



Addis Ababa Institute of Technology  
School of Information Technology and  
Engineering  
Department of IT/SE Eng.

## Agelgil online Hotel Reservation System Software Design Specification

### Team Members

Name	Section	ID
1. Abraham Atlaw	3	UGR/3423/12
2. Aklile Yilma	3	UGR/7107/12
3. Henok Mekuanint	1	UGR/2272/12
4. Joshua Tesfaye	2	UGR/0359/12
5. Samuel Girma	1	UGR/6303/12
6. Temesgen zewde	2	UGR/3848/12

Advisors: Ms. Nuniyat Kifle

Date: Jan 5, 2021

## Revision History

Date	Description	Author	Comments
Jan 2,2022	Version 1	All team members	Initial draft
Jan 4, 2022	Version 2	All team members	First Revision
Jan 5, 2022	Version 3	All team members	Final draft

## Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Signature	Printed Name	Title	Date
	Ms. Nuniyat Kifle	.	

## Table of Contents

List of Tables.....	i
List of figures.....	ii
Definitions, Acronyms, Abbreviations.....	iii
1. Introduction.....	1
1.1 Purpose.....	1
1.2 General Overview.....	1
1.3 Development Methods & Contingencies.....	2
2. System design model.....	3
2.1 Subsystem decomposition.....	3
2.2 Hardware/software mapping.....	4
3. Object Model.....	5
3.1 Class Diagram.....	5
3.2 Sequence Diagram.....	6
4. Detailed Design.....	18
References.....	33

## List of Tables

Table 4.1 Attribute Description for User Class .....	18
Table 4.2 Operation Description for User Class .....	19
Table 4.3 Attribute Description for Hotel .....	21
Table 4.4 Operation Description for Hotel Class.....	22
Table 4.5 Attribute Description for Location .....	23
Table 4.6 Operation Description for location class .....	23
Table 4.7 Attribute Description for Service .....	24
Table 4.8 Operation Description for Service .....	26
Table 4.9 Attribute Description for Order .....	27
Table 4.10 Operation Description for Order Class .....	27
Table 4.11 Attribute Description for CartItem .....	28
Table 4.12 Operation Description for Cart Item .....	29
Table 4.13 Attribute Description for Cart .....	29
Table 4.14 Operation Description for Cart Class.....	29
Table 4.15 Attribute Description for CustomerModel Class .....	30
Table 4.16 Operation Description for Customer Class .....	31
Table 4.17 Attribute Description for Admin Class .....	32
Table 4.18 Operation Description for Admin Class .....	32

## List of figures

Figure 1.1 The Generalized 3 tier structure for Agelgil online reservation System .....	2
Figure 2. 1 Agelgel subsystem decomposition	
(UML diagram, layers are shown as UML package) .....	3
Figure 2.2 Agelgel Deployment Diagram.....	4
Figure 3.11 Class diagram .....	5
Figure 3.21 Sequence diagram for Sign up .....	6
Figure 3.22 Sequence diagram for Login .....	7
Figure 3.23 Sequence diagram for Searching Hotel .....	8
Figure 3.24 Sequence diagram for choosing Hotel .....	9
Figure 3.25 Sequence diagram for Selecting Hotel .....	10
Figure 3.26 Sequence diagram for Personal details .....	11
Figure 3.27 Sequence diagram for Making payments .....	12
Figure 3.28 Sequence Diagram for Cancellation of Order .....	13
Figure 3.29 Sequence Diagram for Hotel Sign Up .....	14
Figure3.30 Sequence Diagram for Hotel Login .....	15
Figure 3.31 Sequence Diagram for Updating the Status of the hotel .....	16
Figure 3.32 Sequence Diagram for User Log Out .....	17
Figure 4.1 User Model .....	18

<b>Figure 4.2 Hotel Model .....</b>	<b>19</b>
<b>Figure 4.3 Location class .....</b>	<b>23</b>
<b>Figure 4.4 Service class .....</b>	<b>24</b>
<b>Figure 4.5 Order Class .....</b>	<b>27</b>
<b>Figure 4.6 CartItem class .....</b>	<b>27</b>
<b>Figure 4.7 Cart class .....</b>	<b>28</b>
<b>Figure 4.8 Customer Model Class .....</b>	<b>30</b>
<b>Figure 4.9 Admin Model Class .....</b>	<b>31</b>

## **Definitions, Acronyms, Abbreviations**

**UML- Unified modeling language**

**CSS- Cascading Style Sheet**

**HTML- Hyper text markup language**

## **1. Introduction**

### **1.1 purpose**

This SDS document will define the design of the Agelgil online reservation system. It contains specific information about the expected input, output, classes and functions. The SDS will break down the project into components to describe in detail what the purpose of each component is and how it will be implemented. The interaction between different classes and functions will be briefly discussed

### **1.2 General overview**

Generally Agelgil provides an online hotel reservation system for the customer and service promotion as well offering services for the customer by the hotels. The distinct features allow customers to

- Search from different hotels using filters such as quality, location, price and rating.
- Select the preferred hotel then make payments online.
- Book a room or get different other types of services provided by the hotel.
- Reserve table, gym memberships and other services provided by the hotel.

The system also provides several features for the hotel such as it allows

- Hotels to create account and post the services they have.
- Hotels to verify payments made by the user.
- Hotels to see the orders made by the user.
- Hotels to get comments and ratings from the user side.

Agelgil online hotel reservation system will use the three tier architecture design approach. Three tier architecture is a client server architecture in which the functional process, logic, data access, computer data storage and user interface are developed and maintained as independent modules on separated platforms.

#### **The presentation tier**

is the top most level of the web application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand. Our presentation tier is built with HTML5,Cascading style sheet(CSS) and JavaScript is deployed to a computing device through a web browser or a web based application.

#### **The logic or Application tier**

this layer coordinates the application, process commands, make logical decisions and evaluations for example when the customer or the hotel tries to log in into the desired account this layer validated whether and makes a logical decision depending on the entered password and user name entered. It also moves and process data between the presentation tier



and Data tier for example when the user tries to search different hotels this tier communicates with the database located at the Data tier and retrieves the desired result.

## Data tier

Here the information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing database.



Figure 1.1 The Generalized 3 tier structure for Agelgil online reservation System

## 1.3 Development method and Contingencies

The project will follow a standard object oriented programming approach. The front end will be implemented using HTML CSS and JavaScript frame works. For application tier Java & spring boot will be implemented. We will use the staged delivery Software life cycle during the development of the web application. We have used pencil to create the UI design for our web application.

The project might face some contingencies during the development of the back end. Changes are made accordingly the change management system.

