

Dear?

COMP-246 StuNest - A Software for Students made by students

Section 1: Problem Statement	1
Name: StuNest	1
Section 2: General Overview Modelling	4
Section 3: Functional Requirements. UML Use Case Modelling.....	4
Section 4: Domain Class Diagram	9
Section 6: Sequence Diagrams.....	11
Section 7: State Diagrams	13
Section 8: Technologies.....	15
Section 9: Project Management (Gantt Chart)	16

Part A: Project Scope and Requirements

Section 1: Problem Statement

Name: StuNest

1.1 a) Problem & Need:

a) Problem: College students often struggle to find suitable residential and rental properties. They have to rely on multiple applications or sources, which can be time-consuming and overwhelming. Hard to find housing just for students in other big marketplaces like kijiji and Facebook

b) Need: There is a need for a dedicated software solution and a website that specifically caters to the housing needs of college students, providing them with a centralized platform to search for and secure residential and rental properties easily which will be only rented to students.

1.1.b. List of Capabilities and Benefits

(i) Capabilities

1-Property Search: Students can search for residential and rental properties based on their preferences, such as location, price range, amenities, and proximity to their college/university.

2-Property Listings: Property owners/landlords can list their available properties with detailed information, including descriptions, photos, rental terms, and contact information.

3-User Profiles: Students can create profiles, manage their preferences, save favourite listings, and track their communication with property owners.

4-Reviews and Ratings: Users can leave reviews and ratings for properties they have rented, helping other students make informed decisions.

5-Communication: The software facilitates communication between students and property owners, allowing them to inquire about properties, schedule visits, and negotiate rental terms.

6-Document Management: The platform enables the exchange and management of rental agreements, lease documents, and other necessary paperwork securely.

7-Background Check: Student ID, Credit Report, Police verification and employment letter or GIC(funds to show)

(ii) Benefits:

1-Convenience: Students can easily find and compare available housing options from a single platform, saving time and effort.

2-Enhanced Decision-making: The software provides comprehensive property information, reviews, and ratings, enabling students to make informed decisions.

3-Improved Communication: Direct communication between students and property owners streamlines the process and eliminates intermediaries, ensuring quick and effective communication.

4-Secure Documentation: The software allows for secure storage and exchange of important rental documents, minimizing the risk of misplaced or lost paperwork.ct

1.2 Identify the stakeholders and their roles

1-Students: They are the main users of the software. Their role is to search for affordable and suitable housing options tailored specifically for students.

2-Landlords/Property Owners: They will use the software to list available housing options exclusively for students. Their role is to provide detailed information about the properties they have for rent and communicate with interested students.

3-Educational Institutions: They can play a role in supporting or partnering with us to promote the software among their student community, ensuring its reach and impact.

4-Administrators/Moderators: These users will manage the software platform. They will verify the authenticity of listings, resolve disputes, ensure compliance with policies, and maintain the overall functionality and security of the software.

5-Support Team: They will provide customer support and assistance to users of the software. Their role is to address any issues or queries raised by students, landlords, or other stakeholders.

1.3 Identify the sub-systems of your application

1-Login and Registration Subsystem: The Login and Registration Subsystem is where students create accounts and log in using email or social media. It offers password reset and account management features. Its purpose is to simplify the process and ensure a smooth experience for students joining our platform.

2-Search Subsystem: This component enables students to search for housing listings based on their preferences. They can filter the search by location, price range, property type, amenities, and availability. It will help them to find the best housing that suits their needs.

3-Listing Management Subsystem: This part of the application empowers landlords to manage their housing listings. They can create new listings, edit existing ones, mark listings as rented when occupied, and remove listings when no longer available.

4-Communication Subsystem: This component facilitates communication between students and landlords. Students can send inquiries to landlords regarding specific listings, and landlords can respond to those inquiries. It enables seamless and efficient communication within the platform.

5-User Profile Subsystem: This sub-system allows users to create and manage their profiles. Students and landlords can update their personal information, upload profile pictures, and view their saved or favorited listings in their profiles.

6-Notification Subsystem: This part of the application handles notifications to keep users informed about important events. Users receive notifications for new listings that match their preferences, incoming messages, and any changes in the status of their listings.

7-Admin/Moderation Subsystem: It provides tools for administrators and moderators to manage the software. They can verify listings, handle user reports, enforce policies, and resolve any disputes that may arise.

8-Review and Rating Subsystem: This component allows users to leave reviews and ratings for landlords or housing listings. It enables students to provide feedback and share their experiences, helping other students make informed decisions.

9-Payment Subsystem: This part of the application facilitates any payment-related functionalities. It handles rental payments or security deposits, ensuring a secure and convenient payment process for students and landlords.

10-Analytics and Reporting Subsystem: This sub-system collects and analyzes data to generate insights and reports. It helps track usage statistics, popular search criteria, and user feedback.

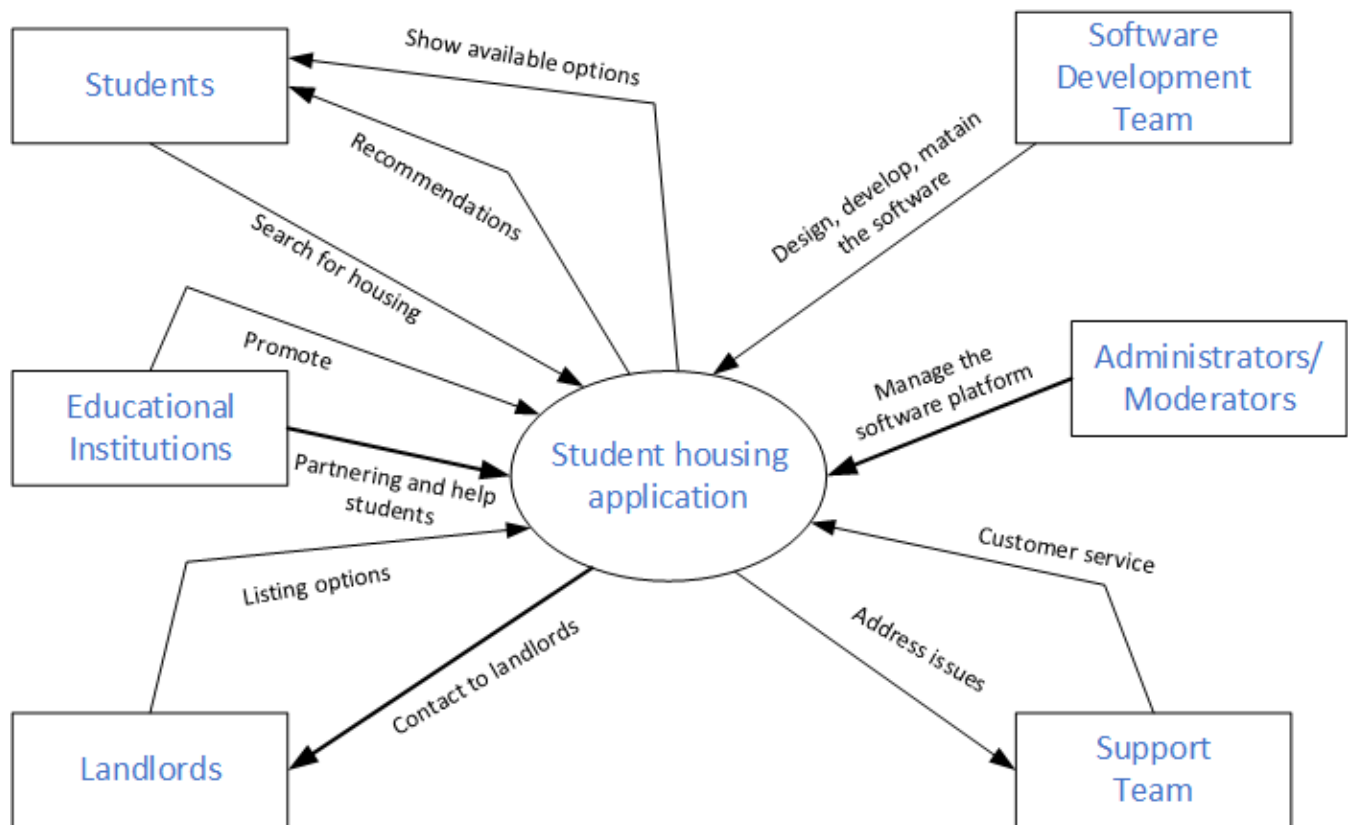
1.4 Who are the intended users of the SRS documentation?

The key players in our software project are the intended audience for the SRS documentation. Our competent software development team, which consists of project managers, software architects, designers, developers, and testers, is one example of this. The SRS document will serve as their primary reference during the whole development process. It makes sure that everyone is on the same page in their understanding of what needs to be done.

The paperwork is useful to a variety of stakeholders, including staff, students, landlords and administrators. Through reading the paper, they learn important details about the project's objectives, specifications, and features. They get a clear understanding of the software's goals and how it will assist students in locating the best and most relevant information.

Section 2: General Overview Modelling

2.1 Context Flow Diagram (CFD)



Section 3: Functional Requirements. UML Use Case Modelling

3.1 Goal Use Cases

FR №	Goal Use Case	Role Player	Description
FR 1	Registration	Student	A student has to go through a registration process in which he provides all information about himself.
FR 2	Search Listings	Student	A student performs a search for available housing listings based on criteria such as location, price range, property type, and amenities.
FR 3	Create Listing	Property owner	An owner creates a new housing listing by providing confirmed details such as location, rent amount, property type, amenities, and any specific requirements. Every information provided first being confirmed and moderated by administrator and only then posted on the website.
FR 4	Choose Listing	Student	Student can choose a listing, and have time either to move in or check the place out. Because he holds the place until the final decision.

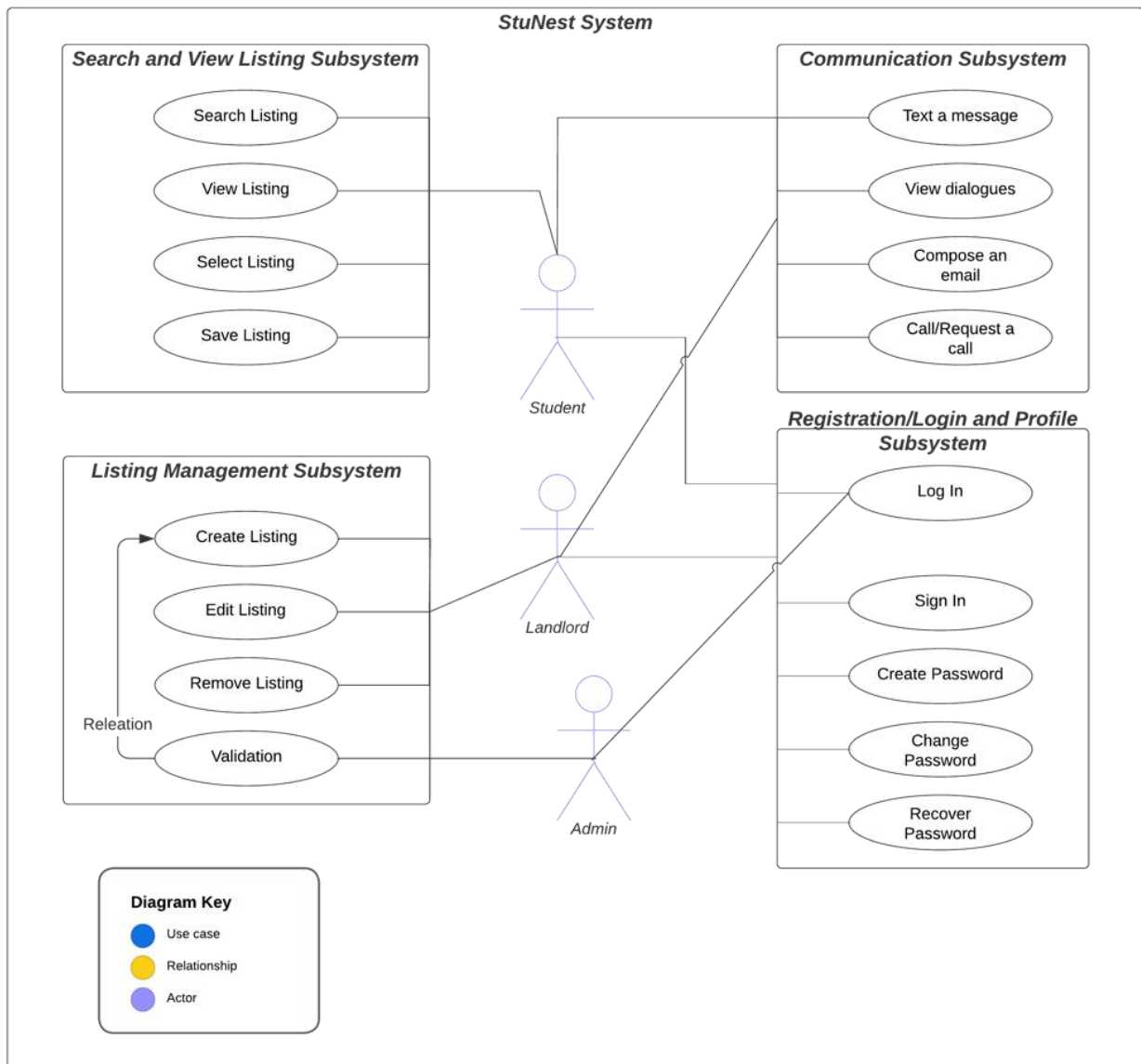
3.2 Use case Diagrams

1. Login and Registration Subsystem:

Actors: Student, Landlord, Administrator

Use Cases: Register Account (Student), Register Account (Landlord), Login (Student), Login (Landlord), Reset Password, Manage Account (Student), Manage Account (Landlord)

Description: The Student actor can register a new account, log in to their account, reset their password if needed, and manage their account settings. The Landlord actor has the same set of use cases for registration, login, and account management. The Administrator actor can manage user accounts and handle administrative tasks related to authentication and access control.

