



**Module Code & Module Title**

**CS5002NI SOFTWARE ENGINEERING**

**Assessment Weightage & Type 35%**

**Individual Coursework**

**Year and Semester**

**2020-21 Autumn**

**Student Name: Siddhartha Ghimire**

**London Met ID: 19031691**

**College ID: NP01CP4A190148**

**Assignment Due Date: 2<sup>nd</sup> May, 2021**

**Assignment Submission Date: 2<sup>nd</sup> May, 2021**

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

## Table of Contents

<b>1. Introduction .....</b>	<b>1</b>
<b>2. Gantt Chart .....</b>	<b>1</b>
<b>3. Use Case Diagram.....</b>	<b>3</b>
3.1. Use Case Diagram .....	3
3.2. High Level Use Case Description .....	5
3.3. Expanded Use Case Description .....	6
<b>4. Communication Diagram .....</b>	<b>8</b>
4.1. Collaboration Diagram .....	8
4.2. Sequence Diagram .....	13
<b>5. Class Diagram .....</b>	<b>17</b>
<b>6. Overall Software Development Process .....</b>	<b>19</b>
<b>7. Prototype .....</b>	<b>23</b>
<b>8. Conclusion .....</b>	<b>31</b>

## Table of Figures

Figure 1: Gantt Chart .....	2
Figure 2: Use Case Diagram.....	4
Figure 3: Communication Diagram: Object for Domain Classes. ....	9
Figure 4: Communication Diagram: Control Object .....	10
Figure 5: Communication Diagram: Boundary Object.....	10
Figure 6: Communication Diagram: Actor.....	11
Figure 7: Communication Diagram: Associations .....	11
Figure 8: Communication Diagram: Messages.....	12
Figure 9: Communication Diagram for Payment of Customer.....	12
Figure 10: Sequence Diagram: Control Object Lifeline.....	13
Figure 11: Sequence Diagram: Boundary Object Lifeline .....	14
Figure 12: Sequence Diagram: Actor Lifeline .....	15
Figure 13: Sequence Diagram: Messages .....	15
Figure 14: Sequence Diagram: Object Lifeline .....	16
Figure 15: Sequence Diagram for Payment of Customer .....	17
Figure 16: Class Diagram .....	18
Figure 17: Design mockup for mobile and desktop users .....	20
Figure 18: Design Rules for Software.....	21
Figure 19: Customer Login.....	23
Figure 20: Customer Registration.....	24
Figure 21: Customer Timeline .....	25
Figure 22: Customer Payment .....	26
Figure 23: Staff Login.....	27
Figure 24: Staff Timeline .....	28
Figure 25: Admin Login.....	29
Figure 26: Admin Timeline .....	30
Figure 27: Staff Registration.....	31

## **1. Introduction**

The task's goal is to create an electronic system to keep track of customers and their payments for a music institute called Sound Strong. The center, which is currently experiencing several problems with keeping up consumer data, needs a system to deal with the records of their clients as well as their employees. The architecture will be built using the RUP method of programming development.

## **2. Gantt Chart**

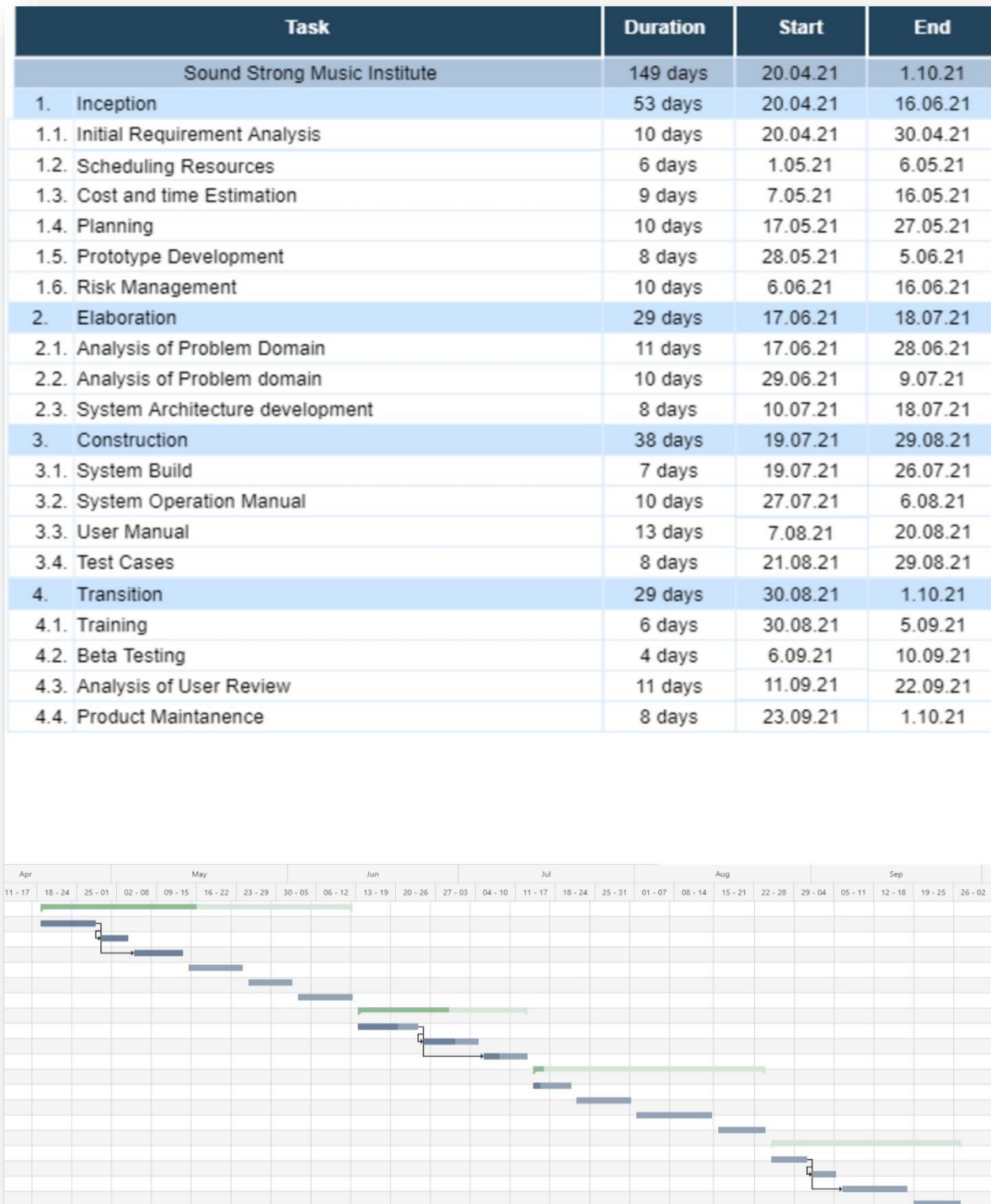


Figure 1: Gantt Chart

### **3. Use Case Diagram**

#### **3.1. Use Case Diagram**

A Use Case Diagram depicts the relationship between a customer and a device. It illustrates various methods for a person to interface with the machine.

The use case diagram for Sound Strong Music Institute is shown below.

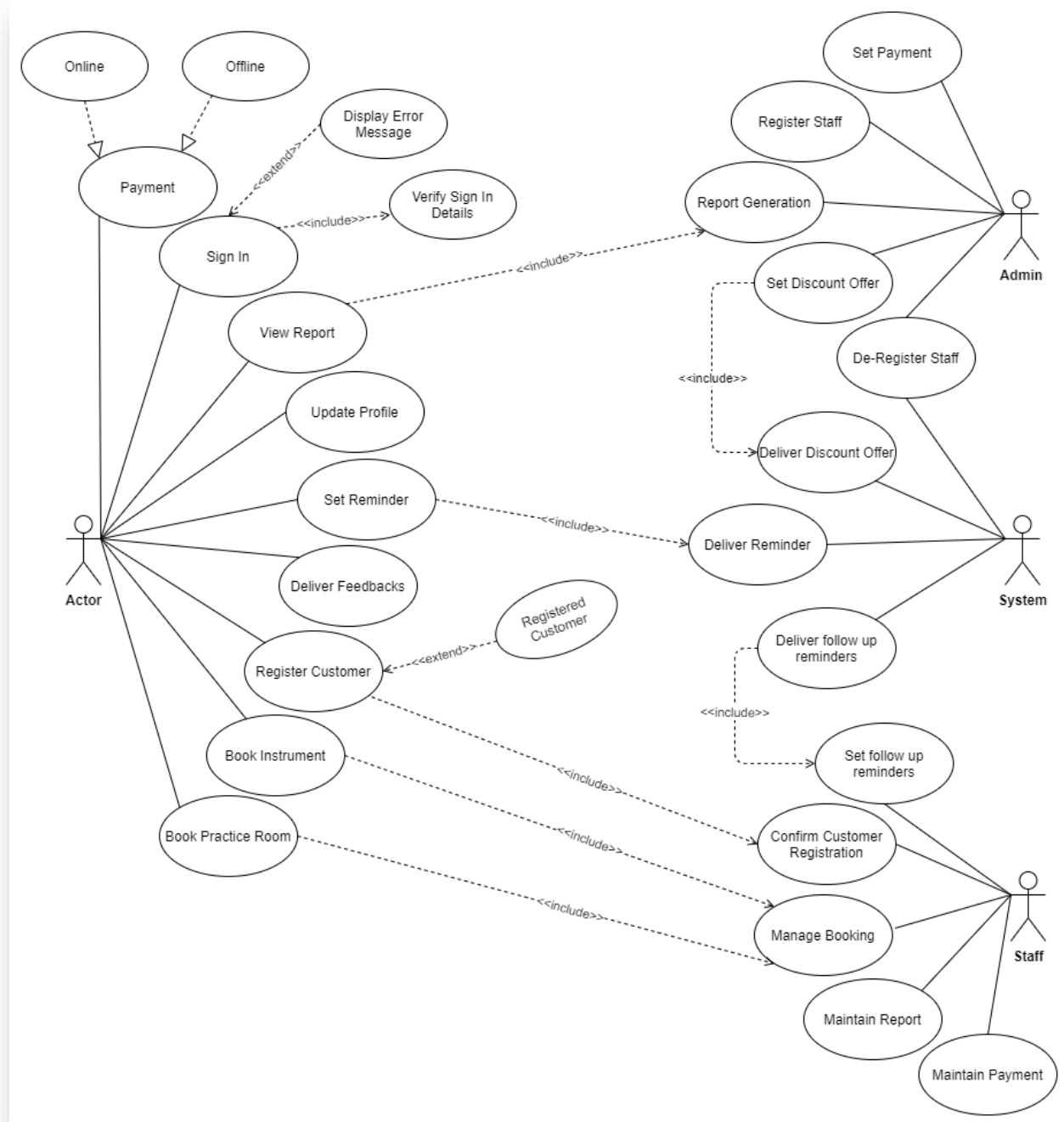


Figure 2: Use Case Diagram

### 3.2. High Level Use Case Description

- **Name:** Register customer
- **Actor(s):** Staff, Customer
- **Description:** The staff enters the customer's details.

