THE VIKING HUB

JORJA HOLLAND, MICHAEL OGUNNAIKE, GRAHAM HEENEY

1 CONTENTS

2 PRODUCT VISION

2.1 FOR (TARGET CUSTOMER) MICHAEL

This app is made specifically for the staff and students of SETU Waterford.

2.2 WHO (STATEMENT OF THE NEED OR OPPORTUNITY) GRAHAM

As students, we find ourselves opening the WIT website and searching for different services and then going onto the bus apps and looking at times and having to keep so many tabs open for my own convenience. Having an app that will keep it all together will make students' lives so much easier. The convenience on this app makes life as a student much better. As the library gets increasingly busier during the semester, our app allows students to book IT rooms and grinds in them as well. As many students may find themselves struggling and having nowhere on campus to study, our app is the solution to their problems.

2.3 THE (PRODUCT NAME) IS A (PRODUCT CATEGORY) JORJA

The Viking Hub is a mobile application that provides services for students and college affiliates such as bus information, book study rooms, the ability to host educational classes and rewards.

2.4 THAT (KEY BENEFIT, COMPELLING REASON TO BUY) GRAHAM

We believe as a student-made app for students it increases the accessibility of all the college services online on different webpages and bus times for buses that run by the college and into the city. The price for our app is €10 which is for the services we offer in our app including free bus rides and whatnot, we think that price point is fair and a strong reason to buy alone, not including all the features we have in our app. Our app offers complimentary bus rides and is priced right which we believe may be a big reason to buy.

We have also created a booking system for the IT rooms which is not available on the WIT website. As these rooms are not used after 6 pm and yet are kept open, students should be able to book a room for an hour or two in the evening. This would also allow for the group area in the library to free up and allow more people to use PCs. There are only 160 spaces available in the group study area and there are much more than 160 people in groups looking for a place where they can study and speak. Opening the IT rooms to be used as study rooms will allow for more groups to study on the campus.

These benefits are worth the money for the app as these are things the college isn't offering itself. It will allow more convenience for students in their day-to-day lives, and we think that is a strong selling point.

2.5 UNLIKE (PRIMARY COMPETITIVE ALTERNATIVE) MICHAEL

The Viking Hub is specifically tailored to the SETU Waterford community, distinguishing it from other applications. With the Viking Hub, there is a guarantee that users will enjoy the personalization, efficiency, and depth that this application has like no other.

The Viking Hub is a unique service app focusing solely on campus-specific events, features and resources. The application's advanced algorithm provides personalized recommendations for transportation. Its user-friendly interface is designed to address the unique needs of the SETU Waterford community, enhancing their sense of belonging and engagement which makes it a top choice for comprehensive campus services.

The primary competitive alternative to this application is the college website itself which is our only competitor as it is for the students and the college staff. But we believe and we are without doubt that our users will agree that our app is superior to the college website as we provide a sense of caring through our service by catering to the students' and college affiliates' needs.

The college website provides information on events, timetables for both lecturers and students, library open hours and more in-depth on the library such as searching for books. Comparing this to the Viking Hub our application provides a more immersive experience for the user as we provide services the college website does not such as the ability to host grinds in the college facilities and provide the opportunity to book study rooms. This displays that our application is more passionate about the students by showing the understanding of how college life can get very busy and difficult, and we aim to make it the best experience to enhance learning.

Our product is aimed at students, lecturers, and college employees by providing the ability to access bus timetables, the capacity, and forms of payment which are very important pieces of information for the users who commute to college every day. We aim to cater to all users whether they live on campus or commute we want to satisfy all criteria. We compared the Viking Hub to the Dublin Coach which only allows you to book bus fares and see what times they are due to arrive. This once again amplifies the pride and care we have for our users, unlike simple applications such as Dublin Coach.

2.6 OUR PRODUCT (STATEMENT OF PRIMARY DIFFERENTIATION) JORJA

The main difference between our product and competing products is we provide more information; we are giving students more access to services both digitally and physically with the ability to book study rooms within the college to be able to see what grinds are available and to attend grinds after college.

Our product is not only aimed at students but at lecturers and college employees with access to bus information such as times, how busy the bus is and what forms of payment they provide compared to the Dublin coach app which only allows you to book bus fares. Also, the TFI bus card it only provides 30% off to users who are not in possession of a TFI student card which gives 50% off but only during your time as what they consider a “young adult” which is calculated as 4 years. Comparing this to our application we provide a loyalty card which is a virtual card and is applied when you book a bus fare with the app. The benefit of this card is depending on whether you’re a student or affiliated with the college the fare will differ, but you will be awarded a free bus fare for the user's most frequently used bus.

These specific features of our application are more in favour of the students as it provides the ability to further or gain knowledge on certain topics by providing a platform that allows students to book study rooms in the library online to support group projects or learning and the ability to host grinds for specific topics in the FTG rooms after college hours to help students gain more understanding of certain modules.

To compare our application, we chose the college website itself which contains the library facilities and the student learning lab. The college website in terms of the library facilities provides an adequate amount of information about the library such as opening times, and the ability to look up books and e-journals and shows that seminar rooms are available but does not provide the facility to book one of these rooms, whereas our application is aimed more towards what the students want to know about the library such as open times, how busy the library is which will be monitored by how many of the student cards are scanned upon entry and the ability to book seminar rooms online for a maximum of two hours at a time.

Another facility we provide is the power to host grinds which will be taught by students and for students. This eliminates the fear of a classroom environment it creates a more comfortable environment where students are asking their friends to help them and unifies the students with a common issue. This also allows the students who are teaching the grinds to further their knowledge by helping their classmates with their issues, encouraging them to advance their public speaking skills by teaching the whole class in turn people sharing this common problem may even help each other from attending the grinds.

In conclusion, our application is not only providing services to both students and college affiliates but it is encouraging students to advance their knowledge, share their knowledge, create an environment where it is okay to be not as advanced as others and provide more accessible public transport by providing more information on bus times, the capacity occupied on the buses and the payments provided and helping to save money with guaranteed lowered prices depending on what role you occupy at the college and providing a rewards card.

3 Features, Scenarios and Stories

3.1 Develop a feature list for your product. Document how you derived this list. Jorja

To clearly display the feature list of The Viking Hub we have utilized the use of tables to also compare them to similar applications. The Viking Hub team have specifically picked these applications as they are predominately the most frequently used applications among students and college affiliates.

1. **Study Room Booking System:**

   With this feature we enable students to book study rooms within the library which will create a more accessible learning environment. This will also promote individual and group learning.

2. **Grind Hosting Platform:**

   With this feature we allow lecturers to share their knowledge on topics by hosting grinds to help students who are struggling and promote collective learning. This will also help students to improve their communication skills and make new friends.

3. **Enhanced Bus Information and Fare System:**

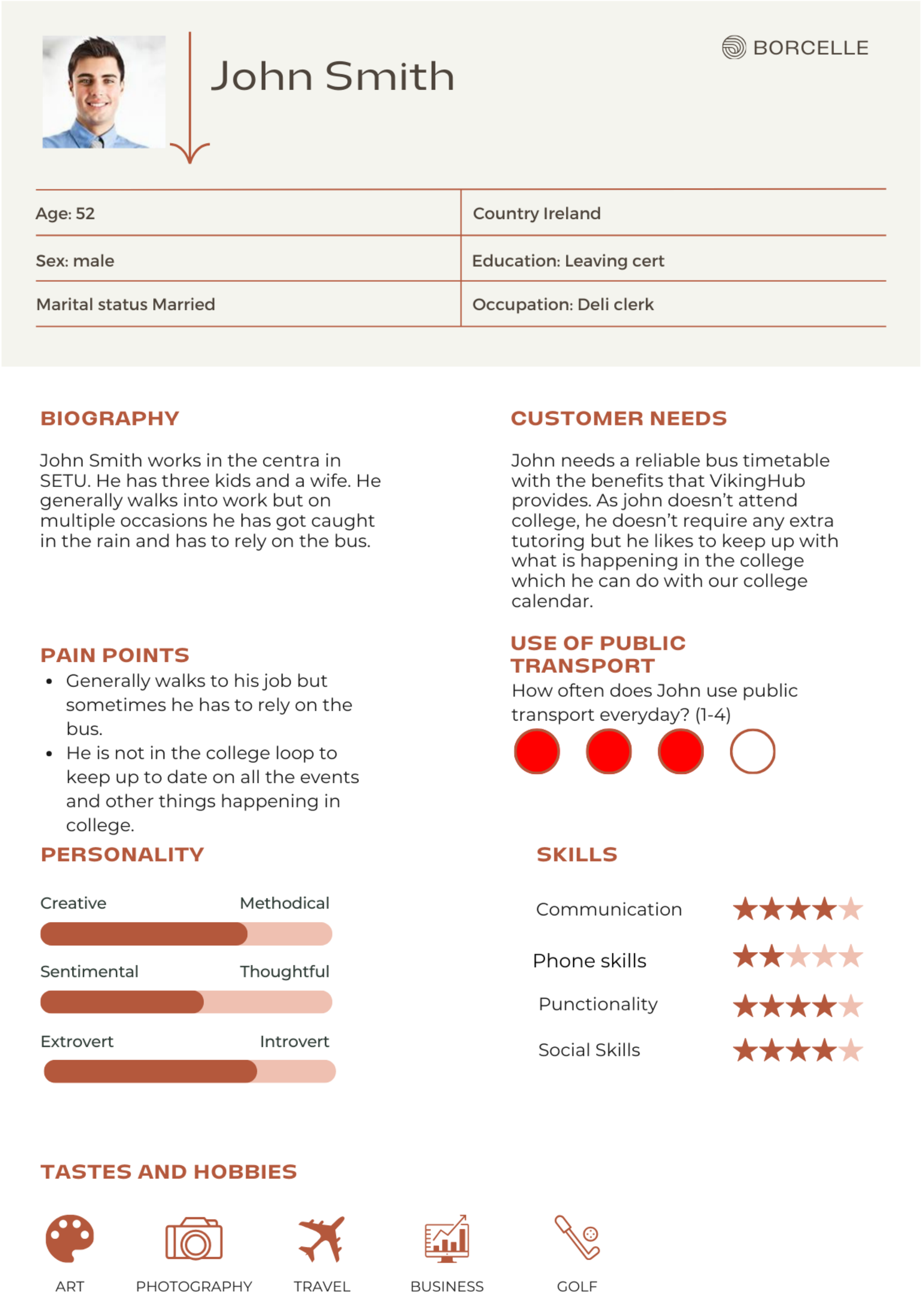
   With this feature this helps both students and college affiliates when commuting to college. This feature provides the times and capacity of the buses, so users have a more accurate representation of the bus services. We also provide a free bus fare for the most used bus of the user.

4. **Interactive Library Services:**

This feature provides the following information which is useful to students when wanting to access the library. Open times, how busy the library is which will be monitored by how many of the student cards are scanned upon entry and the ability to book seminar rooms online for a maximum of two hours at a time.

The Viking Hub team obtained this list of features by analysing existing applications, gathering information, brainstorming. By conducting our research, we were able to create services that optimize our application and showcase the productivity and desire of our product compared to similar applications.

3.2 Create personas. Graham



John Smith

BORCELLE

Age: 52	Country Ireland
Sex: male	Education: Leaving cert
Marital status Married	Occupation: Deli clerk

BIOGRAPHY

John Smith works in the centra in SETU. He has three kids and a wife. He generally walks into work but on multiple occasions he has got caught in the rain and has to rely on the bus.

CUSTOMER NEEDS

John needs a reliable bus timetable with the benefits that VikingHub provides. As john doesn't attend college, he doesn't require any extra tutoring but he likes to keep up with what is happening in the college which he can do with our college calendar.

PAIN POINTS

- Generally walks to his job but sometimes he has to rely on the bus.
- He is not in the college loop to keep up to date on all the events and other things happening in college.

USE OF PUBLIC TRANSPORT

How often does John use public transport everyday? (1-4)

● ● ● ○

PERSONALITY

Creative — Methodical

Sentimental — Thoughtful

Extrovert — Introvert

SKILLS

Communication	★★★★☆
Phone skills	★★☆☆☆
Punctionality	★★★★☆
Social Skills	★★★★☆

TASTES AND HOBBIES

ART PHOTOGRAPHY TRAVEL BUSINESS GOLF

Jane Doe

BORCELLE

Age: 20 years	Country Ireland
Sex: Female	Education: Undergraduate
Marital status Single	Occupation: Kitchen Porter

BIOGRAPHY

Jane Doe is a full time student at SETU who works part-time after college and weekends. Jane is passionate about college and strives to achieve the best grades she can. Jane does not drive and relies on the bus for transport. She often finds herself in the library catching up on work throughout the semester.

PAIN POINTS

- Doesn’t make enough money
- Doesn’t drive
- Struggles with college time to time
- Her job is far from his college

CUSTOMER NEEDS

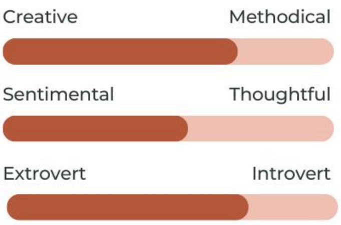
Jane needs a reliable and live bus time table as she cannot to late to work after college. As jane is always in college or working, she sometimes falls behind in his classes and he often needs some extra help . Jane has searched for grinds but they are all to expensive on her current budget.

USE OF PUBLIC TRANSPORT

How often does Jane use public transport everyday? (1-4)

● ● ● ○

PERSONALITY



SKILLS

Internet	★★★★☆
Studying	★★☆☆☆
Punctionality	★★★★☆
Social networks	★★★★☆

TASTES AND HOBBIES

ART

PHOTOGRAPHY

TRAVEL

BUSINESS

GOLF

John Doe

BORCELLE

Age: 35 years	Country Ireland
Sex: male	Education: Postgraduate
Marital status Married	Occupation: Lecturer

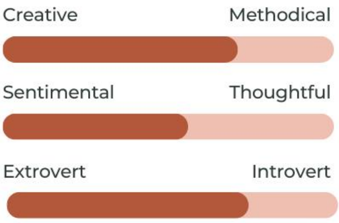
BIOGRAPHY

John Doe is a lecturer is SETU. John teaches computer science. He travels to college by bus as he lives in the town. John offers classes to struggling students after college hours. but sometimes no one shows up as he struggles to get out the information that he is offering classes.

PAIN POINTS

- Waste's his time after college hours when no one shows up to his classes
- He takes the bus and doesn't have a consistent time table on his phone

PERSONALITY



CUSTOMER NEEDS

John uses the timetable on google which is outdated and can be inconsistent so he needs a up to date timetable. John wants to be able to host grinds to struggling students which Vikinghub will allow him to do with ease.

USE OF PUBLIC TRANSPORT

How often does John use public transport everyday? (1-4)

● ● ● ○

SKILLS

Tutoring	★★★★☆
social skills	★★☆☆☆
Punctionality	★★★★☆
Social networks	★★★★☆

TASTES AND HOBBIES

ART PHOTOGRAPHY TRAVEL BUSINESS GOLF

3.3 CREATE SCENARIOS FOR YOUR PRODUCT. (GRAHAM)

John's Daily Routine

John is a student at SETU attempting to juggle his college work, part-time job, and his extracurricular activities. He starts his day by getting the bus to college where he has his classes and study groups throughout the day. As john rushes to catch the bus in the morning, he opens the Viking Hib app to check the timetable for the bus and if there have been any changes to the route as this app provides real time updates to the buses on the app.

After a long day of classes, john needs to find somewhere to study, so rather than going to the library and potentially not being able to get a seat, he books a study room on The Viking Hub app allowing him and his study group to be able to study with no distractions and no delays.

As johns study session wraps up, he remembers he has grinds with a group of students in the FTG building. He seen these grinds were available for free through the Viking Hub app and he was able to join in. this class was hosted by a lecturer for some of his struggling students, but he left the session open so any other students will be able to join for some extra learning.

After the grind session was complete, john check the Viking Hub app to see the times for the busses. On the way home, john thinks about how easy the Viking Hub made his day with everything he needed being in one app with one click.

Adam's Stress-Free Day

Adam, like many others in college, finds himself getting overwhelmed by the requirements of college life. With all the traveling and activities during his day, The Viking Hub has made all this easier to manage.

In the morning Adam opens the app to plan his bus route to college, he knows he can rely on the real time bus routes allowing him to arrive to his classes on time and stress free.

During his breaks, Adam likes to book a study room for him and his peers using the Viking Hub app. This allows for Adam and his friends to have an area that they can study coursework together.

Later in the day, Adam attends a grind session that was organised through the Viking Hub app, he finds it useful, not only for catching up and understanding his college work but also for connecting with his peers.

Adam gets the bus home and reflects on how stress free the Viking Hub made his day.

Janes Productive Study Session

Jane is a lecturer at SETU Waterford, she often hosts study sessions and class reviews with her students. She understands how important it is to her students that she offers any help she has to them. With the Viking Hub, Jane can improve her student's learning experiences with how easy it is to set up and go to a study session.

At the start of every week, Jane reserves rooms in the FTG building for the different study sessions that may be taking place during the week. Using Viking Hub for this makes her life so much easier as it only takes a few seconds to post about it and it is as easy for her students to find it.

During janes extra classes, she encourages her students to make use of the Viking Hub app and to book rooms for themselves to study and help each other with different problems they have. Jane sees how beneficial the app is to her students and how much more beneficial it can be.

After each session ends, Jane asks what time will suit the class best next week and when that is decided, with one button click she can have it organised and ready to go for the following week.

