SOFTWARE REQUIREMENTS SPECIFICATION

for

Design and Implementation of a Lightweight Education Data Bay Area (E-DBA)

Version 1.2 Approved

Prepared by:
Taian Yu (FULL)   Yolanda Yin (FULL)
Justin Zhang (FULL)   Aaron Liu (FULL)
Steven Luo (FULL)   Younger Yang (FULL)

Group C01

March 12, 2025

Contents

Revision History

Name	Date	Reason For Changes	Version
All Members	13 March	Completed Transition Diagram	1.2
All Members	5 March	Complete Chapter 3 and supplement Sections 4.1 and 6	1.1
All Members	26 February	Create the document	1.0

Chapter 1

Introduction

All external database interfaces mentioned in the document (such as the student information database, the degree thesis database, the bank account database, etc.) should be integrated and invoked according to the interfaces configured by the providers.

1.1 Purpose

This Software Requirements Specification (SRS) document defines the functional, performance, and system constraints for the Educational Data Platform (E-DBA). It serves as a comprehensive reference for developers, testers, project managers, and other stakeholders to ensure a shared understanding of the system's expected behavior and requirements. This document will act as the foundation for system design, development, testing, and validation while ensuring that the implementation aligns with the specified business and technical needs.

1.2 Document Conventions

This document adheres to the following standardized conventions:

1.2.1 Formatting and Typography

The documentation maintains consistent typographic practices, with body text in a uniform font and size. Headings are bolded and organized in a hierarchical structure through section divisions to clarify content organization.

1.2.2 Terminology Guidelines

Key terms are consistently defined and formatted for clarity. For example, **data providers** and **data consumers** refer to user roles that supply data to the system and retrieve data from the system, respectively.

1.2.3 Modular Structure

Content is structured according to the system's functional components and modules, such as user role management, data-sharing services, payment systems, and other operational divisions, ensuring logical separation and ease of navigation.

1.3   Intended Audience and Reading Suggestions

This document is intended for the following audience groups:

- **Project Managers and Developers:** Should begin with the overview section to understand the system's architecture and objectives, followed by the functional and technical requirements sections to gain insights into implementation details.

- **System Administrators:** Recommended to focus on the system configuration and management sections, including administrator privileges, security settings, and maintenance workflows.

- **End Users (e.g., students, faculty, and staff):** Advised to refer directly to the user guide and role-specific tutorials to learn how to access and utilize the E-DBA platform's services effectively.

By following these reading paths, each audience group can efficiently locate relevant information aligned with their roles and responsibilities.

1.4   Project Scope

The development of the E-DBA platform's user interface, backend management system, and associated service modules, alongside the implementation of role-based access control for diverse user categories such as administrators, organizational coordinators, and data providers. Core functionalities include data provisioning and consumption mechanisms covering student identity verification, degree thesis access, and course information sharing, as well as the integration of a multi-payment-method online transaction system. Additionally, the platform will offer data warehousing capabilities to enable dataset export and collaborative sharing. Emphasis is placed on ensuring cross-institutional compatibility, robust data security protocols, and intuitive user experience design to streamline data interoperability and service delivery across participating educational organizations.

1.5   References

Foundational insights are drawn from the Eclipse Dataspace Connector (EDC) framework documentation, including its technical specifications and implementation guidelines for secure data exchange. Architectural principles align with the reference models and interoperability standards published by the International Data Spaces Association (IDSA). Additionally, compliance considerations incorporate relevant educational data governance frameworks, cross-institutional data-sharing policies, and sector-specific regulatory requirements. Technical implementation details are informed by industry-standard development protocols, API integration best practices, and security benchmarks for distributed systems, ensuring alignment with modern data infrastructure paradigms.

Chapter 2

Overall Description
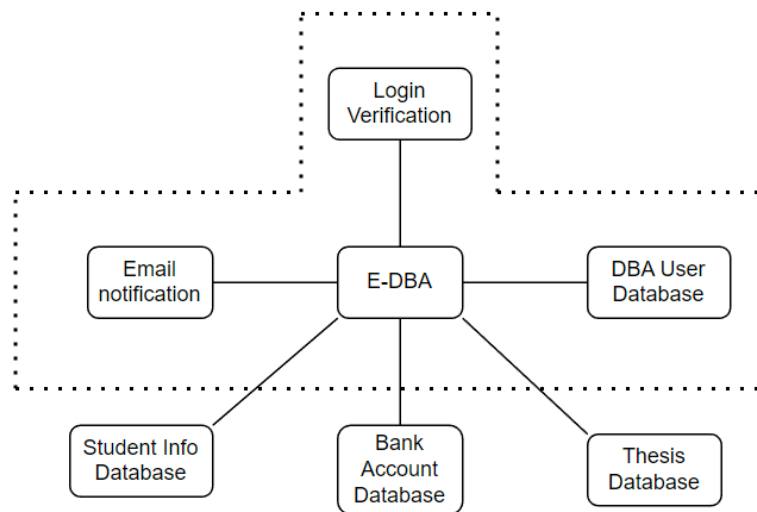
2.1   Product Perspective

