LONDON
METROPOLITAN
UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CS5002NI SOFTWARE ENGINEERING

Assessment Weightage & Type 35% Individual Coursework

Year and Semester 2020/21 Autumn

Student Name: Nabin Gurung

London Met ID: 19031160

College ID: np01cp4a190265

Assignment Due Date: 2 May 2021

Assignment Submission Date: 2 May2021

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.

Table of Contents

Table of Tables

Table of Figures

1. Introduction

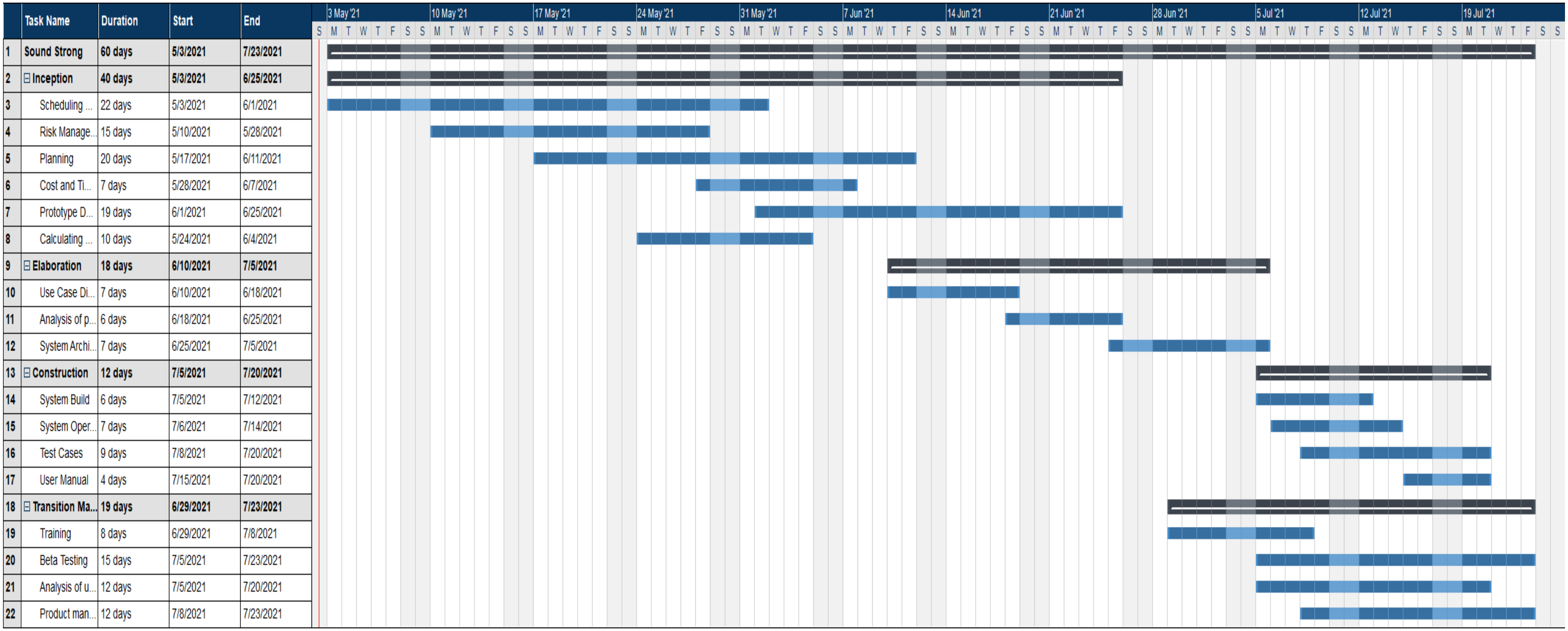
This is the second coursework of “Software Engineering” Module in 2nd semester. In this coursework we are assigned to develop a software building plan for company called “Sound Strong” using Unified Modelling Language(UML) to provide online computerized facility for the customer. This coursework carries 35% of overall module marks. The software needs to be fully functional. We are provided with the scenario of designing software by adding features like Registration, Booking etc. As we are developing the software on the Object-Oriented approach, we need to build use case, collaboration, and sequence diagram. This coursework helps us to learn about the preplanning, designing and quality of software.

An online system is to be developed to using RUP technique. RUP stands for Rational Unified Process which resembles the development cycle of software. It is divided into different phases i.e. modelling, analysis, design, implementation, testing and application. The Rational Unified Process explains how software development teams can successfully deploy commercially proven approaches to software development. They are considered "best practices" not because they are easy to follow, but because they are widely used in business by effective companies, rather than being able to accurately measure their worth. The Rational Unified Process provides each team member with the instructions, models, and tool advisors they need to fully use the following best practices, among others *(rational, 1998)*:

- Develop software iteratively
- Manage requirements
- Use component-based architectures
- Visually model software
- Verify software quality
- Control changes to software

2. Gantt Chart

A Gantt chart is a bar chart that shows how activities are scheduled over time. A Gantt chart is a handy way to indicate what work is planned to be completed on a given day that is used for organizing tasks of all sizes. It can also be used to display a project's start and end dates in a single graph. It is used to display activities or task against time in the form of chart. It is a project management tool which assist in planning and scheduling the complex project *(gantt, 2021)*.



	Task Name	Duration	Start	End
1	**Sound Strong**	**60 days**	**5/3/2021**	**7/23/2021**
2	**⊟ Inception**	**40 days**	**5/3/2021**	**6/25/2021**
3	Scheduling ...	22 days	5/3/2021	6/1/2021
4	Risk Manage...	15 days	5/10/2021	5/28/2021
5	Planning	20 days	5/17/2021	6/11/2021
6	Cost and Ti...	7 days	5/28/2021	6/7/2021
7	Prototype D...	19 days	6/1/2021	6/25/2021
8	Calculating ...	10 days	5/24/2021	6/4/2021
9	**⊟ Elaboration**	**18 days**	**6/10/2021**	**7/5/2021**
10	Use Case Di...	7 days	6/10/2021	6/18/2021
11	Analysis of p...	6 days	6/18/2021	6/25/2021
12	System Archi...	7 days	6/25/2021	7/5/2021
13	**⊟ Construction**	**12 days**	**7/5/2021**	**7/20/2021**
14	System Build	6 days	7/5/2021	7/12/2021
15	System Oper...	7 days	7/6/2021	7/14/2021
16	Test Cases	9 days	7/8/2021	7/20/2021
17	User Manual	4 days	7/15/2021	7/20/2021
18	**⊟ Transition Ma...**	**19 days**	**6/29/2021**	**7/23/2021**
19	Training	8 days	6/29/2021	7/8/2021
20	Beta Testing	15 days	7/5/2021	7/23/2021
21	Analysis of u...	12 days	7/5/2021	7/20/2021
22	Product man...	12 days	7/8/2021	7/23/2021

3 May '21
10 May '21
17 May '21
24 May '21
31 May '21
7 Jun '21
14 Jun '21
21 Jun '21
28 Jun '21
5 Jul '21
12 Jul '21
19 Jul '21
S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S

Figure 1: Gantt Chart

The above Gantt chart is about the Sound Strong Online Application. The estimated time to build a complete software is 5 to 5.5 month. The inception phase is estimated to be completed in 70 days, 32 days for elaboration phase, 26 days for construction phase and 19 days for transition phase.

3. Use Case

A use case is a list of all the different ways a consumer needs to interact with a device. These "uses" are like device requests, and use cases explain how the system responds to such requests. Usage cases, in other words, define the interaction between a device and its users, often known as actors. Use cases may refer to facilities, computers, or business processes, even though the system is normally automated (such as an Order system) *(Larson Elizabeth, 2004)*.

- **Actor**: An Actor in a use case plays vital role. Actor is someone who interacts with system. He/She interacts with every use cases of the system by providing required input. Actor can be divided into two parts Primary and Secondary. Primary actors are those who without which use cases cannot be activated like Customer, whereas Secondary actors are those who depends on primary actors.
- **System**: System is the whole application/software developed. The system carries all the use cases, actors, relations, and data. System is denoted by a rectangular box containing multiple types of use cases.
- **Use Case**: Use case also known as functions/method is a key element in use case diagram. Use case describes about the features to be added in application. It represents about the function of system without which system is useless. It is denoted with oval shape in use case diagram.
- **Relationship:** The important element in use case diagram is relation. A system contains multiple use case to which multiple actors are connected. The relationship between use case and actor determines the working mechanism of system. A use case can have multiple actors associated. Relationship can be divided into 2 parts i.e. Included and Extended. When a use calls another use case automatically it is included relationship between use cases, whereas if there is a condition to call another use case then it is Extended Relationship. It is denoted by dotted line.