- **Name:** Report Generation
- **Actor(s):** Admin, Customer, Staff
- **Description:** The admin obtains the customer's final report and produces files that the customer can use.

- **Name:** Payment of customer
- **Actor(s):** Customer, Admin, Staff
- **Description:** The customer is offered the option of making a purchase (online or offline), and users record it.

- **Name:** Maintain report
- **Actor(s):** Staff
- **Description:** The staff keeps and manages the customer's report, as well as records it.

- **Name:** Set reminder
- **Actor(s):** Customer, Staff
- **Description:** The customer is free to keep their reminder.



- **Name:** Set follow up reminder
- **Actor(s):** Staff
- **Description:** The staff records the reminder and sends a follow-up booking reminder to the customer.

- **Name:** Register a staff
- **Actor(s):** Admin
- **Description:** The admin assigns new staff to the system.

- **Name:** De-register staff
- **Actor(s):** Admin, System
- **Description:** After the contract term has expired, the admin de-registers the staff, and the system stores the staff's files.

### 3.3. Expanded Use Case Description

i.

- **Name:** Register customer
- **Actor(s):** Customer, Staff
- **Purpose:** Record customer's details
- **Overview:** A customer fills the necessary enrollment structure and submits it to the staff. The staff verify the details of the customer and finalizes the registration.
- **Type:** Primary
- **Action steps:**

Action Actor	System Response
1. The customer submits a registration form to the staff.	

2. Staff checks the registrations and records it in system.	
	3. Stores the customer details.
	4. Generate customer id and register its details.
5. Staff issues the Customer ID.	

- **Alternative course of action**

Line 2: If any records are absent or rounded out inaccurately, at that point the staff will request that the customer fill the structure again.

Line 3: If the customer is already registered, then the system will automatically follow up with the record.

## ii.

- **Name:** Payment of the Customer
- **Actor(s):** Customer, Staff, Admin
- **Purpose:** To complete the payment process and store the payment details of the customer
- **Overview:** The customer makes the payment and chooses the mode of payment(online/offline)
- **Type:** Customer
- **Action steps:**

Actor Action	System Response
1. The admin makes the amount of payment.	
2. The customer does the payment.	
	3. The payment method and the price is shown by the system.

4. The customer chooses online payment method and enters the payment type.	
	5. Verifies the bank details/mobile number.
	6. Final processing is done, and bill is provided.
7. The staff records the payment details	
	8. Stores the payment details.

- **Alternative course of action**

Line 4: If Customer chooses cash payment; the system stores the data from the reception of the Institute.

Line 5: If the bank details is not valid then the system shows popup message with Enter the valid bank details.

## 4. Communication Diagram

### 4.1. Collaboration Diagram

UML collaboration diagrams, including sequence diagrams - a kind of interaction diagram - illustrate how objects communicate or collaborate with one another. A collaboration diagram is an extension of an object diagram that depicts the objects as well as the signals that flow between them. In addition to the relations between objects, the collaboration diagram depicts the signals that the objects transmit to one another.

## Steps Involved in Drawing Collaboration Diagram

The steps involved in drawing collaboration diagram are as follows:

### 1. Finding the domain classes.

For the use case payment of customer, the domain classes are paymentOptions and paymentDetails.

- An object of class paymentOptions will be created so that the options for payment will be shown and the customer can make choice
- An object of class paymentDetails will be created to show the payment details of the customers and also to update whether the amount is paid.

### 2. Drawing an object symbol for each of the domain classes.



*Figure 3: Communication Diagram: Object for Domain Classes.*

### 3. Adding a control object

Control object manages the communication between objects.

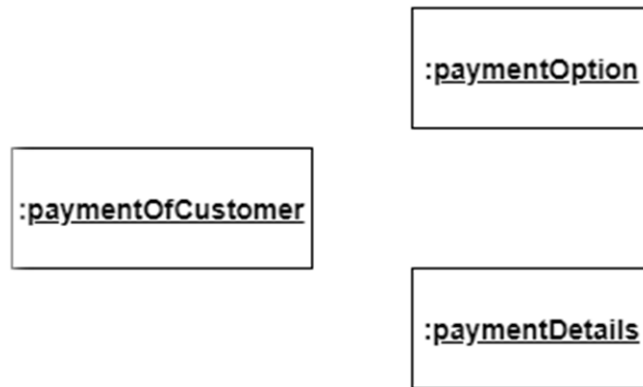


Figure 4: Communication Diagram: Control Object

#### 4. Adding a boundary object

Boundary object manages the screen interaction of the system. It is the object with which the user interacts.

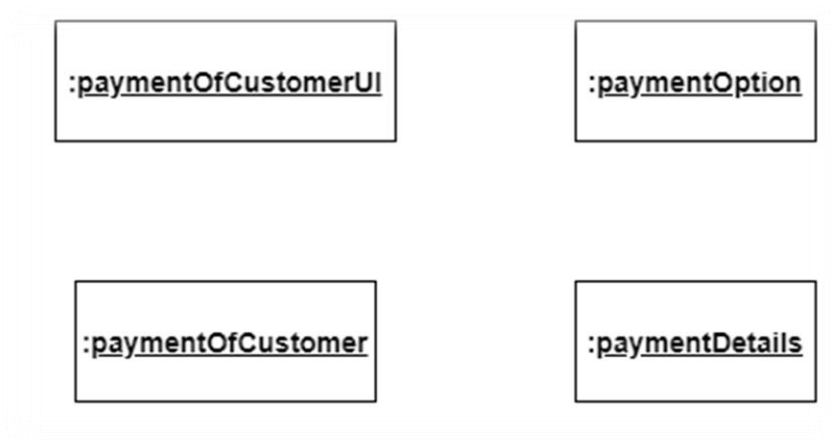


Figure 5: Communication Diagram: Boundary Object

#### 5. Adding an actor

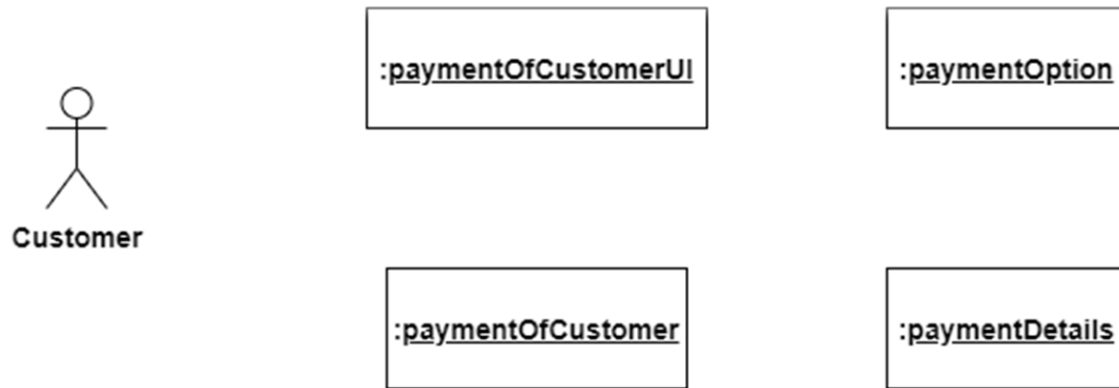


Figure 6: Communication Diagram: Actor

## 6. Adding associations

Associations are lines connecting any two objects which communicates with each other.

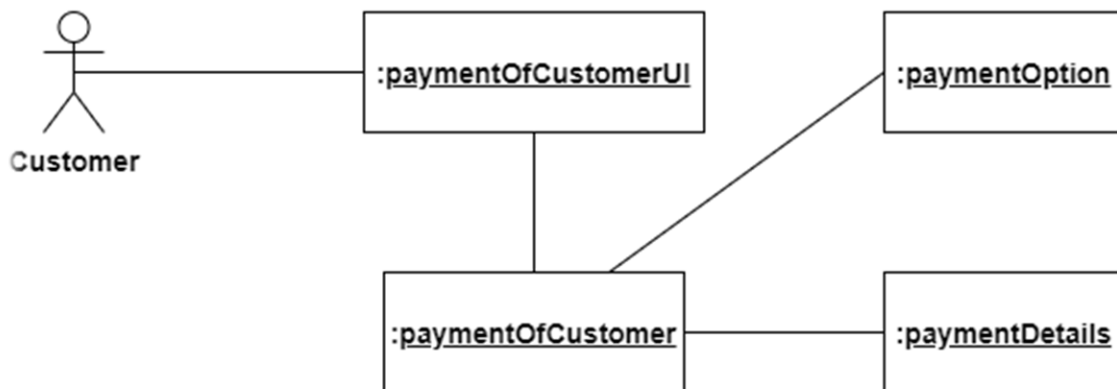


Figure 7: Communication Diagram: Associations

## 7. Adding messages

Messages are written above an arrow which points to direction of that object where the message needs to be delivered.

