

CSE5450
Software Architecture and Design
Final Project Report

1. Problem

The problem's goal is to design and develop a software app to assist incoming first-year and graduate students at the university. Many incoming first-year students face issues due to being alone for the first time. A student may encounter difficulties in all aspects of life, including academic, social, health, and hygiene. Likewise, many new graduate students face various educational, emotional, social, cultural, and health issues. Without family support, such people would greatly benefit from a system that acts as a daily counselor, recording and observing activities and then offering advice based on evaluating these observations. The data collection should require minimal input from the student.

The user interface allows students to search for help in various categories, including academics, health, social, cultural, emotional/mental issues, and hygiene.

2. Use Cases

2.1. About App

- 2.1.1. The student opens the app and is prompted to create an account.
- 2.1.2. The student inputs their personal information, including their name, school email, and contact information.
- 2.1.3. The student is asked to set up a password, and the app prompts them to ensure it is secure.
- 2.1.4. The student is directed to the main menu, where they can select from the following options:
- 2.1.5. Academic: This option allows students to search for resources and support related to their coursework and academic performance. They can also schedule appointments with academic advisors or tutors.
- 2.1.6. Health: This option allows the student to search for resources and support related to their physical and mental health. They can also schedule appointments with health professionals or access self-help resources.

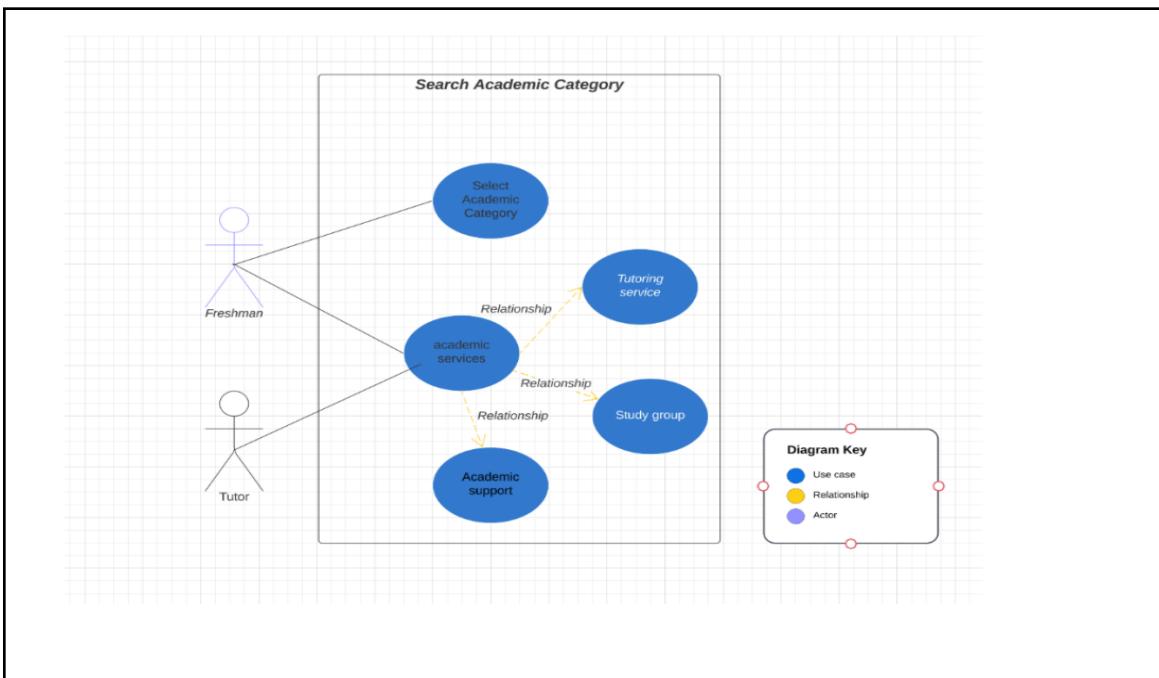
- 2.1.7. Social: This option allows the student to search for resources and support related to making friends and building a social network. They can also access information about campus events and organizations.
- 2.1.8. Cultural: This option allows the student to search for resources and support related to adjusting to a new culture. They can also access information about cultural events and organizations on campus.
- 2.1.9. Emotional/Mental Issues: This option allows the student to search for resources and support related to emotional and mental health. They can also schedule appointments with mental health professionals or access self-help resources.
- 2.1.10. Hygiene: This option allows the student to search for resources and support related to personal hygiene and cleanliness. They can also access information about campus facilities and resources.
- 2.1.11. The student can select one or more options to view the available resources and support.
- 2.1.12. The student can also schedule appointments or access resources directly from the app.
- 2.1.13. The student can also access the app's data collection feature, which allows the app to record and analyze their activities and provide guidance based on the observations. The student can turn this feature on or off at any time.
- 2.1.14. The student can also access the account settings to update their personal information or change their password. The app also has security features to protect student's personal information and keep their data private and secure.

2.2. Support Group Use Case

Use Case Name	Search for Support Group
Scope	Search for Support
Level	User Goal Primary
Primary Actor	Incoming Freshmen and Graduate Students
Stakeholders and Interests	1. Freshman/ Graduate Student: Wants fast and accurate information to be revealed when they search. Wants the information to be relevant to what was searched.

	<p>2. System: Wants quick access to the support group's data to give to the user.</p>
Preconditions	<ol style="list-style-type: none"> 1. The student must have the app installed. 2. The student must be logged in.
Success Guarantee	<ol style="list-style-type: none"> 1. The student has accessed a list of resources related to their category.
Main Success Scenario	<ol style="list-style-type: none"> 1. The student opens the app and is presented with a list of categories (academic, health, social, cultural, emotional/mental issues, hygiene) that they can search for support. 2. They select one of the categories and are presented with a list of resources (e.g., articles, videos, contact info for support services) related to that category.
Exceptions	N / A

2.3. Academic Use Case



Actor Goal List	<ol style="list-style-type: none"> 1. Freshman / Graduate Student: Select Academic Category, find Academic Services, find Academic Support, find a Tutoring Service, and find a Study Group. 2. Tutor: View requests from students for tutoring.
Use Case Name	Search Academic Category
Scope	Academic Category in App
Level	User Goal
Primary Actor	Incoming Freshmen and Graduate Students
Stakeholders and Interests	<ol style="list-style-type: none"> 1. Freshman / Graduate Student: Wants fast and accurate information to be revealed when they search. Wants the information to be relevant to what was searched. 2. Tutors: Wants to update their tutoring services at regular intervals 3. System: Wants to have quick access to academic data to give to the user.
Preconditions	<ol style="list-style-type: none"> 1. The student must have the app installed and be logged in.
Success Guarantee	<ol style="list-style-type: none"> 1. The student has accessed a list of resources related to academic support.
Main Success Scenario	<ol style="list-style-type: none"> 1. The student opens the app and selects the "academic" category from the list of options. 2. They are then presented with a list of academic support resources, such as tutoring services, study groups, and academic advisors.
Exceptions	<ol style="list-style-type: none"> a. System fails: Fails to read search information and retrieve results.

	<ol style="list-style-type: none"> 1. System requests information and retrieves results. 2. Request fails. 3. The system notifies the student of a failed request and gives an error code. 4. Student receives an error message. <p>b. Inaccurate Information: The system needs to have more accurate academic information.</p> <ol style="list-style-type: none"> 1. The student searches for specific academic information. 2. The system retrieves information. 3. The student sees the information needs to be more accurate. 4. The student reports the information as inaccurate using an option presented. 5. The system flags the information for review.
--	--

Use Case Name	Finding Tutoring Services
Scope	Academic Category in App
Level	User Goal
Primary Actor	Incoming Freshmen and Graduate Students
Stakeholders and Interests	<ol style="list-style-type: none"> 1. Freshman / Graduate Student: Wants fast and accurate tutoring info to be revealed when searched. Wants the information to be relevant to what was searched. 2. Tutor: they can communicate with students and solve their doubts in live chat.

	<p>3. System: Wants quick access to the tutor's data to give to the user.</p>
Preconditions	<p>1. The student must have the app installed and be logged in.</p>
Success Guarantee	<p>1. The student has accessed a list of available tutoring services and has contacted a tutor to schedule a session.</p>
Main Success Scenario	<p>1. The student opens the app and selects the "academic" category from the list of options. 2. They then select "tutoring services" from a list of academic support options. 3. The app displays a list of available tutoring services, including information about the subjects offered, the services' location, and the tutors' contact information. 4. The student can then select and contact the tutor to schedule a tutoring session.</p>
Exceptions	<p>a. If the student tries to schedule an appointment with a tutor, but all the slots are already filled, the app will display an error message and prompt the student to try again after some time. b. If the student tries to access an unavailable or broken resource, the app will display an error message and prompt the student to try again later or contact customer support for assistance.</p>

2.4. Health Use Case

<p>The diagram illustrates a UML Use Case Diagram. At the top, a rectangular box labeled "Health Category" contains two blue circles representing use cases: "select medical clinics" and "schedule appointment". A yellow arrow labeled "Relationship" points from "select medical clinics" to "schedule appointment". To the left of the box, a stick figure labeled "Freshman" is connected to "select medical clinics" by a line. To the right, a stick figure labeled "Health Professionals" is connected to "schedule appointment" by a line. A legend box titled "Diagram Key" defines the symbols: a blue circle for "Use case", a yellow arrow for "Relationship", and a stick figure for "Actor".</p>	
Actor Goal List	<ol style="list-style-type: none"> 1. Freshman / Graduate Student: Select medical clinics, and schedule appointments. 2. Health Professionals: Receive Appointments, Update Schedule on the App
Use Case Name	Search Health Category
Scope	Finding Medical Clinics
Level	User Goal
Primary Actor	Incoming Freshmen and Graduate Students
Stakeholders and Interests	<ol style="list-style-type: none"> 1. Freshman / Graduate Student: Wants fast and accurate medical clinic information to be revealed when searched. Wants the information to be relevant to what was searched. 2. Health professionals: can update or make student appointments. 3. System: Wants to have quick access to health data for the user.

Preconditions	<ol style="list-style-type: none"> 1. The student must have the app installed and be logged in.
Success Guarantee	<ol style="list-style-type: none"> 1. The student has accessed a list of available medical clinics and has contacted a clinic to schedule an appointment.
Main Success Scenario	<ol style="list-style-type: none"> 1. The student opens the app and selects the “health” category from the list of options. 2. They then select “medical clinics” from health support options. The app displays a list of available medical clinics, including information about the types of services offered, the location of the clinics, and the contact information for the clinics. 3. The student can then select a clinic and contact the clinic to schedule an appointment.
Exceptions	<ol style="list-style-type: none"> a. If the student tries to schedule an appointment with a health professional from a selected clinic, but all the slots are already filled, the app will display an error message and prompt the student to try again later. b. If the student tries to access an unavailable or broken resource, the app will display an error message and prompt the student to try again later or contact customer support for assistance.

2.5. Social Use Case

<p>Social Category</p> <p>Diagram Key</p> <ul style="list-style-type: none"> Use case (Blue Circle) Relationship (Yellow Circle) Actor (Purple Person) <pre> graph TD Freshman((Freshman)) --- R1(()) R1 --- UC1((select campus organization or create group)) R1 --- UC2((campus organizations events/groups)) R1 --- CampusOrgs((Campus organizations)) R1 --- SSS((Student support services)) UC1 --- SSS UC2 --- CampusOrgs CampusOrgs --- R2(()) R2 --- SSS </pre>	
Actor Goal List	<ol style="list-style-type: none"> 1. Freshman / Graduate Student: Select Campus Organization or create a group; find information on campus organizations events/groups. 2. Campus Organizations: Upload information on campus organizations events/groups. 3. Student Support Services: Find information on campus organizations, events/groups.
Use Case Name	Meeting or making new friends
Scope	Meeting New Friends
Level	User Goal
Primary Actor	Incoming Freshmen and Graduate Students
Stakeholders and Interests	<ol style="list-style-type: none"> 1. Freshman / Graduate Student: Wants fast and accurate social information to be revealed when searched. Wants the information to be relevant to what was searched. 2. Organizations: can create and update social events groups.

	<ol style="list-style-type: none"> 3. System: Wants to have quick access to social data for the user.
Preconditions	<ol style="list-style-type: none"> 1. The student must have the app installed and be logged in.
Success Guarantee	<ol style="list-style-type: none"> 1. The students have made some connections and made new friends.
Main Success Scenario	<ol style="list-style-type: none"> 1. The student opens the app and navigates to the “Social” group, and selects the option to meet new friends. 2. The app prompts the students to fill out a short survey about their interests and hobbies, and the app matches the student with other students with similar interests and creates a chat room for the group. 3. The students can now communicate with their new friends and plan activities together.
Exceptions	<ol style="list-style-type: none"> a. If a student tries to access information about a matching friend, but it gives details about a person with different interests, or else the information is unavailable, the app will display an error message and prompt the student to try again later or contact the customer service for assistance.

2.6. Cultural Use Case

<pre> classDiagram actor FS { Freshman or Graduate Student } actor CEO { Cultural Event Organizers } actor S { <<System>> } useCase SC { Cultural Category } useCase FE { Find Event } useCase SE { Search } useCase CE { Create Events } FS->>FE FS->>SE CEO->>CE FE-->> SC : <<include>> FE-->> S SE-->> S CE-->> S </pre>	
Actor Goal List	<ol style="list-style-type: none"> 1. Freshman / Graduate Student: Search Category, Find Cultural Events. 2. Cultural Event Organizers: Create Events 3. System: Produce Search Results, Store Cultural Information, and Update Cultural Information.
Use Case Name	Find a Cultural Event
Scope	Cultural Category in App
Level	User Goal
Primary Actor	Incoming Freshmen and Graduate Students
Stakeholders and Interests	Freshman / Graduate Student: Wants to see details from the specific item selected from the list of cultural information on campus.

	<p>Event Organizers on Campus: Wants to have their information reach students using the app and knowledge to be accurate.</p> <p>System: Wants to have quick access to cultural data for the user.</p>
Preconditions	<ol style="list-style-type: none"> 1. The student is logged into the app.
Success Guarantee	<ol style="list-style-type: none"> 1. The student can find a cultural event.
Main Success Scenario	<ol style="list-style-type: none"> 1. The student searches the cultural category (refer to 1st use case). 2. The student selects an event from the list of cultural information presented. 3. System displays full details of the event selected. 4. The student can find the details presented to find the cultural event.
Exceptions	<ol style="list-style-type: none"> a. System fails. <ol style="list-style-type: none"> i. Search Failure (Refer to Use case 1, extensions, the system fails). ii. System crashes. <ol style="list-style-type: none"> 1. Student is kicked out of the app. 2. System saves the last action of the student. 3. The student logs back into the app. 4. Students are placed where they left off at the saved point.

2.7. Emotional Use Case

Actor Goal List	<ol style="list-style-type: none"> 1. Freshman / Graduate Student: Search Category, Schedule Appointment. 2. Health Professionals: Receive Accurate Schedule, Update schedule on the system 3. System: Produce Search Results, Store Emotional Information, Update Emotional Information, and Update Schedule for Health Professionals.
Use Case Name	Schedule Appointment with Campus Health Professionals.
Scope	Emotional Category
Level	User Goal
Primary Actor	Incoming Freshmen and Graduate Students
Stakeholders and Interests	<ol style="list-style-type: none"> 1. Freshman / Graduate Student: Wants accurate information on health professionals and schedules. 2. Health Professional: Wants to be notified of scheduled appointments.
Preconditions	<ol style="list-style-type: none"> 1. The student is logged in on the app. 2. The student is in the Emotional category.

Success Guarantee	<ol style="list-style-type: none"> 1. The student can schedule an appointment with a health professional.
Main Success Scenario	<ol style="list-style-type: none"> 1. The student selects an appointment with a health professional option. 2. The system displays a list of professionals to choose from that the campus provides. 3. The student selects a health professional. 4. The system asks the student to select an available date to set the appointment. 5. The student selects a date for the appointment. 6. The system asks the user to fill out the information the health professional needs to confirm the appointment request. 7. The student fills out the form and submits it. 8. The system verifies the information is valid and sends the information to the health professionals.
Exceptions	<ol style="list-style-type: none"> a. scheduling issues. <ol style="list-style-type: none"> i. Failed to represent an accurate schedule availability. <ol style="list-style-type: none"> 1. The system asks the student to input a date for an appointment based on available days. 2. The date and time selected were falsely shown as available. 3. The health professional is notified of the appointment. 4. The health professional verifies the appointment time. 5. Health Professional notifies the student through the system that the appointment time needs to be changed to an available slot. 6. The student changes the appointment time and date.

