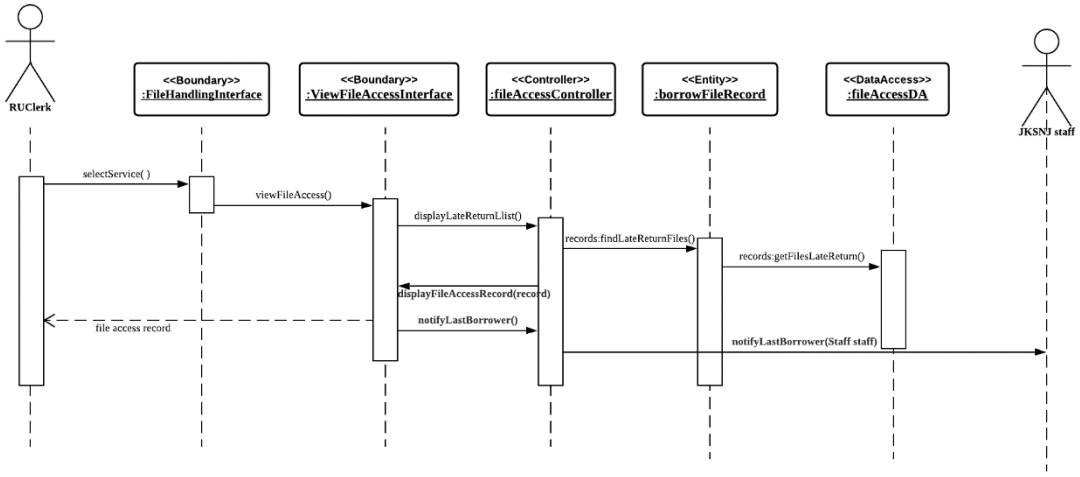

Figure 4.14: Sequence Diagram for <Detect the late return of File>

i) **SD010**: Sequence diagram for Return File

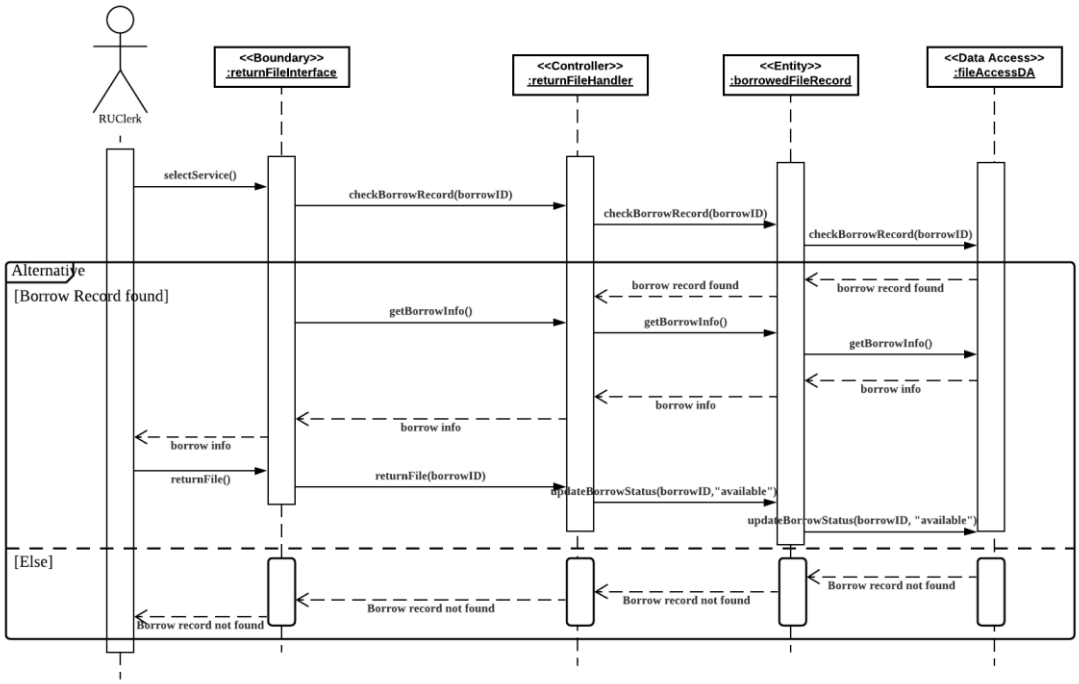

Figure 4.15: Sequence Diagram for <Return File>

5. Data Design

5.1. Data Description

The major data or systems entities are stored into a relational database named as Mipanzu Online File Management System processed and organized into *7* entities as listed in Table 5.1.

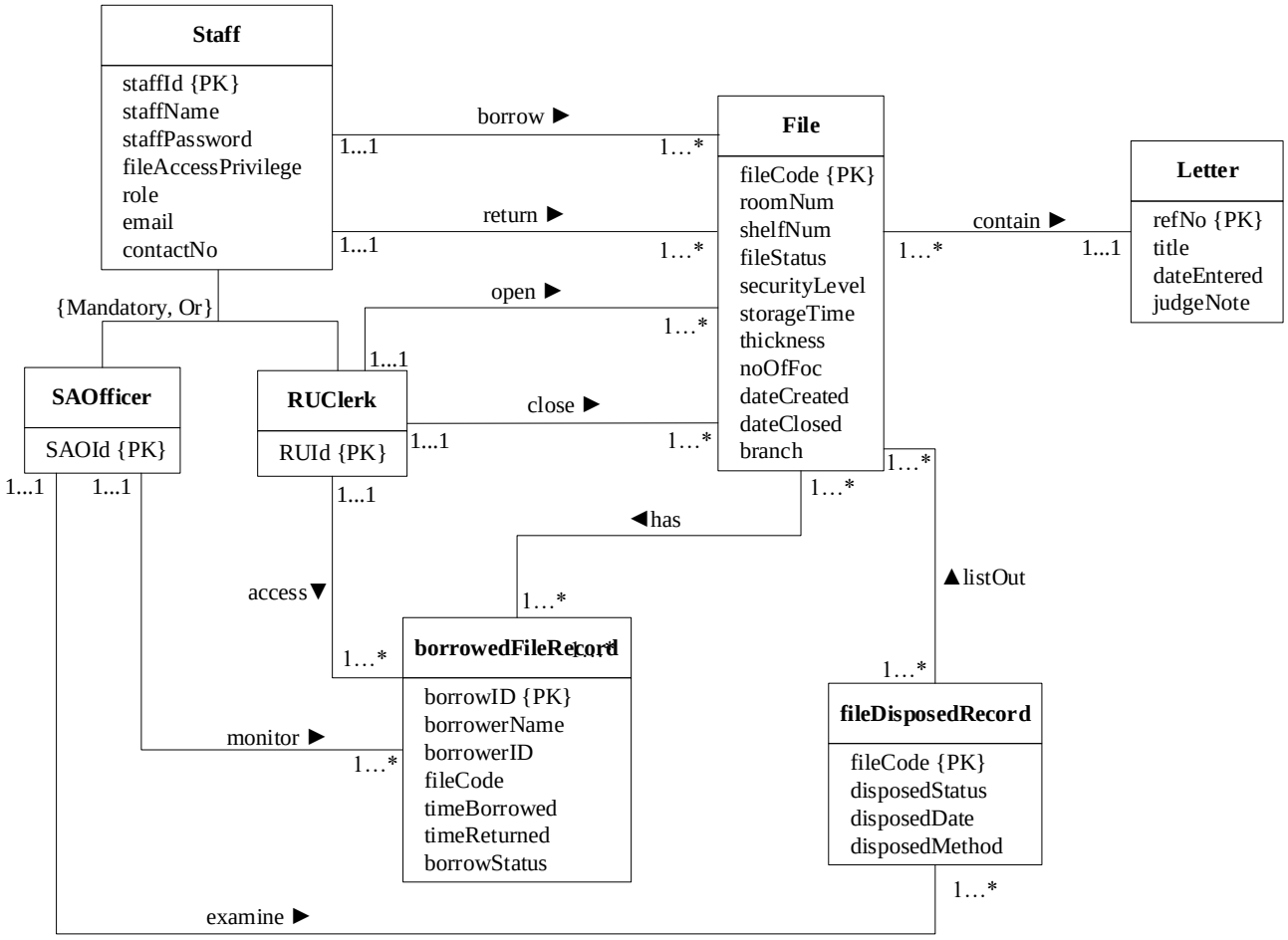

Table 5.1: Description of Entities in the Database

No.	Entity Name	Description
1.	Staff	Store basic information of staff
2.	SAOfficer	Store staff ID of Syariah Assistant Officer
3.	RUClerk	Store staff ID of Clerks of Record Unit
4.	File	Store File information
5.	Letter	Store Letter information
6.	borrowedFileRecord	Store borrowed File Record information
7.	fileDisposedRecord	Store File disposed Record information

5.2. Data Dictionary

5.2.1. Entity: <Staff>

Attribute Name	Type	Description
staffId	VARCHAR2(10)	Staff ID, Primary Key
staffName	VARCHAR2(50)	Full name of staff
staffPassword	VARCHAR2(50)	Password for login
fileAccessPrivilege	VARCHAR2(10)	File access privilege of Staff
role	VARCHAR2(25)	Staff's role
email	VARCHAR2(15)	Staff's email address
contactNo	VARCHAR2(12)	Staff's contact number

5.2.2. Entity: <SAOfficer>

Attribute Name	Type	Description
SAOId	VARCHAR2(10)	staff ID of Syariah Assistant Officer, Primary Key

5.2.3. Entity: <RUClerk>

Attribute Name	Type	Description
RUId	VARCHAR2(10)	staff ID of Clerks of Record Unit, Primary Key

5.2.4. Entity: <File>

Attribute Name	Type	Description
fileCode	VARCHAR2(10)	Code number of File, Primary Key
roomNum	INT	Room number of file storing
shelfNum	INT	Shelf number of File storing
fileStatus	VARCHAR2(10)	Current Status of File
securityLevel	VARCHAR2(10)	Security Level of File
storageTime	TIME	Storage Time of File
thickness	DOUBLE	Thickness of File
noOfDoc	INT	Number of File Document
dateCreated	DATE	Date Created of File
dateClosed	DATE	Date Closed of File
branch	VARCHAR2(50)	Branch of File

5.2.5. Entity: <Letter>

Attribute Name	Type	Description
refNo	INT	Reference number of Letter, Primary Key
title	VARCHAR2(50)	Title of Letter
dateEntered	DATE	Date Entered of Letter
judgeNote	VARCHAR2(50)	Judge Note of Letter

5.2.6. Entity: <borrowedFileRecord>

Attribute Name	Type	Description
borrowID	VARCHAR2(10)	ID of Borrowed File, Primary Key
borrowerName	VARCHAR2(50)	Borrower Name of File
borrowerID	VARCHAR2(10)	Borrower ID of File
fileCode	VARCHAR2(50)	Code number of File
timeBorrowed	DATETIME	Time Borrowed of File
timeReturned	DATETIME	Time Returned of File
borrowStatus	VARCHAR2(10)	Borrow Status of File

5.2.7. Entity: <fileDisposedRecord>

Attribute Name	Type	Description
fileCode	VARCHAR2(10)	Code number of File, Primary Key
disposedStatus	VARCHAR2(10)	Disposed status of File
disposedDate	DATE	Disposed Date of File
disposeMethod	VARCHAR2(10)	Disposed Method of File

6. User Interface Design

6.1. Overview of User Interface

Firstly, the base design of our system is the login function. The user needs to log in their ID and password before accessing other features in the system. After finish key in all the information, the system will bring the user to the menu page. Here, since they are three different users in our system which are JKSNJ staff, Clerk of the Record Unit and Syariah Assistant Officer therefore they will be different user interface for the menu page. However, Clerk of the Record Unit and Syariah Assistant Officer will have the same menu interface as they need to access the same features. Before we start to explain about the user interface of each feature, we need to state that the text which is colored in red is for the user to key in their input meanwhile text colored in black is for the user output.

For Clerk of the Record Unit and Syariah Assistant Officer, they will be six features that will be displayed which are Open File, Close File, Return File, Borrow File, File Disposal and File Handling meanwhile for JKSNJ staff, they will be only two features that will be displayed which are Borrow File and Return File. Therefore, the user just needs to click on the button to access the features of the system.

For borrow file, the user just needs key in the file code. After that, the system will display the borrower's name and ID before the system determine either the users are privilege or not to borrow the file. At the same time, the system also will determine either if the file is available or not for the users to borrow it. If the users are privilege and the file is available, then the user can proceed to borrow the file clicking the borrow button. Lastly, the system will display either the file is successfully borrowed or not.

For return file, the user just needs to key in the borrow ID in the system when the JKSNJ staff bring the file to return the file. Then, the system will display the borrower's information which are the borrower's name and the date returns. Then, the user can proceed to return the file. The process needs to be done in front of the Clerk of the Record Unit. If they are any file lost, they can click the report link so that the system can bring them to the report file loss page.

For dispose file, the user needs to classify the file either they are outdated or not. Then, the system shall display the file information which are the file code, room number, title and shell number of the file that need to be disposed. Then, they will be some time required for the system to approve the request send for file disposal. Lastly, the system will proceed with the file disposal.

For open file, the user needs to key in the file code to make sure that they are no file already created in the system. If the file is not created in the system, then the system shall ask the user to key in the new file code, room number, security level of the file and the shell number. Then, the user also needs to key in either they are any letter in the new file and the information of the letter registration which are reference number, title, and the judge note. If they are no letter assigned to the new file, the system will display that they are no file registration.

For close file, the user just needs to find the file by their file code. Then, the system will display the information of the file. After that, the user can check either they want to open a new file or just delete the file. They are some requirements to open a new file which are the file need to have thickness over 4sm or 100 attachments meanwhile for delete file, the file need to be no longer in use.

For file handling, they will be two different user interfaces based on the two different users which are Clerk of the Record Unit and Syariah Assistant Officer. For Clerk of the Record Unit, they can track file location, view file access record and late return file. For Syariah Assistant Officer, they need to report file loss, view file access record and track file location. So, for them to access the features they just need to click the button before accessing each feature.

For track file location, the user just needs to search the file code and then the system shall display all the file information regarding the file which are the file code, room number, title, and shell number of the file. For view file access record, the user also needs to search for the file code and then the system shall display all the file information regarding the file which are the file code, room number, title, and shell number of the file. For report file loss, the Syariah Assistant Officer needs to key in the borrower's ID and the file code of the file loss. After that, they can search for the borrower file information to makes sure that the borrower exists or not. Then, they can report it to the system. Lastly, for late return file, the clerk of the record unit can choose or search for any late return of file. Then, the system will ask the user to key in the borrower's ID, name, and phone number. After that, they can message the borrower to inform them to return the file.