3.1. Use Case Diagram

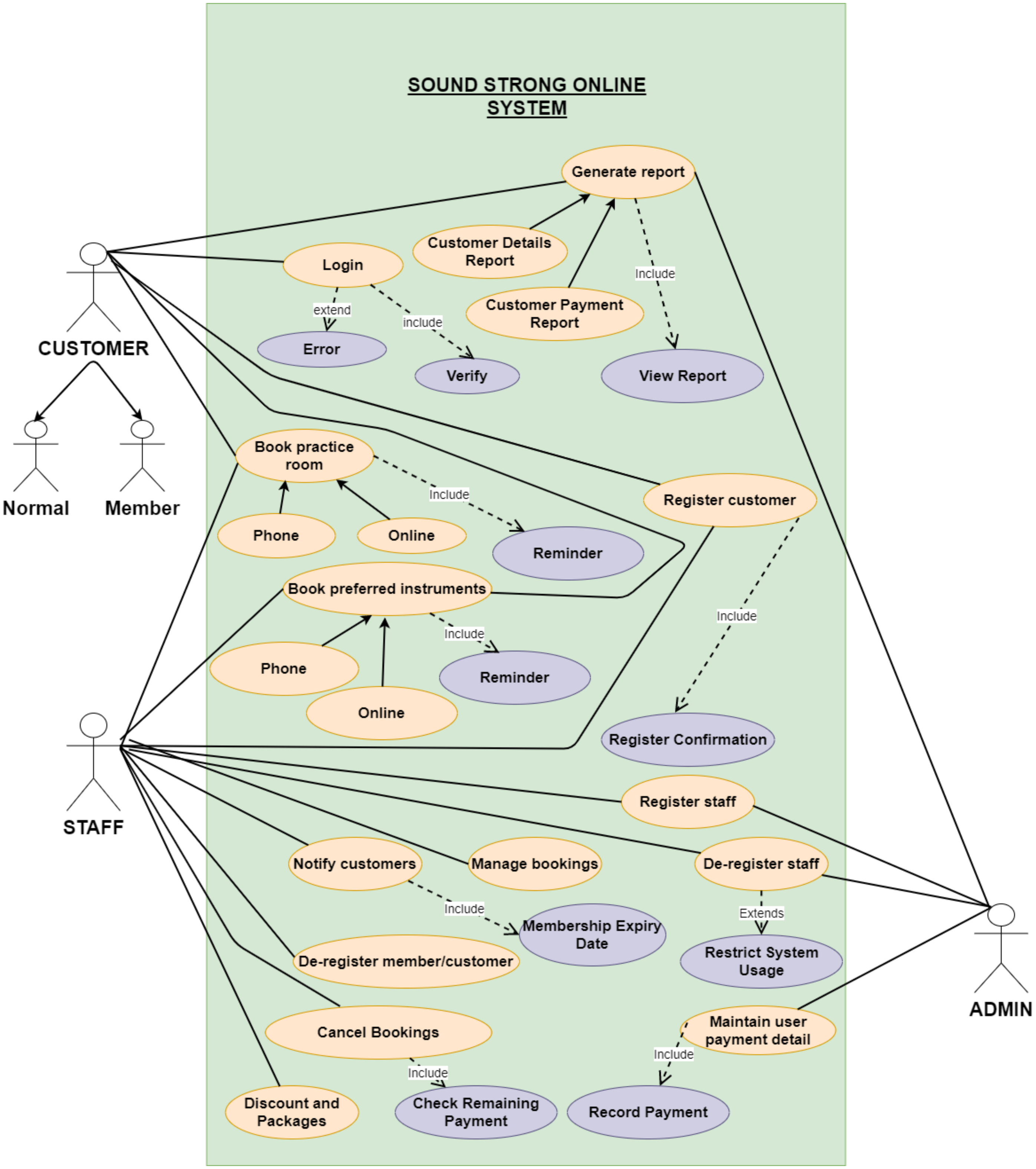


Figure 2:Use Case Diagram

4. High Level Use Case Description

4.1. For Register Customer

Name: Register customer

Actors: Customer, Staff

Description: A Customer visits Sound Strong Online System. He/She checks availability of rooms, instruments, and time. Customer enters his/her personal details, upload citizenship number, photo and register him/herself. After completing registration process successfully, he/she enters the system and make necessary payments. Customer receive notification of payment on his/her mobile number. Staff can also register customer with his/her document.

4.2. For Register Staff

Name: Register staff

Actors: Staff, Admin

Description: A Staff him/herself visits Sound Strong Online System. Staff enters his/her personal details, upload citizenship number, photo and register him/herself. After completing registration process successfully, he/she enters the system. An Administrator can also register customer with his/her document.

4.3. For De-Register Member/Customer

Name: De-register member/customer

Actors: Staff

Description: The staff de-registers customers, if customers do not want to be the member or do not want to use the service provided by the company.

4.4. For De-Register Staff

Name: De-register staff

Actors: Staff, Admin

Description: The admin de-registers staff who leaves the job before the expiration of their contract and restricts them from using the system. The staff him/herself can de-register before quitting the job by using the system.

4.5. For Book Practice Room

Name: Book practice room

Actors: Customer, Staff

Description: The customer is able to make booking through the use of the system if they are registered customer/member or they can book practice room through the use of phone by directly calling in company as companies number is provided at the bottom of the app. Staff can also book the room for customer.

4.6. For Maintain User Payment Detail

Name: Maintain user payment detail

Actors: Admin

Description: An Admin enters the system. He/She checks the payments made by customers. Admin records all the histories of payment done by customer. Admin uploads payment record in system.

4.7. For Generate Report

Name: Generate Report

Actors: Admin, Customer

Description: The registered customer can generate and get their reports from the system. They can view their own report only. To view the report phone number and password is required. The admin can also generate any customer payment and detail report to get data of the customer.

4.8. For Notify Customers

Name: Notify Customers

Actors: Staff

Description: The staff sends message if the customers membership date is about to expire. The staff also set reminder for the bookings done by the customers. The system also sends automatic message if the customer does not visit the company at booking time.

4.9. For Manage Bookings

Name: Manage Bookings

Actors: Staff

Description: The staff checks the bookings done by customers. He/She record the booking dates and instruments in the system. He/She make sure that rooms and instruments available at the booking time and date.

4.10. For Book Instruments

Name: Book Instruments

Actors: Customer, Staff

Description: The customer is able to make booking through the use of the system if they are registered customer/member or they can book instruments through the use of phone by directly calling in company as companies number is provided at the bottom of the app.

Staff can also book the instruments selected by customers based on phone call.

5. Expanded Level Use Case Diagram

5.1. Expanded level description for “register customer”

Use Case Name: Register Customer

Purpose: To register customer

Description: A Customer visits Sound Strong Online System. He/She checks availability of rooms, instruments, and time. Customer enters his/her personal details, upload citizenship number, photo and register him/herself. After completing registration process successfully, he/she enters the system and make necessary payments. Customer receive notification of payment on his/her mobile number. Staff can also register customer with his/her document.

Primary Actors: Staff, Customer

Secondary Actors: None

Pre-Condition: Customer needs to provide valid data.

Post Condition: None

Main Flow/ Typical Course of Events	
Actor Action	**System Response**
1.Customer enters his/her required personal and required information in the system and submits.	
2.Staff fills the required information of customer in the system based on data the he/she has provided.	

	3.The system validates, verifies, and register the customer. 4.The system records the data of customer. 5.The system displays the dashboard for customer. 6.The use case ends.

Table 1: Table of Action step for registration of customer

Alternative Flow

Actor Action	System Response
3.a. Missing Full/Last Name and/or email.	1. The system prompts for Name and email. 2. Use case resume at the main flow step 1.
3.b. Entered numbers/symbols in Name and/or email.	1. The system prompts for Valid Name and email. 2. Use case resume at the main flow step 1.
3.c. Entered already registered email.	1.The system displays “This email has been already registered. Please sign up with new email or just login” message. 2. Use case resume at the main flow step 1.

Table 2:Table of Alternative Flow for registration of customer

5.2. Expanded level description for “book practice room”

Use Case Name: Book Practice Room

Purpose: To book practice room.

Description: The customer is able to make booking through the use of the system if they are registered customer/member or they can book practice room through the use of phone by directly calling in company as companies number is provided at the bottom of the app. Staff can also book the room for customer.

Primary Actors: Staff, Customer

Secondary Actors: None

Pre-Condition: Customer needs to have login in the system.

Post Condition: None

Main Flow/ Typical Course of Events	
Actor Action	**System Response**
1.Customer enters his/her ID and password.	
2.Customer submits hi/her ID and password and login.	
3.Customer enters his/her dashboard.	
4.Customer enters booking time and room.	
	5.The system validates, verifies, and login the customer. 6.The system displays the dashboard for customer. 7.The use case ends.

Table 3: Table of Action step for booking practice room

Alternative Flow

Actor Action	System Response
4.a. Missing Booking Time/Room.	1. The system prompts for Booking Time and Room. 2. Use case resume at the main flow step 1.
4.b. Entered time that is not available.	1. The system prompts for entering another booking time. 2. Use case resume at the main flow step 1.
4.c. Entered room that is not available or already booked by other.	1. The system prompts for entering another room. 2. Use case resume at the main flow step 1.
4.d. Re booking same room and time.	1. The system displays “You have already booked a room at this time“ message. 2. The use case ends.

Table 4: Table of Alternative Flow for booking of practice

5.3. Collaboration and sequence diagram for “register customer”

5.3.1. Collaboration Diagram

Collaboration Diagram is a diagrammatical representation of message flow between the objects. This diagram is used to show the interaction way of use cases. It is used to clarify and define the roles of objects. It delivers the functionalities of use case and operations *(visual-paradigm, 2020)*.In order to develop a collaboration diagram of extended use case, I will take following steps to build it.

- **Step 1:** We need to identify the domain classes in from use case in first step. The domain class can be an actor or object. Here, System, Customer, Register and Staff are domain classes.

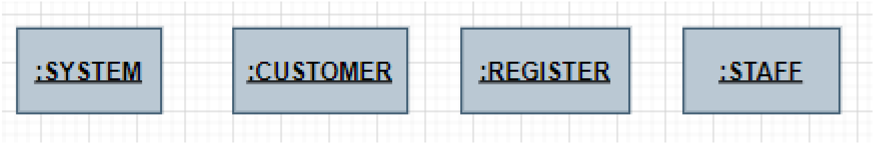


Figure 3: Screenshot of domain classes

- **Step 2:** In second step, control object, boundary object and actor are added as RegistrationDetailUI, RegistrationDetail, User respectively.

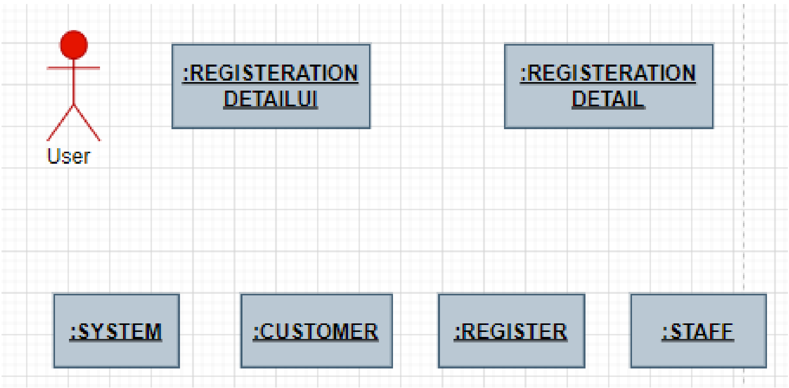


Figure 4: Screenshot of domain classes, actors and control objects

- **Step 3:** In third step, connection is done between the objects. The actors is connected to control object, object is connected to boundary object and at last domain classes are connected to boundary object.