	<p>b. System fails.</p> <p>i. Information received by the student needs to be validated.</p> <ol style="list-style-type: none"> 1. The student fills out the form and submits it. 2. The system verifies the form information is valid. 3. The information needs to be validated. 4. The system provides an error message with the necessary steps to correct the invalid information.
--	--

2.8. Hygiene Use Case

	<pre> graph TD subgraph HygieneCategory [Hygiene Category] direction TB S1((Search)) S2((View Campus Resources and Facilities)) S1 --- F1[Freshman or Graduate Student] S2 --- C1(Campus) S1 --- Syst[<<actor>> System] S2 --- Syst Syst --- S1 Syst --- S2 S1 --- S2 end </pre>
Actor Goal List	<ol style="list-style-type: none"> 1. Freshman / Graduate Student: Search Category, Find Hygiene Events. 2. System: Produce Search Results, Store Hygiene Information, and Update Hygiene Information. 3. Campus: Upload Campus Resources and Facilities Information.

Use Case Name	Access information on Campus Resources and Facilities
Scope	Hygiene Category in App
Level	User Goal
Primary Actor	Incoming Freshmen and Graduate Students
Stakeholders and Interests	<ol style="list-style-type: none"> 1. Freshman / Graduate Student: Wants quick and accurate information on campus resources and facilities. 2. Campus: Wants accurate information to be presented to students about campus resources and facilities.
Preconditions	<ol style="list-style-type: none"> 1. The student is logged into the app. 2. The student is in the Hygiene Category.
Success Guarantee	<ol style="list-style-type: none"> 1. The student can find information on campus resources and facilities.
Main Success Scenario	<ol style="list-style-type: none"> 1. The system displays a selection option for viewing Campus Resources and Facilities information. 2. The student selects an option to see Campus Resources and Facilities information. 3. System displays accurate information about the campus resources and facilities. 4. Student reads the information.
Exceptions	<ol style="list-style-type: none"> a. If the student tries to access info about a campus facility or resource, but the information is not available or broken, the app will display an error message and prompt the student to try again later or contact customer support for assistance.

2.9. Tracking Daily Activities

Use Case Name	Tracking Daily Activities
Scope	Tracking Daily Activities
Level	User Goal
Primary Actor	Incoming Freshmen and Graduate Students
Stakeholders and Interests	<ol style="list-style-type: none"> 1. Freshman/Graduate Student: Wants accurate information about the category they are interested in. 2. Student Support Services: They want to present accurate information to the student about their data. 3. Data analyst: The data analysts use the data collected by the app to identify trends and patterns to help support students.
Preconditions	<ol style="list-style-type: none"> 1. The student must have the app installed and be logged in.
Success Guarantee	<ol style="list-style-type: none"> 1. The students' daily activities have been recorded and analyzed to provide guidance.
Main Success Scenario	<ol style="list-style-type: none"> 1. The student is encouraged to input information about their daily activities, such as how many hours they slept, how many hours they studied, how many hours they exercised, etc. 2. This info is recorded and analyzed to provide guidance and recommendations to the student.

Exceptions	N / A
-------------------	-------

2.10. Security

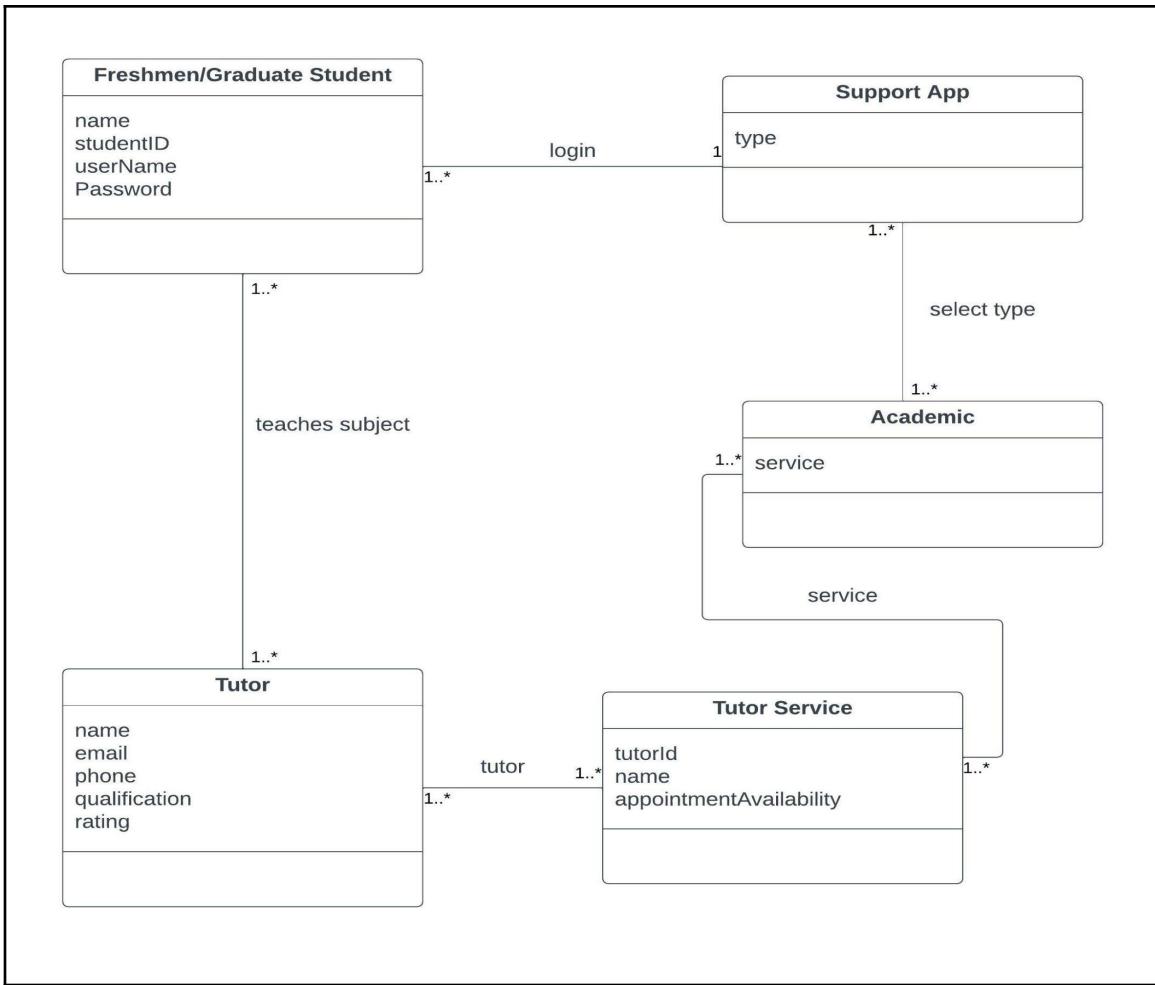
Use Case Name	Security
Scope	Ensuring Data Security
Level	User Goal
Primary Actor	Incoming Freshmen and Graduate Students
Stakeholders and Interests	<ol style="list-style-type: none"> 1. Freshman/Graduate student: Students have a stake in protecting and securing their info. They expect their privacy to be respected and for the app to maintain the confidentiality of their data. 2. Data analysts: The data analysts use the data collected by the app to identify trends and patterns to help support students. 3. Developer and maintenance team: These individuals are responsible for designing, developing, and maintaining the app. 4. Organization responsible for the app: They are responsible for overseeing the development and use of the app and may be held legally liable for any security breaches or data breaches that occur. 5. Student support services: These are responsible for supporting the students and accessing the activity data to provide guidance and support.

Preconditions	<ol style="list-style-type: none"> 1. The app must be installed and configured with security settings.
Success Guarantee	<ol style="list-style-type: none"> 1. The student's personal and daily activity data is protected from unauthorized access.
Main Success Scenario	<ol style="list-style-type: none"> 1. The app is designed with security measures to protect the student's personal info and daily activity data. 2. This includes measures such as encryption, password protection, and restricted access to data.
Exceptions	N / A