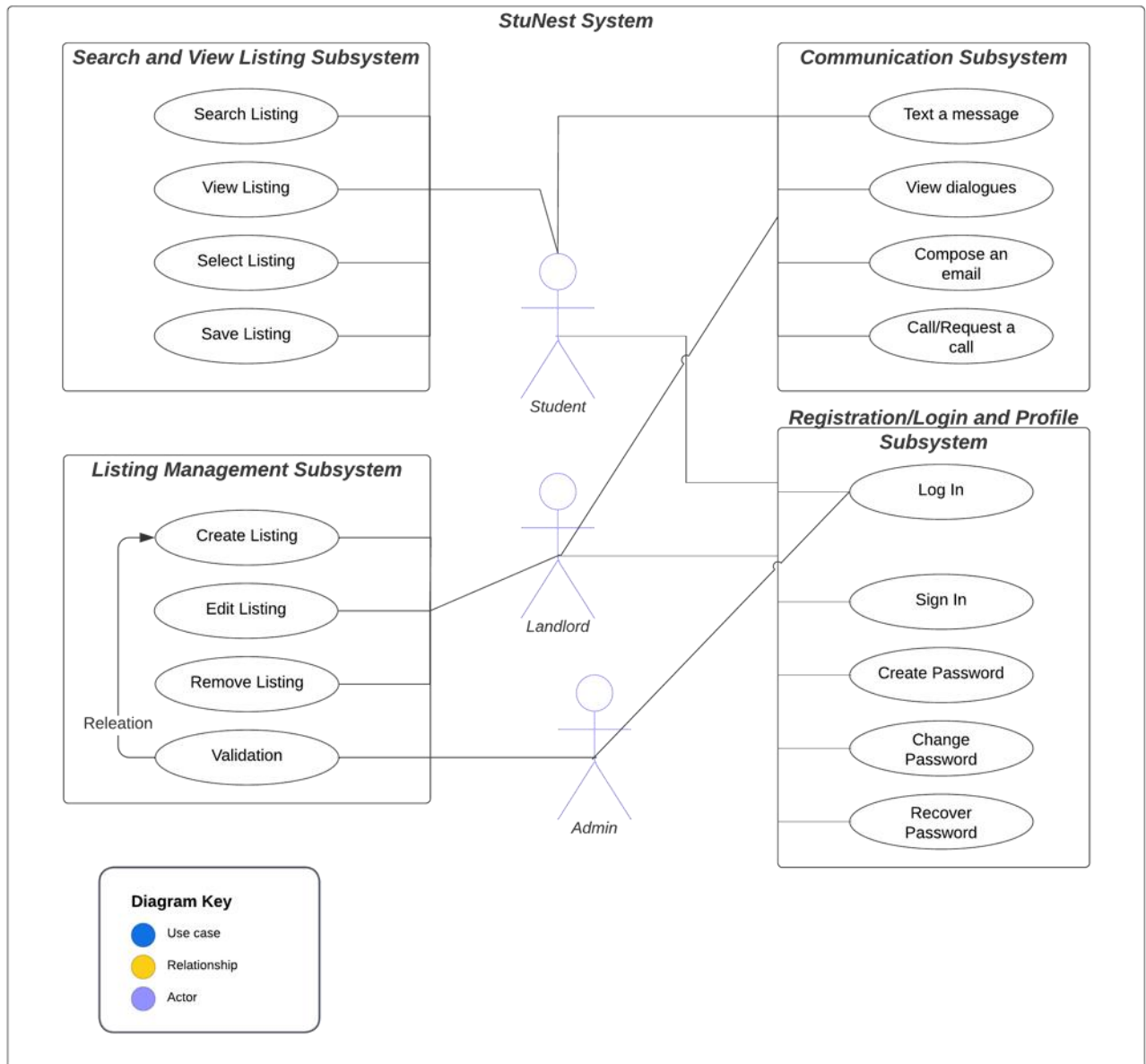


2. Search Subsystem:

Actors: Student

Use Cases: Search Properties, Filter Search Results, View Property Details

Description: The Student actor can search for properties, apply filters to narrow down the search results based on their preferences, and view detailed information about individual property listings.

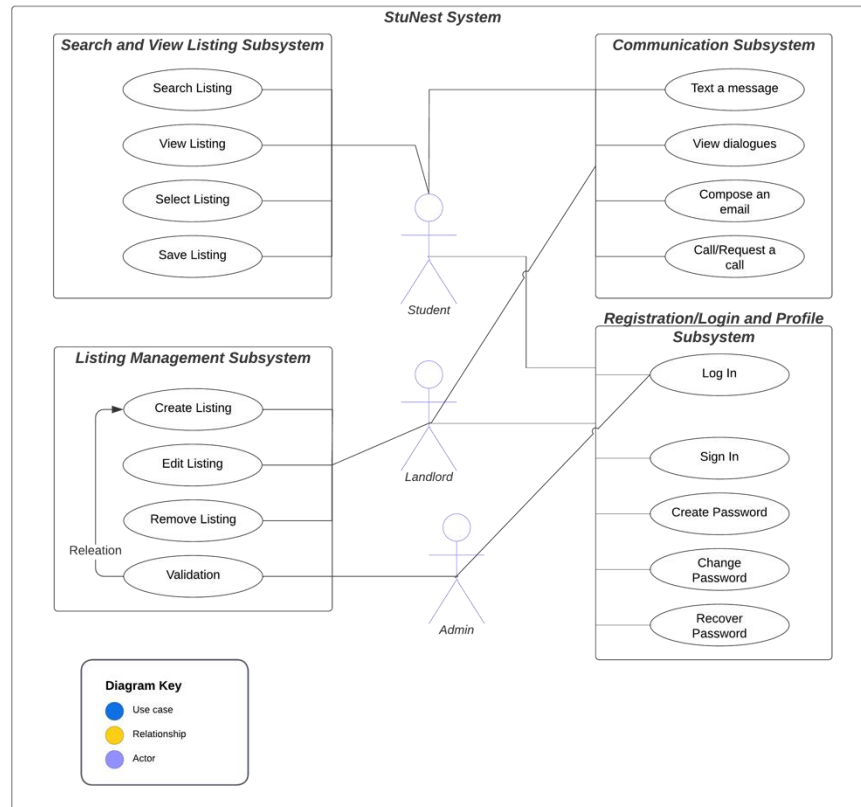


3. Listing Management Subsystem:

Actors: Landlord

Use Cases: Create Listing, Edit Listing, Remove Listing, Mark Listing as Rented

Description: The Landlord actor can create new property listings, edit existing listings, remove listings when they are no longer available, and mark listings as rented once they are occupied.



4. Communication Subsystem:

Actors: Student, Landlord

Use Cases: Send Inquiry (Student), Respond to Inquiry (Landlord), Schedule Visit (Student), Confirm Visit (Landlord), Negotiate Rental Terms (Student), Respond to Rental Terms (Landlord)

Description: The Student actor can send inquiries to landlords regarding specific listings, schedule property visits, and negotiate rental terms. The Landlord actor can respond to student inquiries, confirm property visits, and negotiate rental terms.

3.3 User Stories:

1) Login and Registration Subsystem:

As a student, I want to register an account with my email or social media credentials so that I can log in easily and access the housing platform.

- Acceptance Criteria:

The registration process accepts both email and social media registration options.

After successful registration, an account is created, and the user can log in.

As a student, I want to reset my password if I forget it, ensuring secure access to my account.

- Acceptance Criteria:

The password reset functionality allows the user to initiate a password reset.

The user receives an email with a secure link to reset their password.

Upon clicking the link, the user is redirected to a password reset page where they can set a new password.

2) Search Subsystem:

As a student, I want to search for available housing properties based on my preferences so that I can find suitable options and.

- Acceptance Criteria:

The search functionality allows students to enter preferences such as location, price range, amenities, and proximity to their college/university.

The search results display relevant housing properties based on the provided preferences.

3) Listing Management Subsystem:

As a landlord, I want to create new listings for my available housing properties, providing detailed information so that students can find and inquire about them.

- Acceptance Criteria:

The listing creation form allows landlords to enter detailed information about the property, including descriptions, photos, rental terms, and contact information.

After submitting the listing, it is successfully created and available for students to view.

As a landlord, I want to edit or remove my listings when necessary to ensure accurate and up-to-date information is available.

- Acceptance Criteria:

The listing management interface provides options to edit or remove existing listings.

When editing a listing, the updated information is reflected correctly in the listing details.

4) Communication Subsystem:

As a student, I want to send inquiries to landlords regarding specific listings to gather more information or express interest in renting.

- Acceptance Criteria:

The communication interface allows students to select a listing and send an inquiry to the respective landlord.

The landlord receives the inquiry and can view the student's message.

As a student, I want to schedule property visits and negotiate rental terms directly with landlords to facilitate the rental process.

- Acceptance Criteria:

The communication interface provides options for scheduling property visits and initiating rental term negotiations.

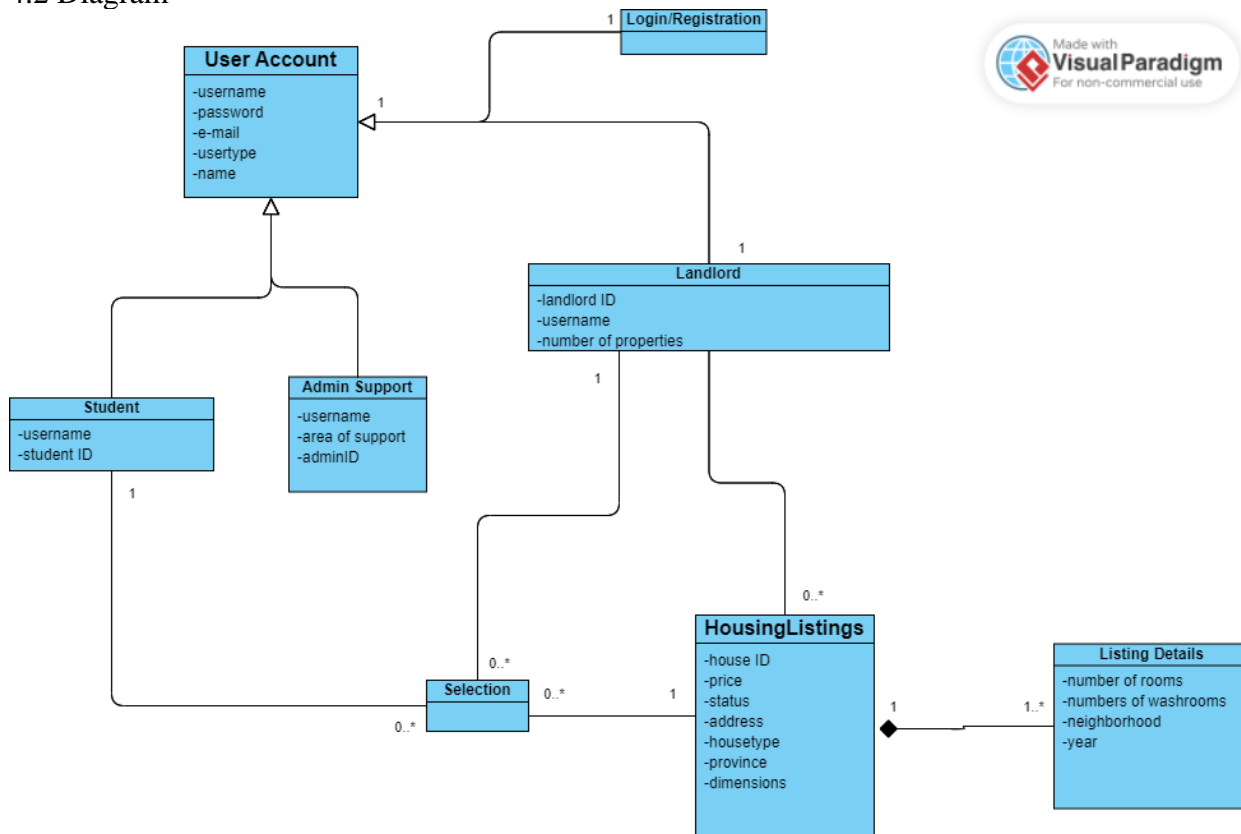
Both the student and landlord receive notifications and can view the communication history related to the visit or negotiation.

Section 4: Domain Class Diagram

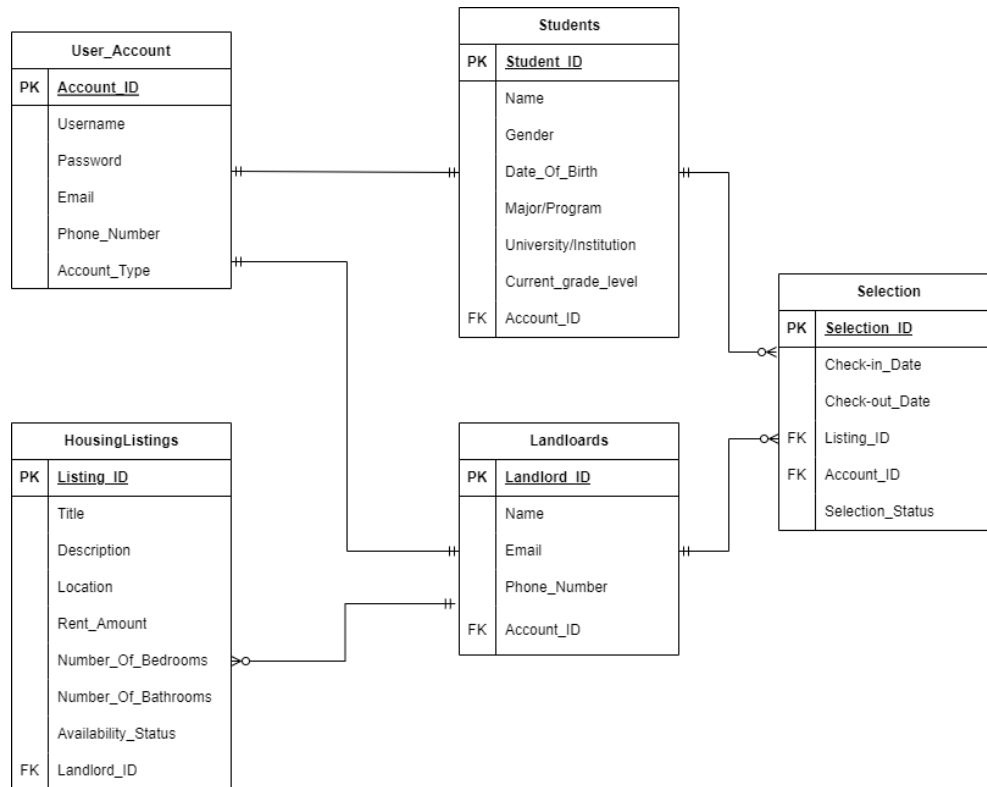
4.1 List of classes

1. Student
2. Landlord
3. HousingListings
4. User Account
5. Admin Support
6. Listing Detail

4.2 Diagram



Section 5: Entity Relationship Diagram (ERD)



Section 6: Sequence Diagrams

6.1 Login (use case: Login to the system)

a. Description:

This use case describes the process of logging in to the Rented House Finder System, a platform specifically designed to provide or find rented houses for students outside of their college or university. Users, both students seeking accommodation and property owners or agents, are required to log in to access the system's features and services.

b. Actors:

Student: A user who is seeking accommodation outside of their college.