Figure 2.1: Product Perspective
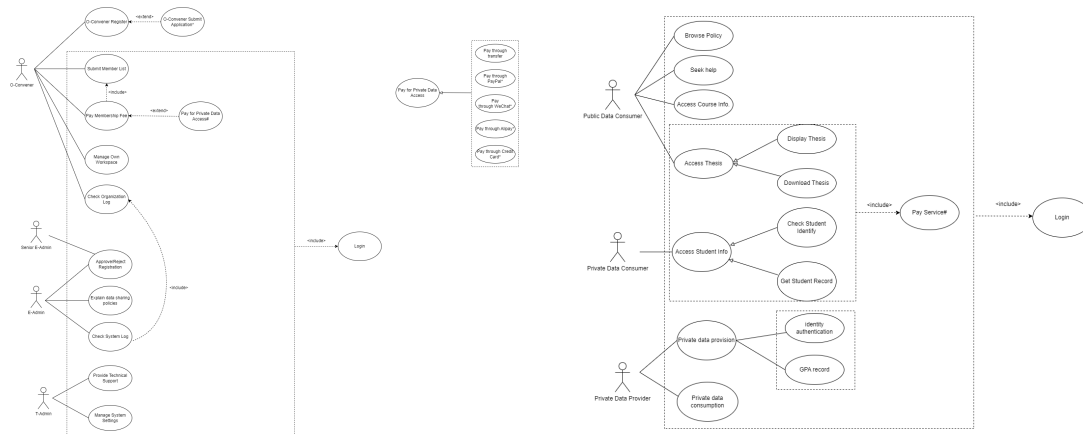
2.2 Product Functions

Figure 2.2: Product Function

2.2.1 Basic Scenario

The registration process initiates when the **O-Convener** completes the registration form (Step 1), providing contact information, institutional affiliation, and required documentation. Subsequently, the **E-Admin** or **Senior E-Admin** conducts a comprehensive review (Step 2), including material verification, authenticity validation, and background screening. Upon approval, system validation occurs (Step 3), generating a unique convener ID and activating service access privileges. Finalization (Step 4) triggers automated credential distribution and system onboarding initiation.

2.2.2 Alternative Scenario

Registration Rejection: In cases of administrative rejection, the system issues an automated notice detailing rejection rationale and appeal procedures, while preserving records for 30 days per GDPR compliance.
Incomplete Submission: System-level validation identifies missing elements through automated checks, flagging incomplete fields and initiating follow-up protocols requiring convener remediation before process continuation.

2.3 User Classes and Characteristics

- **Admin**:

  - **T-admin**: Responsible for overseeing and managing system functionality. Has access to the full suite of administrative controls.

  - **E-admin**: Handles user and system management, including configuration of system features and settings.

  - **Senior E-admin**: Manages both system and user configurations at an advanced level.

Characteristics: Must be responsive and make decisions to resolve user issues efficiently.

- **O-Convener**: Focuses on managing user lists, workspace organization, and service configurations for the system. **Characteristics:** Must manage user access and configure services.

- **Data Provider and Data Consumer**:

  - **Data Provider**: Supplies the system with data for processing and consumption.

  - **Data Consumer**: Engages with the system to access and use provided data, including processing payments for data consumption.

  Characteristics: Must be skilled in operating systems and data sorting.

- **End Users**: Individuals who interact with the system on a more basic level to access or utilize specific functionalities. End users are the primary consumers of the system's services, with access tailored to their specific needs. **Characteristics:** Can basically use the system through instruction.

2.4   Operating Environment

We will use windows or linux as our poerating environment. We will also use database management system like MySQL to store our data

2.5   Design and Implementation Constraints

- The system must adhere to a high level of security to protect sensitive data and user privacy.

- The system must be capable of connecting to and interacting with a database management system to store and retrieve data efficiently.

2.6   User Documentation

- **User Manuals**: Comprehensive guides tailored to each user class (admin, end user, etc.), providing step-by-step instructions for system setup and usage.

- **On-line Help**: An integrated help system within the software that offers real-time support to solve user problems.

- **Tutorials**: Both video and text-based tutorials designed to help users understand how to operate the system effectively.

- **Release Notes**: A summary of updates and changes made in each version of the system to inform users of new features, fixes, or enhancements.

2.7 Assumptions and Dependencies

Assumptions:

- The stability and reliability of third-party services integrated with the system.

Dependencies:

- The project will rely on a database management system for development and data storage.

Chapter 3

System Features

3.1   Login

3.1.1   Description

Login function for all users: T-admin, senior E-admin, E-admin, O-convener, Public Data provider, Private data consumer, Public Data consumer. The login page can be accessed from the main page of the software. Different types of users have different privileges. Medium priority.