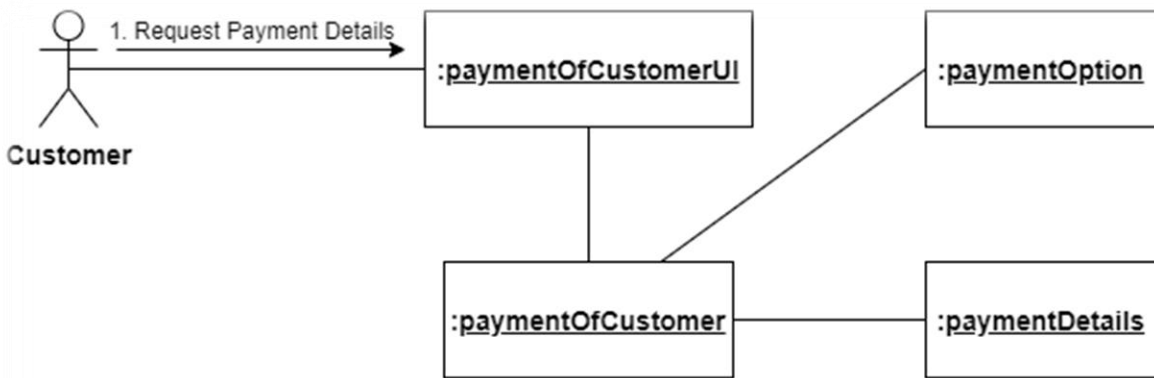


Figure 8: Communication Diagram: Messages

## 8. Final Collaboration Diagram

The final collaboration diagram for the use case payment of customer is shown below:

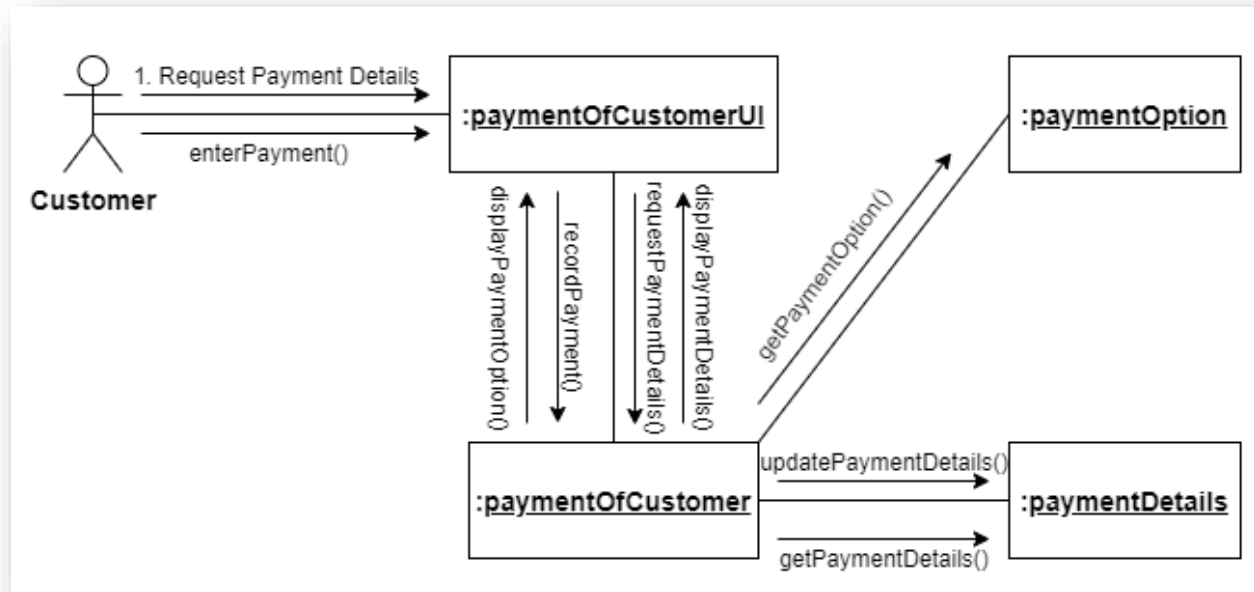


Figure 9: Communication Diagram for Payment of Customer

## 4.2. Sequence Diagram

Sequence Diagrams are interaction diagrams that show how processes are performed. They document the presence between objects in the form of a collaborative effort. Sequence Diagrams are time oriented, and they physically represent the order of the relationship by using the vertical axis of the diagram to represent time, what messages are received, and when.

### Steps Involved in Drawing Sequence Diagram

The steps involved in drawing a Sequence Diagram are as follows:

#### 1. Finding the domain classes.

The possible domain classes for use case paymentOfCustomer are paymentOption and paymentDetails.