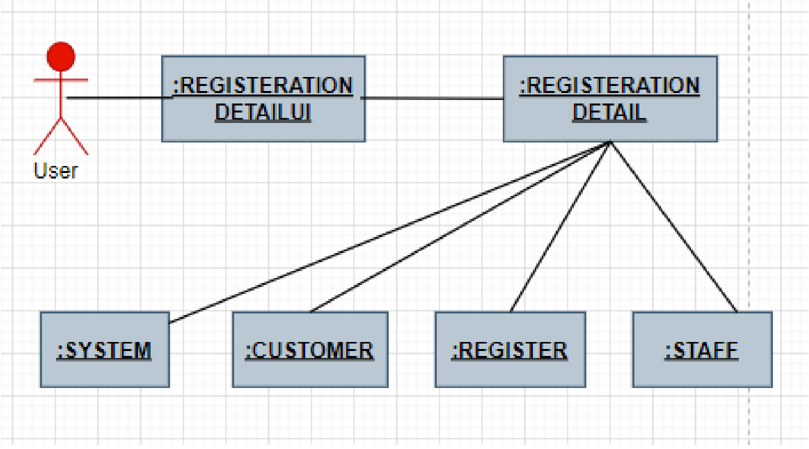


Figure 5: Screenshot of connecting objects,actors,classes

- **Step 4:** In last step, message flow is shown between the objects using arrow headed line with proper message.

5.3.2. Final Collaboration Diagram

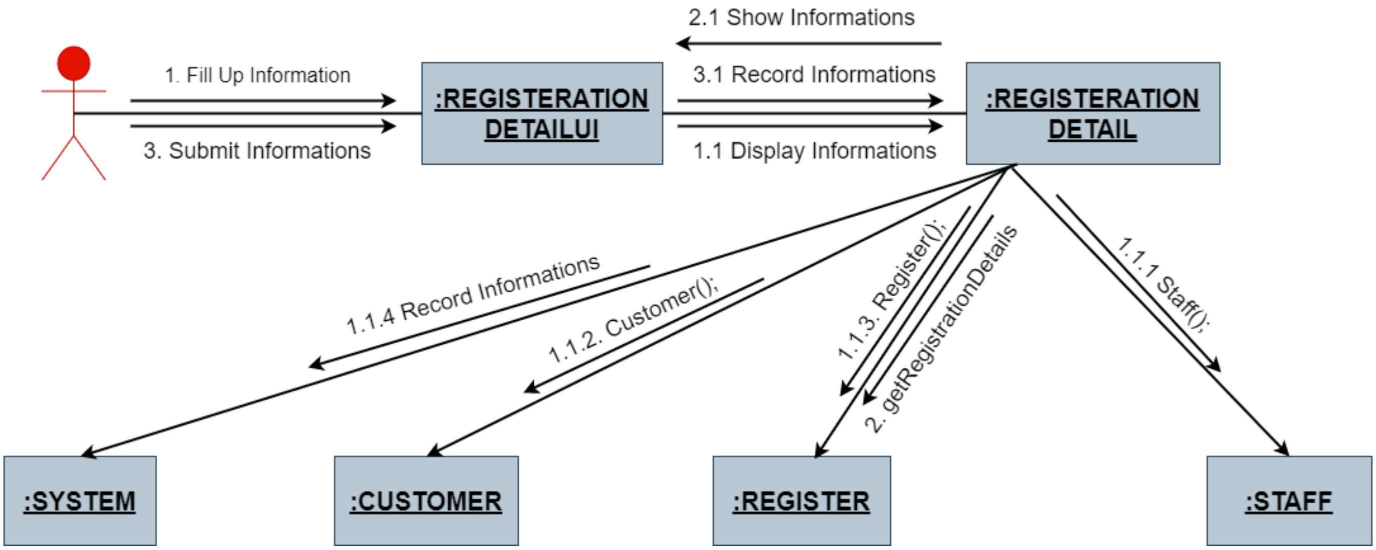


Figure 6: Collaboration Diagram

5.3.3. Sequence Diagram

A sequence diagram clearly defines the order in which objects interact, or the order in which these interactions occur. A sequence diagram may also be referred to as an event diagram or an event scenario. Sequence diagrams show how and in what order the components of a machine work together. Businesspeople and software engineers also use these diagrams to log and understand specifications for current and existing applications *(geeksforgeeks, 2018)*. To develop a collaboration diagram of extended use case, I will take following steps to build it.

The following steps are taken to make sequence diagram:

- **Finding Domain Class**

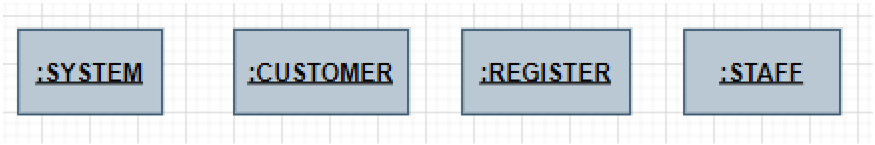


Figure 7: Screenshot of domain classes

- **Control Object Lifeline**

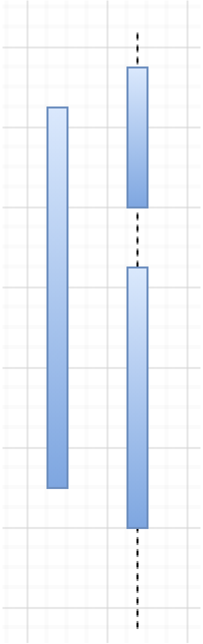


Figure 8: Screenshot of control object

- **Boundary object lifeline**

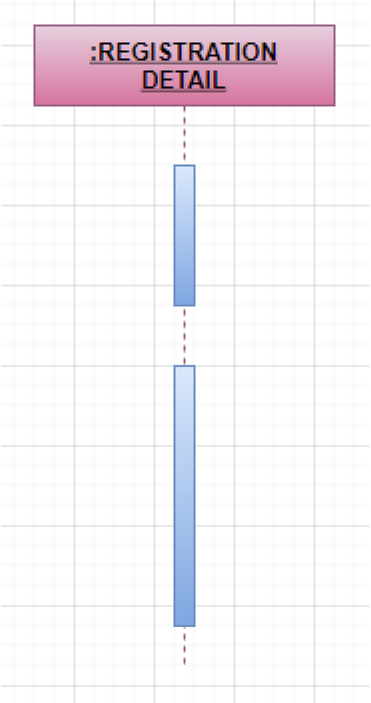


Figure 9: Screenshot of Boundary object lifeline

- **Actor lifeline**

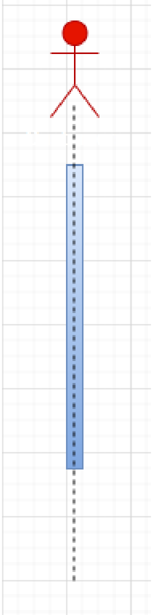


Figure 10: Screenshot of Actor lifeline

5.3.4. Final Sequence Diagram

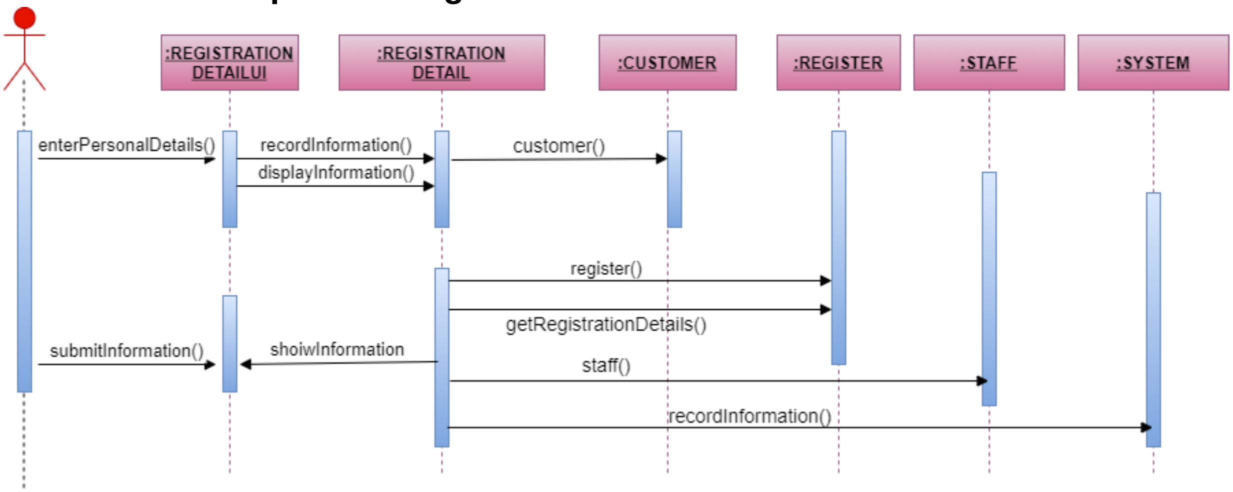


Figure 11: Sequence Diagram

Here, in the sequence diagram the customer enters personal detail in registration UI. After entering all the necessary details, the customer clicks submit button in UI. The details entered by the customer will be recorded in registration function, the overall information about the registration detail will be shown to customer.

6. Class Diagram

A class diagram is a static diagram. It reflects an application's static view. A class diagram illustrates a class's properties and operations, as well as the system's constraints. Since class diagrams are the only UML diagrams that can be specifically mapped for object-oriented languages, they are commonly used in the modeling of object-oriented structures. A set of classes, interfaces, associations, collaborations, and constraints. Are seen in a class diagram. A structural diagram is another name for it.