3.1.2   State diagrams / UI
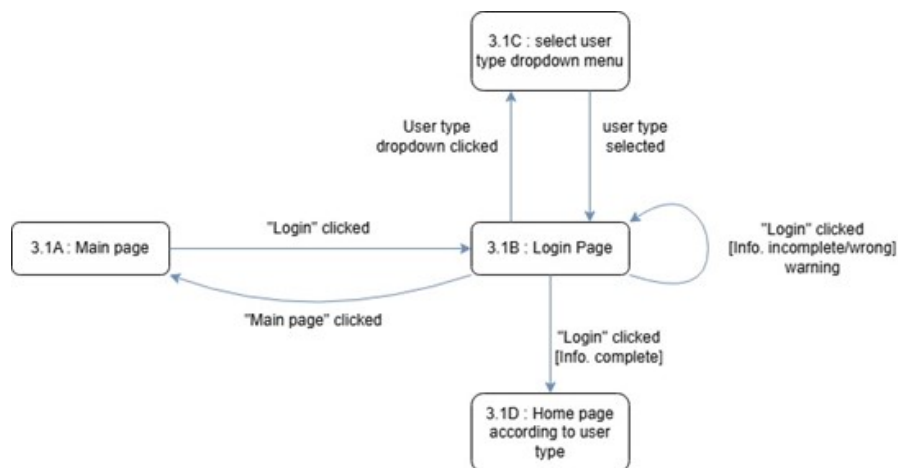

Figure 3.1: State Diagrams

Note: Please refer to document in 4.1 for UI diagrams.

Scenario 1: Login Success

- Assume users are on the main page.
- Users click the login button on the main page.
- System displays the login page.
- Users select user type and enter credentials.
- Users click the login button.
- System directs users to the home page.

Scenario 2: Login Fail

- Assume users are on the main page.

- Users click the login button on the main page.

- System displays the login page.

- Users select user type and enter credentials.

- Users click the login button.

- System displays a login fail message.

- System returns to the login page.

3.1.3 Functional Requirements

TBD

3.2 O-Convener

3.2.1 Description

O-convener is specific to the organization that provides the managment of users and access rights. The O-convener can manage their own private workspace, maintain and grant new users access permission from their organization to use the E-DBA and as well as check the organization log. The addition of users is done either through system UI or from Excel files for batch input. Medium priority

3.2.2 State diagrams / UI
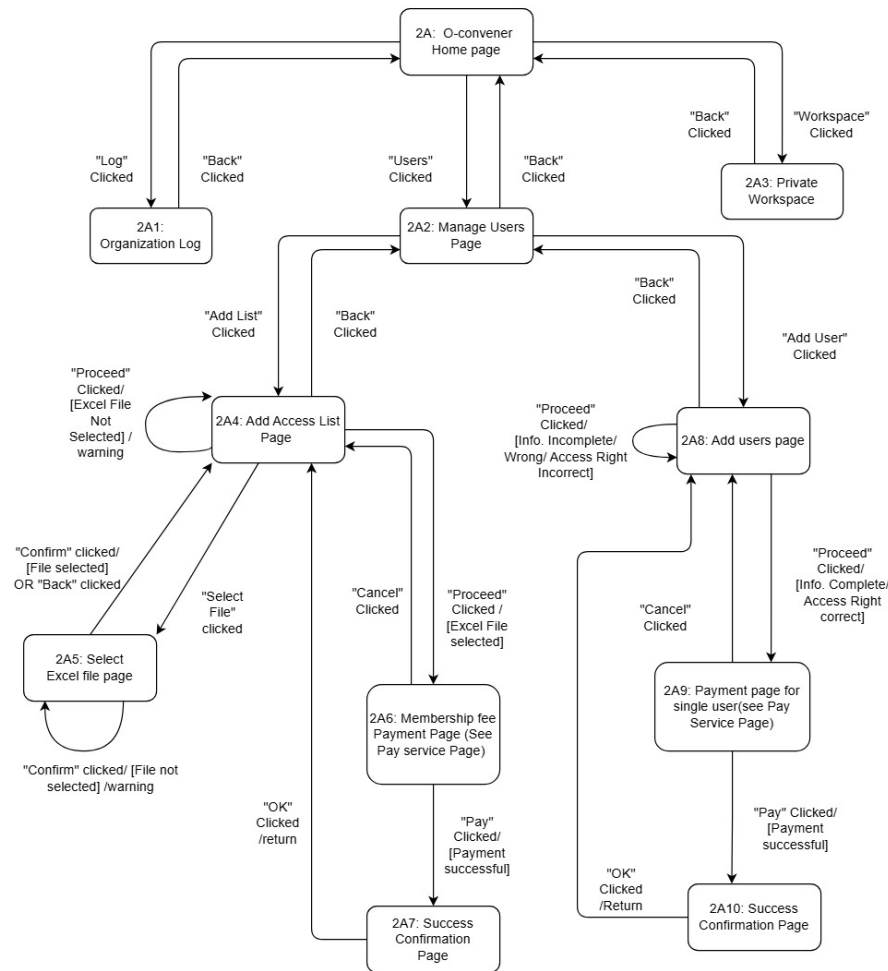

Figure 3.2: State Diagrams

Scenario 1: Manage Own Private Workspace

- From O-Convener home page.
- Enter Private Workspace.