3.4 CREATE USER STORIES. (MICHAEL)

- As a SETU student, I want to be able to have access to real-time bus information and timetables through The Viking Hub so I can plan a stress-free commute to college.
- As a commuter student, I want to be able to receive rewards such as free bus fares through the Viking hub app, encouraging me to use public transportation hence also helping me save money on my daily commute.
- As a user, I want the convenience of accessing all college related resources and services in one app eliminating the need for multiple apps and platforms optimising my daily activities as a student or a staff member.
- As a student I want to be able to book study rooms in advance ensuring a dedicated place for group study or I have a quiet space for study without any last-minute rush or uncertainty.
- As a student I want to be able to host or join grind sessions organised through The Viking Hub aiding me with understanding difficult modules and connecting with my peers through collaborative learning.
- As a staff member I want to be able to encourage collaborative learning and student engagement by promoting the use of The Vicking Hub for booking study rooms and organising educational activities, facilitating a conducive learning environment for students.

3.5 REVISIT FEATURE LIST AND AMEND BASED ON PERSONAS, SCENARIOS, AND USER STORIES. JORJA

From reading the user personas, scenarios, and stories I have identified some changes to be made to our feature list for our application. From reading the user personas and specifically their needs I identified that the consumers of this product lack a student and lecturers' timetable and an college calendar. With these adjustments to our feature list our list of services is as follows:

1. **Study Room Booking System:**

   With this feature we enable students to book study rooms within the library which will create a more accessible learning environment. This will also promote individual and group learning.

2. **Grind Hosting Platform:**

   With this feature we allow students to share their knowledge on topics and allow lecturers to help struggling students by hosting grinds to help others and promote collective learning. This will also help students to improve their communication skills and make new friends.

3. **Enhanced Bus Information and Fare System:**

   With this feature this helps both students and college affiliates when commuting to college. This feature provides the times and capacity of the buses, so users have a more accurate representation of the bus services. We also provide a free bus fare for the most used bus of the user.

4. **Interactive Library Services:**

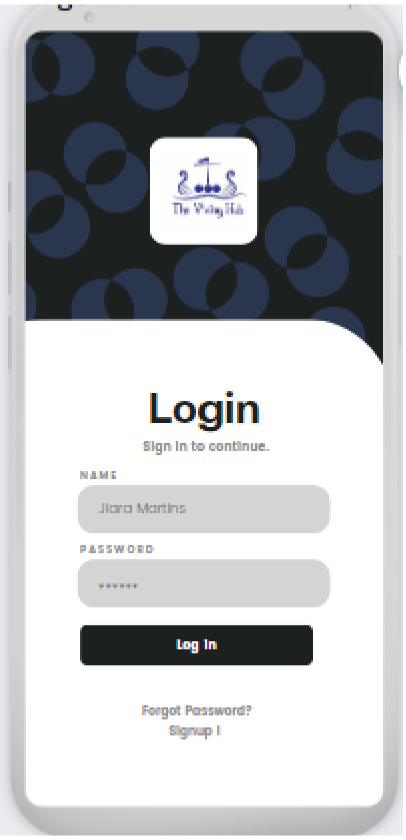
   This feature provides the following information which is useful to students when wanting to access the library. Open times, how busy the library is which will be monitored by how many of the student cards are scanned upon entry and the ability to book seminar rooms online for a maximum of two hours at a time.

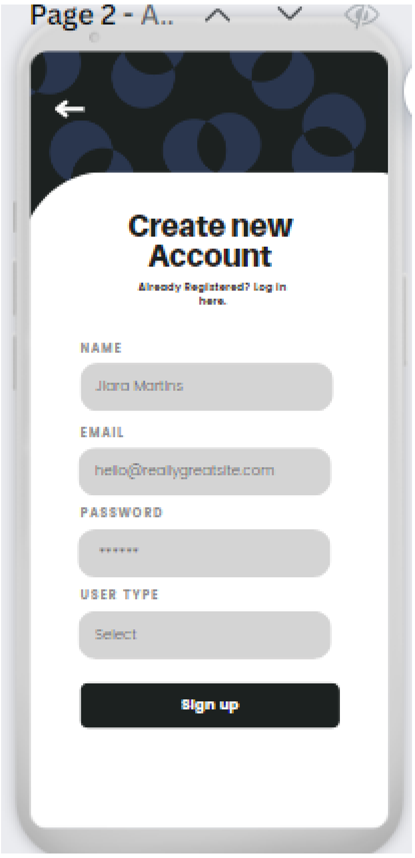
5. **Student and lecturers Timetables:**
   This feature was brought in to make students and lecturers lives easier which a tap of a screen. With this feature we will have a section on the application which allows you to view your timetable and to save time scanning over it to highlight exactly what class you have at that time.

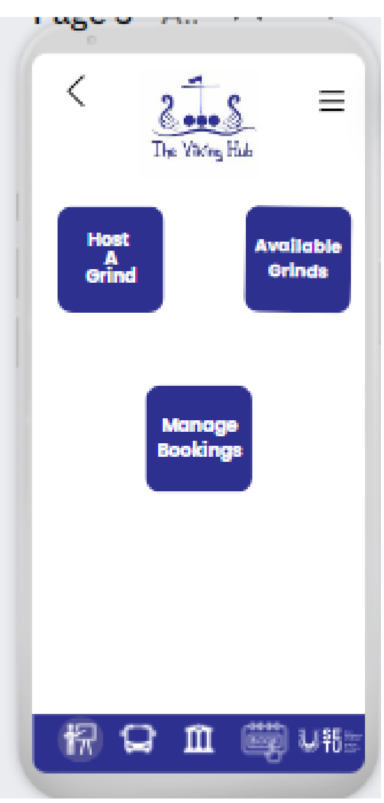
6. **College calendar:**
   This feature was introduced as a result of analysing the users experience of our application. This feature will allow users to see any upcoming events with the college such as women in stem, any holidays etc. this will also help students stay on top of deadlines as they mark in the calendar when assignments are due and any CA exams that occur during the semester.

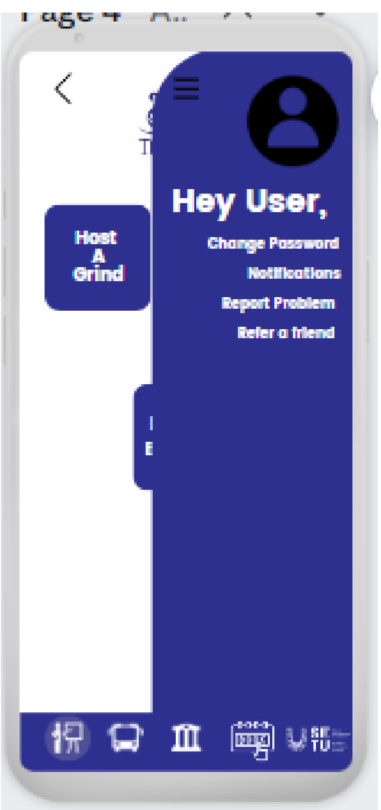
With these changes applied to feature list we believe our application stands out against our competitors and that we can provide the best services to our users which hopefully will make college life less stressful and more enjoyable.

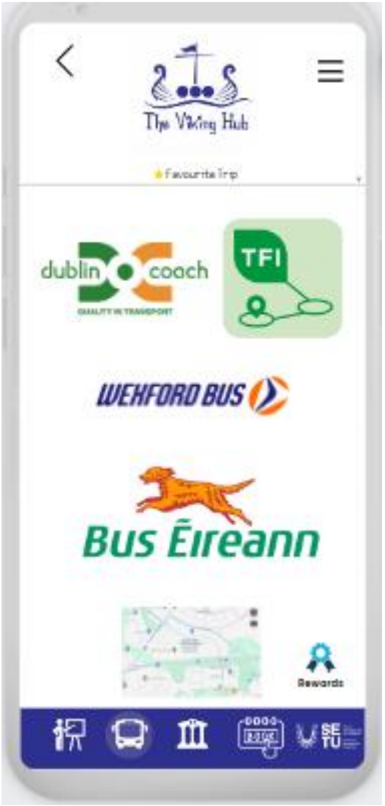
3.6 Create a prototype using one of the tools discussed in class.(Michael)

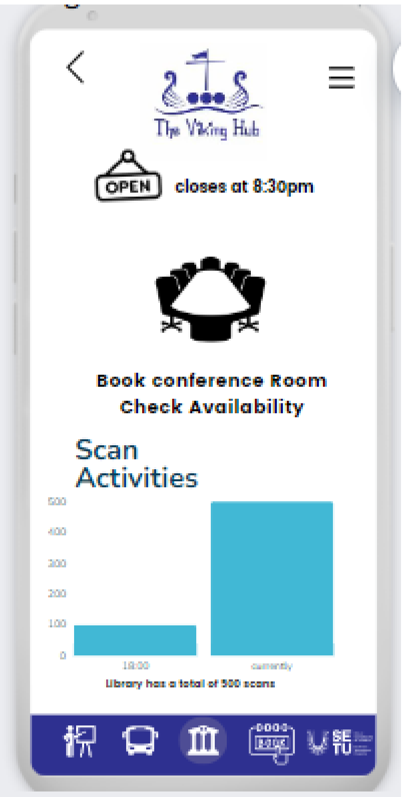


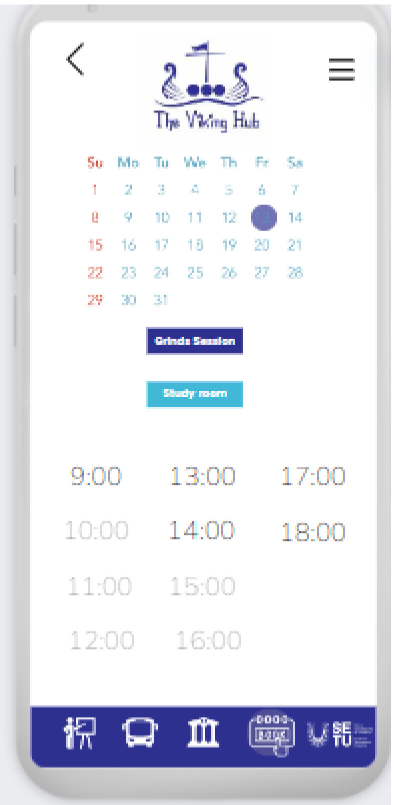


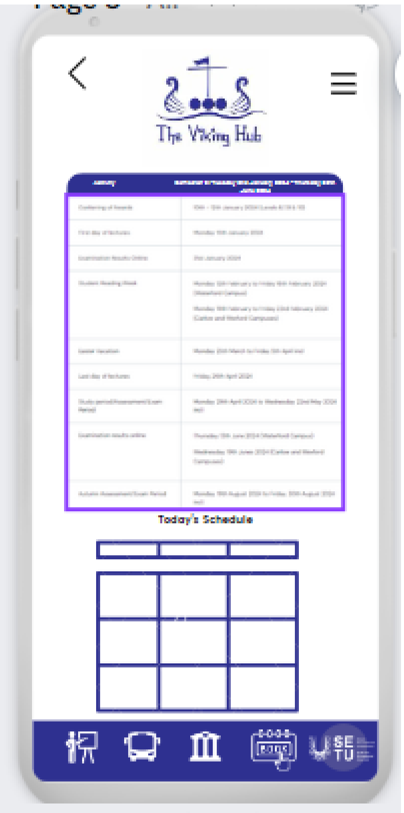










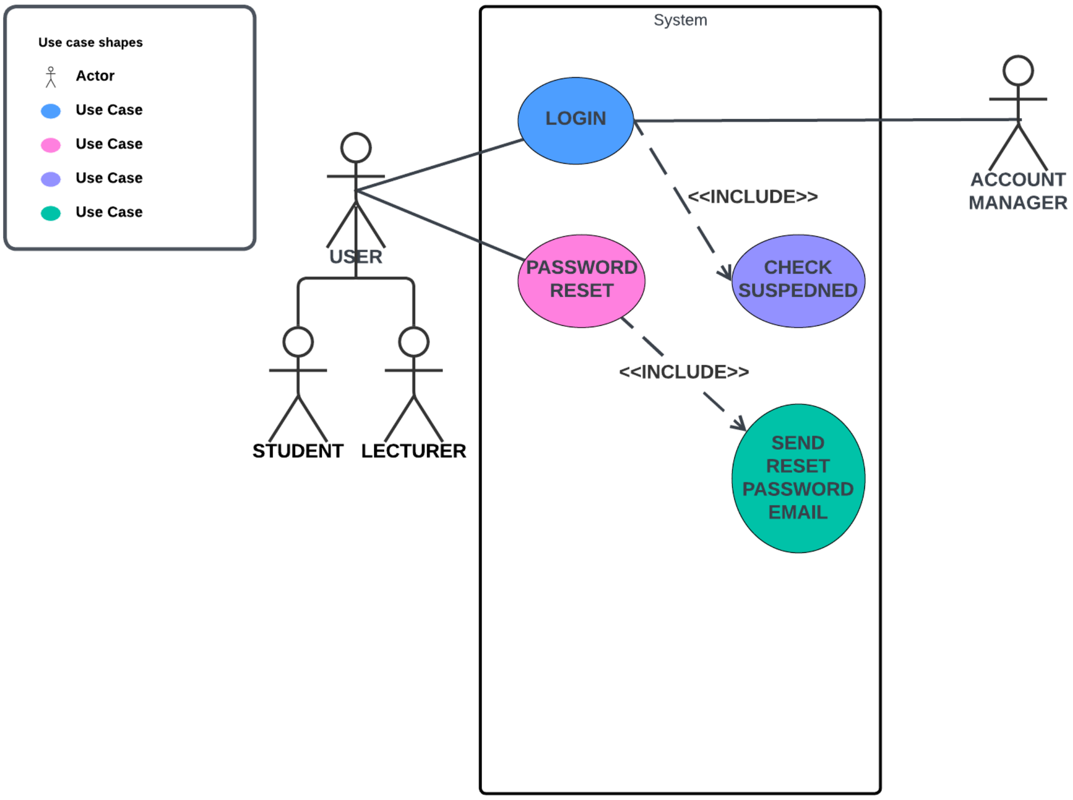


Activity	[illegible]
Conferring of Awards	10th - 12th January 2024 (Levels 8/9 & 10)
First day of lectures	Monday 15th January 2024
Examination Results Online	31st January 2024
Student Reading Week	Monday 12th February to Friday 16th February 2024 (Waterford Campus) Monday 19th February to Friday 23rd February 2024 (Carlow and Wexford Campuses)
Easter Vacation	Monday 25th March to Friday 5th April incl
Last day of lectures	Friday 26th April 2024
Study period/Assessment/Exam Period	Monday 29th April 2024 to Wednesday 22nd May 2024 incl
Examination results online	Thursday 13th June 2024 (Waterford Campus) Wednesday 19th June 2024 (Carlow and Wexford Campuses)
Autumn Assessment/Exam Period	Monday 19th August 2024 to Friday 30th August 2024 incl