#### 2. Drawing control object lifeline



Figure 10: Sequence Diagram: Control Object Lifeline



### 3. Drawing boundary object lifeline

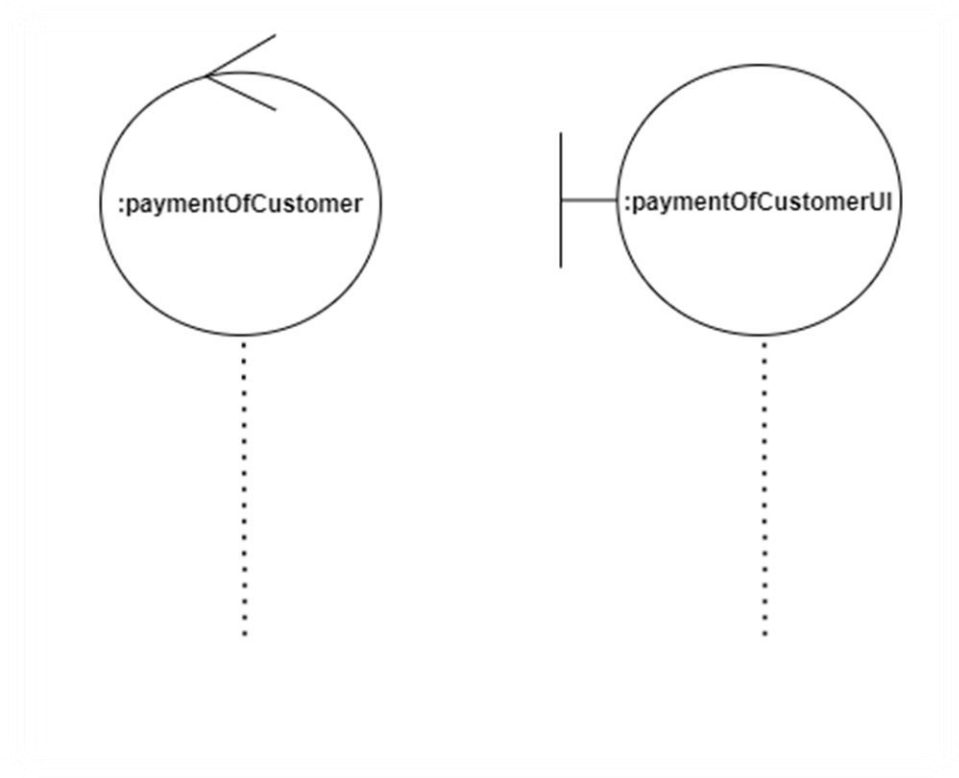


Figure 11: Sequence Diagram: Boundary Object Lifeline

### 4. Drawing actor lifeline

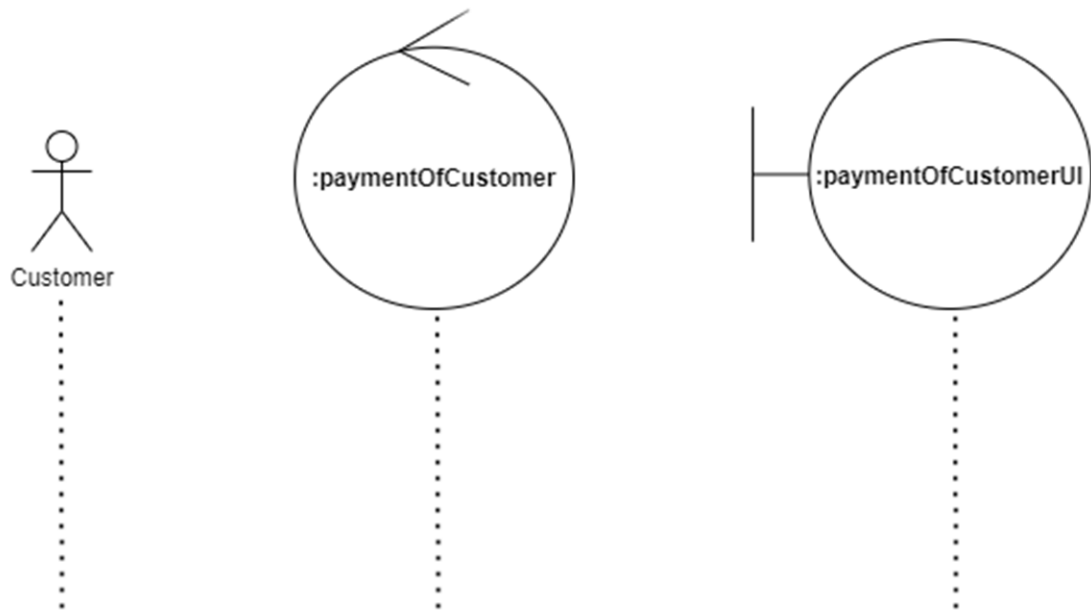


Figure 12: Sequence Diagram: Actor Lifeline

## 5. Adding messages

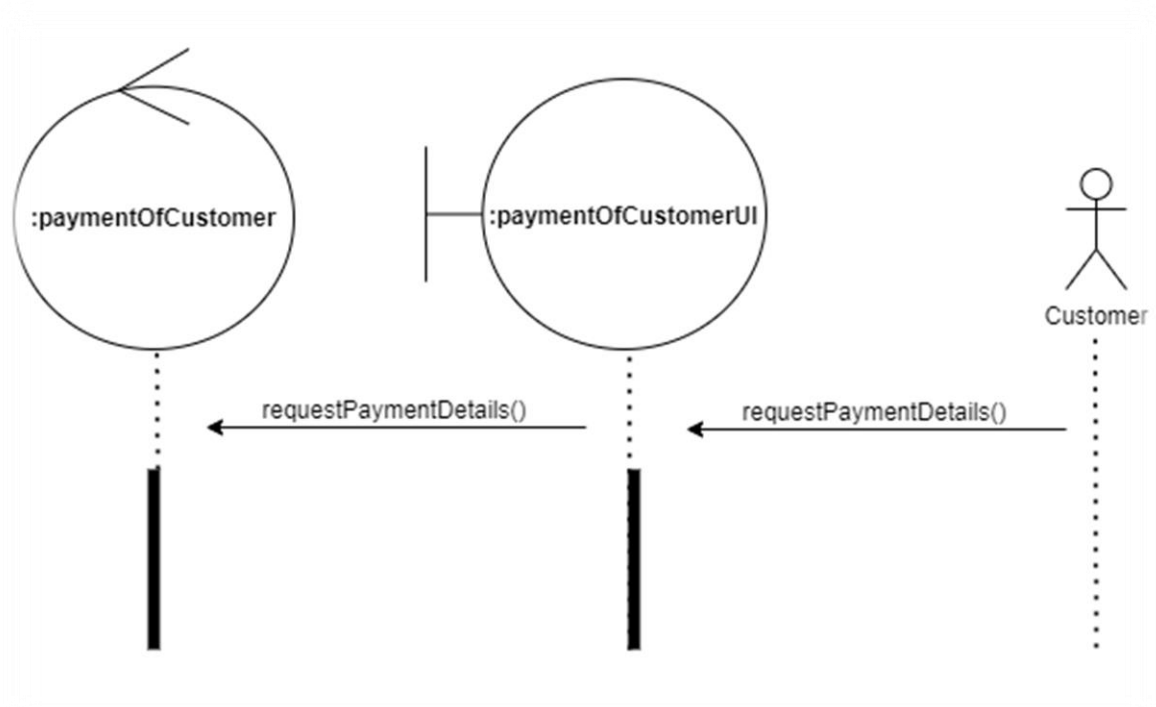


Figure 13: Sequence Diagram: Messages

## 6. Drawing object lifeline

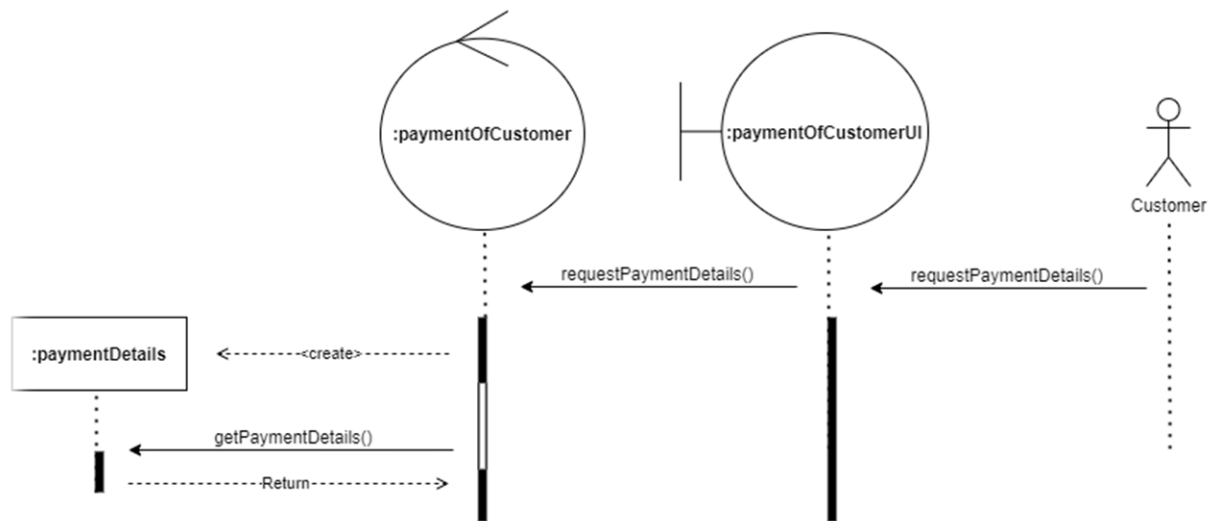


Figure 14: Sequence Diagram: Object Lifeline

## 7. Final Sequence Diagram

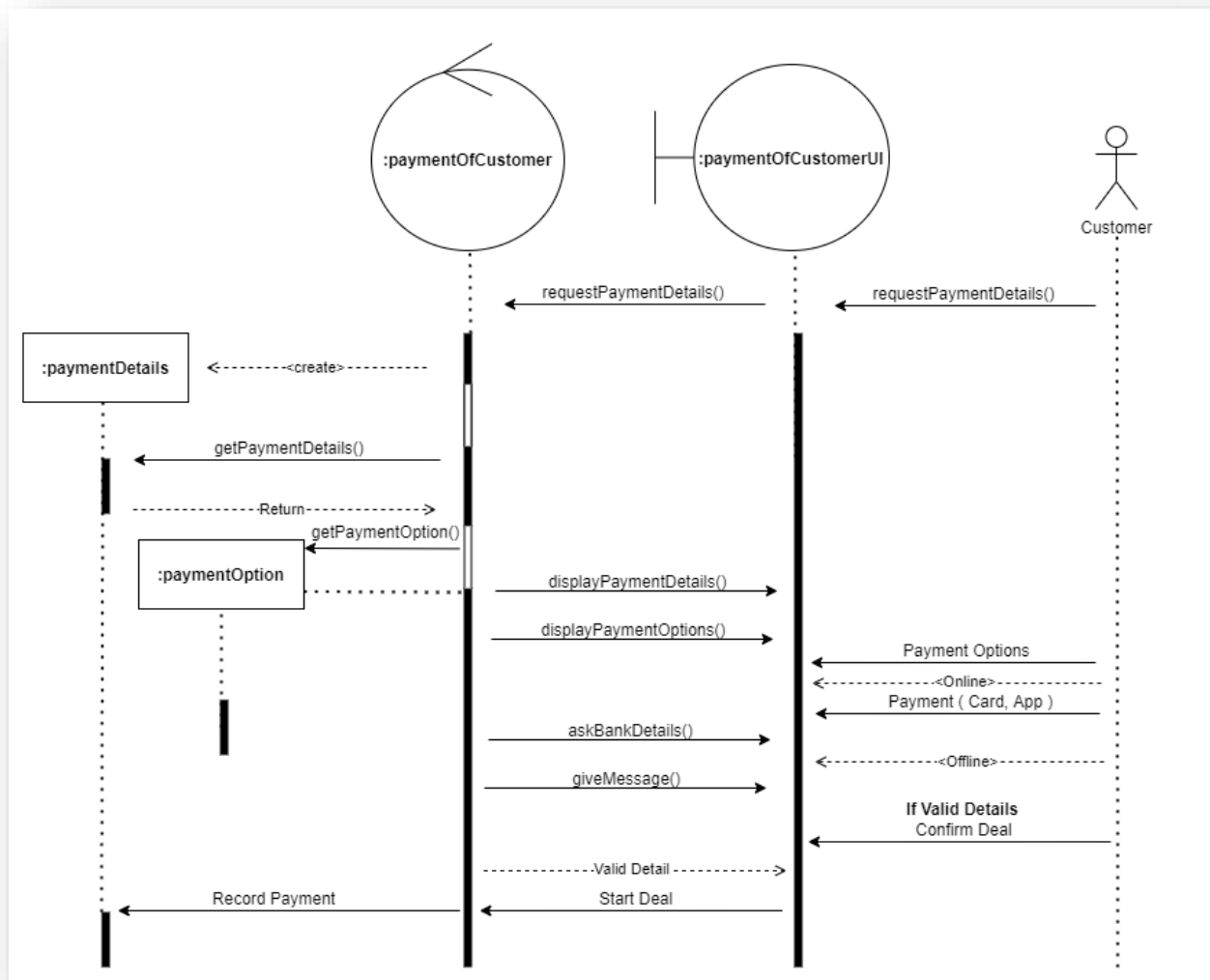


Figure 15: Sequence Diagram for Payment of Customer

## 5. Class Diagram

A class diagram is a hierarchical diagram that depicts the system's composition as well as the relationships between the classes.

