



**slington college**  
(इस्लिङ्टन कलेज)

**Module Code & Module Title**

**CS5002NI SOFTWARE ENGINEERING**

**Assessment Weightage & Type**

**35% Individual Coursework**

**Year and Semester**

**2020-21Autumn**

**Student Name: Summit Shrestha**

**London Met ID: 19032520**

**College ID: NP01CP4A190113**

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

**Assignment Submission Date: 1<sup>st</sup> May 2021**

**Word Count (Where Required):2909**

*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

1. Introduction .....	1
2. Gantt Chart .....	3
2.1. Description.....	3
2.2. Gantt Chart .....	3
3. Use Case Diagram .....	4
3.1. Description.....	4
3.2. Use-Case Diagram .....	4
3.3. High Level Description.....	5
3.4. Expanded Level Description .....	8
4. Communication Diagram – Sequence Diagram.....	10
4.1. Steps to Draw Communication Diagram .....	10
4.2. Communication Diagram .....	10
4.3. Steps to Draw Sequence Diagram .....	12
4.4. Sequence Diagram .....	14
5. Class Diagram.....	15
5.1. Steps to Draw Class Diagram.....	15
5.2. Use Cases and Domain Classes.....	16
5.3. Class Diagram.....	17
6. Explanation .....	18
7. Prototype .....	20
7.1. Customer .....	20
7.2. Staff.....	24
7.3. Admin.....	28
8. Conclusion.....	31
References .....	32

## Table of Figures

Figure 1: Gantt chart .....	3
Figure 2: Use Case Diagram .....	4
Figure 3: Communication Diagram .....	11
Figure 4: Sequence Diagram.....	14
Figure 5: Class Diagram .....	17
Figure 6: Register Customer .....	20
Figure 7: Customer Login .....	21
Figure 8: Customer Home.....	21
Figure 9: Customer Book Room .....	22
Figure 10: Customer Payment .....	22
Figure 11: Customer Membership .....	23
Figure 12: Staff Login.....	24
Figure 13: Staff Home .....	25
Figure 14: Staff Customer Details .....	25
Figure 15: Staff Booking Details .....	26
Figure 16: Staff Payment Details.....	26
Figure 17: Staff generate Report.....	27
Figure 18: Admin Login .....	28
Figure 19: Admin Home .....	28
Figure 20: Admin Register Staff.....	29
Figure 21: Admin Staff Details.....	29
Figure 22: Admin Report Details.....	30

## Table of Tables

Table 1: Expanded Description of Register Customer .....	8
Table 2: Expanded Description of Make Payment. ....	9
Table 3: Use Cases and Domain Classes .....	16

## 1. Introduction

The coursework requires us to develop a system on a music store named “Sound Strong Music” which is currently tackling the issues in managing records of customer, member and their bookings. According to the guidelines of the coursework the methodology that is required to be used in software development is RUP (Rational Unified Process) methodology. The Rational Unified Process® is a Software Engineering Process. It provides a disciplined approach to assigning tasks and responsibilities within a development organization. Its goal is to ensure the production of high-quality software that meets the needs of its end-users, within a predictable schedule and budget (Software, 2021). The RUP methodology breaks the development cycle of a software into four consecutive phases.

**i) Transition Phase:** During the inception phase, you develop the system's business case and define the project's scope. To do so, you must first identify all external entities (actors) with which the system will interact and then define the nature of this interaction at a high level. This entails identifying all use cases and describing a few key ones. The business case includes success criteria, a risk assessment, an estimate of the resources required, and a phase plan with key milestone dates.

**ii) Elaboration Phase:** The elaboration phase's goal is to analyze the problem domain, lay a solid architectural foundation, create the project plan, and eliminate the project's highest risk elements. To achieve these goals, you must have a “mile wide and inch deep” view of the system. Architectural decisions must be made with a thorough understanding of the entire system, including its scope, major functionality, and nonfunctional requirements such as performance.

**iii) Construction Phase:** All remaining components and application features are developed and integrated into the product during the construction phase, and all features are thoroughly tested. In some ways, the construction phase is similar to a manufacturing process in that the emphasis is on managing resources and controlling operations to optimize costs, schedules, and quality. In this sense, the management mindset shifts from intellectual property development during inception and elaboration to product development during construction and transition.

**iv) Transition Phase:** The transition phase's goal is to introduce the software product to the user community. After the product has been delivered to the end user, problems usually arise that necessitate the development of new releases, the correction of some issues, or the completion of previously postponed features. (Software, 2021)

## 2. Gantt Chart

### 2.1. Description

A Gantt chart, which is widely used in project management, is one of the most popular and useful methods of displaying activities (tasks or events) against time. A list of the activities is shown on the left side of the chart, and a time scale is shown along the top. Each activity is represented by a bar, with the position and length of the bar reflecting the activity's start, duration, and end dates. This allow you to see at glance: (Team, 2021)

- What the various activities are
- When each activity begins and ends
- How long each activity is scheduled to last
- Where activities overlap with other activities and by how much
- The start and end date of the whole project

The Gantt chart was produced from draw.io, which is an online platform to create various types of diagrams, for this project.

### 2.2. Gantt Chart

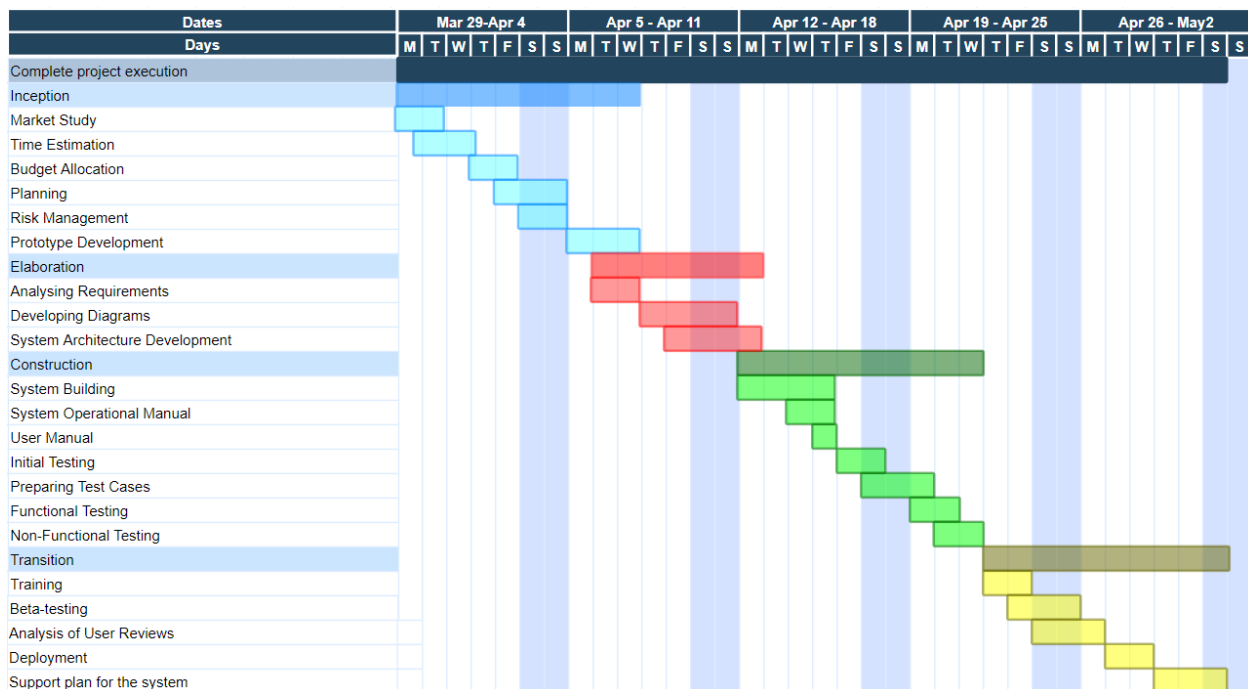


Figure 1: Gantt chart

### 3. Use Case Diagram

#### 3.1. Description

Use case diagrams, also known as behavior diagrams, are commonly used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors). Each use case should produce some observable and valuable result for the system's actors or other stakeholders. (Editors, 2009)