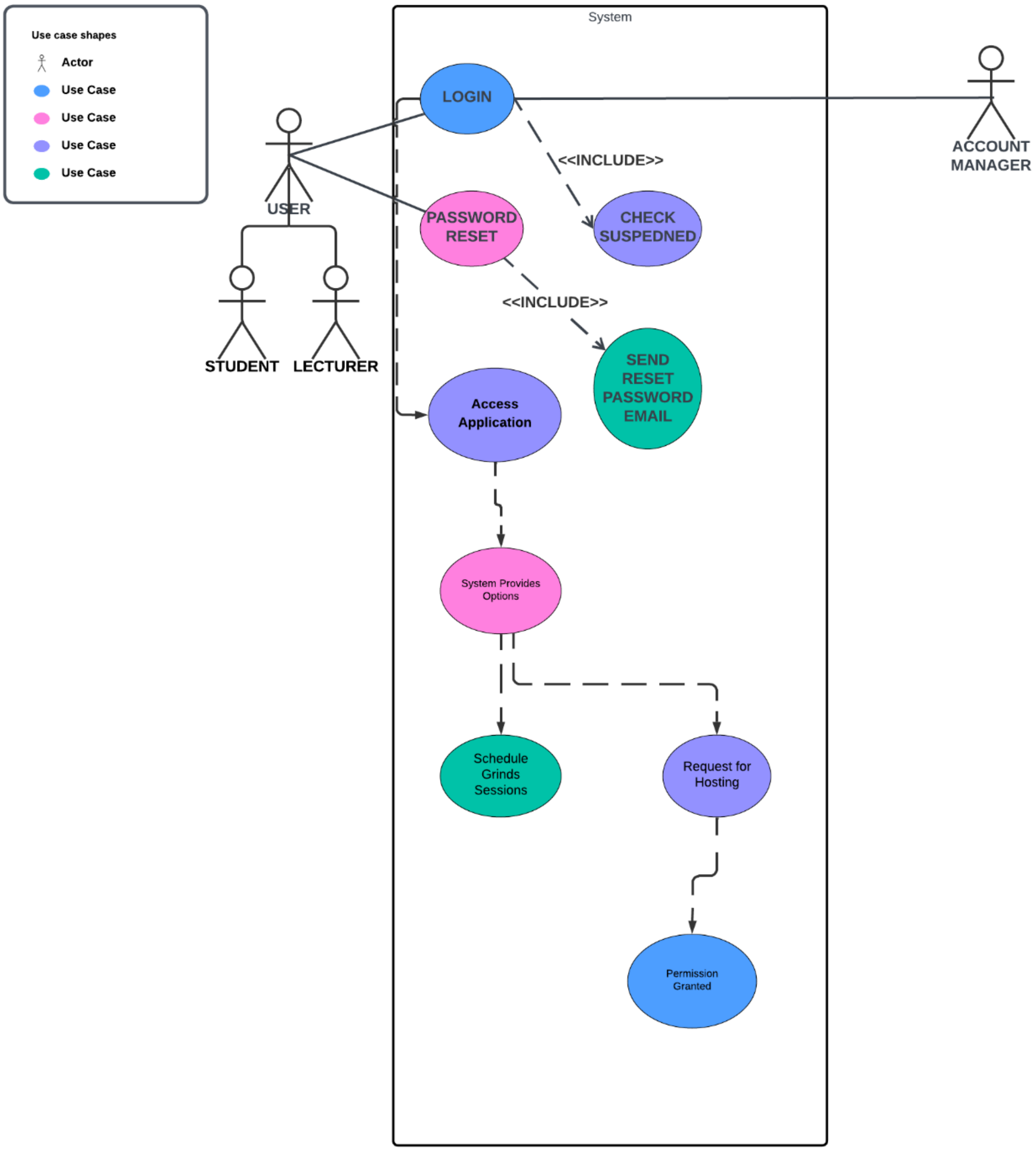
Today's Schedule

3.7 Use Case Models

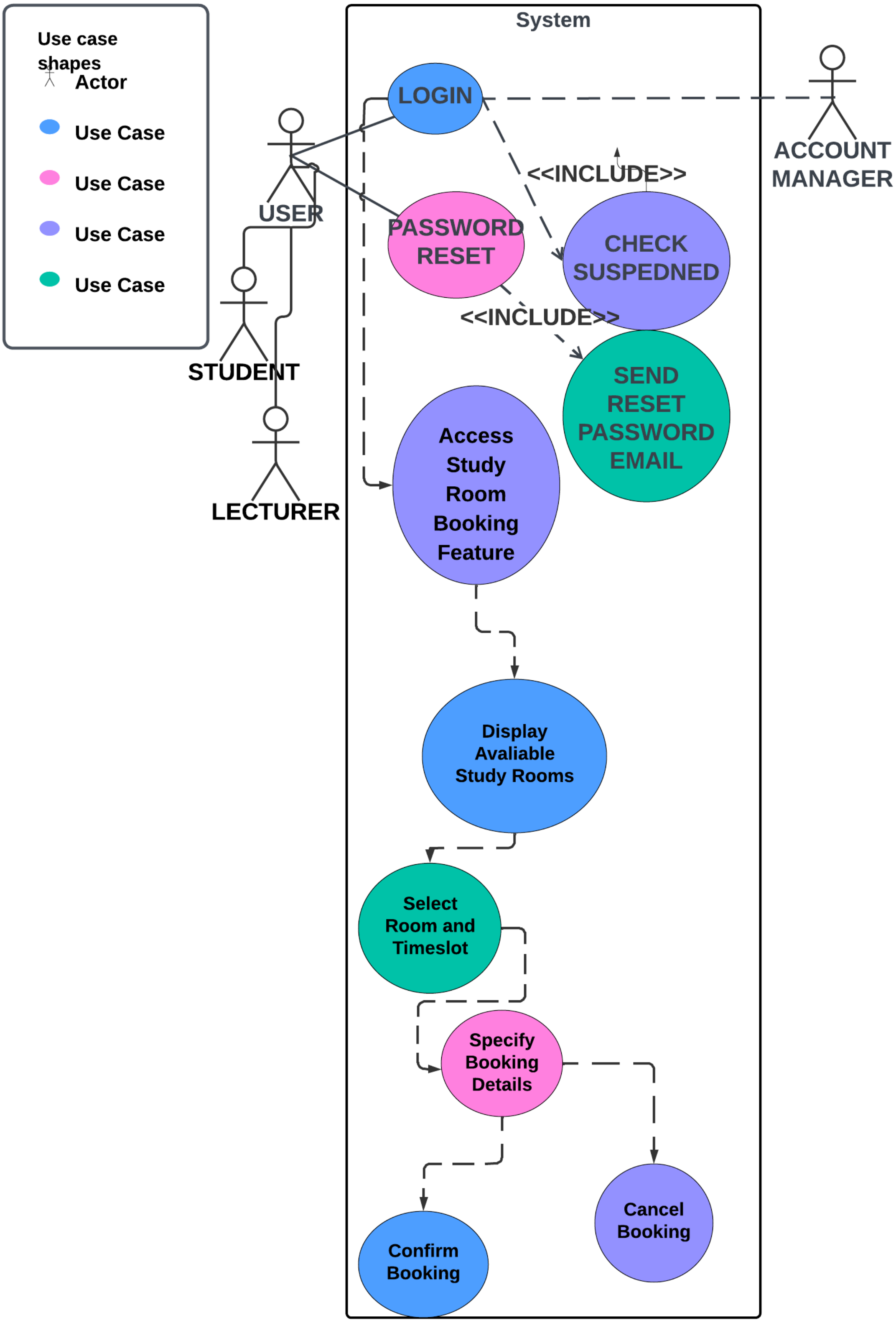
Login Use Case Model



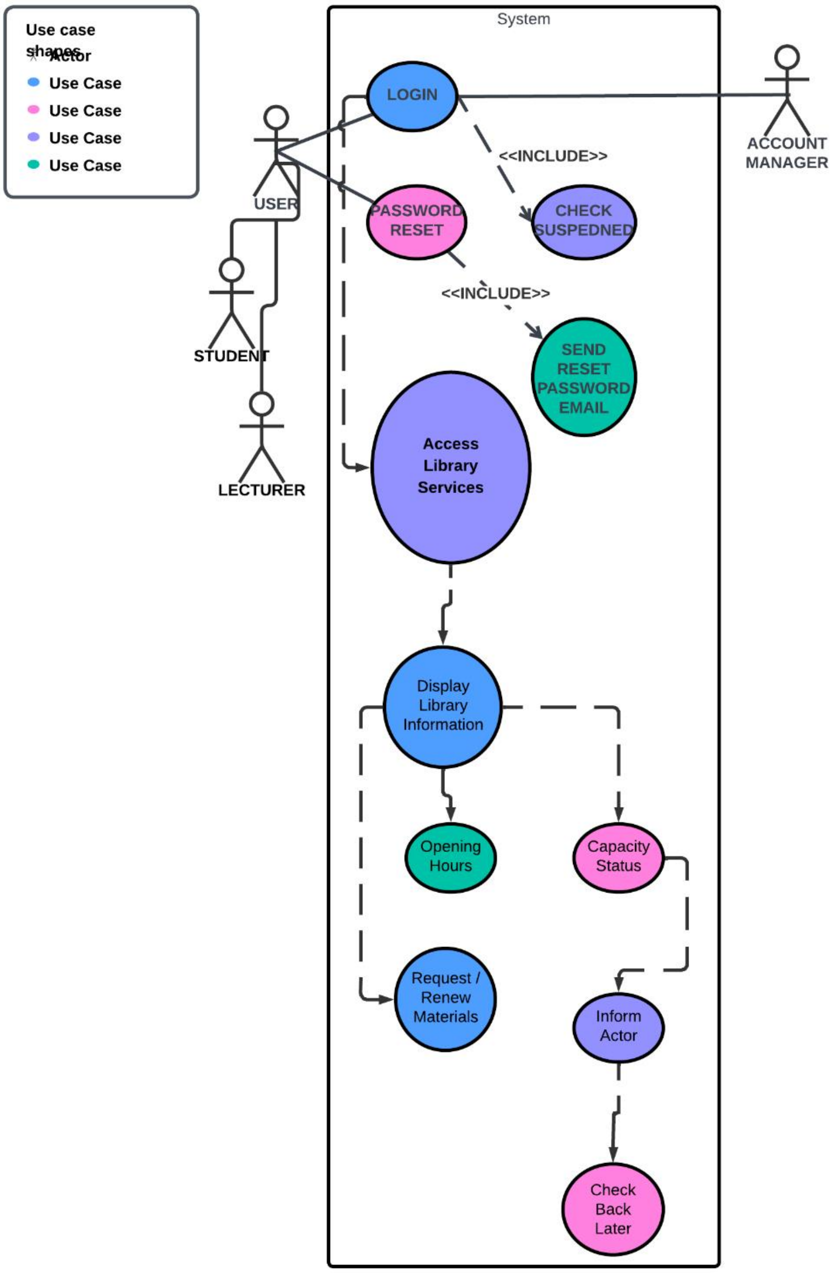
Host Grinds Use Case Model



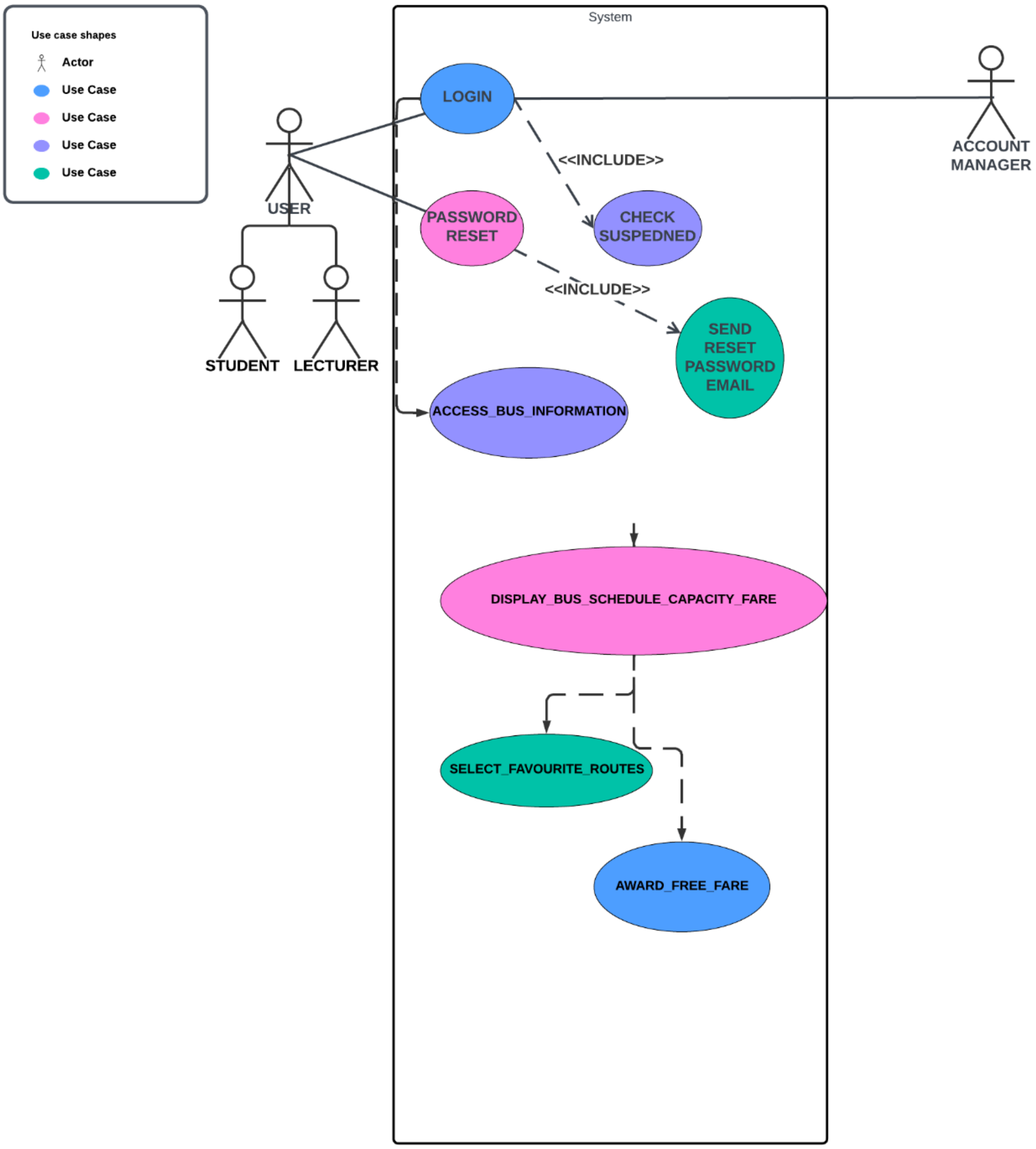
Book Study Room Use Case Model



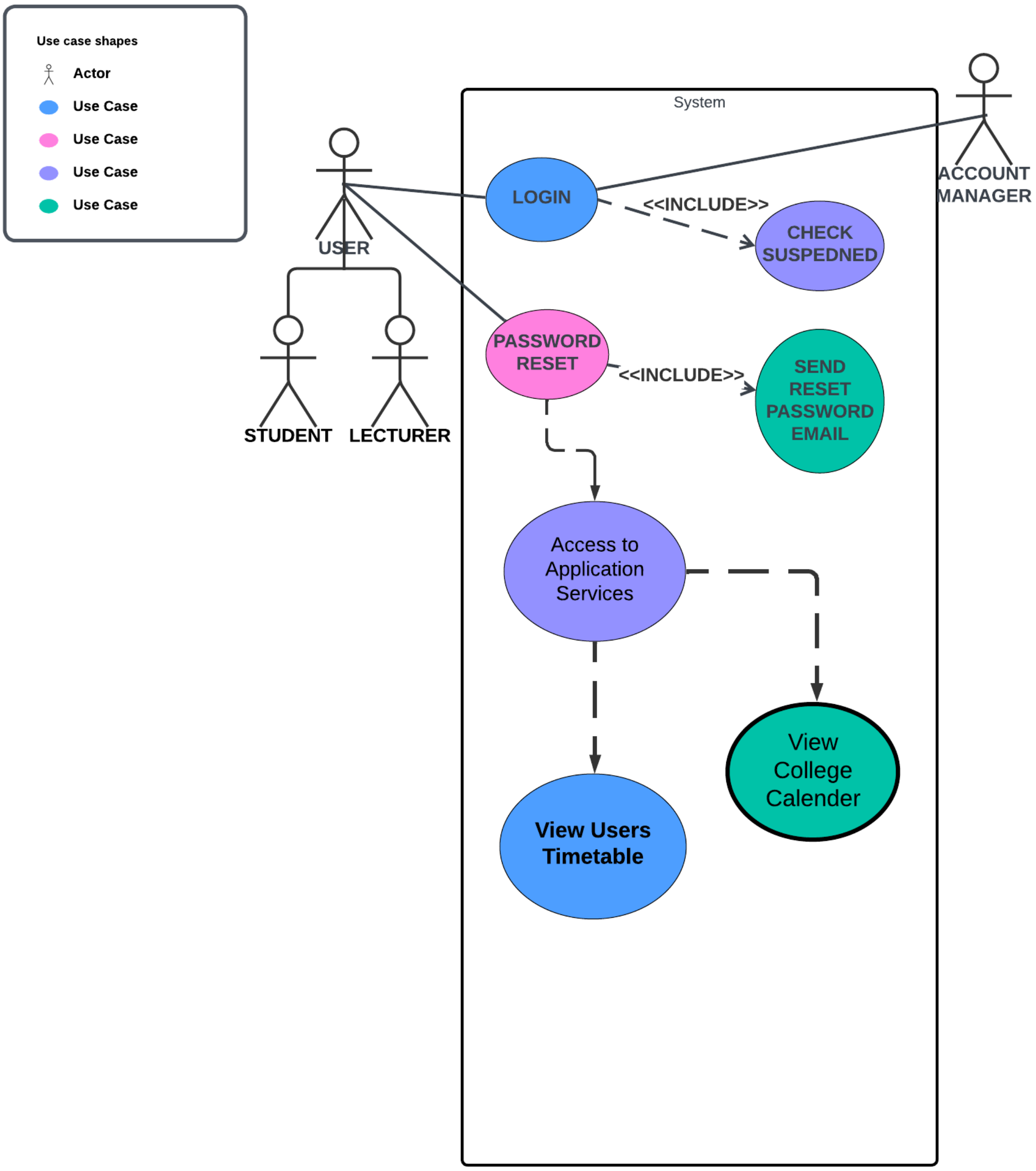
Library Services Use Case Model



Bus Services Use Case Model

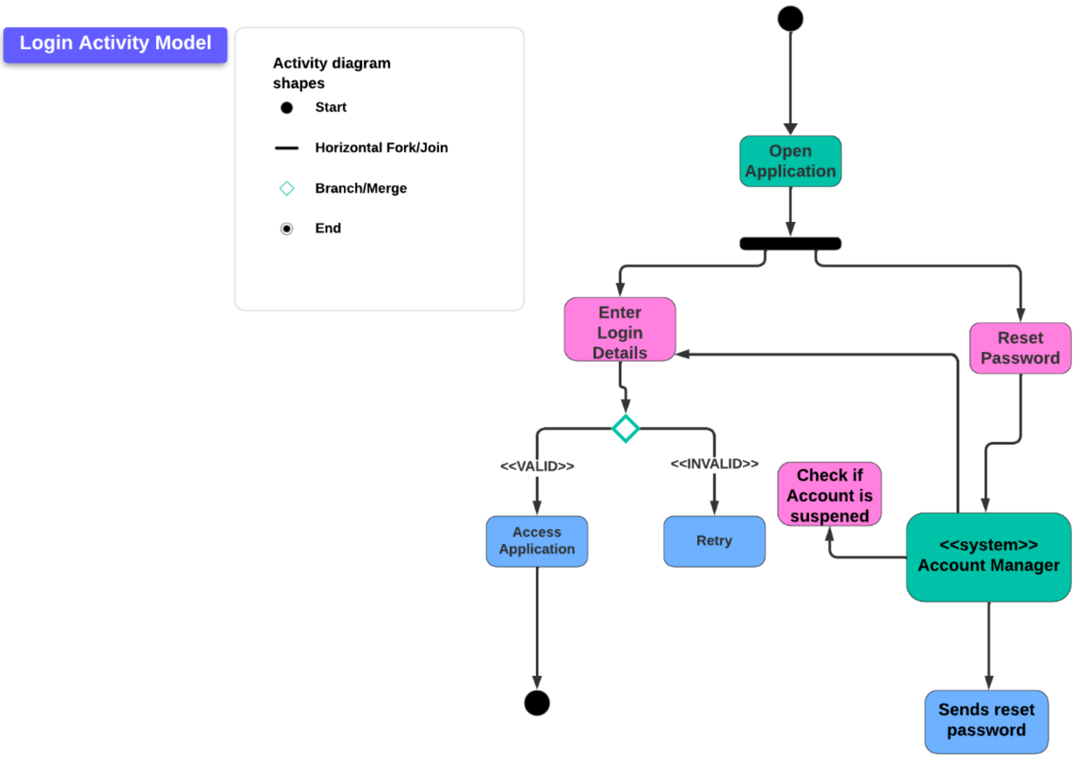


Timetable and Calendar Use Case Model

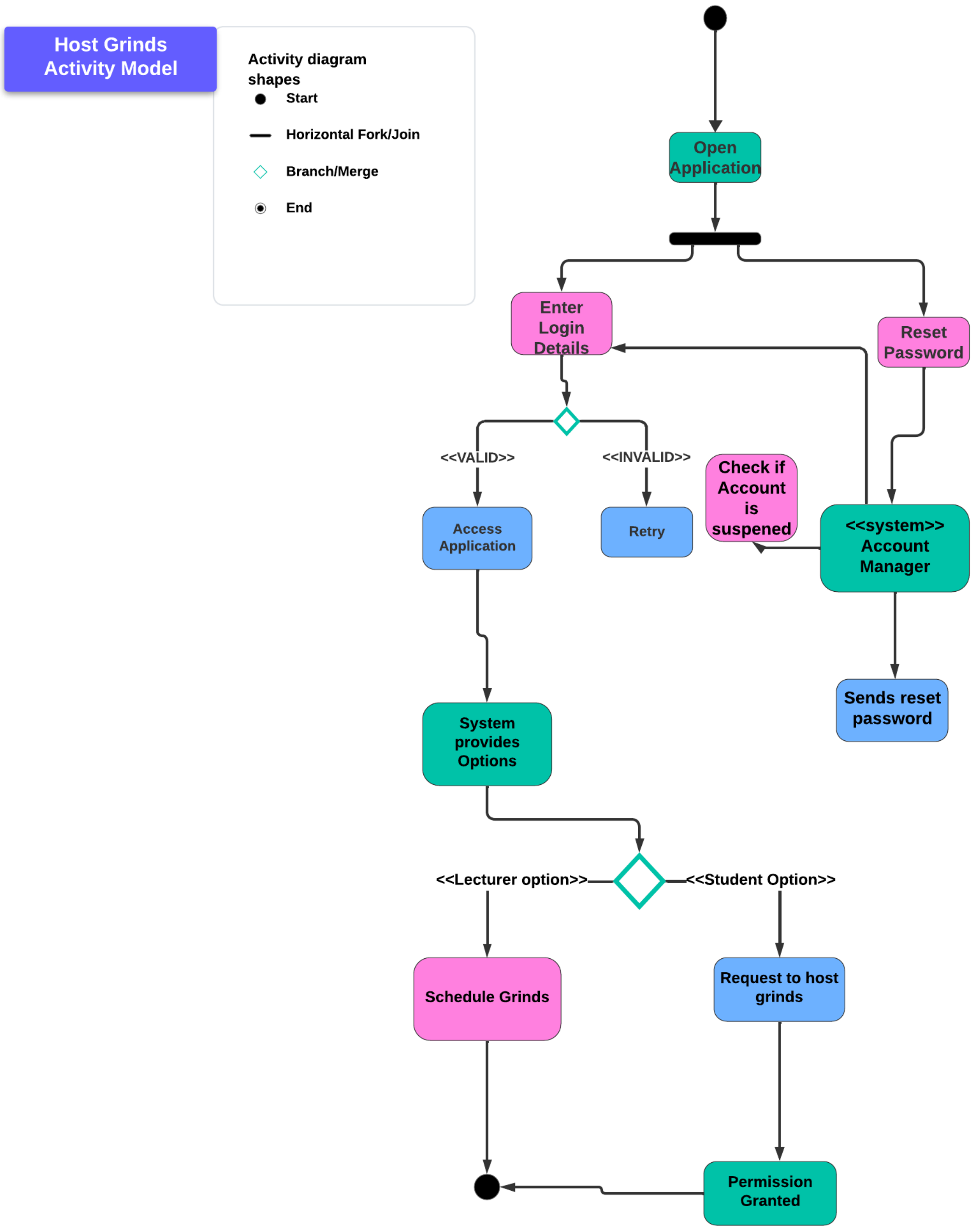


3.8 Activity Models

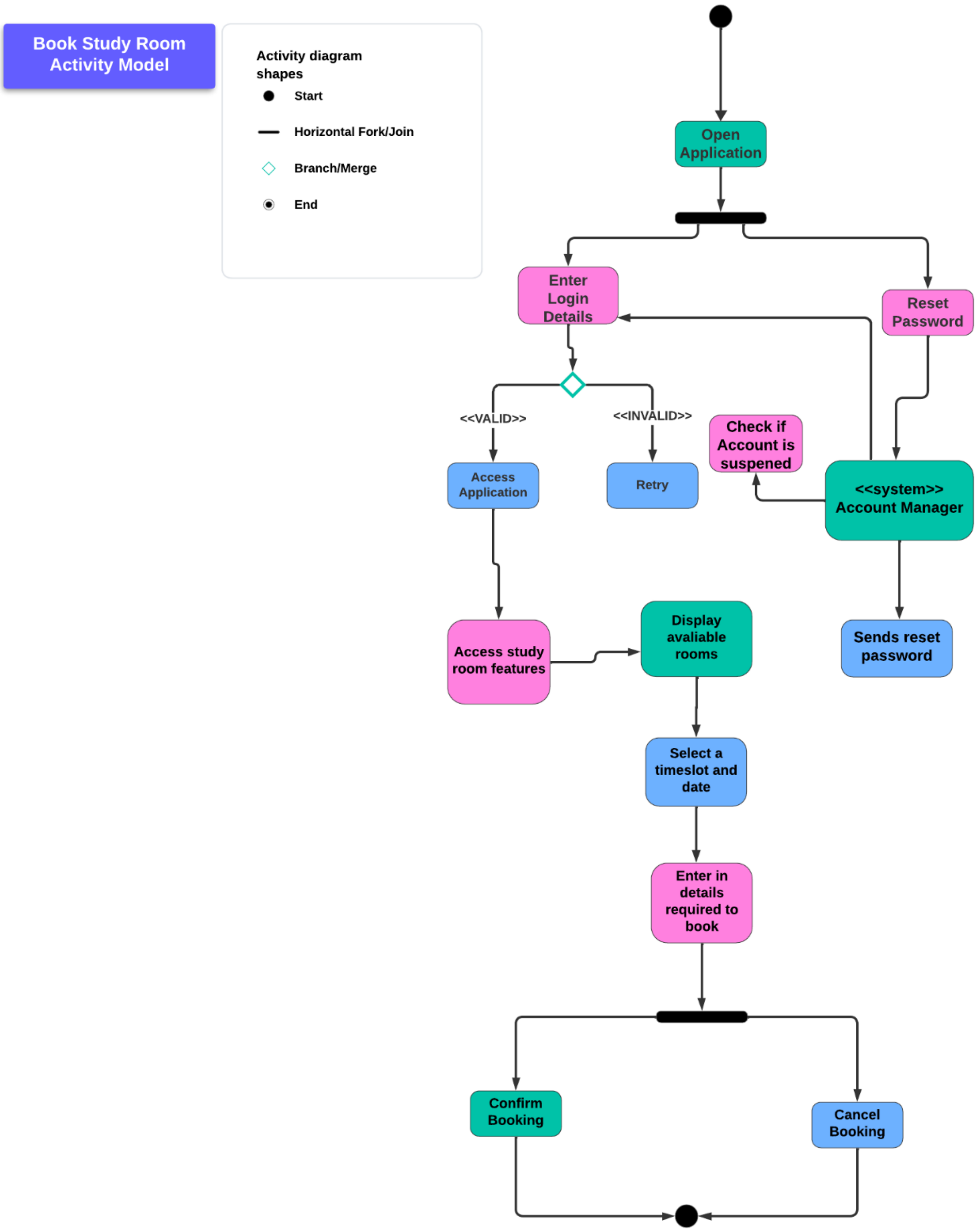
Login Activity Model



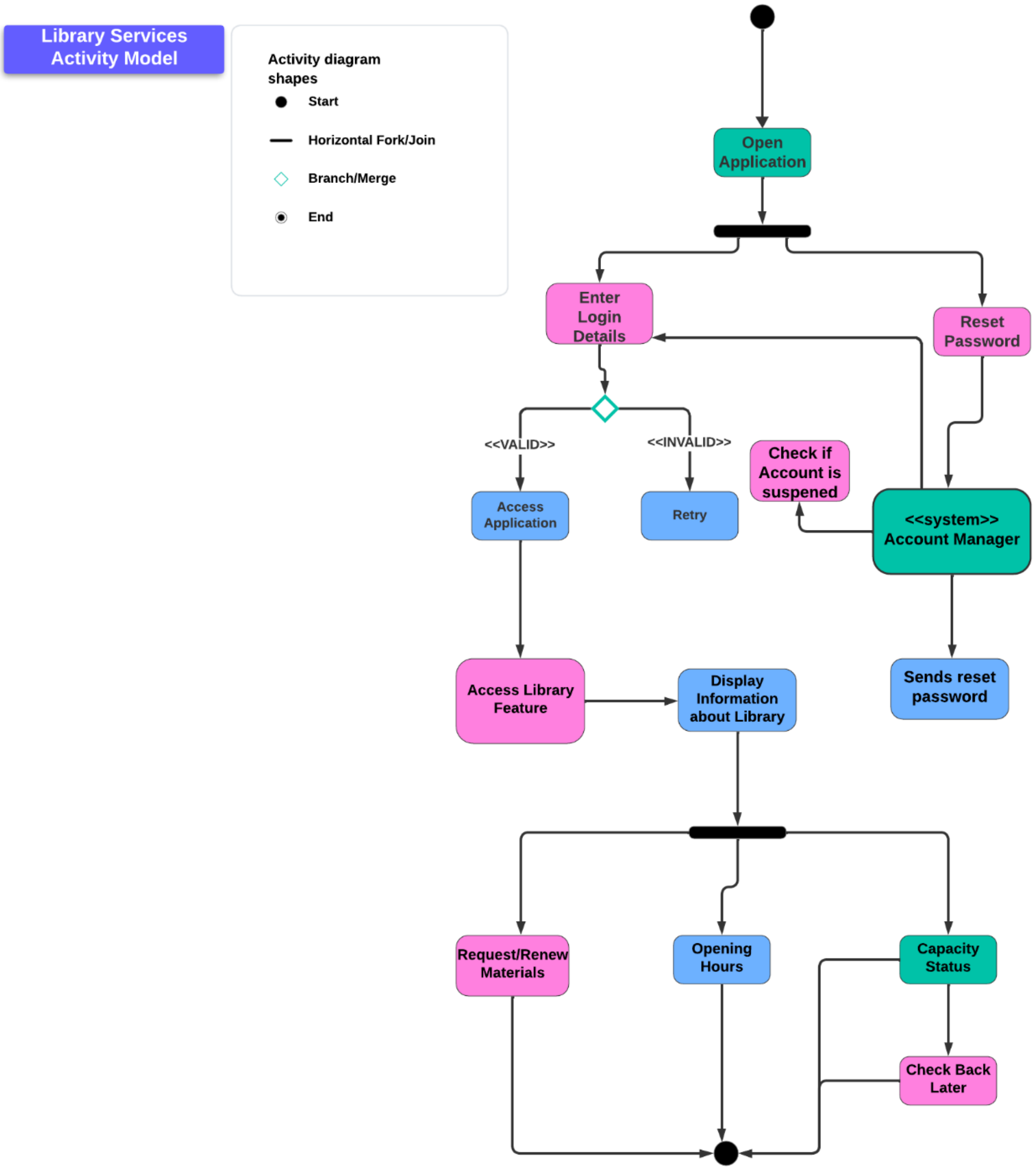
Host Grinds Activity Model



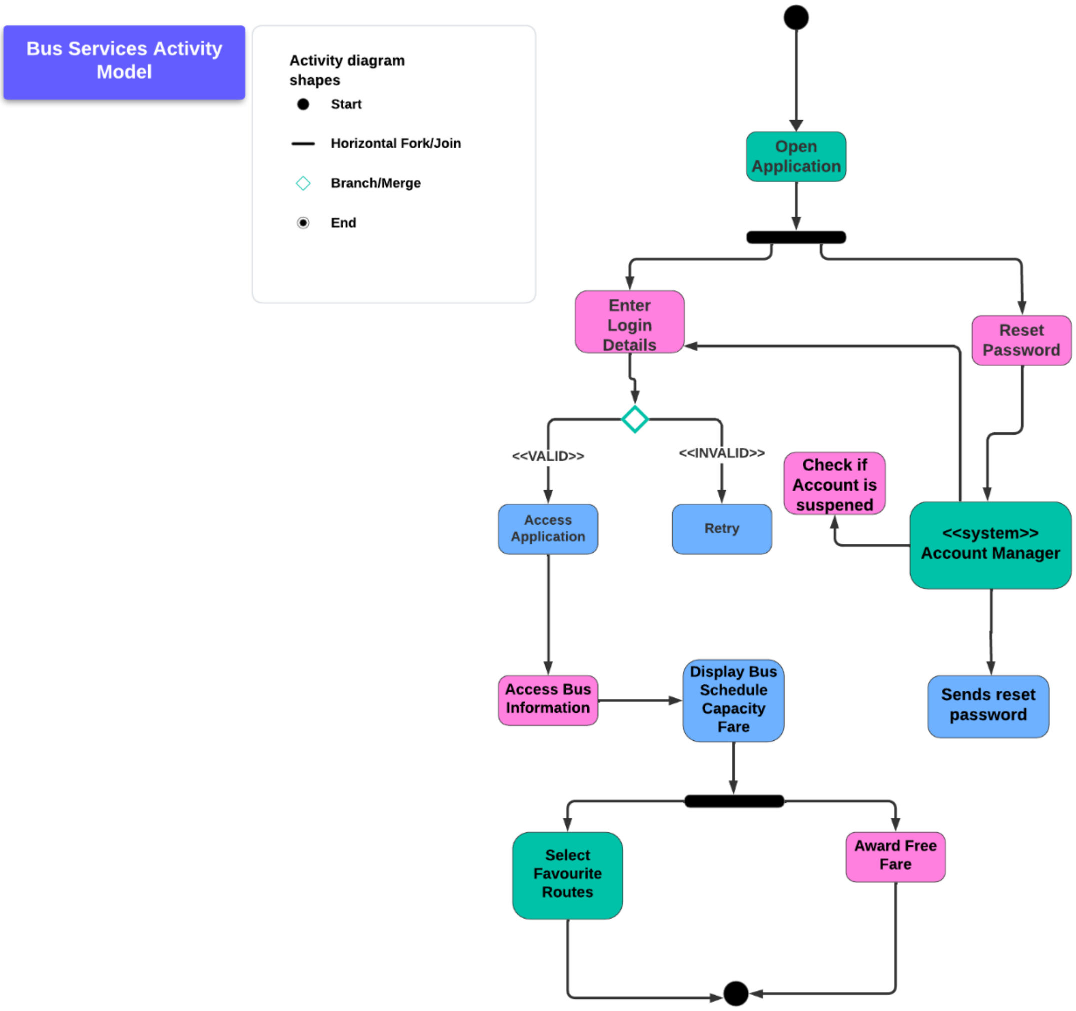
Book Study Room Activity Model



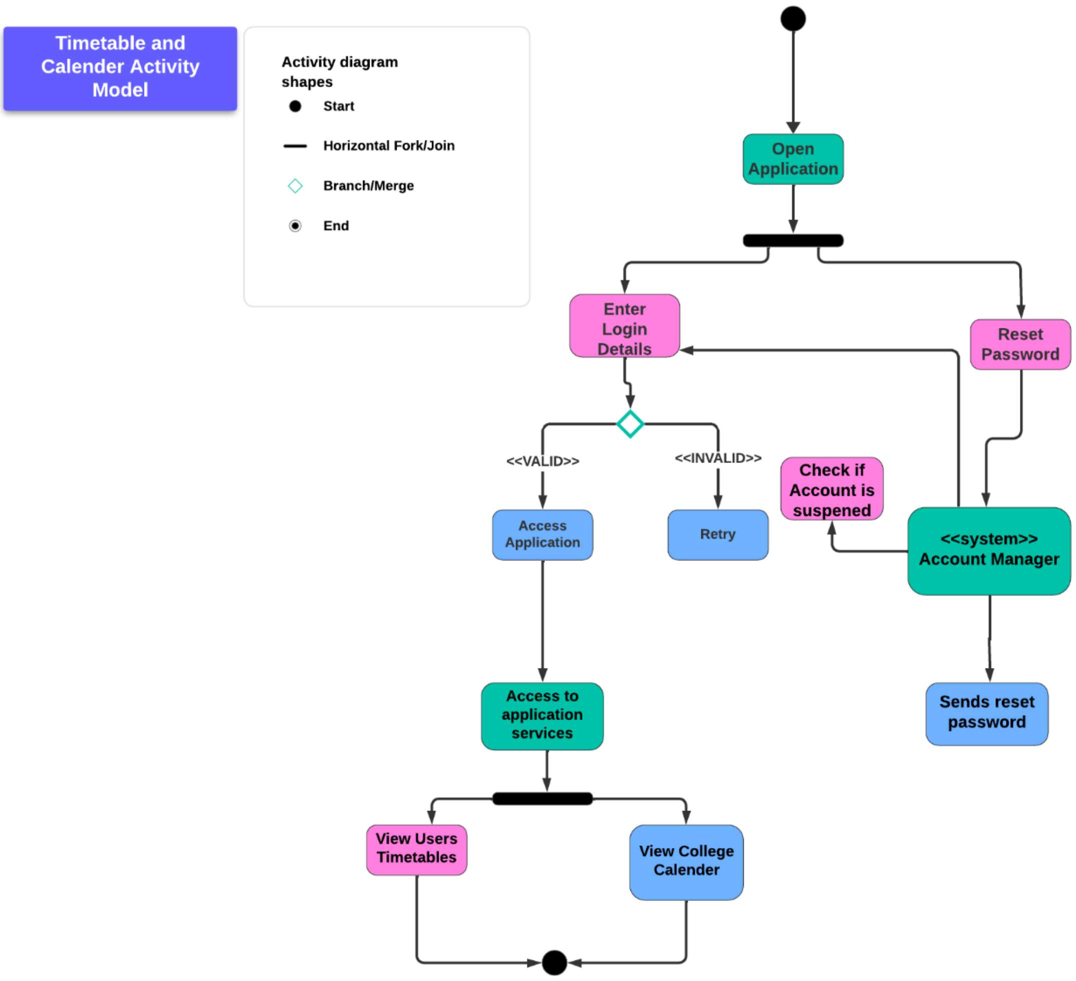
Library Services Activity Model



Bus Services Activity Model

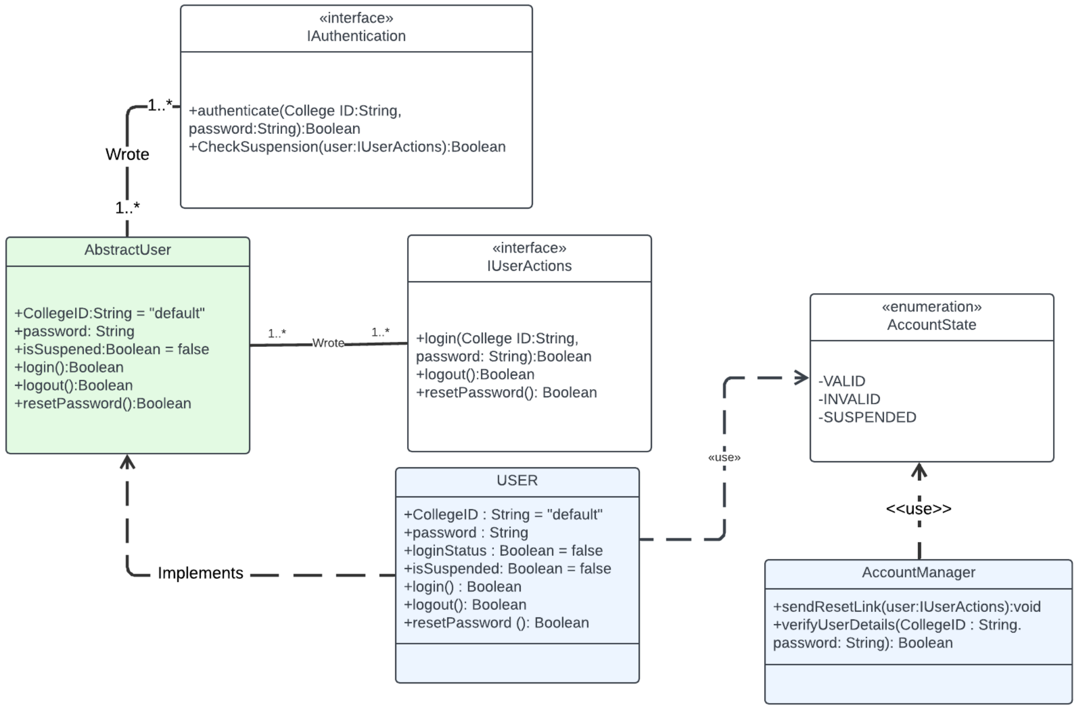


Timetable and Calender Activity Model

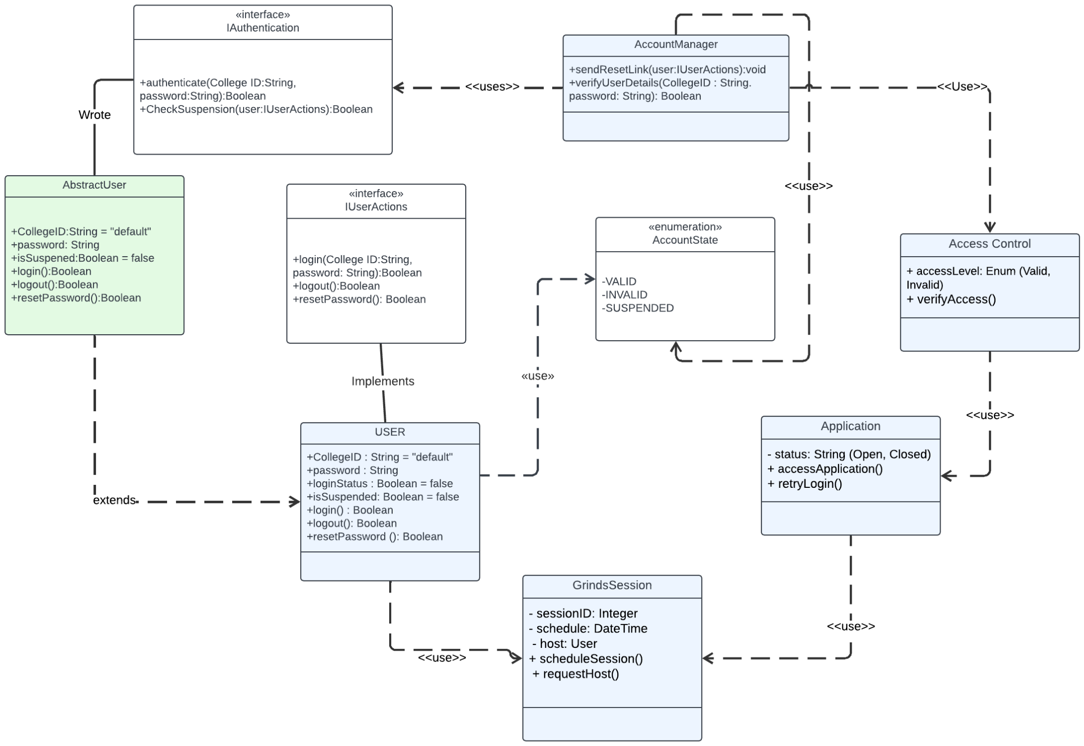


3.9 CLASS MODELS

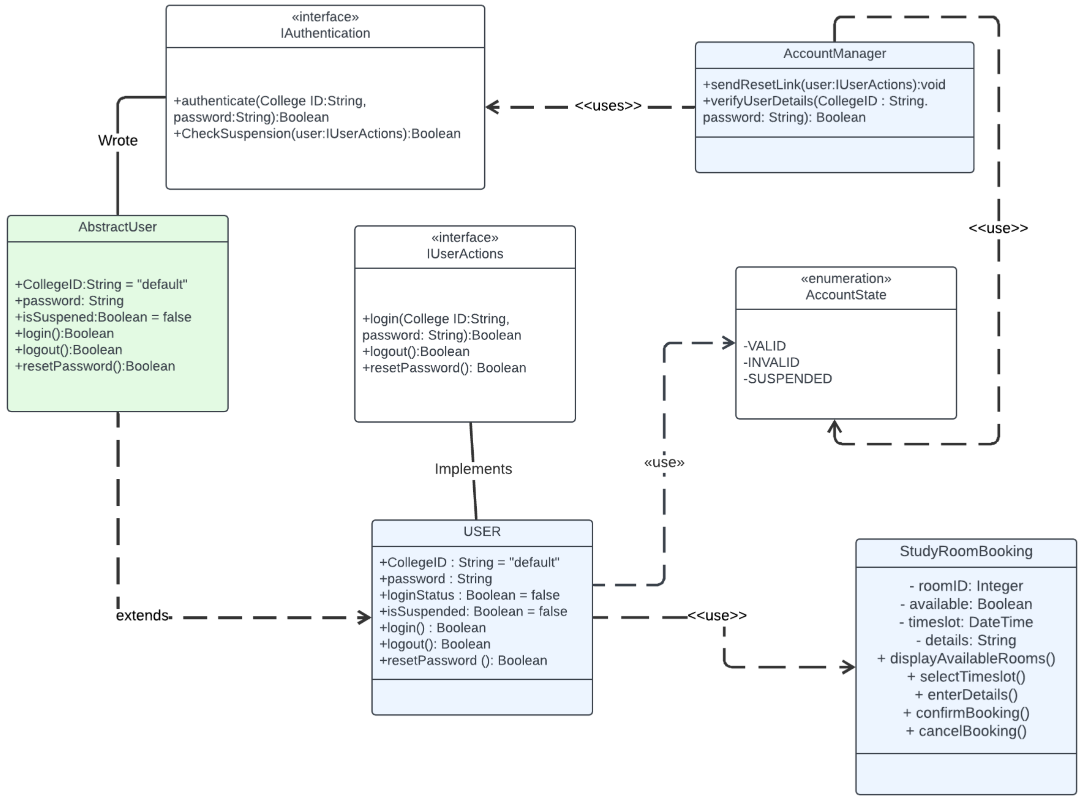
Login Class Model


«interface»
IAuthentication
+authenticate(College ID:String,
password:String):Boolean
+CheckSuspension(user:IUserActions):Boolean
1..*
Wrote
1..*
AbstractUser
+CollegeID:String = "default"
+password: String
+isSuspened:Boolean = false
+login():Boolean
+logout():Boolean
+resetPassword():Boolean
1..*
Wrote
1..*
«interface»
IUserActions
+login(College ID:String,
password: String):Boolean
+logout():Boolean
+resetPassword(): Boolean
«enumeration»
AccountState
-VALID
-INVALID
-SUSPENDED
«use»
<<use>>
USER
+CollegeID : String = "default"