6.2. Screen Images

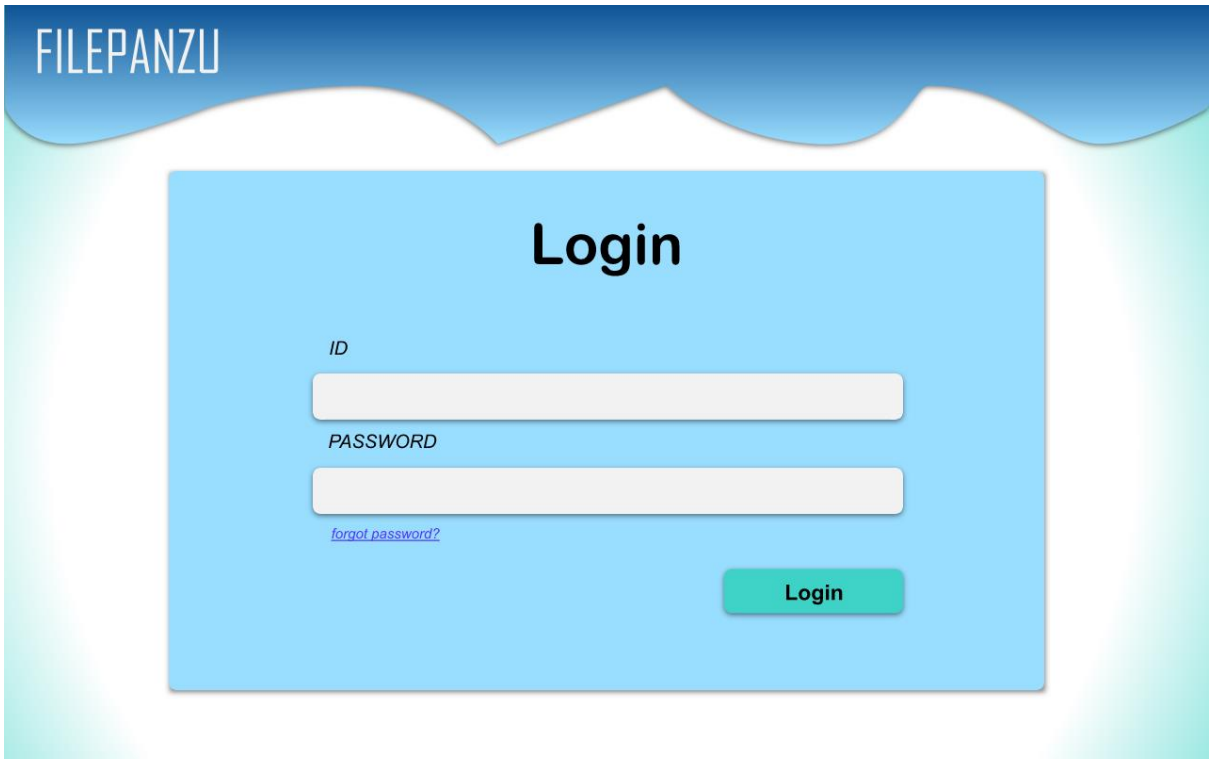

Figure 6.1: Interface for <Login>

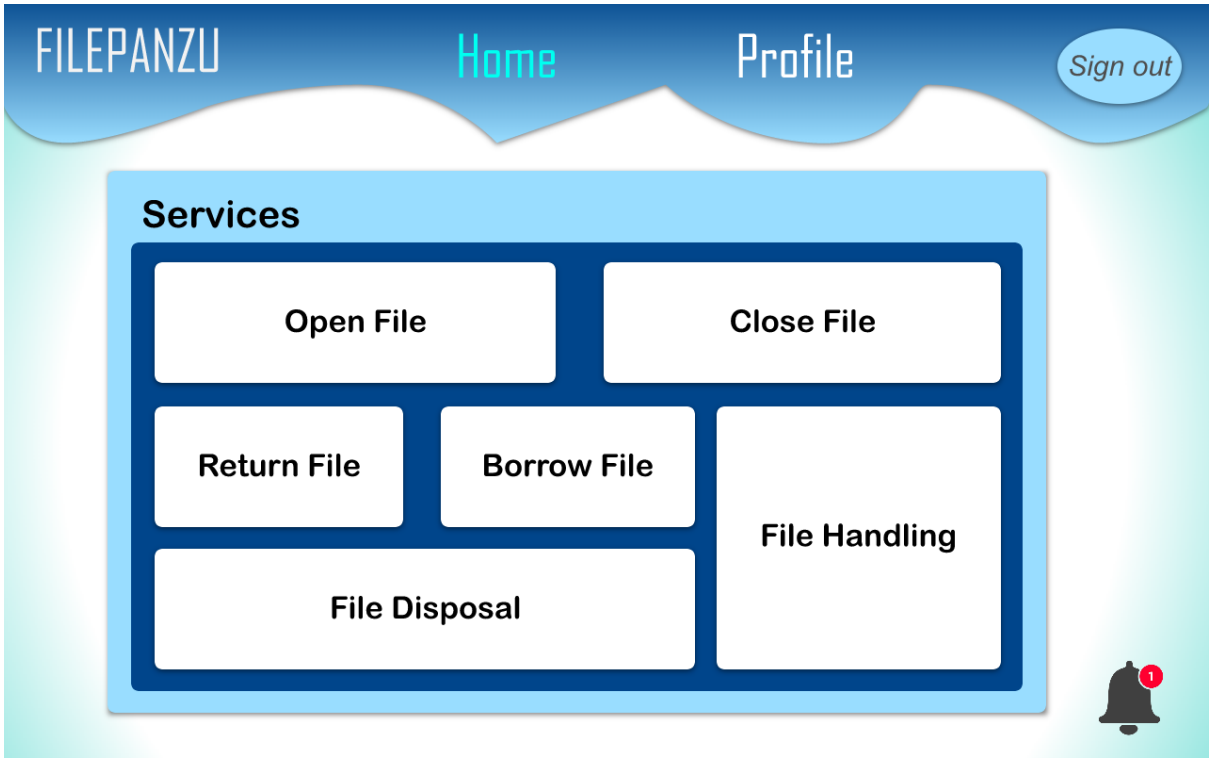

Figure 6.2: Interface for <Home Page for Clerk of Record Unit>

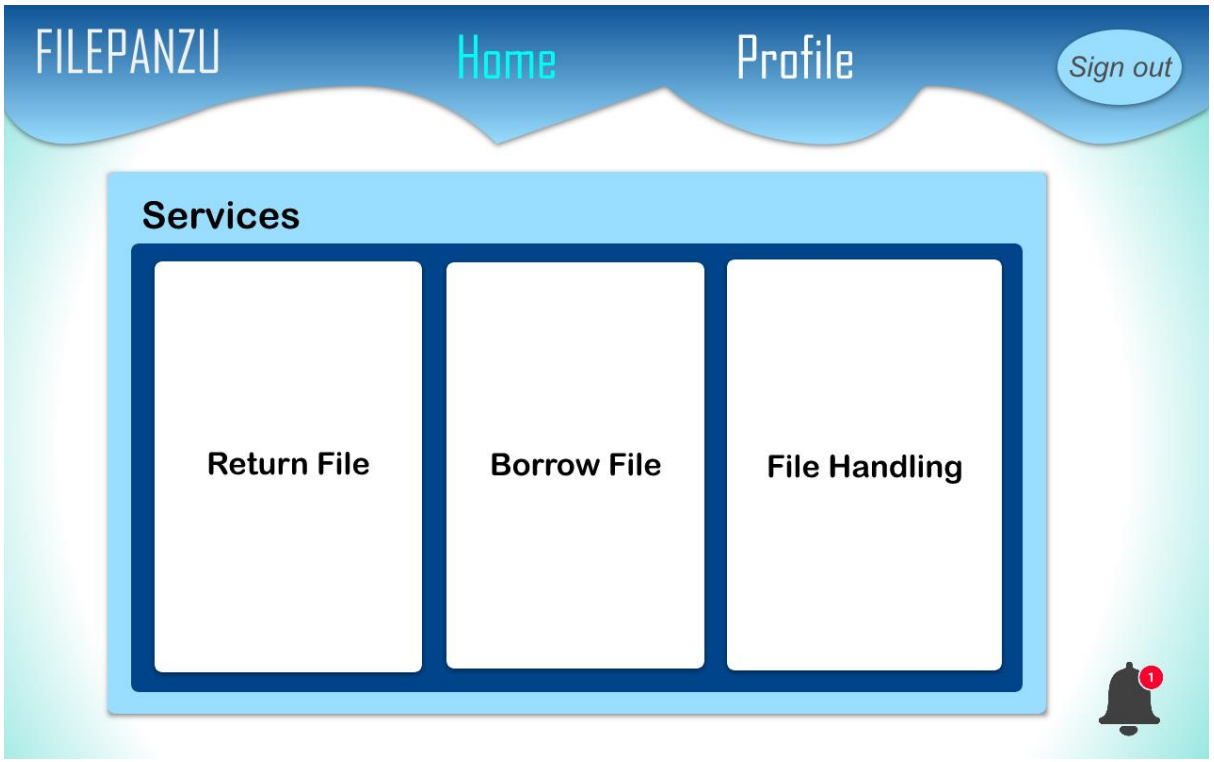

Figure 6.3: Interface for <Home Page for Syariah Assistant Officer>

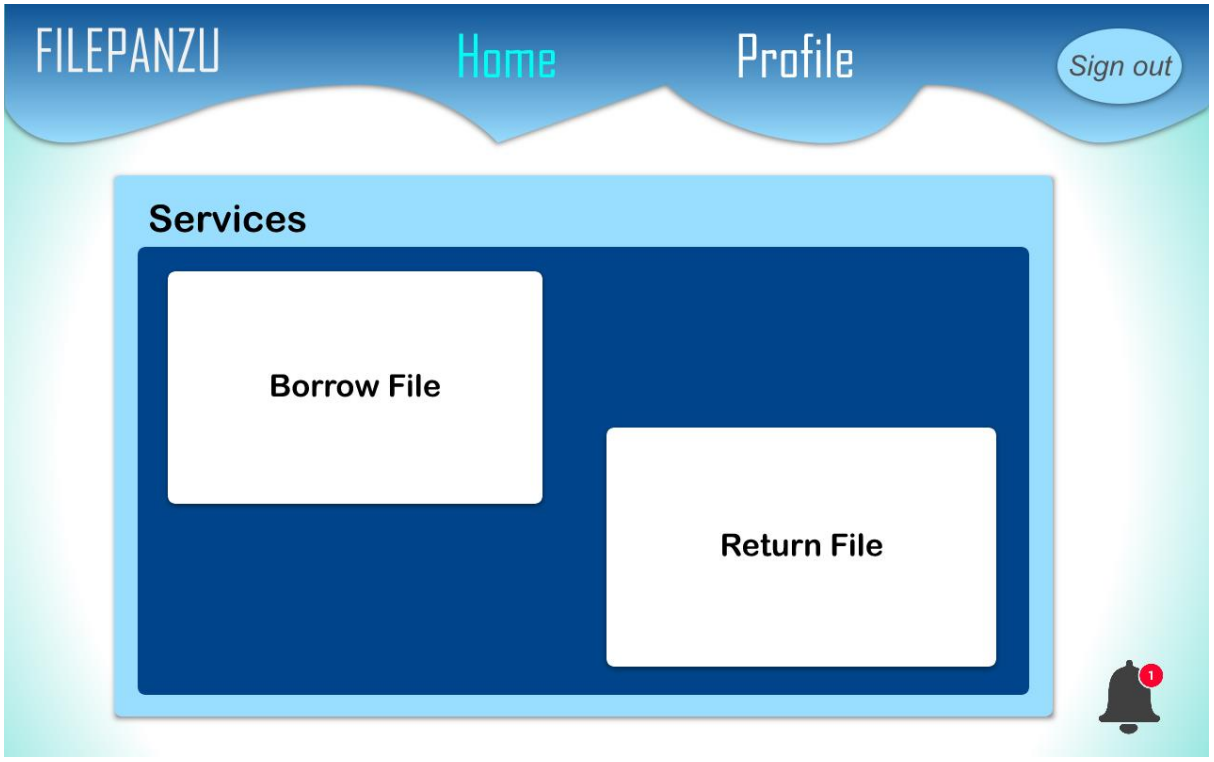

Figure 6.4: Interface for <Home Page for Normal Staff>

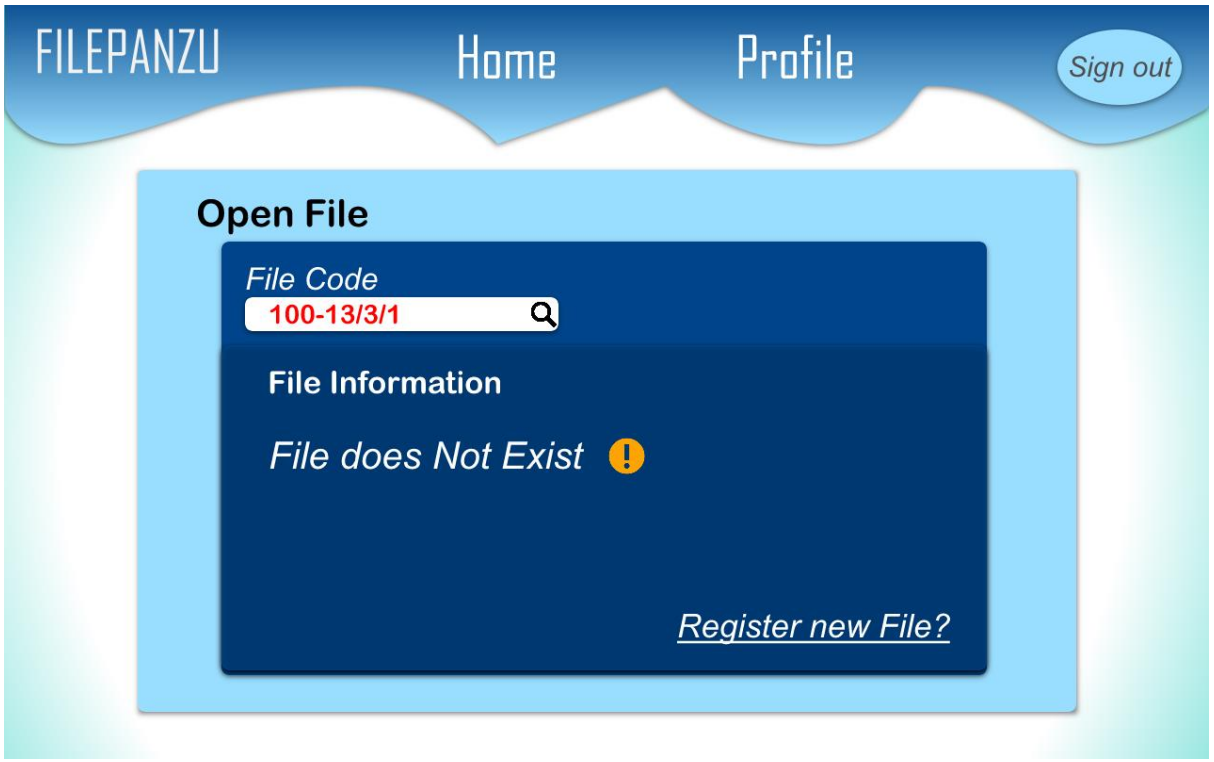

Figure 6.5: Interface for <Open File>

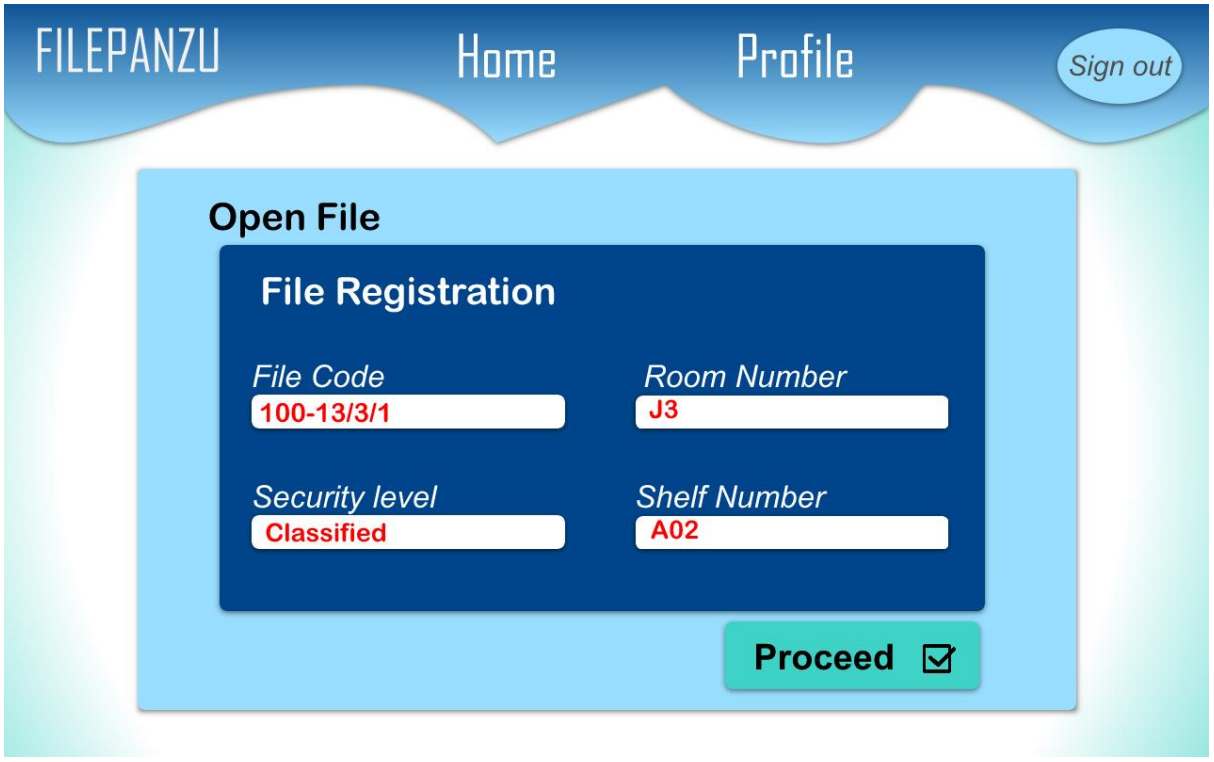