3. Domain Models

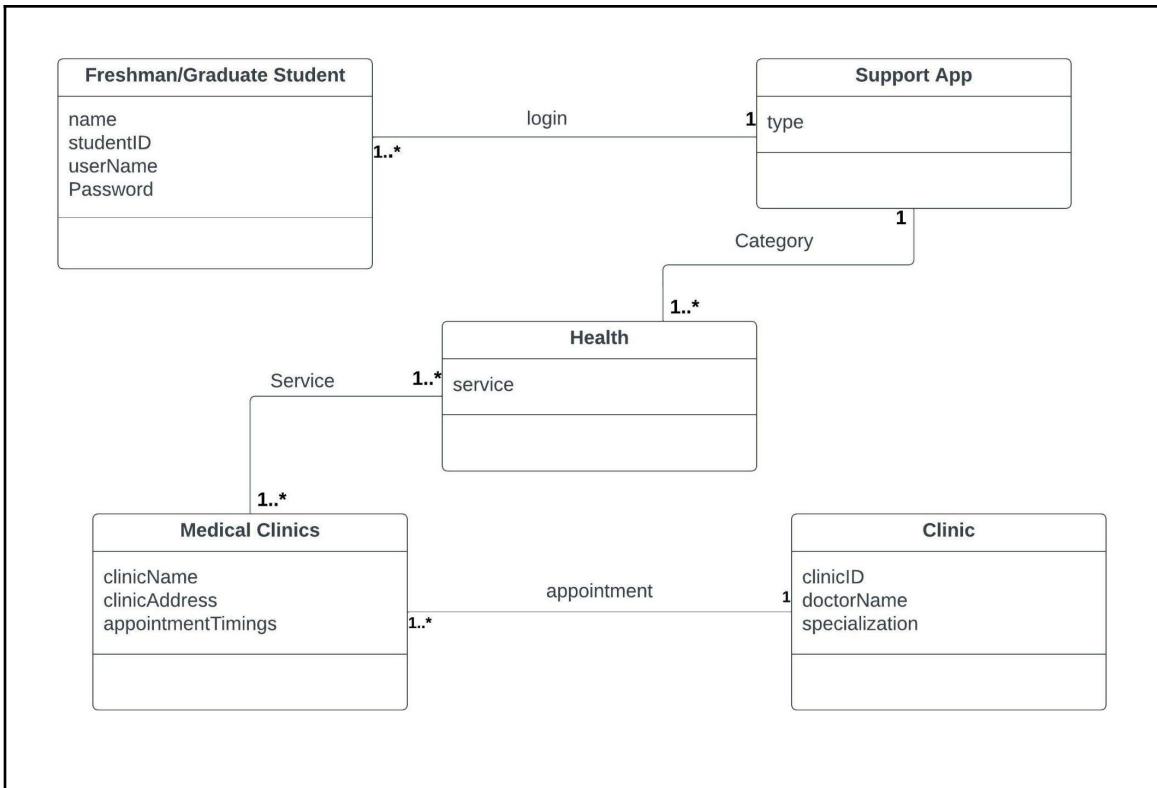
3.1. Academics

Domain Model for Academics

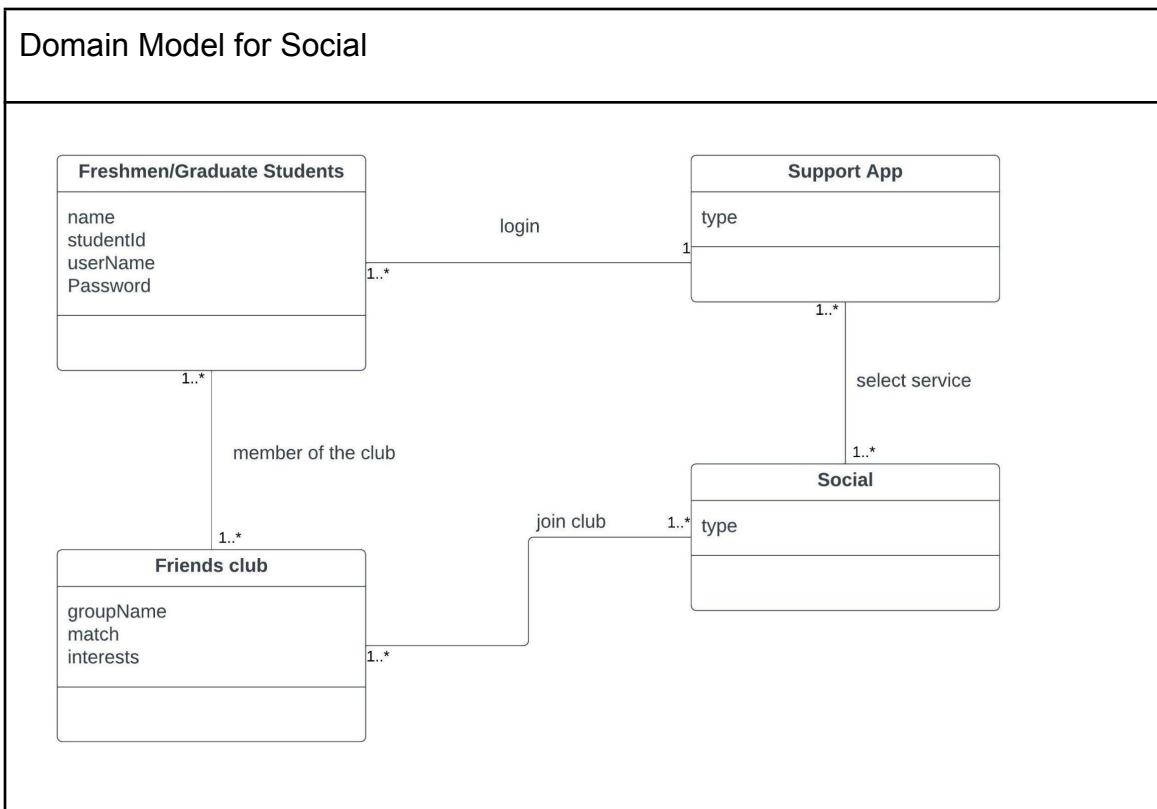


3.2. Health

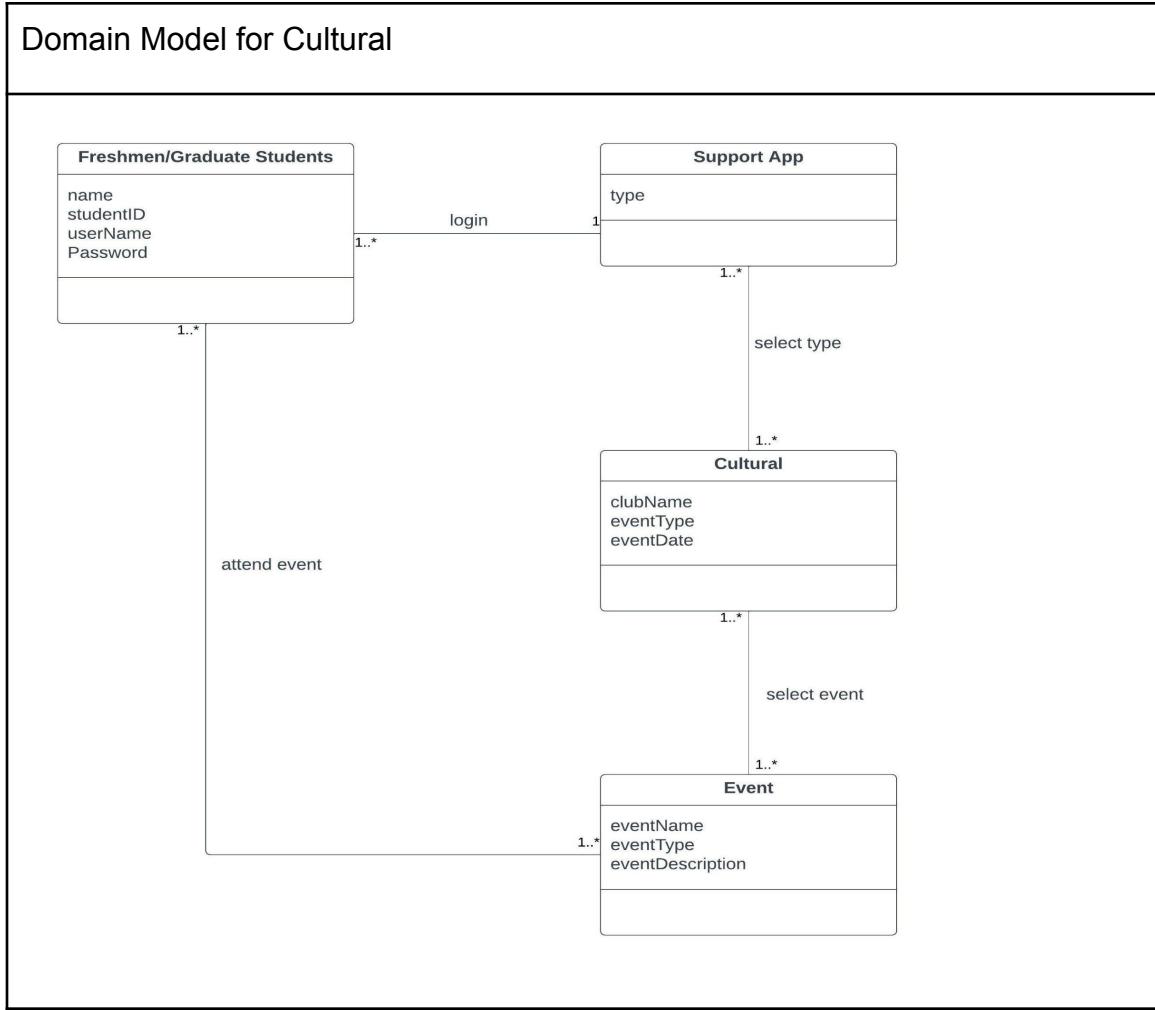
Domain Model for Health



3.3. Social

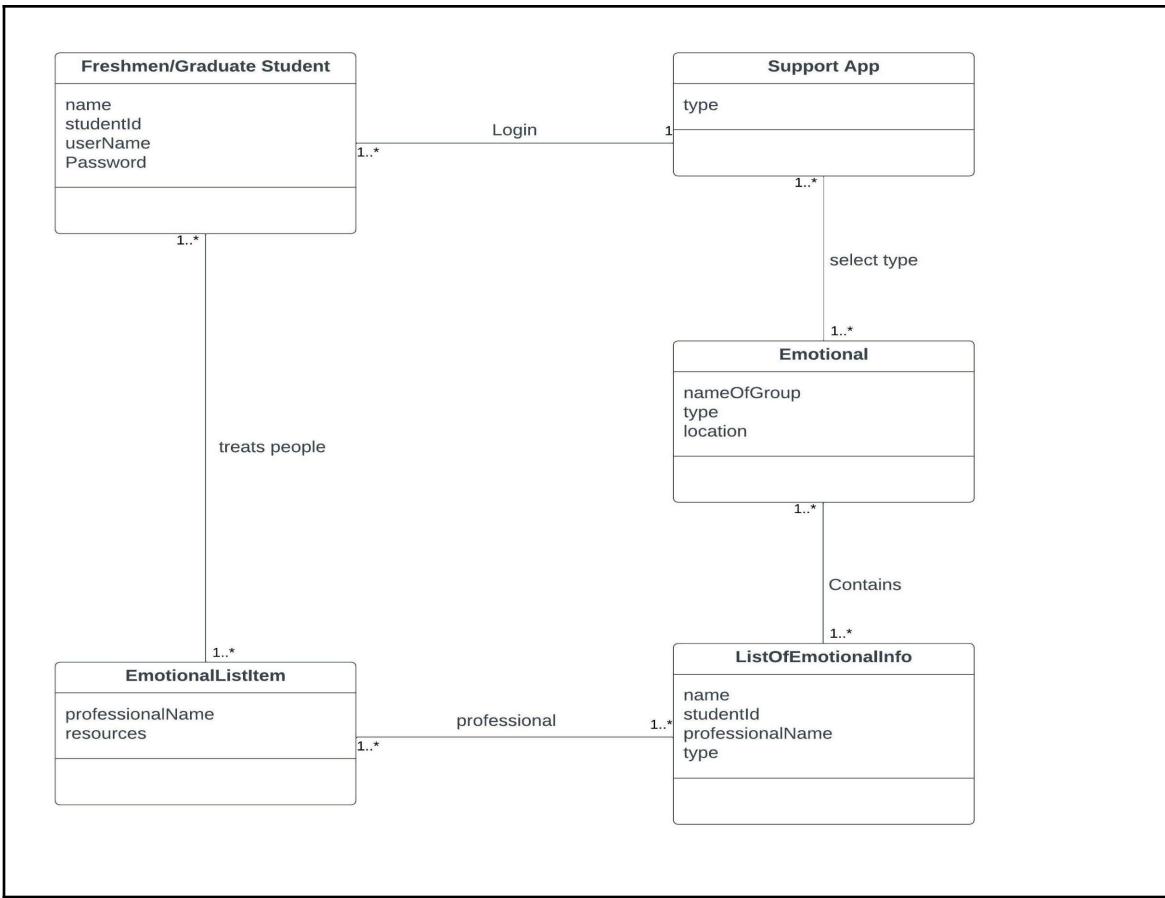


3.4. Cultural



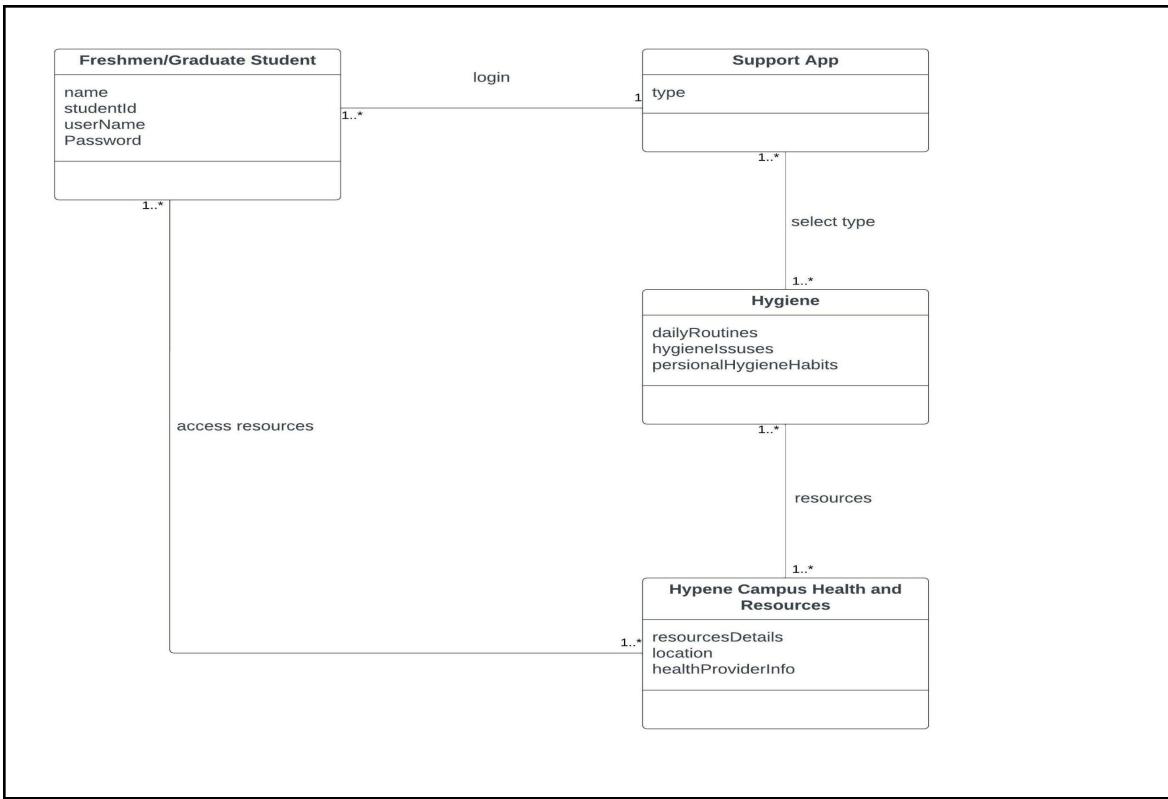
3.5. Emotional





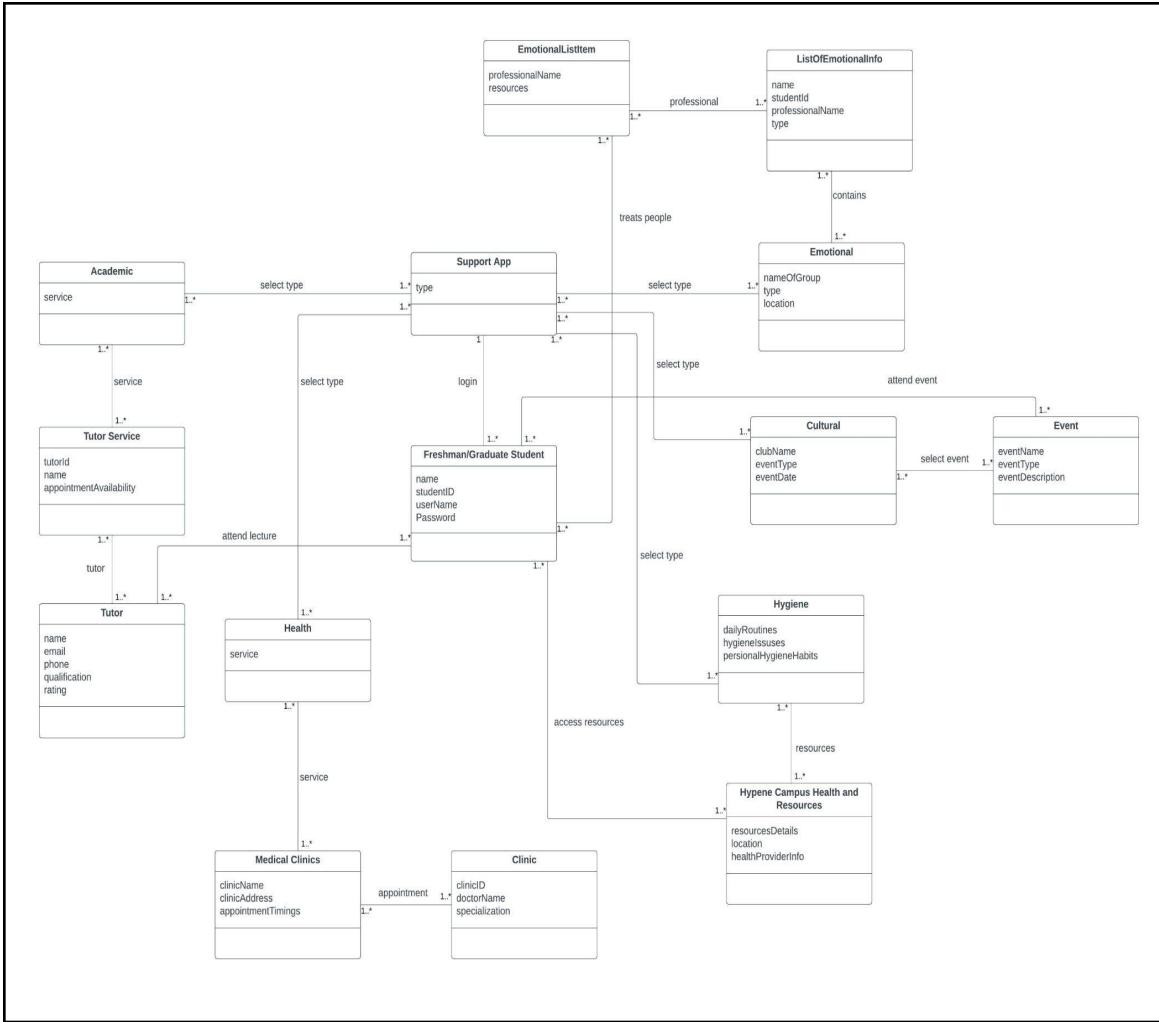
3.6. Hygiene

Domain Model for Hygiene



3.7. Overall Domain Model

Domain Model For App

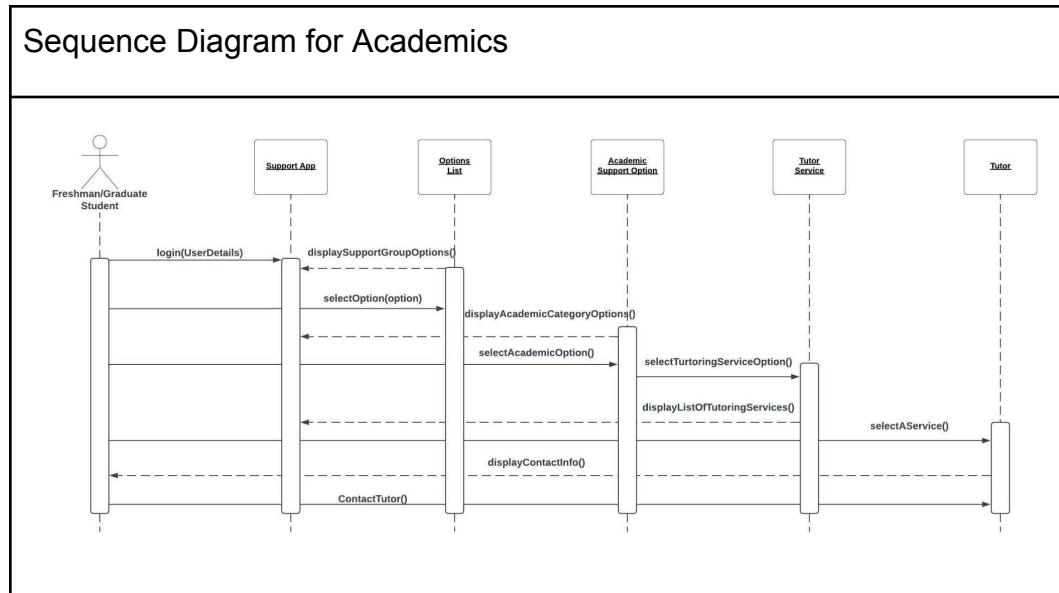


4. System Sequence Diagram

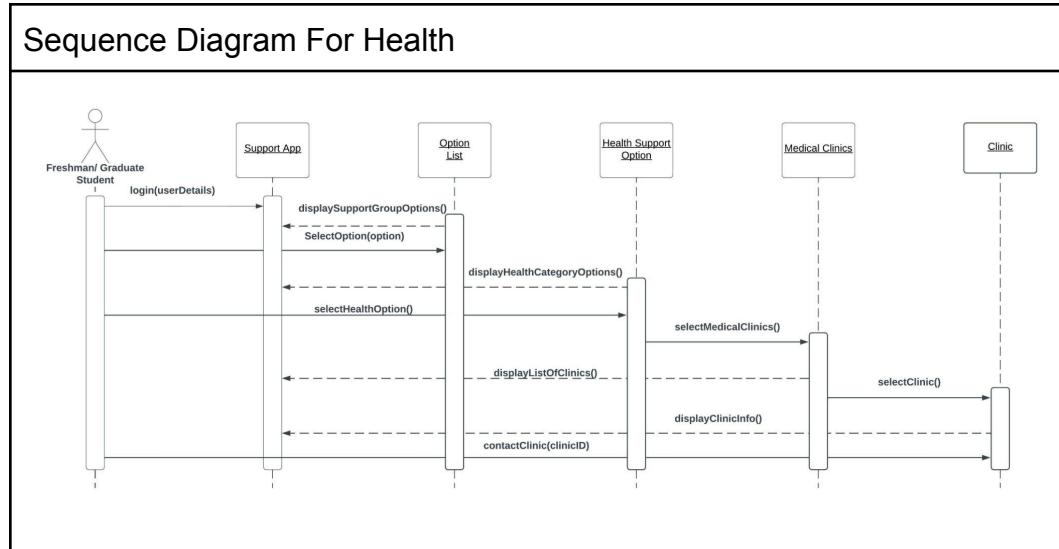


5. Sequence Diagrams

5.1. Academics

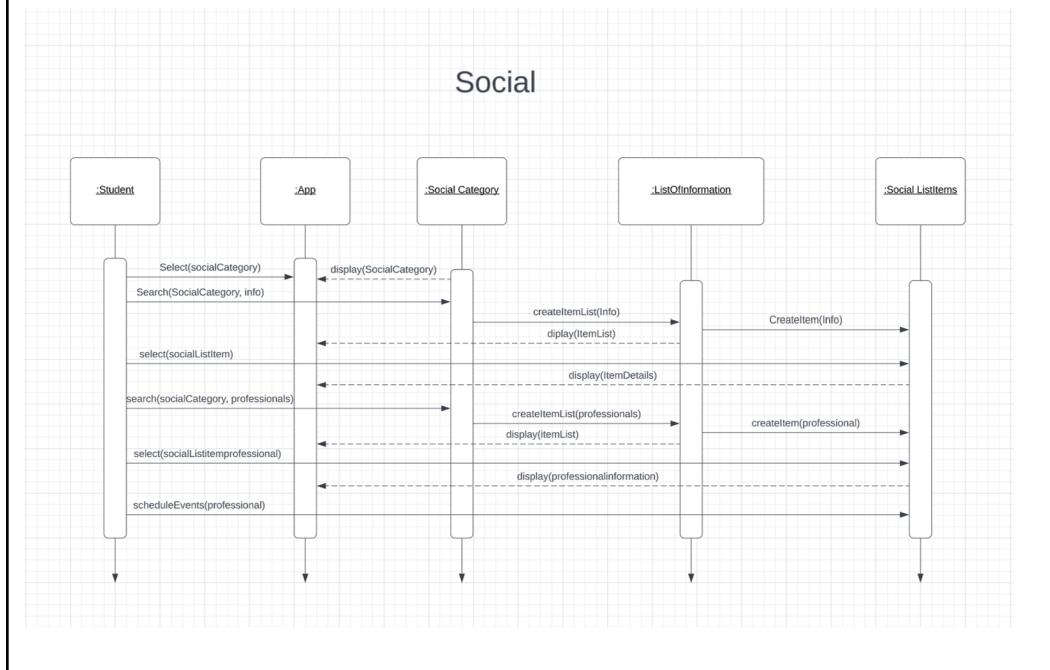


5.2. Health



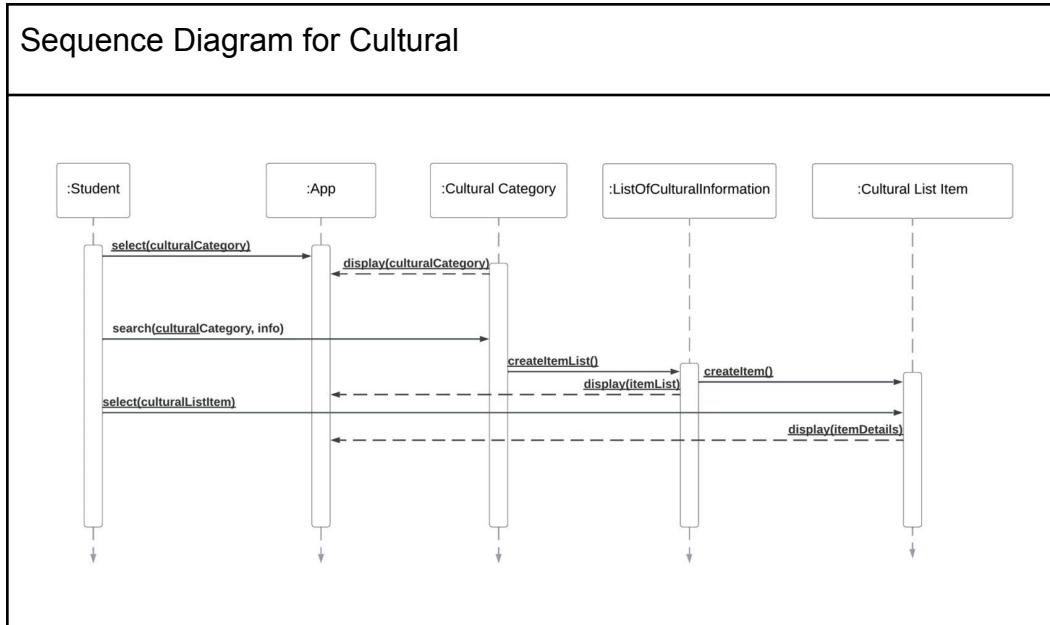
5.3. Social

Sequence Diagram for Social



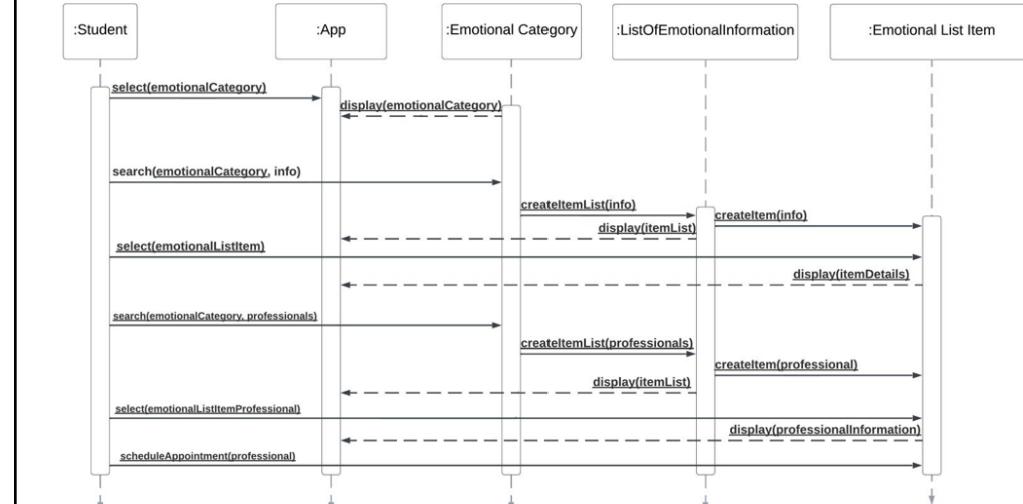
5.4. Cultural