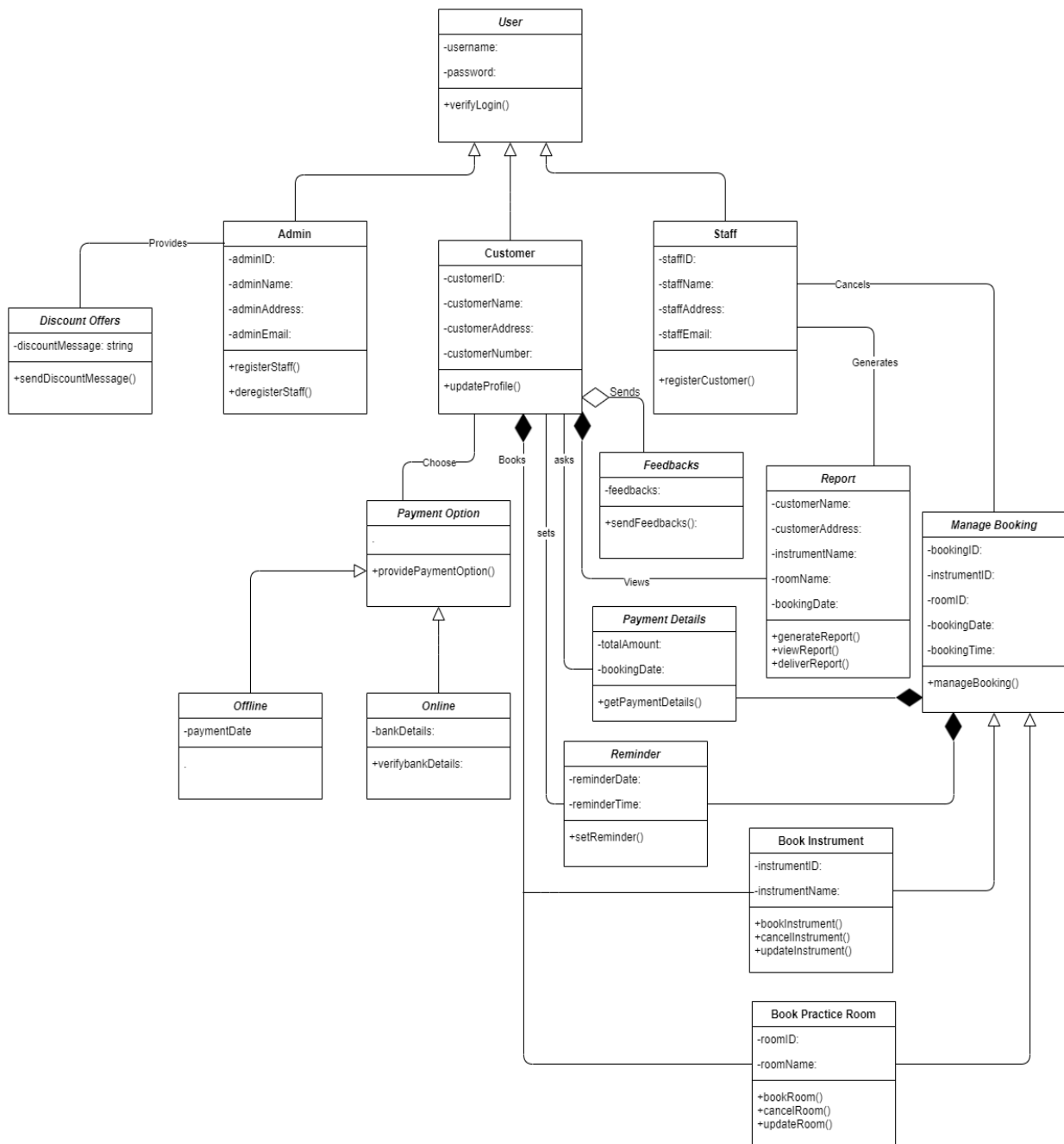


Figure 16: Class Diagram

## 6. Overall Software Development Process

The overall software development process follows Waterfall model and the requirement is all mentioned below:

- Requirement Analysis
  1. Customers must be able to book available rooms and instruments from their timeline as they log in through the portal.
  2. Customers must have to register in order to use the system.
  3. Staff must handle the overall registration verification process of customer and verifies registration through phone number or email.
  4. Staff can provide a discount scheme and a special package for customers who have membership by texts or messages.
  5. Staff should be registered through admin and must be verifies through texts or messages.
  6. There must be a feedback section for customers to rate and review the software, and the admin can only accept and respond to feedback through text or messages.
  7. Staff should get all the booking information in manage booking section from customers.
  8. Customer's timeline should include sections for online and offline payment gateways, and if the payment is completed, the customer should be notified through texts or messages. The online payment gateways should have the option of banking and mobile payment. Mobile payment should contain fonenpay, esewa, Khalti and paypal. Banking should contain card or cheque.
  9. Customer should able to view the reports which will be provided by the staff.
  10. System should auto generate reports according to the information provided by the customer and staff can able to view and forward reports to customer.
- Design

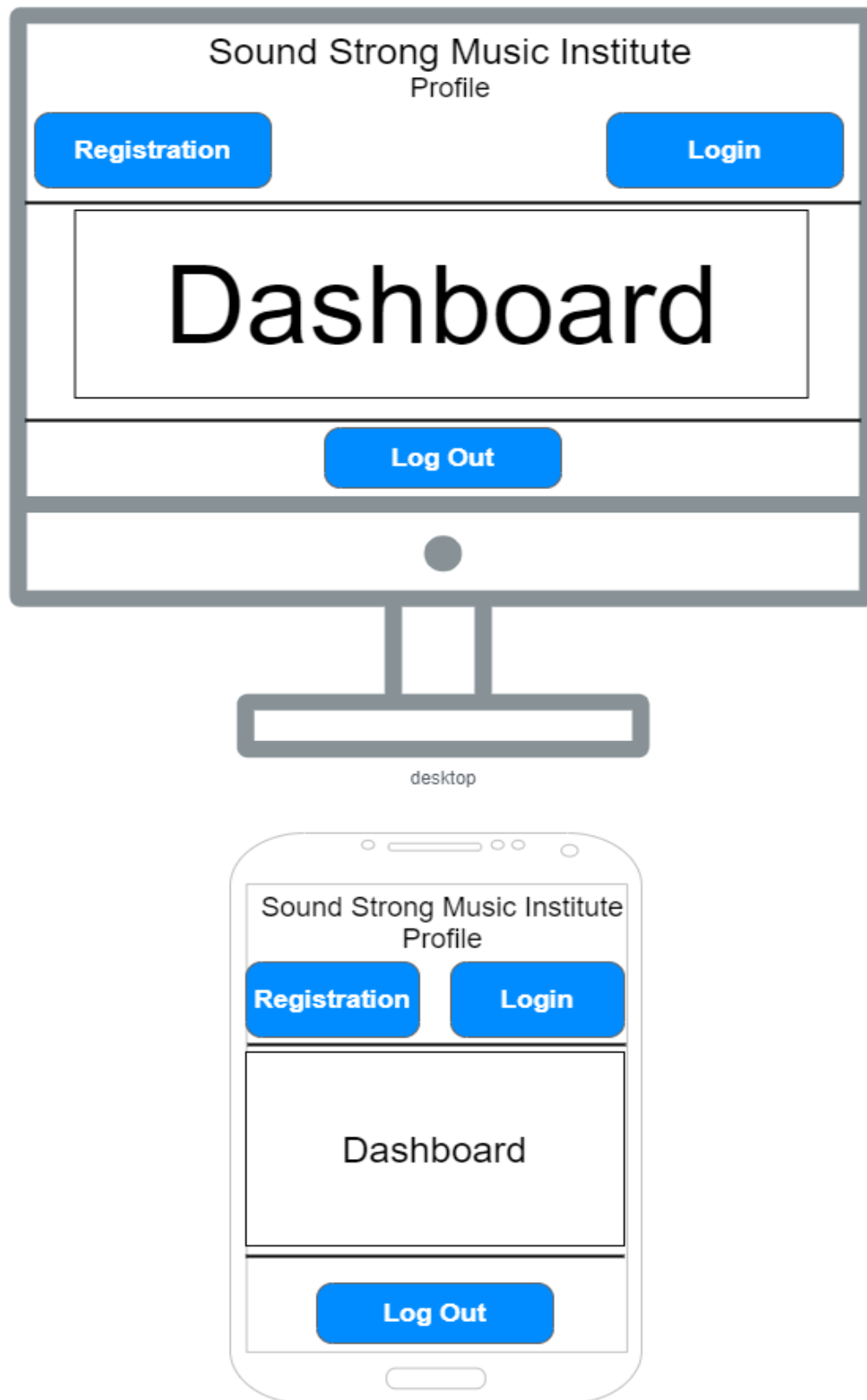


Figure 17: Design mockup for mobile and desktop users



Figure 18: Design Rules for Software.

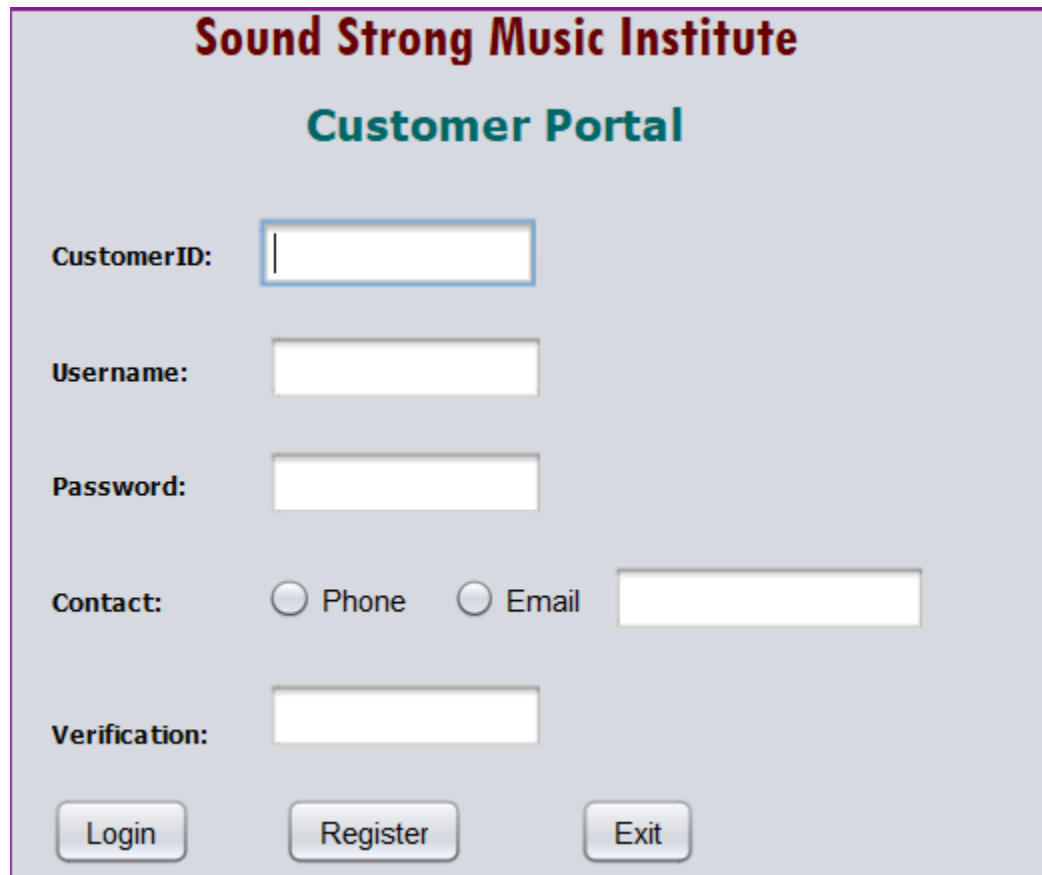


- Implementation or coding
  1. The above mockup design should be translated into the source code.
  2. Database should be implement with proper relationship which is mentioned in the class diagram above.
  3. Java programming language must be implement for both mobile and desktop users.
  4. The software should not throw any exception error and proper validation must be done.
- Testing
  1. The testing must start once the coding is done.
  2. The developed software should be tested thoroughly and if any defects were found then the software should assigned to the developers to get them fixed.
  3. Testing should be done until the point at which the software is as per the requirement analysis expectation.
- Deployment
  1. Once the software is tested properly, it must be deployed in the production environment depending upon the clients expectation.
  2. The production environment should consists of clients along with developers and testers.
  3. Every thing must be completed according to the clients expectation without any error.
- Maintenance
  1. On the maintenance of the product if any issues arrise, then the developer must fix the error until it doesnot throw any exception.