Figure 6.6: Interface for <Open File (File registration)>

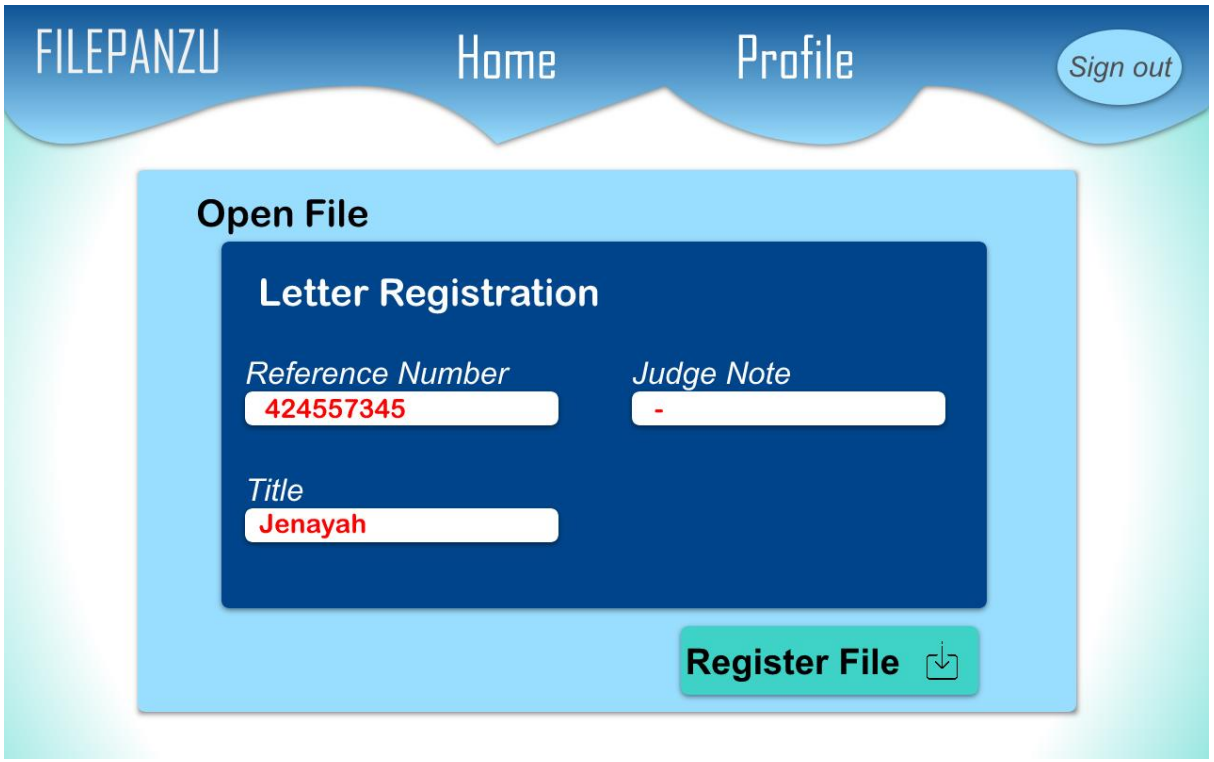

Figure 6.7: Interface for <Open File (Letter Registration)>

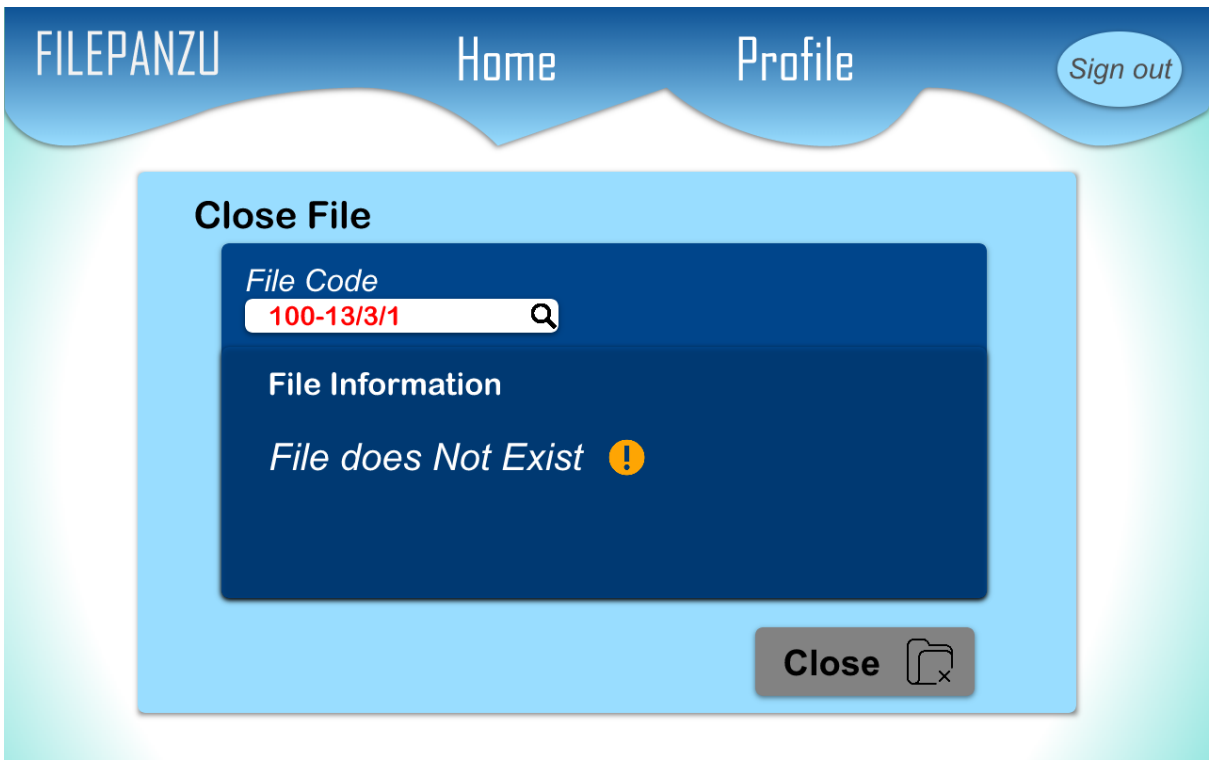

Figure 6.8: Interface for <Close File>

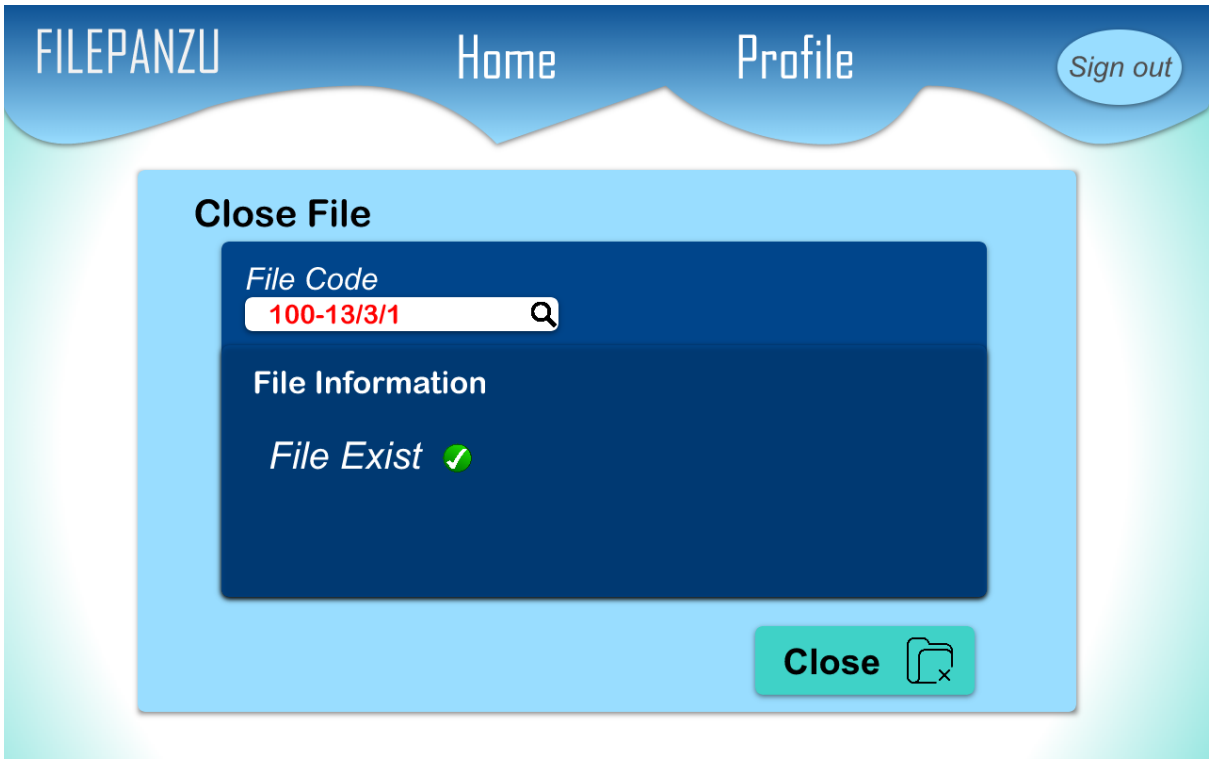

Figure 6.9: Interface for <Close File (File Exist)>

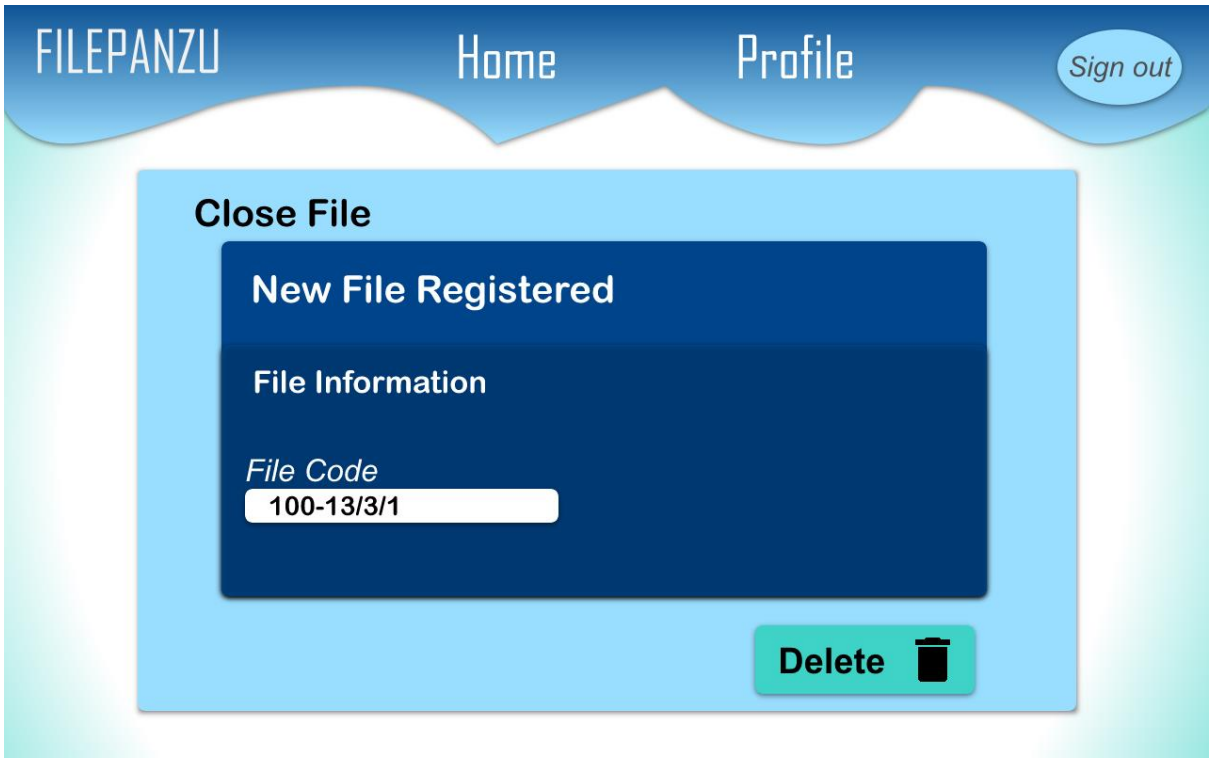

Figure 6.10: Interface for <Close File (New File Opened)>

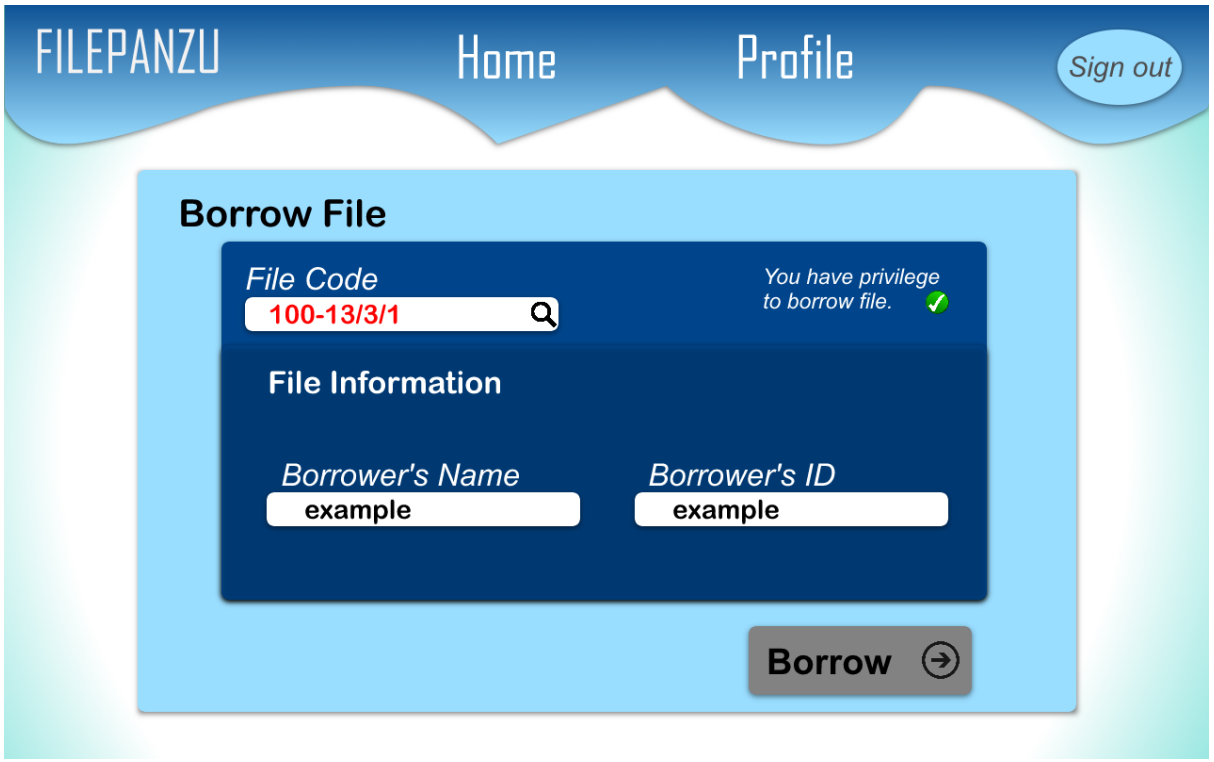

Figure 6.11: Interface for <Borrow File>