Sequence Diagram for Cultural



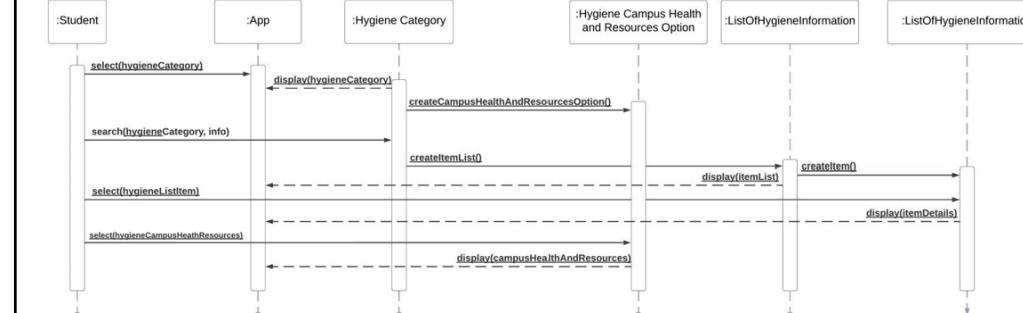
5.5. Emotional

Sequence Diagram for Emotional



5.6. Hygiene

Sequence Diagram for Hygiene



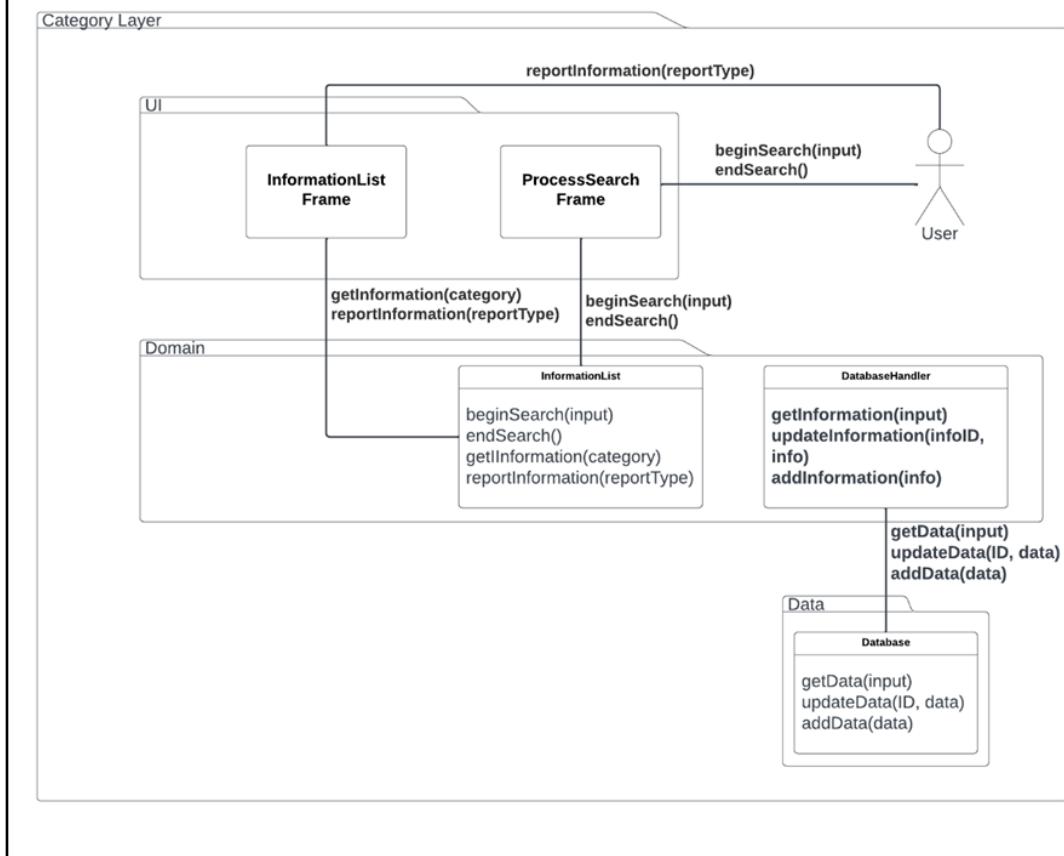
6. Architecture For Each Use Case

6.1. Individual Use Case Architecture

6.1.1. Category Architecture

Category Architecture:

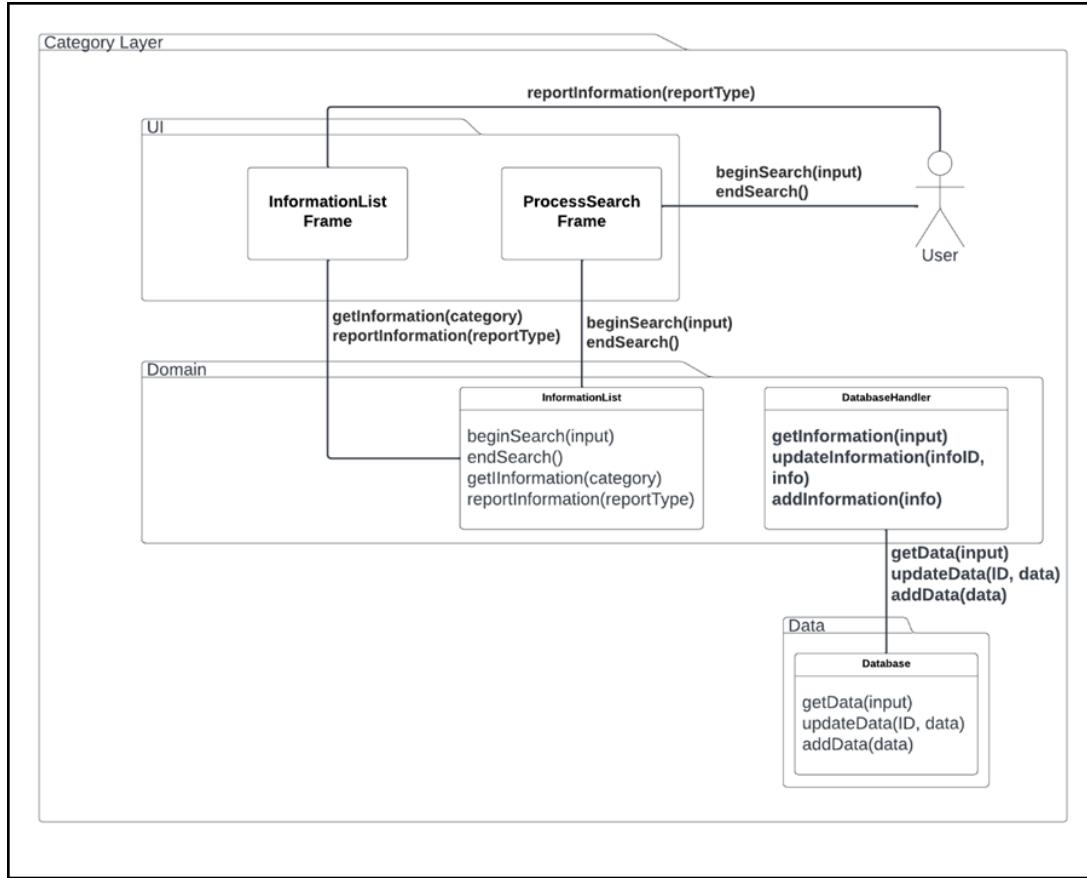
Abstracted out version of what each category uses, which is essentially a list of information and a search function. The category package will be used to model each category's individual use case.



6.1.2. Academic

Academic Use Cases:

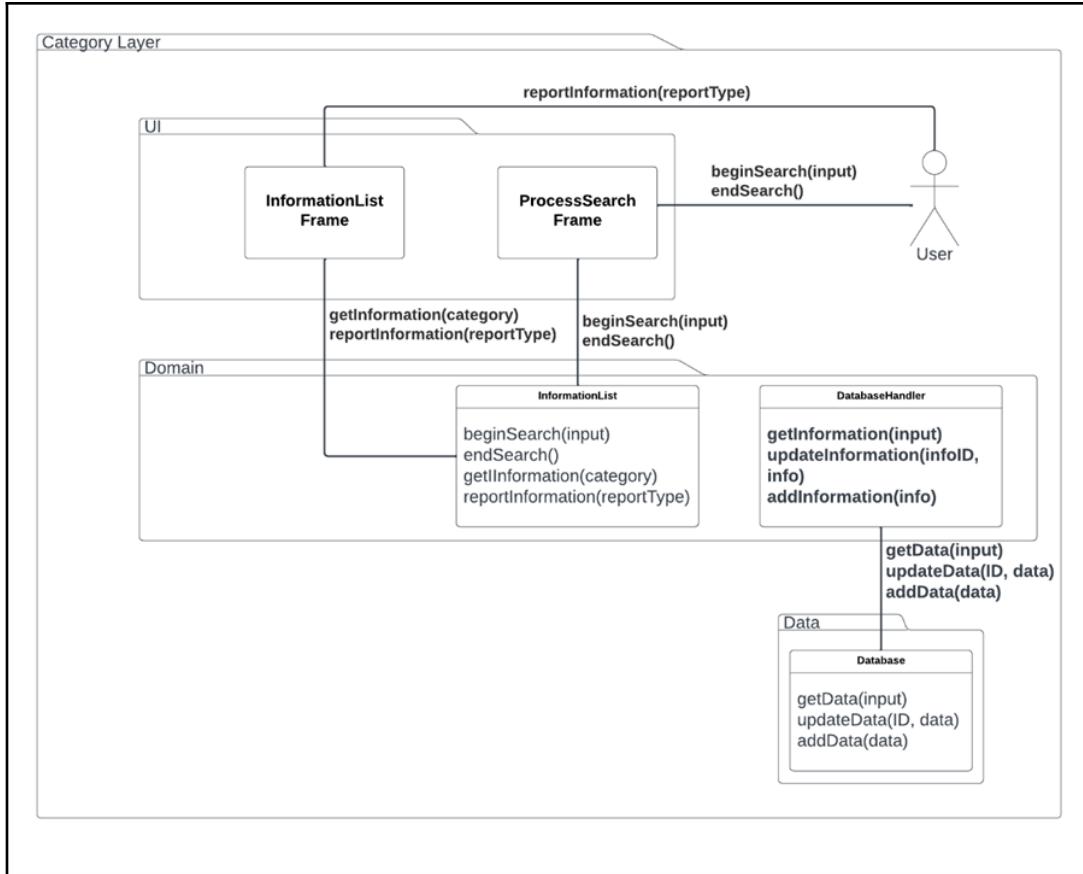
Finding a Tutoring Service and Searching Academic Category



6.1.3. Health

Health:

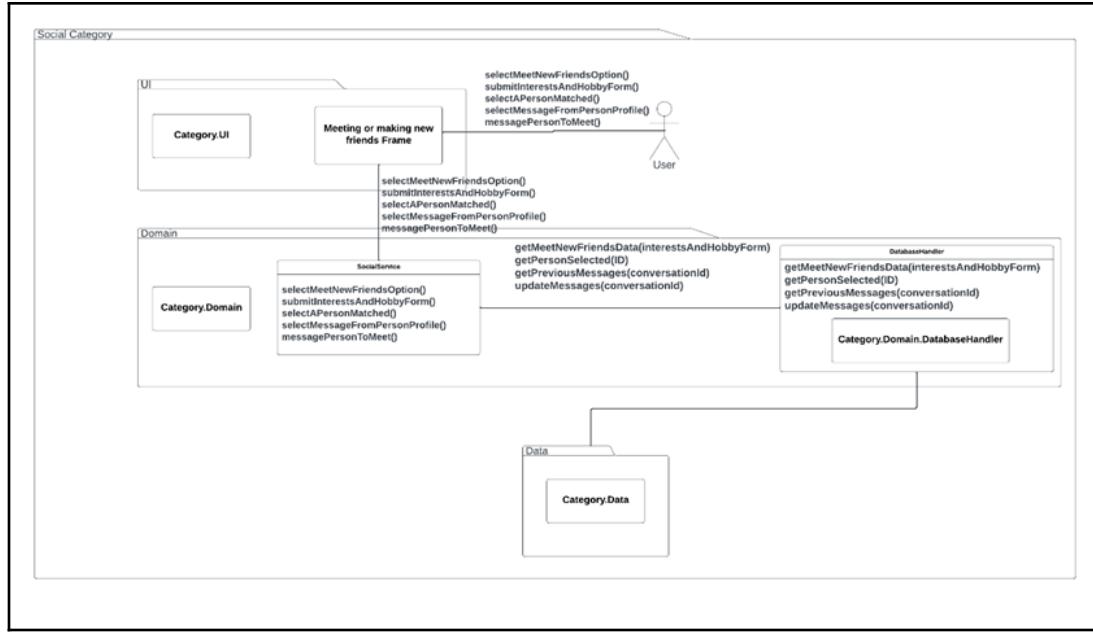
Finding medical clinics and Searching Health Category



