

SDD-xx

SOFTWARE DESIGN DOCUMENT

<Software Name>

Client:

<User>


Prepared by:

<Group number & Members>

Bachelor of Informatics

Faculty of Informatics

Jl. Telekomunikasi 1, Dayeuhkolot Bandung

	Bachelor of Informatics Telkom University	Document Number		Number of Pages
		<i>SDD-xx</i>		<#>
		Revision	<i><revision number></i>	<i>Date: <date></i>

List of Revisions

Revision	Description
A	
B	
C	
D	
E	
F	
G	

DATE INDEX	-	A	B	C	D	E	F	G
Revised by								
Checked by								
Approved by								

List of Revised Pages

Page	Revision	Page	Revision

Table of Content

Table of Content.....	4
List of Tables	5
List of Figures	6
1. Introduction.....	7
1.1 Aims.....	7
1.2 Problem scope.....	7
1.3 Terms and Definitions	7
1.4 References	7
2 General Design (High-Level)	8
2.1 Software Implementation Environment Plan	9
2.2 Software Architecture Description	9
3 Detailed Design (Low-Level).....	11
3.1 Use Case Realization.....	11
3.1.1 Use Case #1 <name of use case 1>.....	11
3.1.1.1 Use Case Scenario #1 <name of use case 1>.....	11
3.1.1.2 UI Design and UI Object Description #1 <name of use case 1>.....	11
3.1.1.3 Object Identification and Class Type #1 <name of use case 1>.....	12
3.1.1.4 Sequence Diagram #1 <name of use case 1>.....	13
3.1.2 Use Case #2 <name of use case 2>.....	13
3.1.2.1 Use Case Scenario #2 <name of use case 2>.....	13
3.1.2.2 UI Design dan UI Object Description #2 <name of use case 2>.....	14
3.1.2.3 New Object Identification & Class Type #2 <name of use case 2>.....	15
3.1.2.4 Sequence Diagram #2 <name of use case 2>.....	16
3.2 Overall Class Diagram	23
3.3 Detailed Class Design.....	24
3.4 Algorithm and/or Query Design.....	25
4 Requirement Traceability Matrix.....	26

After the Table of Contents, you may include a List of Tables and a List of Figures

List of Tables

List of Figures

1. Introduction

1.1 Aims

This document outlines the use case scenarios for a software system designed to enhance the user experience within a virtual library environment. The software is aimed at providing a seamless and efficient platform for users, administrators, and members to interact with library resources, engage in educational activities, and manage various aspects of the library ecosystem. The purpose of this document is to detail specific use cases, their primary and alternative flows, as well as pre- and post-conditions.

1.2 Problem scope

The software whose requirements are specified in this document is a comprehensive Library Management System (LMS). This SRS outlines the software requirements for the entire Library Management System, encompassing various modules and functionalities that collectively contribute to an efficient and user-friendly virtual library environment. The system is designed to cater to the needs of multiple user roles, including actors such as students, lecturers, and administrators.

1.3 Terms and Definitions

User Roles:

User: Individuals engaging with the virtual library system.

Administrator: Responsible for system management.

Library Management System (LMS): Comprehensive software for cataloging, circulation, user management, and reporting within the virtual library.

Use Case Terminology:

Primary Flow: Main sequence of steps in a scenario.

Alternative Flow: Deviation to address specific conditions.

Pre-conditions: Necessary conditions before executing a use case.

Post-conditions: Desirable outcomes following successful use case execution.

1.4 References

<https://openlibrary.telkomuniversity.ac.id/>

<https://chromewebstore.google.com/detail/google-dictionary-by-goog/mgijmajocgfcbeboacabfgobmjgjcja>

<https://www.oxfordlearnersdictionaries.com/definition/english>

<https://languages.oup.com/google-dictionary-en/>

2 General Design (High-Level)

2.1 Software Implementation Plan

The software architecture for the "Closed Library" mobile application will follow a typical three-tier architecture, consisting of the presentation layer, the application layer, and the data layer.

a. Presentation Layer

- The front-end will be developed using a mobile application framework, such as React Native or Flutter, for cross-platform compatibility.
- The user interface will include screens for user login, book browsing and searching, reservation requests, and library updates.

b. Application Layer

- The application layer will handle business logic and be an intermediary between the presentation and data layers.
- Key components include user authentication, reservation processing, and library inventory management.
- This layer will communicate with the data layer to retrieve and update information about books, users, and reservations.

c. Data Layer

- The data layer will manage the storage and retrieval of data related to users, books, reservations, and library resources.
- A relational database, such as MySQL or PostgreSQL, will be employed to store structured data efficiently.
- Object-relational mapping (ORM) tools may be used to interact with the database.

d. Communication

- RESTful APIs or GraphQL will facilitate communication between the application and data layers.
- Secure protocols will be implemented to ensure the confidentiality and integrity of data during transmission.

e. Security

- Security measures, including encryption for sensitive data, secure authentication mechanisms, and authorization controls, will be implemented to protect user information and reservation details.

f. Scalability

- The architecture will be designed with scalability in mind to accommodate potential growth in user base and library collections.
- Load balancing and caching strategies may be employed to optimize performance.

g. Integration

- Integration points with external services may include notification systems for reservation updates or library closure alerts.

2.2 Software Architecture Description

Fill in with a list of modules. The list of modules can be in the form of the following table:

No	Component Name	Description
1	User Authentication	Allows users to securely register, log in, and access their library accounts. Manages user profiles and ensures data privacy.
2	Book Catalog	Provides a comprehensive online catalog for users to browse, search, and filter library books. Includes detailed book information and availability status.
3	Reservation System	Enables users to reserve books online for pickup. Manages reservation status, waitlists, and notifications.
4	Digital Content Management	Manages access to the preview books. Integrates with digital content providers and ensures secure access controls.
5	Library Inventory	Interacts with the database to store and retrieve information about books, users, reservations, and library resources. Ensures accurate inventory tracking and efficient data management.
6	Security	Implements robust security measures to protect user data and library resources. Includes encryption, authentication, authorization, and vulnerability management.
7	Scalability	Designs architecture for system scalability to handle increasing user base and expanding digital collections. Employs load balancing and caching strategies for optimal performance.
8	Integration	Integrates with external services such as digital content providers, notification systems, and library catalog aggregators. Enables seamless communication and data exchange.