## 1. System Architecture

### 2.1 Subsystem decomposition

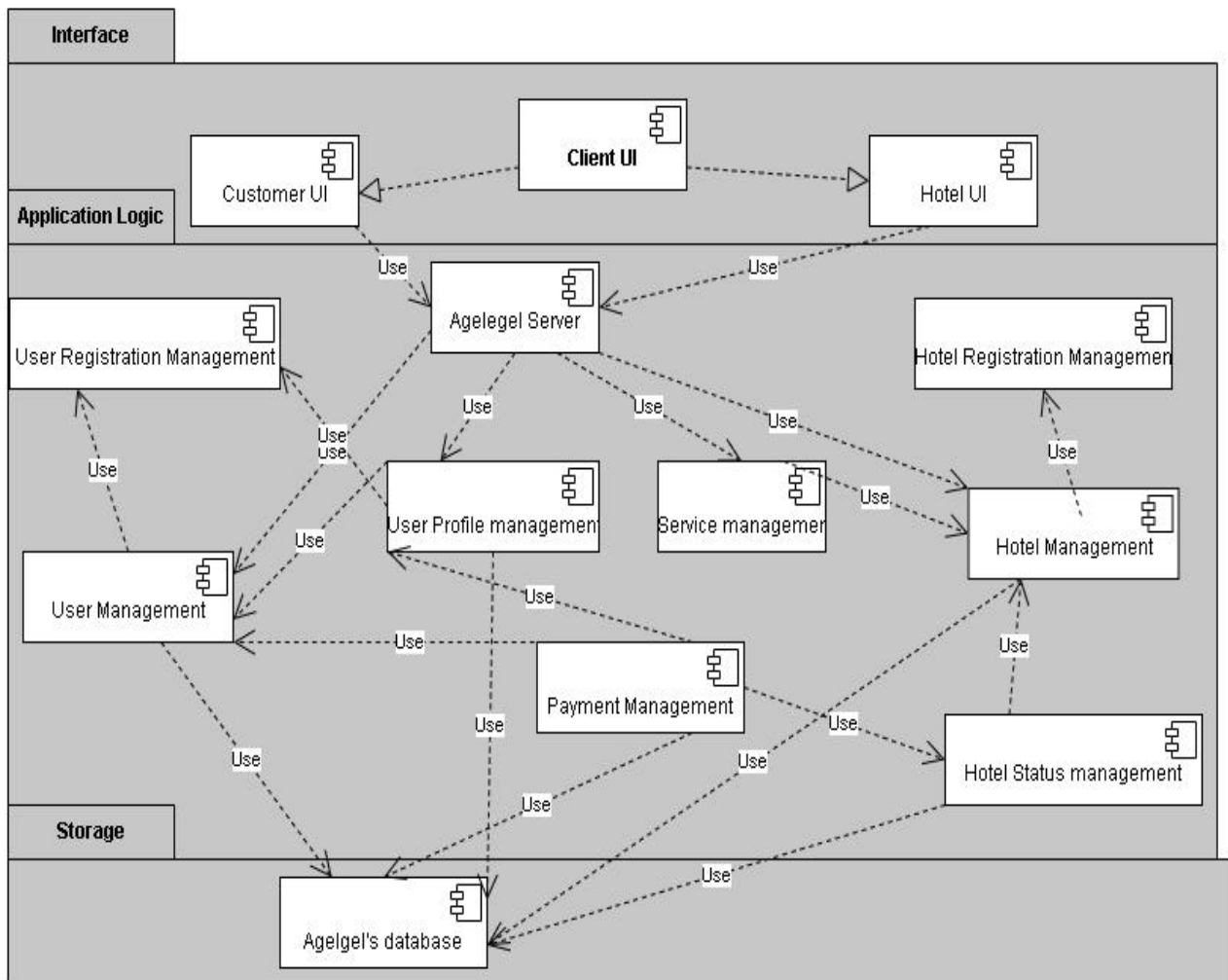


Figure 2. 1 Agelgil subsystem decomposition (UML diagram, layers are shown as UML package)

## 2.2 Hardware/software mapping

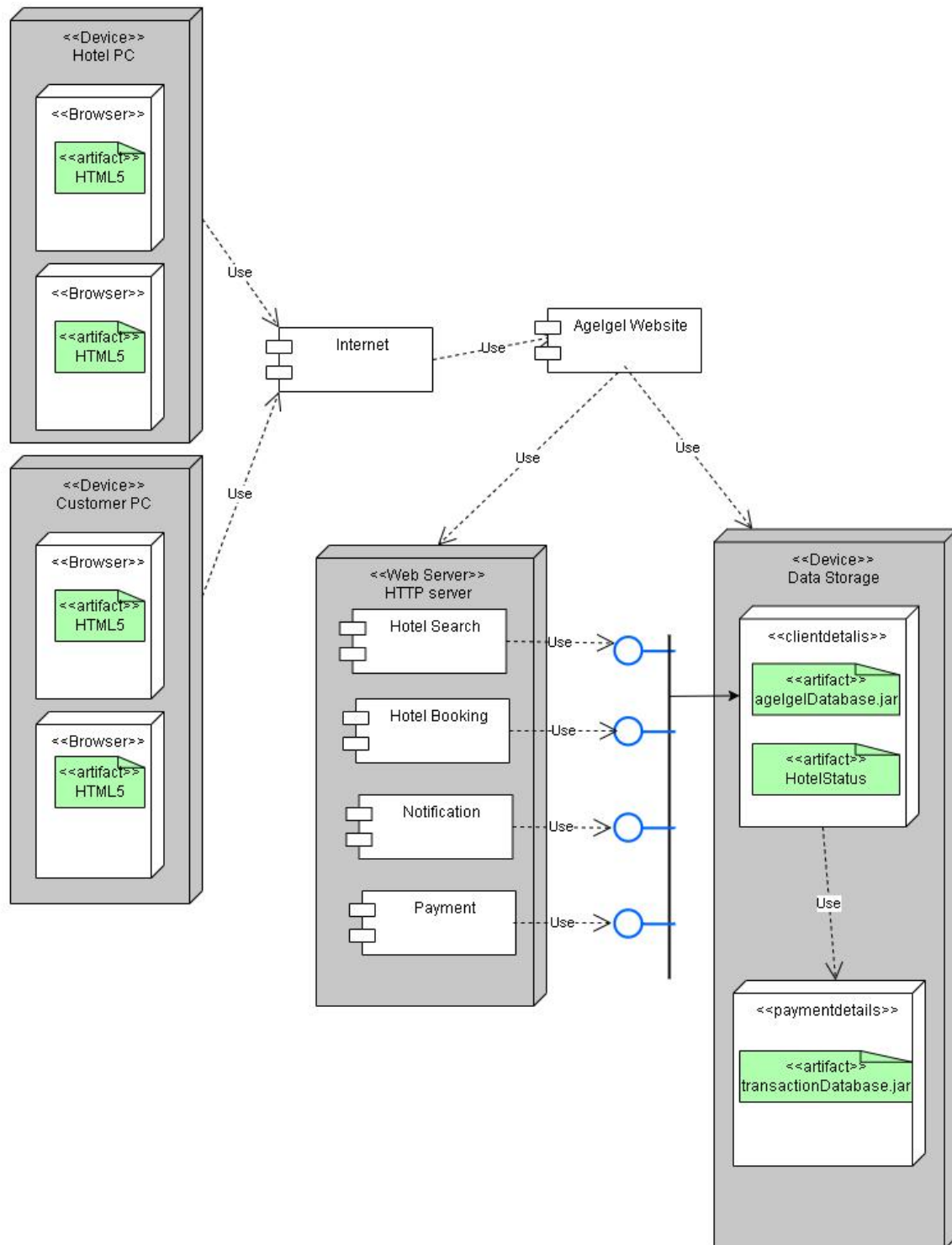


Figure 2.2 Agelgil Deployment Diagram

### 3.1 Class Diagram

FIGURE 3.11 CLASS DIAGRAM

Our class diagram took a lot of space more than we expected and it isn't viewable from here so we provided the link for a better view.