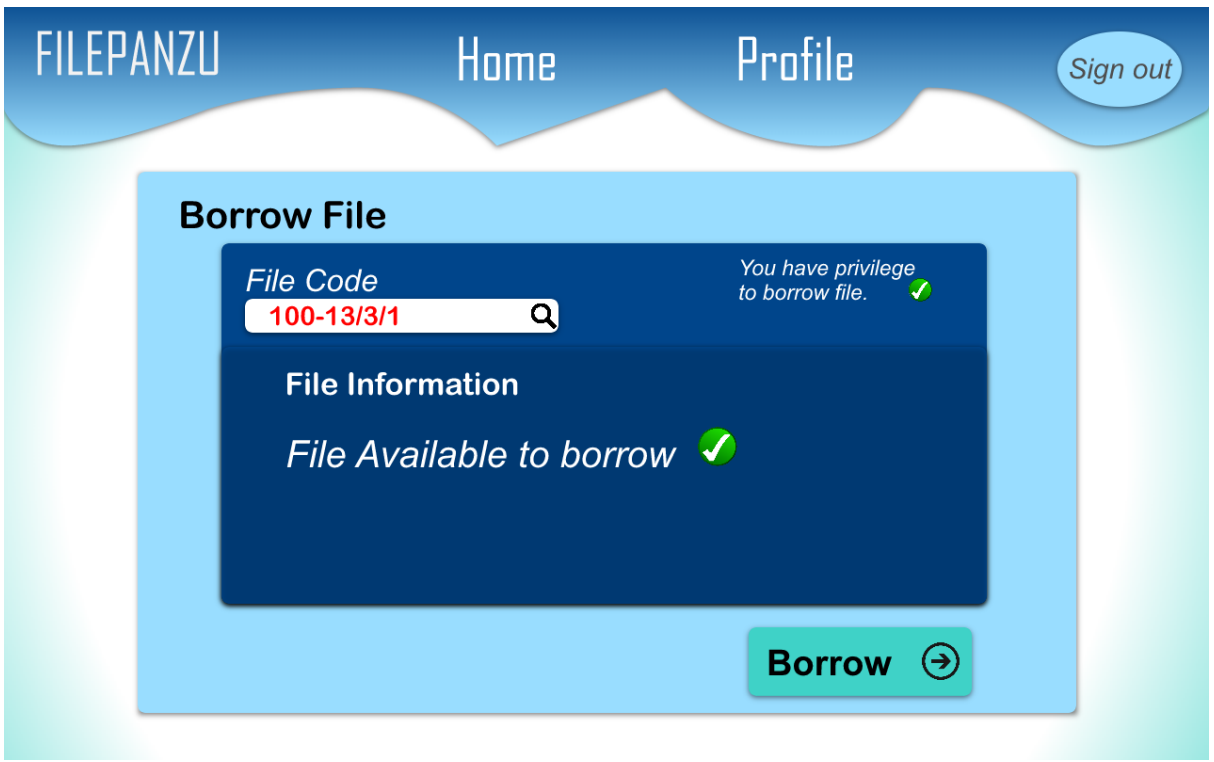

Figure 6.12: Interface for <Borrow File (Allowed to Borrow)>

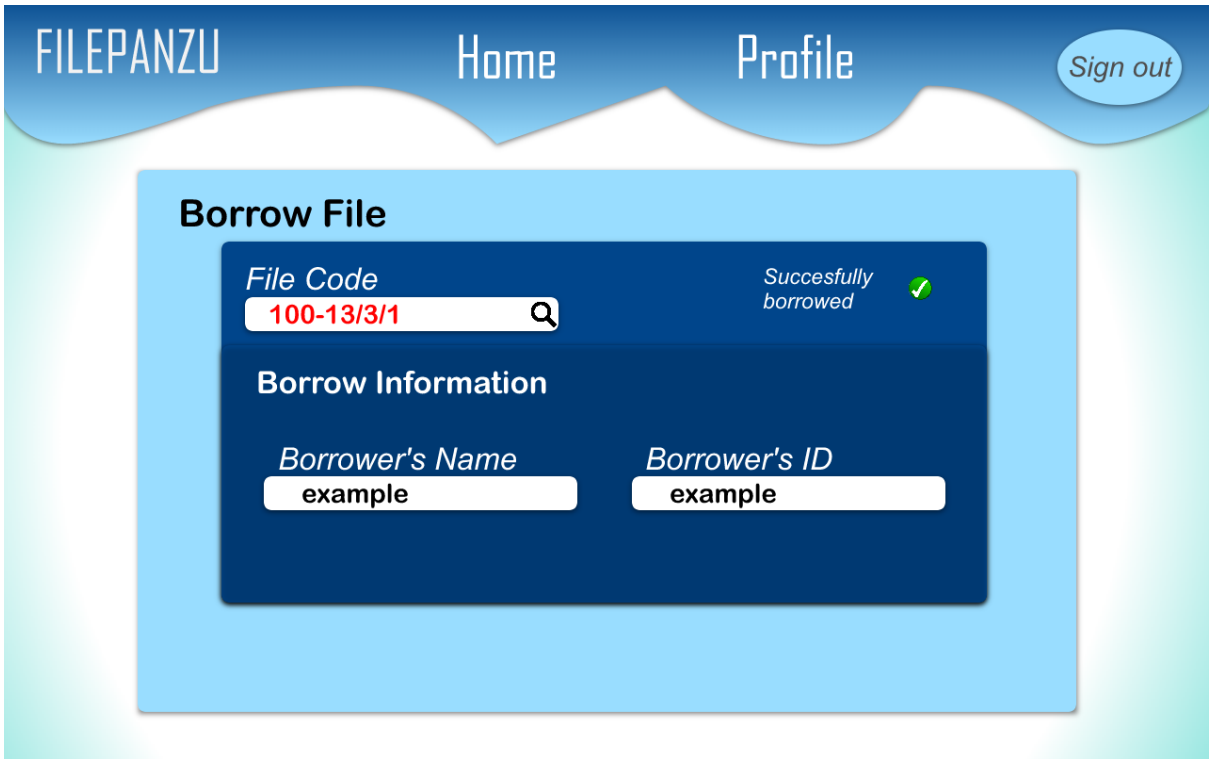

Figure 6.13: Interface for <Borrow File (Borrow Succesfully)>

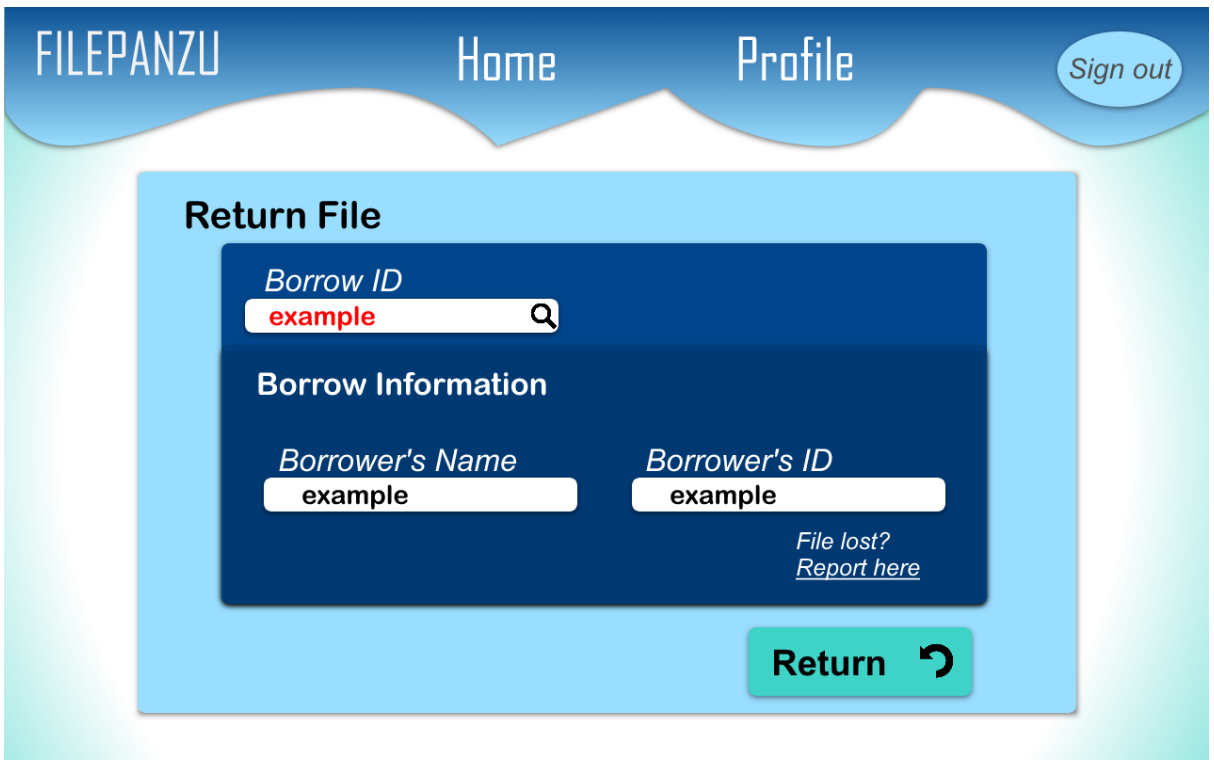

Figure 6.14: Interface for <Return File>

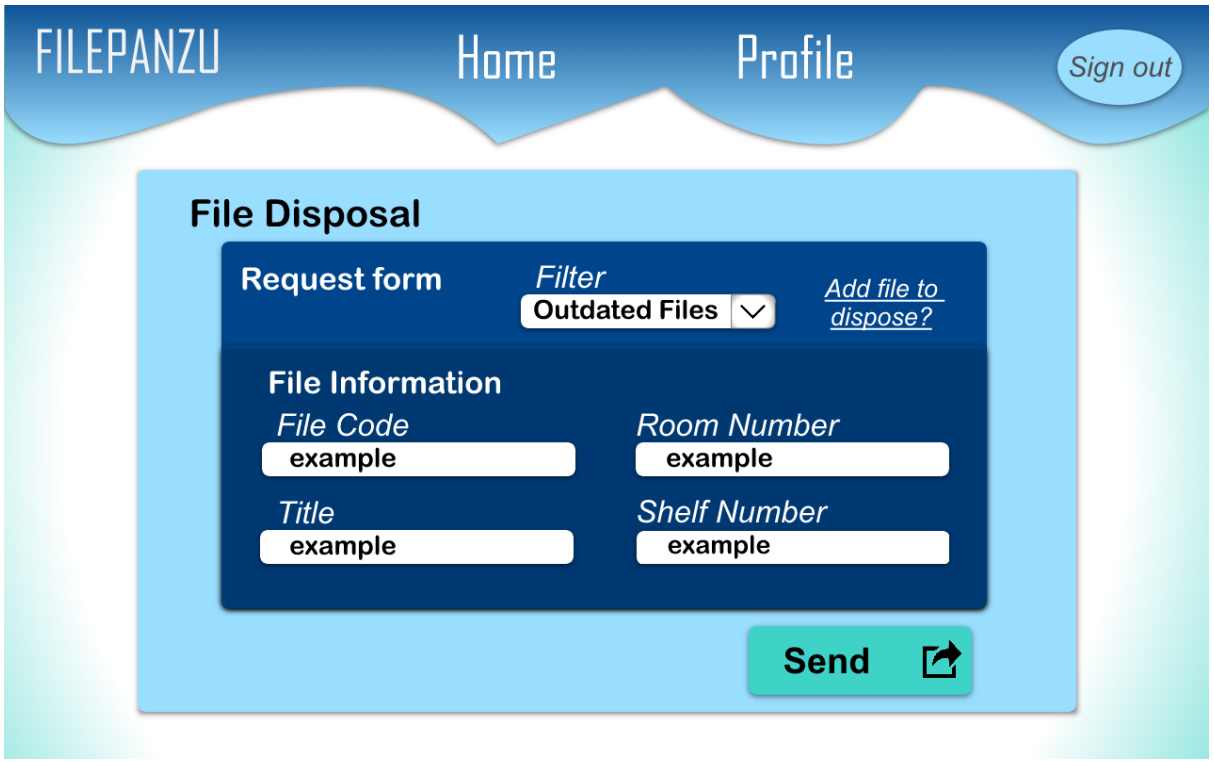

Figure 6.15: Interface for <File Disposal>

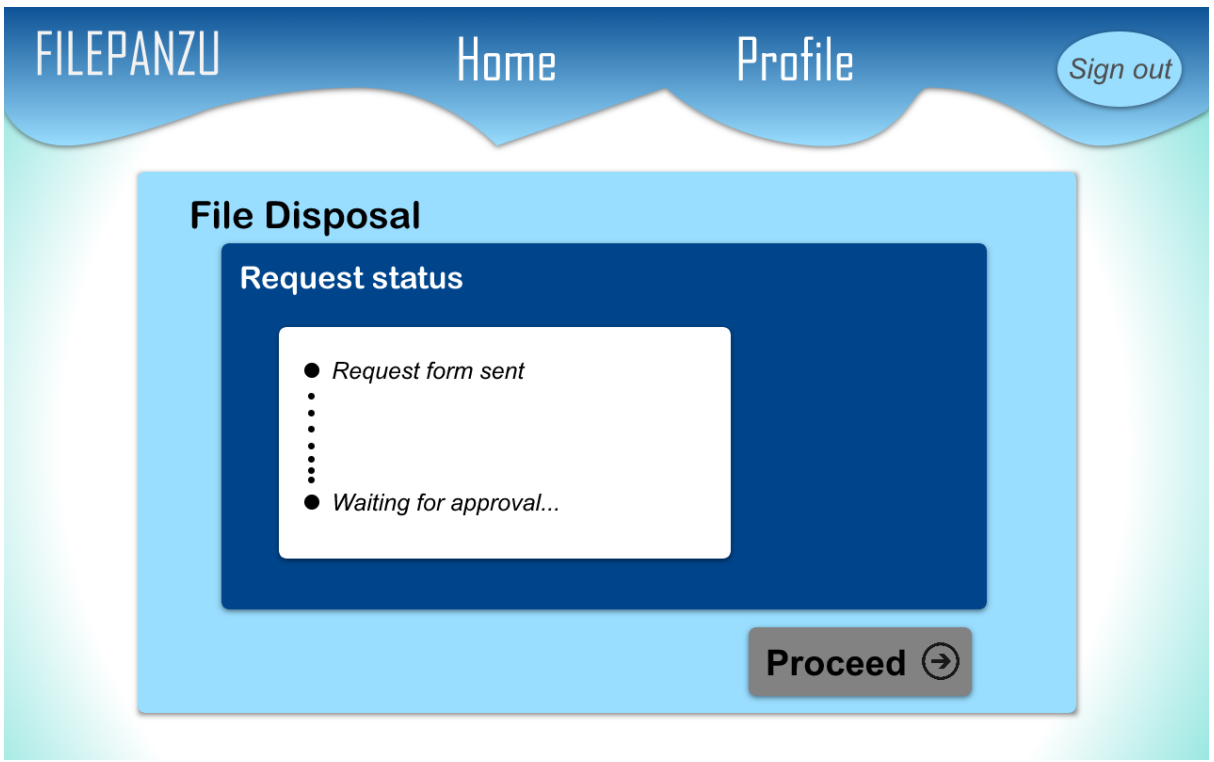

Figure 6.16: Interface for <File Disposal (Approval status)>