3 Detailed Design (Low-Level)

3.1 Use Case Realization

Contains the USE CASE TABLE as follows:

No	Use Case Name	Use Case Description
#1	Preview Page	Allows the user to preview the book
#2	Login Page	Allows the user to login by using their designated email and password
#3	Register Page	Allows the user to register new accounts
#4	Borrow Page	Allows the user to borrow the book after previewing the book
#5	Sign Out Page	Allow the user to sign out

3.1.1 Use Case #1 <Preview Book> Lukman Alfaris Hidayatullah


3.1.1.1 Use Case Scenario #1 <Preview Book> - Lukman Alfaris H

Use Case Scenario #1 :


- i. Pre-Condition
The user has successfully login and want to previewing a book before borrowing it
- ii. Use Case Description
 - a. Primary Flow
 - i. The user choose the book they want to preview
 - ii. The system display the informations of the book
 - b. Alternative Flow
- iii. Post-Condition
The system has display the information of the book user choose


3.1.1.2 UI Design and UI Object Description #1 Preview Book – Lukman alfaris H

Fill in this section with the initial version of the interface prototype for each use case.
Afterwards, for each interface/screen, provide detailed specifications.



[Home](#)
[Book](#)
[Genre](#)
[Authors](#)






Calculus for Pre-School

Lukman Alfaris
2023


"In the vibrant city of Arcadia, a reluctant hero emerges as an unexpected threat looms. Our protagonist, a young and ordinary individual, discovers hidden powers that thrust them into a world of mystique and danger. As they navigate through a tapestry of alliances and betrayals, the fate of Arcadia hangs in the balance. Fueled by friendship, love, and a sense of duty, our hero faces epic challenges, unraveling secrets that redefine their understanding of reality. With stunning visuals and a gripping narrative, this comic takes readers on an exhilarating journey through a realm where every choice carries profound consequences."

★★★★★

Borrow



Author Info



Lukman Alfaris

"In the vibrant city of Arcadia, a reluctant hero emerges as an unexpected threat looms. Our protagonist, a young and ordinary individual, discovers hidden powers that thrust them into a world of mystique and danger. As they navigate through a tapestry of alliances and betrayals, the fate of Arcadia hangs in the balance. Fueled by friendship, love, and a sense of duty, our hero faces epic challenges, unraveling secrets that redefine their understanding of reality. With stunning visuals and a gripping narrative, this comic takes readers on an exhilarating journey through a realm where every choice carries profound consequences."

Table of UI Object Descriptions

SCREEN ID	SCREEN NAME	DESCRIPTION
	Preview book Page	Page tat will be landed if the user click the preview page

FOR EACH INTERFACE/PAGE, create detailed specifications.

Interface XXX: {fill in with screen number or interface design image number}

PreviewBook Page

Object_ID	TYPE	LABEL*	Description**
Button 1	Button	Home	If clicked, it will go to home page
Button 2	Button	Book	If clicked, it will go to book page
Button 3	Button	Genre	If clicked, it will go to genre page
Button 4	Button	Authors	If clicked, it will go to author page
Botton 5	Button	Borrow	If clicked, it will active the function BorrowBook
Text Box 1	Text Box	Search Book	If clicked, it will allows the user to search book

3.1.1.3 Object Identification and Class Type #1 <Preview Book> - Lukman Alfaris H

Identify objects related to this use case.

Classes in the design phase are different from classes in the analysis phase.

Use the table below:

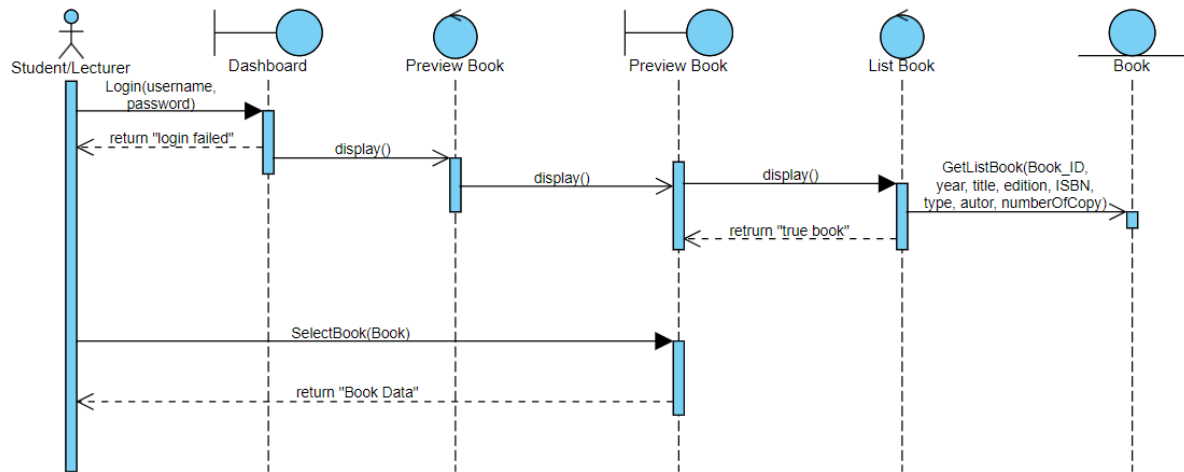
TABEL OF DESIGN OBJECTS

No	New Object Name	Class Type
1	Student	Actor
2	Lecturer	Actor
3	Dashboard	Boundary
4	Preview Book	Controller

5	Preview Book	Boundary
6	List Book	Controller
7	Book	Entity

*Class types such as Boundary (Interface), Entity (Database), Controller

3.1.1.4 Sequence Diagram #1 <Preview Book> - Lukman Alfaris H



3.1.2 Use Case #2 <Login> Andi Ziyadh Ghafa Ramadhan

- i. Pre-Condition : The user has already created an account, and their data is stored in the system.
- ii. Use Case Description
 - a. Primary Flow :
 1. The user navigates to the landing page of the login section on the website.
 2. The user enters their username or email and password on the login page.
 3. The system validates the entered credentials against the stored data in the database.
 4. If the credentials are found in the database, the system grants access to the website and allows the user to proceed to the dashboard.
 5. The user is redirected to their personalized dashboard.
 - b. Alternative Flow :
 1. Wrong Credentials :
 - A. If the inputs are not found in the database, the system denies access to the website thus returning the user back to the login page.
 - B. The user can re-enter their credentials back to the designated space.
 2. Forgot password :
 - A. If the user wants to reset their password they can click the designated button, and the system will provide some recovery options.
 - B. After choosing a recovery option, the system will send an instruction and provide the needs to reset the user's password.
 - C. The user follows the instructions to set a new password.
 - D. After following the instruction, the system will replace the new password with the old password
- iii. Post-Condition :
 - If the user inputs the correct credentials, the system will display the home screen or dashboard to the website
 - The user can use the dashboard however their wish included with the functionalities and feats.
 - The system employs a stateful session model, preserving website data and user interactions across multiple pages and actions until the session is terminated by the user.