## 7. Prototype

Type: Customer

- Log In

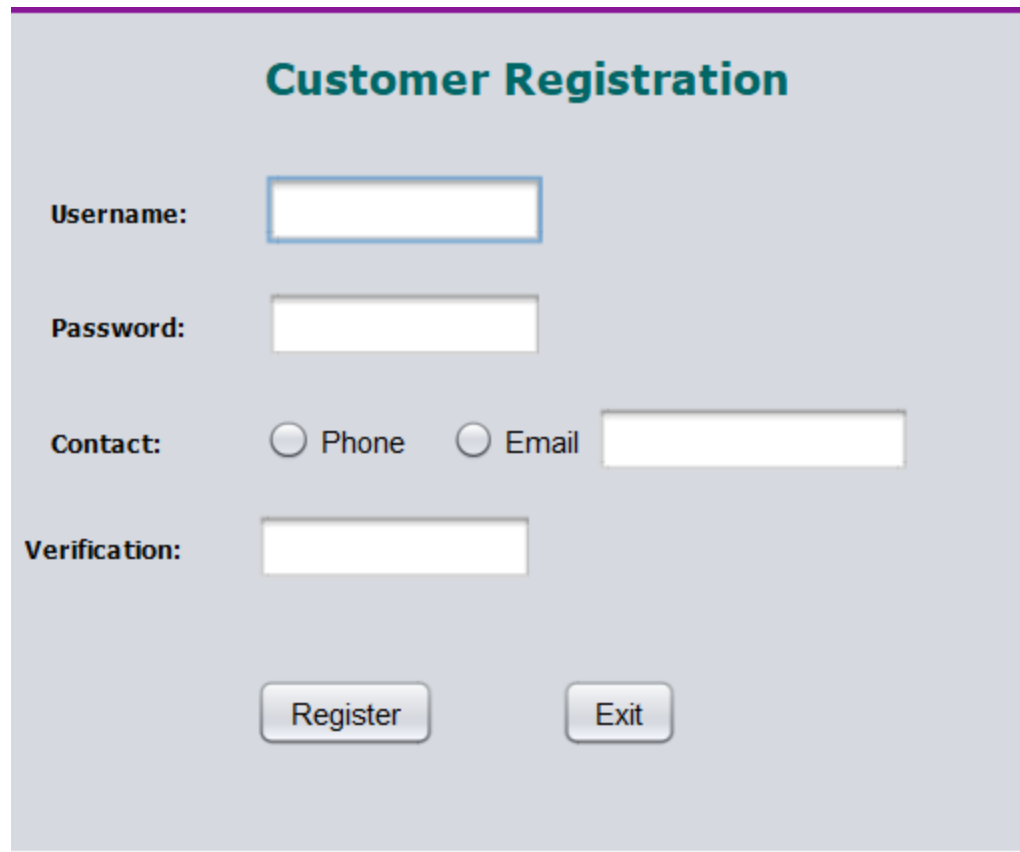


The image shows a web form titled "Sound Strong Music Institute Customer Portal". The form is set against a light gray background with a purple border. It contains the following fields and controls:

- CustomerID:** A text input field with a blue border.
- Username:** A text input field.
- Password:** A text input field.
- Contact:** Two radio buttons labeled "Phone" and "Email", followed by a text input field.
- Verification:** A text input field.
- Buttons:** Three buttons at the bottom labeled "Login", "Register", and "Exit".

Figure 19: Customer Login

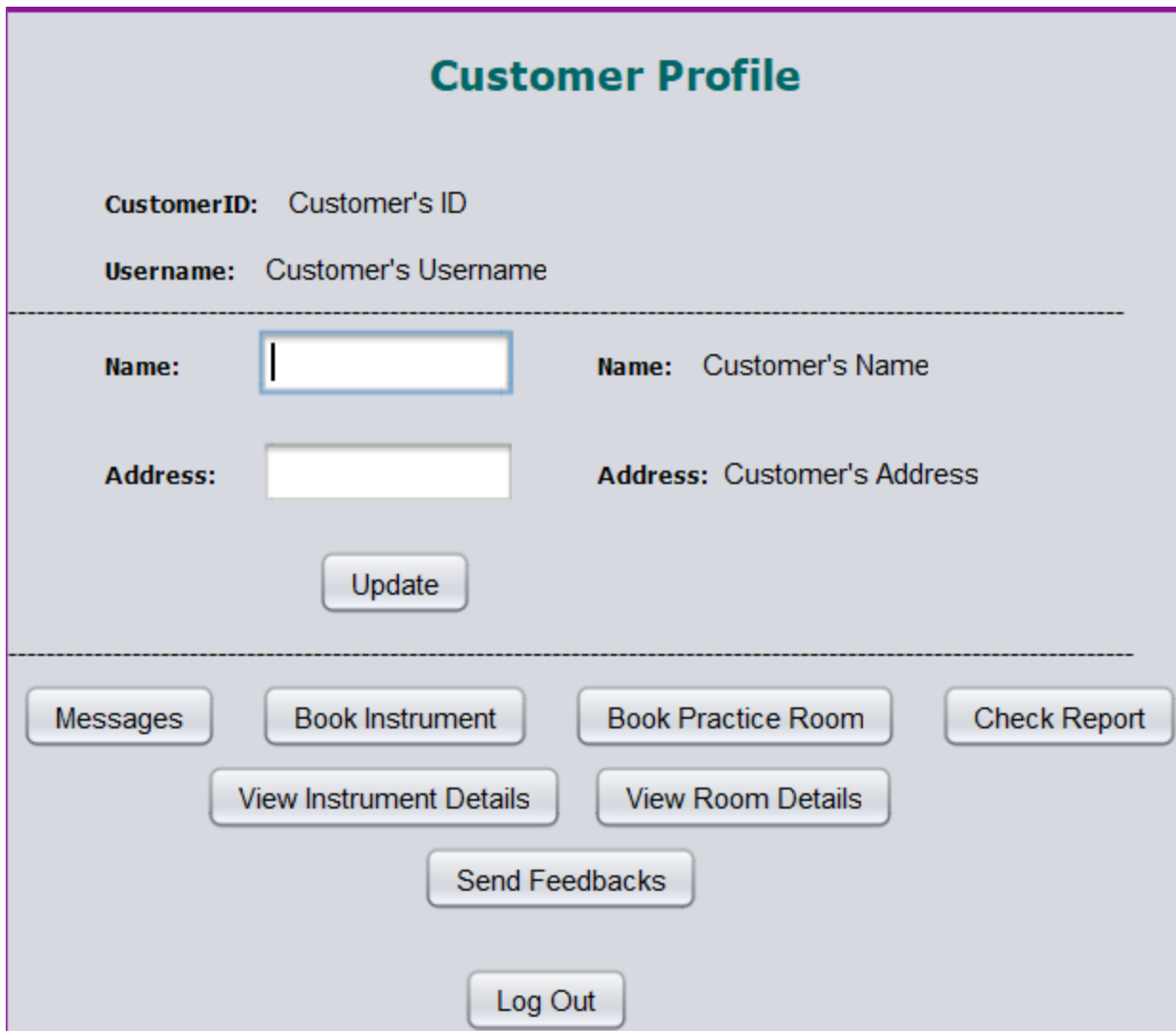
- Register Customer



The image shows a 'Customer Registration' form with a light gray background and a purple border. The title 'Customer Registration' is centered at the top in a bold, dark teal font. Below the title, there are four labeled input fields: 'Username:' with a white rectangular box, 'Password:' with a white rectangular box, 'Contact:' with two radio buttons labeled 'Phone' and 'Email' followed by a white rectangular box, and 'Verification:' with a white rectangular box. At the bottom of the form, there are two buttons: 'Register' and 'Exit', both with a light gray gradient and rounded corners.

*Figure 20: Customer Registration*

- Customer Profile



The image shows a web form titled "Customer Profile" with a light blue background and a purple border. At the top, the title "Customer Profile" is in bold teal. Below it, there are two rows of labels: "CustomerID: Customer's ID" and "Username: Customer's Username". A dashed horizontal line separates these from the input fields. The input section has two columns. The left column has "Name:" followed by a text input field with a blue border, and "Address:" followed by a text input field. The right column has "Name: Customer's Name" and "Address: Customer's Address". Below the input fields is a rounded "Update" button. Another dashed horizontal line follows. Below this, there are five buttons: "Messages", "Book Instrument", "Book Practice Room", "Check Report", and "View Instrument Details" (which is centered below the first two). Below these is "View Room Details" (centered below the next two), then "Send Feedbacks" (centered below the next two), and finally "Log Out" (centered at the bottom).