Following steps are taken to build a class diagram from use case.

- **Identifying domain classes:** Domain classes are identified from the use case diagram.

Use cases	Domain classes
Login	User
Book Practice Room	Customer, Booking, Staff
Book Instrument	Customer, Booking, Staff
Notify Customer	User, Admin
Generate report	Customer, Admin
Register customer	Customer, Staff
Manage User Payment Detail	Admin
Manage Booking	Staff
Register staff	Customer, Staff
Customer Detail Report	Staff, Customer
Customer Report	Customer, Staff
De-register staff	Admin, Staff
De-register customer	Customer, Staff
Cancel Booking	Customer, Staff

Table 5: Table of Use case and domain classes

- **Creation of Class:** Each domain classes are properly added in diagram

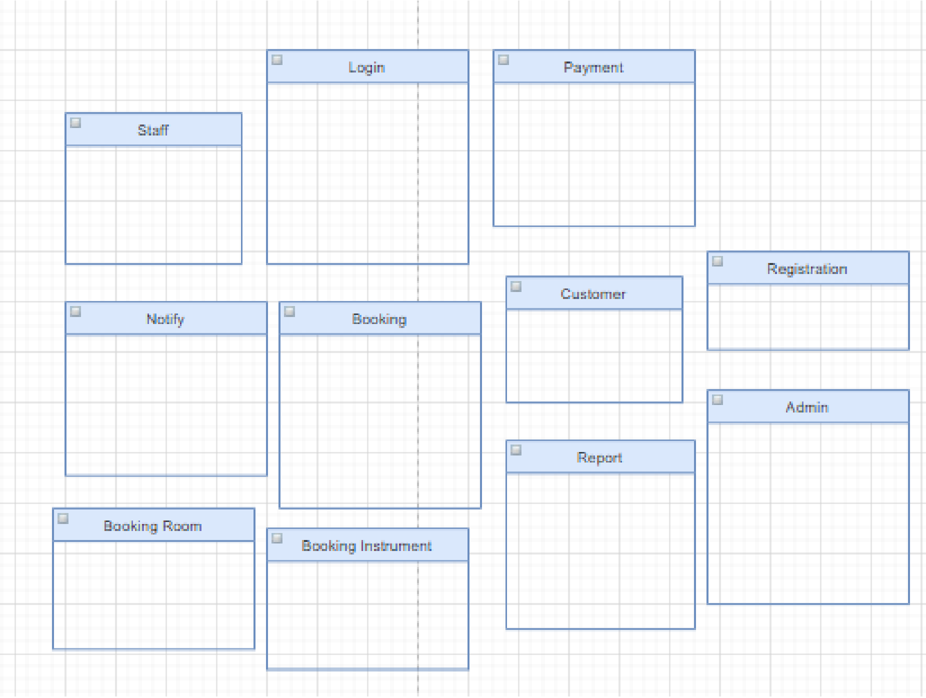


Figure 12: Screenshot of domain classes for class diagram

- **Method Creation:** Each method is added in respective classes with proper modifier

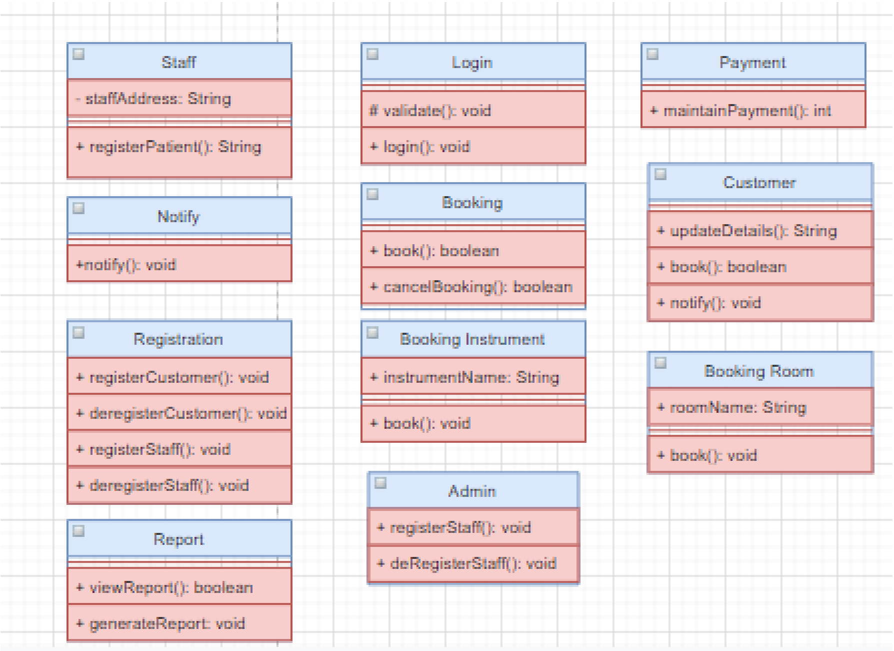


Figure 13: Screenshot of method creation in class diagram

- **Attribute Creation:** The attributes/ data that a class can hold are properly added in diagram respectively

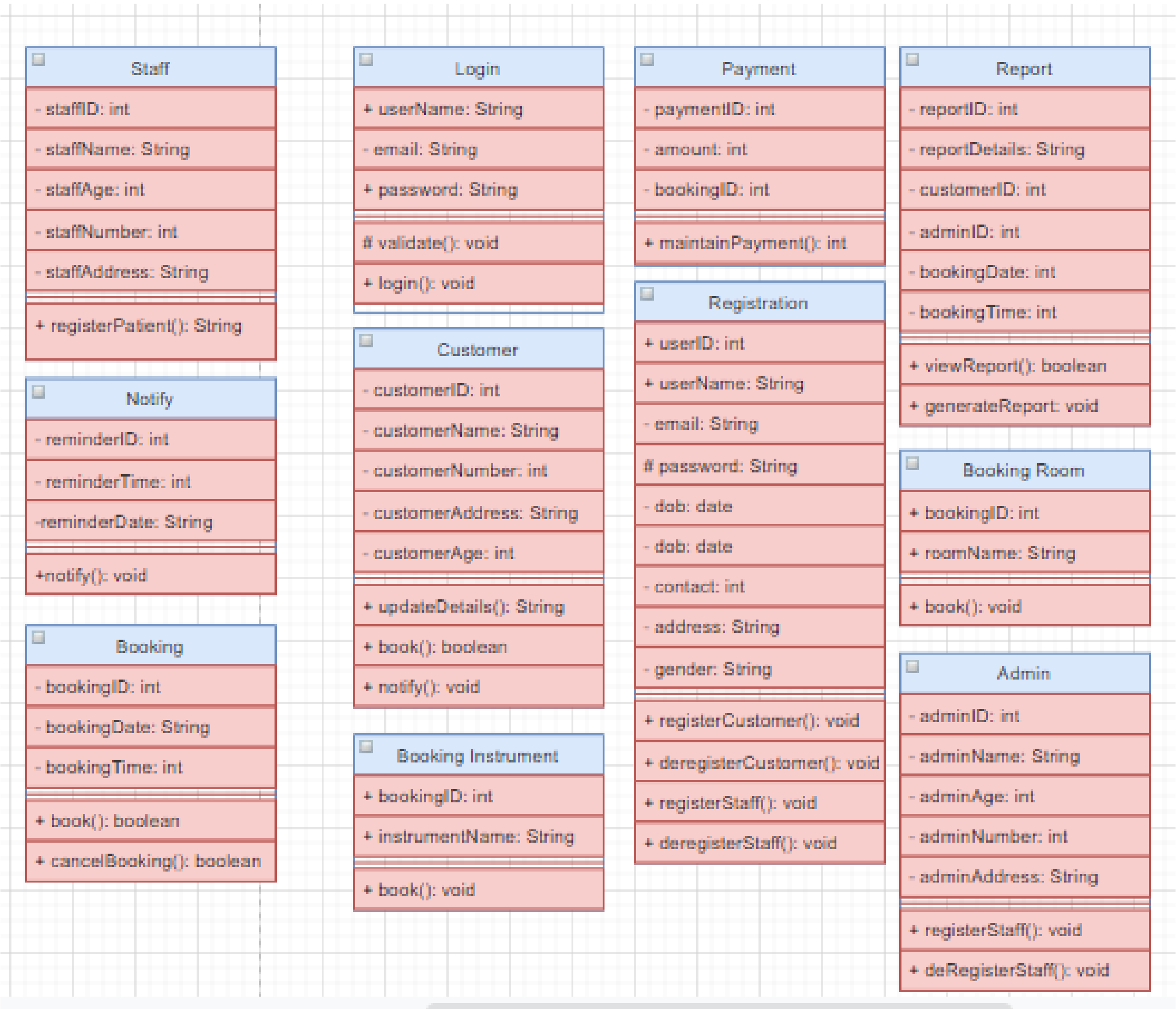


Figure 14: Screenshot of attributes creation in class diagram

- **Relationship of classes:** Every classes are connected using Association and proper dependencies.