#### 3.2. Use-Case Diagram

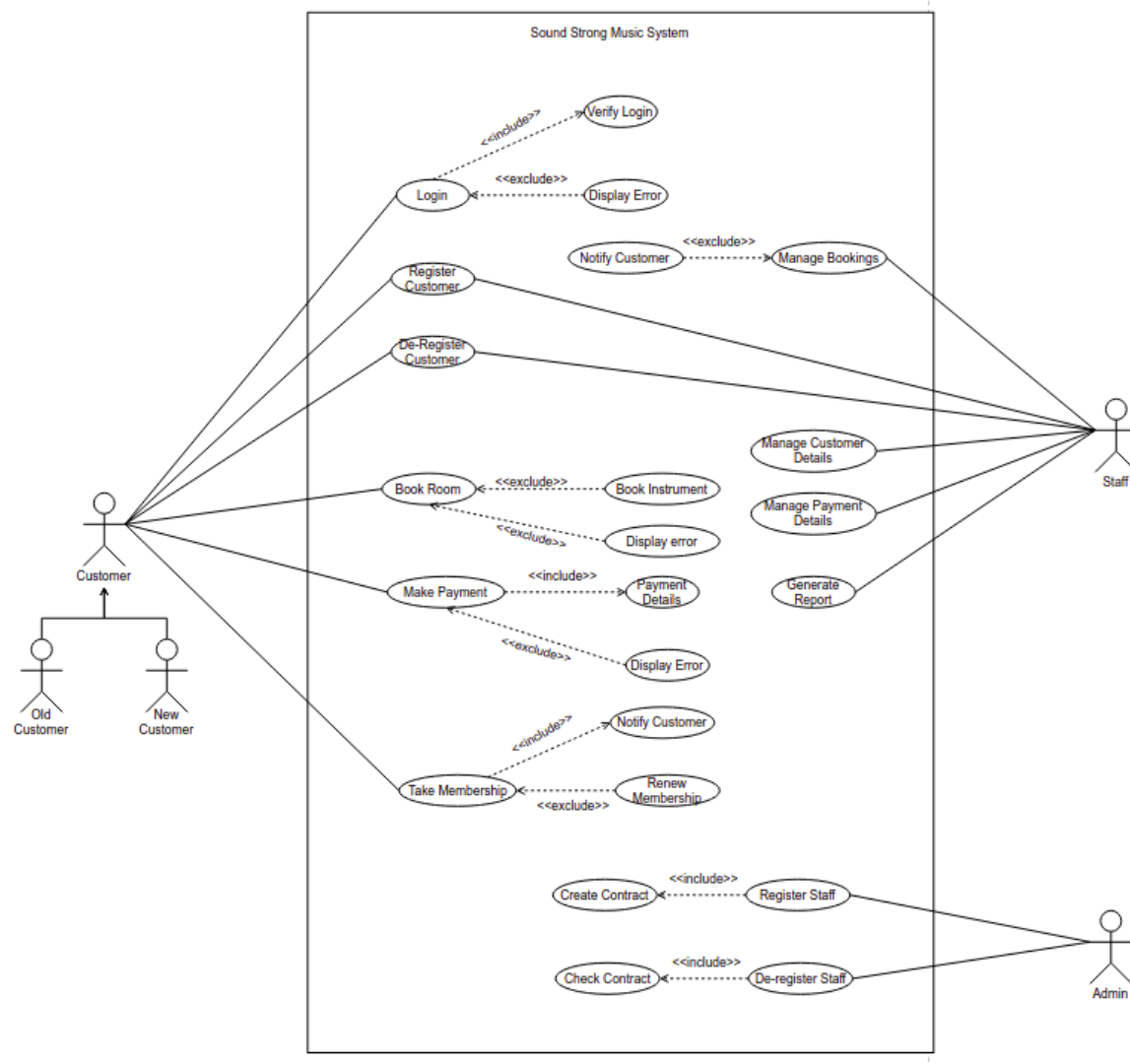


Figure 2: Use Case Diagram



### 3.3. High Level Description

#### i) Login

**Actor:** Customer

**Description:** A customer enters his/her username and password into the system. The system check the validity of customer's login details. If it is valid the customer is welcome to use the system else access won't be allowed and error message is displayed.

#### ii) Register customer

**Actor:** Customer, Staff

**Description:** A customer/staff enter details of a customer to register him/her into the system. The details are stored in the customer details creating a new user which the customer can use to log in.

#### iii) De-register customer

**Actor:** Customer, Staff

**Description:** A customer can request his/her account to be taken down. Also the staff can remove a customer's details from the system on a customer's request or any given condition inappropriate to the terms and conditions of the music store. The customer won't be able to access the account after de-registration.

#### iv) Book Room

**Actor:** Customer

**Description:** A customer books a room by entering booking details asked by the system. After that the customer is required to make payment before the request is sent to staff.

#### v) Book Instrument

**Actor:** Customer

**Description:** If a customer checks instrument box in booking details asked by system in book room use case then customer is also asked to write down name of the instruments needed. After that the customer is required to make payment before the request is sent to staff.

**vi) Take Membership**

**Actor:** Customer

**Description:** A customer chooses a plan for membership. Then the customer is required to make a payment before getting membership.

**vii) Make Payment**

**Actor:** Customer

**Description:** A customer is required to enter his payment details for payment. If the balance required is available it extracts the amount from user's balance and book the rooms or makes the customer a member else if the balance required is unavailable then error message is displayed.

**viii) Manage Bookings**

**Actor:** Staff

**Description:** The staff gets requests from customers and members for booking of room, checks the availability of the rooms/instruments and either accepts or rejects the bookings according to the circumstance.

**ix) Manage Bookings**

**Actor:** Staff

**Description:** The staff gets requests from customers and members for booking of room, checks the availability of the rooms/instruments and either accepts or rejects the bookings according to the circumstance.

**x) Manage Customer Details**

**Actor:** Staff

**Description:** The staff gets the registration details and membership details from the customers and manages them accordingly.

**xi) Manage Payment Details****Actor:** Staff

**Description:** The staff gets payment details from the payments made by the customers and manages the details based on booking and membership payment.

**xii) Notify customer****Actor:** System

**Description:** Once the staff accepts booking a reminder is set to remind the customer half hour before the booking time. It also notifies a member a week before his/her membership expires.

**xiii) Generate Report****Actor:** Staff

**Description:** The staff accumulates all the details relating to a customer and makes a report for the organizations purpose that is sent to the admin.

**xiv) Register Staff****Actor:** Admin

**Description:** It allows the admin to hire a staff to run the organization.

**xv) De-register Staff****Actor:** Admin

**Description:** It allows the admin to fire a staff in case of breach in contract terms and conditions. It also allows the admin to de-register a staff in case the contract expires.

### 3.4. Expanded Level Description

#### i) Register Customer

**Actor:** Customer, Staff

**Purpose:** To create a user for a customer to log in to the system.

**Description:** A customer enter details of a customer to register him/her into the system. In case the customer is unknown to this process then staff registers the details for the customer. The details are stored in the customer details creating a new user which the customer can use to log in.

Actor Action	System Response
1. A customer/staff opens the system. The customer doesn't have a user to log in to so s/he proceeds to register page.	
	2. Opens registration page
3. Required details are filled in.	
	4. Checks the validity of the customer details like username to make it unique
	5. If the details are valid, system stores the details and gives access to log in
	6. Else error message that the username has already been take is displayed to user.

Table 1: Expanded Description of Register Customer

**i) Make Payment****Actor:** Customer, Staff**Purpose:** Record Payment Details**Description:** A customer enter details of payment to gain membership or book a room. The details is recorded by the system which is later managed by the staff.

Actor Action	System Response
1. A customer is taken to payment page to make payment for booking or membership.	
	2. Opens payment page.
3. Required details are filled in.	
	4. Checks the if the Balance required is available.
	5.If the Balance is available payment is made then booking request or membership request is sent along with the details.
	6. If the balance is not available then error message is displayed the use case also ends.
7. After payment is successful staff accepts the booking.	

Table 2: Expanded Description of Make Payment.

## 4. Communication Diagram – Sequence Diagram

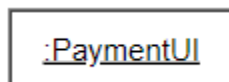
### 4.1. Steps to Draw Communication Diagram

**Step 1:** Select a Use Case Diagram

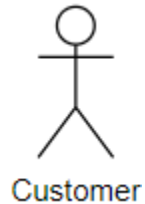
**Step 2:** Analyse the expanded level description of the use case diagram.

**Step 3:** Identify the objects from expanded level description.

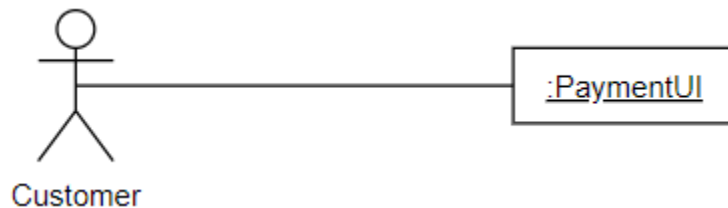
**Step 4:** Objects are represented by a rectangular box and the object is underlined and begins with colon (:) symbol.



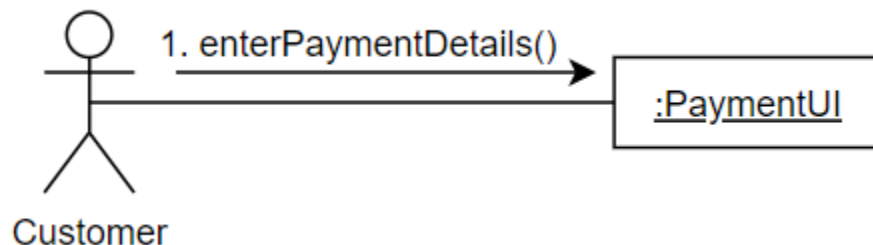
**Step 5:** Actors are represented by stick man.