+password : String
+loginStatus : Boolean = false
+isSuspended: Boolean = false
+login() : Boolean
+logout(): Boolean
+resetPassword (): Boolean
Implements
AccountManager
+sendResetLink(user:IUserActions):void
+verifyUserDetails(CollegeID : String.
password: String): Boolean

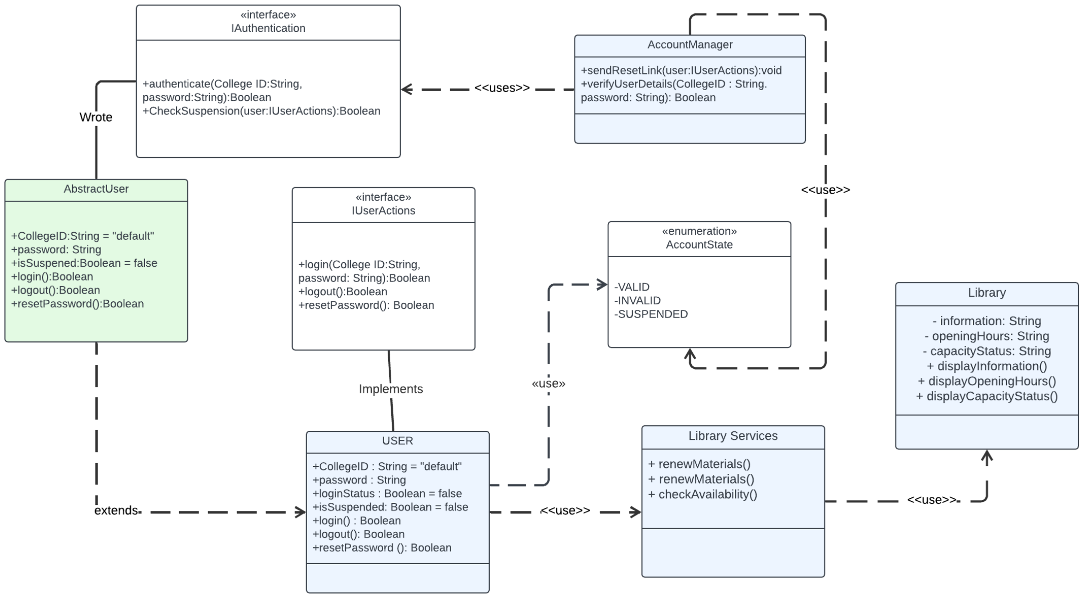
Host Grinds Class Model


«interface»
IAuthentication
+authenticate(College ID:String, password:String):Boolean
+CheckSuspension(user:IUserActions):Boolean
<<uses>>
AccountManager
+sendResetLink(user:IUserActions):void
+verifyUserDetails(CollegeID : String. password: String): Boolean
<<Use>>
Wrote
AbstractUser
+CollegeID:String = "default"
+password: String
+isSuspened:Boolean = false
+login():Boolean
+logout():Boolean
+resetPassword():Boolean
«interface»
IUserActions
+login(College ID:String, password: String):Boolean
+logout():Boolean
+resetPassword(): Boolean
<<use>>
«enumeration»
AccountState
-VALID
-INVALID
-SUSPENDED
Access Control
+ accessLevel: Enum (Valid, Invalid)
+ verifyAccess()
Implements
«use»
<<use>>
Application
- status: String (Open, Closed)
+ accessApplication()
+ retryLogin()
extends
USER
+CollegeID : String = "default"
+password : String
+loginStatus : Boolean = false
+isSuspended: Boolean = false
+login() : Boolean
+logout(): Boolean
+resetPassword (): Boolean
<<use>>
GrindsSession
- sessionID: Integer
- schedule: DateTime
- host: User
+ scheduleSession()
+ requestHost()
<<use>>

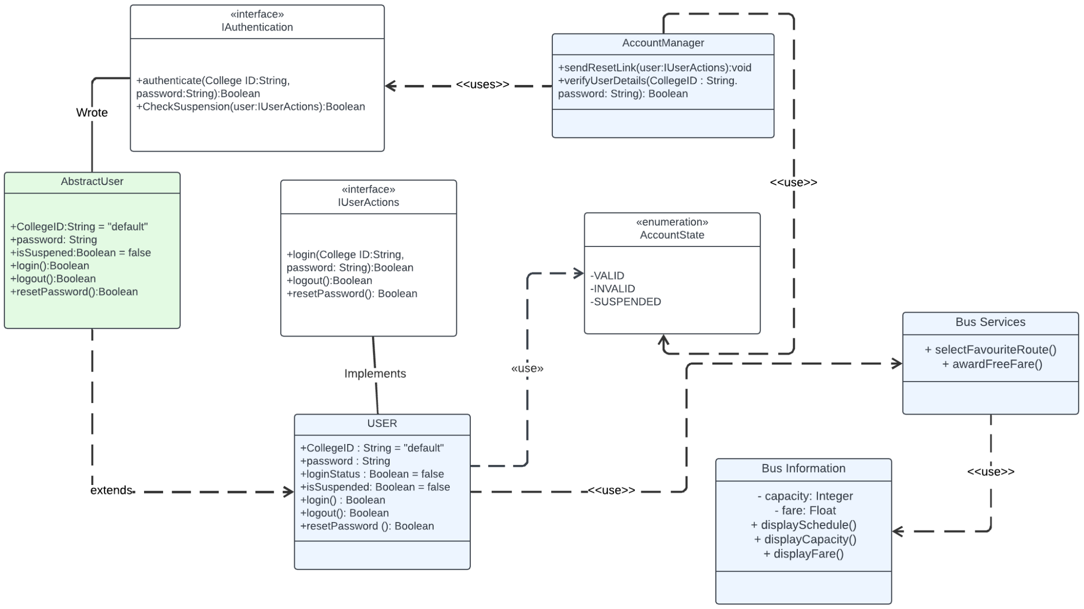
Book Study Room Class Model



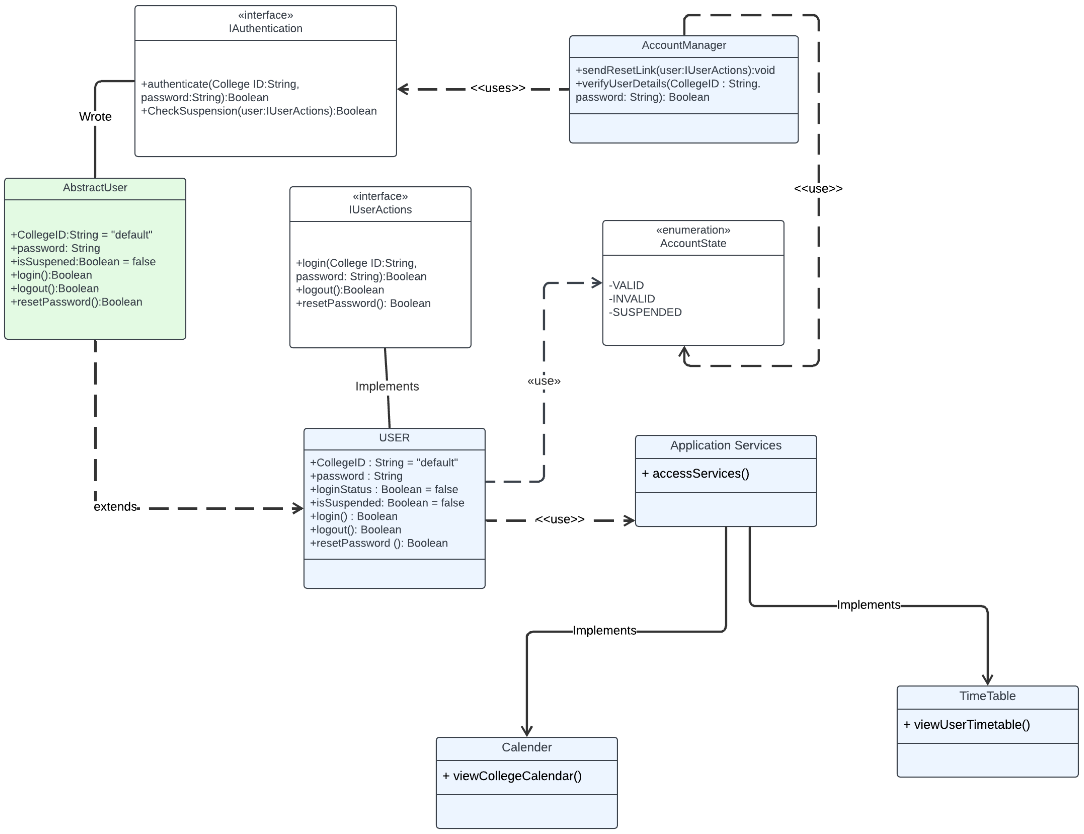
Library Services Class Model