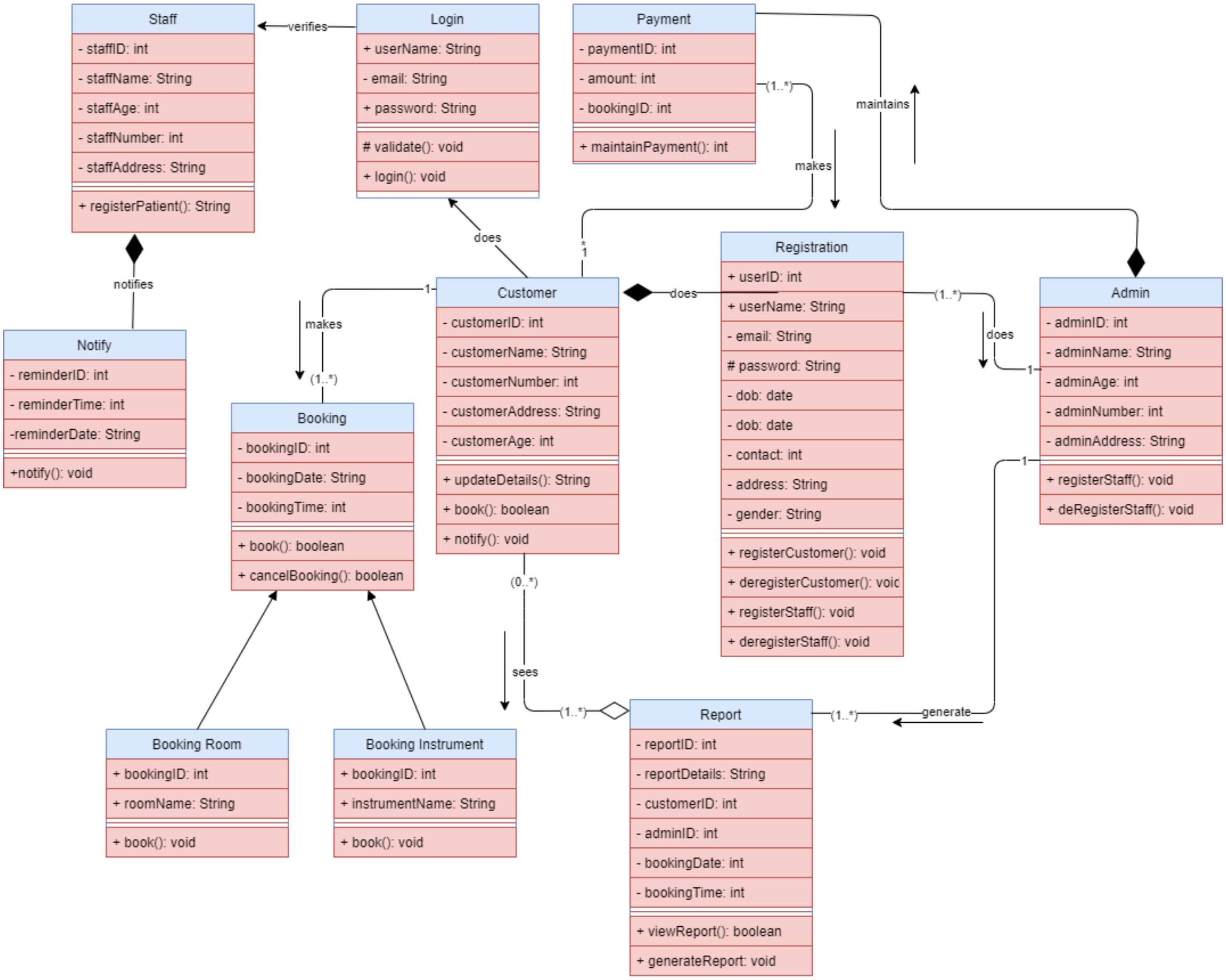


Figure 15: Screenshot of relation in classes

6.2. Final Class Diagram

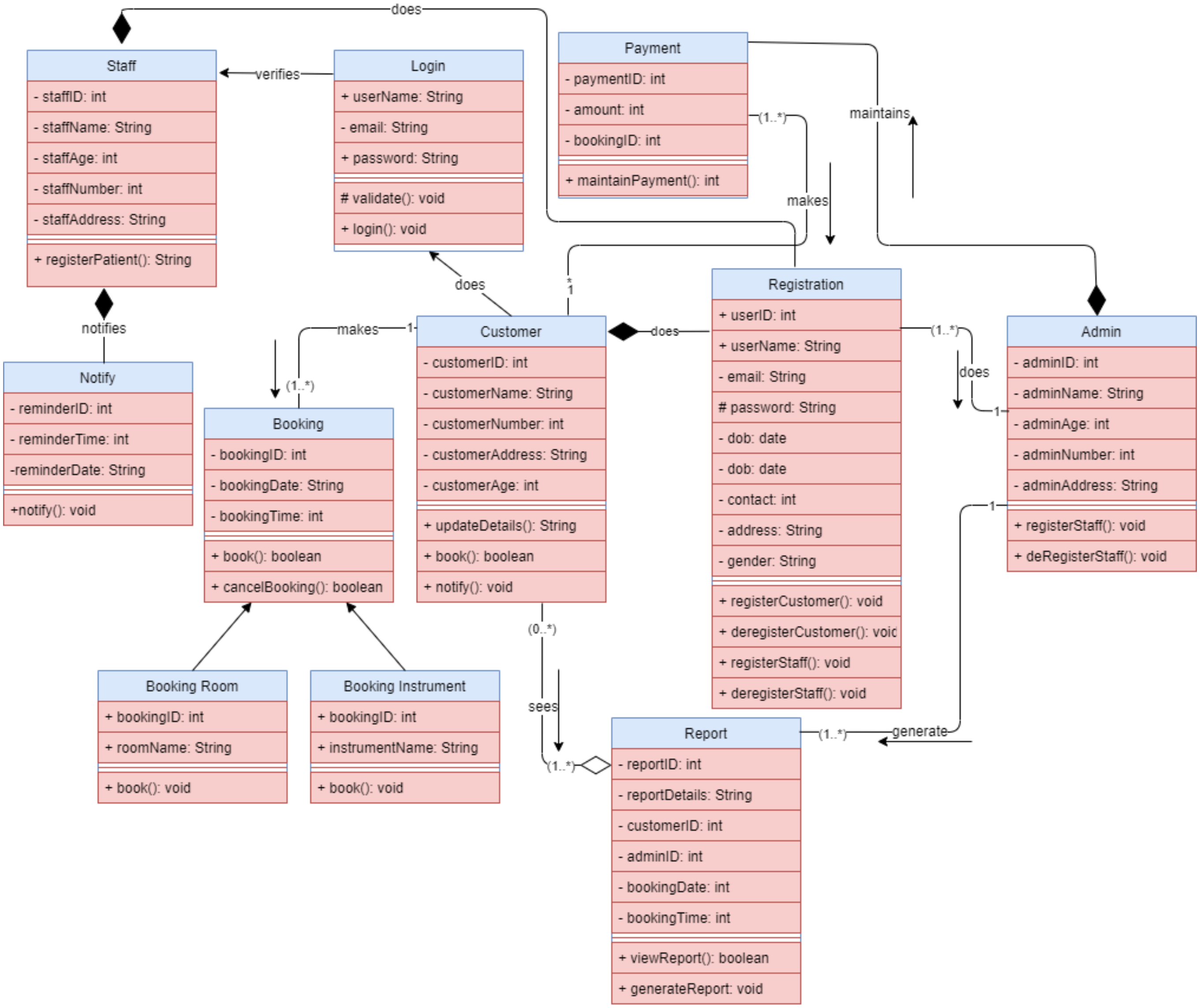


Figure 16: Class Diagram

7. Further Development Process

Coursework requirement is only to model the system diagrammatically. But, it’s not enough for the fully functional system to be build. The project is not a huge, but it can cost a lot of time, money, and employee. From the scenario, that we are given we build different diagrams such as use case diagram, collaboration diagram, sequence diagram and most important Gantt chart of project.

As we have used RUP development method to develop a system for Sound Strong Company. It helps deter money from being lost and avoids unforeseen construction costs because it has a clear schedule for each phase of the development process. With the help of this coursework we designed and planned to build the system programmatically. As we are in the midway of project i.e. elaboration phase, the task is not complete yet because many new requirements may rise during the process. Since this project was developed using an iterative approach, the various phases may run in parallel, but their initialization dates will vary because any information from the previous step is needed to begin the next phase. Information from one step is required to begin the next, and things become simpler as the project progresses. In all stages of the project, we are willing to make modifications and adjustments in accordance with their changing client requirements. And after the activities are completed, we will help and manage the system.

After completing design further planning’s are made to develop system. Documentation of system is carried out to properly manage the project. Every step we have done, every function, features and their uses are documented properly to make record and submit to client. After that, coding part of the project is carried out. Firstly, prototype of application as in Beta version is given to client for the feedback. Client will check the application and will suggest different changes to be implemented. Finally, application will be ready to release and launched, but transition phase will remain even after completing because different bugs may appear so, to remove bug the testing will be running all the time.