**Step 6:** Two Objects are connected using a linear line.



**Step 7:** The task carried out by object is represented on top of the line with another straight line point toward the object that the task is directed to.



## 4.2. Communication Diagram

A communication diagram (also known as a collaboration diagram in UML 1.x) is a type of UML interaction diagram that depicts interactions between objects and/or parts (represented as lifelines) through the use of sequenced messages in a free-form arrangement. A communication diagram corresponds to (can be converted to/from or replaced by) a simple sequence diagram that lacks structuring mechanisms such as interaction uses and combined fragments. It is also assumed that message overtaking (in which the order of reception differs from the order of sending of a given set of messages) will not occur or is irrelevant. (Editors, 2009)

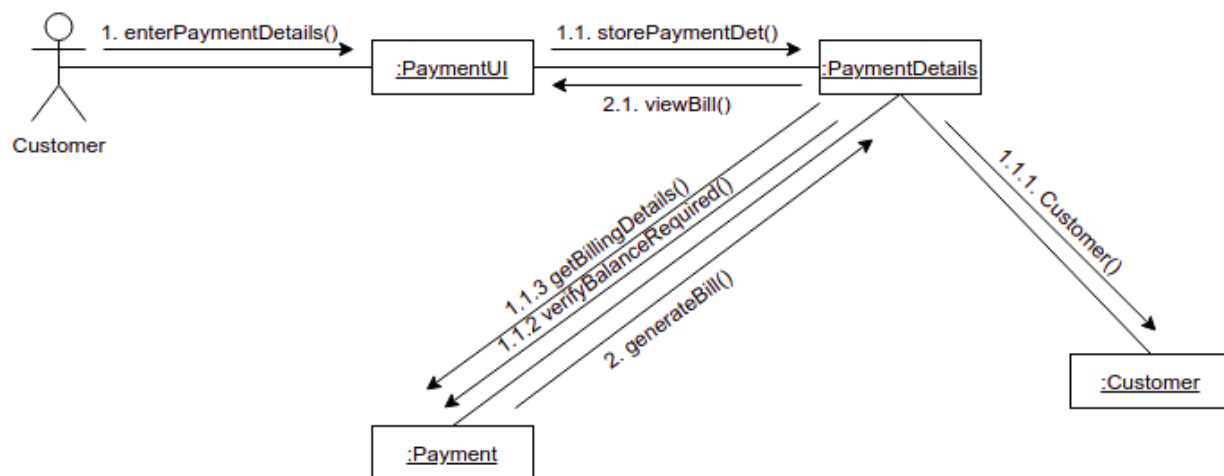


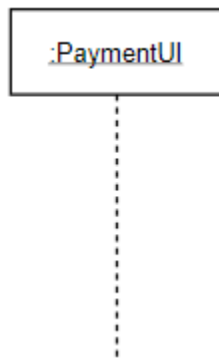
Figure 3: Communication Diagram

### 4.3. Steps to Draw Sequence Diagram

**Step 1:** Analyse the expanded level description of a use case diagram.

**Step 2:** Identify the objects from expanded level description.

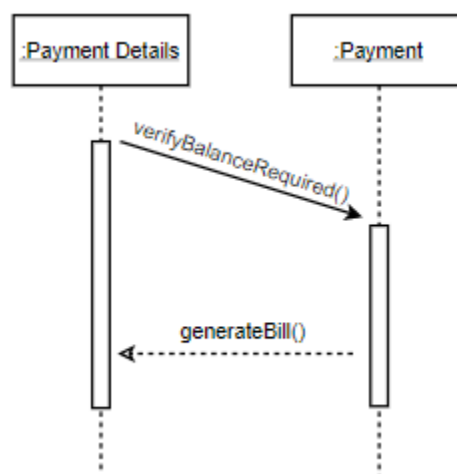
**Step 3:** Objects are represented by a rectangular box and the object is underlined and begins with colon (:) symbol. A horizontal dotted line is drawn under objects to represent passing of time.



**Step 4:** Actors are represented by stick man.

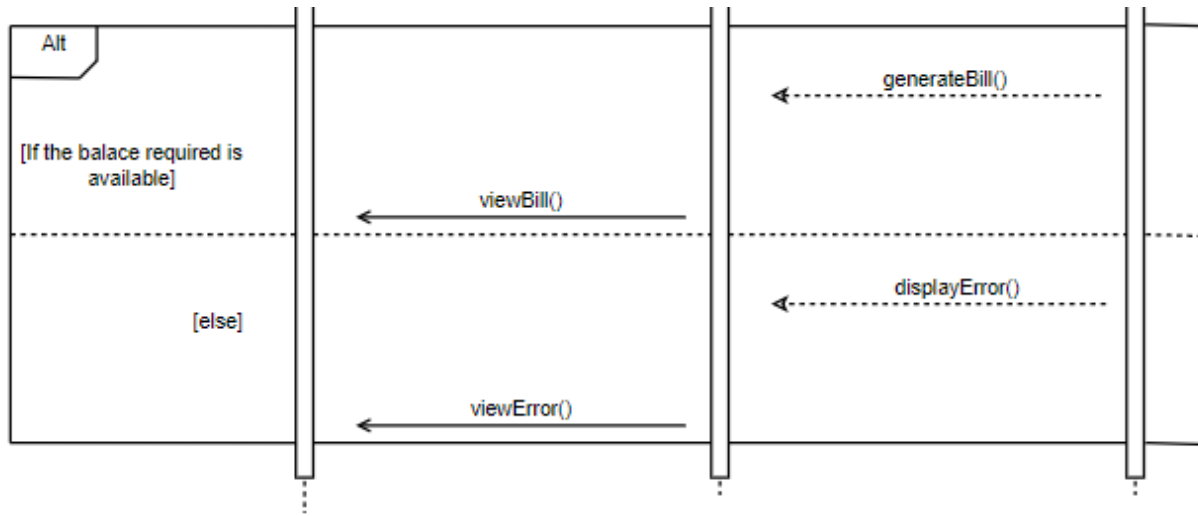


**Step 5:** The message sent from an object to another is shown by an arrow line pointed towards the object it is sent to and the return message is represented by dash arrow.





**Step 6:** Alternative is shown inside a rectangular box.



## 4.4. Sequence Diagram

UML Sequence Diagrams are interaction diagrams that show how operations are performed. They document the interaction of objects in the context of a collaborative effort. Sequence Diagrams are time focused, and they visually represent the order of the interaction by using the vertical axis of the diagram to represent time, what messages are sent, and when. (Paradigm, 2021)