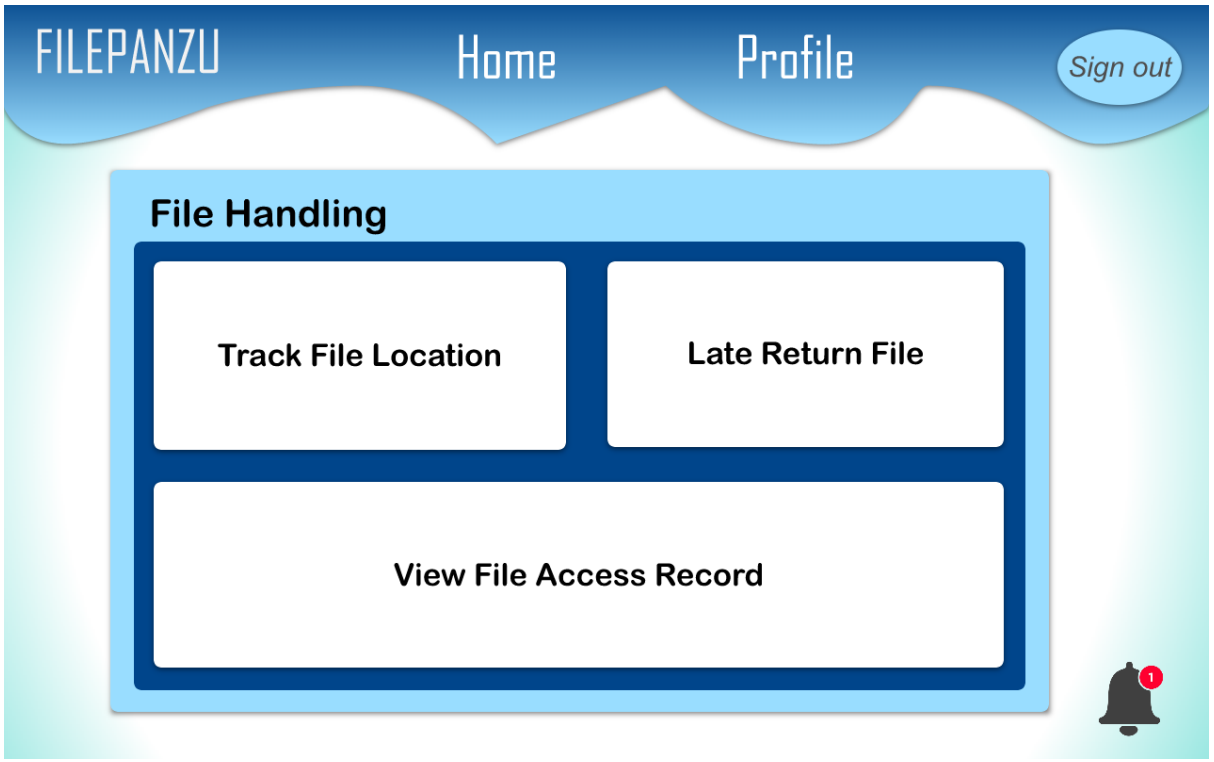

Figure 6.17: Interface for <File Handling for Clerk of Record Unit>

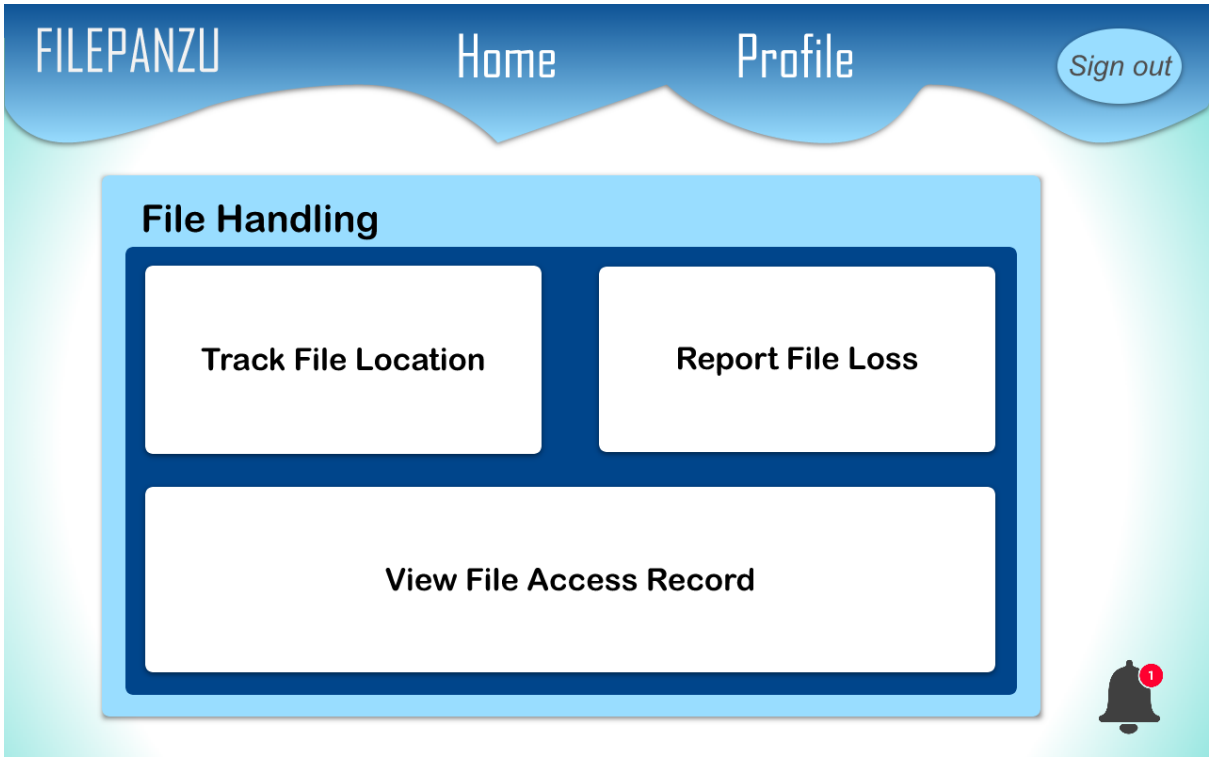

Figure 6.18: Interface for <File Handling for Syariah Assistant Officer>

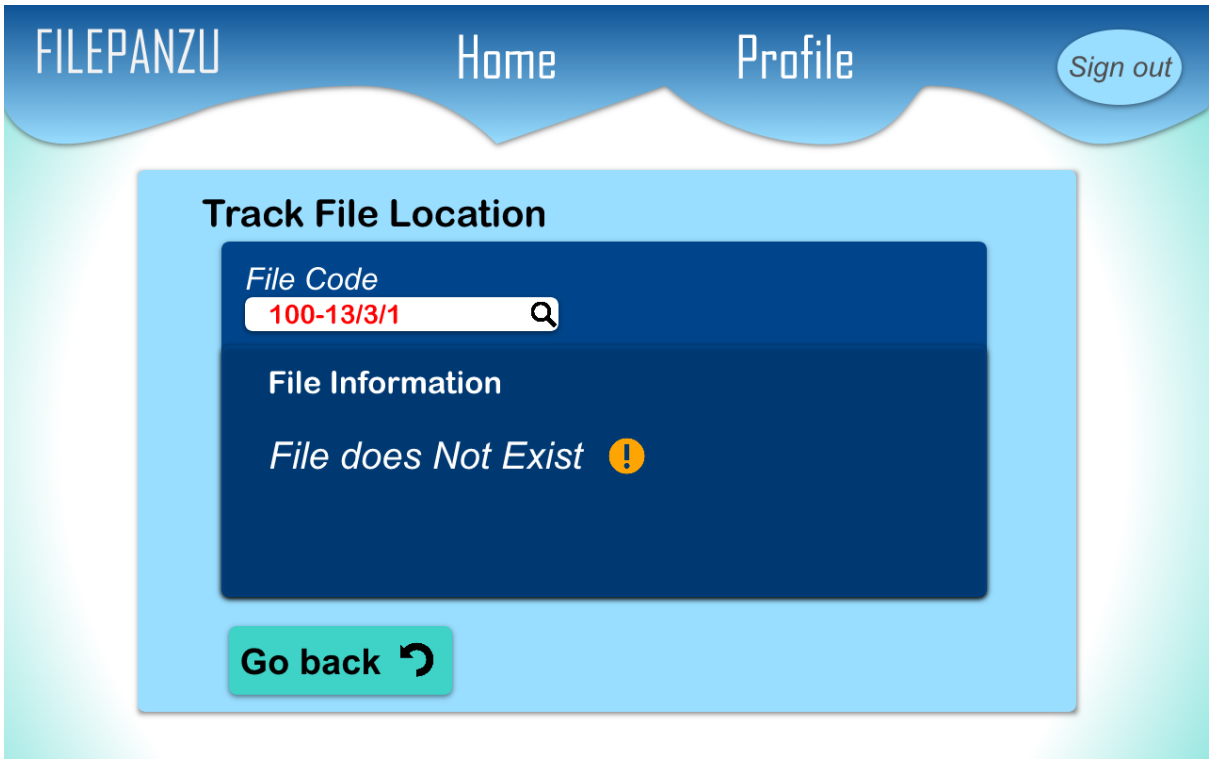

Figure 6.19: Interface for <Track File Location>

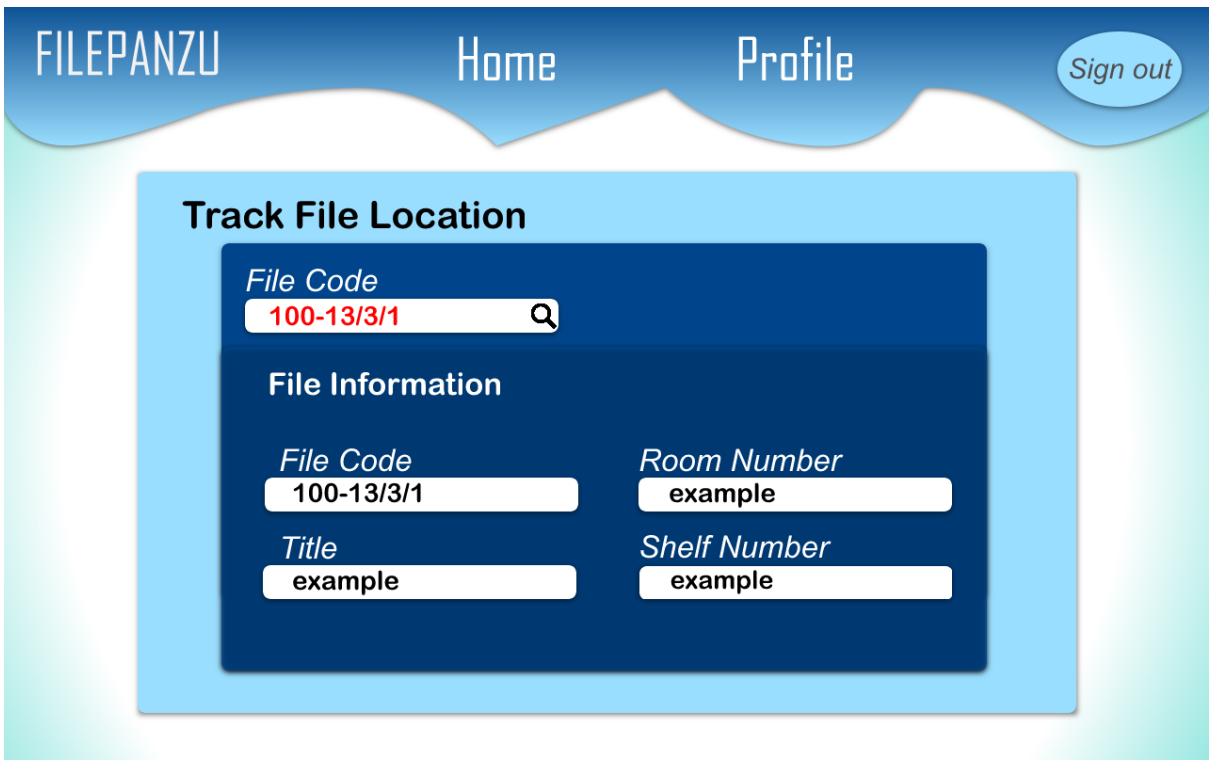

Figure 6.20: Interface for <Track File Location (Information Displayed)>

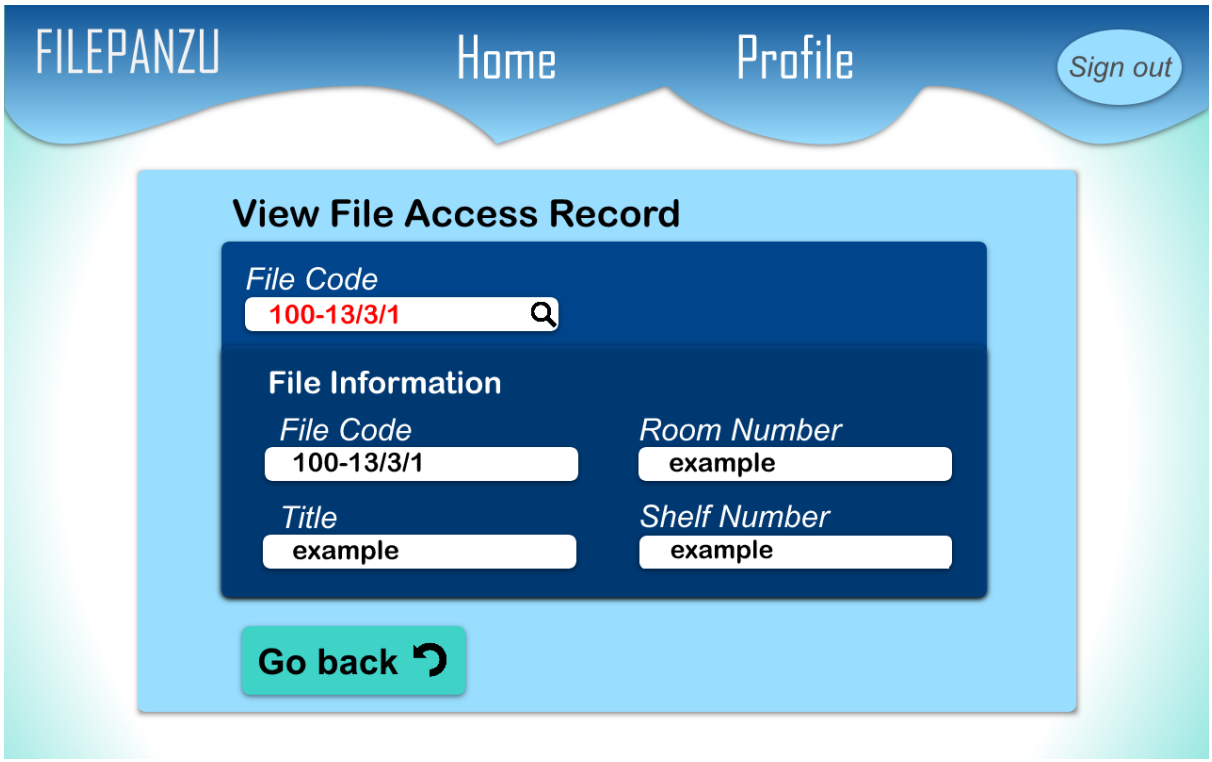

Figure 6.21: Interface for <View File access Record>