Property Owner/Agent: A user who owns or represents a property available for rent.

Preconditions:

The Rented House Finder System is operational and accessible.

The user is registered and has valid login credentials.

The user has an active internet connection.

c. Basic Flow:

The user launches the Rented House Finder System application or visits the website.

The system presents the login page, displaying fields for username and password.

The user enters their username and password.

The user submits the login credentials.

The system validates the entered credentials.

If the credentials are valid, the system logs the user into their account.

The system redirects the user to the dashboard or home page of the application.

Alternative Flows:

d. Invalid Credentials:

In step 6, if the system determines that the entered credentials are invalid:

The system displays an error message indicating the invalid credentials.

The user is prompted to re-enter their credentials or initiate the password recovery process.

The user can make another attempt to log in or reset their password.

e. Registration:

If the user does not have an account:

The system provides a registration option or link.

The user clicks on the registration option.

The system redirects the user to the registration page.

The user enters the required details for registration.

The user submits the registration form.

The system validates the entered information and creates a new account.

The user is automatically logged in, and the system redirects them to the dashboard or home page.

f. Postconditions:

Upon successful login, the user gains access to the features and services provided by the Rented House Finder System.

The user remains logged in until they manually log out or their session expires.

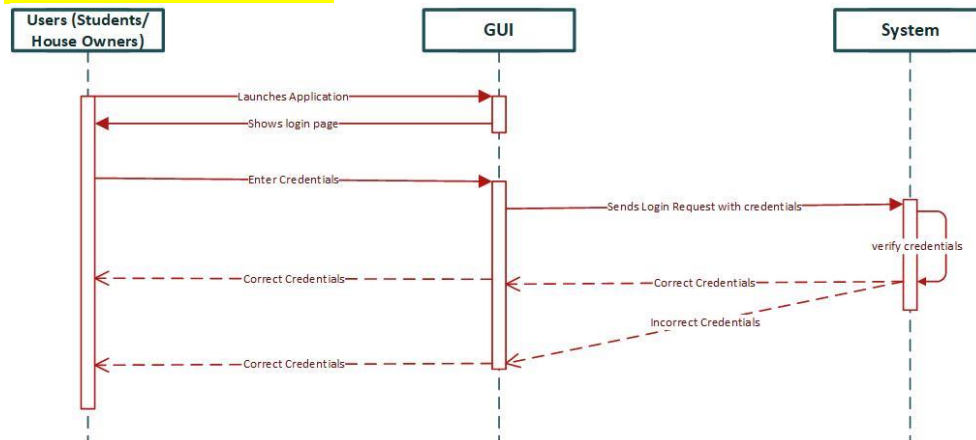
g. Notes:

Security measures such as encryption, secure storage of passwords, and session management should be implemented to ensure the confidentiality and integrity of user data.

The login process may involve additional security measures like two-factor authentication to enhance the system's security.

Password recovery mechanisms should be provided to assist users who forget their login credentials.

h. Diagram Use case Create account



6.2 Search (Use cases: enter information and retrieve information)

a. Use case Description

In this sequence diagram, the student interacts with the GUI to initiate a search. The GUI displays a search form where the student can enter their search criteria. The student enters the criteria, and the GUI sends a search request to the system. The system processes the request and retrieves a list of matching rental houses, which is then sent back to the GUI. The GUI displays the list of rental houses to the student, who can select a specific rental house to view more details. The GUI sends a request for the rental house details to the system, which retrieves and sends the details back to the GUI for display to the student.

b. Basic Flow:

The student interacts with the GUI to initiate a search for rental houses.

The GUI presents a search form or interface where the student can enter their search criteria, such as location, price range, number of bedrooms, amenities, etc.

The student enters the desired search criteria.

The GUI sends a search request to the system, including the entered search criteria.

The system processes the search request and retrieves a list of rental houses that match the specified criteria.

The system sends the list of rental houses back to the GUI.

The GUI receives the list of rental houses from the system.

The GUI displays the list of rental houses to the student.

The student reviews the list of rental houses and can choose to view more details for a specific rental house.

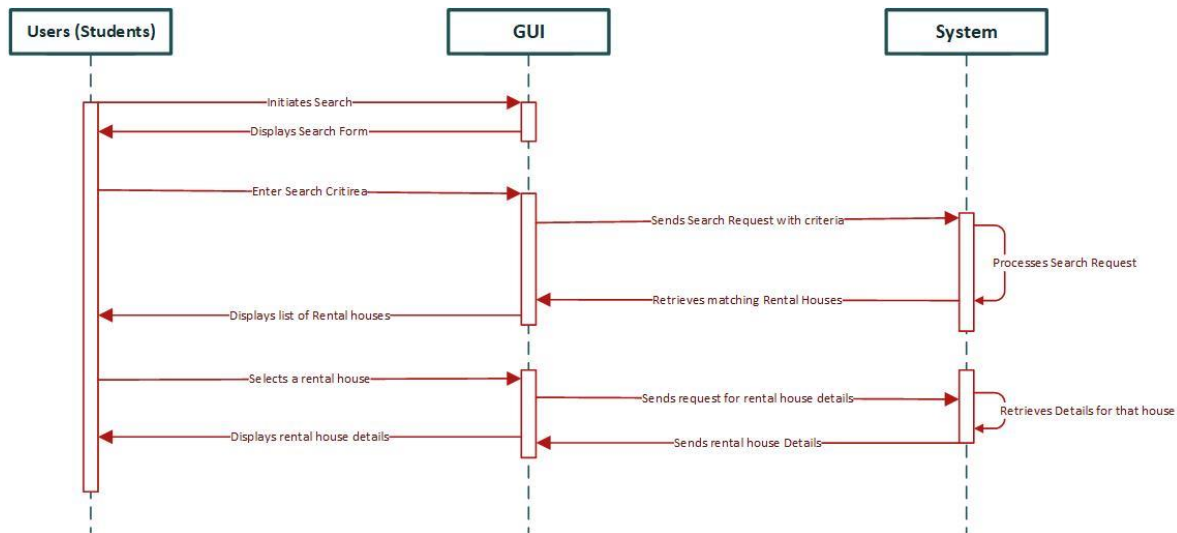
c. Postconditions:

The student is presented with a list of rental houses matching their entered search criteria. The student can choose to view more details for a specific rental house.

d. Exceptions:

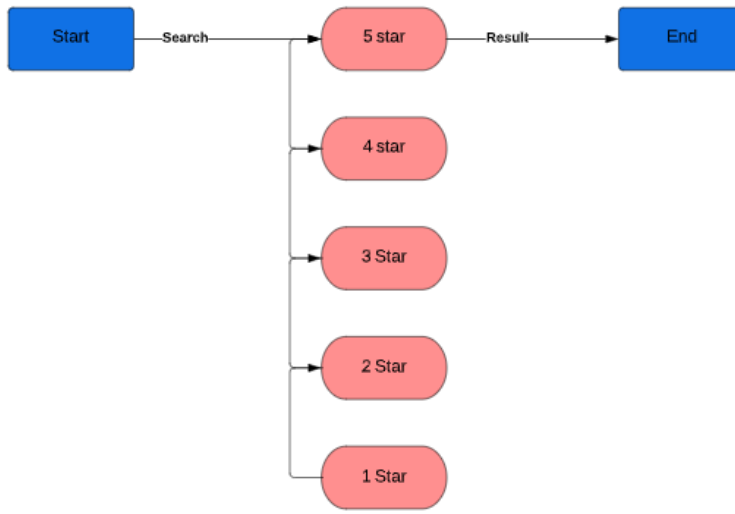
If there are any technical issues during the search process, such as network connectivity problems or system errors, the system provides appropriate error messages and guidance to the student.

e. Diagram Sequence Diagram for Search use case

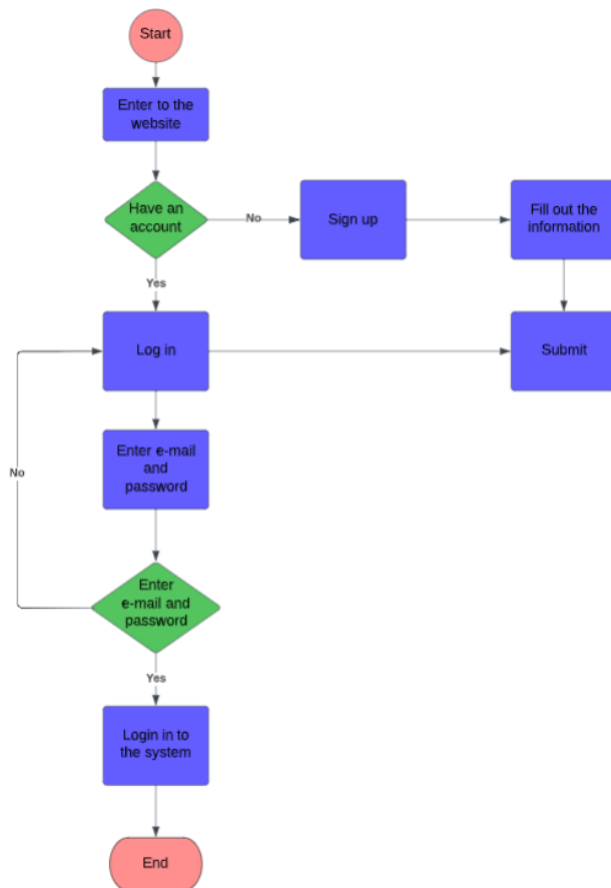


Section 7: State Diagrams

7.1 Rating system state machine object for landlord



7.2 State machine for search object



Section 8: Technologies

a) Mobile Application:

❖ Client Side (Front-end - GUI):

Programming Languages: Java, Kotlin, Swift, or React Native (for cross-platform development)

Development Frameworks: Android SDK (for Android), iOS SDK (for iOS)

UI/UX Frameworks: React Native, Flutter, Xamarin

IDEs: Android Studio (for Android), Xcode (for iOS)

UI Design Tools: Sketch, Adobe XD, Figma

Mobile App Testing: Appium, XCTest, Espresso

❖ Business Logic (Middle Layer - Class methods, etc.):

Programming Languages: Java, Kotlin (for Android), Swift (for iOS)

Development Frameworks: Spring Boot, Node.js, Ruby on Rails

API Integration: RESTful APIs, GraphQL

Authentication and Authorization: OAuth, JWT (JSON Web Tokens)

Mobile App Frameworks: React Native, Flutter

❖ Data Side (Database):

Database Technologies: SQLite (for local storage), Firebase Realtime Database, Firebase Cloud Firestore, PostgreSQL, MySQL

Database Query Languages: SQL (Structured Query Language)

Object-Relational Mapping (ORM): Room (for Android), CoreData (for iOS), Firebase Realtime Database SDK, Firebase Cloud Firestore SDK

b) Web Application:

❖ Client Side (Front-end - GUI):

Programming Languages: HTML, CSS, JavaScript

Front-end Frameworks: React.js, Angular, Vue.js

UI/UX Frameworks: Bootstrap, Material-UI, Bulma

IDEs: Visual Studio Code, Sublime Text, Atom

UI Design Tools: Sketch, Adobe XD, Figma

Front-end Testing: Jest, Cypress, Selenium

❖ Business Logic (Middle Layer - Class methods, etc.):

Programming Languages: JavaScript, TypeScript, Python, Ruby

Back-end Frameworks: Node.js (Express.js, Koa.js), Ruby on Rails, Django, Flask

API Integration: RESTful APIs, GraphQL

Authentication and Authorization: OAuth, JWT (JSON Web Tokens)

Server-side Rendering: Next.js, Nuxt.js

Web Application Frameworks: React.js, Angular, Vue.js

❖ Data Side (Database):

Database Technologies: MongoDB, MySQL, PostgreSQL, Firebase Realtime Database, Firebase Cloud Firestore

Database Query Languages: SQL (Structured Query Language), NoSQL (MongoDB Query Language)

Object-Relational Mapping (ORM): Mongoose (for MongoDB), Sequelize (for SQL databases), Firebase Realtime Database SDK, Firebase Cloud Firestore SDK

Section 9: Project Management (Gantt Chart)