3.1.2.1 UI Design dan UI Object Description #2 Login – Andi Ziyadh Ghafa R

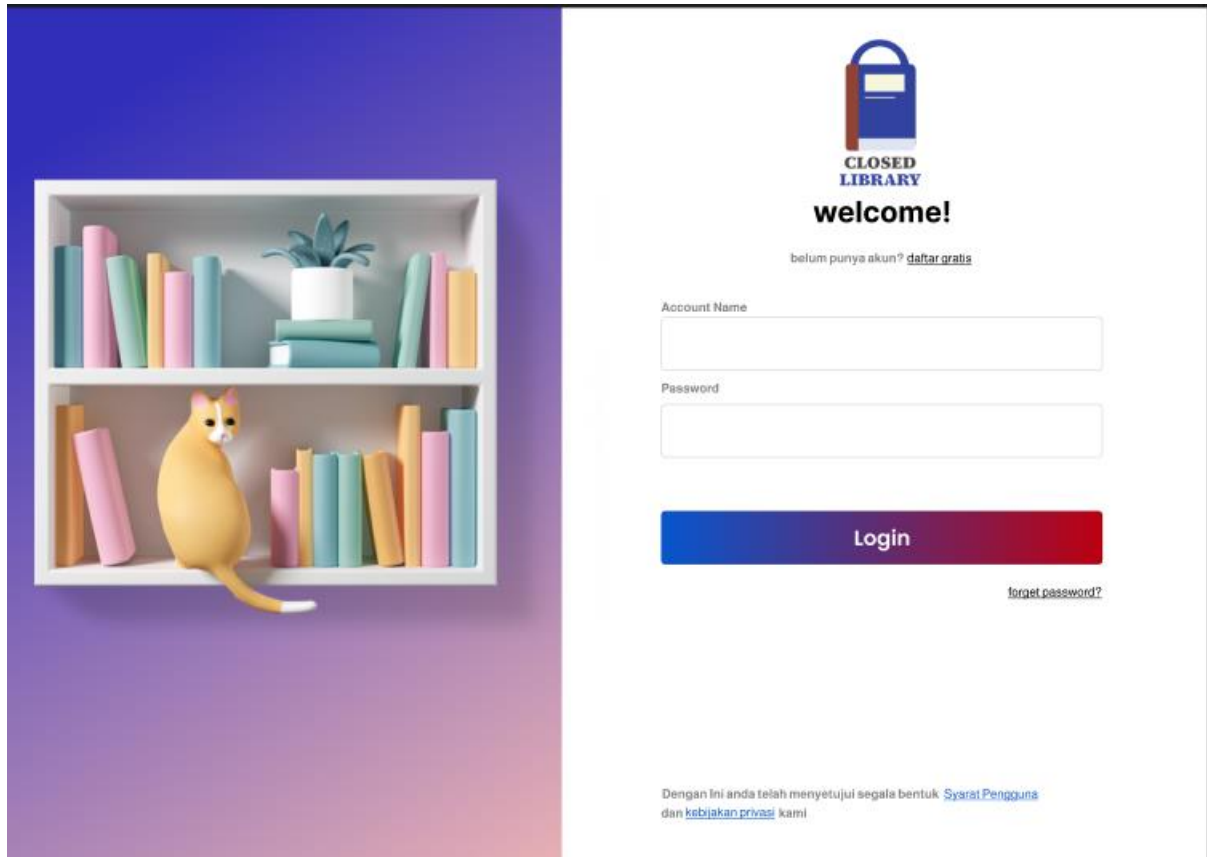


Table of UI Object Descriptions

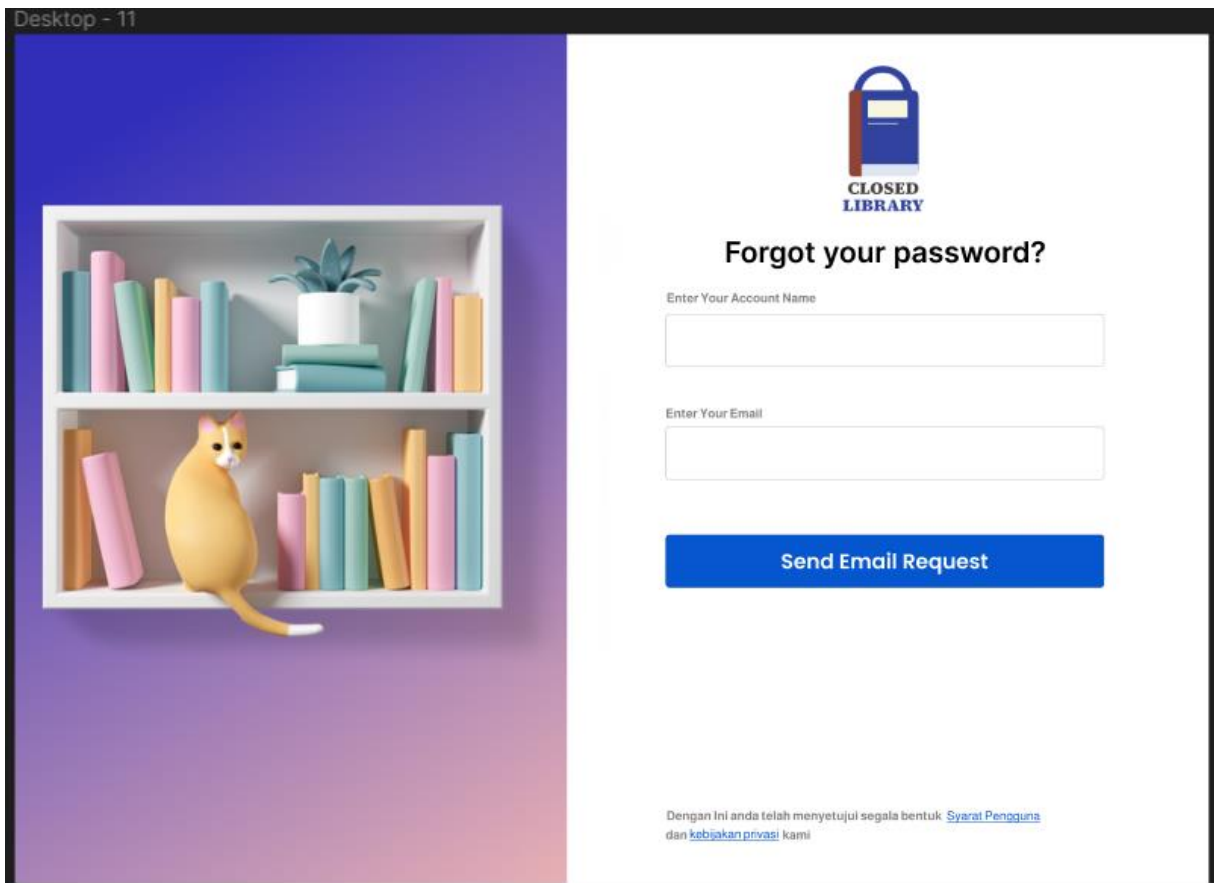
SCREEN ID	SCREEN NAME	DESCRIPTION
	Login Page	Page that is first landed if the user has not login.
	Forgot Password Page	Page that will be landed if the user click the forgot password button.

FOR EACH INTERFACE/PAGE, create detailed specifications.

Interface XXX: {fill in with screen number or design image number}

Login Page

Object_ID	TYPE	LABEL*	Description**
Button 1	Button	Login	After Clicking this button, the user can go to the dashboard page given the credentials are correct.
Button 2	Button	Forgot password?	If this button is clicked the user can go reset their password after using a recovery option.
Button 3	Button	Daftar gratis	This button allow user to register if they have not registered yet.
TextBox 1	TextBox	Username	This box is for the user to enter their username
TextBox 2	TextBox	Password	This box is for the user to enter their Password
Button 4	Button	Syarat Pengguna	This button will allow the user to view the terms and Condition
Button 5	Button	Kebijakan Privasi	This button will allow the user to view the Privacy Policy



Forgot Password Page

Object_ID	TYPE	LABEL*	Description**
TextBox 1	TextBox	Username	This box is for the user to enter their username
TextBox 2	TextBox	Email	This box is for the user to enter their email
Button 1	Button	Send	This button allows the system to send the recovery instruction to the email.

3.1.2.2 New Object Identification & Class Type #2 <Login> - Andi Ziyadh Ghafa R

Identify objects related to this use case.

Classes in the design phase are different from classes in the analysis phase.