Book Bus Class Model



Timetable and Calendar Class Model


«interface»
IAuthentication
+authenticate(College ID:String, password:String):Boolean
+CheckSuspension(user:IUserActions):Boolean
AccountManager
+sendResetLink(user:IUserActions):void
+verifyUserDetails(CollegeID : String. password: String): Boolean
<<uses>>
Wrote
AbstractUser
+CollegeID:String = "default"
+password: String
+isSuspened:Boolean = false
+login():Boolean
+logout():Boolean
+resetPassword():Boolean
«interface»
IUserActions
+login(College ID:String, password: String):Boolean
+logout():Boolean
+resetPassword(): Boolean
<<use>>
«enumeration»
AccountState
-VALID
-INVALID
-SUSPENDED
Implements
«use»
USER
+CollegeID : String = "default"
+password : String
+loginStatus : Boolean = false
+isSuspended: Boolean = false
+login() : Boolean
+logout(): Boolean
+resetPassword (): Boolean
extends
<<use>>
Application Services
+ accessServices()
Implements
Implements
TimeTable
+ viewUserTimetable()
Calender
+ viewCollegeCalendar()

3.10 Conclusion

In conclusion, The Viking Hub is an important, all-in-one platform for SETU Waterford students and staff, combining key services such as bus schedules, study room bookings, and educational support into a single, easily accessible application. The app provides significant value for €10 by providing access to university resources and including benefits such as free bus tickets. As it grows and adapts, The Viking Hub hopes to set new standards for how educational services are delivered, considerably improving student life and the overall educational experience at SETU Waterford.