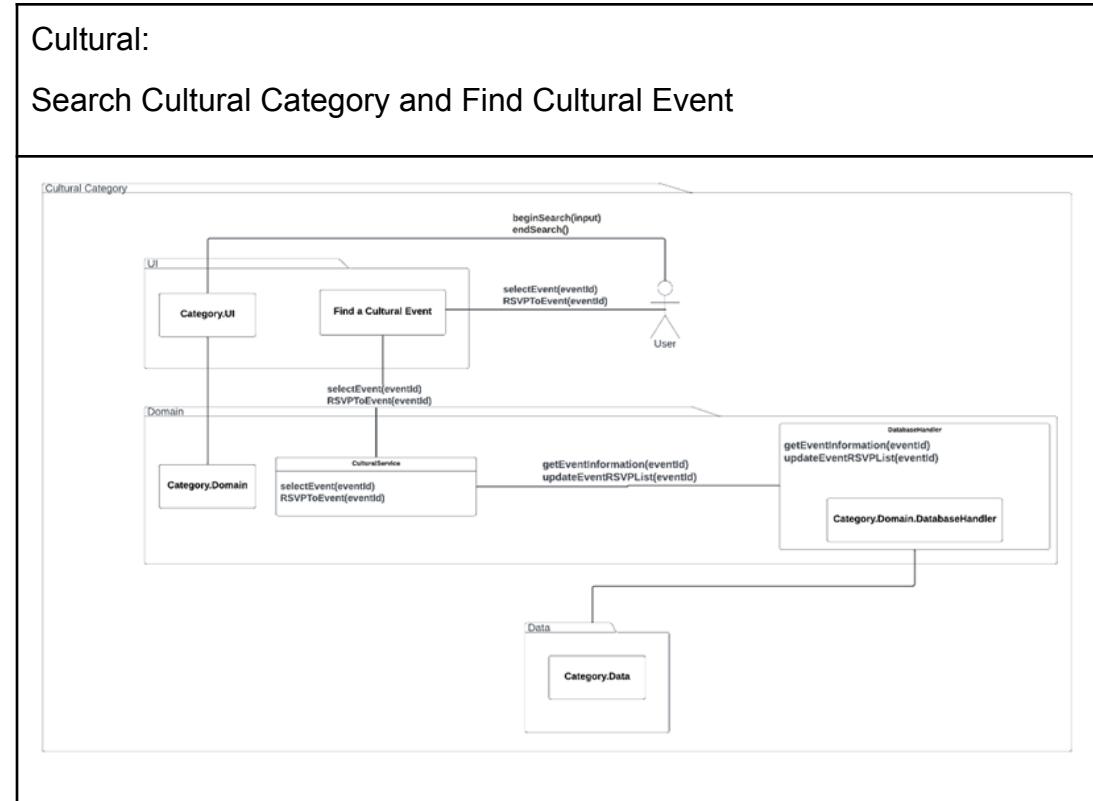
6.1.4. Social

Social:

Meeting or making new friends and Searching Social Category



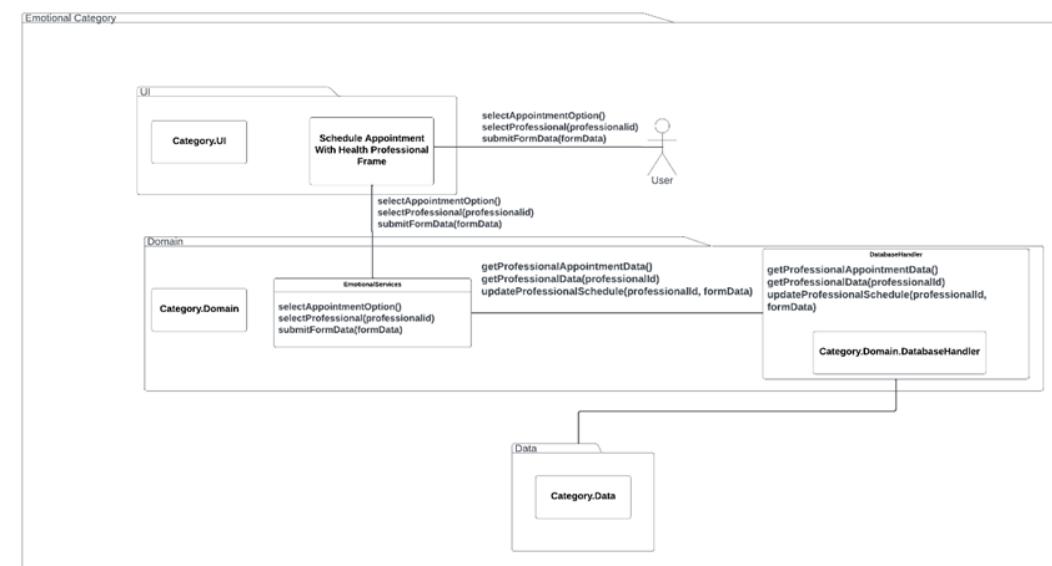
6.1.5. Cultural



6.1.6. Emotional

Emotional:

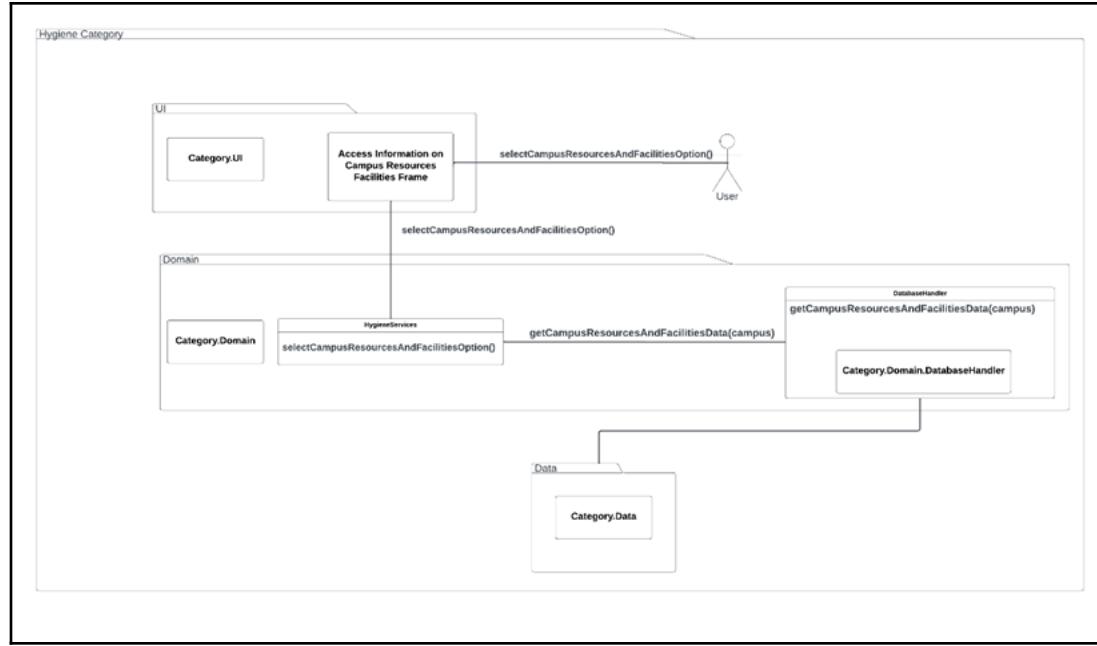
Schedule an Appointment with Campus Health Professionals and Search Emotional Category



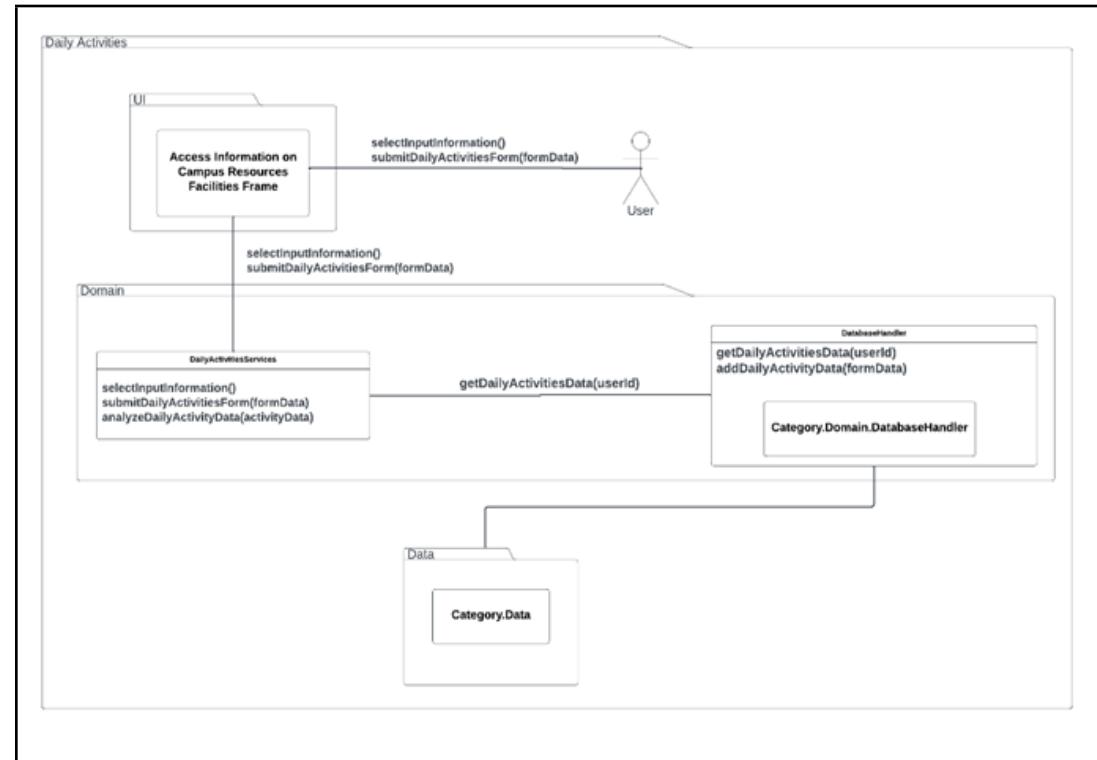
6.1.7. Hygiene

Hygiene:

Access Information on Campus Resources and Search Hygiene Category.