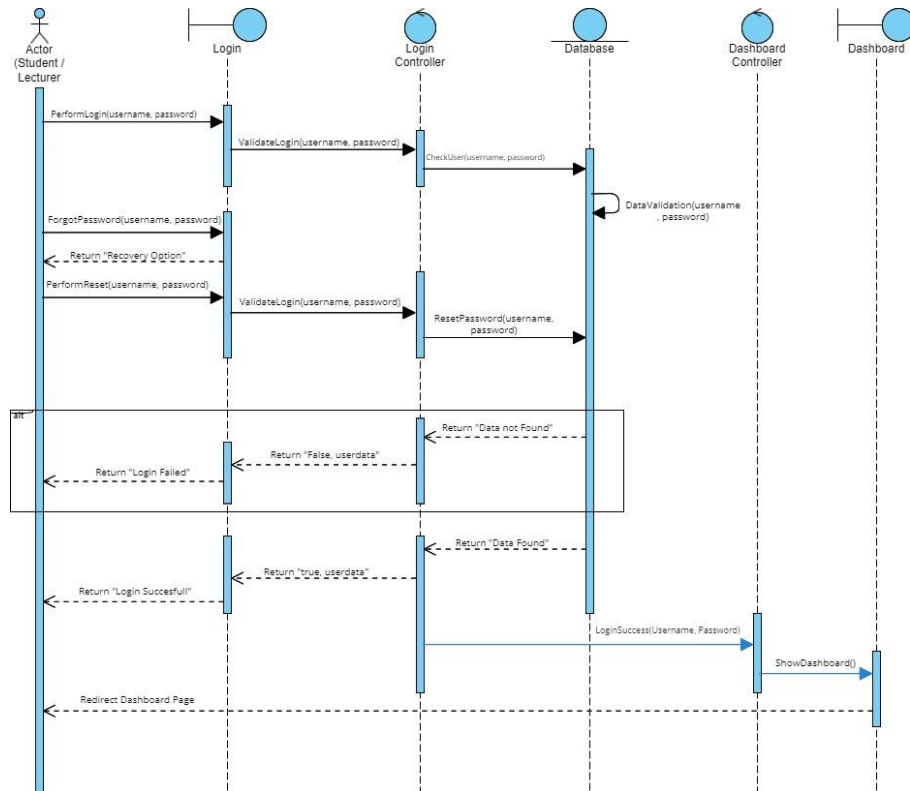
Use the table below:

TABLE OF DESIGN OBJECTS

No	New Object Name	Class Type
1	Student	Actor
2	Lecturer	Actor
3	Login	Boundary
4	Login Controller	Controller
5	Database	Entity
6	Dashboard Controller	Controller
7	Dashboard	Entity

*Class types such as Boundary (Interface), Entity (Database), Controller

3.1.2.3 Sequence Diagram #2 Login – Andi Ziyadh Ghafa R




3.1.3 Use Case #3 <Register> Jonathan David Brillyanto


Use Case Scenario #1 :

- i. Pre-Condition

The user has no account and wants to create an account
- ii. Use Case Description
 - a. Primary Flow
 1. The user navigates to the register pages on the website
 2. The user enter their data (email, account name, password)
 3. The system validates the data that goes into the database.
 4. If there is no data in database, the system stored the new user data in database
 5. The user is directed to the login page.
 - b. Alternative Flow
 1. User data is already available in the database:
 2. The user is directed to the login page.
- iii. Post-Condition
 - If the user has finished plugging in the data and gets a confirmation email, the system will display the login page.
 - The User is able to login to their account

3.1.3.1 UI Design dan UI Object Description #3 Register – Jonatan David Brillyanto





Register Your Account

Email

Account Name

Password

Repeat Password

Dengan Ini anda telah menyetujui segala bentuk [Syarat Pengguna](#) dan [kebijakan privasi](#) kami

Register

Tabel of UI Object Descriptions

SCREEN ID	SCREEN NAME	DESCRIPTION
	Register Page	Page that is first landed if the user Users do not have an account

FOR EACH INTERFACE/PAGE, create detailed specifications.