The Viking Hub team developed their communication, teamwork and time management skills when developing this application. This was a very enjoyable experience as we as a team were able to be as creative and open about our ideas and to be able to put our thoughts together and create a successful application this has boosted our confidence in our ability to work in teams and this attitude and skills will be used in the future for more reports like The Viking Hub.

3.11 References

Home test - setu (local) (no date) *SETU*. Available at: https://www.setu.ie/ (Accessed: 03 February 2024).

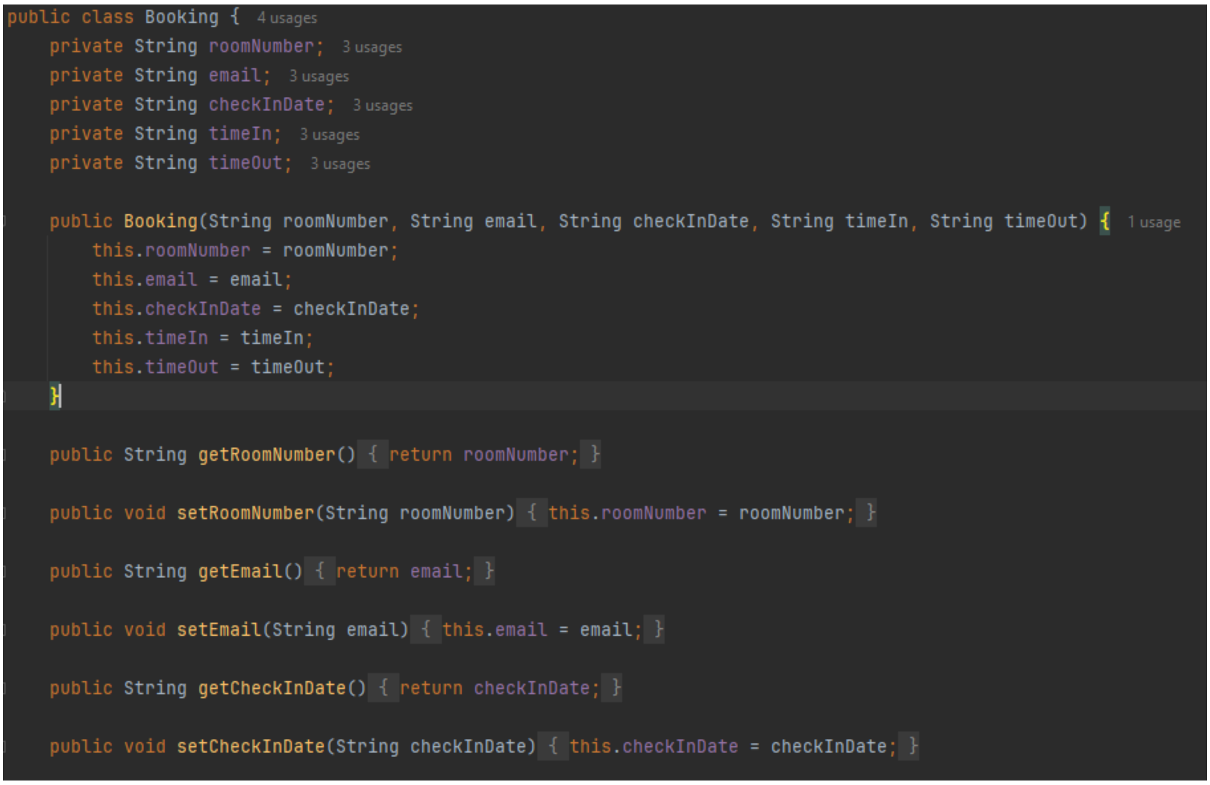
Home (2023) *Dublin Coach*. Available at: https://www.dublincoach.ie/ (Accessed: 03 February 2024).

Leap card (no date) *Leap Card*. Available at: https://about.leapcard.ie/ (Accessed: 03 February 2024).

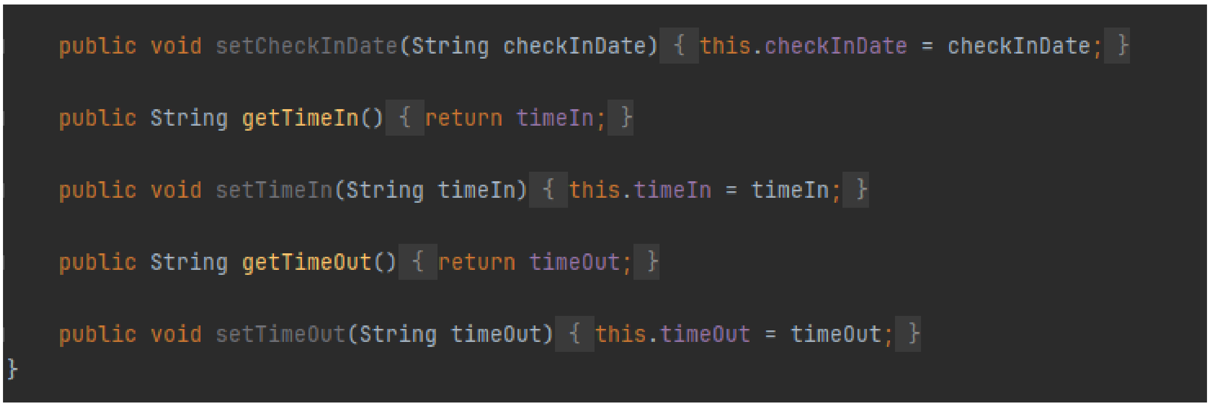
3.12 Appendix Code

Classes

Booking Class

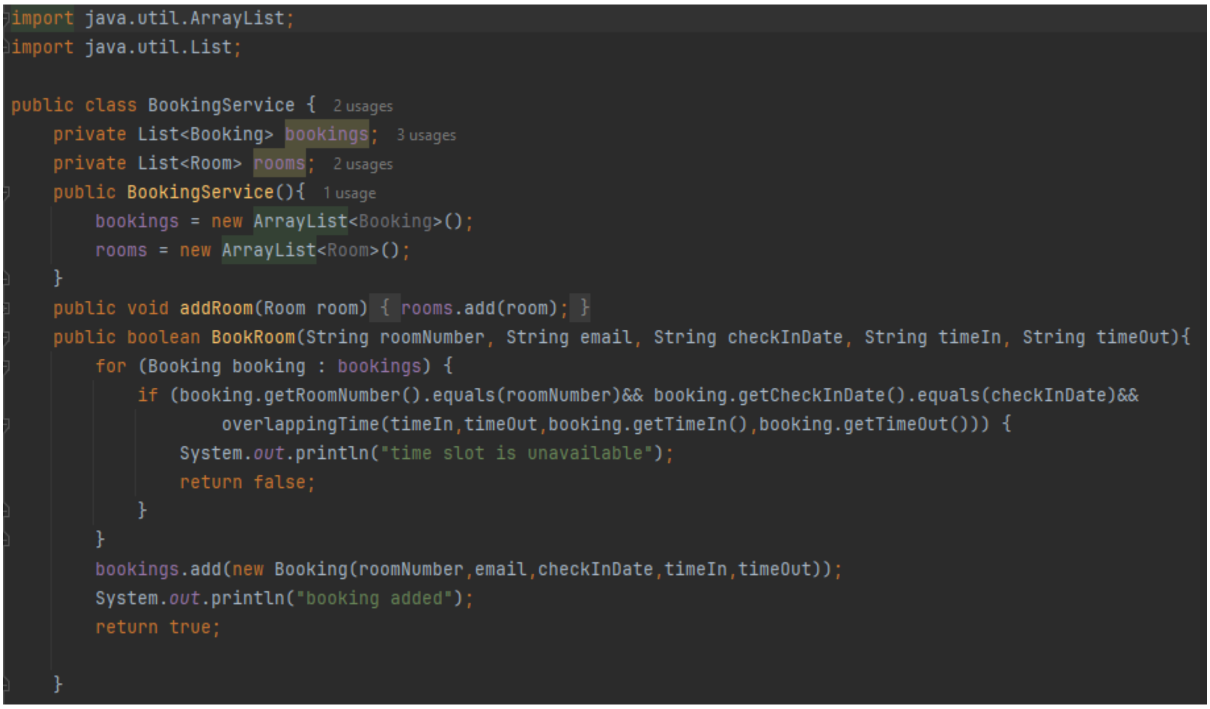


```
public class Booking {  4 usages
    private String roomNumber;  3 usages
    private String email;  3 usages
    private String checkInDate;  3 usages
    private String timeIn;  3 usages
    private String timeOut;  3 usages

    public Booking(String roomNumber, String email, String checkInDate, String timeIn, String timeOut) {  1 usage
        this.roomNumber = roomNumber;
        this.email = email;
        this.checkInDate = checkInDate;
        this.timeIn = timeIn;
        this.timeOut = timeOut;
    }

    public String getRoomNumber() { return roomNumber; }

    public void setRoomNumber(String roomNumber) { this.roomNumber = roomNumber; }

    public String getEmail() { return email; }

    public void setEmail(String email) { this.email = email; }

    public String getCheckInDate() { return checkInDate; }

    public void setCheckInDate(String checkInDate) { this.checkInDate = checkInDate; }
```

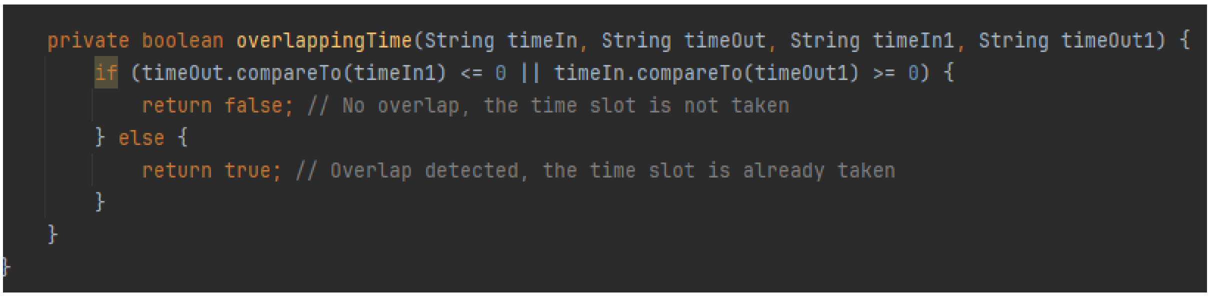


```
    public void setCheckInDate(String checkInDate) { this.checkInDate = checkInDate; }

    public String getTimeIn() { return timeIn; }

    public void setTimeIn(String timeIn) { this.timeIn = timeIn; }

    public String getTimeOut() { return timeOut; }

    public void setTimeOut(String timeOut) { this.timeOut = timeOut; }
}
```

BookingService Class

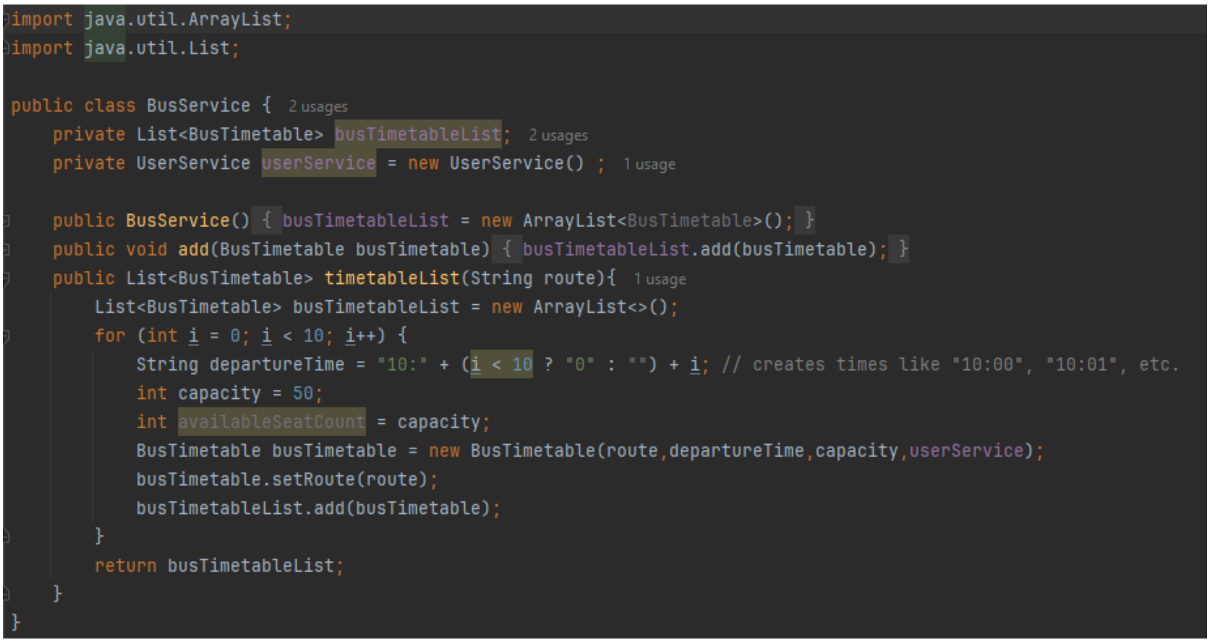


```
import java.util.ArrayList;
import java.util.List;

public class BookingService {  2 usages
    private List<Booking> bookings;  3 usages
    private List<Room> rooms;  2 usages
    public BookingService(){  1 usage
        bookings = new ArrayList<Booking>();
        rooms = new ArrayList<Room>();
    }
    public void addRoom(Room room) { rooms.add(room); }
    public boolean BookRoom(String roomNumber, String email, String checkInDate, String timeIn, String timeOut){
        for (Booking booking : bookings) {
            if (booking.getRoomNumber().equals(roomNumber)&& booking.getCheckInDate().equals(checkInDate)&&
                    overlappingTime(timeIn,timeOut,booking.getTimeIn(),booking.getTimeOut())) {
                System.out.println("time slot is unavailable");
                return false;
            }
        }
        bookings.add(new Booking(roomNumber,email,checkInDate,timeIn,timeOut));
        System.out.println("booking added");
        return true;

    }
```



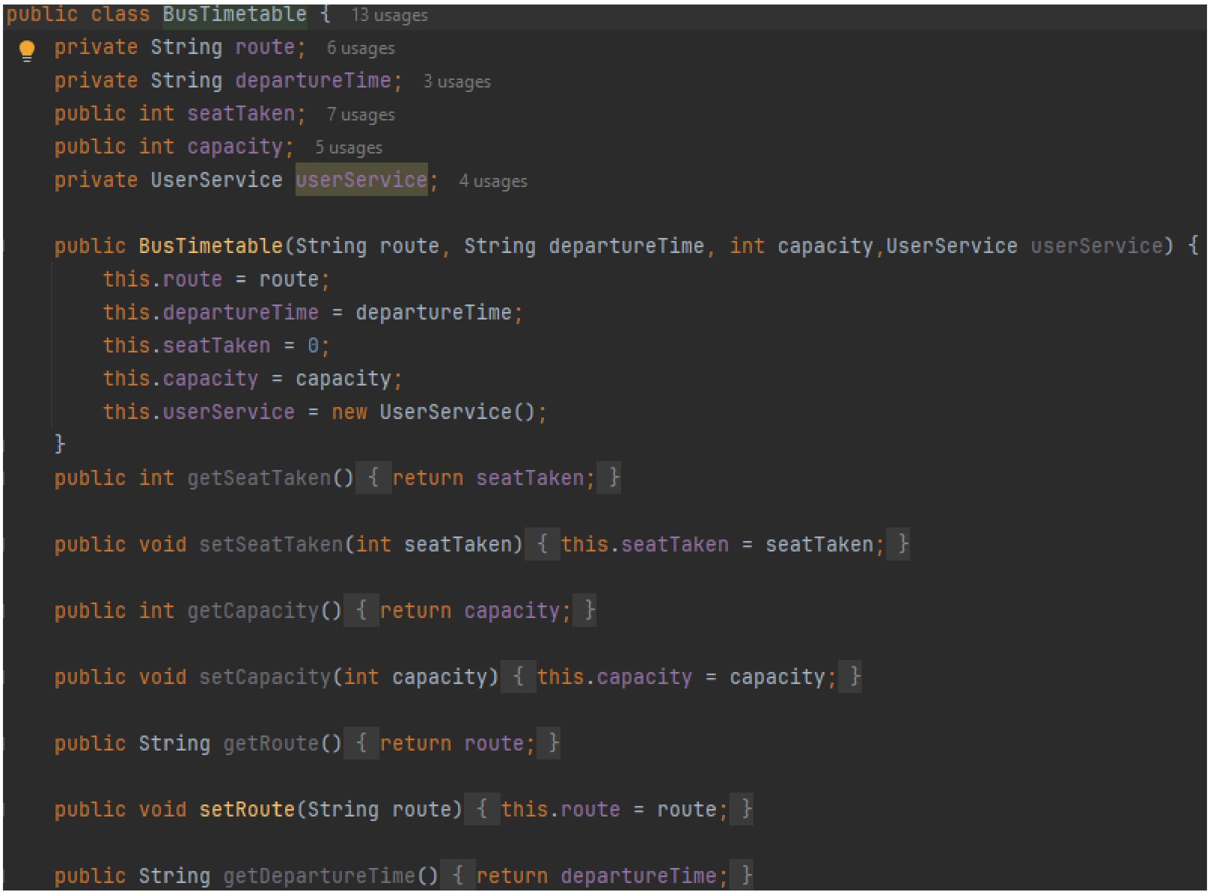
```
    private boolean overlappingTime(String timeIn, String timeOut, String timeIn1, String timeOut1) {
        if (timeOut.compareTo(timeIn1) <= 0 || timeIn.compareTo(timeOut1) >= 0) {
            return false; // No overlap, the time slot is not taken
        } else {
            return true; // Overlap detected, the time slot is already taken
        }
    }
}
```