Scenario 2: Manage Users – Add Users Through System UI

- From O-Convener home page.
- Enter Manage User page.
- Click "Add Users".
- Enter credentials and access rights.
- Save choice and proceed to payment.
- Finish and return to Manage User page.

Scenario 3: Manage Users – Add Users Through Excel Batch File

- From O-Convener home page.

- Enter Manage User page.

- Click "Add Access List".

- Input Excel file.

- Save choice and proceed to payment.

- Finish and return to Manage User page.

Scenario 4: Check Organization Log

- From O-Convener home page.

- Enter Check Log page.

- Return to Home page.

3.3   E-Admin & Senior E-Admin

3.3.1   Description

In the system, E-Admin could check system log through inspecting the system log status of each user, while they could explain data sharing policy through write the explanation for selected data sharing policy. And they could also approve or reject application through checking the content of application and deciding whether approve the application or reject. However, Senior E-Admin could only approve or reject application.
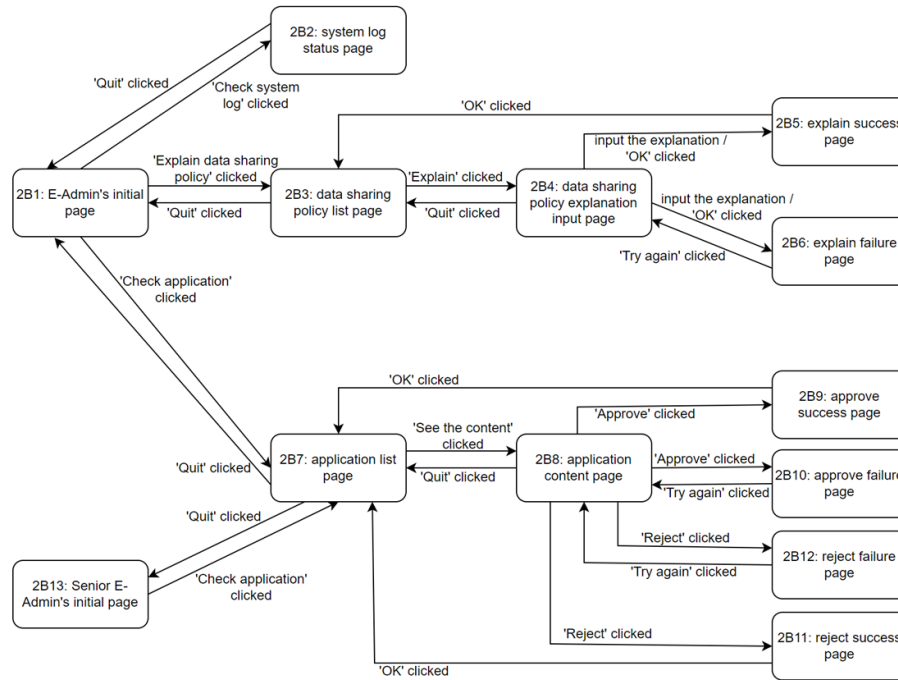
3.3.2 Stimulus/Response Sequence

Figure 3.3: State Diagrams

Basic Scenario

- The E-Admin/Senior E-Admin logs in.
- The system displays the E-Admin's/Senior E-Admin's initial page.
- The E-Admin clicks 'Check System Log'.
- The system displays the system log status page.
- The E-Admin clicks 'Explain Data Sharing Policy'.
- The system displays the data sharing policy list page.
- The E-Admin clicks 'Explain'.
- The system displays the data sharing policy explanation input page.
- The E-Admin inputs the explanation and clicks 'OK'.
- The system displays the explanation success page if successful, otherwise, displays the explanation failure page.
- The E-Admin clicks 'OK' or 'Try Again'.
- The system returns to the data sharing policy list page or the data sharing policy explanation input page.
- The E-Admin clicks 'Quit'.
- The system returns to the E-Admin's initial page.

15

- The Senior E-Admin/E-Admin clicks 'Check Application'.

- The system displays the application list page.

- The Senior E-Admin/E-Admin clicks 'See the Content'.

- The system displays the application content page.

- The Senior E-Admin/E-Admin clicks 'Approve' or 'Reject'.

- The system displays the approve success page or reject success page if successful, otherwise, displays the approve failure page or reject failure page.

- The Senior E-Admin/E-Admin clicks 'OK' or 'Try Again'.

- The system returns to the application list page or the application content page.