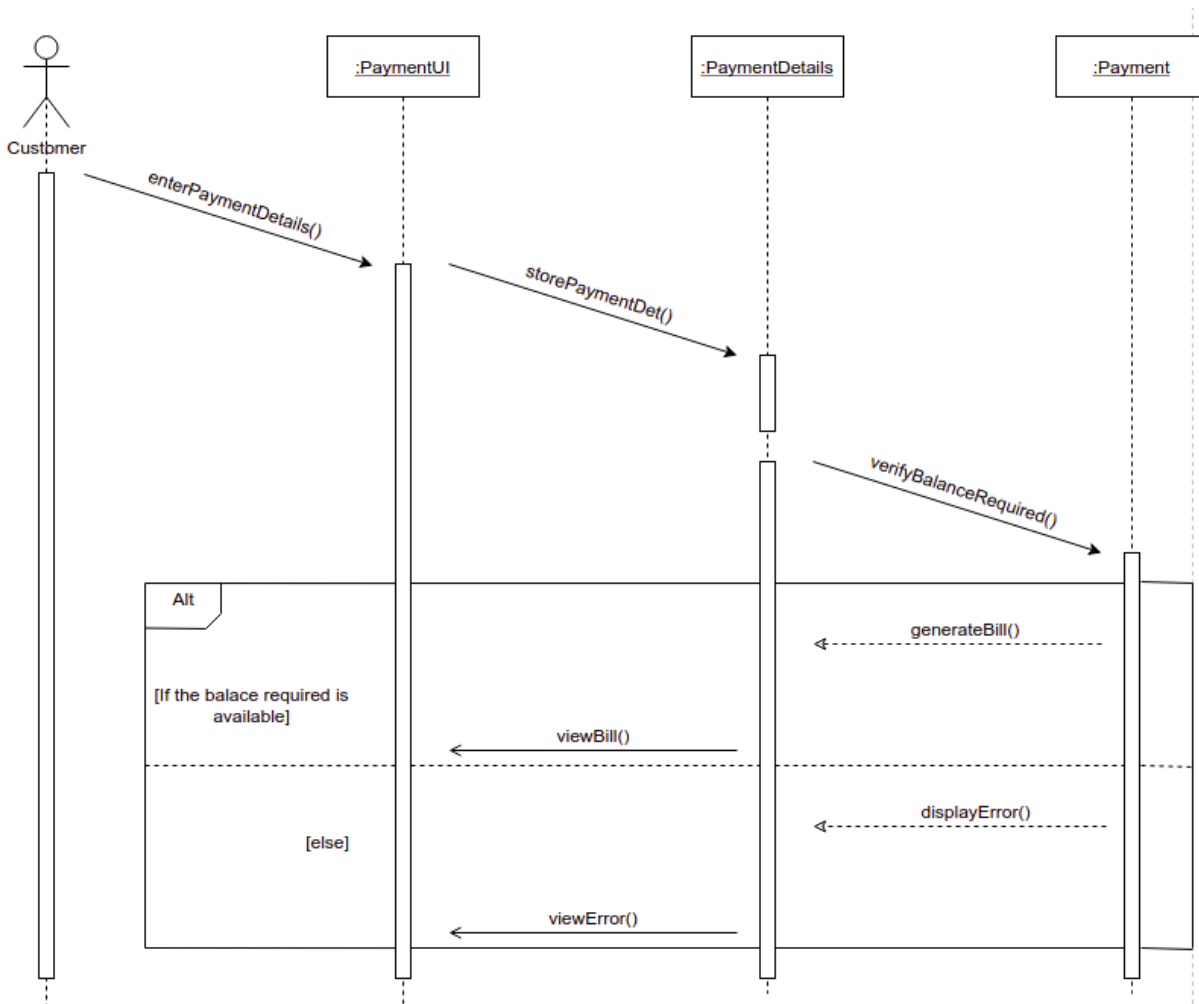


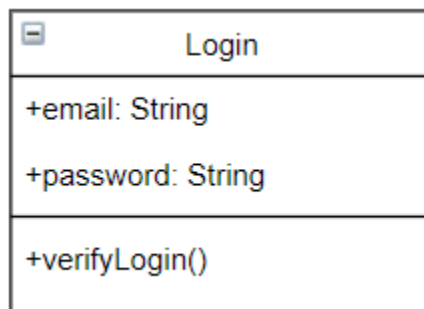
Figure 4: Sequence Diagram

## 5. Class Diagram

### 5.1. Steps to Draw Class Diagram

**Step 1:** Identify domain classes then draw all them all first.

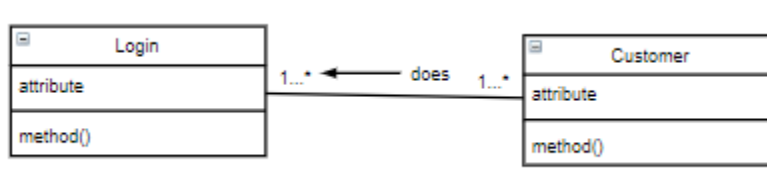
**Step 2:** Classes are represented by rectangular box with attributes and methods inside it separated by a line.



**Step 3:** Related classes are interconnected with a line. Each relation is represented by a different sign.



**Step 4:** Verbs are written over the relation lines to specify the relation between classes clearly.



**Step 5:** Cardinality is represented by writing numbers above relation lines.

Only one: 1

Zero or One = 0...1

Zero or More = 0...\*

One or More = 1...\*

Range = 5...7 (5 to 7)

Specific Number = 3, 6 (3 or 6)

**5.2. Use Cases and Domain Classes**

S.N	Use Cases	Domain Classes
1	Login	Customer, Login
2	Register Customer	Customer, Register_Customer, Staff
3	Book Room	Customer, Book_Room, Staff
4	Book Instrument	Customer, Book_Instrument, Staff
5	Take Membership	Customer, Take_Membership
6	Make Payment	Customer, Payment, Bill
7	Generate Report	Staff, Report, Admin
8	Manage Booking	Staff, Manage_Booking
9	Notify Customer	Reminder, Customer
10	Register Staff	Admin, Register_Staff
11	De-register Staff	Admin, De-register_Staff

Table 3: Use Cases and Domain Classes

### 5.3. Class Diagram

Class diagram is a static diagram. It represents an application's static view. A class diagram is used not only for visualizing, describing, and documenting various aspects of a system, but also for building executable code for a software application. A class diagram describes a class's attributes and operations, as well as the constraints imposed on the system. Because they are the only UML diagrams that can be mapped directly to object-oriented languages, class diagrams are widely used in the modeling of object-oriented systems. A class diagram is a visual representation of a collection of classes, interfaces, associations, collaborations, and constraints. It is also referred to as a structural diagram. (Point, 2021)

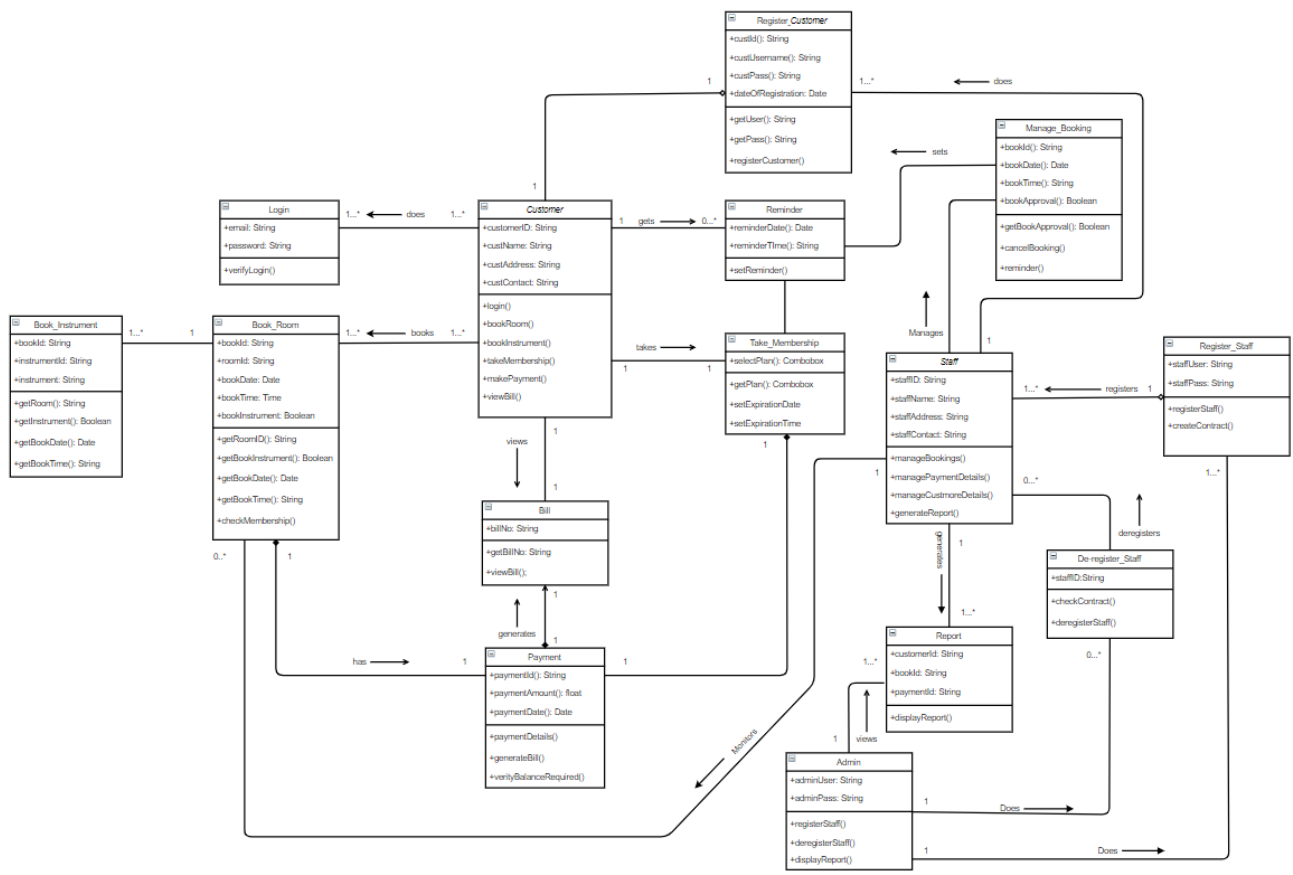


Figure 5: Class Diagram

## 6. Explanation

All the required diagram have been made. Now, we start the development of the software UI. The wireframes and prototypes will be made with the designing tools like Balsmiq and Adobe XD respectively. When the prototypes were finalized then the development of the UI was initiated. Angular JS was used for the framework of the UI.

With the prototype that was finalized, the first line code for the software was written. Angular JS will be used for the framework. AngularJS is a set of tools for creating the best framework for application development. It is completely extensible and integrates well with other libraries. Every feature can be customized or replaced to meet your specific development workflow and feature requirements. Angular JS uses data binding. Data-binding is a method of automatically updating the view when the model changes, as well as the model when the view changes. This is fantastic because it removes DOM manipulation from the list of things to be concerned about. The code editor that will be is used Visual Studio Code.