BusService Class

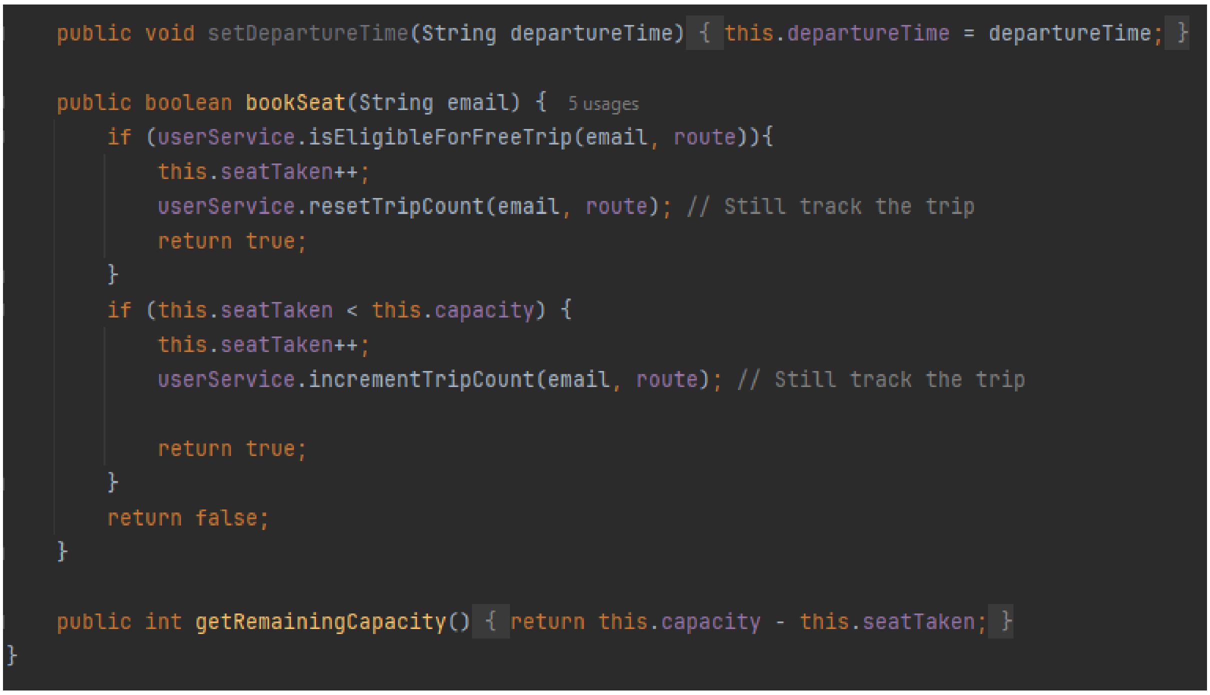


```
import java.util.ArrayList;
import java.util.List;

public class BusService {  2 usages
    private List<BusTimetable> busTimetableList;  2 usages
    private UserService userService = new UserService() ;  1 usage

    public BusService() { busTimetableList = new ArrayList<BusTimetable>(); }
    public void add(BusTimetable busTimetable) { busTimetableList.add(busTimetable); }
    public List<BusTimetable> timetableList(String route){  1 usage
        List<BusTimetable> busTimetableList = new ArrayList<>();
        for (int i = 0; i < 10; i++) {
            String departureTime = "10:" + (i < 10 ? "0" : "") + i; // creates times like "10:00", "10:01", etc.
            int capacity = 50;
            int availableSeatCount = capacity;
            BusTimetable busTimetable = new BusTimetable(route,departureTime,capacity,userService);
            busTimetable.setRoute(route);
            busTimetableList.add(busTimetable);
        }
        return busTimetableList;
    }
}
```

BusTimetable Class

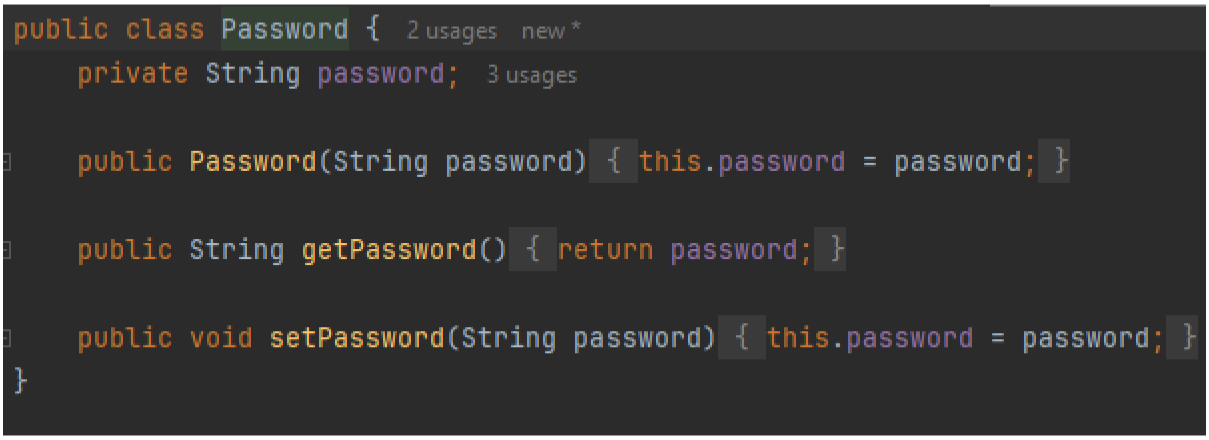


```
public class BusTimetable {  13 usages
    private String route;  6 usages
    private String departureTime;  3 usages
    public int seatTaken;  7 usages
    public int capacity;  5 usages
    private UserService userService;  4 usages

    public BusTimetable(String route, String departureTime, int capacity,UserService userService) {
        this.route = route;
        this.departureTime = departureTime;
        this.seatTaken = 0;
        this.capacity = capacity;
        this.userService = new UserService();
    }
    public int getSeatTaken() { return seatTaken; }

    public void setSeatTaken(int seatTaken) { this.seatTaken = seatTaken; }

    public int getCapacity() { return capacity; }

    public void setCapacity(int capacity) { this.capacity = capacity; }

    public String getRoute() { return route; }

    public void setRoute(String route) { this.route = route; }

    public String getDepartureTime() { return departureTime; }
```



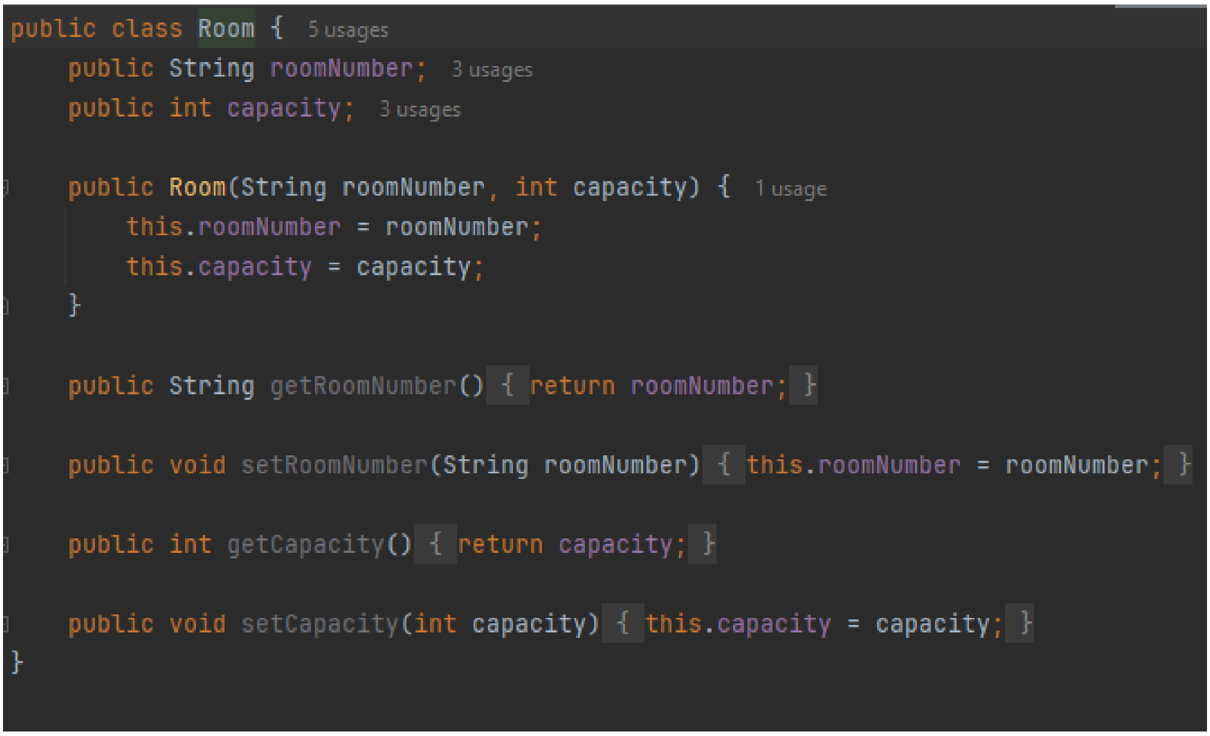
```
    public void setDepartureTime(String departureTime) { this.departureTime = departureTime; }

    public boolean bookSeat(String email) {  5 usages
        if (userService.isEligibleForFreeTrip(email, route)){
            this.seatTaken++;
            userService.resetTripCount(email, route); // Still track the trip
            return true;
        }
        if (this.seatTaken < this.capacity) {
            this.seatTaken++;
            userService.incrementTripCount(email, route); // Still track the trip

            return true;
        }
        return false;
    }

    public int getRemainingCapacity() { return this.capacity - this.seatTaken; }
}
```

Password Class



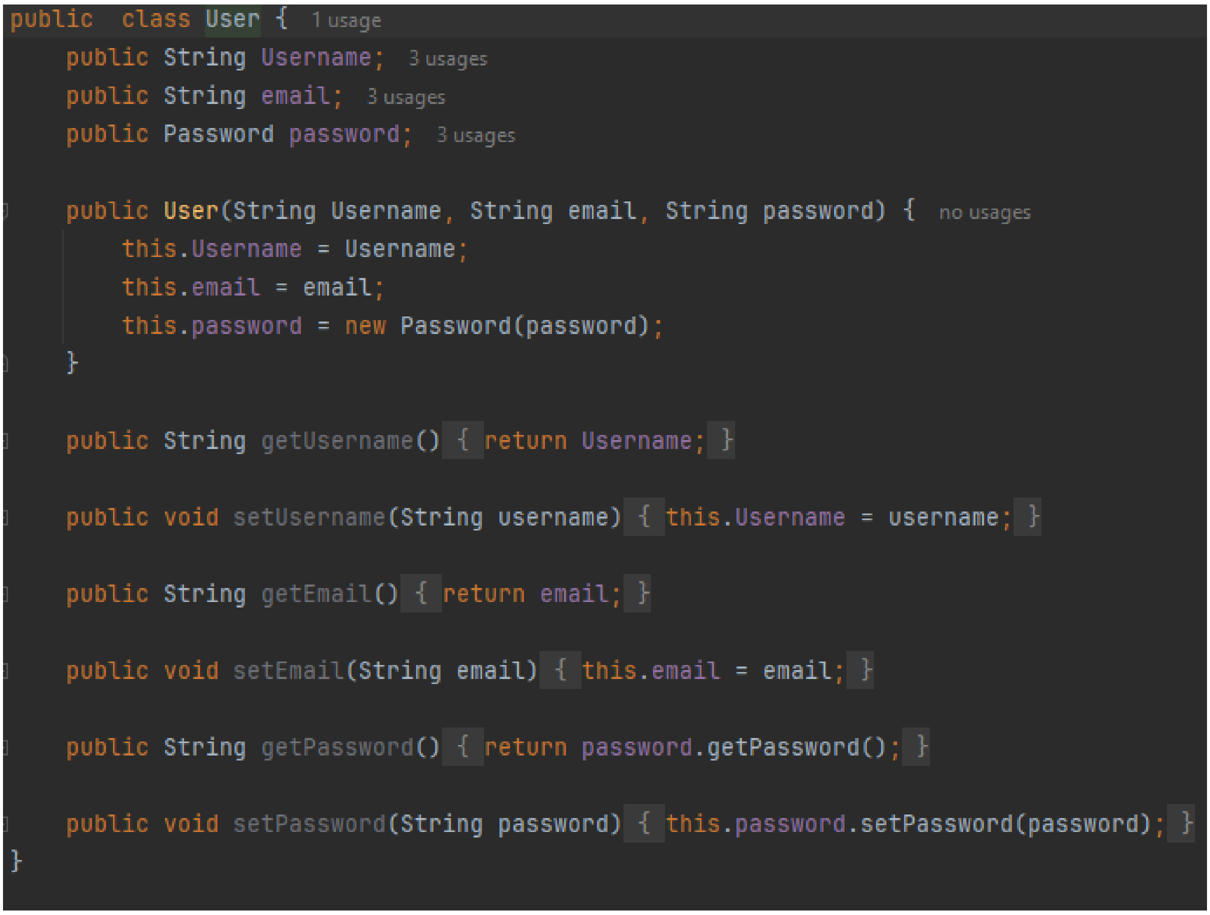
```
public class Password {  2 usages   new *
    private String password;  3 usages

    public Password(String password) { this.password = password; }

    public String getPassword() { return password; }

    public void setPassword(String password) { this.password = password; }
}
```

Room Class



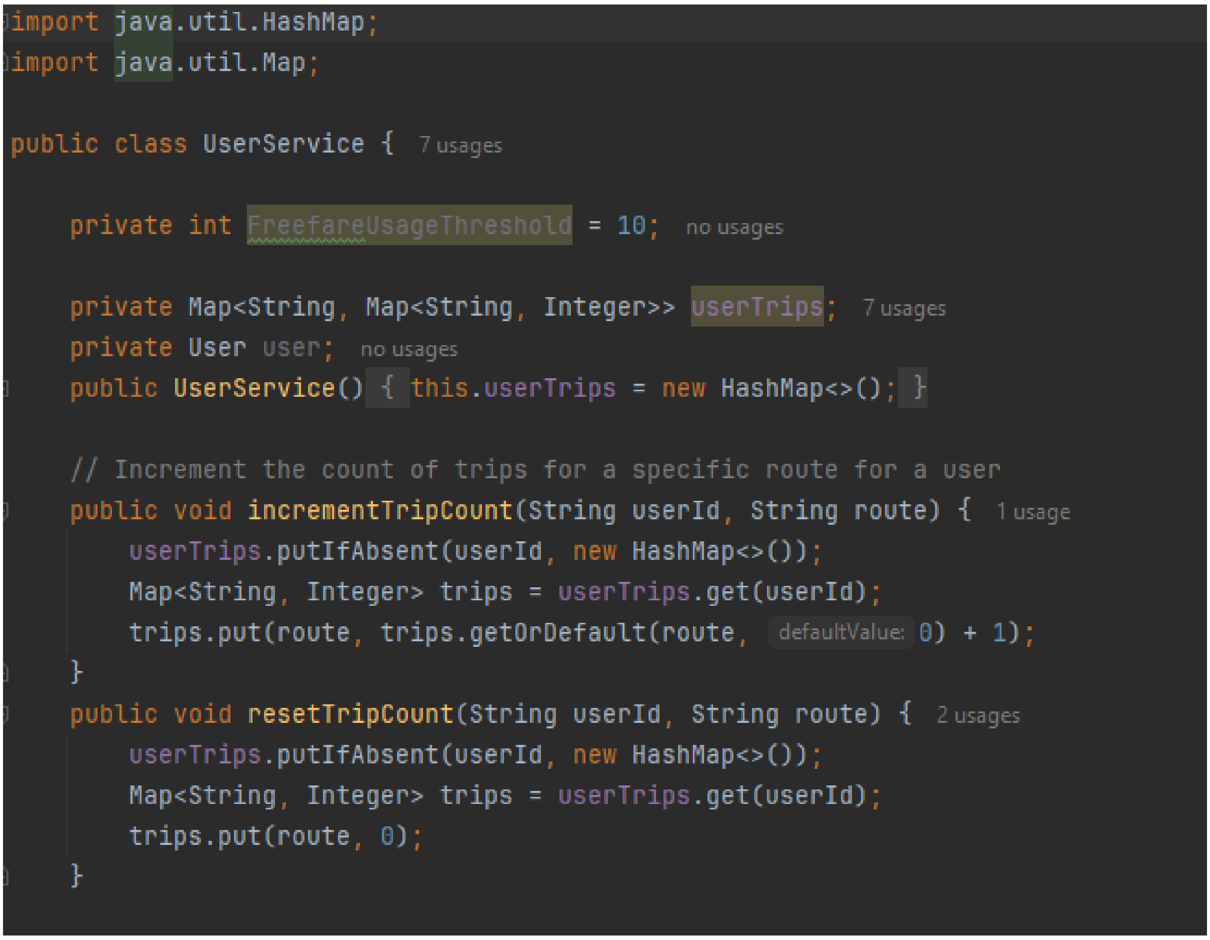
```
public class Room {  5 usages
    public String roomNumber;  3 usages
    public int capacity;  3 usages

    public Room(String roomNumber, int capacity) {  1 usage
        this.roomNumber = roomNumber;
        this.capacity = capacity;
    }

    public String getRoomNumber() { return roomNumber; }

    public void setRoomNumber(String roomNumber) { this.roomNumber = roomNumber; }

    public int getCapacity() { return capacity; }

    public void setCapacity(int capacity) { this.capacity = capacity; }
}
```