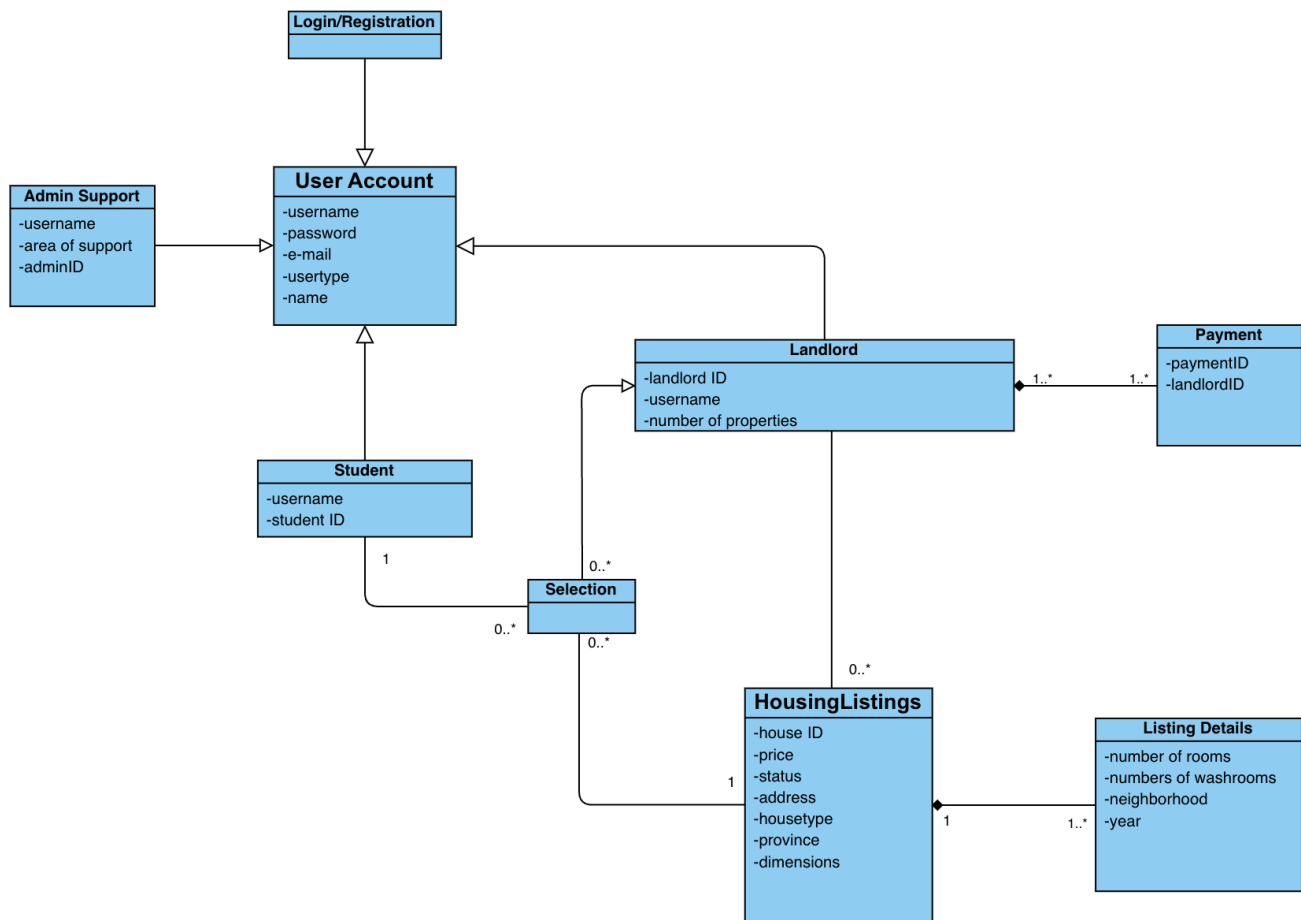
Task Name		Duration	Start	Finish	Predecesso...	Assigned To	% Complete	Status	May					Jun		
									30	May 7	May 14	May 21	May 28	Jun 4	Jun 11	Jun 18
1	Forming Group	1d	05/10/23	05/10/23		Professor	100%	Complete	<div><div></div></div> Professor							
2	Finalizing Company	6d	05/12/23	05/19/23		All	100%	Complete	<div><div></div></div> All							
3	Naming Company	1d	05/12/23	05/12/23		All	100%	Complete	<div><div></div></div> All							
4	Allotting Tasks	1d	05/12/23	05/12/23		All	100%	Complete	<div><div></div></div> All							
5	Section 1	6d	05/19/23	05/26/23		Abhishek	100%	Complete		<div><div></div></div> Abhishek						
6	Section 2	6d	06/02/23	06/09/23		Zhangir	100%	Complete			<div><div></div></div> Zhangir					
7	Section 3	6d	06/02/23	06/09/23		Zhangir	100%	Complete			<div><div></div></div> Zhangir					
8	Section 4	6d	06/02/23	06/09/23		Genya	100%	Complete			<div><div></div></div> Genya					
9	Section 5	5d	06/02/23	06/08/23		Sanghak	100%	Complete			<div><div></div></div> Sanghak					
10	Section 6	5d	06/02/23	06/08/23		Ranvir	100%	Complete			<div><div></div></div> Ranvir					
11	Section 7	6d	06/02/23	06/09/23		Dhruvit	100%	Complete			<div><div></div></div> Dhruvit					
12	Section 8	6d	06/02/23	06/09/23		Abhishek	100%	Complete			<div><div></div></div> Abhishek					
13	Section 9	5d	06/02/23	06/08/23		Dhruvil	100%	Complete			<div><div></div></div> Dhruvil					
14	Reviewing Part 1	1d	06/09/23	06/09/23		All + Professor	100%	Complete			<div><div></div></div> All + Professor					
15	Submitting Part 1	1d	06/11/23	06/11/23		Dhruvil	100%	Complete			<div><div></div></div> Dhruvil					
16																

Task Name	May					Jun				
	Apr 30	May 7	May 14	May 21	May 28	Jun 4	Jun 11	Jun 18	Jun 25	
1	Forming Group									
2	Finalizing Company									
3	Naming Company									
4	Allotting Tasks									
5	Section 1									
6	Section 2									
7	Section 3									
8	Section 4									
9	Section 5									
10	Section 6									
11	Section 7									
12	Section 8									
13	Section 9									
14	Reviewing Part 1									
15	Submitting Part 1									

Part B: Software Design Architecture

Section 1: Requirements Edits to Part A

1.1 Corrections on Domain Class Diagram (section 4.2)



1.2 ERD (section 5)

- Payment table connected to Users table.

1.3 Technologies (section 8)

- Update Web Application content updated.

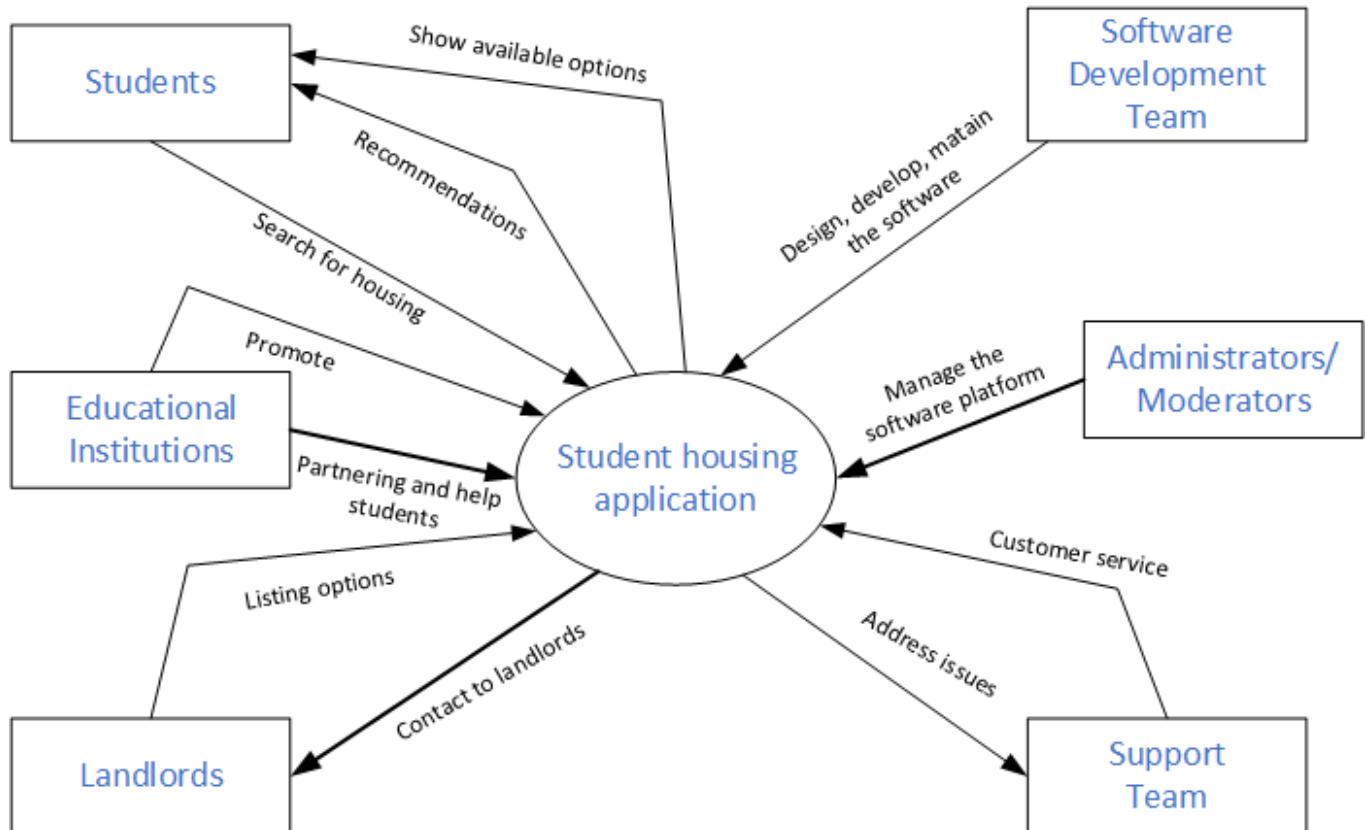
Section 2: Overview Model

2.1 Intended users of the SDD document

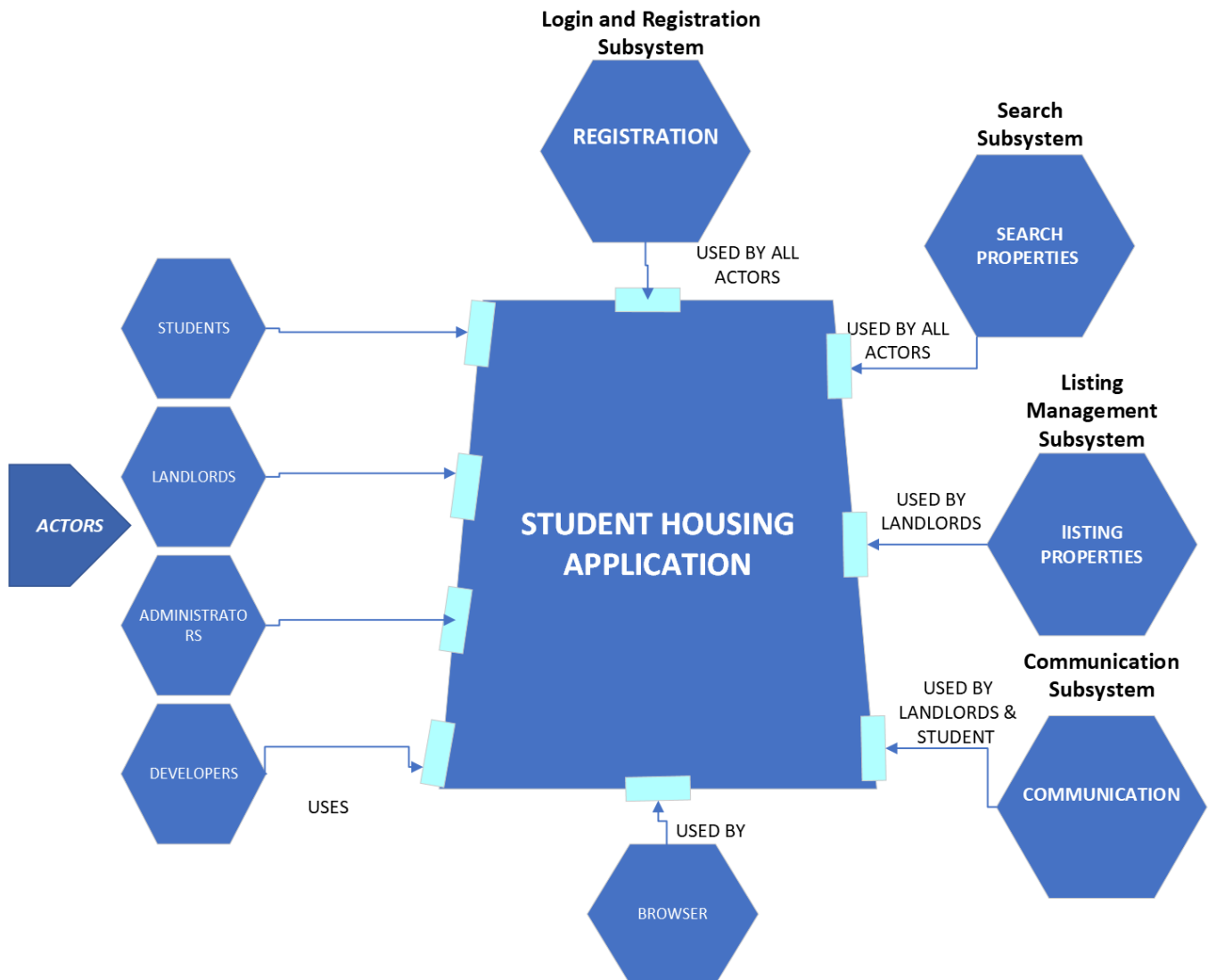
The main objective of the System Design Document (SDD) is to offer a comprehensive overview of the system's design, enabling software developers to gain a thorough understanding of the required development

tasks. As a result, this document is primarily intended for developers and managers who bear the responsibility of overseeing the system's development

2.2.1 SRS Context Flow Diagram --CFD (What diagram)



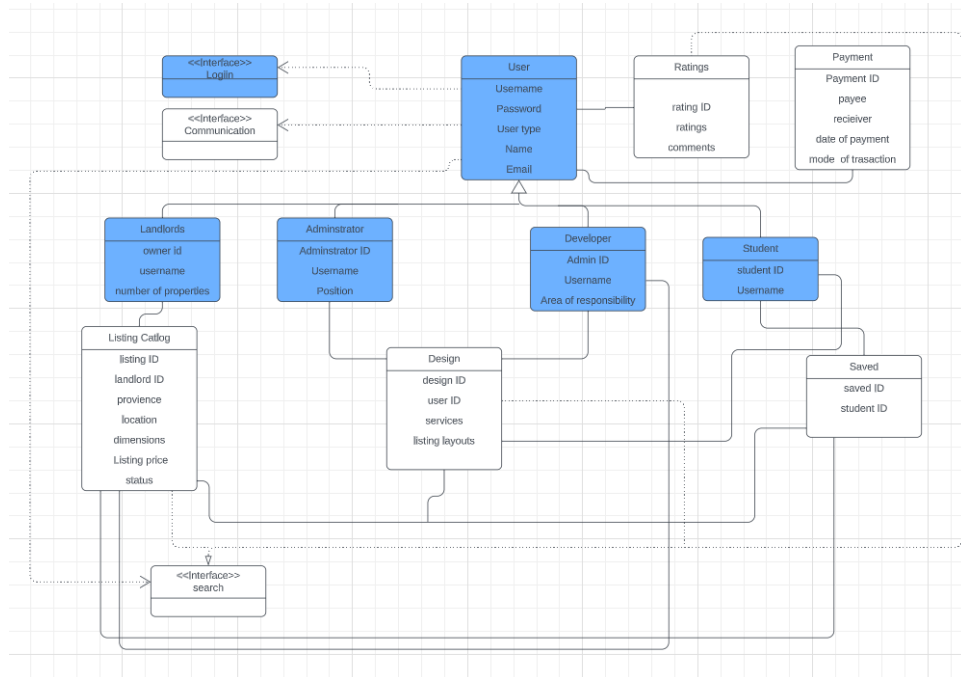
2.2.2 Architectural Context Diagram—ACD (HOW diagram)