Software testing is the process of comparing software to requirements gathered from users and system specifications. Testing occurs at the phase level of the software development life cycle or at the module level of program code. Validation and verification are components of software testing. Rainforest was used to do the testings. Rainforest runs tests in minutes across its network of automated virtual machines as well as its community of on-demand manual QA testers. It is designed from the ground up to work seamlessly with your existing CI/CD pipeline. Using our API, CLI, or CircleCI orb

After the completion of development of software, user manuals were made so that people would be able to know how the software works and beta-testing can also be done

Beta testing is a type of user acceptance testing in which the product team distributes a nearly finished product to a group of target users in order to assess product performance in the real world. The reviews from the targeted group would be retrieved in a certain interval of time. The bugs would be fixed and improvement to the software will be done in accordance to the reviews.

After the beta testing the software will be released to use for the Sound strong company to use. Even after releasing of the software the work is not done. The review will be open for the staffs and customers to report improvement. The maintenance of software will also be important. Its

main purpose is to update software as time passes by and updates to programming languages happens. This will fix the short comings that was unfixable in the past possible with the update. The maintenance usually depends on the type of contract.

## 7. Prototype

According to Cambridge Dictionary, “Prototype is the first example of something, such as a machine or other industrial product, from which all later forms are developed.” The prototypes that are given below are the examples of a software for Sound Strong Music, from which the UI (User Interface) will be developed which will later transition into fully functional software.

### 7.1. Customer

Registration

http://soundstrongmusic.com/registration

Sound Strong Music Login

Full-Name: \_\_\_\_\_

Address : \_\_\_\_\_

Contact : \_\_\_\_\_

Username: \_\_\_\_\_

Create Password: \_\_\_\_\_

Confirm Password: \_\_\_\_\_

☐ Agree Terms and Conditions

Register

Figure 6: Register Customer



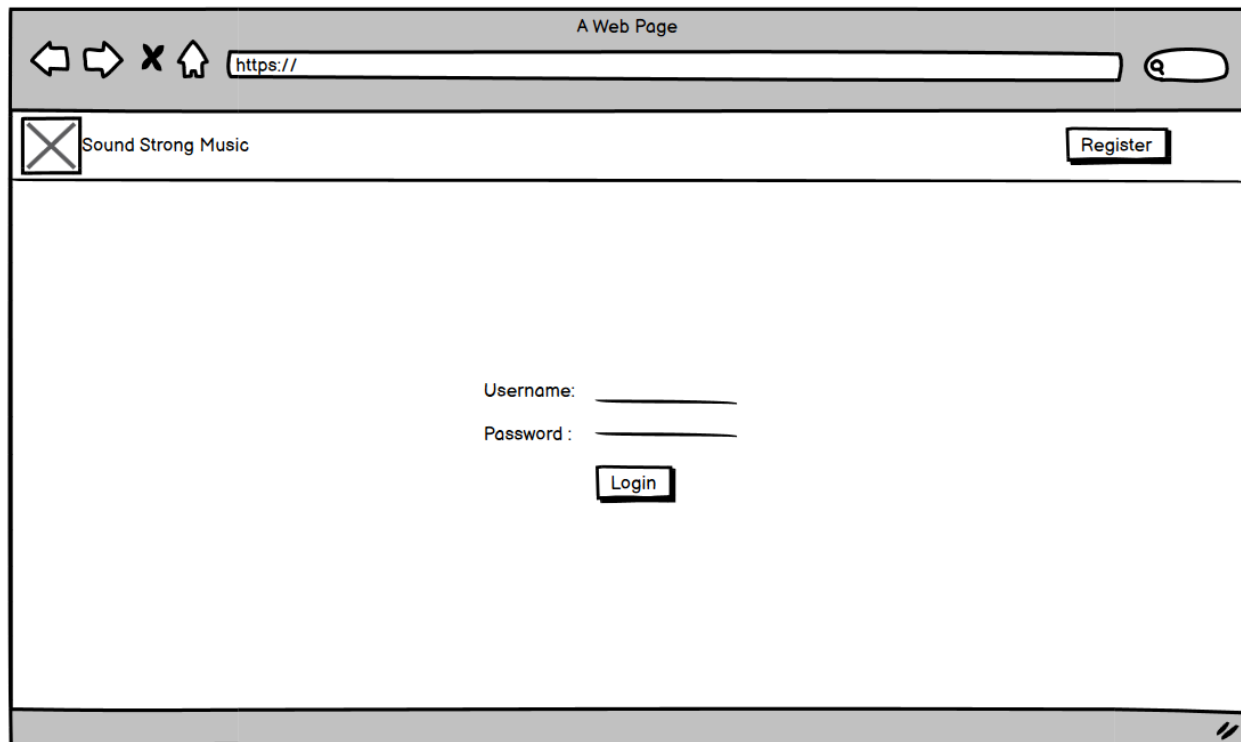


Figure 7: Customer Login

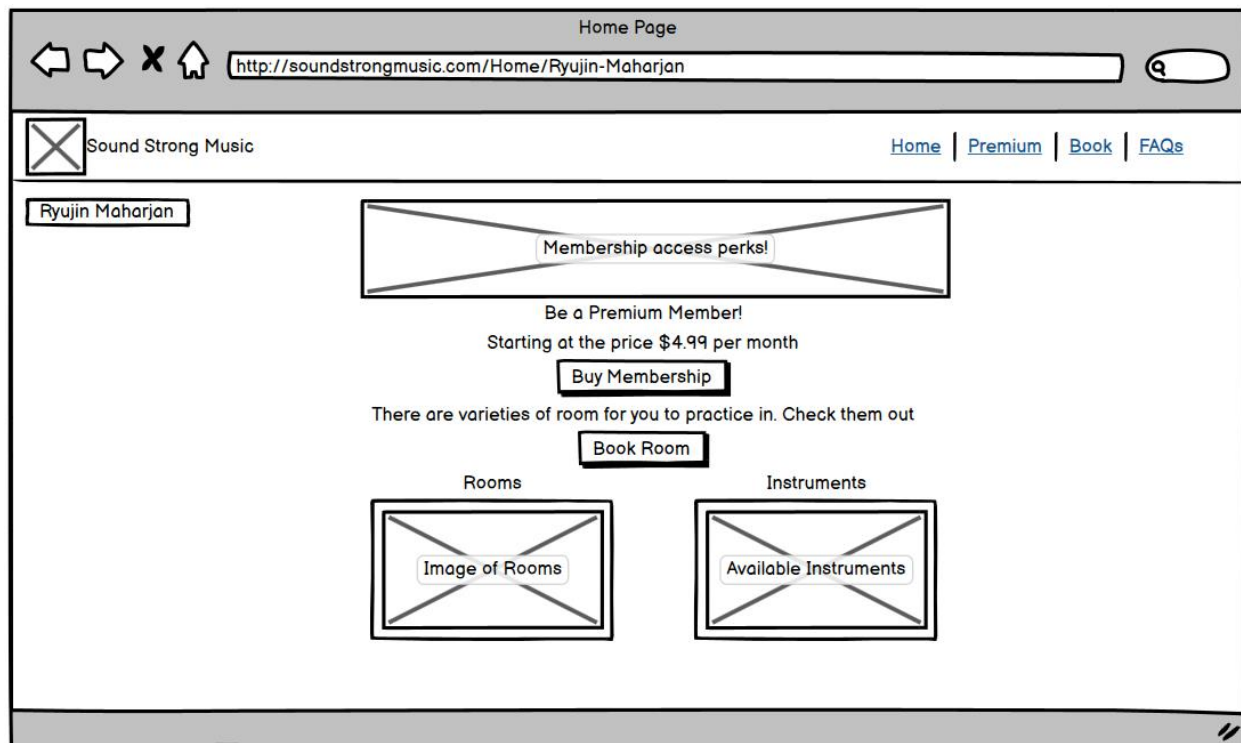


Figure 8: Customer Home

Book Room

http://soundstrongmusic.com/Book

Sound Strong Music

Home | Premium | Book | FAQs

Ryujin Maharjan

Book Room

Try Premium Membership!!!  
\$1.99 OFF while booking  
[Buy Membership](#)

Room: \_\_\_\_\_  
Date:  /  /   
Time:  :  :  am  
Instrument: ☐ Needed

List of Instruments

Room Preview

Book Room Cancel

Figure 9: Customer Book Room

Payment

http://soundstrongmusic.com/Payment

Sound Strong Music

Home | Premium | Book | FAQs

Ryujin Maharjan

Book Details

Customer: Ryujin Maharjan  
Room: Atalantic  
Date: 04/30/2021  
Time: 04:00pm  
Instrument: Piano

Payment

Total Amount: \$15 Discount: \$0.00  
Card No: \_\_\_\_\_  
PIN: \_\_\_\_\_

Atlantic Room Preview

Pay Cancel

Figure 10: Customer Payment

Membership

http://soundstrongmusic.com/membership

Sound Strong Music

Home | Premium | Book | FAQs

Ryujin Maharjan

## Choose a plan

<b>1 Month Plan</b> <b>\$50</b> -\$1.99 per Booking Premium Instruments	<b>Half-Year Plan</b> <b>\$250</b> -\$4.99 per Booking +Access to Record Station	<b>1 Year Plan</b> <b>\$400</b> -\$9.99 per Booking Access to Everything
--	---	---

Premium Access Stuff

Payment

Price: \$400  
Card No: \_\_\_\_\_  
PIN: \_\_\_\_\_

Pay Cancel

Figure 11: Customer Membership

## 7.2. Staff

The screenshot shows a web browser window titled "Staff Login". The address bar displays the URL "http://soundstrongmusic.com/staff-login". The browser's title bar includes navigation icons (back, forward, stop, home) and a search icon. The page header features the "Sound Strong Music" logo on the left and an "Admin Login" button on the right. The main content area contains a login form with the following elements:

- A "Username:" label followed by a text input field.
- A "Passcode:" label followed by a text input field.
- A "Login" button centered below the input fields.