8. Prototype for Sound Strong Online Application

Login

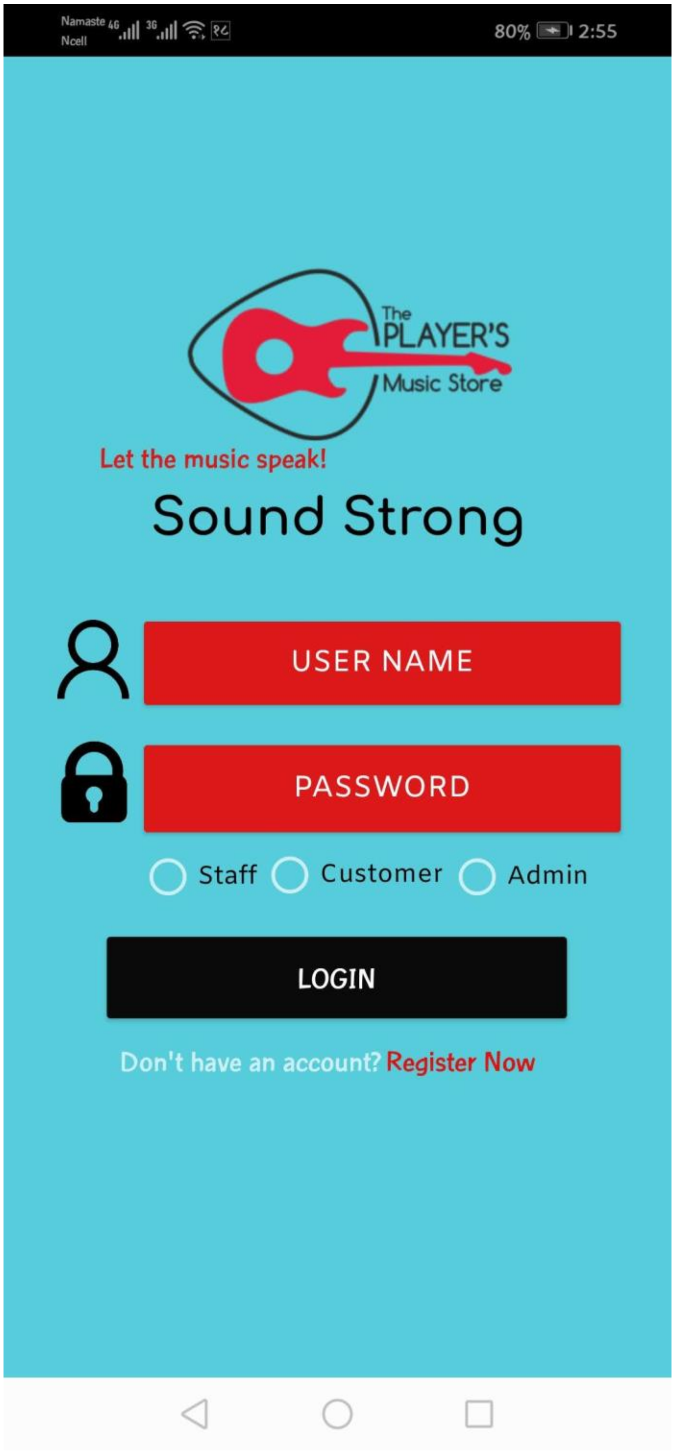
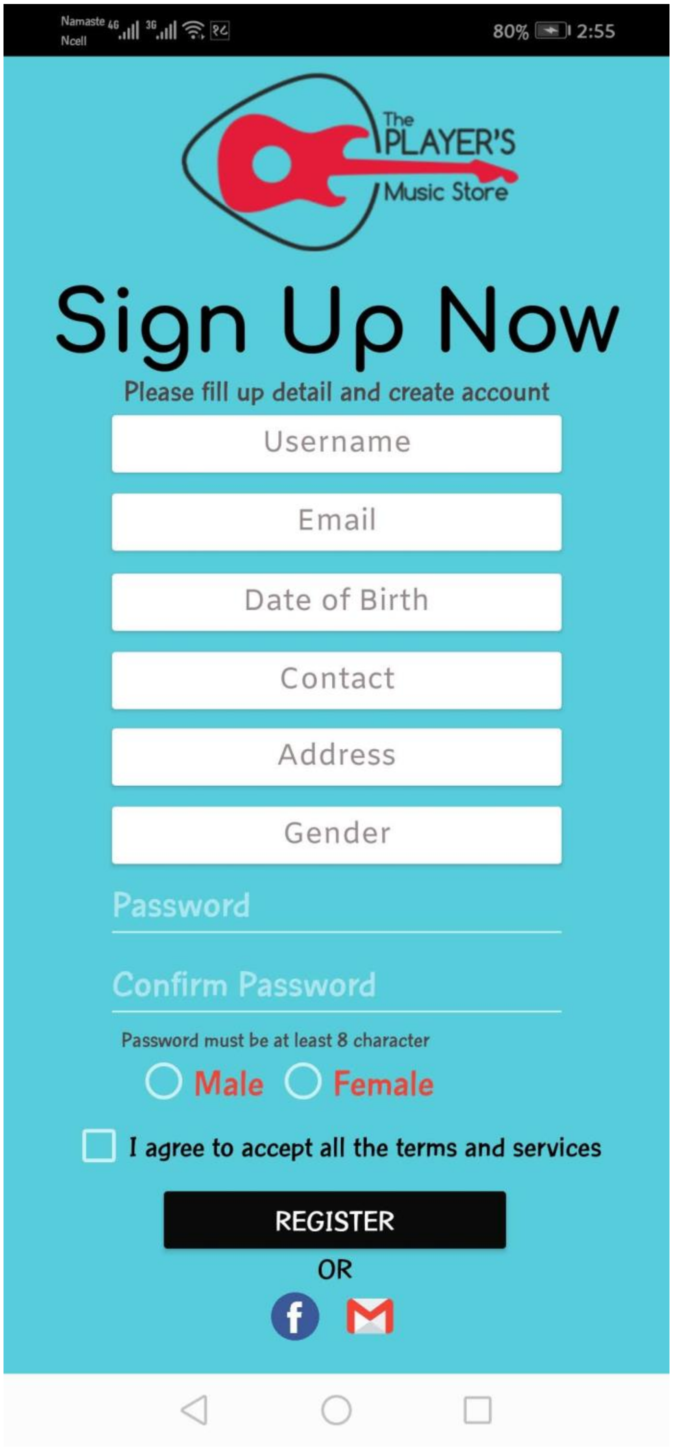