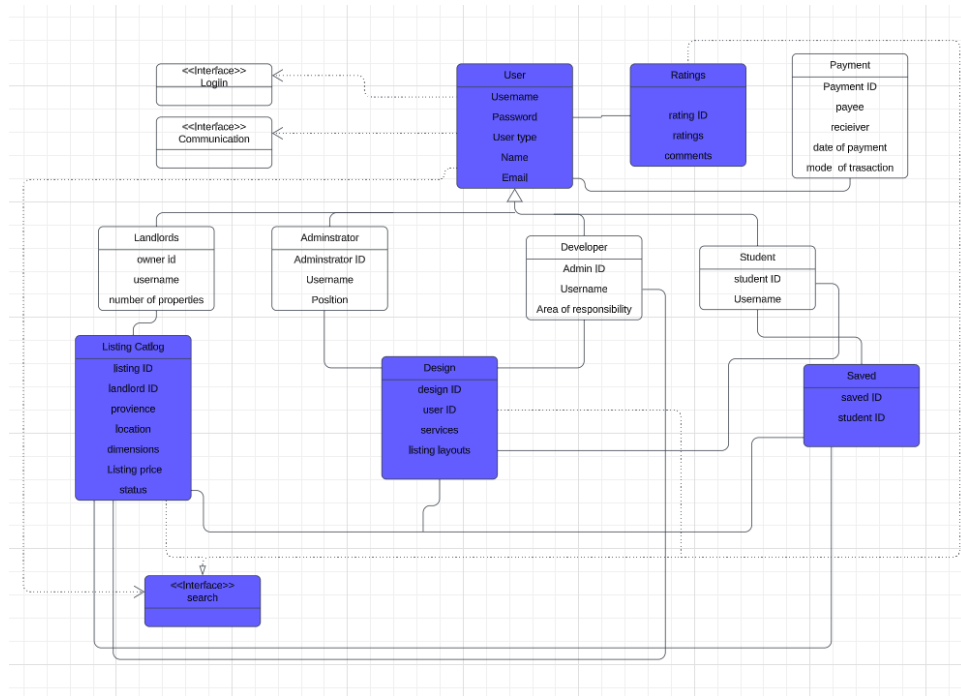
Section 3: Modularization

3.1 Partition the analysis model

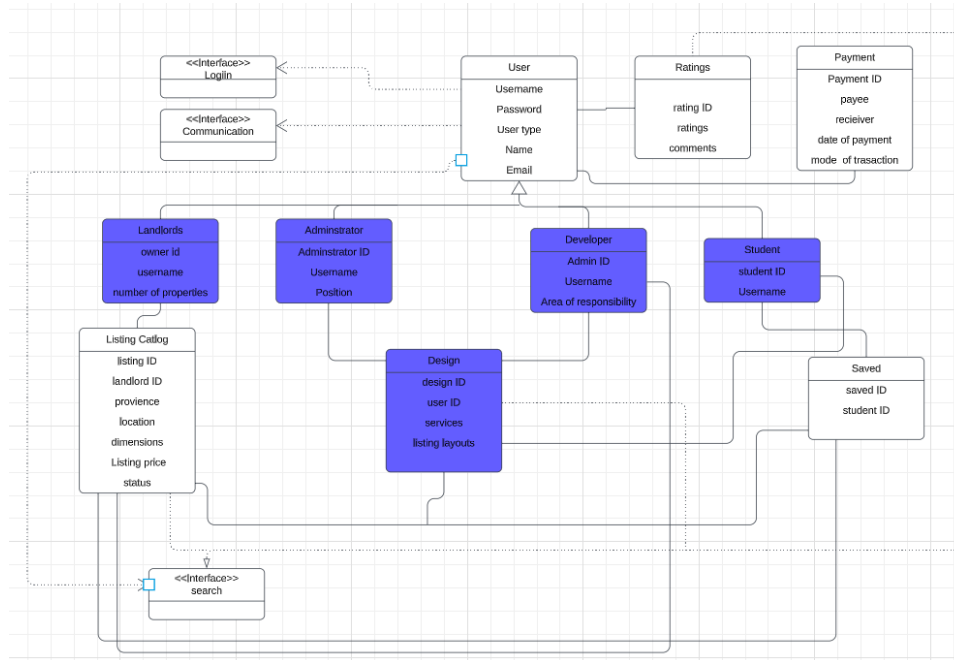
3.1.1 Login subsystem



3.1.2 Search Subsystem

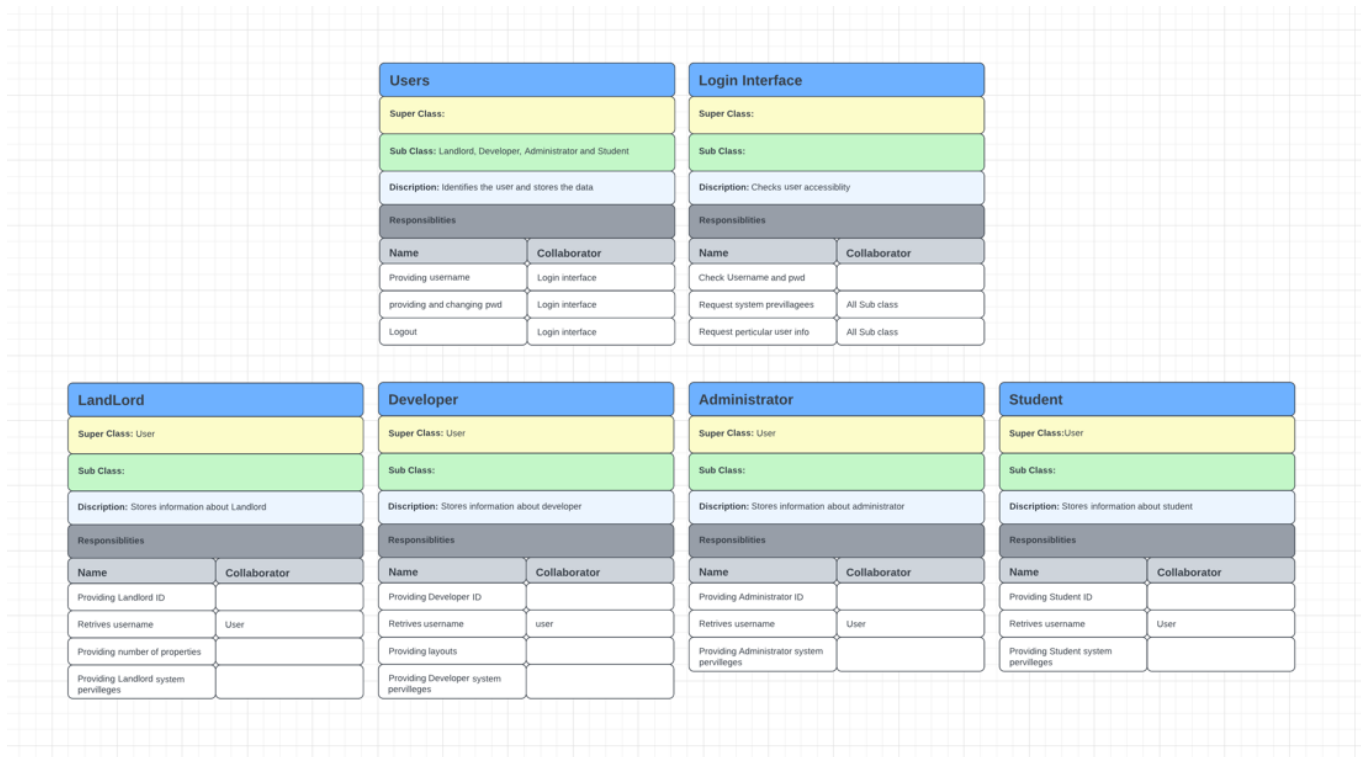


3.1.3 Listing Catalog Subsystem

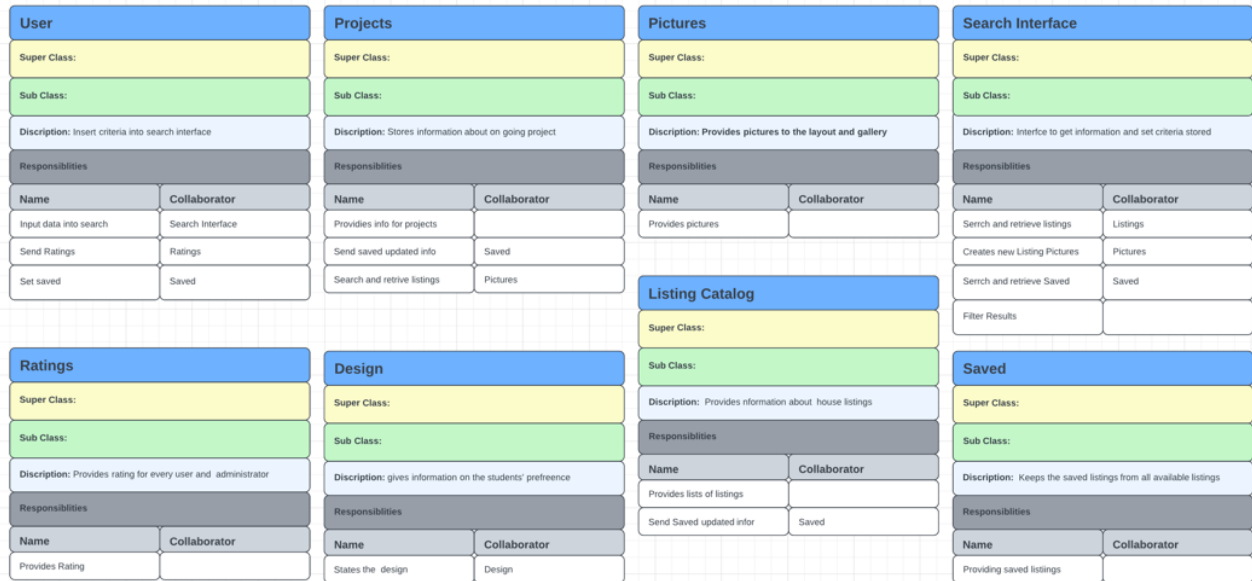


3.2 Class Responsibility Collaboration (CRC)

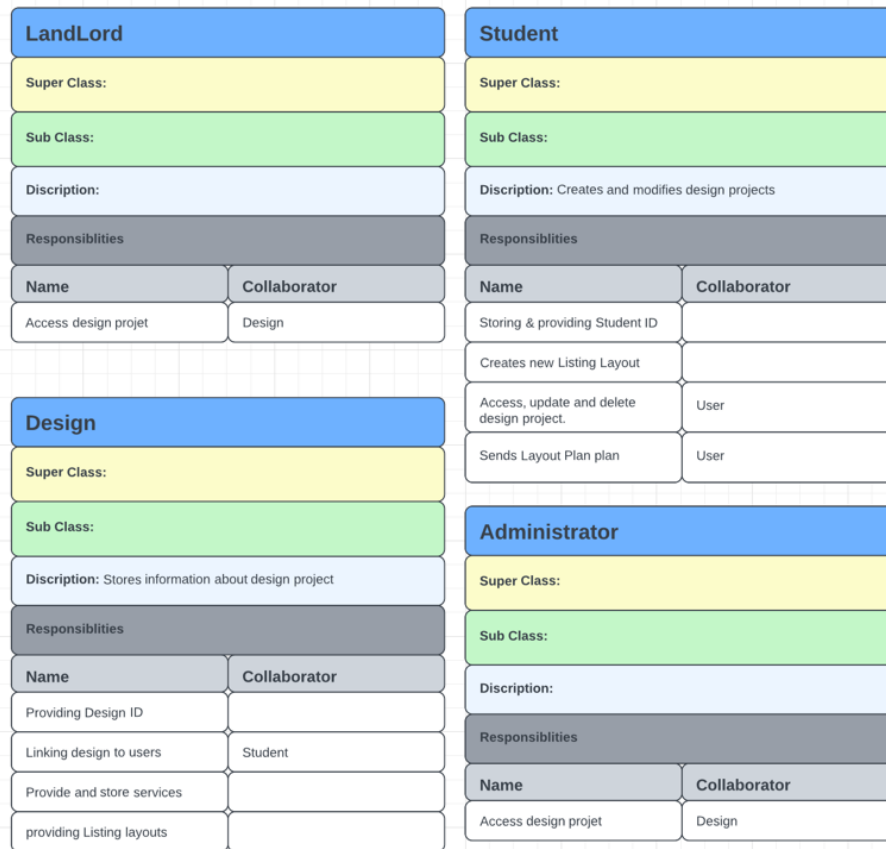
3.2.1 Login Subsystem