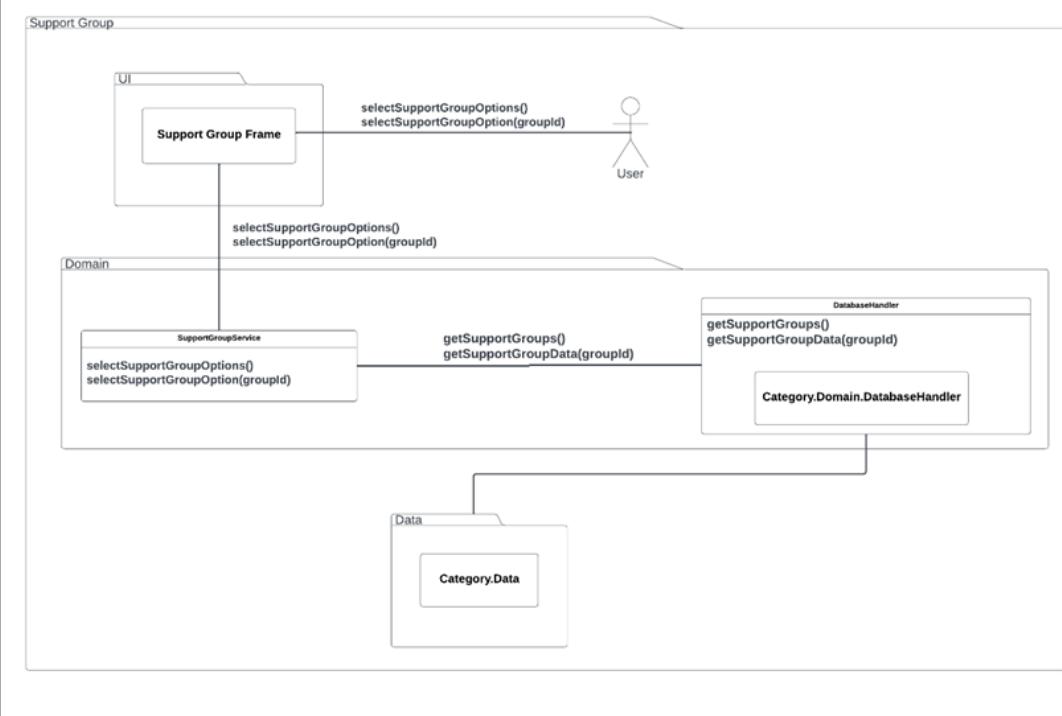


6.1.8. Tracking Daily Activities



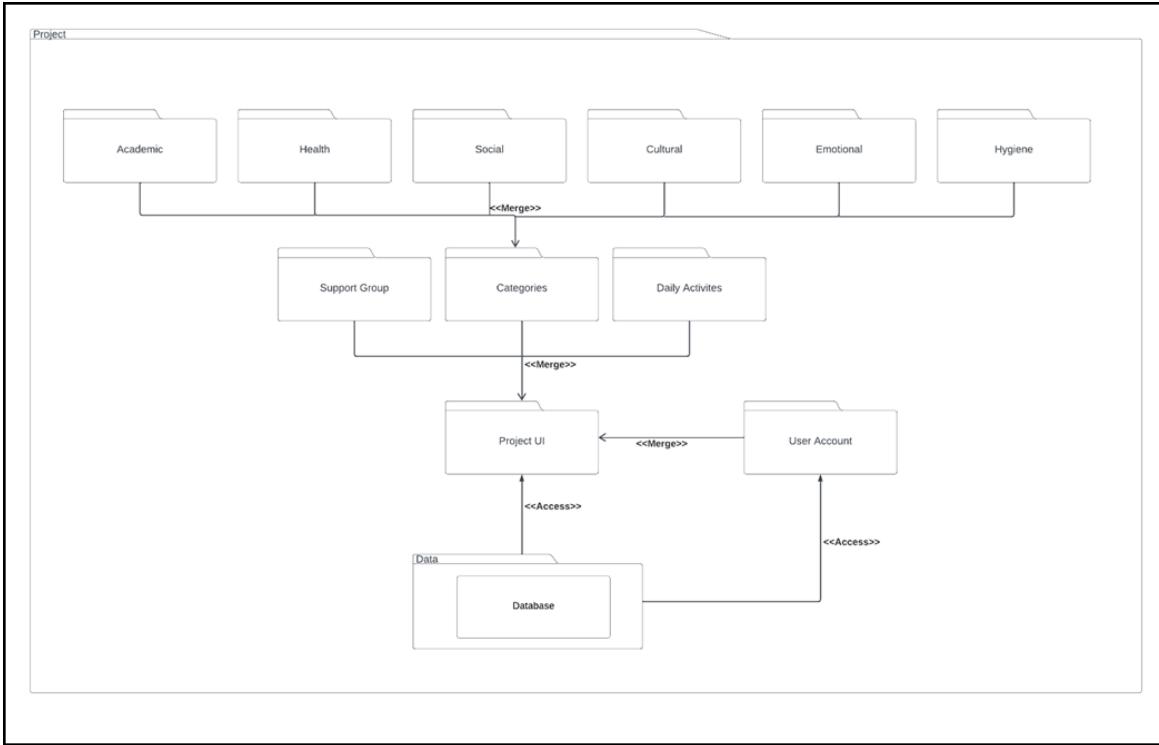
6.1.9. Support Group

Support Group: Search Support Group



6.2. Complete Architecture

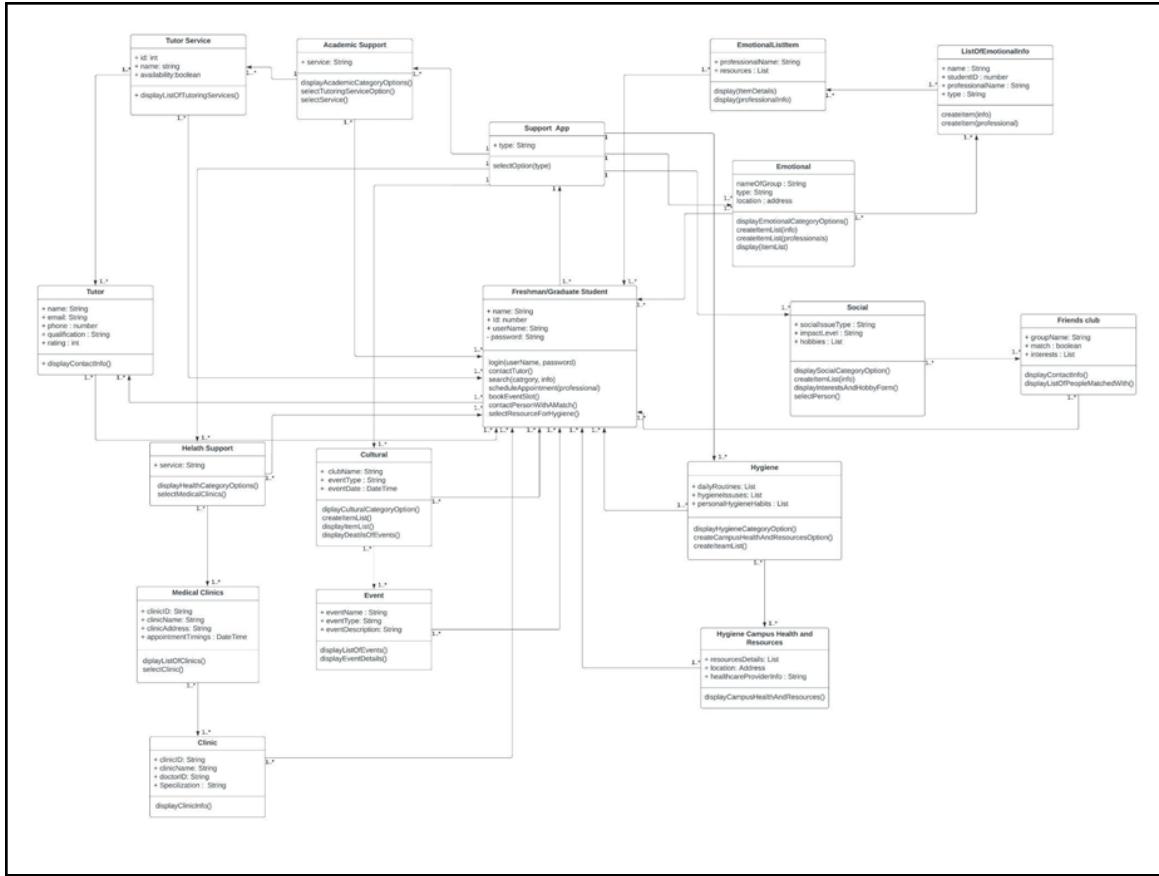
Architecture of Whole App



7. Class Diagrams

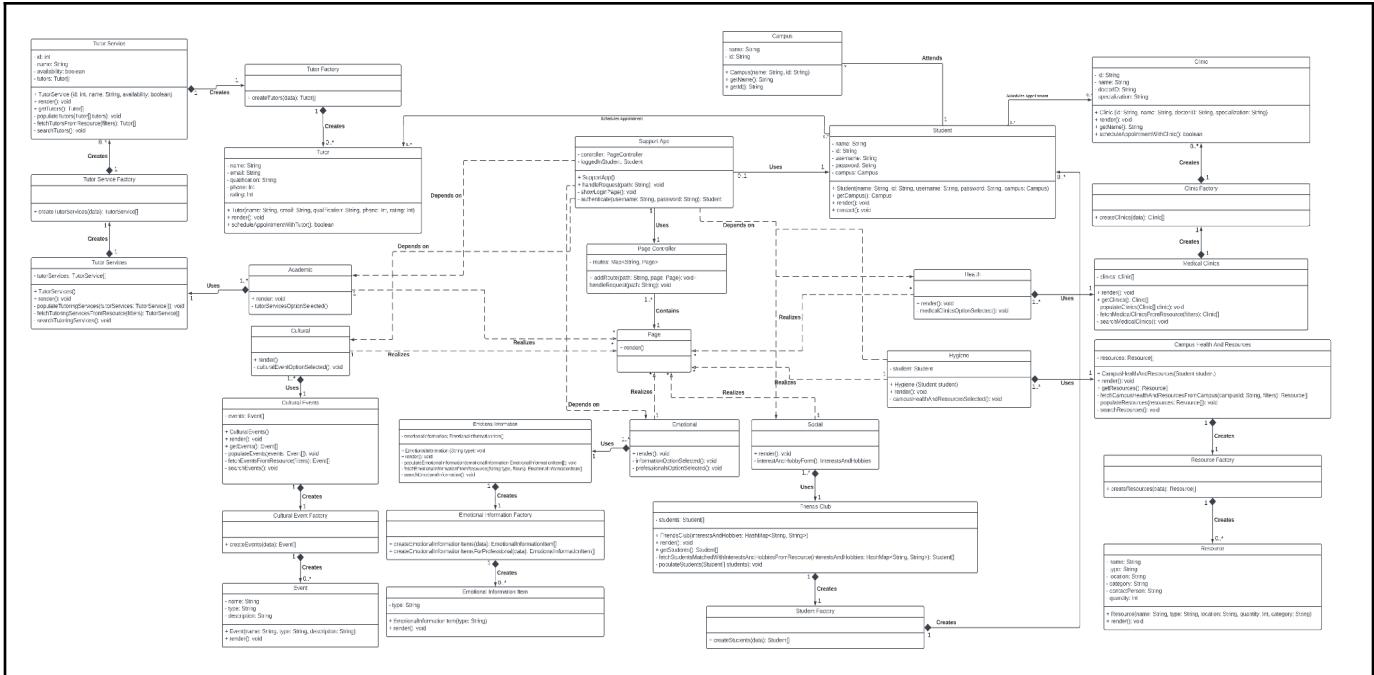
7.1. Initial Full Diagram

DCD of the whole application before code.



7.2. Refactored Full Class Diagram Based on Code

DCD of the whole application after code.

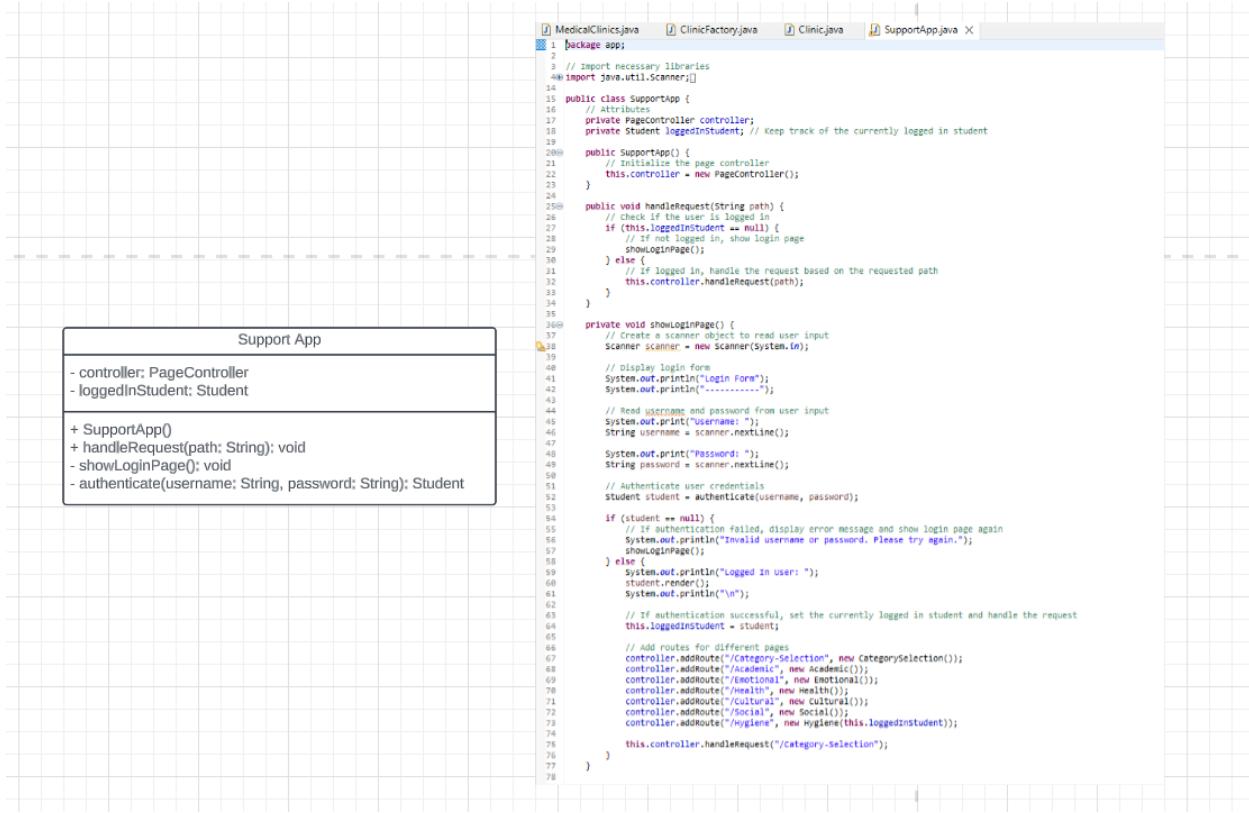


8. Mapping To Code

8.1. Major Classes

Includes: Support App, Student, PageController (Not including small code in report since you have the zip file, it would make this document huge).

8.1.1. Support App

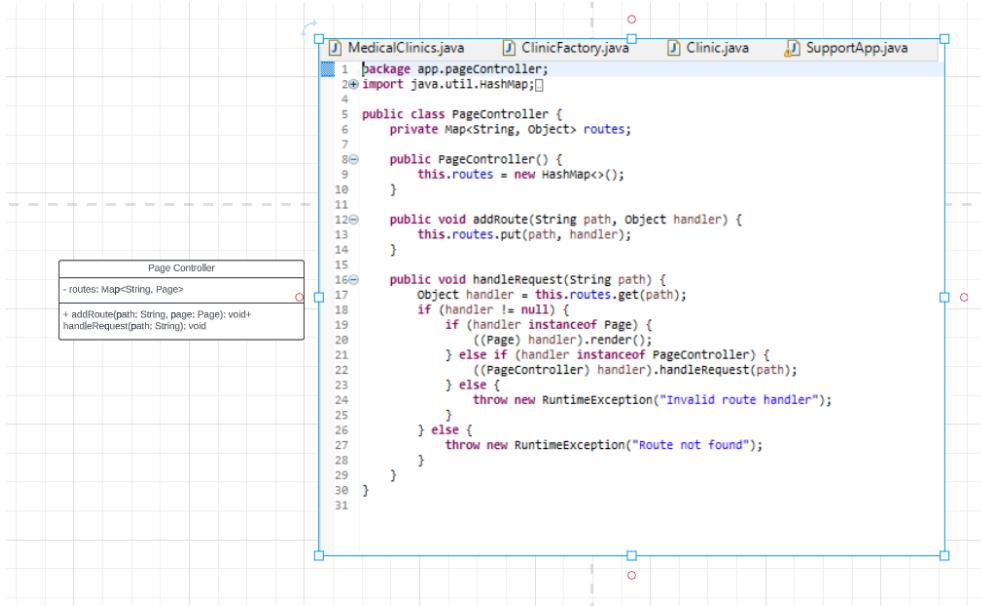


```

1 [package app;
2
3 // Import necessary libraries
4 import java.util.Scanner;
5
6 public class SupportApp {
7     // Attributes
8     private PageController controller;
9     private Student loggedInStudent; // Keep track of the currently logged in student
10
11     public SupportApp() {
12         // Initialize the page controller
13         this.controller = new PageController();
14     }
15
16     public void handleRequest(String path) {
17         // Check if the user is logged in
18         if (this.loggedInStudent == null) {
19             // If not logged in, show login page
20             showLoginPage();
21         } else {
22             // If logged in, handle the request based on the requested path
23             this.controller.handleRequest(path);
24         }
25     }
26
27     private void showLoginPage() {
28         // Create a scanner object to read user input
29         Scanner scanner = new Scanner(System.in);
30
31         // Display login form
32         System.out.println("Login Form");
33         System.out.println("-----");
34
35         // Read username and password from user input
36         System.out.print("Username: ");
37         String username = scanner.nextLine();
38
39         System.out.print("Password: ");
40         String password = scanner.nextLine();
41
42         // Authenticate user credentials
43         Student student = authenticate(username, password);
44
45         if (student == null) {
46             // If authentication failed, display error message and show login page again
47             System.out.println("Invalid username or password. Please try again.");
48             showLoginPage();
49         } else {
50             System.out.println("Logged in user: ");
51             student.print();
52             System.out.println("\n");
53
54             // If authentication successful, set the currently logged in student and handle the request
55             this.loggedInStudent = student;
56
57             // Add routes for different pages
58             controller.addRoute("/Category-Selection", new CategorySelection());
59             controller.addRoute("/Academic", new Academic());
60             controller.addRoute("/Health", new Health());
61             controller.addRoute("/Cultural", new Cultural());
62             controller.addRoute("/Social", new Social());
63             controller.addRoute("/Hygiene", new Hygiene(this.loggedInStudent));
64
65             this.controller.handleRequest("/Category-Selection");
66         }
67     }
68
69     public void handleLoginRequest(String path) {
70         // Implement logic for handling login requests
71     }
72
73     private Student authenticate(String username, String password) {
74         // Implement logic for authenticating users
75     }
76
77 }
78

```

8.1.2. Page Controller



```

1 [package app.pageController;
2
3 import java.util.HashMap;
4
5 public class PageController {
6     private Map<String, Object> routes;
7
8     public PageController() {
9         this.routes = new HashMap<>();
10    }
11
12    public void addRoute(String path, Object handler) {
13        this.routes.put(path, handler);
14    }
15
16    public void handleRequest(String path) {
17        Object handler = this.routes.get(path);
18        if (handler != null) {
19            if (handler instanceof Page) {
20                ((Page) handler).render();
21            } else if (handler instanceof PageController) {
22                ((PageController) handler).handleRequest(path);
23            } else {
24                throw new RuntimeException("Invalid route handler");
25            }
26        } else {
27            throw new RuntimeException("Route not found");
28        }
29    }
30 }
31

```

8.1.3. Student

```

classDiagram
    class Student {
        -name: String
        -id: String
        -username: String
        -password: String
        -campus: Campus

        +Student(name: String, id: String, username: String, password: String, campus: Campus)
        +getCampus(): Campus
        +render(): void
        +contact(): void
    }

```

```

public class Student {
    // Attributes
    private String name;
    private String id;
    private String username;
    private String password;
    private Campus campus;

    // Constructor
    public Student(String name, String id, String userName, String password, Campus campus) {
        this.name = name;
        this.id = id;
        this.username = username;
        this.password = password;
        this.campus = campus;
    }

    // Campus
    public Campus getcampus() {
        System.out.println("\n\t\t\tgetting campus from student...");
        return this.campus;
    }

    // Display
    public void render() {
        System.out.println("\n\t\t\tDisplaying: student");
        System.out.println("\t\t\t\tID " + this.id);
        System.out.println("\t\t\t\tusername " + this.username);
        if (this.campus != null) {
            System.out.println("\t\t\t\tCampus " + this.campus.getName());
        }
    }

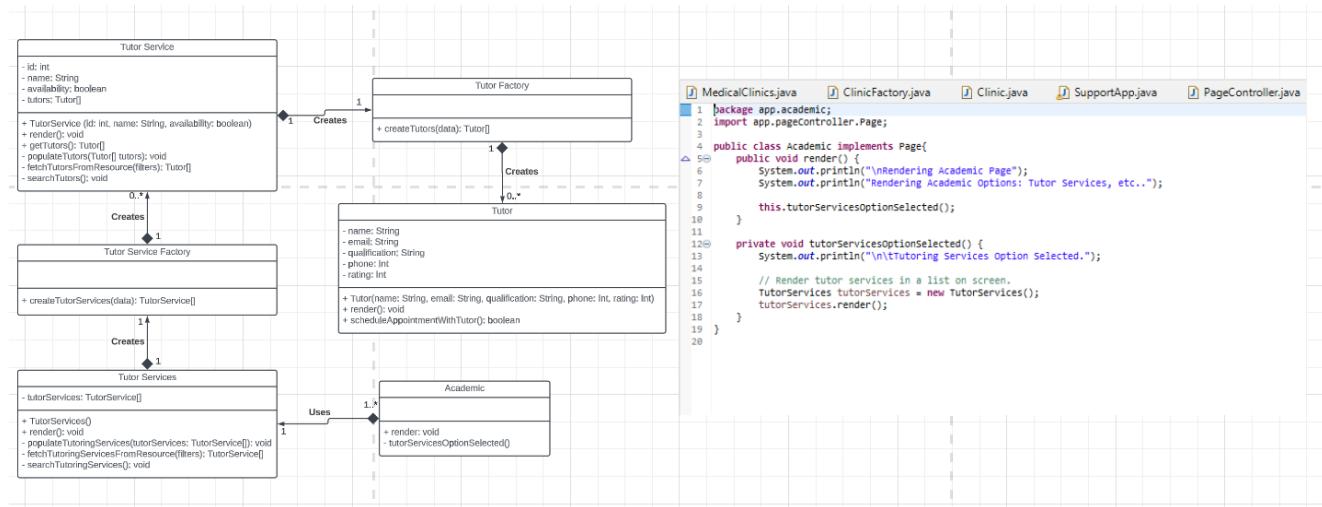
    public void contact() {
        System.out.println("\n\t\t\t" + name + " Contacted.");
    }
}

```

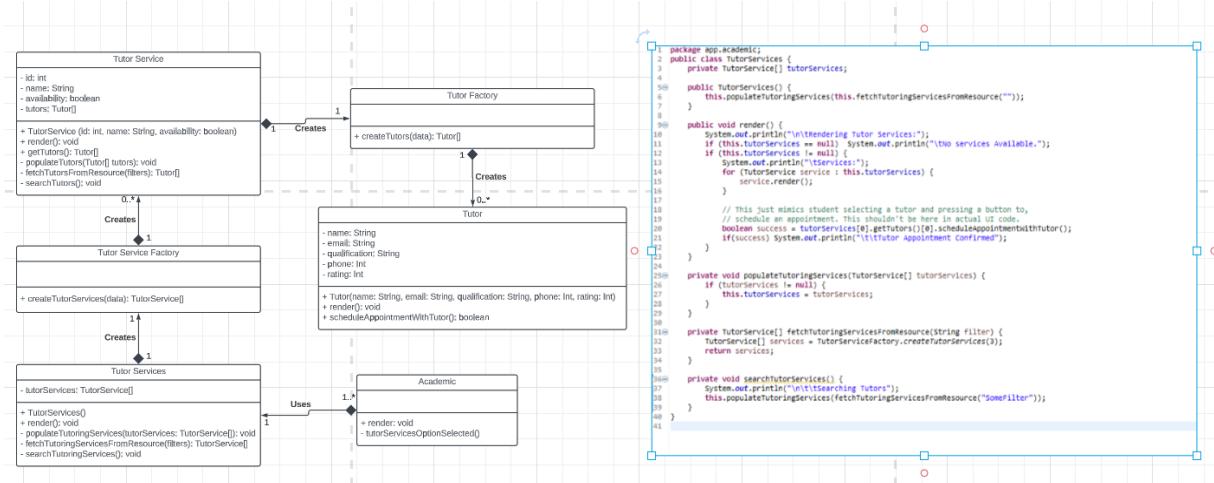
8.2. Use Cases

Includes: The categories and the important classes connected to the categories. Does not include Factory Classes since those are self explanatory.