- The Senior E-Admin/E-Admin clicks 'Quit'.

- The system returns to the E-Admin's/Senior E-Admin's initial page.

3.4   T-Admin

3.4.1   Description

The user navigates through the T-Admin system to manage E-Admin accounts, respond to help requests, and manage vault applications. Actions include editing and updating account information, answering help requests, and navigating back to previous menus.
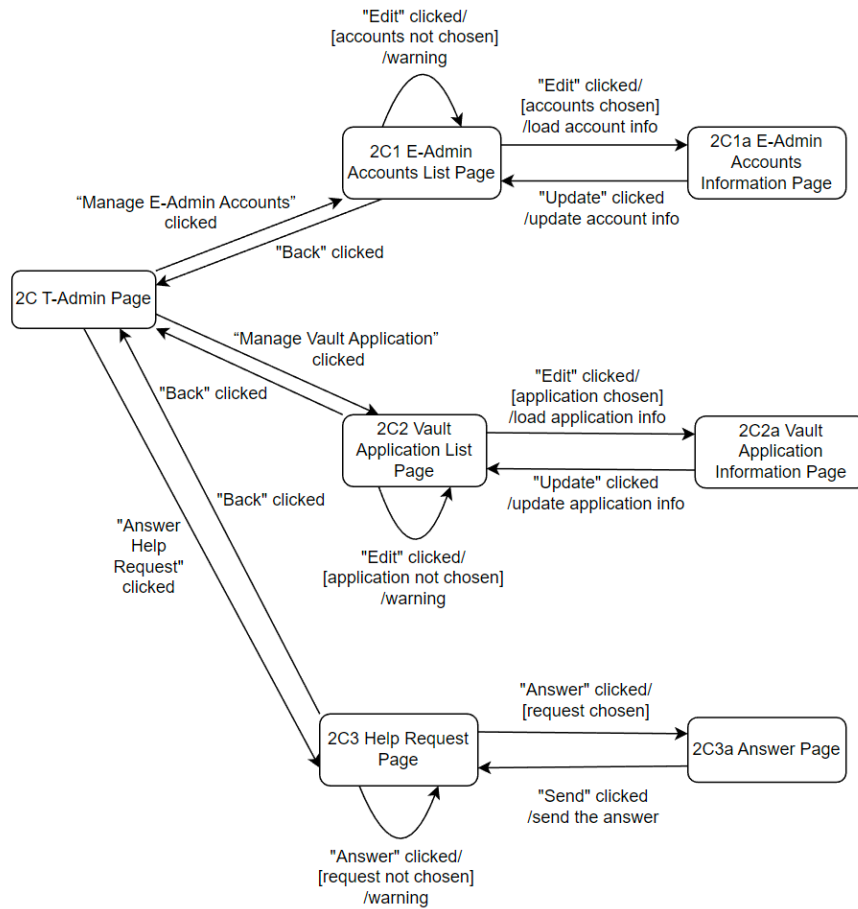
3.4.2 Stimulus/Response Sequence

Figure 3.4: State Diagrams

- The user clicks "Manage E-Admin Accounts". → The system displays the E-Admin accounts list.

- The user clicks "Edit". → The system shows E-Admin account information.

- The user clicks "Update". → The system goes back to the E-Admin accounts list page.

- The user clicks "Back". → The system goes back to the manage page.

- The user clicks "Answer Help Request". → The system shows the help request list.

- The user clicks "Answer". → The system shows the answer page.

- The user clicks "Send". → The system goes back to the help request list page.

- The user clicks "Back". → The system goes back to the manage page.

- The user clicks "Manage Vault Application". → The system displays the vault application list.

- The user clicks "Edit". → The system shows vault application information.
- The user clicks "Update". → The system goes back to the vault application list page.
- The user clicks "Back". → The system goes back to the manage page.
- The user clicks "Back" again. → The system goes back to the T-Admin page.