[https://lucid.app/lucidchart/a77a6d53-7cef-4778-af54-2fb5daacb98e/edit?invitationId=inv\\_0cb06d32-4486-437f-bc2b-d33dac0a8284&page=HWEp-vi-RSFO#](https://lucid.app/lucidchart/a77a6d53-7cef-4778-af54-2fb5daacb98e/edit?invitationId=inv_0cb06d32-4486-437f-bc2b-d33dac0a8284&page=HWEp-vi-RSFO#)

## 3.2 Sequence Diagram

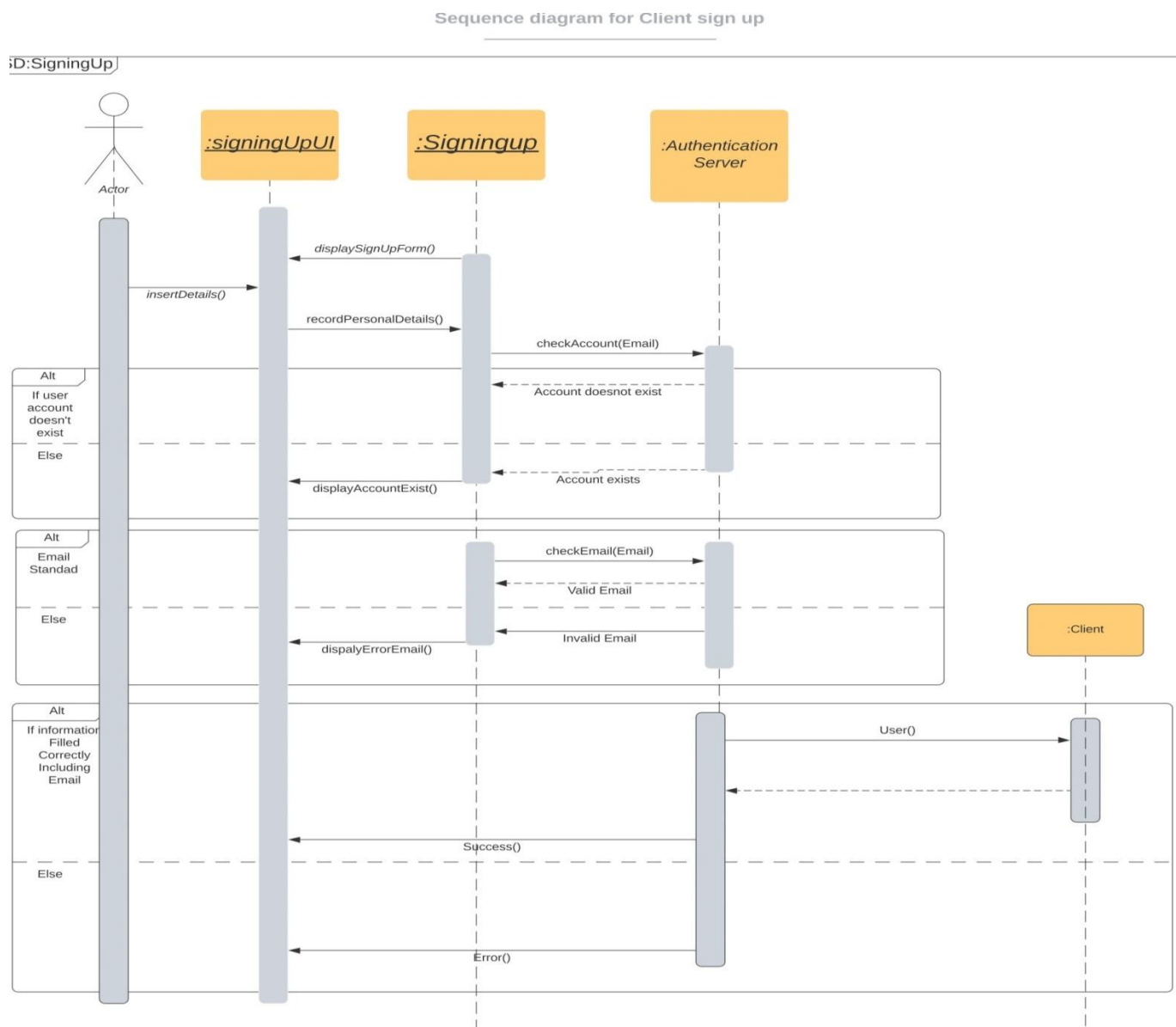


Figure 3.21 Sequence diagram for Sign up

Sequence diagram for Client Login

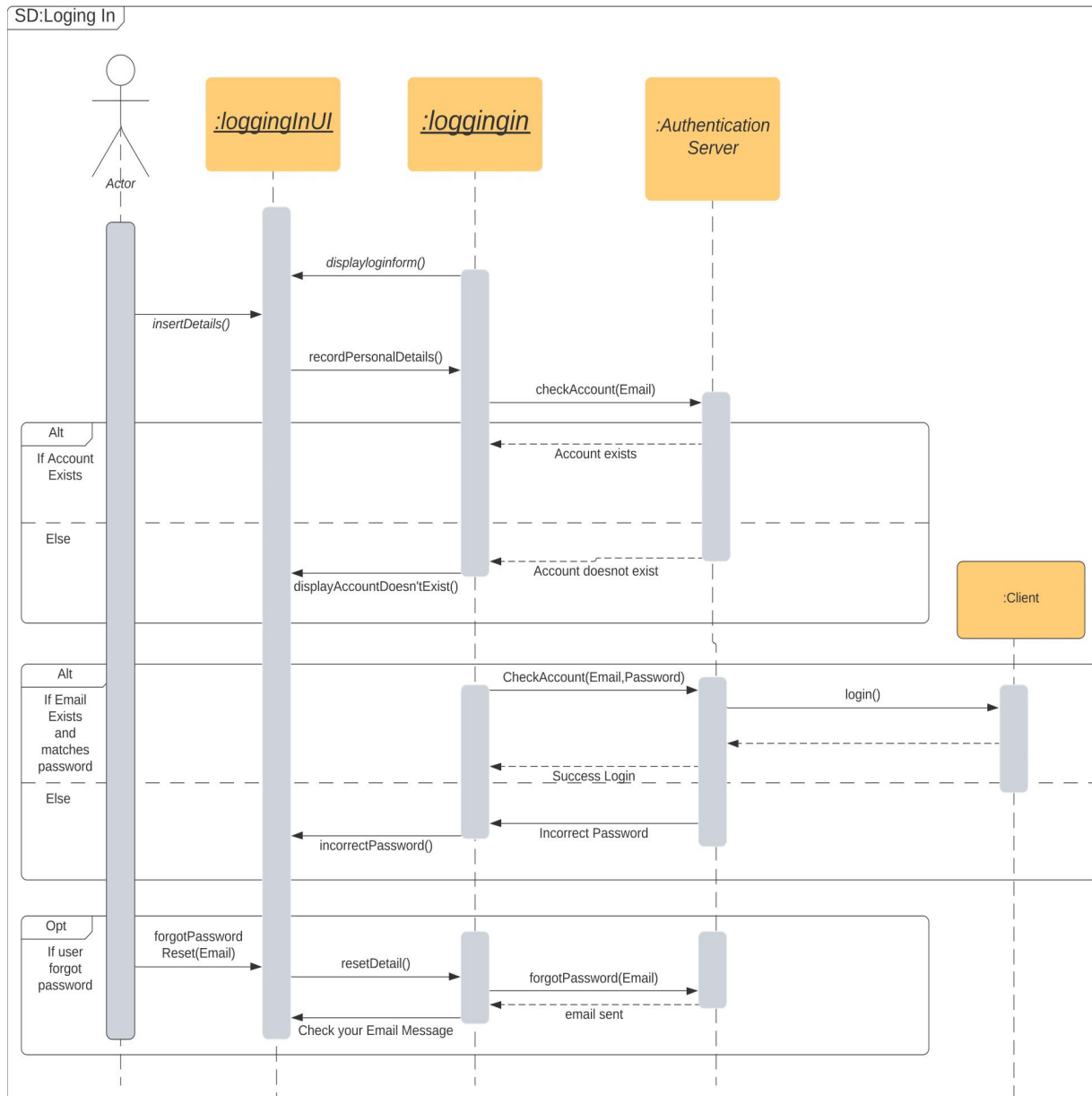
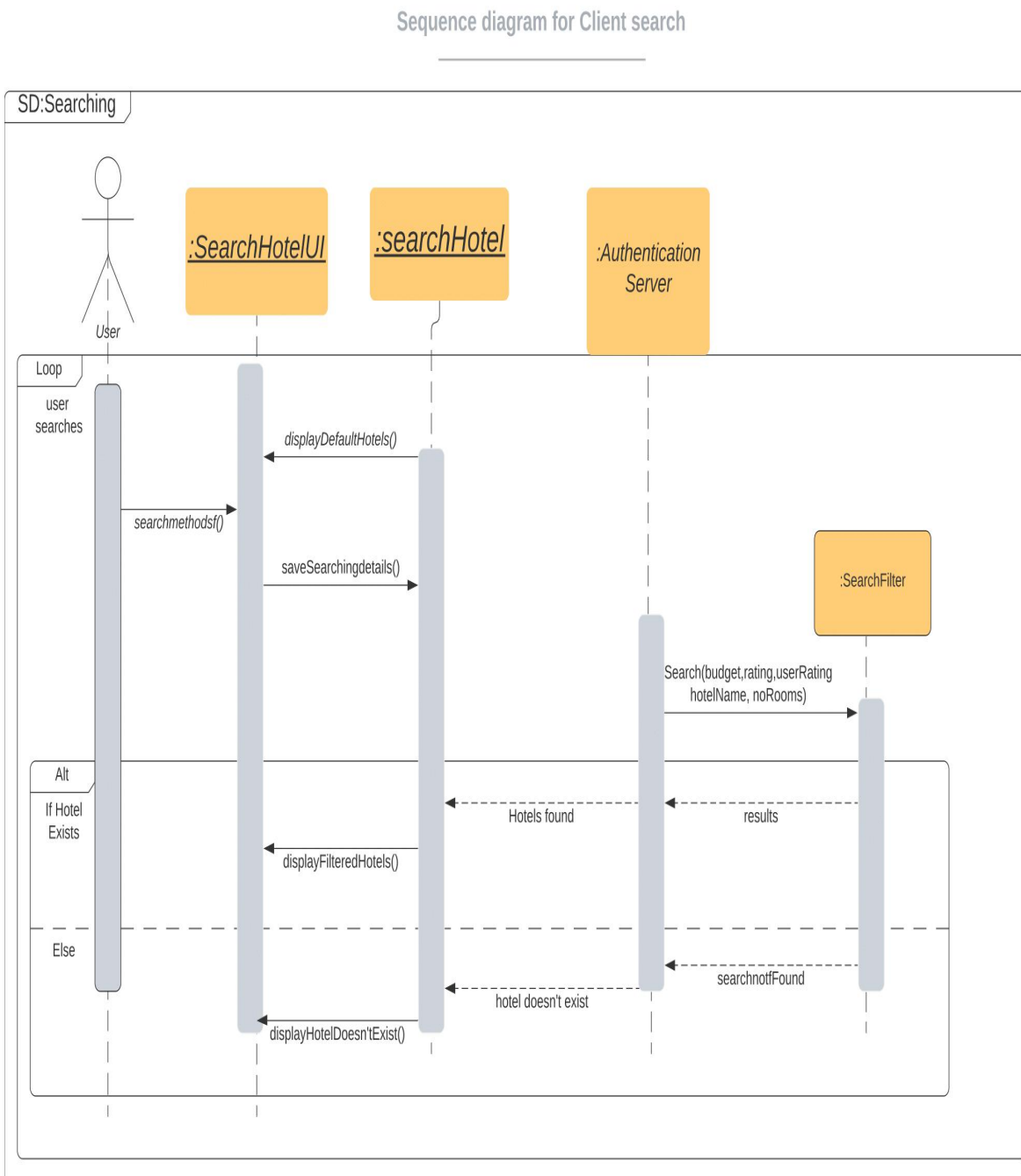