Interface XXX: {fill in with screen number or design image number}

Registration Page

Object_ID	TYPE	LABEL*	Description**
Button1	Button	Register	After Clicking this button, the user can go to the to the login page after getting the confirmation email.
TextBox 1	TextBox	Email	This box is for the user to enter their email
TextBox 2	TextBox	Username	This box is for the user to enter their username
TextBox 1	TextBox	Password	This box is for the user to enter their pass
TextBox 2	TextBox	Repeat password	This box is for the user to enter their username

3.1.3.2 New Object Identification & Class Type #3 Register – Jonatan David Brillyanto

Identify objects related to this use case.

Classes in the design phase are different from classes in the analysis phase.

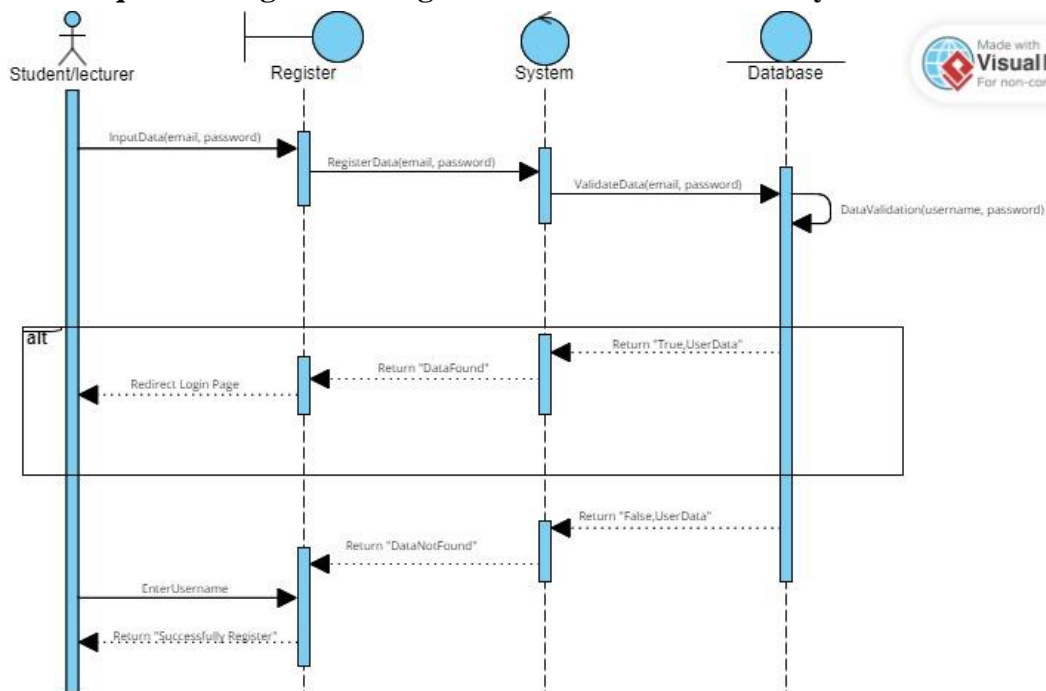
Use the table below:

TABLE OF DESIGN OBJECTS

No	New Object Name	Class Type
1	Student	Actor
2	Lecturer	Actor
3	Register	Boundary
4	Database	Entity
5	System	Controller

*Class types such as Boundary (Interface), Entity (Database), Controller

3.1.3.3 Sequence Diagram #3 Register – Jonathan David Brillyanto



3.1.4 Use Case #4 Borrow Page – Azkal Azkiya Zalfa

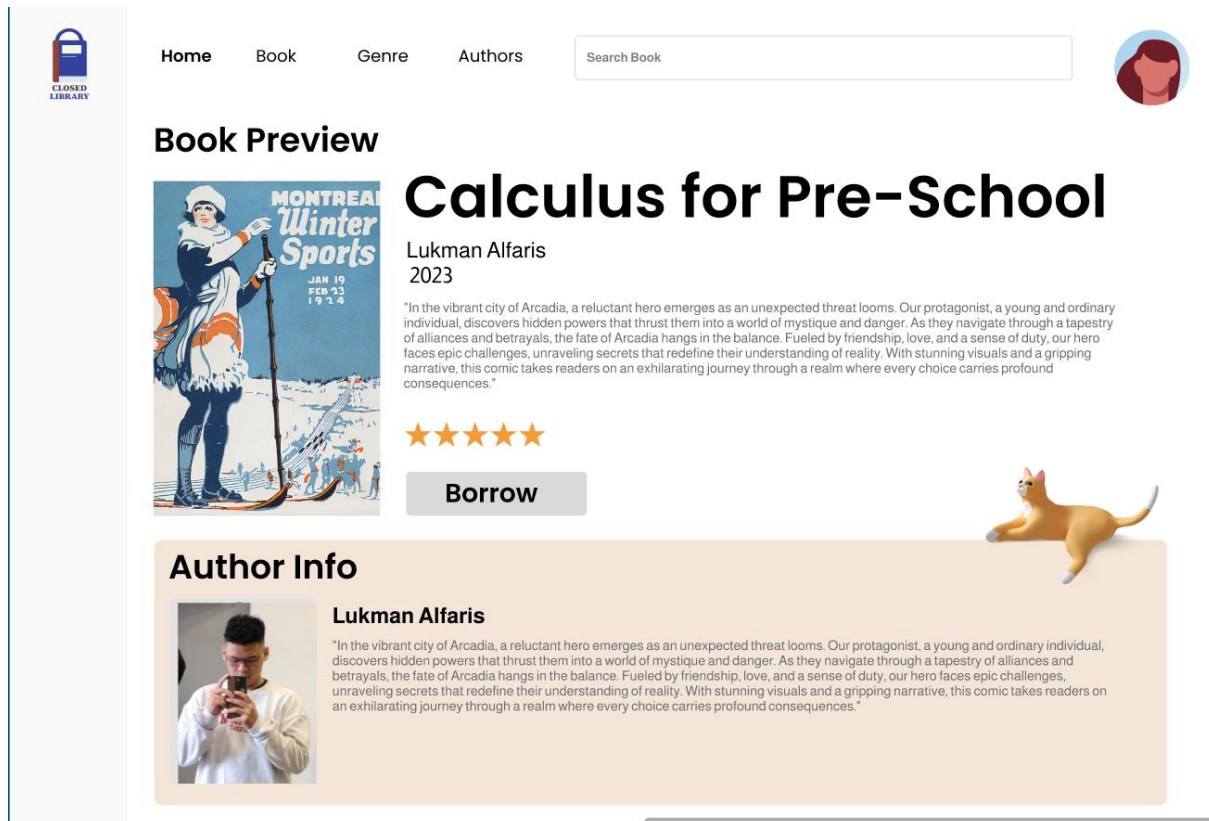
Use Case Scenario #1 :