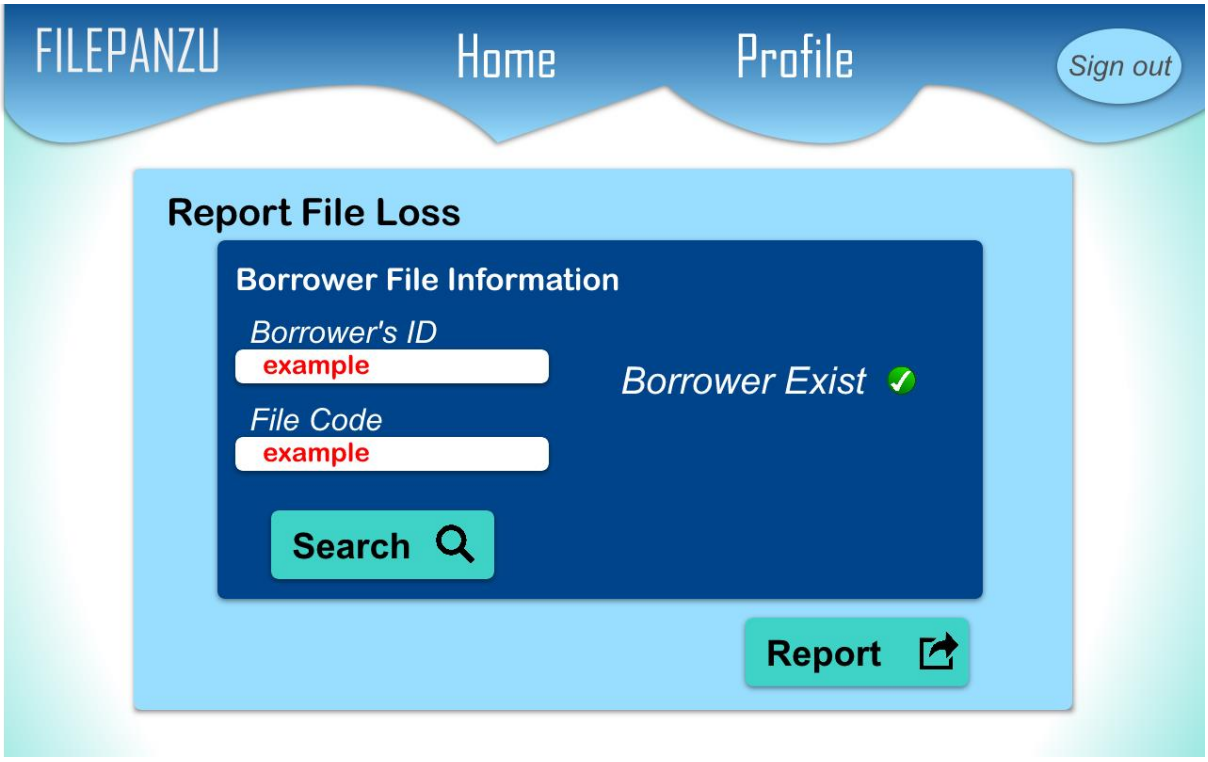

Figure 6.22: Interface for <Report File Loss>

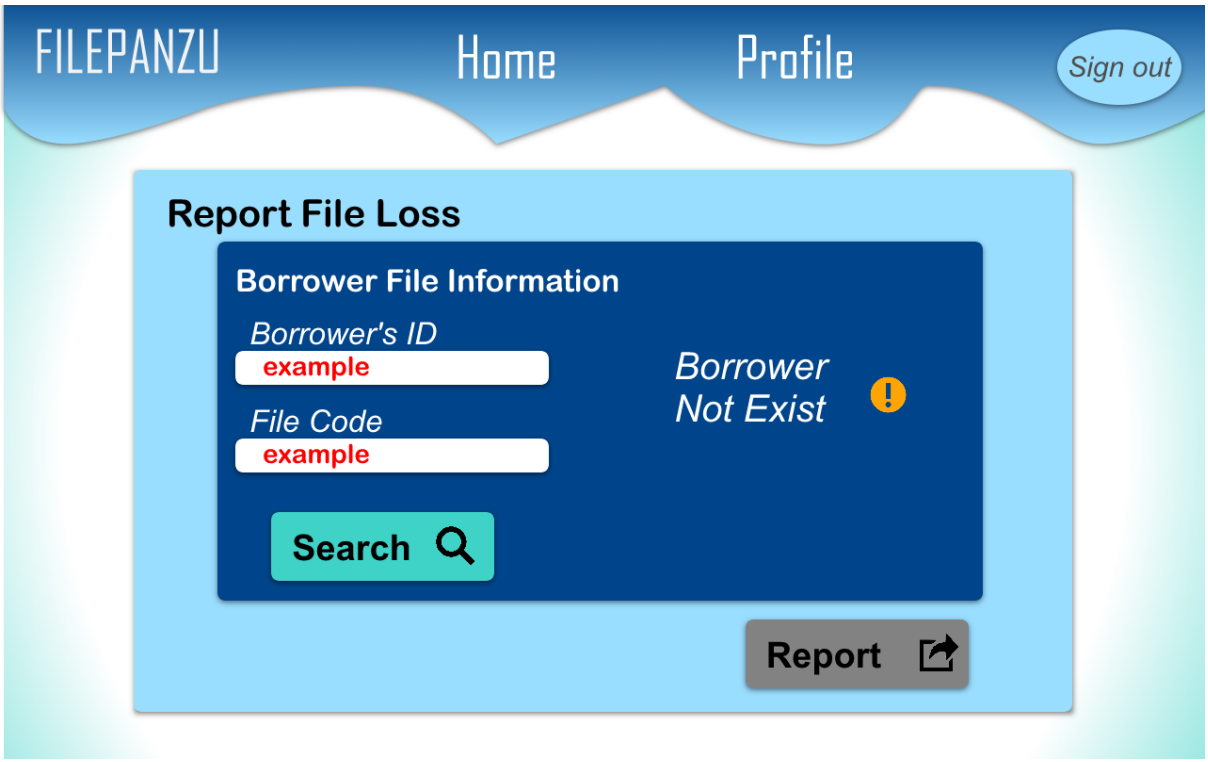

Figure 6.23: Interface for <Report File Loss (Borrower Not Exist)>

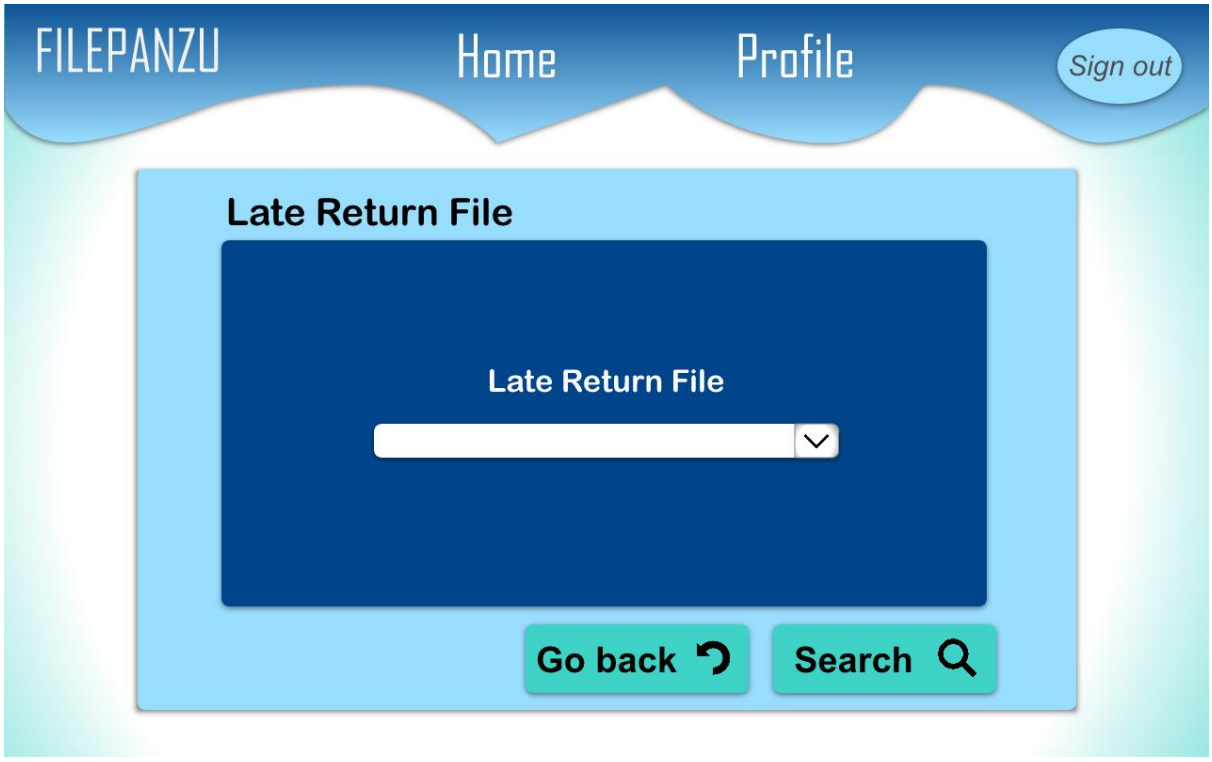

Figure 6.24: Interface for <Late Return of File>

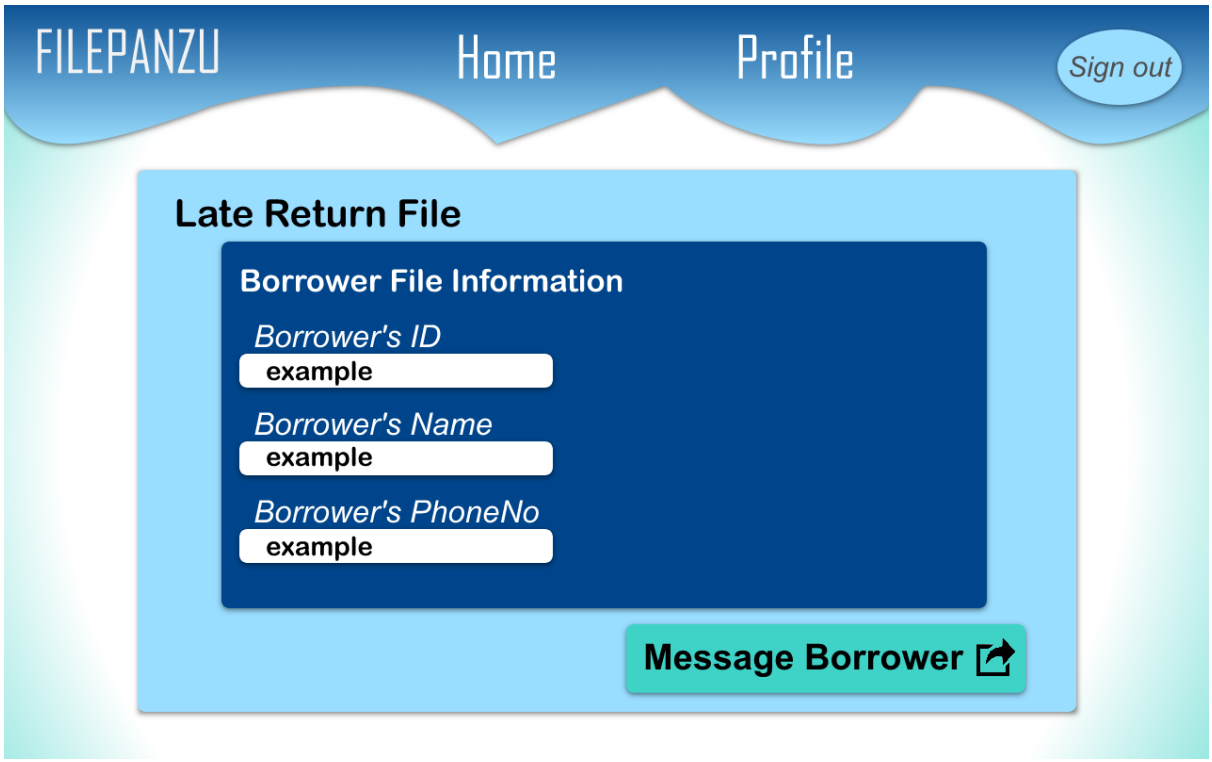

Figure 6.25: Interface for <Late Return of File (Notify Borrower)>

7. Requirements Matrix

The sequence diagrams for each use case vs. corresponding classes (entities) are listed as in Table 7.1.