3.2.2 Search Subsystem

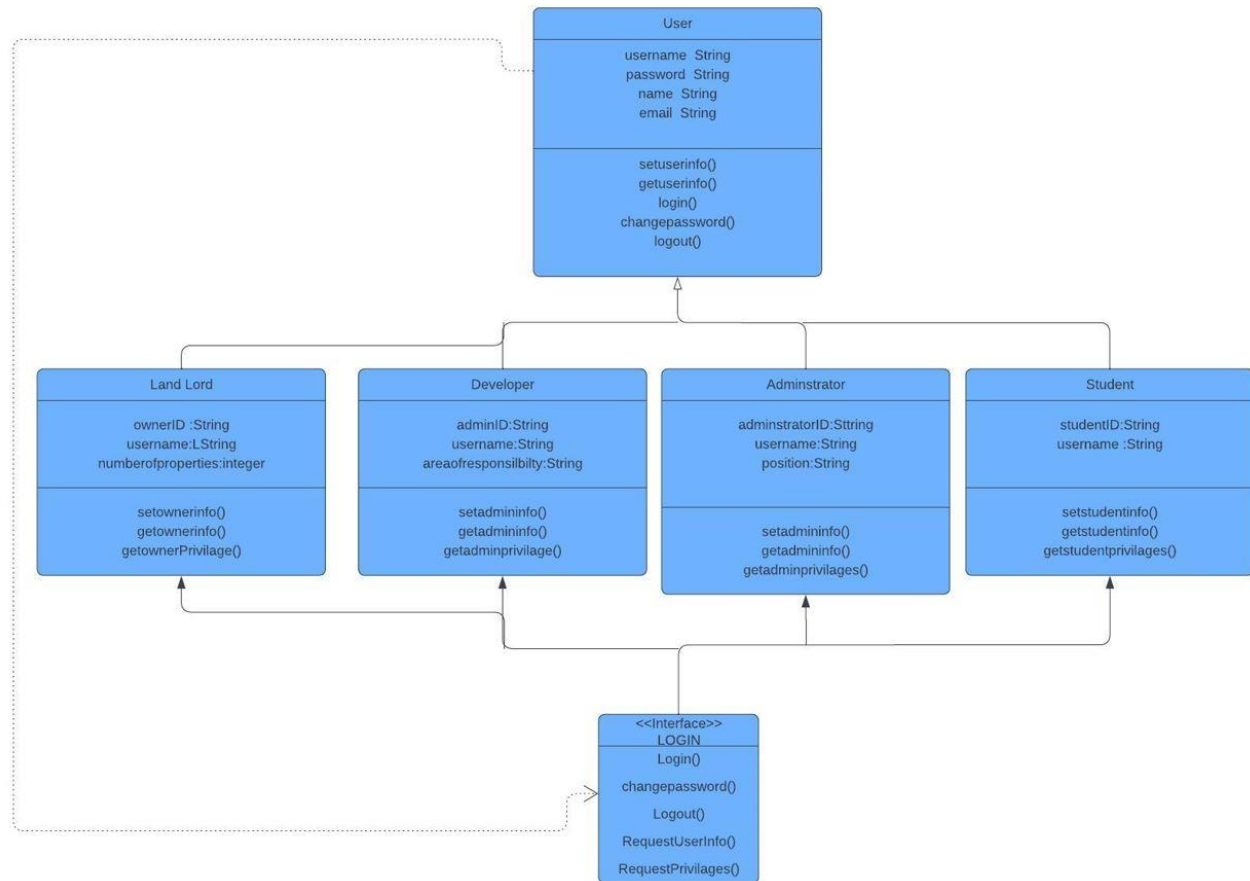


3.2.3 Listing Catalog Subsystem

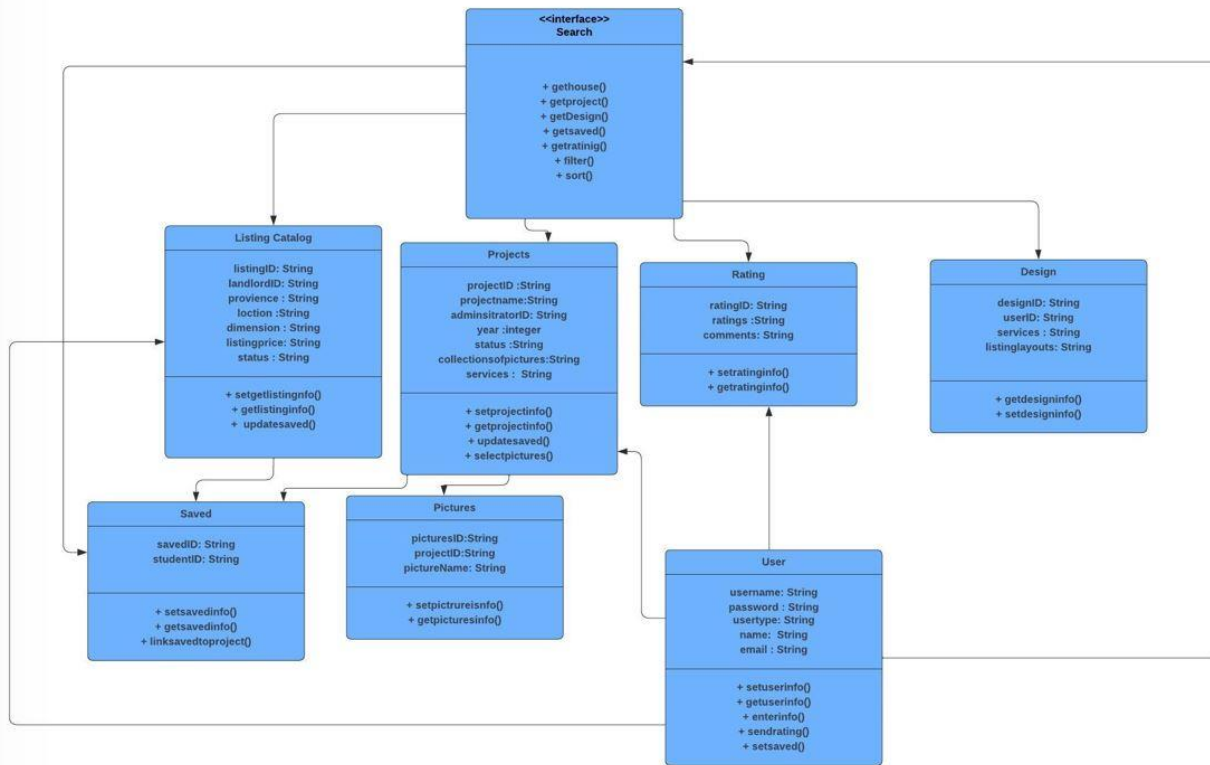


3.3 Design Classes Diagram

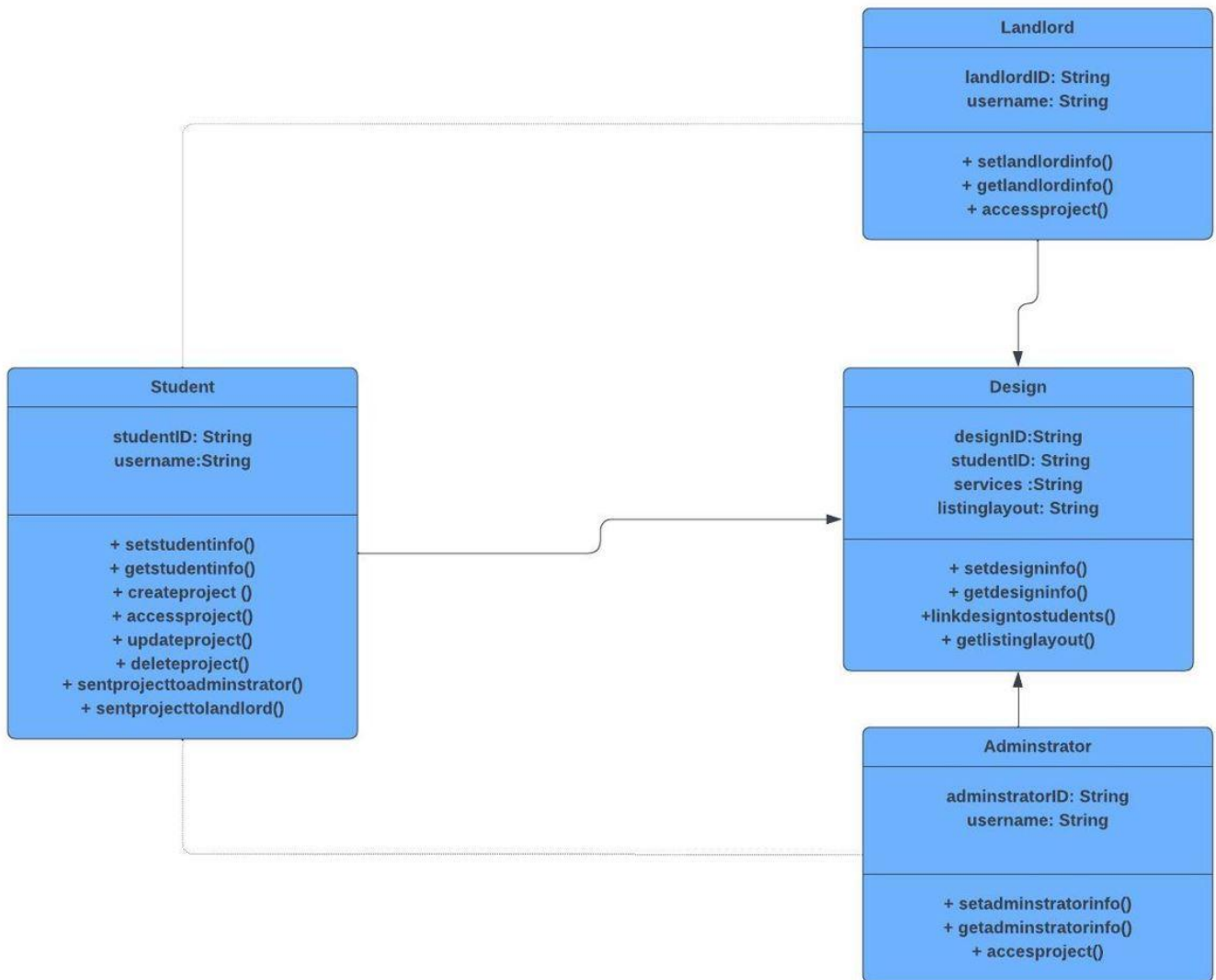
3.3.1 Login Subsystem



3.3.2 Search Subsystem



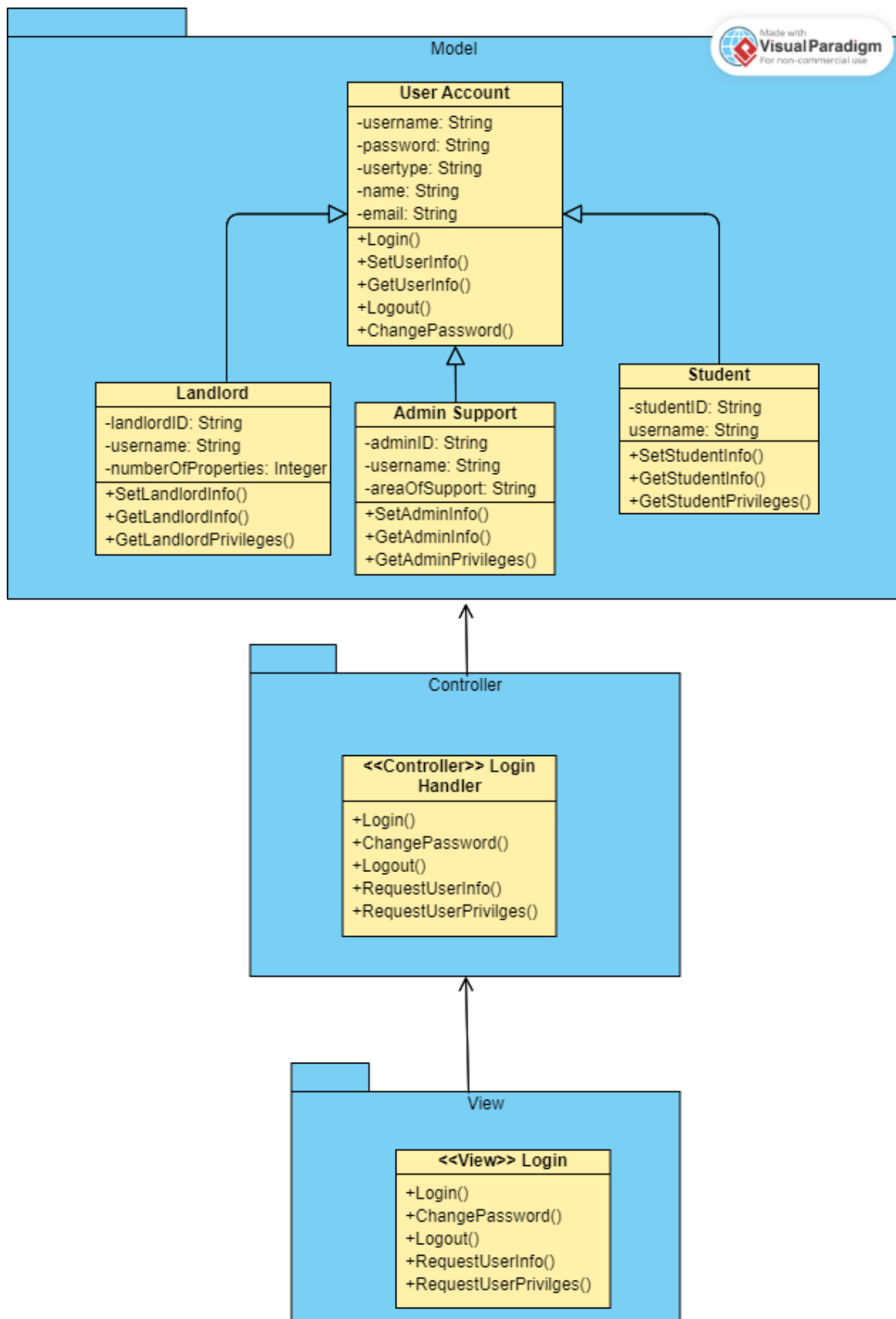
3.3.3 Listing Catalog Subsystem



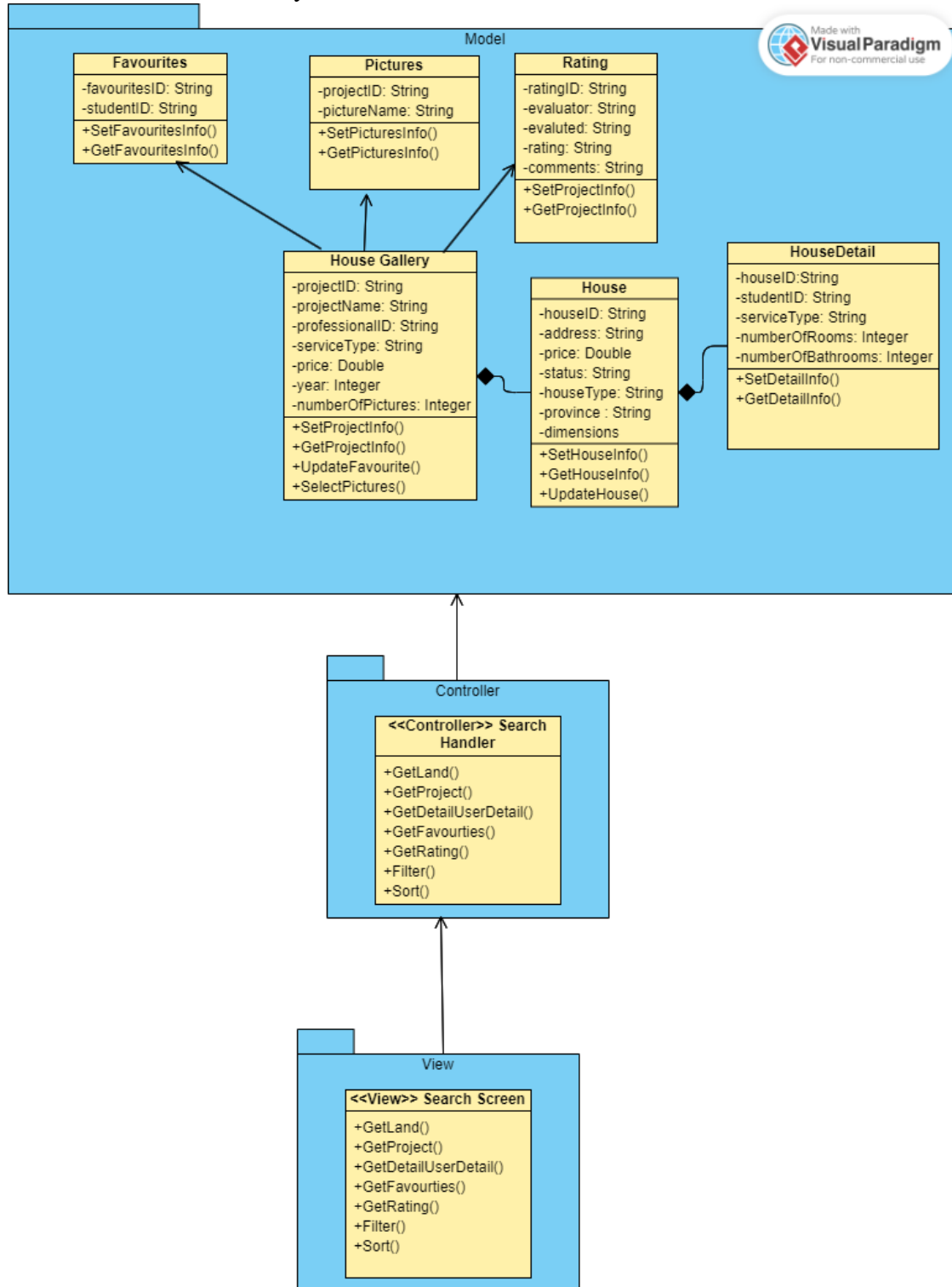
Section 4: Framework M(odel) V(iew) C(ontroller)