- i. Pre-Condition
A user accesses a book's page in an attempt to borrow the book.
- ii. Use Case Description
 - a. Primary Flow
The user successfully borrows the book given the right condition.
 - b. Alternative Flow
The user is unsuccessful in their attempt to borrow a book due to their record of missed deadline when returning a book.
- iii. Post-Condition
The user can pick up the actual, physical copy of a book from the librarian desk.\

3.1.5 UI Design dan UI Object Description #4 Borrow Page – Azkal Azkiya Zalfa

Tabel of UI Object Descriptions

SCREEN ID	SCREEN NAME	DESCRIPTION
	Preview Book Page	The user can preview the book, see the book's metadata/description and its author(s), and a general synopsis of the book.
	Borrow Book Page	The user can borrow the book and set the duration of it.



Homepage

Object_ID	TYPE	LABEL*	Description**
Button1	Button	Home	If clicked, it'll return the user back to the Homepage
Button2	Button	Book	If clicked, it'll return the user to the Dashboard
Button3	Button	Genre	If clicked, it'll return the user to the Dashboard and listed the book by genre
Button4	Button	Authors	If clicked, it'll return the user to the Dashboard and listed the book by author
Button5	Button	Logout	If clicked, it'll log out the user and return them to the login page
Button6	Button	Borrow	If clicked, it'll direct the user to the Borrow Book page
TextBox	Text Box	Search	If clicked, it'll search the book based on what the user inputs



Want to borrow our book?

Borrower Name

Book Title

Duration

Request



Copyright By Closed Library 2023

REGISTRATION Page

Object_ID	TYPE	LABEL*	Description**
TextBox1	Textbox	Borrower Name	It will record the borrower's name.
TextBox2	Textbox	Book Title	It will record the book title.
TextBox3	Textbox	Duration	It will record the borrowing duration.
Button1	Button	Request	It will directly request for the librarian to put the book on hold until the borrower picks it up.

3.1.5.1 New Object Identification & Class Type #4 Borrow Page – Azkal Azkiya Zalfa

Identify objects related to this use case.

Classes in the design phase are different from classes in the analysis phase.

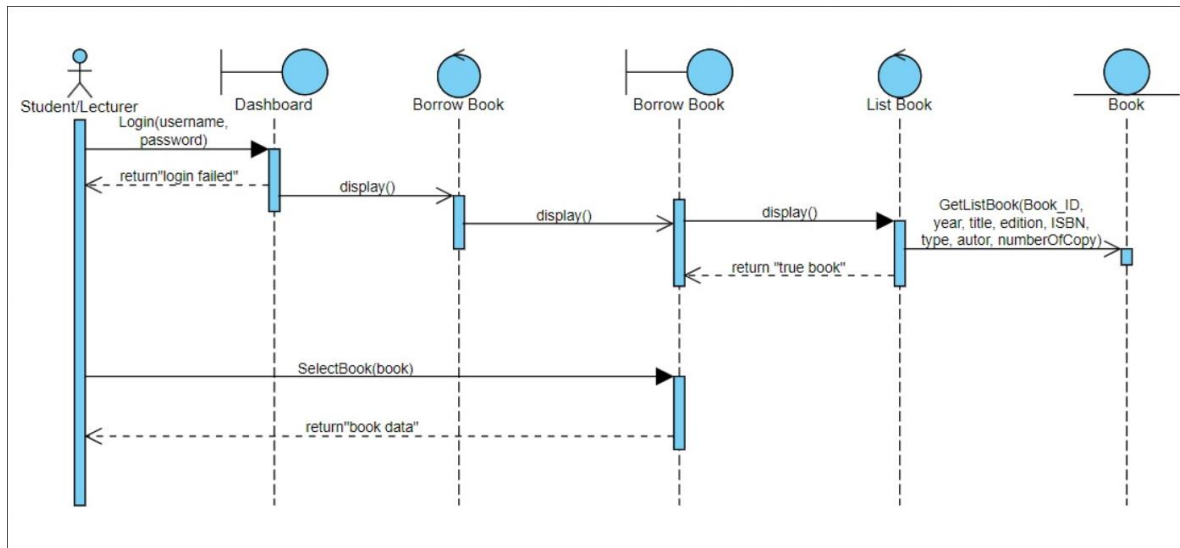
Use the table below:

TABLE OF DESIGN OBJECTS

No	New Object Name	Class Type
1	Student/Lecturer	Actor
2	Dashboard	Boundary
3	Dashboard Controller	Controller
4	Borrow Book	Boundary
5	Book Controller	Controller
6	Book	Entity