Table 7.1: Description of Entities in the Database

	Staff	SAOfficer	RUClerk	File	Letter	borrowedFileRecord	fileDisposedRecord
P001, UC010, SD001	X	X	X				
P002, UC001, SD002			X	X	X		
P002, UC002, SD003			X	X			
P002, UC003, SD004			X	X			
P002, UC004, SD005	X	X	X				
P002, UC005, SD006			X	X			X
P002, UC006, SD007	X		X	X		X	
P002, UC007, SD008			X			X	
P002, UC008, SD009	X		X			X	
P002, UC009, SD010			X			X	

8. Test Cases

8.1. TC001: Test Login Subsystem

This test contains the following test cases:

(a) TC001_01: Test Login (SD001)

8.1.1. TC001_01: Test Login (SD001).

This test contains the following alternate and exception scenarios (if any):

a) TC001_01_01: Test normal flow of login using staff ID and password (SD001)

Test Case ID	TC001_01_01	Test Case Description	Test normal flow of login	
Created By	Tan Chiaw Torng	Reviewed By	Version	
QA Tester's Log				
Tester's Name	Tan Chiaw Torng	Date Tested	16-Jun-21	Test Case (Pass/Fail/

S#	Prerequisites:		S#	Test Data
1	Access to Chrome Browser		1	Staff ID: A10001, Password: 123
2	User is not logged in		2	Staff ID: A10002, Password: 456
3	User's data is in the database		3	Staff ID: A10003, Password: 789
4			4	

Test Scenario	Verify on entering valid staff id and password, the customer can login	

Step#	Step Details	Expected Results	Actual Results	Pass/Fail/Not executed/Suspended
1	Navigate the Mipanzu website	Website can be loaded.		
2	Enter staff ID and password	Data suceessfully entered		
3	Click on Login	Display home page according to the role		

b) TC001_01_02: Test exception flow of login using staff ID and password (SD001)

Test Case ID	TC001_01_02	Test Case Description	Test exception flow of login	
Created By	Tan Chiaw Torng	Reviewed By	Version	
QA Tester's Log				
Tester's Name	Tan Chiaw Torng	Date Tested	16-Jun-21	Test Case (Pass/Fail/

S#	Prerequisites:		S#	Test Data
1	Access to Chrome Browser		1	Staff ID: B10001, Password: 000
2	User is not logged in		2	Staff ID: E10002, Password: 000
3	User's data is not in the database		3	Staff ID: J10003, Password: 000
4			4	

Test Scenario	Fail to verify on entering valid staff id and password, the customer cannot login	

Step#	Step Details	Expected Results	Actual Results	Pass/Fail/Not executed/Suspended
1	Navigate the Mipanzu website	Website can be loaded.		
2	Enter staff ID and password	Data suceessfully entered		
3	Click on Login	Display error message		

8.2. TC002: Test File Subsystem

This test contains the following test cases:

(a) TC002_01: Test Open New File (SD002)

(b) TC002_02: Test Close File (SD003)

(c) TC002_03: Test Track File Location (SD004)

(d) TC002_04: Test Report the loss of file (SD005)

(e) TC002_05: Test Dispose of File (SD006)

(f) TC002_06: Test Borrow File (SD007)

(g) TC002_07: Test View file access record (SD008)

(h) TC002_08: Test Detect the late return of file (SD009)

(i) TC002_09: Test Return File (SD010)

8.2.1. TC002_01: Test Open New File (SD002)

This test contains the following alternate and exception scenarios (if any):

(a) TC002_01_01: Test normal flow of open new file (SD002)

Test Case ID	TC002_01_01	Test Case Description	Test normal flow of open new file	
Created By	Wan Luqman	Reviewed By		Version

QA Tester's Log

Tester's Name		Date Tested		Test Case (Pass/Fail/Not Executed)	

S #	Prerequisites:
1	User is logged in to the system as a clerk of the record unit
2	There are no file with the same code created in the system
3	There will be no letter already created in the system
4	

S #	Test Data Requirement
1	Filecode: 100-19/1/2;Room Number: J3;Security Level: Classified;Shelf Number: A02
2	Reference Number: 424557345 Judge Note:- Title: Jenayah
3	
4	

Test Scenario	Verify on entering a valid file code and letter information, the new file can be open.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Key in detail of the file	Display file is successfully created message		
2	Key in the letter information	Display letter is successfully created message		

(b) TC002_01_02: Test exception flow 1 of open new file (SD002)

Test Case ID	TC002_01_02	Test Case Description	Test exception flow 1 of open new file	
Created By	Wan Luqman	Reviewed By		Version

QA Tester's Log

Tester's Name		Date Tested		Test Case (Pass/Fail/Not Executed)	

S #	Prerequisites:
1	User is logged in to the system as a clerk of the record unit
2	There are file with the same code created in the system
3	There will be letter already created in the system
4	

S #	Test Data Requirement
1	Reference Number: 424557345 Judge Note:- Title: Jenayah
2	
3	
4	

Test Scenario	The file code entered is already exists

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Key in the letter information	Display letter exists message		

8.2.2. TC002_02: Test Close File (SD003)

a) TC002_02_01: Test normal flow of close file (SD003)

Test Case ID	TC002_02_01	Test Case Description	Test normal flow of close file	
Created By	Husny Musharraf	Reviewed By		Version

QA Tester's Log

Tester's Name		Date Tested		Test Case (Pass/Fail/Not Executed)	

S #	Prerequisites:	S #	Test Data Requirement
1	User is logged in to the system as a clerk of the record unit	1	Filecode: 100-19/1/2
2		2	
3		3	
4		4	

Test Scenario	Verify on entering a valid file code, the file can be close.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Key in the file code	Display file exists message		
2	Click on Close button	Display file is successfully closed message		

b) TC002_02_02: Test alternative flow of close file (SD003)

Test Case ID	TC002_02_02	Test Case Description	Test alternative flow of close file	
Created By	Husny Musharraf	Reviewed By		Version

QA Tester's Log

Tester's Name		Date Tested		Test Case (Pass/Fail/Not Executed)	

S #	Prerequisites:	S #	Test Data Requirement
1	User is logged in to the system as a clerk of the record unit	1	-
2	File is successfully closed	2	
3		3	
4		4	

Test Scenario	User does not click 'Delete' of the closed file.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Click Go Home	File is not deleted		

c) TC002_02_02: Test exception flow of close file (SD003)

Test Case ID	TC002_02_03	Test Case Description	Test exception flow of close file	
Created By	Husny Musharraf	Reviewed By		Version

QA Tester's Log

Tester's Name		Date Tested		Test Case (Pass/Fail/Not Executed)	

S #	Prerequisites:	S #	Test Data Requirement
1	User is logged in to the system as a clerk of the record unit	1	-
2	File code is keyed into the system	2	
3		3	
4		4	

Test Scenario	Fail to verify on the entered file code.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	-	Display error message		

8.2.3. TC002_03: Test Track File Location (SD004)

(a) TC002_03_01: Test normal flow of track file location (SD004)

Test Case ID	TC002_03_01	Test Case Description	Test normal flow of track file location	
Created By	Wan Luqman	Reviewed By		Version

QA Tester's Log

Tester's Name		Date Tested		Test Case (Pass/Fail/Not Executed)	

S #	Prerequisites:	S #	Test Data Requirement
1	User is logged in to the system as a clerk of the record unit	1	Filecode: 100-19/1/2
2	File exist in the system	2	
3		3	
4		4	

Test Scenario	Verify on entering a valid file code, the current location of the file can be track.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Key in file code	File exists displayed		
2	Click search button	Location of the file displayed		

(b) TC002_03_02: Test exception flow of track file location (SD004)

Test Case ID	TC002_03_02	Test Case Description	Test exception flow of track file location	
Created By	Wan Luqman	Reviewed By		Version

QA Tester's Log

Tester's Name		Date Tested		Test Case (Pass/Fail/Not Executed)	

S #	Prerequisites:	S #	Test Data Requirement
1	User is logged in to the system as a clerk of the record unit	1	-
2	File not exist in the system	2	
3		3	
4		4	

Test Scenario	Fail to verify on the entered file code.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	-	File not exists and error message displayed		

8.2.4. TC002_04: Test Report the loss of file (SD005)

(a) TC002_04_01: Test normal flow of report the loss of file (SD005)

Test Case ID	TC002_04_01	Test Case Description	Test normal flow of report the loss of file		
Created By	Husny Musharraf	Reviewed By		Version	
QA Tester's Log					
Tester's Name		Date Tested		Test Case (Pass/Fail/Not Executed)	

S #	Prerequisites:	S #	Test Data Requirement
1	User is logged in to the system as JKSNJ staff	1	Filecode: 100-19/1/2;Borrower's ID: B19EC0001
2	Physical form of the file is missing	2	
3		3	
4		4	

Test Scenario	Verify on entering a valid file code and borrower's ID, the report of a loss file can be done.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Key in staff ID and file code	Display borrow file information		
2	Click Report button	Display successfully report message		

(b) TC002_04_02: Test exception flow of report the loss of file (SD005)

Test Case ID	TC002_04_02	Test Case Description	Test exception flow of report the loss of file		
Created By	Husny Musharraf	Reviewed By		Version	
QA Tester's Log					
Tester's Name		Date Tested		Test Case (Pass/Fail/Not Executed)	

S #	Prerequisites:	S #	Test Data Requirement
1	User is logged in to the system as JKSNJ staff	1	-
2	Record not found	2	
3		3	
4		4	

Test Scenario	Fail to verify on the entered file code and borrower's ID.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	-	Display error message		

8.2.5. TC002_05: Test Dispose of File (SD006)

(a) TC002_05_01: Test normal flow of dispose of file (SD006)

Test Case ID	TC002_05_01	Test Case Description	Test normal flow of dispose of file		
Created By	Tan Chiaw Torng	Reviewed By		Version	
QA Tester's Log					
Tester's Name	Tan Chiaw Torng	Date Tested	16-Jun-21	Test Case (Pass/Fail/Not)	

S#	Prerequisites:	S#	Test Data
1	A valid clerk of record unit is logged in.	1	Filecode: 100-13/3/3, file thickness:20&65, disposal status:Approved, disposal method: Moved to ANM, disposal date: 16/6/2021, disposed
2	A valid Syariah Assistant Officer is logged in.	2	Filecode: 105-13/3/4, file thickness:30&68, disposal status:Approved, disposal method: Burned, disposal date: 18/6/2021, disposed status:Disposed
3	Access file disposal page	3	Filecode: 200-19/1/2, file thickness:50&80, disposal status:Approved, disposal method: Moved to ANM, disposal date: 19/6/2021, disposed
4	The database contains 1 file record that exceeds the maximum storage time.	4	

Test Scenario	Verify on the data entered, users complete the file disposal procedure for the outdated file from the database and 1 extra damaged file entered.

Step#	Step Details	Expected Results	Actual Results	Pass/Fail/Not executed/Suspended
1	The clerk of record unit clicks on 'Outdated Files'.	The list of files that exceed the maximum storage time is loaded.		
2	The clerk of record unit enter file code of extra file to be disposed of.	File code is successfully entered.		
3	The clerk of record unit clicks on 'Search'.	Display 'File code is found'		
4	The clerk of record unit clicks on 'Add'.	The details of file with the file code entered is loaded.		
5	The clerk of record unit clicks on 'Submit'.	The file disposal list is sent to Syariah Assistant Officer.		
6	Syariah Assistant Officer refreshes the file disposal page.	The file disposal list is displayed.		
7	Syariah Assistant Officer enters the thickness of file.	The thickness of file is successfully entered.		
8	Syariah Assistant Officer clicks on 'Email'.	Syariah Assistant Officer's Outlook compose page is loaded.		
9	Syariah Assistant Officer returns to file disposal page.	File disposal page is accessed.		
10	Syariah Assistant Officer clicks on 'Add to Disposal Record'.	The file disposal list is inserted into the disposal record database.		
11	Syariah Assistant Officer selects the file disposal status as "Approved".	File disposal status is successfully selected.		
12	Syariah Assistant Officer selects the file disposal method.	File disposal method is successfully selected.		
13	Syariah Assistant Officer clicks on 'Update'.	File disposal status and method are updated in the database.		
14	The clerk of record unit clicks on 'Approved List'	The approved file disposal list is loaded.		
15	The clerk of record unit selects disposal date.	File disposal date is successfully selected.		
16	The clerk of record unit selects disposal status as "Disposed".	File disposal status is successfully selected.		
17	The clerk of record unit clicks on 'Update'.	File disposal status and date are updated in the database.		