4.1 MVC pattern diagram to include

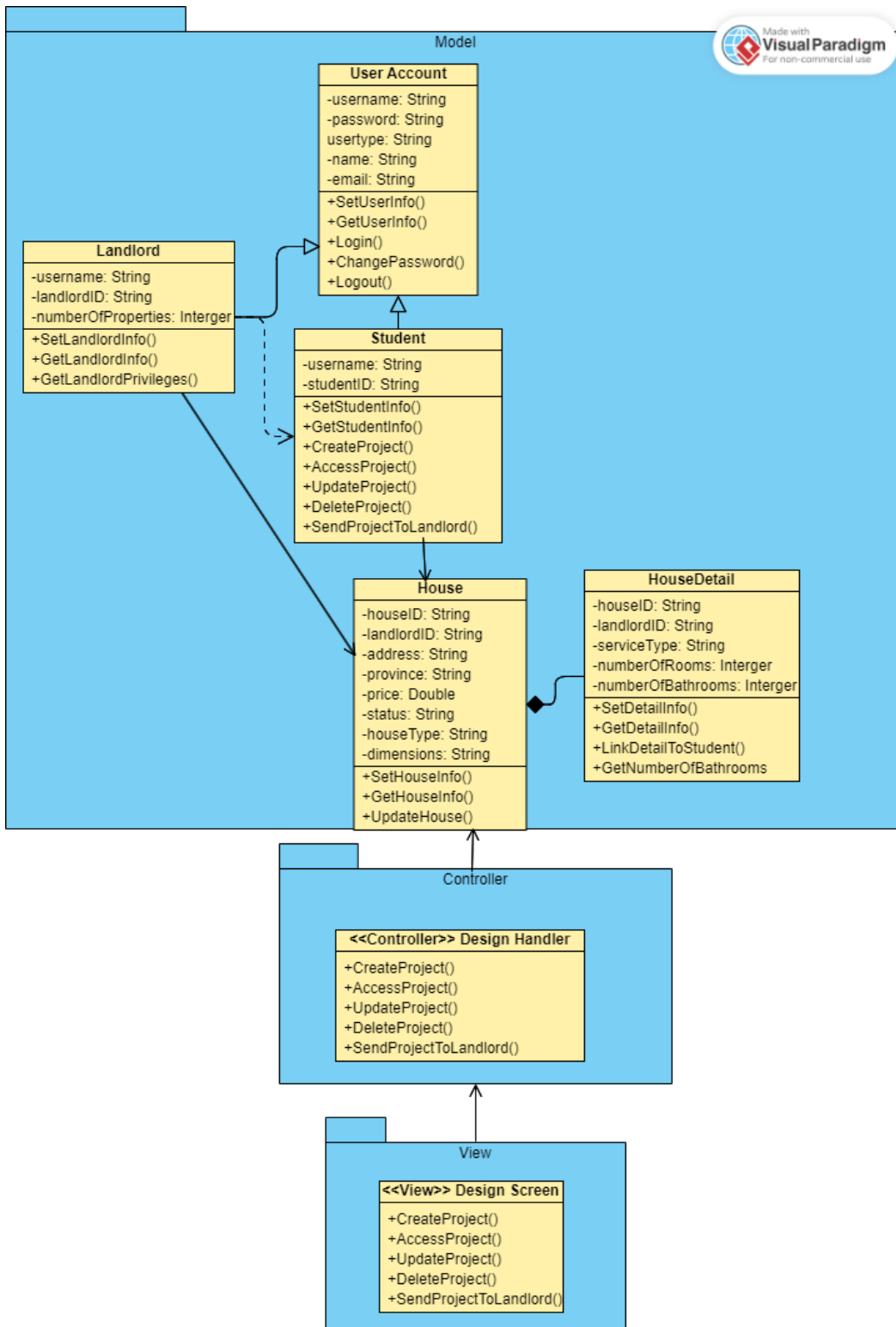
4. 1.1 Login Subsystem



4.1.2 Internal Search Subsystem

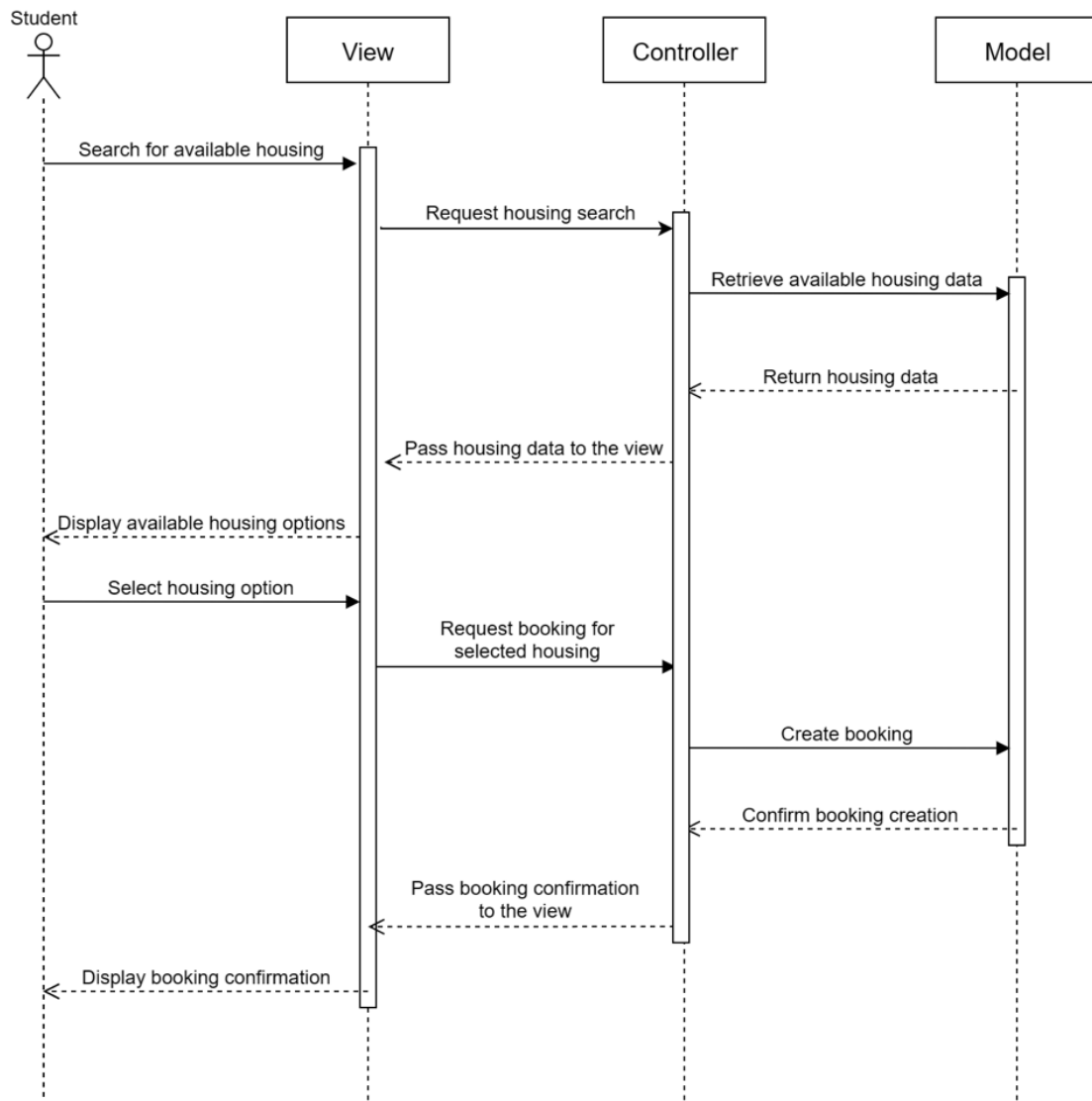


4.1.3 Home Detail Subsystem

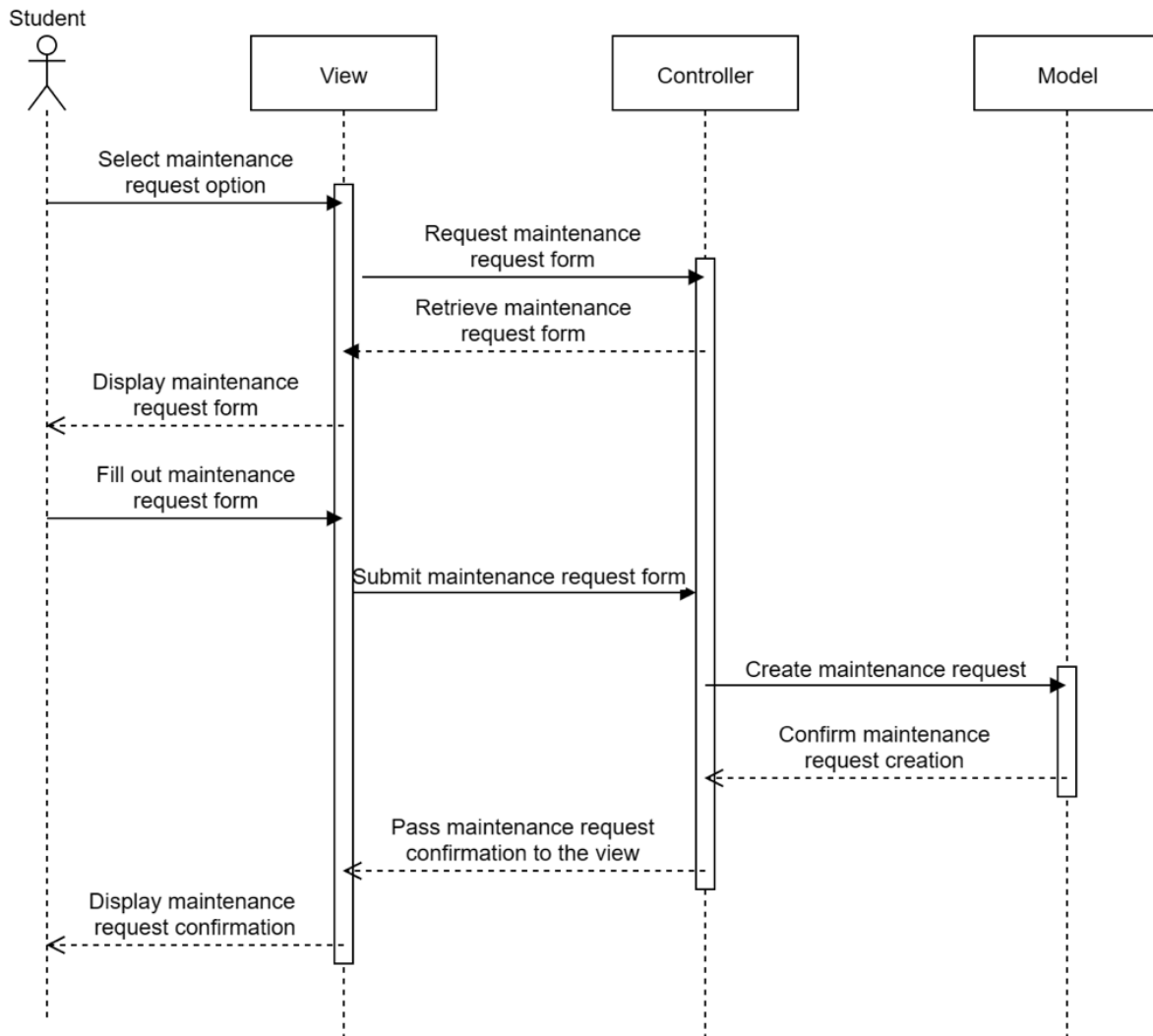


4.2 Full Sequence diagrams

4.2.1. Use case: Search and Book housing (Booking Subsystem)

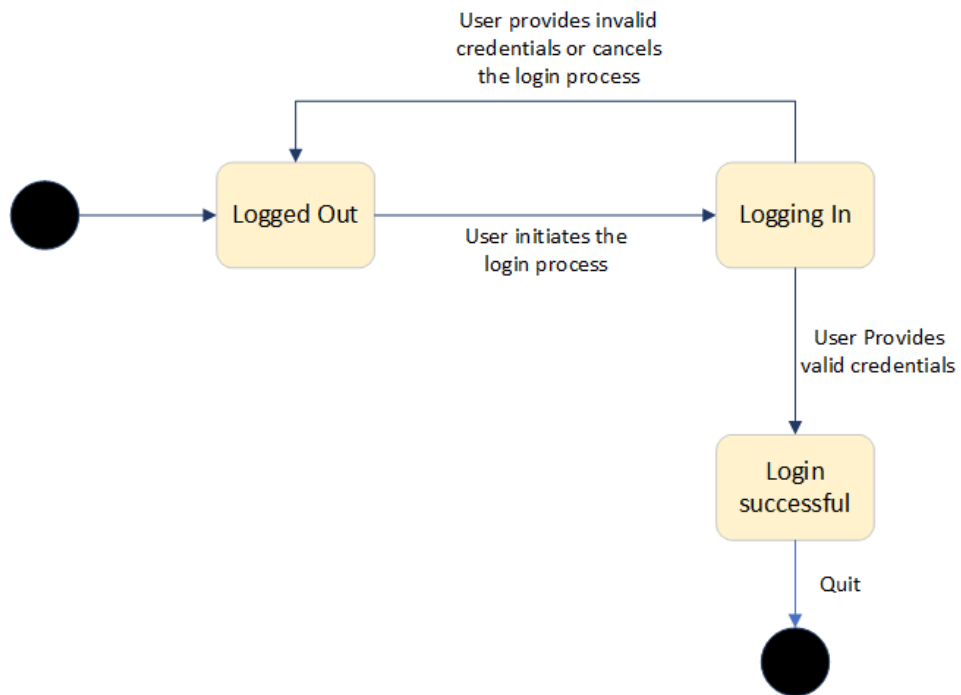


4.2.2. Use cases: Enter information and Retrieve information (Search Subsystem)



4.3 State Machine Diagrams (Needs Improvement – getting confused with flowcharts)

4.3.1. Objects: Login to the system



4.3.2. Objects: Search interface and Rating class (use cases: Enter information and Retrieve information) from Part A

Part C: System Design Documents

Section 1: Corrections do Design Specifications Part B

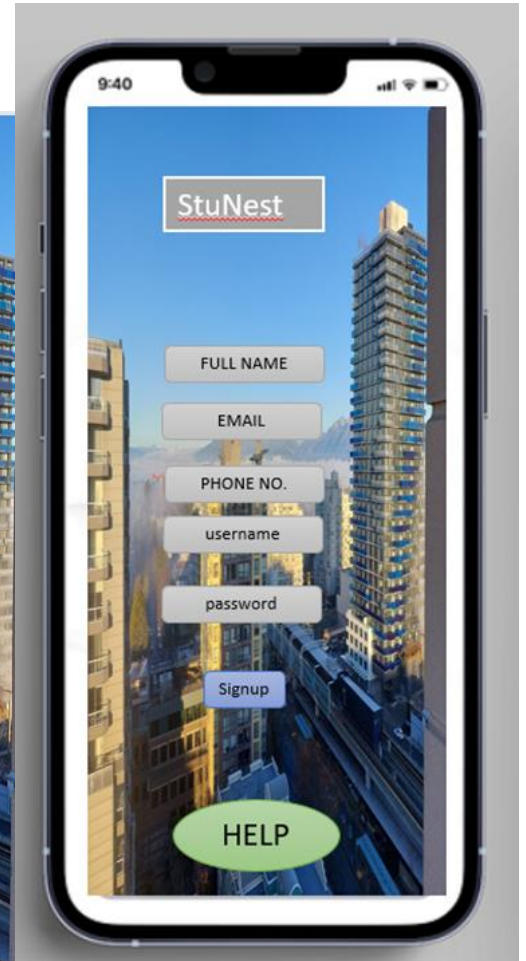
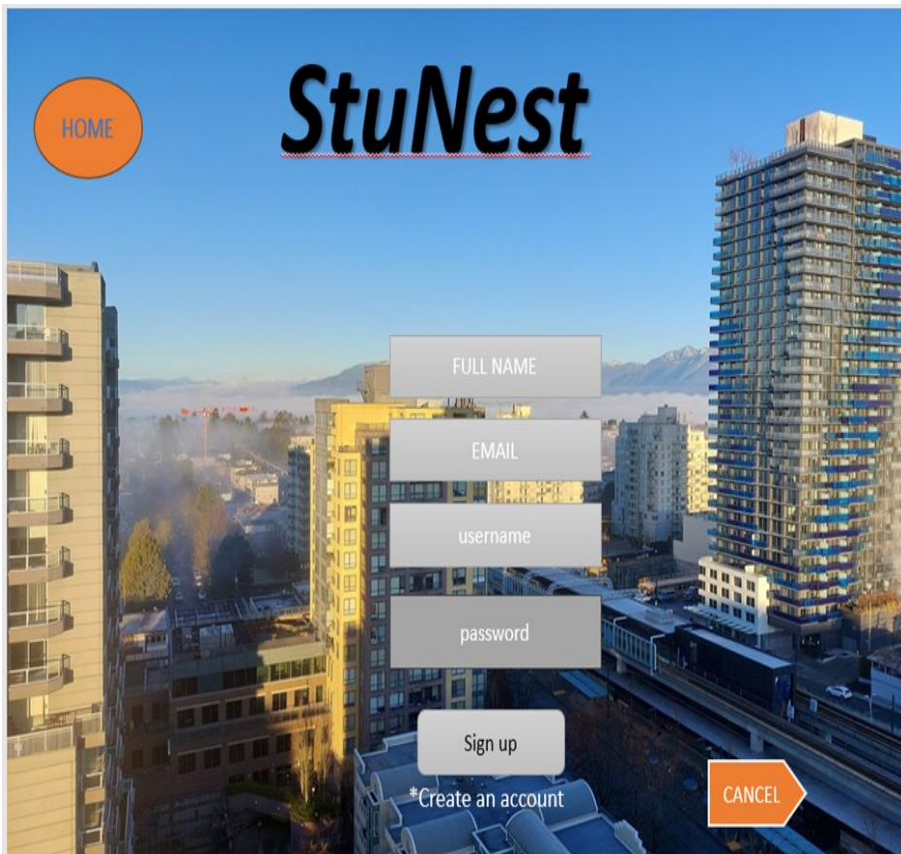
Section 2: Software Design Patterns (Adapter)

Section 3: UI/UX design

3.1 Sign In (login)

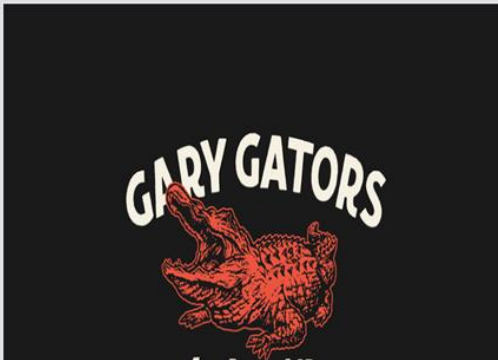


3.2 Sign Up

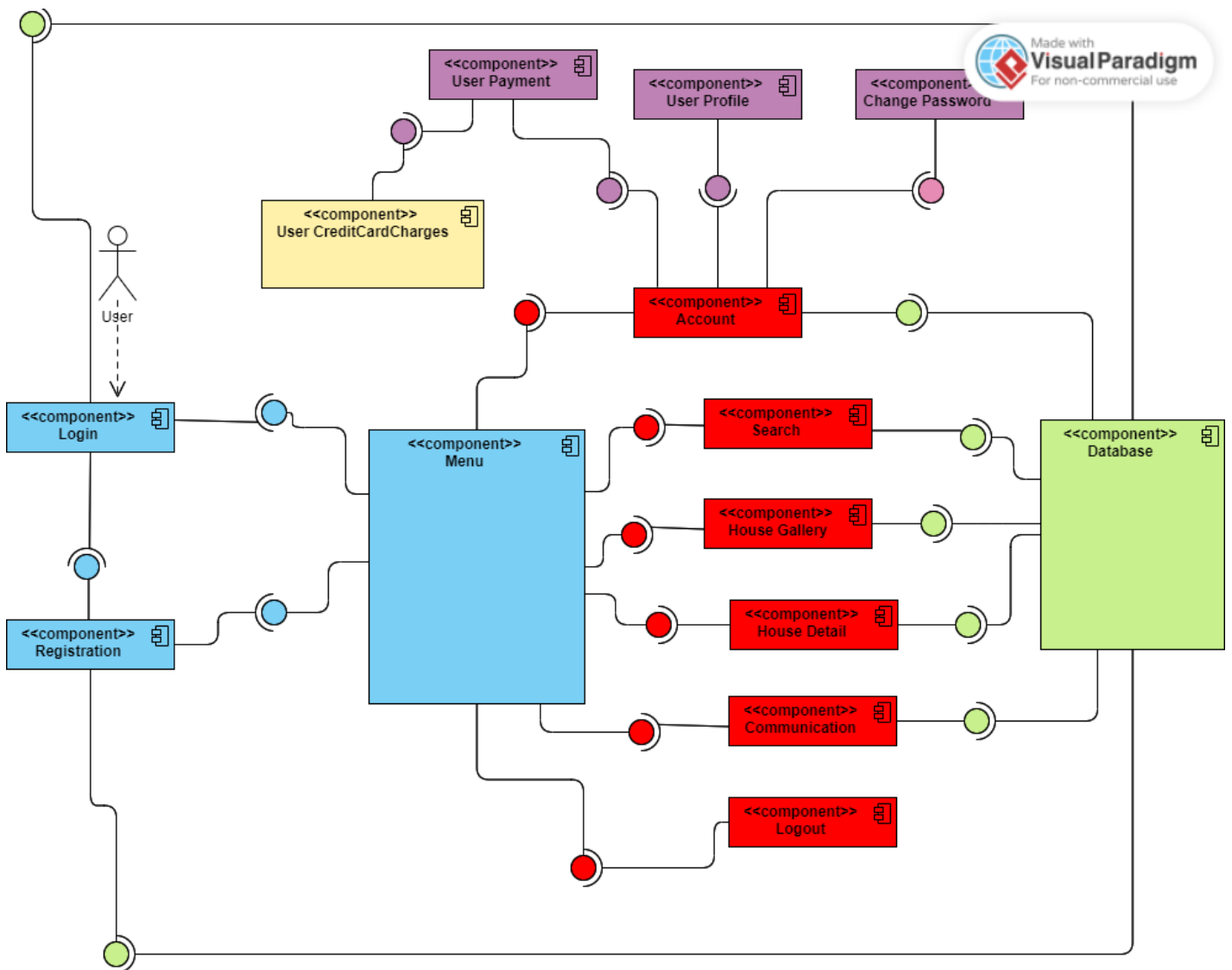


3.3 Home (after login)

FIND YOUR DESIRED SPACE WITH US!



Section 4: High Level Component / Deployment Diagram



Section 5) Gantt Chart

asks		Assigned To	Start Date	End Date	Status	May			Jun			Jul				Aug				Sep						
						Apr 23	Apr 30	May 7	May 14	May 21	May 28	Jun 4	Jun 11	Jun 18	Jun 25	Jul 2	Jul 9	Jul 16	Jul 23	Jul 30	Aug 6	Aug 13	Aug 20	Aug 27	Sep 3	Sep 10
1	Forming Group	Professor	05/10/23	05/10/23	Complete																					
2	Finalizing Company	All	05/12/23	05/19/23	Complete																					
3	Naming Company	All	05/12/23	05/12/23	Complete																					
4	Allotting Part A Tasks	All	05/12/23	05/12/23	Complete																					
5	Part A (sec 1)	Abhishek	05/19/23	05/16/23	Complete																					
	Part A (sec 2)	Zhangir	06/02/23	06/10/23	Complete																					
	Part A (sec 3)	Zhangir	06/02/23	06/10/23	Complete																					