Figure 17: Screenshot of GUI of login screen of prototype app

Register, Customer, Admin



Namaste 4G 3G
Ncell
80% 2:55

The PLAYER'S Music Store

Sign Up Now

Please fill up detail and create account

Username

Email

Date of Birth

Contact

Address

Gender

Password

Confirm Password

Password must be at least 8 character

○ Male ○ Female

□ I agree to accept all the terms and services

REGISTER

OR

Figure 18: Screenshot of GUI of registration screen of prototype app

Generate Report

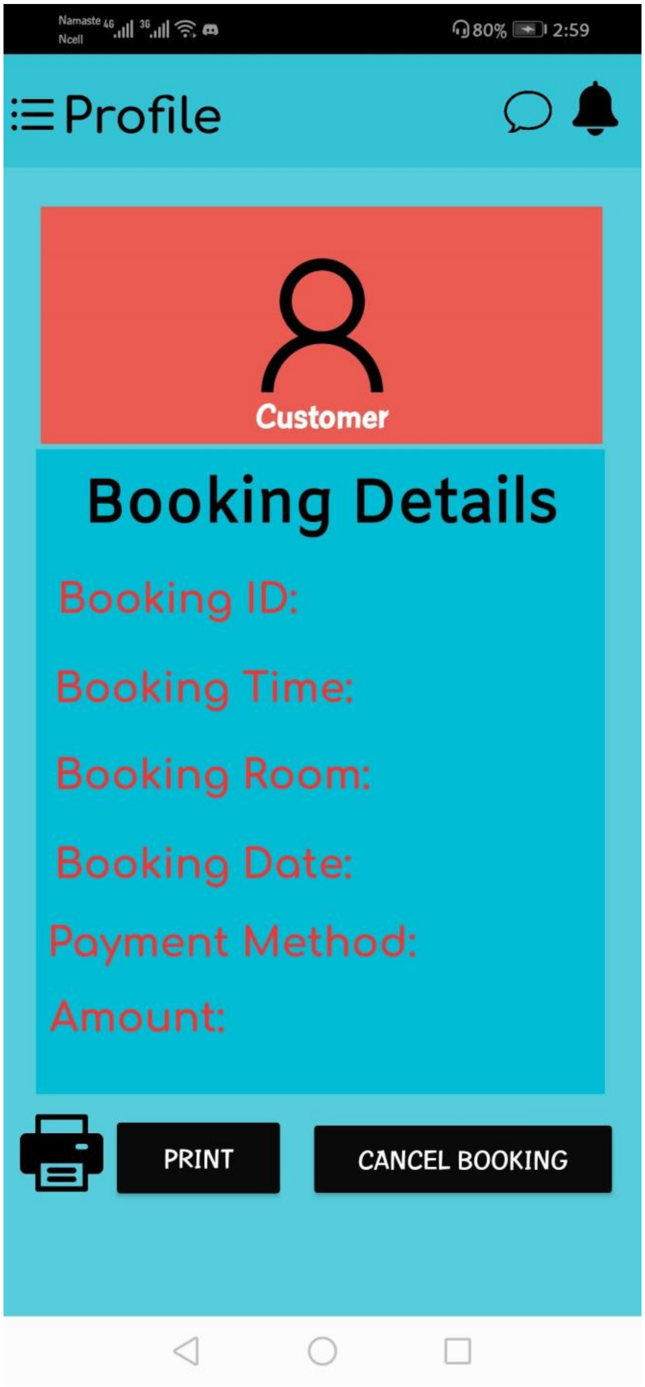


Figure 19: Screenshot of GUI of report screen of prototype app

Booking

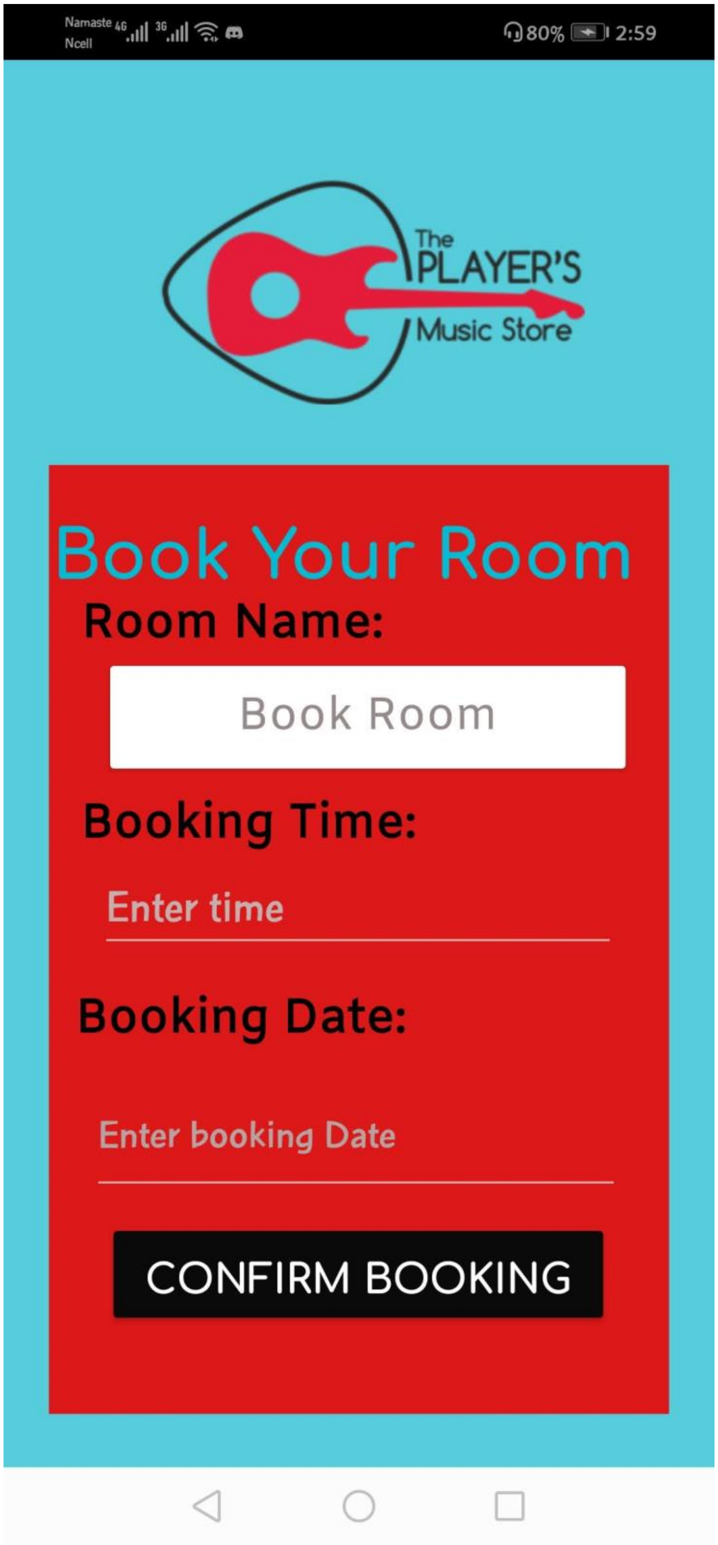


Figure 20: Screenshot of GUI of booking screen of prototype app

Payment

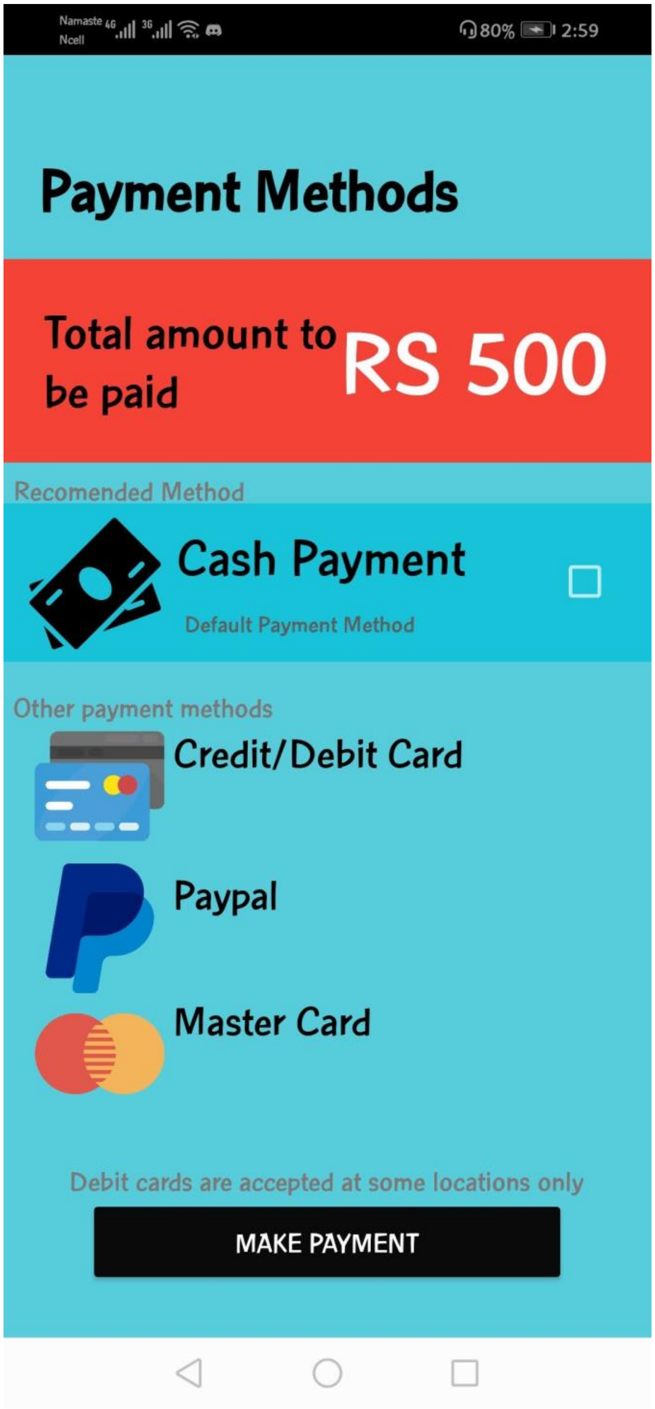


Figure 21: Screenshot of GUI of payment screen of prototype app

Dashboard Customer

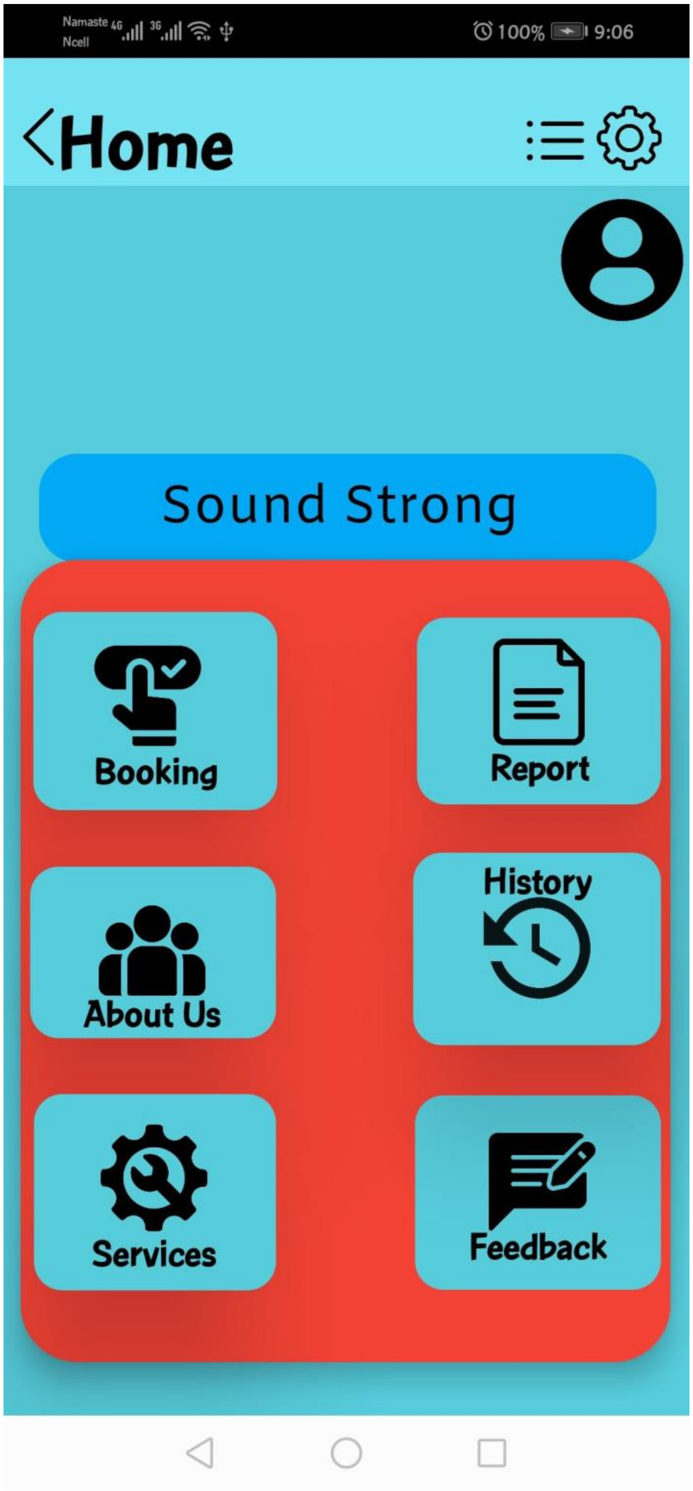


Figure 22: Screenshot of GUI of customer dashboard screen of prototype app

Dashboard Staff

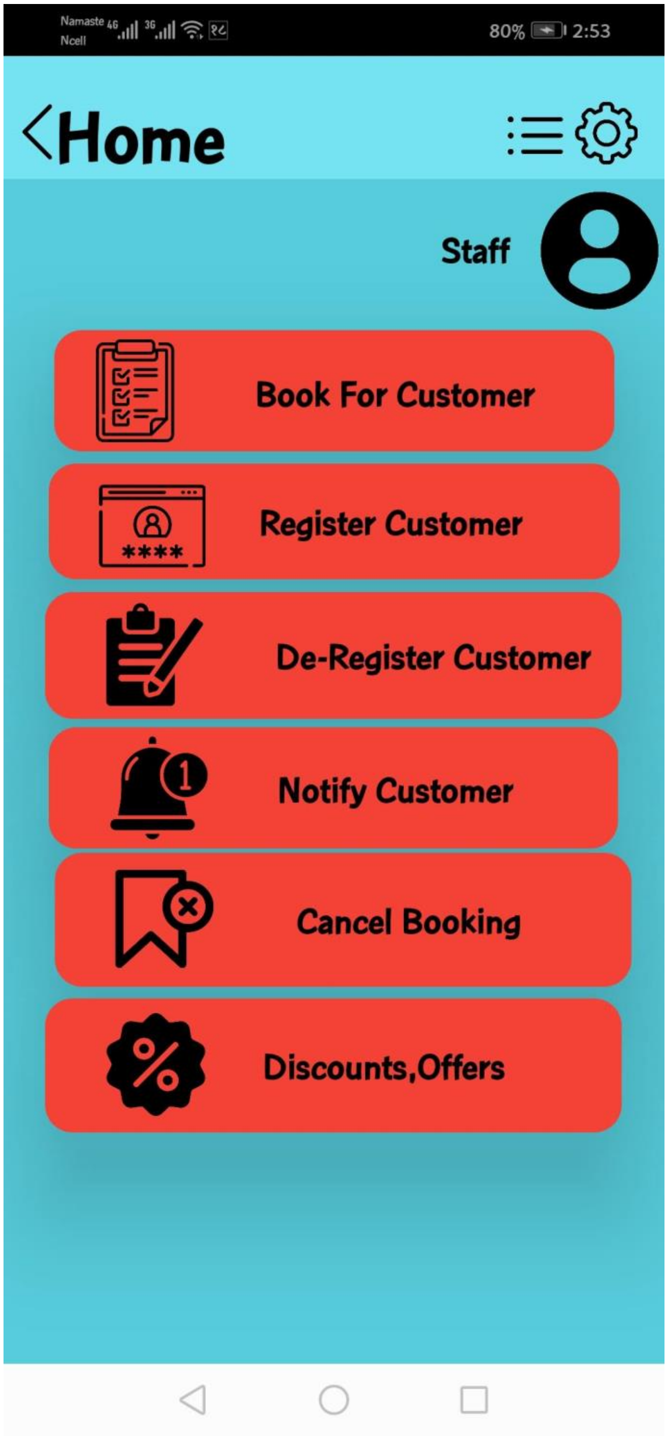


Figure 23: Screenshot of GUI of staff dashboard of prototype app

Dashboard Admin

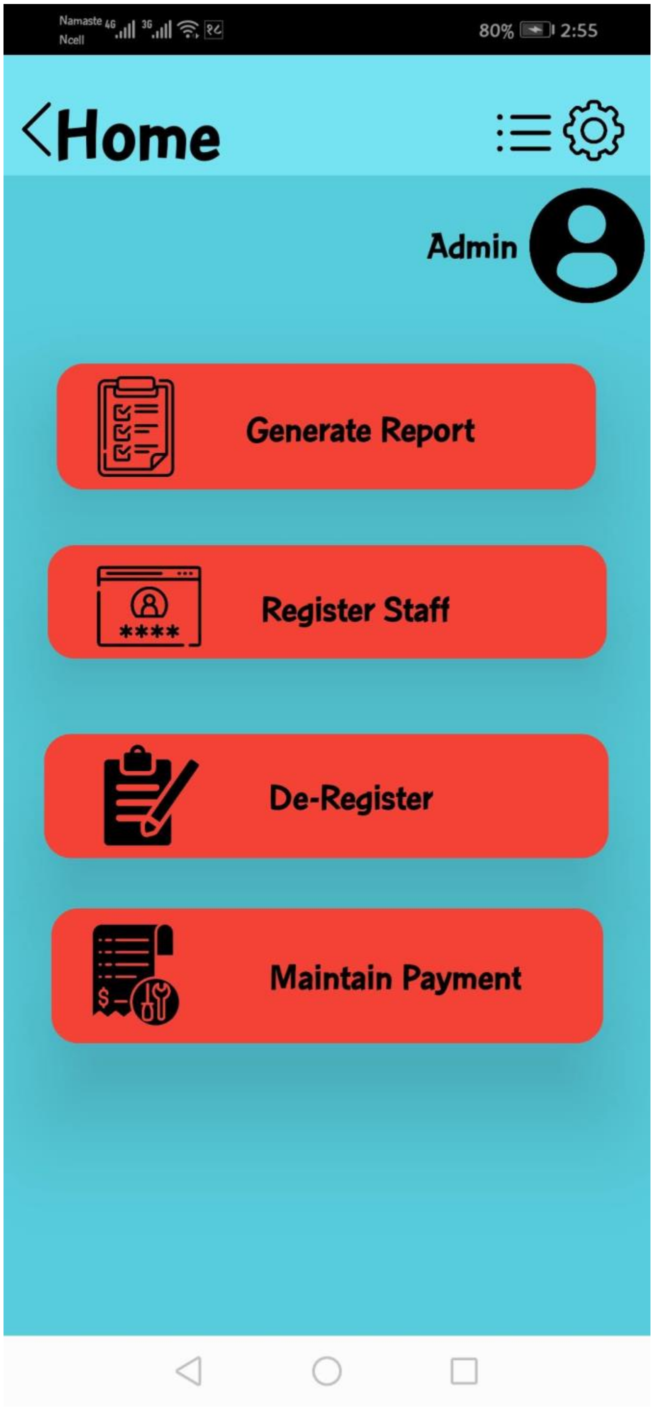
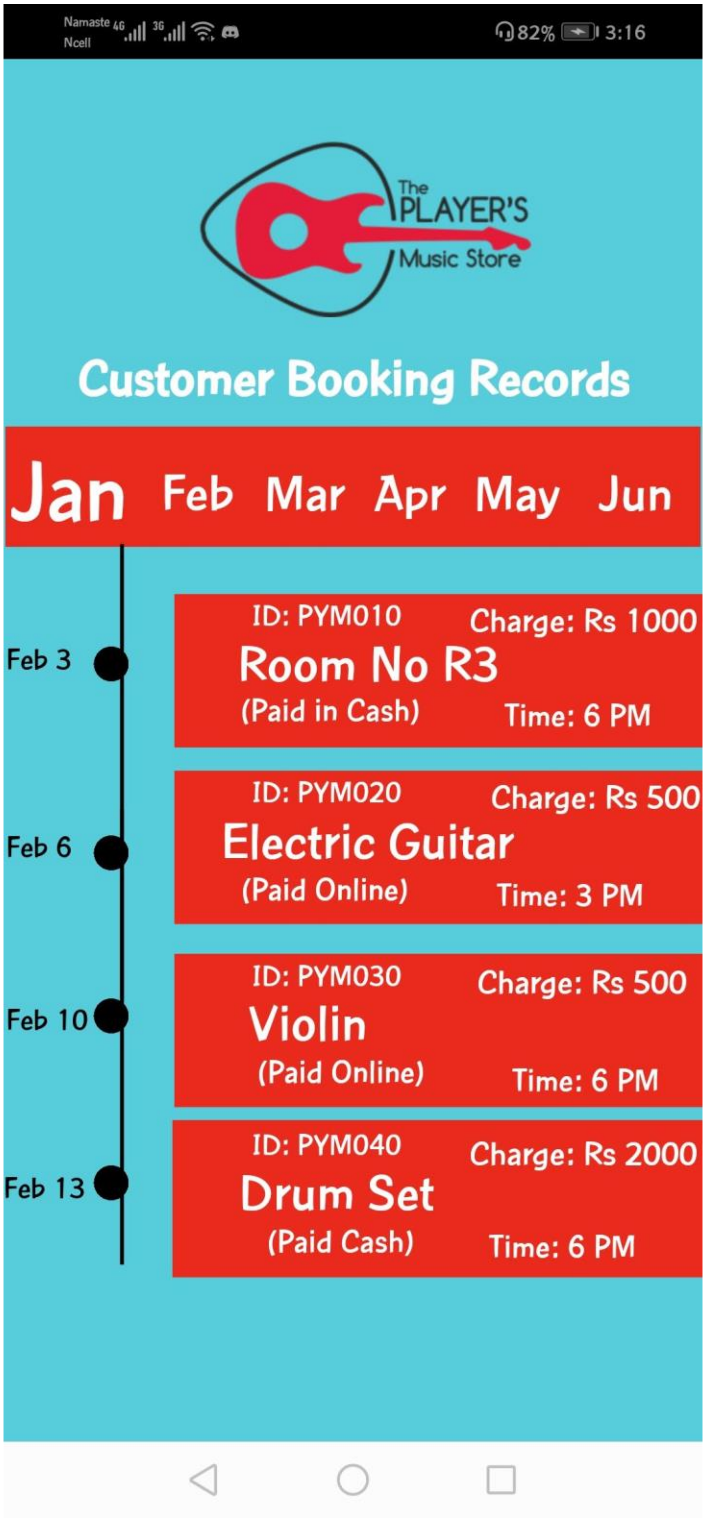


Figure 24: Screenshot of GUI of admin dashboard of prototype app

Booking Records



Customer Booking Records

Feb 3
ID: PYM010 Charge: Rs 1000
Room No R3
(Paid in Cash) Time: 6 PM

Feb 6
ID: PYM020 Charge: Rs 500
Electric Guitar
(Paid Online) Time: 3 PM

Feb 10
ID: PYM030 Charge: Rs 500
Violin
(Paid Online) Time: 6 PM

Feb 13
ID: PYM040 Charge: Rs 2000