Figure 3.22 Sequence diagram for Login



**Figure 3.23 Sequence diagram for Searching Hotel**

Sequence diagram for Client Choose Hotel

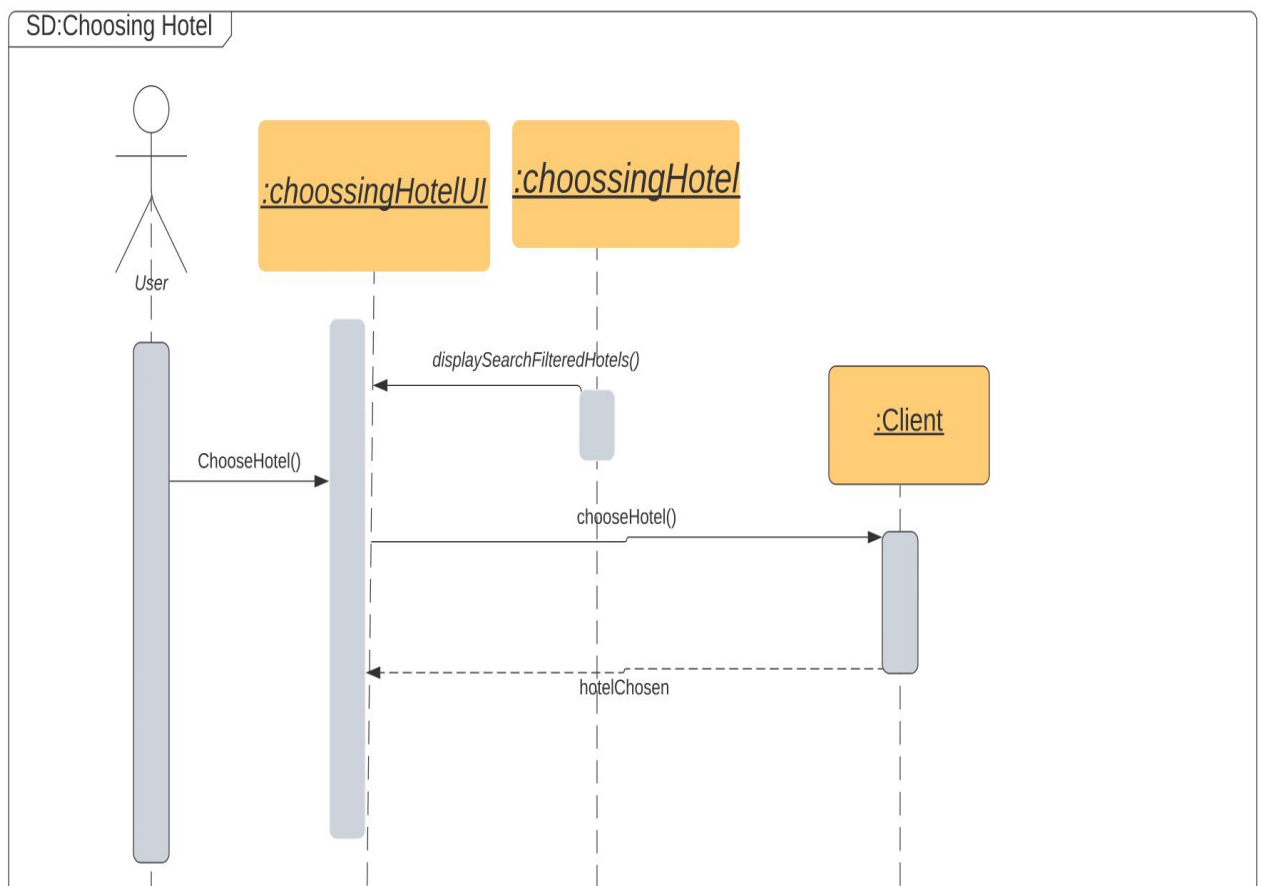


Figure 3.24 Sequence diagram for choosing Hotel



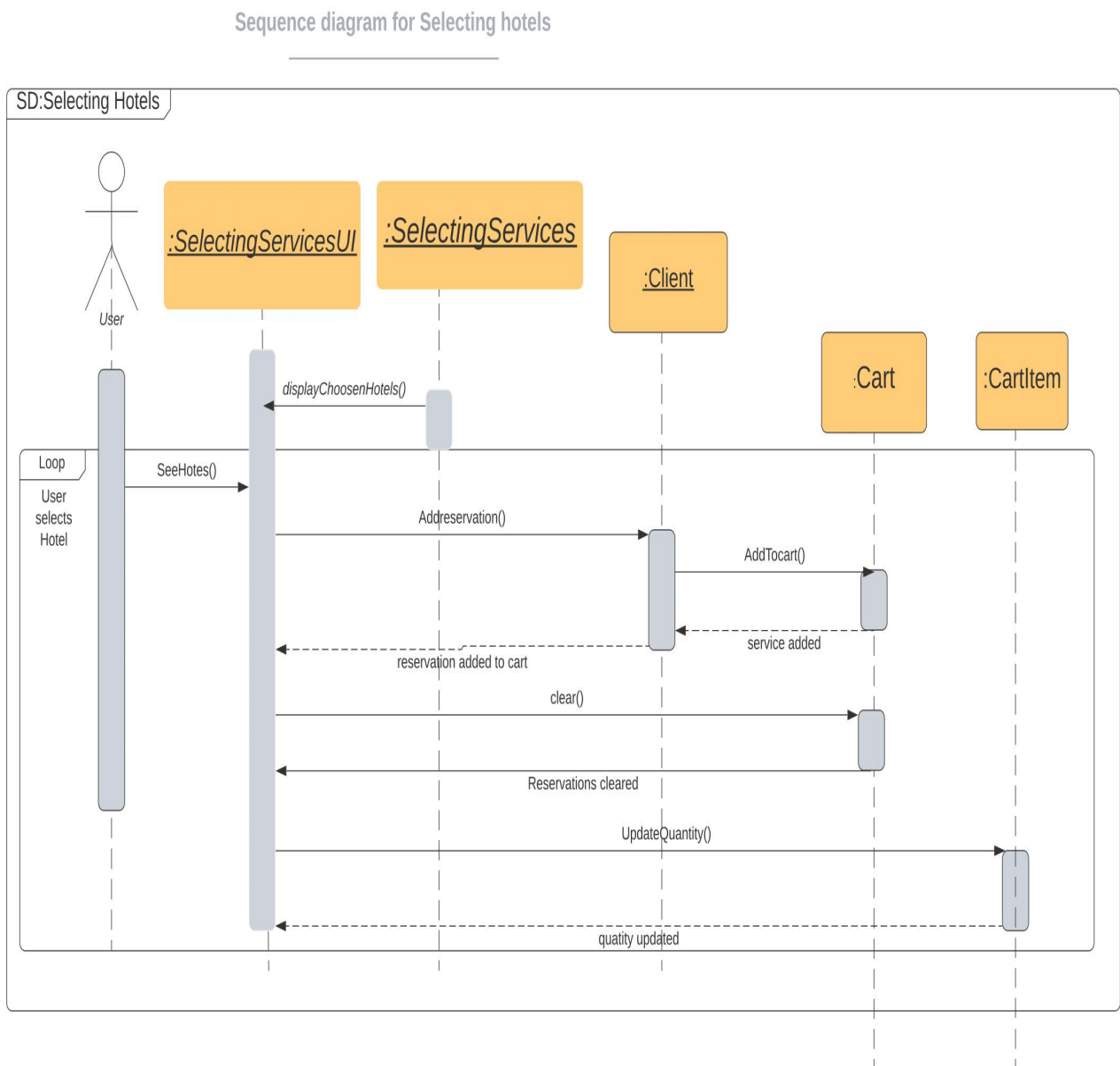


Figure 3.25 Sequence diagram for Selecting Hotel

Sequence diagram for Personal details

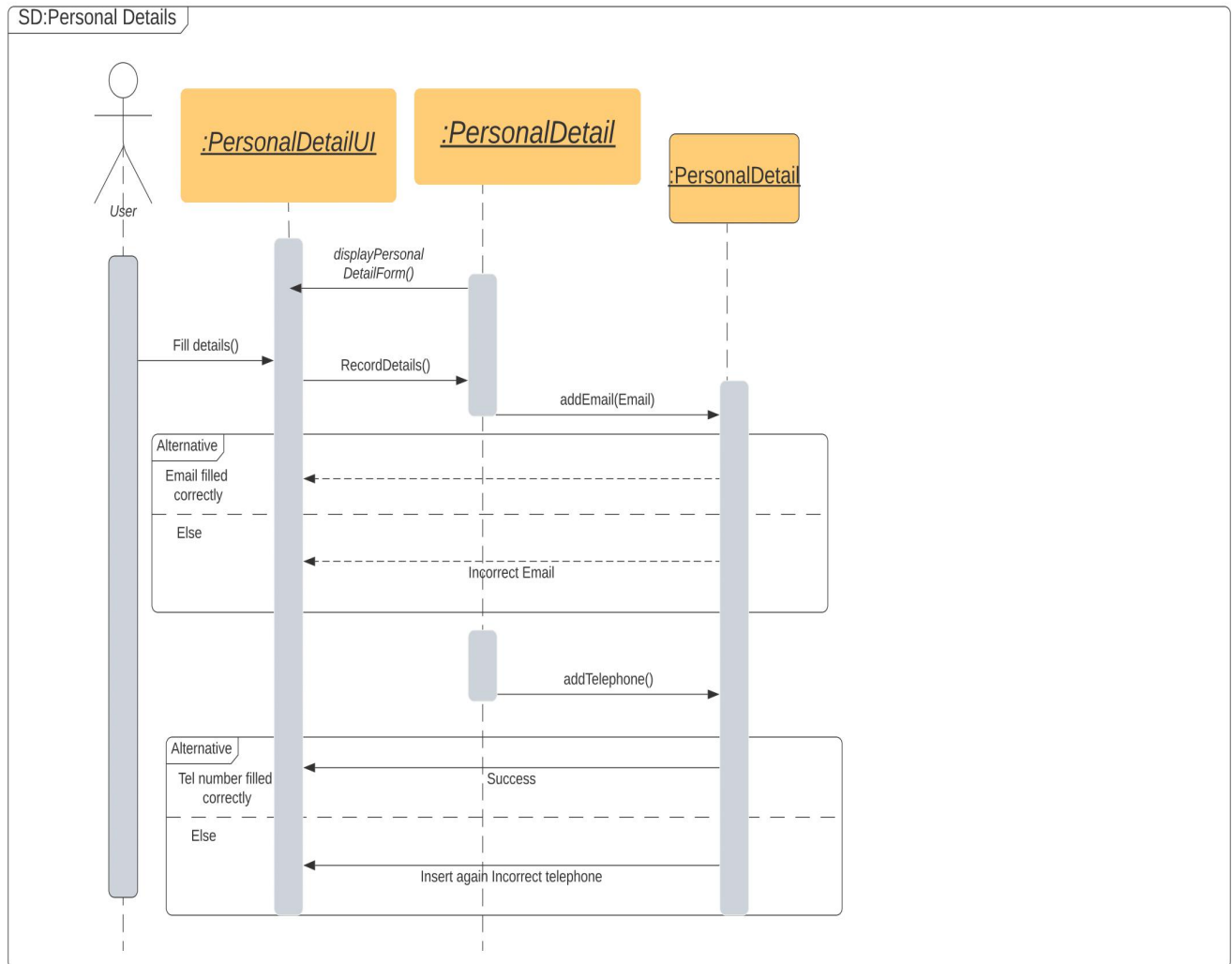


Figure 3.26 Sequence diagram for Personal details

Sequence diagram for Making Payments

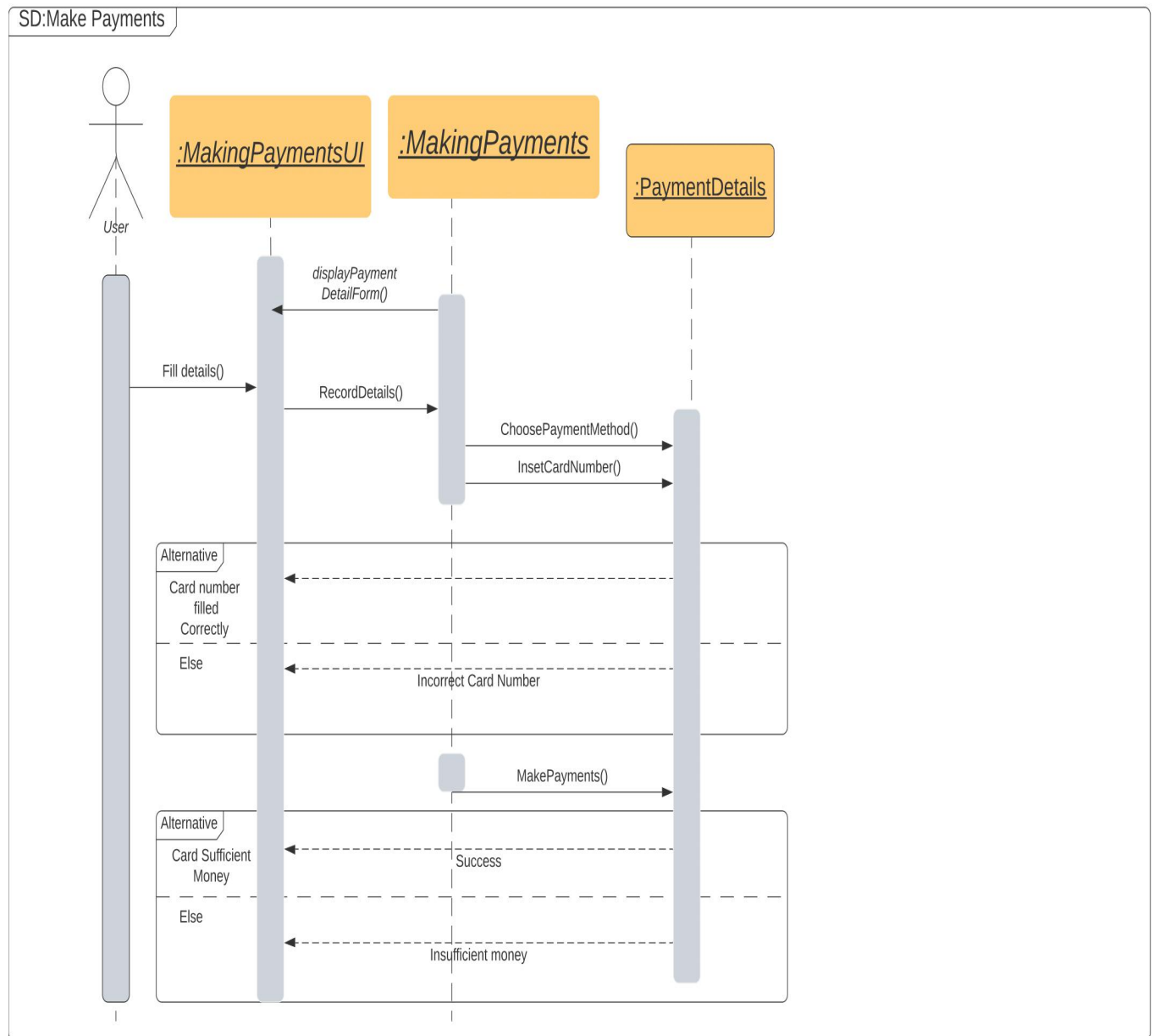


Figure 3.27 Sequence diagram for Making payments

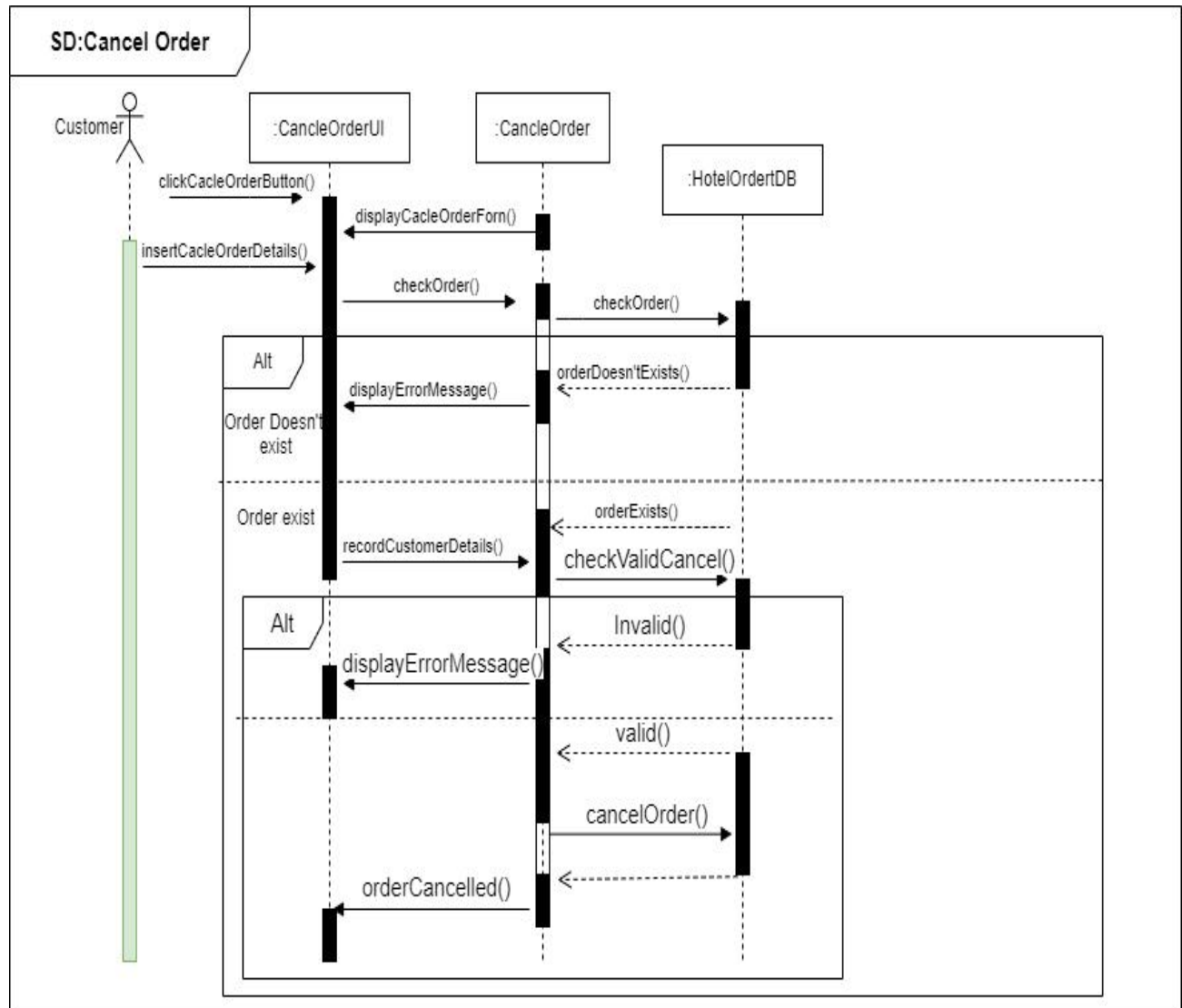


Figure 3.28 Sequence Diagram for Cancellation of Order

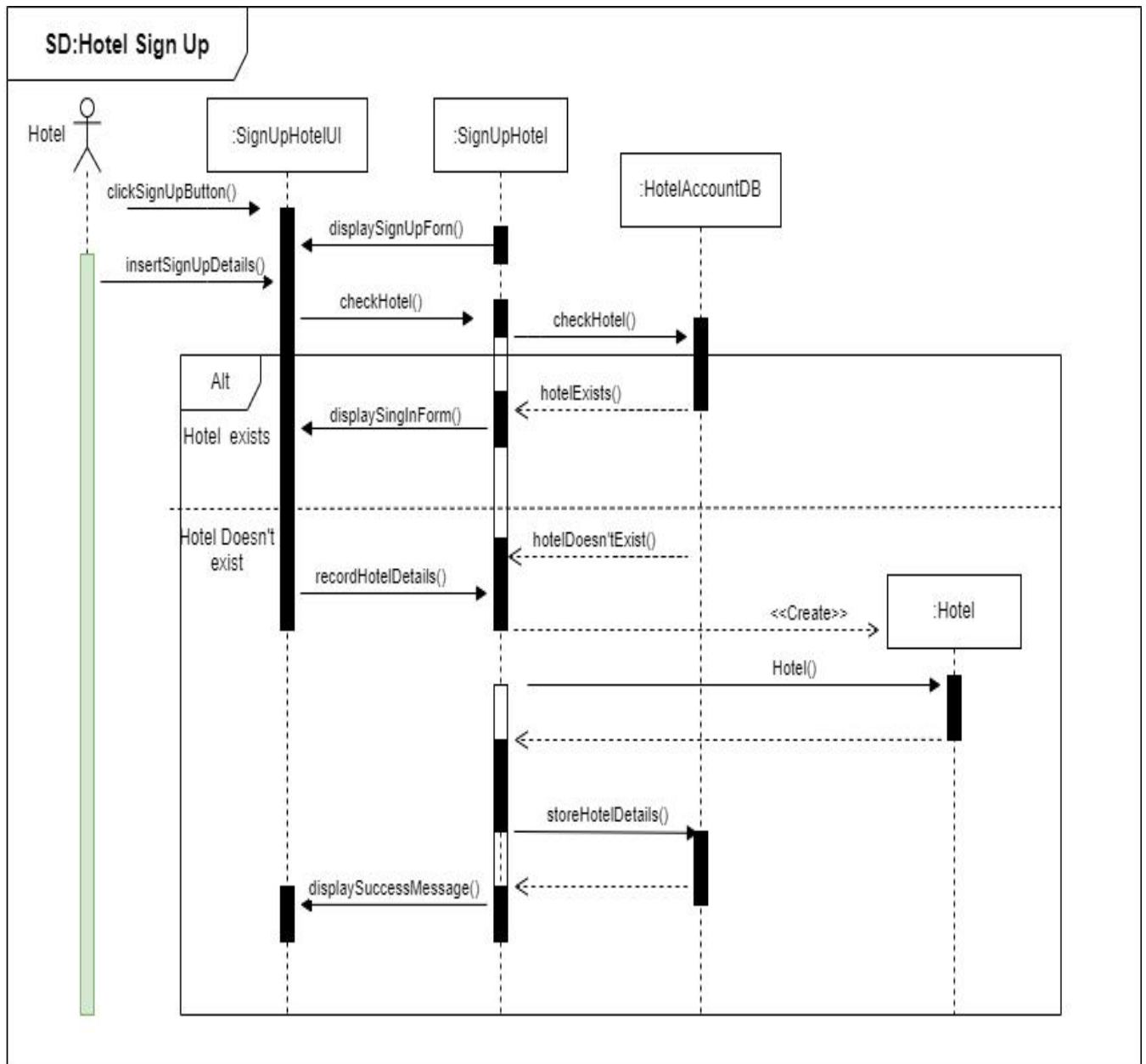


Figure 3.29 Sequence Diagram for Hotel Sign Up

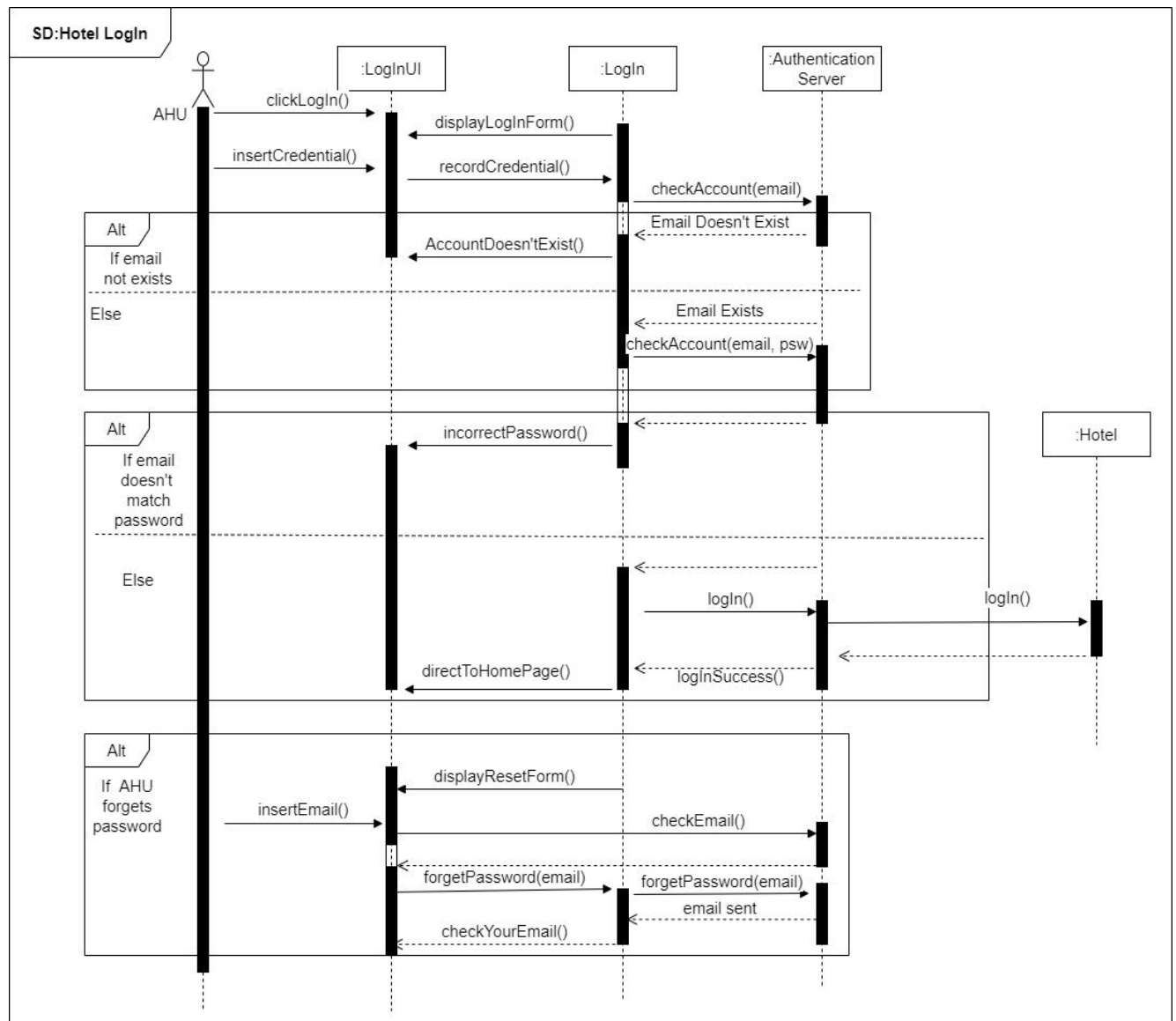


Figure3.30 Sequence Diagram for Hotel Login