	Part A (sec 4)	Genya	06/02/23	06/10/23	Complete																					
	Part A (sec 5)	Sanghak	06/02/23	06/09/23	Complete																					
	Part A (sec 6)	Ranvir	06/02/23	06/09/23	Complete																					
	Part A (sec 7)	Dhruvit	06/02/23	06/10/23	Complete																					
	Part A (sec 8)	Abhishek	06/02/23	06/10/23	Complete																					
	Part A (sec 9)	Dhruvit	06/02/23	06/11/23	Complete																					
14	Reviewing Part A	All + Professor	06/09/23	06/10/23	Complete																					
15	Submitting Part A	Zhangir	06/11/23	06/11/23	Complete																					
16	Allotting Part B Tasks	All	07/04/23	07/04/23	Complete																					
17	Part B (sec 1)	Zhangir	07/04/23	07/12/23	Complete																					
18	Part B (sec 2)	Abhishek	07/04/23	07/13/23	Complete																					
19	Part B (sec 3)	Dhruvit + Dhruvit	07/04/23	07/12/23	Complete																					
20	Part B (sec 4)	Sanghak + Genya	07/04/23	07/12/23	Complete																					
21	Part B (sec 5)	Ranvir	07/04/23	07/14/23	Complete																					
22	Part B (sec 6)	Dhruvit	07/04/23	07/16/23	Complete																					
23	Reviewing Part B	Professor	07/07/23	07/14/23	Complete																					
24	Submitting Part B	Zhangir	07/16/23	07/16/23	Complete																					
25	Allotting Part C Tasks	All	07/28/23	08/08/23	Complete																					
26	Part C (sec 1)	Zhangir	07/28/23	08/08/23	Complete																					
27	Part C (sec 2)	Ranvir + Dhruvit	07/28/23	08/08/23	Complete																					
28	Part C (sec 3)	Abhishek	07/28/23	08/08/23	Complete																					
29	Part C (sec 4)	Kenya	07/28/23	08/08/23	Complete																					
30	Part C (sec 5)	Dhruvit	08/04/23	08/11/23	Complete																					
31	Presentation Slides	Sanghak	08/09/23	08/11/23	Complete																					
32	Presentation	All	08/11/23	08/11/23	Complete																					
33	Submitting Part C	Dhruvit	08/13/23	08/13/23	In Progress																					