*Class types such as Boundary (Interface), Entity (Database), Controller

3.1.5.2 Sequence Diagram #2 Borrow Page – Azkal Azkiya Zalfa

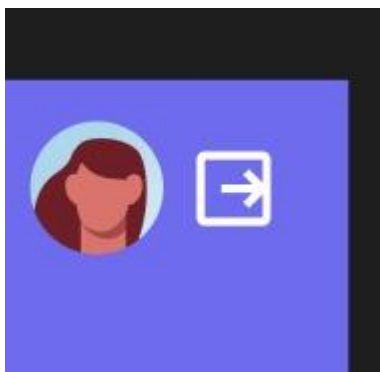


3.1.6 Use Case #5 Sign Out – Faiz Azizan Rashied

Use Case Scenario #1 :

- 1 Pre-Condition
The user is is currently logged into the system
- 2 Use Case Description
 - Primary Flow
The user clicks the designated logout button
The system gets the request for logging out
The system checks the connection of the user
The system confirmed that the actor has successfully logged out
 - Alternative Flow
The user connection timed out:
The user is directed the last page the user viewed.
- 3 Post-Condition
The User is able to logout safely.

3.1.1.1 UI Design dan UI Object Description #5 Sign Out – Faiz Azizan Rasied



Tabel of UI Object Descriptions

SCREEN ID	SCREEN NAME	DESCRIPTION
	Sign Out Button	A button that is present besides the profile icon that serves the purpose of logging out the user if pressed

FOR EACH INTERFACE/PAGE, create detailed specifications.
Interface XXX: {fill in with screen number or design image number}

Sign Out Page

Object_ID	TYPE	LABEL*	Description**
<i>Button1</i>	<i>Button</i>	<i>Sign Out Button</i>	<i>If clicked, it will make the user logged out</i>

3.1.1.2 New Object Identification & Class Type #5 Sign Out Page – Faiz Azizan R

Identify objects related to this use case.

Classes in the design phase are different from classes in the analysis phase.

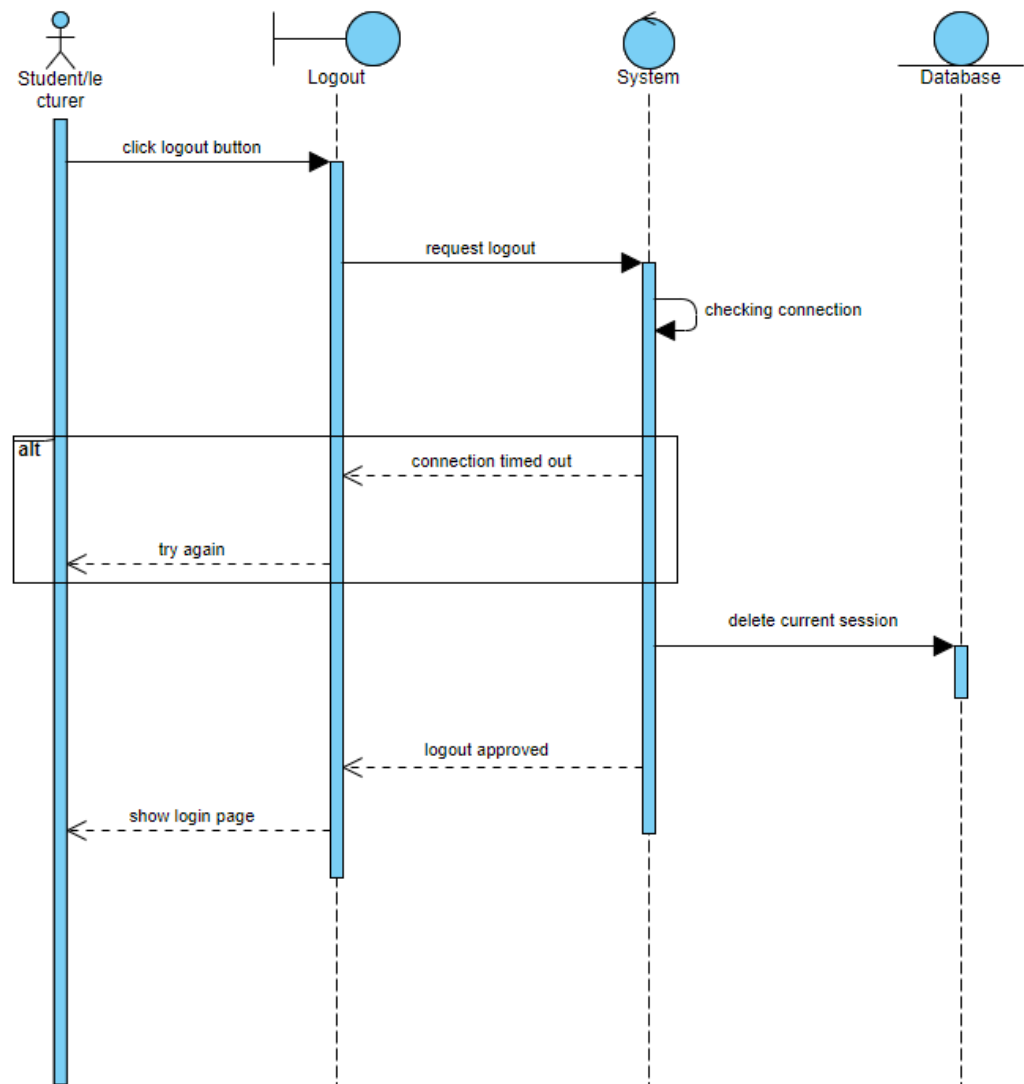
Use the table below:

TABLE OF DESIGN OBJECTS

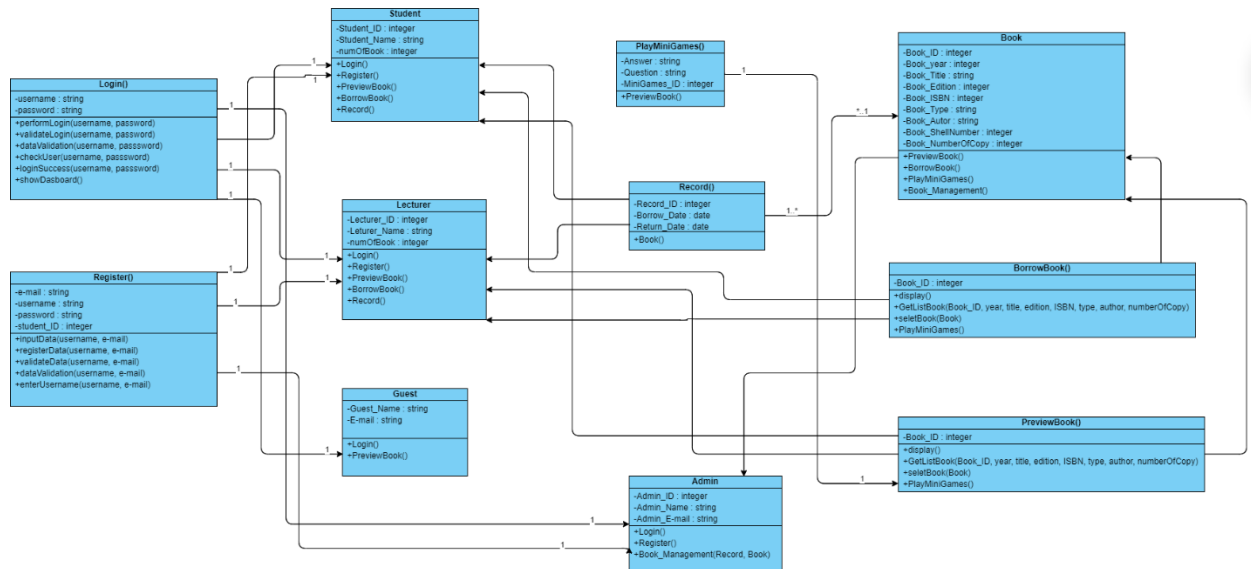
No	New Object Name	Class Type
<i>1</i>	<i>Student</i>	<i>Actor</i>
<i>2</i>	<i>Lecturer</i>	<i>Actor</i>
<i>3</i>	<i>Logout</i>	<i>Boundary</i>
<i>4</i>	<i>System</i>	<i>Controller</i>
<i>5</i>	<i>Database</i>	<i>Entity</i>

**Class types such as Boundary (Interface), Entity (Database), Controller*

3.1.1.3 Sequence Diagram #5 Sign Out Page – Faiz Azizan R



3.2 Overall Class Diagram



3.3 Detailed Class Design

Fill in this section with a comprehensive list of classes that will be used in the software, following the MVC model.

TABLE OF CLASSES :

Class ID	Name of Design Class	Attribute (visibility)	Method / Operation
1	Login	Username, Password	performLogin(), validate Login(), dataValidation(), checkUser(), loginSuccess(), showDashboard()
2	Register	e-mail, username, password, student_ID	inputData(), registerData(), validateData(), dataValidation(), enterUsername()
3	Student	Student_ID, Student_Name, numOfBook	Login(), Register, PreviewBook(), BorrowBook(), Reord()
4	Lecturer	Lecturer_ID, Lecturer_Name, numOfBook	Login(), Register, PreviewBook(), BorrowBook(), Reord()
5	Guest	Guest_Name, e-mail	Login(), PreviewBook()
6	PlayMiniGames	Answer, Question, Minigames_ID	PreviewBook()
7.	Record	Record_ID, Borrow_Date, Return_Date	Book()
8	Admin	Admin_ID, Admin_Name, Admin_e-mail	Login(), Register(), Book_Management()
9	Book	Book_ID, Book_Title, Book_year, Book_Edition, Book_ISBN, Book_Type, Book_Author, Book_ShelfNumber, Book_NumberOfCopy	PreviewBook(), BorrowBook(), PlayMiniGames(), Book_Management()
10	BorrowBook	Book_ID	Display(), GetListBook(), selectBook(), PlayMiniGames()

11	PreviewBook	Book_ID	Display(), GetListBook(), selectBook(), PlayMiniGames()
----	-------------	---------	---

For each class:

- Identify the operations (referring to class responsibilities), as well as the attributes, including their visibility.

3.4 Algorithm and/or Query Design

Complete this section exclusively for the algorithm framework of processes deemed significant. Implementation of the skeleton code can also be carried out for classes defined in the specific programming language. Subsections per class are permissible.

Algorithm #1

Class Name: UserAuthentication

Operation Name: authenticateUser

Algorithm:

1. Receive user's email and password.
2. Connect to the database.
3. ***SELECT * FROM users WHERE email = receivedEmail;***
4. If a matching user is found, compare the received password with the hashed password stored in the database.
5. If the passwords match, create a session for the user and return a success message.
6. Otherwise, return an error message indicating invalid credentials.

Query #2

Query: SELECT * FROM books WHERE title LIKE '%search_term%' OR author LIKE '%search_term%';

Description: This query retrieves all books from the database that have a title or author containing a specified search term, enabling users to search for books by title or author name.

Additional Algorithms and Queries:

Algorithm #3

Class Name: BookReservation

Operation Name: reserveBook

Algorithm:

1. Receive book ID and user ID.
2. Check availability of the book.
3. If available, update the book's status to "reserved" and associate it with the user's account.
4. Send a notification to the user confirming the reservation.

Query #4

Query: SELECT * FROM reservations WHERE user_id = userId AND status = 'pending';

Description: This query retrieves all pending reservations for a specific user, allowing them to view and manage their upcoming reservations.

Algorithm #5

Class Name: DigitalContentManagement

Operation Name: accessDigitalContent

Algorithm:

1. Receive user ID and content ID.
2. Verify user's authorization to access the content.
3. Generate a temporary access link or stream the content directly to the user.
4. Track content usage for licensing and analytics purposes.

Query #6

Query: `SELECT * FROM digital_content WHERE category = 'ebook' AND language = 'English';`

Description: This query retrieves all English-language ebooks from the database, allowing users to browse digital content by category and language.

Query #2

{If referring to a specific query, complete the query below}

*Query : SELECT * FROM data*

Description : Description of the query's purpose and functionality

4 Requirement Traceability Matrix

Fill in with the mapping between the requirement and the realized use case.

FR Code	Name of Functional Requirement	Name of Use Case
ADM-01	Data Validation	Register, Login
USER-01	User Registration	Register
USER-02	Login	Login
USER-03	Book Preview	PreviewBook
USER-04	Borrowing Book	Borrow Book
USER-05	Change Password	Login
USER-06	Forgot Password	Login