3.5 Public Data Consumer

3.5.1 Description

The Public Data Consumer can browse policies, seek help, access course information, and access thesis. They can select policies to view details, submit help requests and check responses, choose courses to access information, and download theses after payment. The system provides success, failure, or warning feedback based on actions and allows navigation between pages.
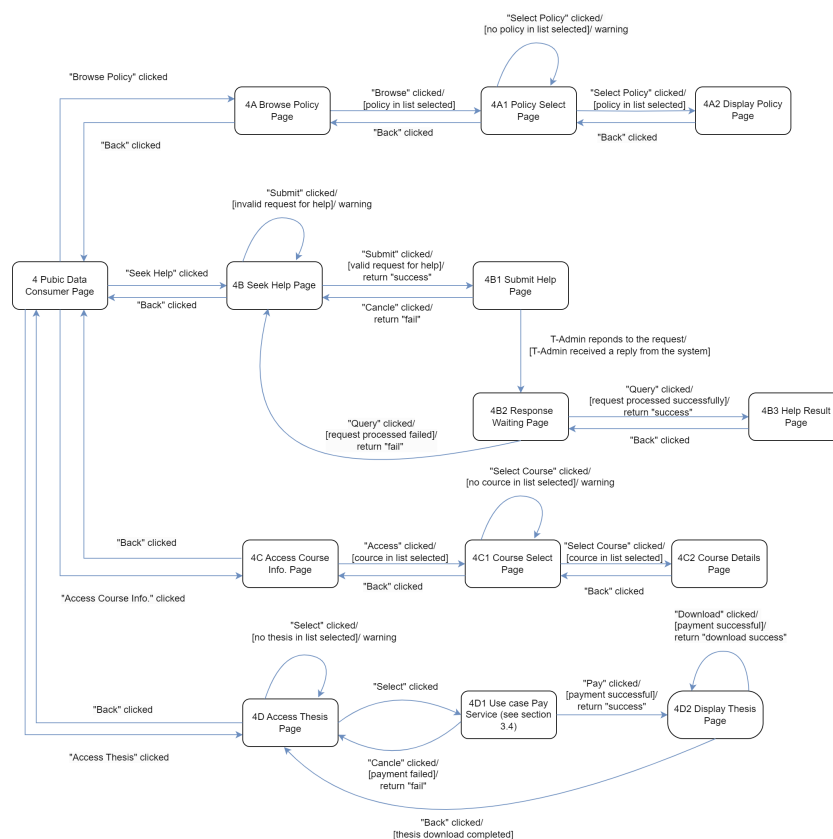
3.5.2 State diagrams / UI

Figure 3.5: State Diagrams

Note: Please refer to document in 4.1 for UI diagrams.

Basic Scenario

- Browse Policy
    1. User browses the policy list.
    2. Views policy details.
    3. Returns to the list at any time.
- Seek Help
    1. User submits a help request.
    2. Waits for T-Admin's reply.
    3. Ends the process after confirming the reply.
- Access Course Info
    1. User browses the course list.
    2. Views course details.
    3. Returns to the list at any time.
- Access Thesis
    1. Selects Thesis.
    2. Displays Thesis Content.
    3. Requests Download.
    4. Downloads Thesis.

3.5.3 Functional Requirements

TBD

3.6 Pay Service

3.6.1 Description

In the Pay Service use case, the actor could pay by PayPal, Credit Card, AliPay, WeChat or by money transfer through scan the QR code on the corresponding page, however, the system would initially only allow money transfer, and if the actor want to pay by other methods, he/she could go to the next page to select the payment method he/she want and pay the money.
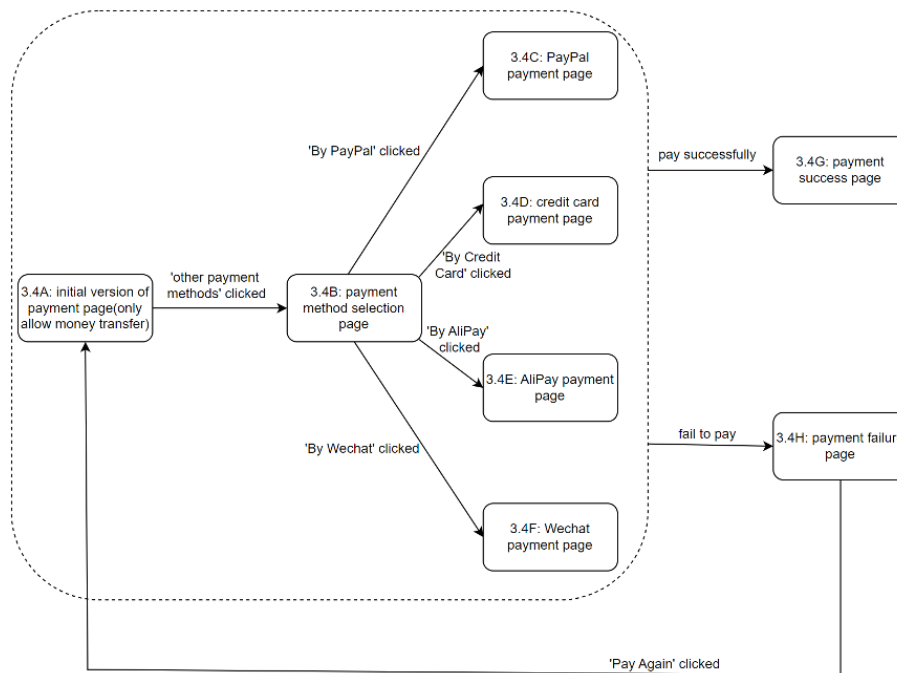
3.6.2 State diagrams / UI

Figure 3.6: State Diagrams

Note: Please refer to document in 4.1 for UI diagrams.

Basic Scenario

The actor scan the QR code on the initial version of payment page and pay the money.

The system display the payment success page if the actor paid successfully, otherwise, the system display payment failure page The actor click 'other payment methods' button.