(b) TC002_05_02: Test exception flow of dispose of file (SD006)

Test Case ID	TC002_05_02	Test Case Description	Test exception flow of dispose of file		
Created By	Tan Chiaw Torng	Reviewed By		Version	
QA Tester's Log					
Tester's Name	Tan Chiaw Torng	Date Tested	16-Jun-21	Test Case (Pass/Fail/Not)	

S#	Prerequisites:	S#	Test Data
1	A valid clerk of record unit is logged in.	1	Filecode: 108-13/3/3, file thickness:65, disposal status:Rejected
2	A valid Syariah Assistant Officer is logged in.	2	Filecode: 111-13/3/4, file thickness:68, disposal status:Rejected
3	Access file disposal page	3	Filecode: 130-19/1/2, file thickness:80, disposal status:Rejected
4	The database contains no file record that exceeds the maximum storage time.	4	

Test Scenario	File disposal application is rejected, users fail to dispose of file

Step#	Step Details	Expected Results	Actual Results	Pass/Fail/Not executed/Suspended
1	The clerk of record unit clicks on 'Outdated Files'.	The empty list of files that exceed the maximum storage time is loaded.		
2	The clerk of record unit enter file code of extra file to be disposed of.	File code is successfully entered.		
3	The clerk of record unit clicks on 'Search'.	Display 'File code is found'		
4	The clerk of record unit clicks on 'Add'.	The details of file with the file code entered is loaded.		
5	The clerk of record unit clicks on 'Submit'.	The file disposal list is sent to Syariah Assistant Officer.		
6	Syariah Assistant Officer refreshes the file disposal page.	The file disposal list is displayed.		
7	Syariah Assistant Officer enters the thickness of file.	The thickness of file is successfully entered.		
8	Syariah Assistant Officer clicks on 'Email'.	Syariah Assistant Officer's Outlook compose page is loaded.		
9	Syariah Assistant Officer returns to file disposal page.	File disposal page is accessed.		
10	Syariah Assistant Officer clicks on 'Add to Disposal Record'.	The file disposal list is inserted into the disposal record database.		
11	Syariah Assistant Officer selects the file disposal status as "Rejected".	File disposal status is successfully selected.		
12	Syariah Assistant Officer clicks on 'Update'.	File disposal status is updated in the database.		

(c) TC002_05_03: Test alternative flow 1 of dispose of file (SD006)

Test Case ID	TC002_05_03	Test Case Description	Test alternate flow 1 of dispose of file		
Created By	Tan Chiaw Torng	Reviewed By		Version	
QA Tester's Log					
Tester's Name	Tan Chiaw Torng	Date Tested	16-Jun-21	Test Case (Pass/Fail/Not)	

S#	Prerequisites:	S#	Test Data
1	A valid clerk of record unit is logged in.	1	Filecode: 000-13/3/3, file thickness:20, disposal status:Approved, disposal method: Moved to ANM, disposal date: 16/6/2021, disposed status:Disposed
2	A valid Syariah Assistant Officer is logged in.	2	Filecode: 888-13/3/4, file thickness:30, disposal status:Approved, disposal method: Burned, disposal date: 16/6/2021, disposed status:Disposed
3	Access file disposal page	3	Filecode: 182-19/1/2, file thickness:50, disposal status:Approved, disposal method: Moved to ANM, disposal date: 16/6/2021, disposed status:Disposed
4	The database contains 1 file record that exceeds the maximum storage time.	4	

Test Scenario	Fail to verify the file code of 1 extra file entered to be disposed of, users complete the file disposal procedure for the outdated file from the database.

Step#	Step Details	Expected Results	Actual Results	Pass/Fail/Not executed/Suspended
1	The clerk of record unit clicks on 'Outdated Files'.	The list of files that exceed the maximum storage time is loaded.		
2	The clerk of record unit enter file code of extra file to be disposed of.	File code is successfully entered.		
3	The clerk of record unit clicks on 'Search'.	Display 'File code is not found'		
4	The clerk of record unit clicks on 'Submit'.	The file disposal list is sent to Syariah Assistant Officer.		
5	Syariah Assistant Officer refreshes the file disposal page.	The file disposal list is displayed.		
6	Syariah Assistant Officer enters the thickness of file.	The thickness of file is successfully entered.		
7	Syariah Assistant Officer clicks on 'Email'.	Syariah Assistant Officer's Outlook compose page is loaded.		
8	Syariah Assistant Officer returns to file disposal page.	File disposal page is accessed.		
9	Syariah Assistant Officer clicks on 'Add to Disposal Record'.	The file disposal list is inserted into the disposal record database.		
10	Syariah Assistant Officer selects the file disposal status as "Approved".	File disposal status is successfully selected.		
11	Syariah Assistant Officer selects the file disposal method.	File disposal method is successfully selected.		
12	Syariah Assistant Officer clicks on 'Update'.	File disposal status and method are updated in the database.		
13	The clerk of record unit clicks on 'Approved List'	The approved file disposal list is loaded.		
14	The clerk of record unit selects disposal date.	File disposal date is successfully selected.		
15	The clerk of record unit selects disposal status as "Disposed".	File disposal status is successfully selected.		
16	The clerk of record unit clicks on 'Update'.	File disposal status and date are updated in the database.		

(d) TC002_05_04: Test alternative flow 2 of dispose of file (SD006)

Test Case ID	TC002_05_04	Test Case Description	Test alternatel flow 2 of dispose of file		
Created By	Tan Chiaw Torng	Reviewed By		Version	
QA Tester's Log					
Tester's Name	Tan Chiaw Torng	Date Tested	16-Jun-21	Test Case (Pass/Fail/Not)	

S#	Prerequisites:		S#	Test Data
1	A valid clerk of record unit is logged in.		1	Filecode: 300-13/3/3, file thickness:20&65, disposal status:Approved, disposal method: Moved to ANM, disposal date: 16/6/2021, disposed status:Disposed
2	A valid Syariah Assistant Officer is logged in.		2	Filecode: 205-13/3/4, file thickness:30&68, disposal status:Approved, disposal method: Burned, disposal date: 18/6/2021, disposed status:Disposed
3	Access file disposal page		3	Filecode: 100-19/1/2, file thickness:50&80, disposal status:Approved, disposal method: Moved to ANM, disposal date: 19/6/2021, disposed status:Disposed
4	The database contains 1 file record that exceeds the maximum storage time.		4	

Test Scenario	Thickness of files in the file disposal list are not fully filled by the Syariah Officer Assistant, users complete the file disposal procedure for the outdated file from the database and 1 extra damaged file entered.

Step#	Step Details	Expected Results	Actual Results	Pass/Fail/Not executed/Suspended
1	The clerk of record unit clicks on 'Outdated Files'.	The list of files that exceed the maximum storage time is loaded.		
2	The clerk of record unit enter file code of extra file to be disposed of.	File code is successfully entered.		
3	The clerk of record unit clicks on 'Search'.	Display 'File code is found'		
4	The clerk of record unit clicks on 'Add'.	The details of file with the file code entered is loaded.		
5	The clerk of record unit clicks on 'Submit'.	The file disposal list is sent to Syariah Assistant Officer.		
6	Syariah Assistant Officer refreshes the file disposal page.	The file disposal list is displayed.		
7	Syariah Assistant Officer enters the thickness of only 1 file.	The thickness of 1 file is successfully entered.		
8	Syariah Assistant Officer clicks on 'Email'.	System displays an error message.		
9	Syariah Assistant Officer enters the thickness of the other file.	The thickness of file is successfully entered.		
10	Syariah Assistant Officer clicks on 'Email'.	Syariah Assistant Officer's Outlook compose page is loaded.		
11	Syariah Assistant Officer returns to file disposal page.	File disposal page is accessed.		
12	Syariah Assistant Officer clicks on 'Add to Disposal Record'.	The file disposal list is inserted into the disposal record database.		
13	Syariah Assistant Officer selects the file disposal status as "Approved".	File disposal status is successfully selected.		
14	Syariah Assistant Officer selects the file disposal method.	File disposal method is successfully selected.		
15	Syariah Assistant Officer clicks on 'Update'.	File disposal status and method are updated in the database.		
16	The clerk of record unit clicks on 'Approved List'	The approved file disposal list is loaded.		
17	The clerk of record unit selects disposal date.	File disposal date is successfully selected.		
18	The clerk of record unit selects disposal status as "Disposed".	File disposal status is successfully selected.		
19	The clerk of record unit clicks on 'Update'.	File disposal status and date are updated in the database.		

8.2.6. TC002_06: Test Borrow File (SD007)

(a) TC002_06_01: Test normal flow of Borrow File (SD007)

Test Case ID	TC002_06_01	Test Case Description	Test normal flow of borrow file		
Created By	See Wen Xiang	Reviewed By		Version	
QA Tester's Log					
Tester's Name				Test Case (Pass/Fail/Not)	

S#	Prerequisites:		S#	Test Data
1	User JKSNJ staff is logged in.		1	File code: 100-19/1/2
2	File exists in the system.		2	
3			3	
4			4	

Test Scenario	Verify on entering valid file code to borrow file.	

Step#	Step Details	Expected Results	Actual Results	Pass/Fail/Not executed/Suspended
1	Keys in the file code.	The system creates a borrow ID, updates the file access database, shows the borrow ID to the staff on the borrow file interface.		
2	Clerk of the record unit keys in the borrow ID.	The system displays the borrow record.		

(b) TC002_06_02: Test exception flow 1 of Borrow file (SD007)

Test Case ID	TC002_06_02	Test Case Description	Test exception flow 1 of borrow file	
Created By	See Wen Xiang	Reviewed By	Version	
QA Tester's Log				
Tester's Name		Date Tested	Test Case (Pass/Fail/Not)	

S#	Prerequisites:		S#	Test Data
1	User JKSNJ staff is logged in.		1	File code: 999-99/9/9
2	File is not exist in the system.		2	
3			3	
4			4	

Test Scenario	Fail to verify on entering valid file code to borrow file.		

Step#	Step Details	Expected Results	Actual Results	Pass/Fail/Not executed/Suspended
1	Keys in the file code.	Display File not found message.		
2				
3				

(c) TC002_06_03: Test exception flow 2 of Borrow file (SD007)

Test Case ID	TC002_06_03	Test Case Description	Test exception flow 2 of borrow file	
Created By	See Wen Xiang	Reviewed By	Version	
QA Tester's Log				
Tester's Name		Date Tested	Test Case (Pass/Fail/Not)	

S#	Prerequisites:		S#	Test Data
1	User JKSNJ staff is logged in.		1	File code: 100-19/1/2
2	File is not is not available.		2	
3			3	
4			4	

Test Scenario	Fail to verify on entering valid file code to borrow file due to file is not available.		

Step#	Step Details	Expected Results	Actual Results	Pass/Fail/Not executed/Suspended
1	Keys in the file code.	Displays the last borrower's name and staff ID.		
2	-	Displays File is not available message.		
3				

(d) TC002_06_04: Test exception flow 3 of Borrow file (SD007)

Test Case ID	TC002_06_04	Test Case Description	Test exception flow 3 of borrow file	
Created By	See Wen Xiang	Reviewed By	Version	
QA Tester's Log				
Tester's Name		Date Tested	Test Case (Pass/Fail/Not)	

S#	Prerequisites:		S#	Test Data
1	User JKSNJ staff is logged in.		1	File code: 100-19/1/2
2	Staff does not have the privilege to borrow the file		2	
3			3	
4			4	

Test Scenario	Fail to verify on entering valid file code to borrow file due to does not have the privilege to borrow the file.		

Step#	Step Details	Expected Results	Actual Results	Pass/Fail/Not executed/Suspended
1	Keys in the file code.	Displays the message No right to borrow the file.		
2				
3				

8.2.7. TC002_07: Test View file access record (SD008)

a) TC002_07_01: Test normal flow of view file access record (SD008)

Test Case ID	TC002_07_01	Test Case Description	Test normal flow of view file access record		
Created By	Husny Musharraf	Reviewed By		Version	
QA Tester's Log					
Tester's Name		Date Tested		Test Case (Pass/Fail/Not Executed)	

S #	Prerequisites:	S #	Test Data Requirement
1	User is logged in to the system as clerk of record unit	1	Filecode: 100-19/1/2
2		2	
3		3	
4		4	

Test Scenario	Verify on entering a valid file code, the file access record can be vie

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Key in file code	Display file information		

b) TC002_07_02: Test exception flow of view file access record (SD008)

Test Case ID	TC002_07_02	Test Case Description	Test exception flow of view file access record		
Created By	Husny Musharraf	Reviewed By		Version	
QA Tester's Log					
Tester's Name		Date Tested		Test Case (Pass/Fail/Not Executed)	

S #	Prerequisites:	S #	Test Data Requirement
1	User is logged in to the system as clerk of record unit	1	-
2		2	
3		3	
4		4	

Test Scenario	Fail to verify on the entered file code.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	-	Display error message		

8.2.8. TC002_08: Test Detect the late return of file (SD009)

a) TC002_08_1: Test normal flow of track file location (SD009)

Test Case ID	TC002_08_01	Test Case Description	Test normal flow of detect the late return of file		
Created By	Wan Luqman	Reviewed By		Version	
QA Tester's Log					
Tester's Name		Date Tested		Test Case (Pass/Fail/Not Executed)	

S #	Prerequisites:	S #	Test Data Requirement
1	User is logged in to the system as a clerk of the record unit	1	
2		2	
3		3	
4		4	

Test Scenario	Detecting the late return of file that has been borrowed.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Clicks on the View File Late Return button	List of file exceed the deadline of return displayed		
2	Notifies the borrower of the file	JKSNJ staff receives the notification		

8.2.9. TC002_09: Test Return File (SD010)

(a) TC002_09_01: Test normal flow of Return File (SD010)

Test Case ID	TC002_09_01	Test Case Description	Test normal flow of return file	
Created By	See Wen Xiang	Reviewed By	Version	
QA Tester's Log				
Tester's Name			Test Case (Pass/Fail/Not)	

S#	Prerequisites:		S#	Test Data
1	A JKSNJ staff is taking the file that they borrowed to the counter of JKSNJ.		1	borrow ID: R001
2	User clerk of the record unit is logged in.		2	
3	Borrow record exists in the system		3	
4			4	

Test Scenario	Verify on entering valid borrow ID.		

Step#	Step Details	Expected Results	Actual Results	Pass/Fail/Not executed/Suspended
1	Keys in the borrow ID.	Display the file borrow info.		
2	Click on the Return File button	Borrow status is updated in the file access database.		

(b) TC002_09_02: Test exception flow of Return file (SD010)

Test Case ID	TC002_09_02	Test Case Description	Test exception flow of return file	
Created By	See Wen Xiang	Reviewed By	Version	
QA Tester's Log				
Tester's Name		Date Tested	Test Case (Pass/Fail/Not)	

S#	Prerequisites:		S#	Test Data
1	A JKSNJ staff is taking the file that they borrowed to the counter of JKSNJ.		1	borrow ID: R999
2	User clerk of the record unit is logged in.		2	
3	Borrow record does not exist in the system		3	
4			4	

Test Scenario	Fail to verify on entering valid borrow ID.		

Step#	Step Details	Expected Results	Actual Results	Pass/Fail/Not executed/Suspended
1	Keys in the borrow ID.	Display an error message		

Appendix A: Traceability Matrix

Test Case ID	Use Case ID/ Sequence Diagram ID	Package ID
TC001 for <Login> Subsystem 1. TC001_01	UC010 • SD001	P001
TC002 for <File> Subsystem 1. TC002_01 2. TC002_02 3. TC002_03 4. TC002_04 5. TC002_05 6. TC002_06 7. TC002_07 8. TC002_08 9. TC002_09	• UC001/SD002 • UC002/SD003 • UC003/SD004 • UC004/SD005 • UC005/SD006 • UC006/SD007 • UC007/SD008 • UC008/SD009 • UC009/SD010	P002