Drum Set
(Paid Cash) Time: 6 PM

Figure 25: Screenshot of GUI of booking records of prototype app

9. Conclusion

The main aim of the coursework is to give us knowledge about how the software's are built with proper planning, designing and good teamwork. We learned about different diagrams such as use case diagram, collaboration diagram, sequence diagram and Gantt chart. The coursework helps me to get the knowledge of how designs are built for use cases, how the message communicates in use case, how the objects, classes are extracted from use cases. As the coursework is individual, it takes lots of time to complete on own. But, with the help of teachers, seniors, and friends it was completed in time.

10. References

gantt, 2021. [Online]
Available at: https://www.gantt.com/
[Accessed 29 4 2021].

geeksforgeeks, 2018. [Online]
Available at: https://www.geeksforgeeks.org/unified-modeling-language-uml-sequence-diagrams/#:~:text=A%20sequence%20diagram%20simply%20depicts,which%20these%20interactions%20take%20place.&text=These%20diagrams%20are%20widely%20used,for%20new%20and%20existing%20systems.
[Accessed 29 4 2021].

Larson Elizabeth, L. R., 2004. [Online]
Available at: https://www.pmi.org/learning/library/use-cases-project-manager-know-8262
[Accessed 29 4 2021].

rational, 1998. [Online]
Available at:
https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf
[Accessed 29 4 2021].

visual-paradigm, 2020. [Online]
Available at: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml-collaboration-diagram/
[Accessed 29 4 2021].