The system display the payment method selection page.

The actor click 'By PayPal'/'By Credit Card'/'By Alipay'/'By Wechat' button.

The system display PayPal payment page/credit card payment page/AliPay payment page/Wechat payment page.

The actor scan the QR code on the PayPal payment page/credit card payment page/AliPay payment page/Wechat payment page and pay the money.

The system display the payment success page if the actor paid successfully, otherwise, the system display payment failure page.

3.6.3 Functional Requirements

TBD

3.7 Data Provider

3.7.1 Description

The data provider can add info to a new course or edit the info of an existing course. He can also set an interface for thesis sharing services by set corresponding links according to different services.

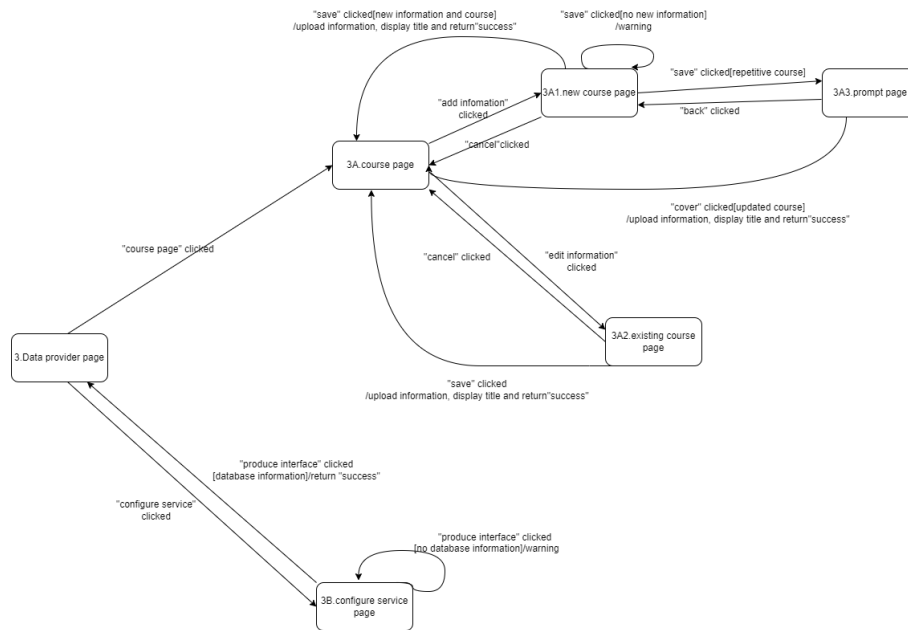### 3.7.2 Stimulus/Response Sequence

Figure 3.7: State Diagrams

Basic Scenario

- The Private Data Provider clicks "add information/edit information".
- The system displays the information page.
- The Private Data Provider adds/edits information.
- The system saves the information and displays it.
- The Private Data Provider then clicks "configure service".
- The system displays the configure service page.
- The Private Data Provider adds a database and clicks "produce interface".
- The system produces the interface.

Alternative Scenario

- The Private Data Provider clicks "add information/edit information".
- The system displays the information page.
- The Private Data Provider does not add new information or adds a repetitive

course.

- If no new course is added, the system returns to the new course page.
- If a repetitive course is added, the system will go to the prompt page to check whether the author wants to overwrite the original course.
- The Private Data Provider then clicks "configure service".
- The system displays the configure service page.
- The Private Data Provider does not add a database and clicks "produce interface".
- The system gives a warning and returns to the configure service page.

3.8 Private Data Consumer

3.8.1 Description

The user accesses the system and navigates through various options to view and manage student information. The user can check student details, proceed to payment services (low priority), and return to previous menus as needed.
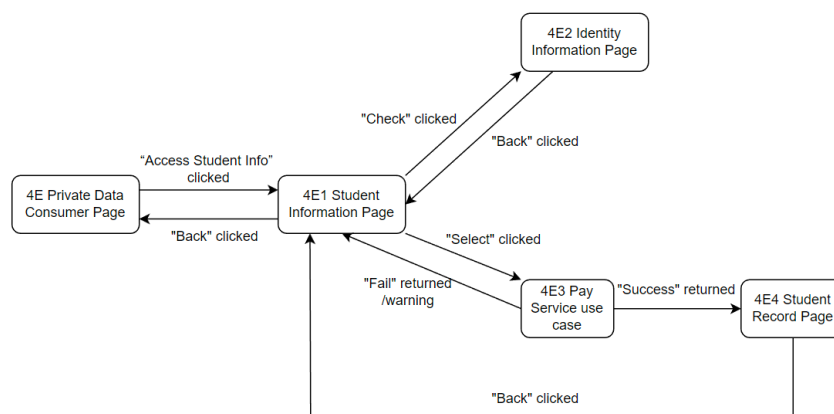
3.8.2 Stimulus/Response Sequences

Figure 3.8: State Diagrams

Basic Scenario

- The user clicks "Access Student Info". → The system displays student information.
- The user clicks "Check". → The system shows student identity information.
- The user clicks "Back". → The system goes back to the student information

page.