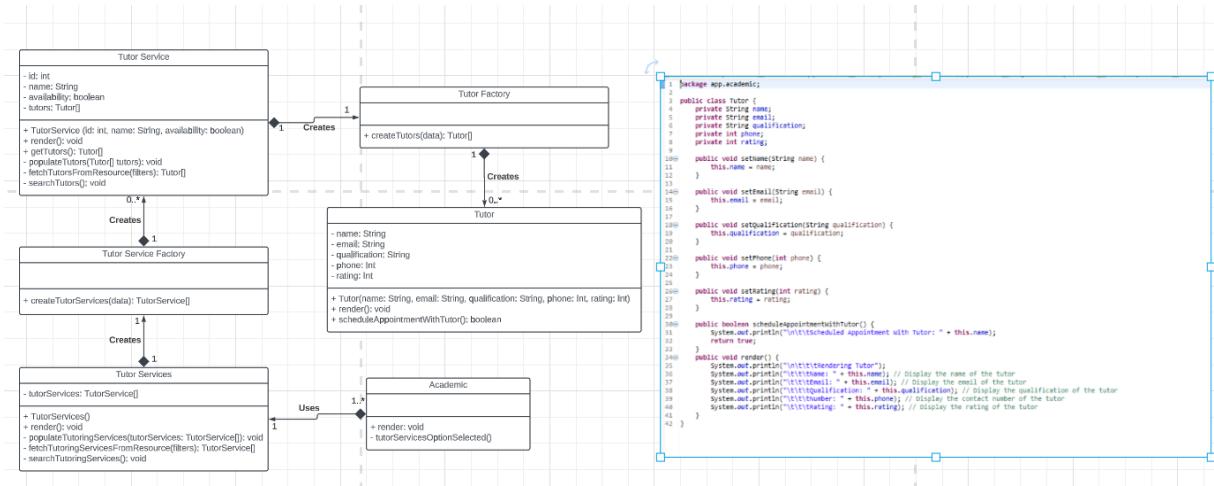
8.2.1. Academic



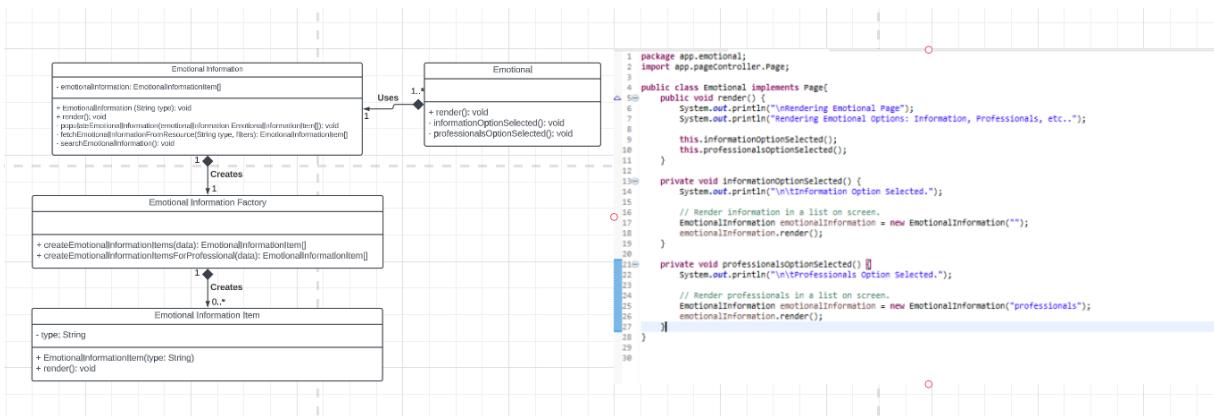
8.2.1.1. TutorServices



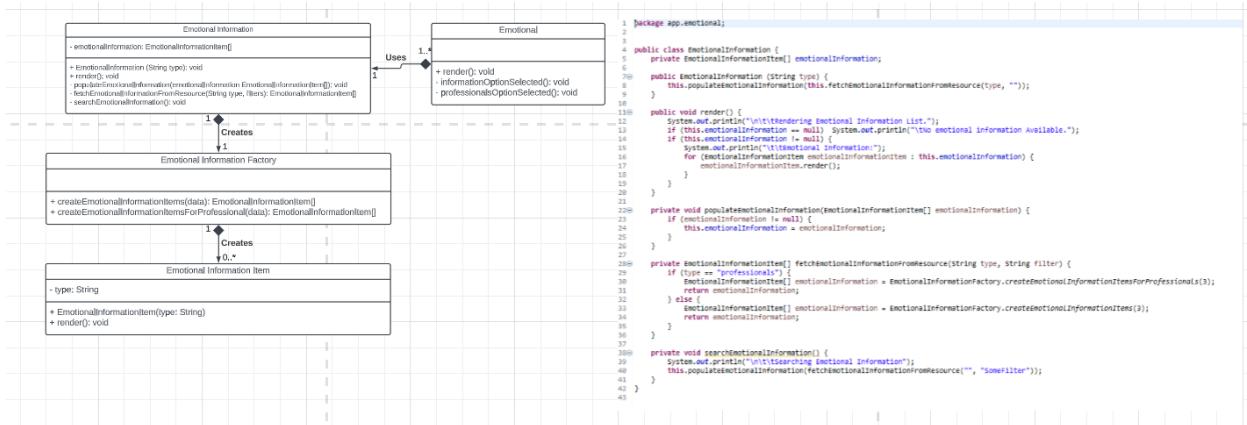
8.2.1.2. Tutor



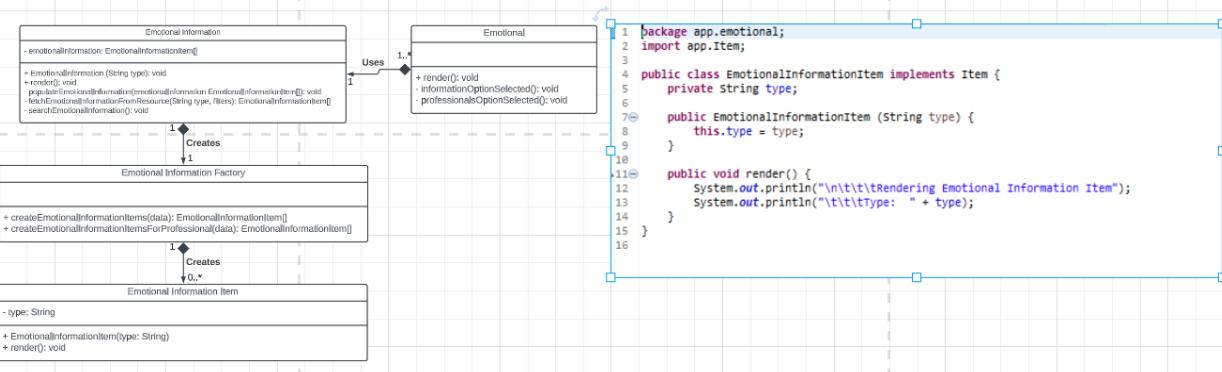
8.2.2. Emotional



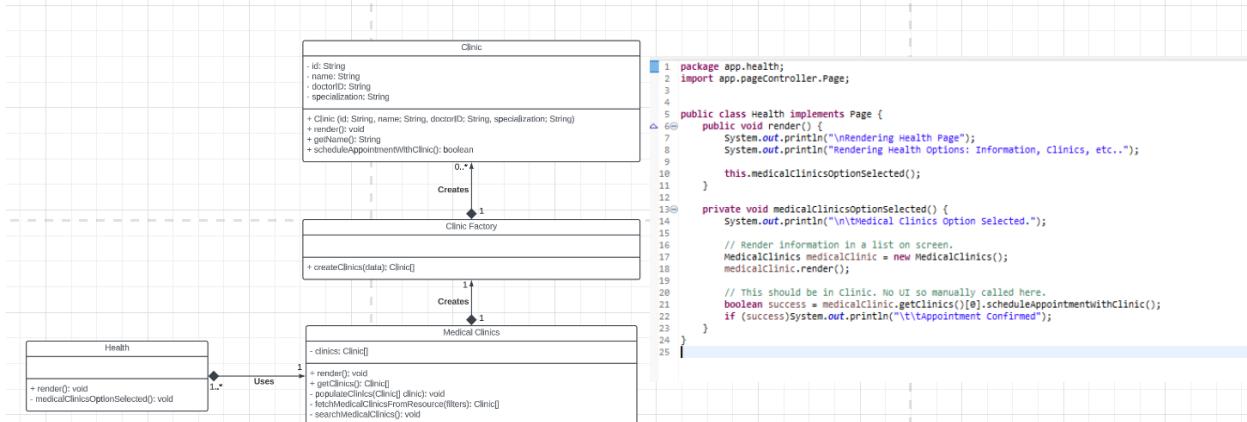
8.2.2.1. Emotional Information



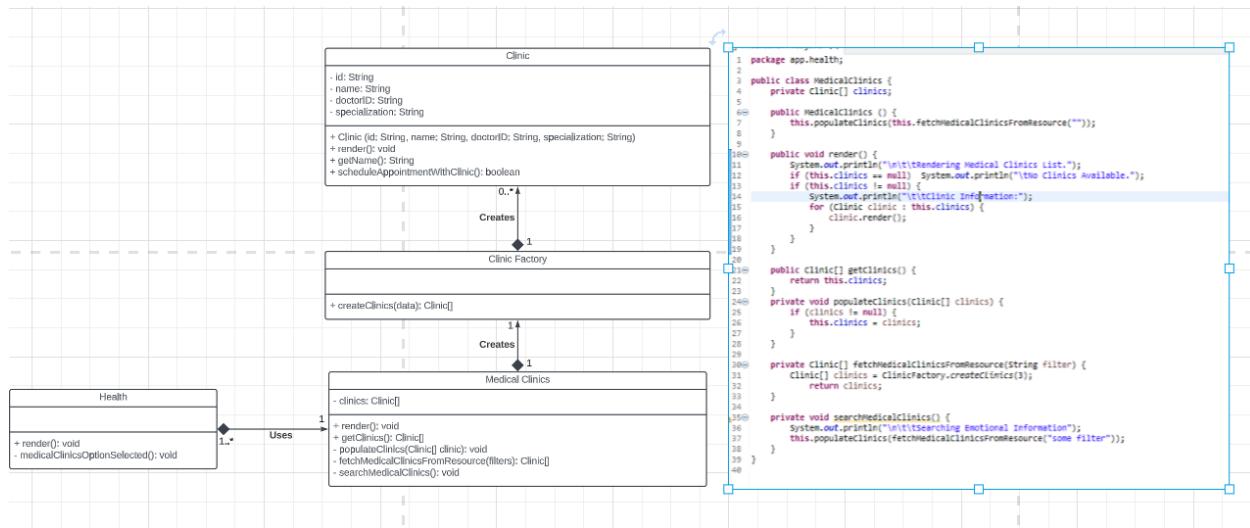
8.2.2.2. Emotional Information Item



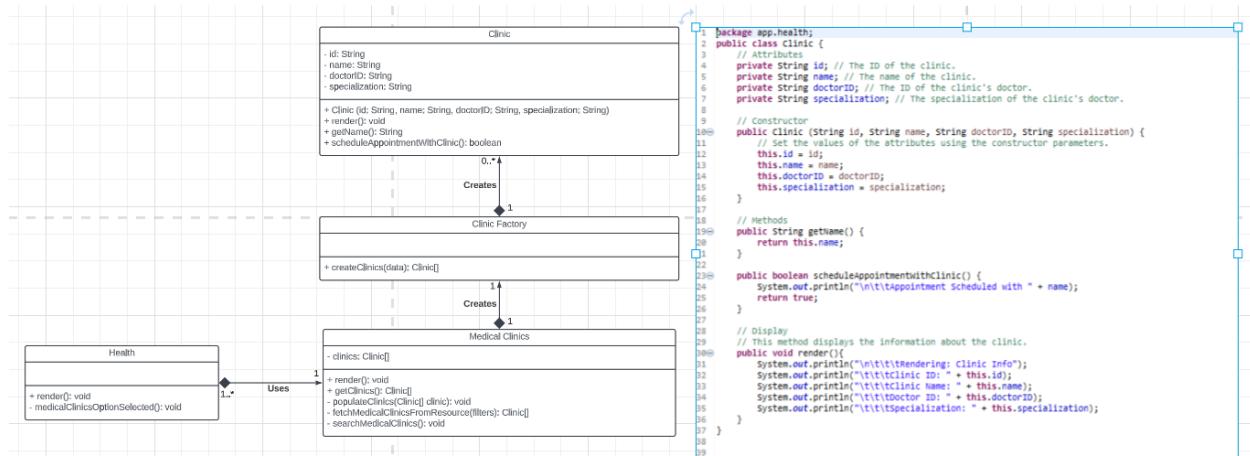
8.2.3. Health



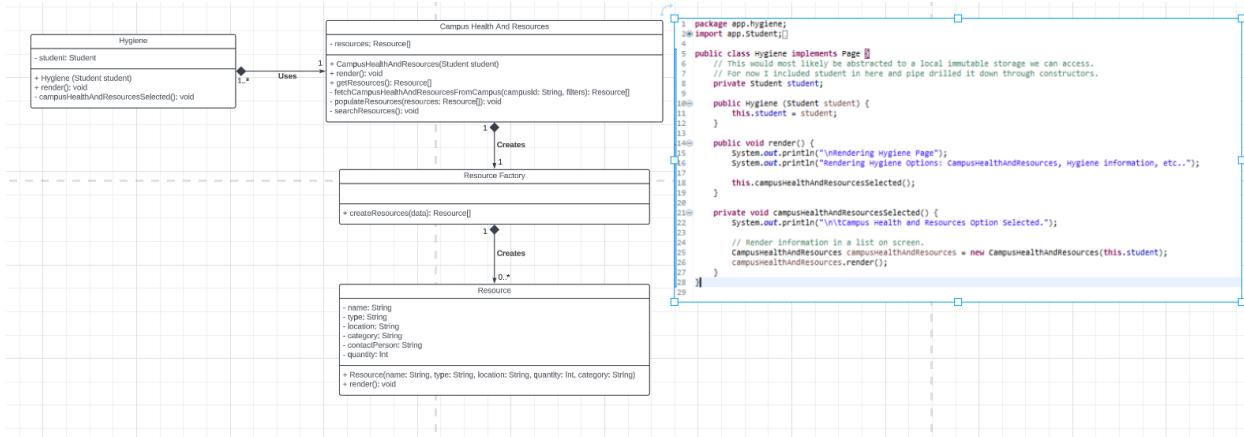
8.2.3.1. Medical Clinics



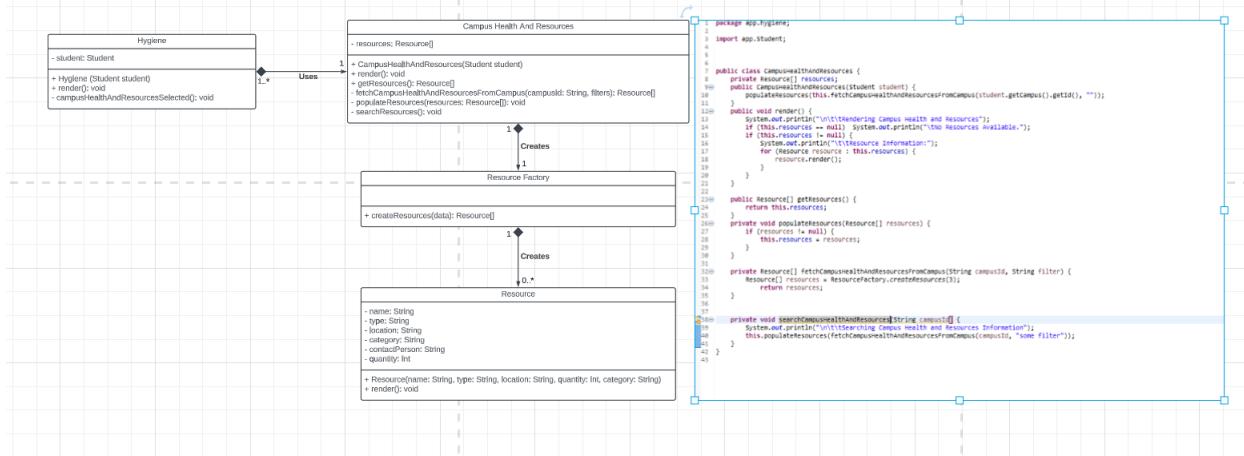
8.2.3.2. Clinic



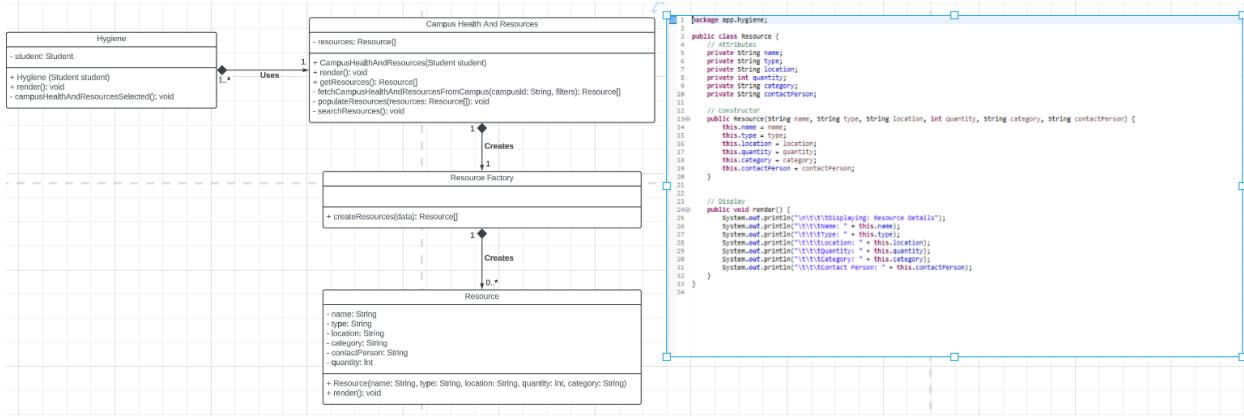
8.2.4. Hygiene



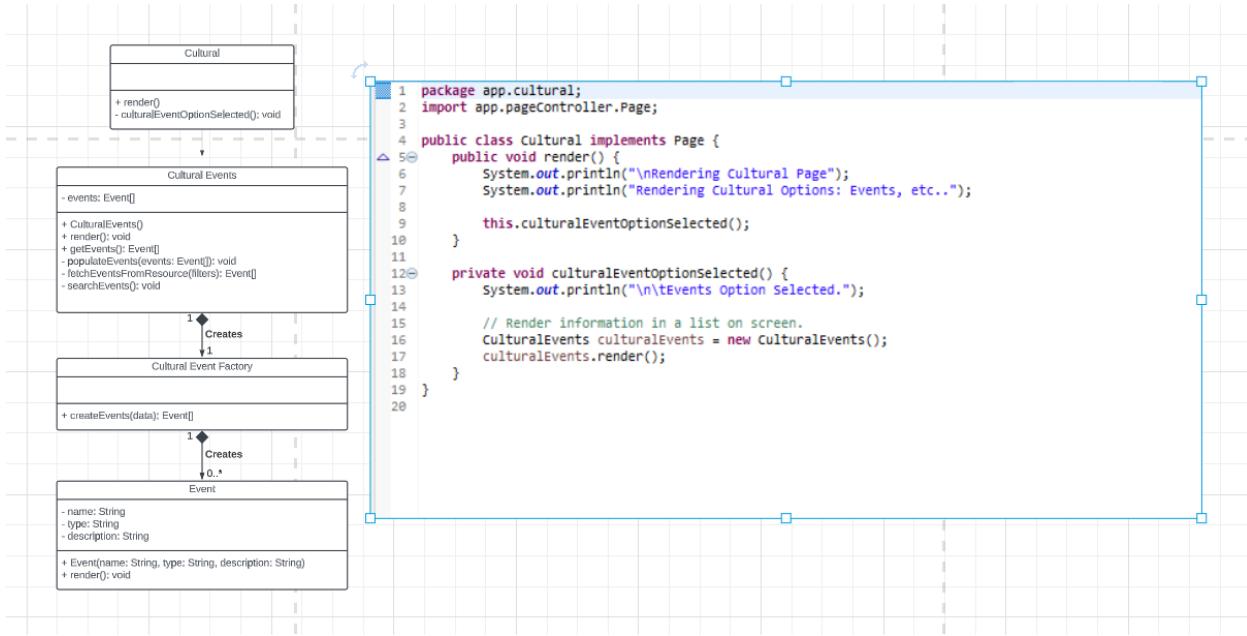
8.2.4.1. Campus Health And Resources



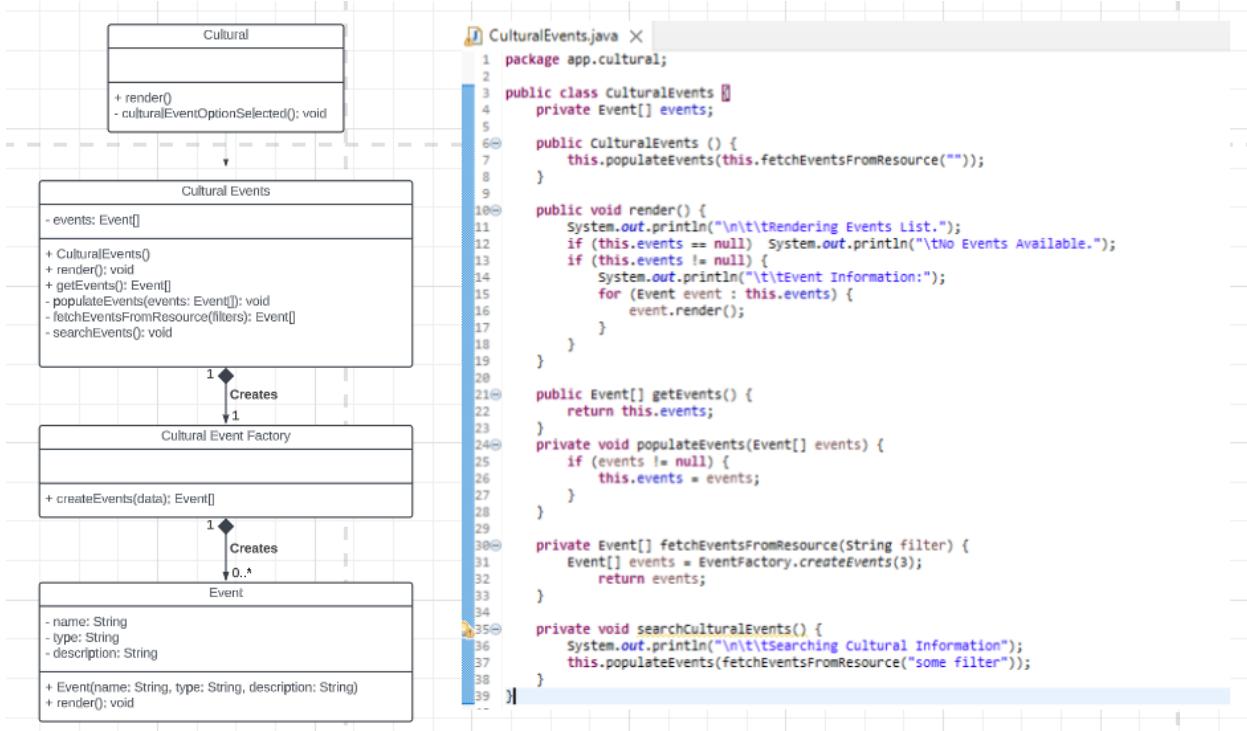
8.2.4.2. Resource



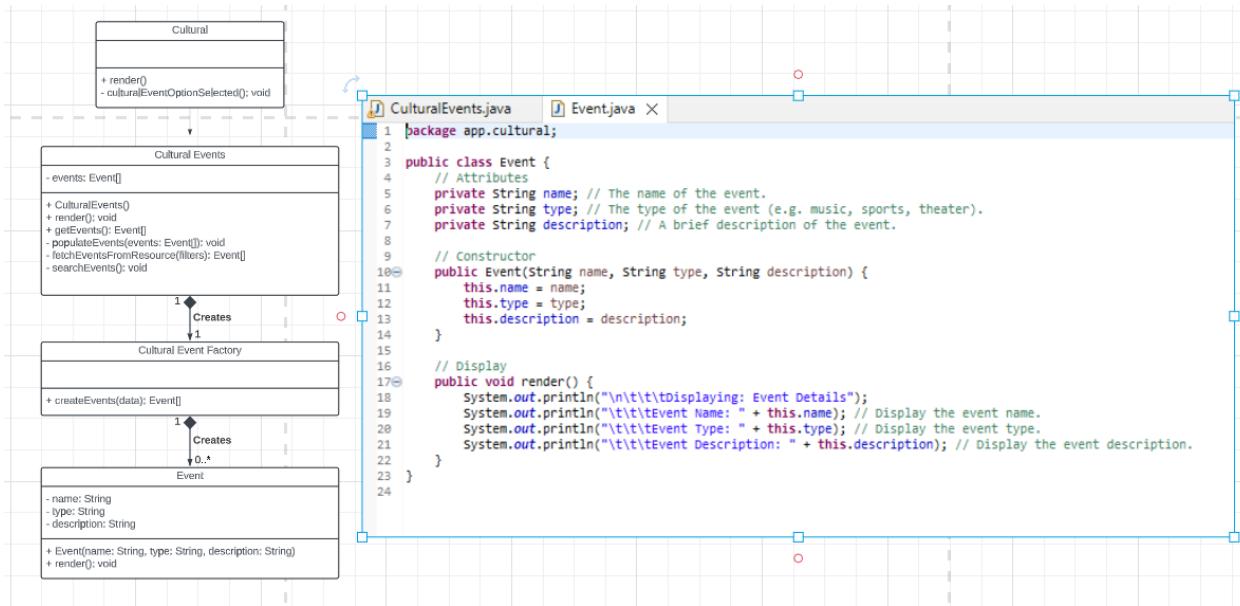
8.2.5. Cultural



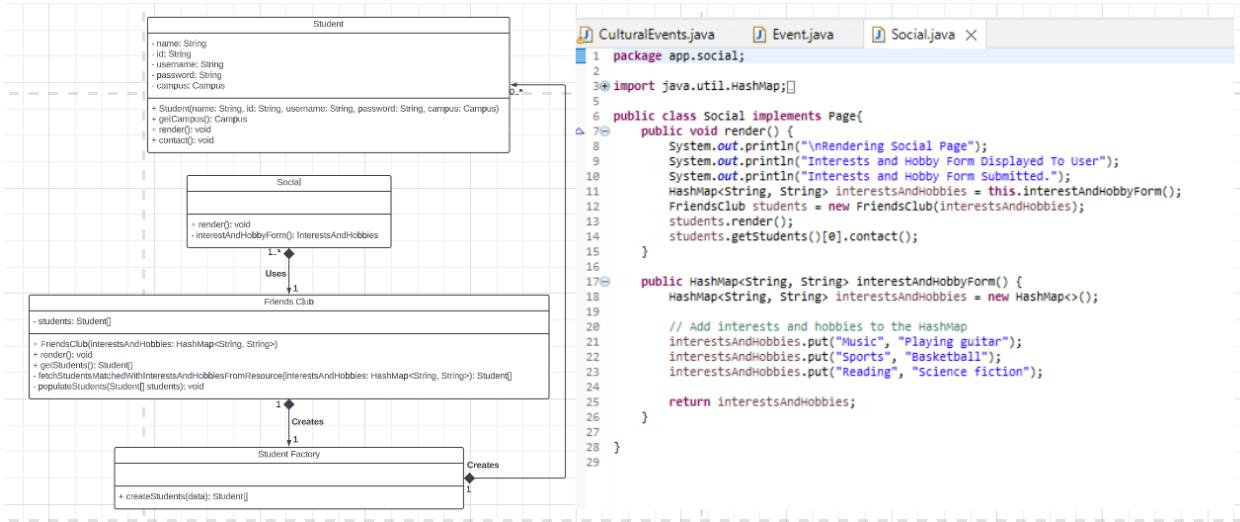
8.2.5.1. Cultural Events



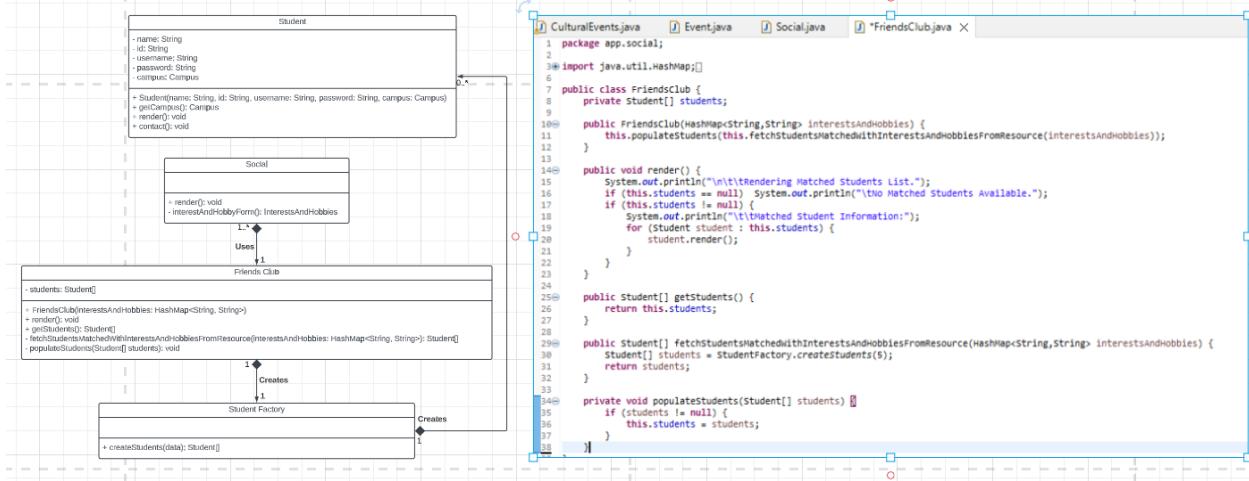
8.2.5.2. Event



8.2.6. Social



8.2.6.1. FriendsClub



9. Conclusion

In conclusion, developing a software app to assist incoming first-year and graduate students at a university is a step towards providing services and support to students at universities all around the world. There were many challenges in the software development process. We learned to work as a team, and the value of detailed design in the early stages; the design doesn't always map to the code as accurately as wanted, and reviewing and refactoring design based on changing requirements and views is essential in great software development team/process, the importance of using design patterns in developing software and how that can change complex code to simple and reusable code/design. Finally, through these challenges, we completed the software development design and code cycle for creating an application by creating the Use cases, Context Diagrams, Domain models, System sequence diagrams, Architectures, Sequence diagrams, Design class diagrams, and code.

CONTRIBUTIONS	
NAME	PERCENTAGE
Venkata Satya Jaya Prakash Banka	33.33%
Smit Jesalbhai Patel	33.33%
Seth Heinzman	33.34%