**Customer Profile**

**CustomerID:** Customer's ID

**Username:** Customer's Username

---

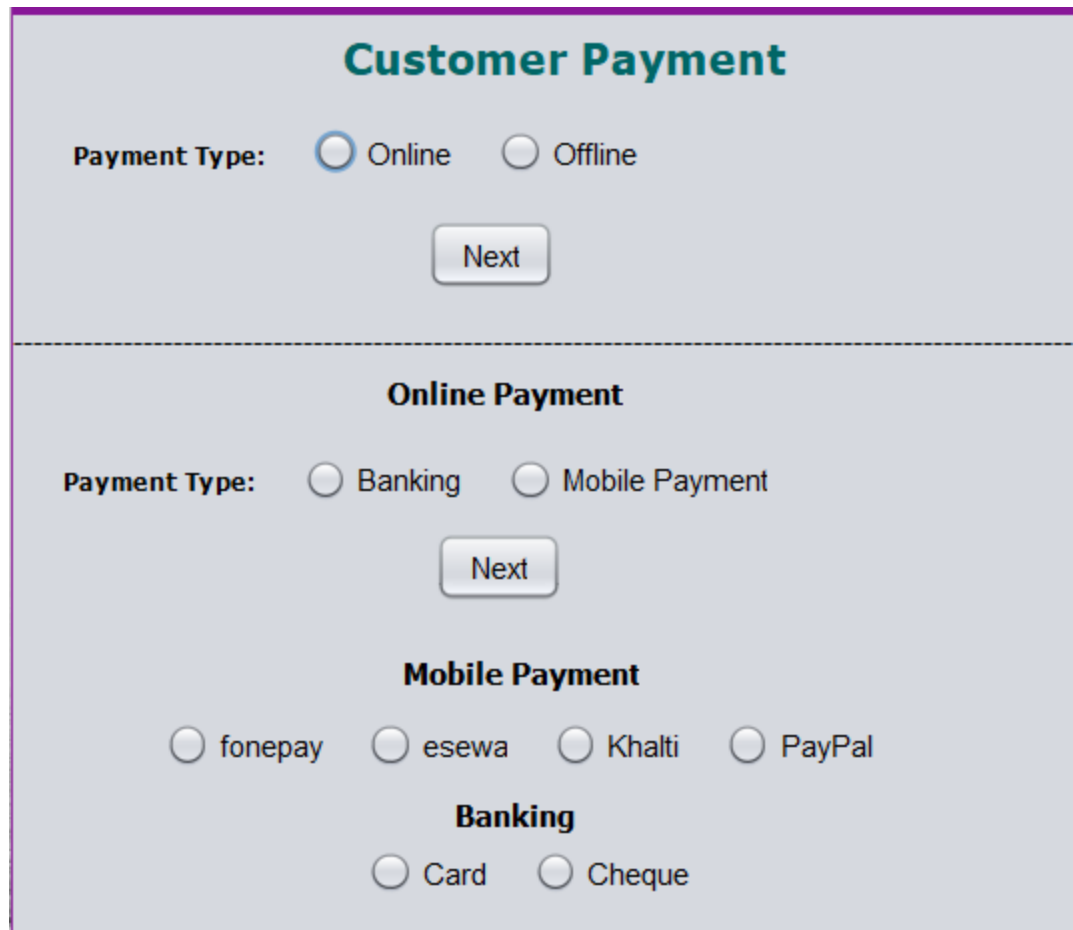
**Name:**  **Name:** Customer's Name

**Address:**  **Address:** Customer's Address

---

Figure 21: Customer Timeline

- Customer Payment



The form is titled "Customer Payment" in a large, bold, teal font. It is divided into three sections by horizontal dashed lines. The first section, "Customer Payment", contains a "Payment Type:" label followed by two radio buttons: "Online" (which is selected) and "Offline". Below these is a "Next" button. The second section, "Online Payment", contains a "Payment Type:" label followed by two radio buttons: "Banking" and "Mobile Payment". Below these is another "Next" button. The third section, "Mobile Payment", contains four radio buttons: "fonepay", "esewa", "Khalti", and "PayPal". Below this section is a "Banking" section with two radio buttons: "Card" and "Cheque".

**Customer Payment**

Payment Type: ☒ Online ☐ Offline

Next

---

**Online Payment**

Payment Type: ☐ Banking ☐ Mobile Payment

Next

**Mobile Payment**

☐ fonepay ☐ esewa ☐ Khalti ☐ PayPal

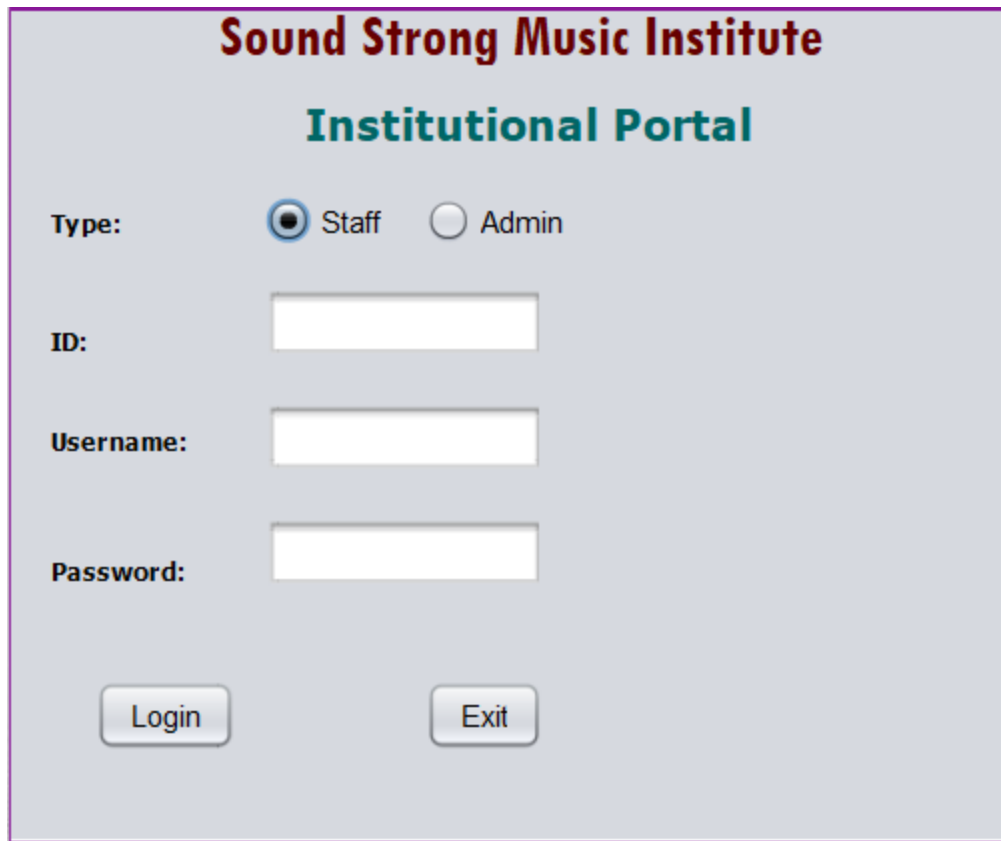
**Banking**

☐ Card ☐ Cheque

Figure 22: Customer Payment

**Type: Staff**

- Staff Login



The image shows a web form titled "Sound Strong Music Institute Institutional Portal". The form is set against a light gray background with a purple border. It contains the following elements:

- Title:** "Sound Strong Music Institute" in a large, bold, dark red font, and "Institutional Portal" in a bold, teal font below it.
- Type:** A label followed by two radio buttons. The first radio button is selected (indicated by a black dot) and is labeled "Staff". The second radio button is unselected and is labeled "Admin".
- ID:** A label followed by a white rectangular input field.
- Username:** A label followed by a white rectangular input field.
- Password:** A label followed by a white rectangular input field.
- Buttons:** Two buttons at the bottom: "Login" and "Exit", both with a light gray gradient and rounded corners.

*Figure 23: Staff Login*

- Staff Profile

**Staff Profile**

**StaffID:** Staff's ID

**Username:** Staff's Username

---

**Name:**  **Name:** Staff's Name

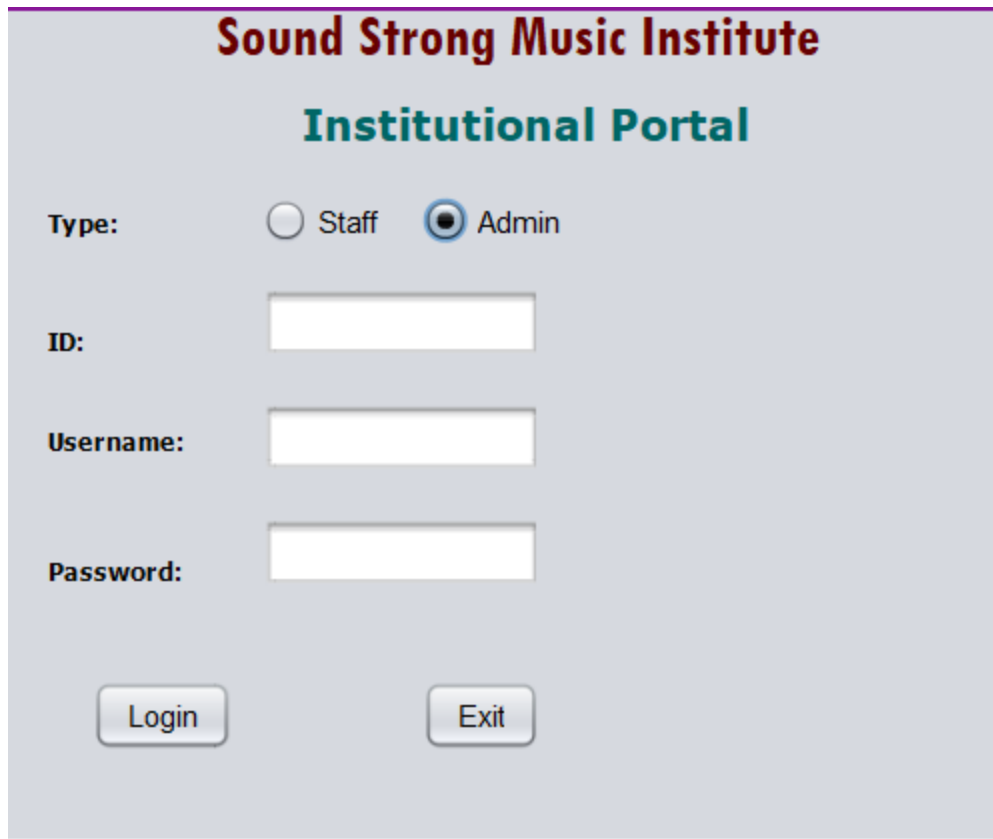
**Address:**  **Address:** Staff's Address

---

Figure 24: Staff Timeline

**Type: Admin**

- Admin Login



The image shows a login form for the 'Sound Strong Music Institute Institutional Portal'. The form has a light gray background with a purple border. At the top, the title 'Sound Strong Music Institute' is in a large, bold, dark red font, and 'Institutional Portal' is in a bold, teal font. Below the title, there are two radio buttons for 'Type': 'Staff' (unselected) and 'Admin' (selected). Below this, there are three text input fields labeled 'ID:', 'Username:', and 'Password:'. At the bottom, there are two buttons: 'Login' and 'Exit'.