The browser window has a standard gray border with a double-slash icon in the bottom right corner.

Figure 12: Staff Login

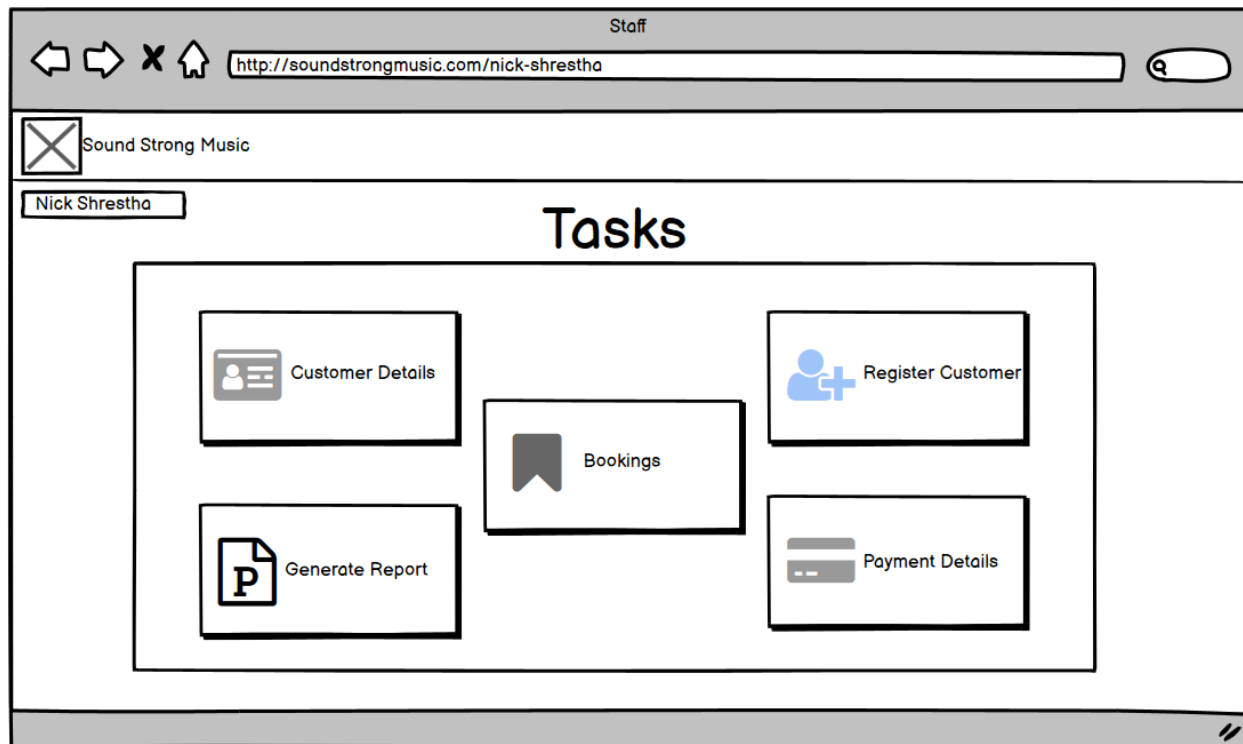


Figure 13: Staff Home

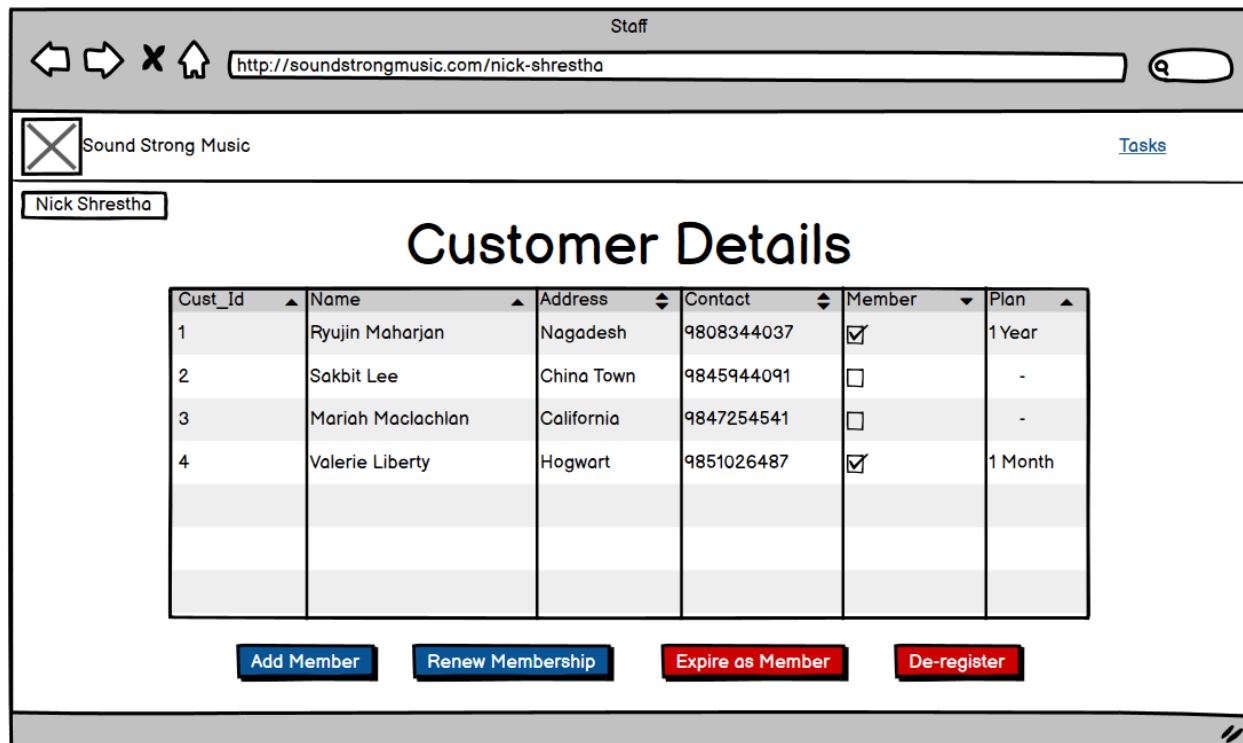


Figure 14: Staff Customer Details

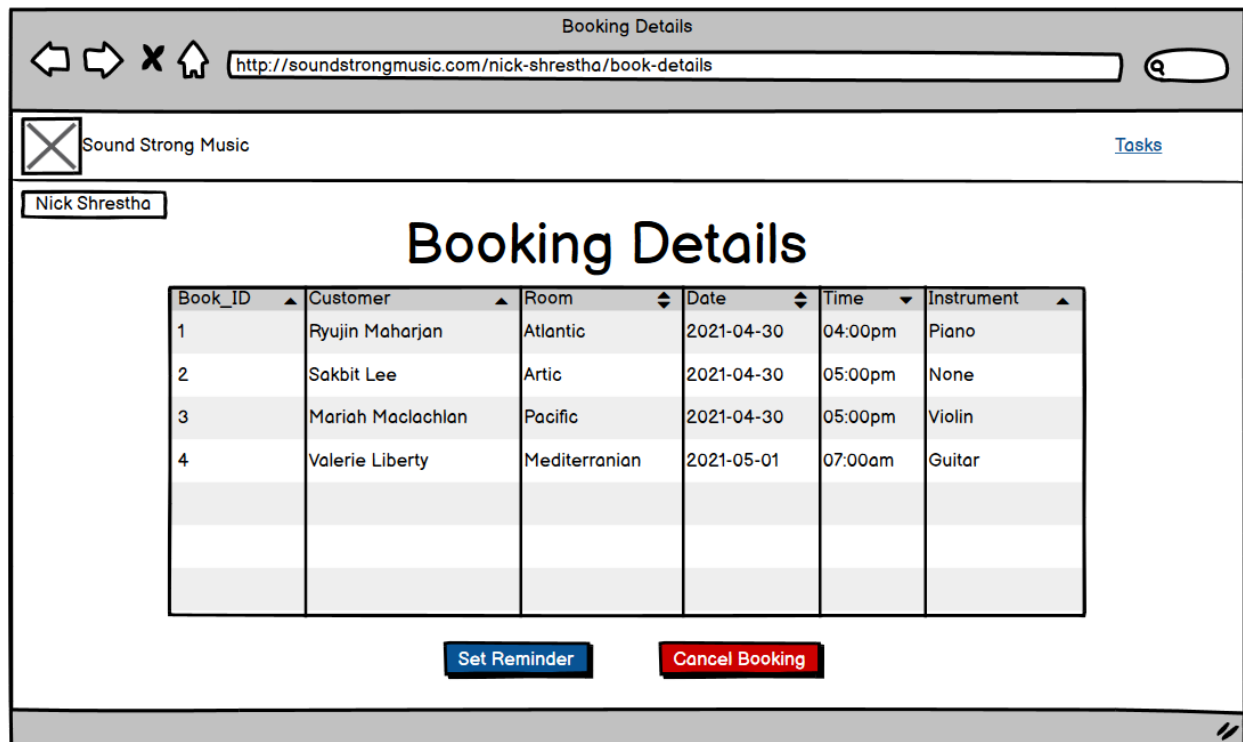


Figure 15: Staff Booking Details