User Class

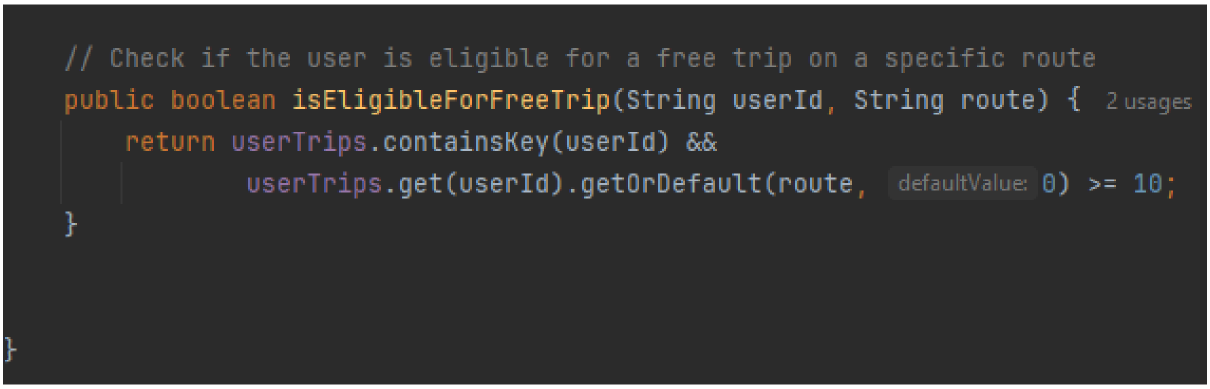


```
public  class User {  1 usage
    public String Username;  3 usages
    public String email;  3 usages
    public Password password;  3 usages

    public User(String Username, String email, String password) {  no usages
        this.Username = Username;
        this.email = email;
        this.password = new Password(password);
    }

    public String getUsername() { return Username; }

    public void setUsername(String username) { this.Username = username; }

    public String getEmail() { return email; }

    public void setEmail(String email) { this.email = email; }

    public String getPassword() { return password.getPassword(); }

    public void setPassword(String password) { this.password.setPassword(password); }
}
```

UserService Class



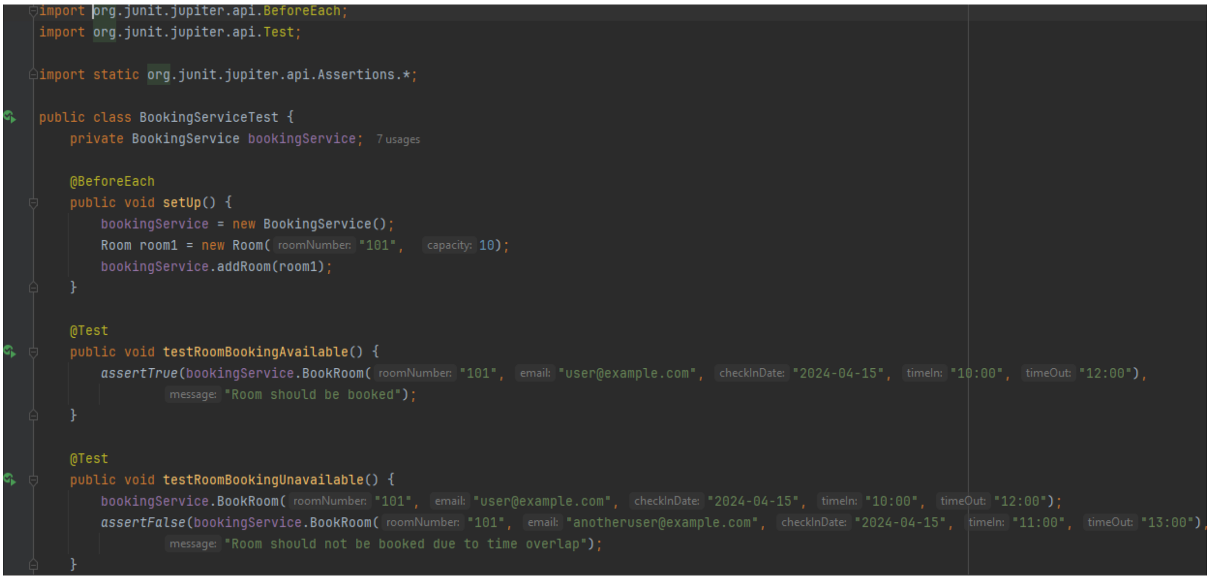
```
import java.util.HashMap;
import java.util.Map;

public class UserService {  7 usages

    private int FreefareUsageThreshold = 10;  no usages

    private Map<String, Map<String, Integer>> userTrips;  7 usages
    private User user;  no usages
    public UserService() { this.userTrips = new HashMap<>(); }

    // Increment the count of trips for a specific route for a user
    public void incrementTripCount(String userId, String route) {  1 usage
        userTrips.putIfAbsent(userId, new HashMap<>());
        Map<String, Integer> trips = userTrips.get(userId);
        trips.put(route, trips.getOrDefault(route,  defaultValue: 0) + 1);
    }
    public void resetTripCount(String userId, String route) {  2 usages
        userTrips.putIfAbsent(userId, new HashMap<>());
        Map<String, Integer> trips = userTrips.get(userId);
        trips.put(route, 0);
    }
```



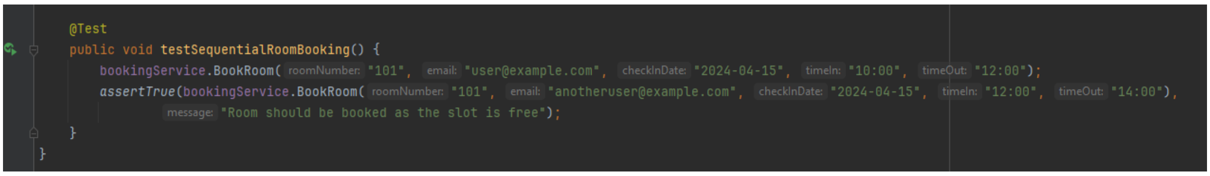
```
    // Check if the user is eligible for a free trip on a specific route
    public boolean isEligibleForFreeTrip(String userId, String route) {  2 usages
        return userTrips.containsKey(userId) &&
                userTrips.get(userId).getOrDefault(route,  defaultValue: 0) >= 10;
    }



}
```

<u>Tests</u>

BookingServiceTest

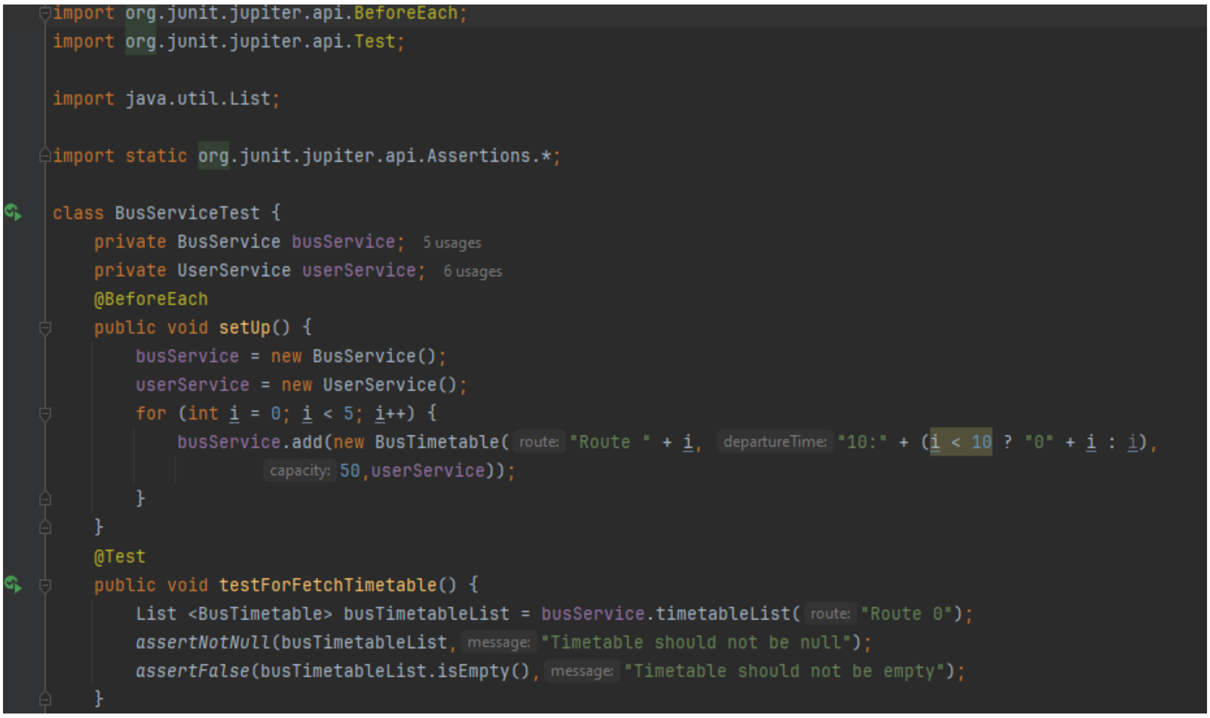


```java
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.*;

public class BookingServiceTest {
    private BookingService bookingService;

    @BeforeEach
    public void setUp() {
        bookingService = new BookingService();
        Room room1 = new Room( "101",   10);
        bookingService.addRoom(room1);
    }

    @Test
    public void testRoomBookingAvailable() {
        assertTrue(bookingService.BookRoom( "101",  "user@example.com",  "2024-04-15",  "10:00",  "12:00"),
                 "Room should be booked");
    }

    @Test
    public void testRoomBookingUnavailable() {
        bookingService.BookRoom( "101",  "user@example.com",  "2024-04-15",  "10:00",  "12:00");
        assertFalse(bookingService.BookRoom( "101",  "anotheruser@example.com",  "2024-04-15",  "11:00",  "13:00"),
                 "Room should not be booked due to time overlap");
    }
```

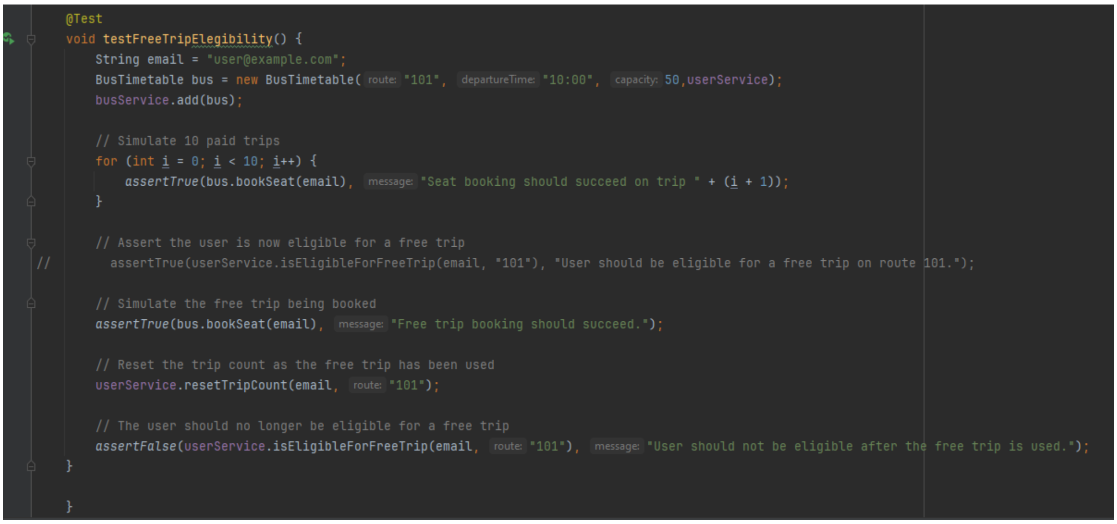


```java
    @Test
    public void testSequentialRoomBooking() {
        bookingService.BookRoom( "101",  "user@example.com",  "2024-04-15",  "10:00",  "12:00");
        assertTrue(bookingService.BookRoom( "101",  "anotheruser@example.com",  "2024-04-15",  "12:00",  "14:00"),
                 "Room should be booked as the slot is free");
    }
}
```

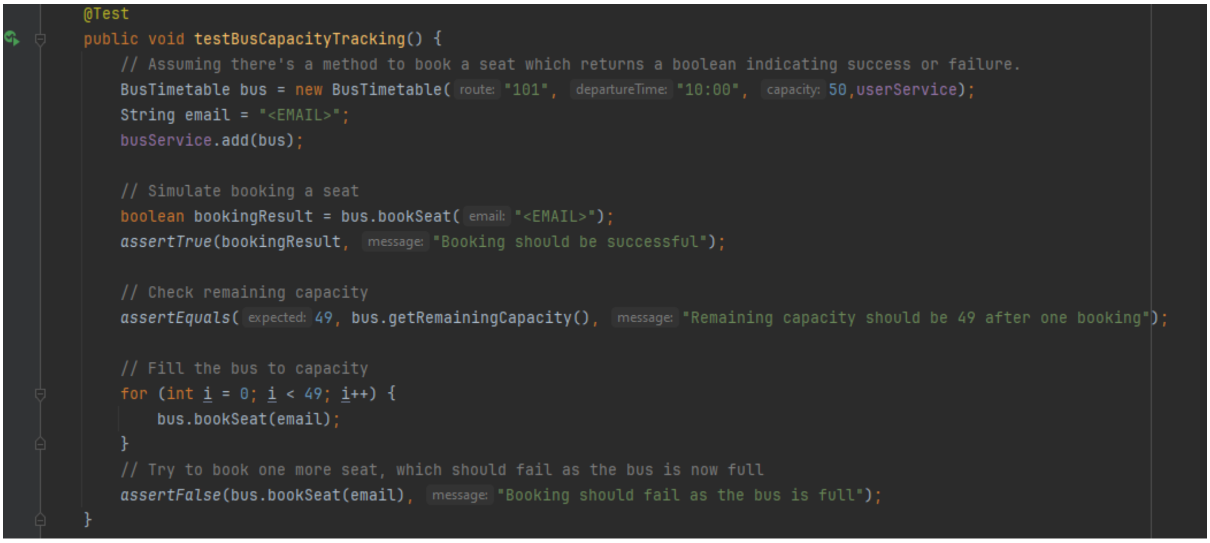
BusServiceTest



```java
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;

import java.util.List;

import static org.junit.jupiter.api.Assertions.*;

class BusServiceTest {
    private BusService busService;  5 usages
    private UserService userService;  6 usages
    @BeforeEach
    public void setUp() {
        busService = new BusService();
        userService = new UserService();
        for (int i = 0; i < 5; i++) {
            busService.add(new BusTimetable( route: "Route " + i,  departureTime: "10:" + (i < 10 ? "0" + i : i),
                    capacity: 50,userService));
        }
    }
    @Test
    public void testForFetchTimetable() {
        List <BusTimetable> busTimetableList = busService.timetableList( route: "Route 0");
        assertNotNull(busTimetableList, message: "Timetable should not be null");
        assertFalse(busTimetableList.isEmpty(), message: "Timetable should not be empty");
    }
```



```java
    @Test
    void testFreeTripElegibility() {
        String email = "user@example.com";
        BusTimetable bus = new BusTimetable( route: "101",  departureTime: "10:00",  capacity: 50,userService);
        busService.add(bus);

        // Simulate 10 paid trips
        for (int i = 0; i < 10; i++) {
            assertTrue(bus.bookSeat(email), message: "Seat booking should succeed on trip " + (i + 1));
        }

        // Assert the user is now eligible for a free trip
//          assertTrue(userService.isEligibleForFreeTrip(email, "101"), "User should be eligible for a free trip on route 101.");

        // Simulate the free trip being booked
        assertTrue(bus.bookSeat(email), message: "Free trip booking should succeed.");

        // Reset the trip count as the free trip has been used
        userService.resetTripCount(email, route: "101");

        // The user should no longer be eligible for a free trip
        assertFalse(userService.isEligibleForFreeTrip(email, route: "101"), message: "User should not be eligible after the free trip is used.");
    }

    }
```



```java
    @Test
    public void testBusCapacityTracking() {
        // Assuming there's a method to book a seat which returns a boolean indicating success or failure.
        BusTimetable bus = new BusTimetable( route: "101",  departureTime: "10:00",  capacity: 50,userService);
        String email = "<EMAIL>";
        busService.add(bus);

        // Simulate booking a seat
        boolean bookingResult = bus.bookSeat( email: "<EMAIL>");
        assertTrue(bookingResult, message: "Booking should be successful");

        // Check remaining capacity
        assertEquals( expected: 49, bus.getRemainingCapacity(), message: "Remaining capacity should be 49 after one booking");

        // Fill the bus to capacity
        for (int i = 0; i < 49; i++) {
            bus.bookSeat(email);
        }
        // Try to book one more seat, which should fail as the bus is now full
        assertFalse(bus.bookSeat(email), message: "Booking should fail as the bus is full");
    }
```