- The user clicks "Select". → The system turns into the payment service.
- The user pays successfully. → The system shows the student record.
- The user clicks "Back". → The system goes back to the information page.
- The user clicks "Back" again. → The system goes back to the private data consumer page.

Chapter 4

External Interface Requirements

4.1 User Interfaces

Please Visit the document User_Interface.pdf

4.2 Hardware Interfaces

TBD

4.3 Software Interfaces

1 Checking account validity and executing transfer processes through payment transfer methods (Transfer) (Paypal, WeChat, Alipay, Credit is TBD)

4.4 Communications Interfaces

1 Email addresses and electronic forms, including proof documents required by the O-convener for registration.

2 Excel files required by a registered organization to authenticate student identity through batch input.

Chapter 5

Other Nonfunctional Requirements

5.1 Performance Requirements

- **Response Time**: To ensure a good user experience, the system should minimize delays, especially in critical operations such as real-time data access and identity verification.

- **Concurrency Handling**: During peak periods, such as the start of the school term or thesis submission deadlines, the system may experience a large number of concurrent users. The system must have strong concurrency handling capabilities to manage multiple user requests simultaneously.

- **Scalability**: The system should be scalable to accommodate future growth. New features should be easily added, and more users should be supported without affecting existing functionalities.

- **Data Consistency and Reliability**: For key operations such as payments and identity verification, data loss or inconsistencies must be prevented to avoid serious consequences. The system should have robust transaction management and error recovery mechanisms.

- **Resource Consumption**: Developers should optimize the code and architecture to ensure efficient use of resources, avoiding excessive consumption of server resources during operation.

5.2 Safety Requirements

- **Data Breach Risk**: The system must prevent privacy violations and identity theft by encrypting data transmission and implementing strict access control mechanisms. Unauthorized access must be prohibited, and all user activities should be monitored.

- **Payment Security Risk**: The system should mitigate financial transaction risks by using multi-factor authentication and adopting certified secure payment gateways, ensuring the safety and reliability of payment channels.

- **System Stability Risk**: To prevent service interruptions, data loss, or corruption due to system instability, important data must be backed up regularly, and performance metrics should be continuously monitored.

5.3   Security Requirements

- **Data Sharing Security**: The system must ensure that data is shared securely and transparently between different organizations. Data transmission must be encrypted, and only authorized parties should be able to access specific data.

- **Online Payment Security**: All payment operations must be secure to prevent financial fraud. Strict security measures must be taken to protect users' payment information, especially when handling personal financial details.

- **Non-tamperable Log Records**: System logs should be automatically categorized based on user activity and be non-modifiable to ensure the integrity of log data.

5.4   Software Quality Attributes

5.4.1   For Users

- **Usability**: The system's user interface should be easy to use for individuals in different roles, with detailed guidance to help them get started quickly.

- **Reliability**: The system should run stably, minimizing the risk of failures and ensuring a quick recovery if issues arise.

- **Efficiency**: The system should respond quickly to user actions, whether browsing public information or querying private data.

5.4.2   For Developers

- **Maintainability**: Code structure should be clear, following best practices, to ensure easier updates and iterations.

- **Compatibility**: The system should function correctly across various operating systems and browsers.

- **Flexibility**: The system should have flexible service interfaces for easy integration with external systems and support customizable configurations based on organizational needs.

- **Testability**: A comprehensive testing framework should be established, covering unit tests, integration tests, stress tests, and others.

- **Compliance**: The system should adhere to relevant laws and regulations, particularly those relating to personal data protection, following strict privacy guidelines.

5.5 Business Rules

- **Roles and Permissions**: Different users and organizations have specific access rights and permissions. Only authorized users can perform critical operations like identity verification and payment processing.

- **Data Privacy**: The system must ensure that sensitive personal information is protected according to privacy regulations, restricting access to authorized personnel only.

Chapter 6

Other Requirements

6.1 Appendix A: Glossary

- **Reliability**: The ability of the database software to perform its required functions under stated conditions for a specified period of time without failure.
- **Efficiency**: The ability of the database software to perform tasks quickly and with minimal resource consumption.
- **Scalability**: The ability of the database software to handle increasing amounts of data and users.
- **Maintainability**: The ability of the database software to be modified, updated, or repaired over time.
- **Compatibility**: The ability of the database software to work the same as other systems, platforms, or tools.
- **Flexibility**: The ability of the database software to adapt to changing requirements or environments.
- **Testability**: The ability of the database software to be tested to ensure it meets functional and non-functional requirements.

6.2 Appendix B: Analysis Models

TBD

6.3 Appendix C: Issues List

- 4.2 TBD
- 6.2 TBD