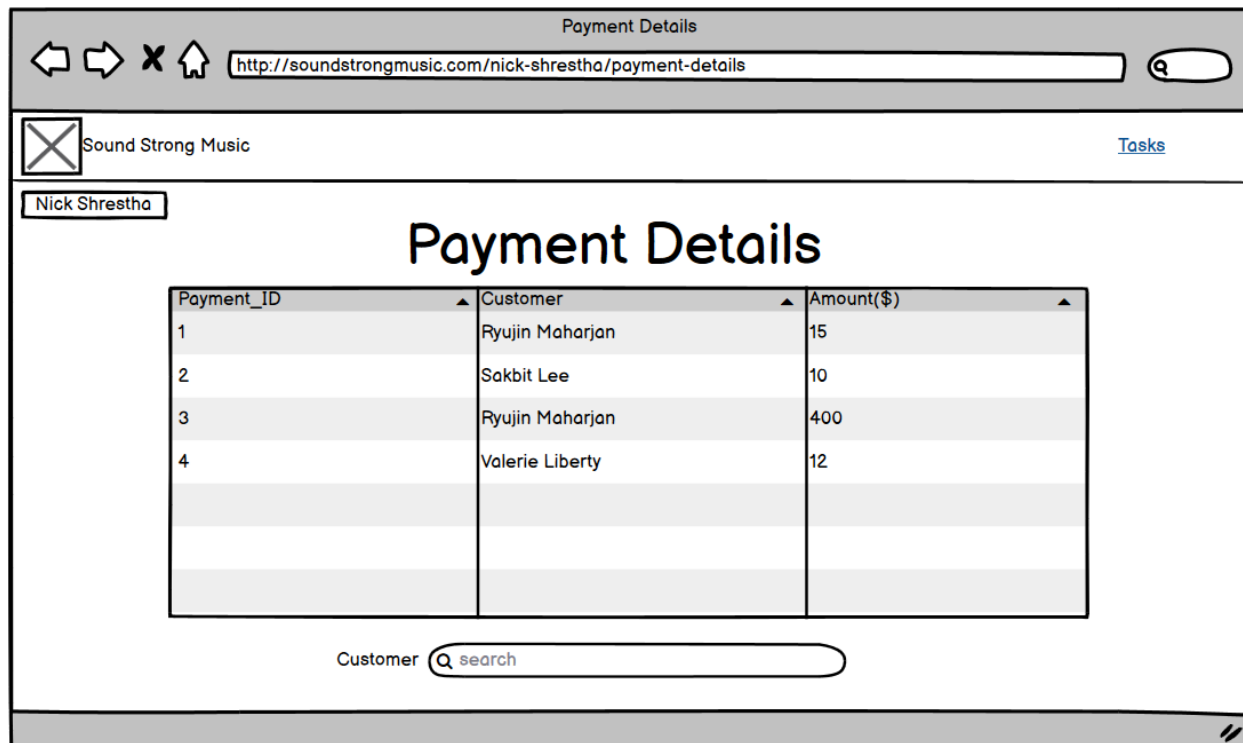


Figure 16: Staff Payment Details

Generate Report

http://soundstrongmusic.com/nick-shrestha/generate-report

Sound Strong Music [Tasks](#)

Nick Shrestha

## Generate Report

Customer Name: \_\_\_\_\_

Address : \_\_\_\_\_

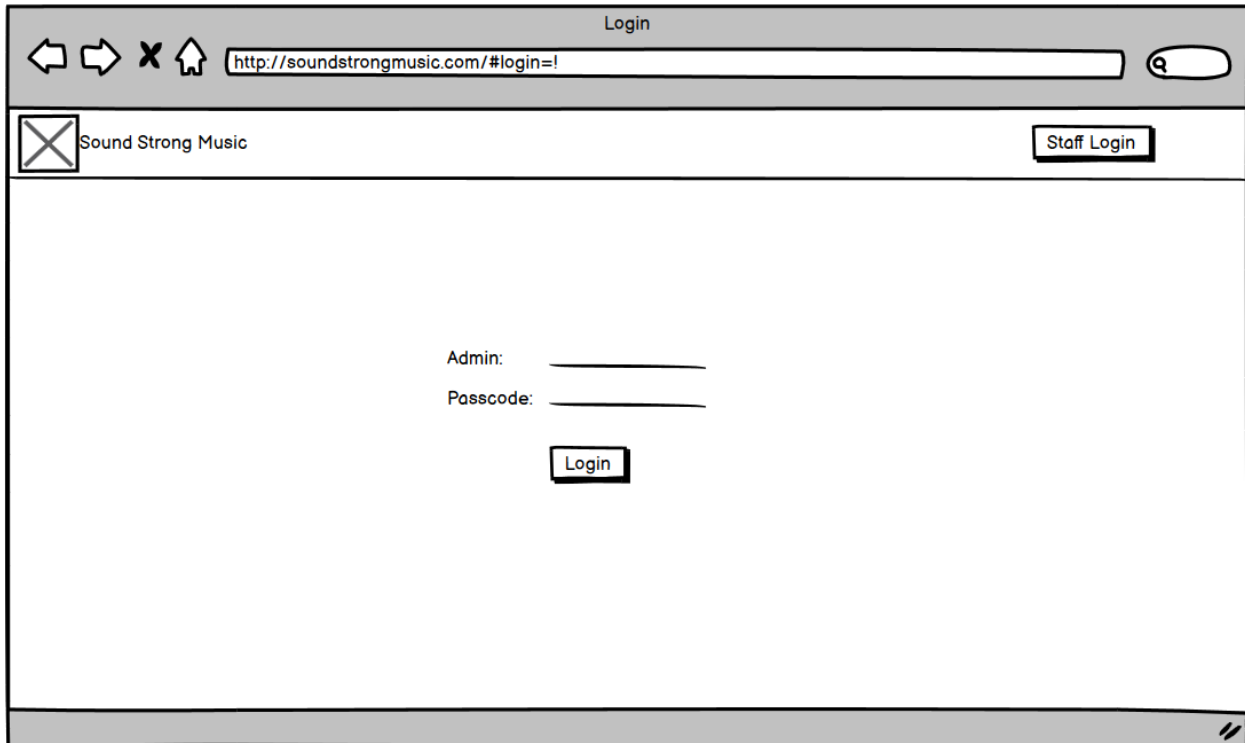
Contact: \_\_\_\_\_

Report:

\_\_\_\_\_

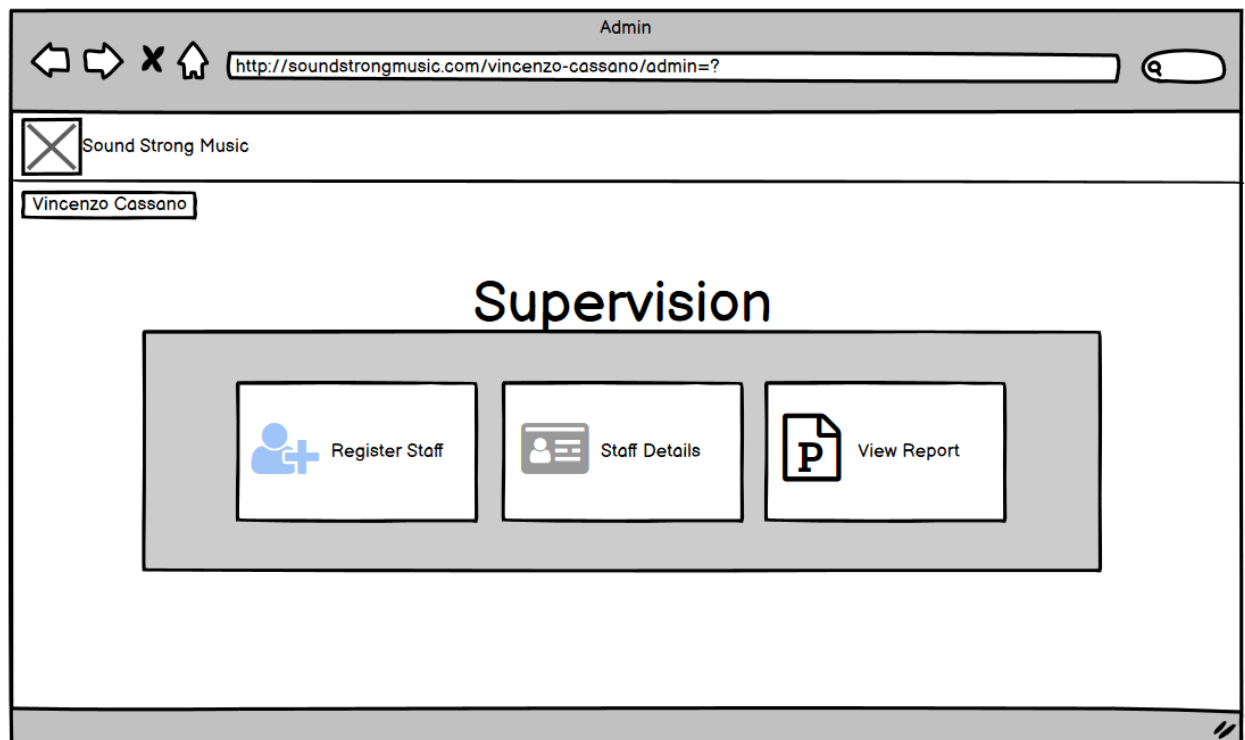
Figure 17: Staff generate Report

### 7.3. Admin



A screenshot of a web browser window titled "Login". The address bar shows the URL "http://soundstrongmusic.com/#login=!". The page header includes a "Sound Strong Music" logo and a "Staff Login" button. The main content area contains a login form with two input fields: "Admin:" and "Passcode:". Below these fields is a "Login" button.

Figure 18: Admin Login



A screenshot of a web browser window titled "Admin". The address bar shows the URL "http://soundstrongmusic.com/vincenzo-cassano/admin=?". The page header includes a "Sound Strong Music" logo and a "Vincenzo Cassano" button. The main content area features a large heading "Supervision" above a container with three buttons: "Register Staff" (with a person icon), "Staff Details" (with a person icon), and "View Report" (with a document icon).

Figure 19: Admin Home



Register Staff

Sound Strong Music

Vincenzo Cassano

## Register Staff

Name: \_\_\_\_\_

Address: \_\_\_\_\_

Contact: \_\_\_\_\_

Work-Time: \_\_\_\_\_

Contract Length: \_\_\_\_\_

Issue-Date:  /

☒ I accept Contract terms

Signature

Figure 20: Admin Register Staff

Staff Details

Sound Strong Music

Vincenzo Cassano

Customer Name

## Staff Details

Staff_ID	Staff Name	Address
s01	Nick Shrestha	Gatthaghar
s02	Sakbit Lee	Worcester
s03	Mariah MacLachlan	Louisiana
s04	Valerie Liberty	Texas

Figure 21: Admin Staff Details

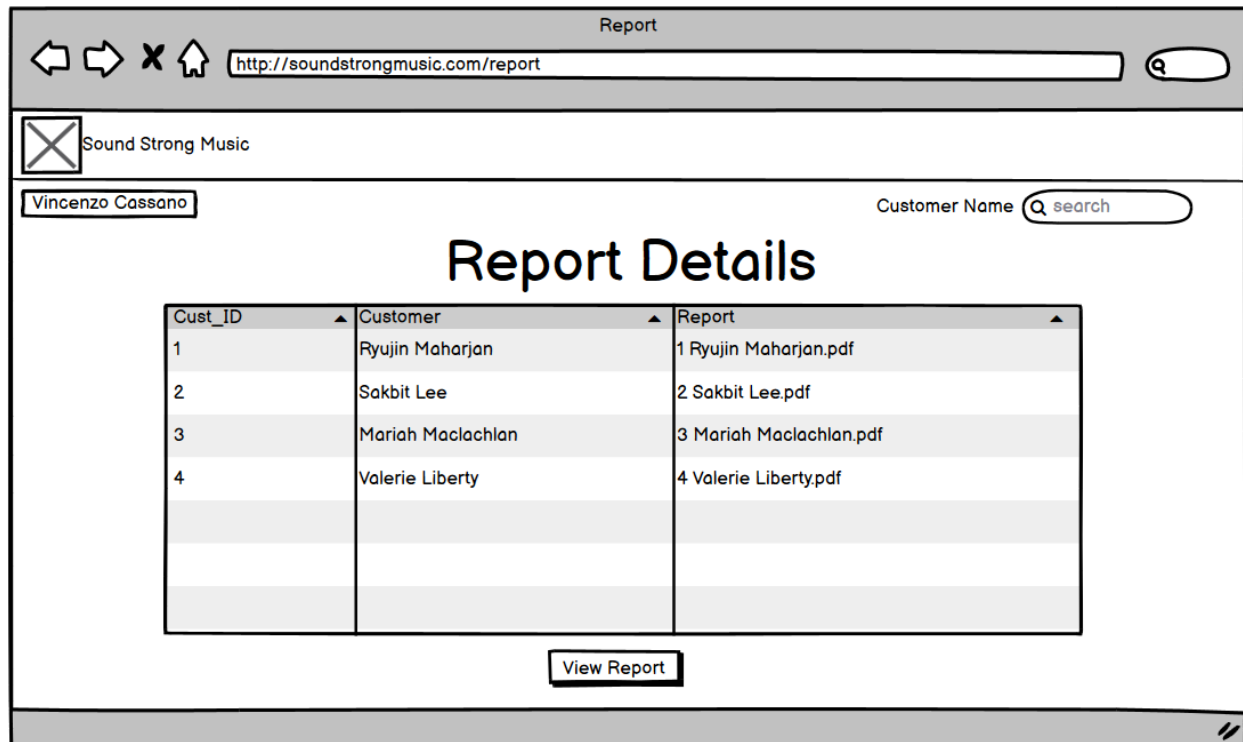


Figure 22: Admin Report Details

## 8. Conclusion

The coursework was finally fulfilled after a whole month of development. The coursework had a whole new level of requirement with different diagrams. Understanding how to make the diagrams was much easier than actually making the diagrams. Tutorials provided by the teachers and online books, websites and reports were used to broaden the knowledge about software development life cycle. It was found that a lot goes into software developing than just coding. Coding is merely just a part of software developing to make the software function for the purpose it was built for. The diagrams that were developed such as use case diagram, class diagram and other supports the whole coding of software development. The coursework was a lot to handle at first but as time went by the software development progressed and finally the coursework was completed. It was a headache to begin with but learning the diagrams was profitable and making the prototype was fun. Completing the coursework has its own advantage for gaining marks but as a side perk it has given a learning experience on systematically building a software with RUP methodology which is used in real time.

## References

- Code, V. S., 2021. *Visual Studio Code*. [Online] Available at: <https://code.visualstudio.com/docs/editor/whyvscode#:~:text=For%20serious%20coding%2C%20you'll,tough%2C%20the%20tough%20get%20debugging.> [Accessed 28th April 2021].
- Editors, U., 2009. *UML Communication Diagrams Overview*. [Online] Available at: <https://www.uml-diagrams.org/communication-diagrams.html> [Accessed 27 April 2021].
- Editors, U. D., 2009. *Use case diagram are UML diagrams*. [Online] Available at: <https://www.uml-diagrams.org/use-case-diagrams.html> [Accessed 27 April 2021].
- Paradigm, V., 2021. *What is Sequence Diagram?*. [Online] Available at: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/> [Accessed 27 April 2021].
- Point, T., 2021. *UML - Class Diagram*. [Online] Available at: [https://www.tutorialspoint.com/uml/uml\\_class\\_diagram.htm](https://www.tutorialspoint.com/uml/uml_class_diagram.htm) [Accessed 27 April 2021].
- Software, R., 2021. *Microsoft Word - TP026B.doc*. [Online] Available at: [https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251\\_bestpractices\\_TP026B.pdf](https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf) [Accessed 27 April 2021].
- Team, G. C., 2021. *What is Gantt Chart?*. [Online] Available at: <https://www.gantt.com/> [Accessed 27 April 2021].