**Sound Strong Music Institute**  
**Institutional Portal**

Type: ☐ Staff ☒ Admin

ID:

Username:

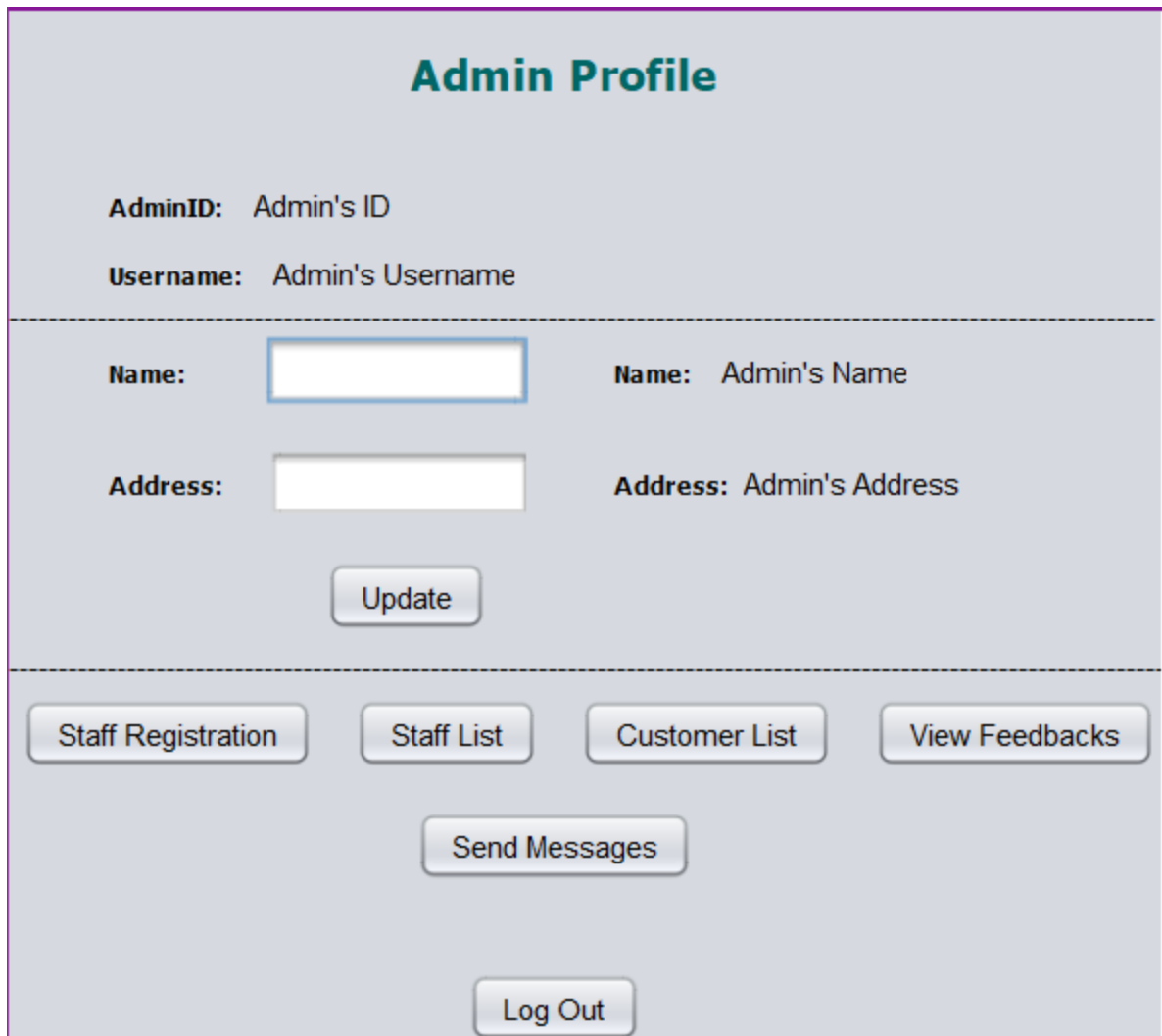
Password:

Login Exit

Figure 25: Admin Login

- Admin Profile





The image shows a web form titled "Admin Profile" in a teal font. The form is divided into two main sections by a dashed horizontal line. The top section contains labels for "AdminID: Admin's ID" and "Username: Admin's Username". The bottom section contains labels for "Name:" and "Address:" next to empty text input fields. To the right of these fields are labels "Name: Admin's Name" and "Address: Admin's Address". Below the input fields is an "Update" button. Below the "Update" button is a row of four buttons: "Staff Registration", "Staff List", "Customer List", and "View Feedbacks". Below these buttons is a "Send Messages" button. At the bottom of the form is a "Log Out" button.

**Admin Profile**

**AdminID:** Admin's ID

**Username:** Admin's Username

---

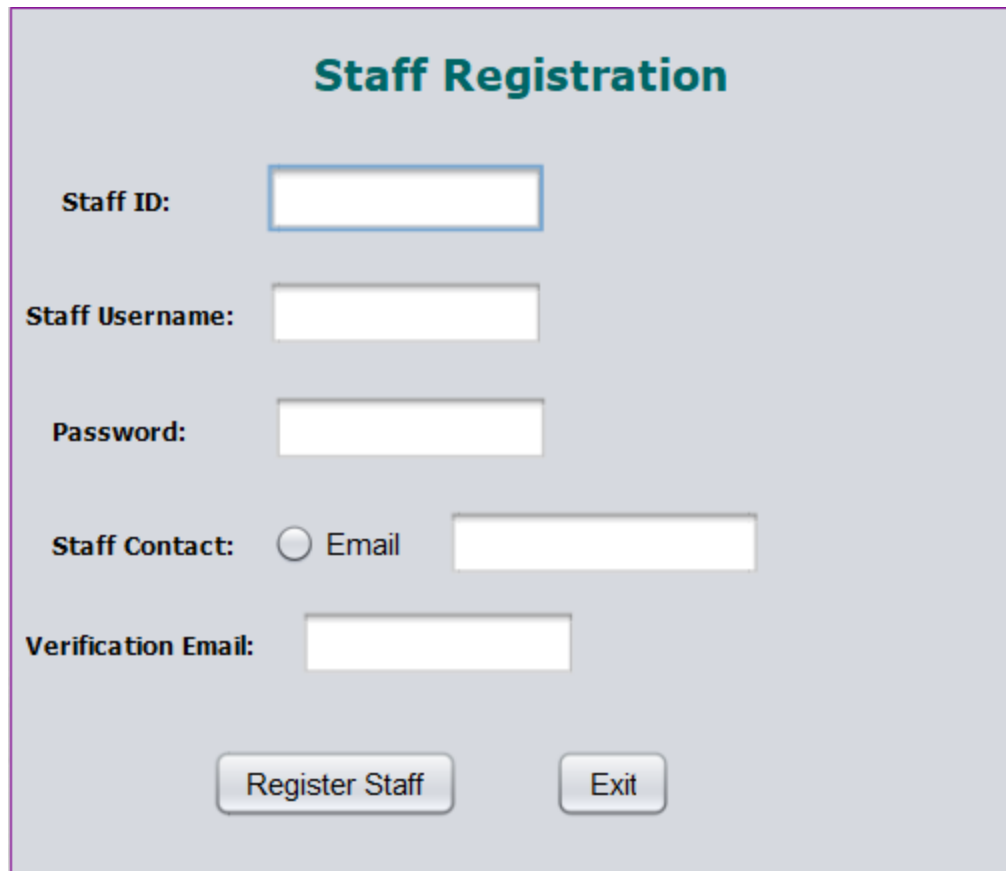
**Name:**  **Name:** Admin's Name

**Address:**  **Address:** Admin's Address

---

Figure 26: Admin Timeline

- Register Staff



The image shows a web form titled "Staff Registration" in a bold, dark blue font. The form is set against a light gray background. It contains five input fields: "Staff ID:" (a text box), "Staff Username:" (a text box), "Password:" (a text box), "Staff Contact:" (a radio button labeled "Email" followed by a text box), and "Verification Email:" (a text box). At the bottom of the form, there are two buttons: "Register Staff" and "Exit", both with a light blue gradient and rounded corners.

*Figure 27: Staff Registration*

## 8. Conclusion

The coursework which is been implemented in this report is all about the Sound Strong Music Institute that can now able to handle every record of information provided. This report is been finalized after the great research, consulting with the instructor (Mr. Rajesh Dware) time to time, patience and hard work. Implementing the concepts of RUP methodology, use case model, communication diagram and class diagram was quite fun and thrilling. This report is been completed after getting the concepts of scenario provided from the coursework.