## 4. Detailed Design

Figure 4.1 User Model



Table 4.1 Attribute Description for User Class

Attribute	Type	Visibility	Invariant
username	String	private	username $\neq$ NULL ✓ Must contain "@" ✓ Must contain "."(dot) ✓ Position of ">"1 ✓ Position of (dot)>position of @+2 ✓ Position of "."+3 <= total length of email ✓ Total length of email >= 5
password	String	private	Password $\neq$ NULL

			✓ Must be at least 8 characters.
loginStatus	bool	private	✓ Login status indicates the status of the user account

**Table 4.2 Operation Description for User Class**

Operation	Visibility	Return type	Argument	Pre-Condition	Post-Condition
login	Public	void		Login button is clicked	The user should be logged in.
logout	Public	void		Logout button is clicked and the user should be Logged In	The user should be logged out.
getLoginStatus	Public	bool		none	Returns Boolean value for login status

**Figure 4.2 Hotel Model**

## HotelModel

```
-hotelId: Integer  
-hotelName:String  
-isVerified: bool = false  
-location: Location  
-legalDocs: String  
-services: List<Services>  
-images: List<String>  
  
+Hotel(hotel_name,user_id):Object  
+addServices(List<Service> services)  
+verify()  
+getGallery()
```

**Table 4.3 Attribute Description for Hotel**

Attribute	Type	Visibility	Invariant
name	String	private	. username $\diamond$ NULL ✓ Hotel name should be a String
isVerified	bool	private	✓ Indicates whether the Hotel has official recognition by the admin
legalDocs	String	private	LegalDocs $\diamond$ NULL ✓ should end with “.pdf”
location	Location	Private	✓ It is an object of longitude and latitude ✓ Must be 6-digit decimal place
Services	List	Private	Services $\diamond$ NULL ✓ List of Services
images	List	private	

**Table 4.4 Operation Description for Hotel Class**

Operation	Visibility	Return type	Argument	Pre-Condition	Post-Condition
Hotel	public	Object	String, int	none	The method is a constructor which creates a hotel object
addServices	Public	void	List<Service>	The hotel must have an account and must be logged in	The services will be added to the hotels profile
verify	Public	void	none	The hotel object must be created or the hotel must have an account	The hotel's isVerified attribute's value will be true.
getGallery	public	List	none	A hotel Object must be created	A list of images will be returned

**Figure 4.3 Location class**



**Table 4.5 Attribute Description for Location**

Attribute	Type	Visibility	Invariant
longitude	double	Private	✓ must be in a six-digit Decimal Degrees
latitude	double	Private	✓ must be in a six-digit Decimal Degrees

**Table 4.6 Operation Description for location class**

Operation	Visibility	Return type	Argument	Pre-Condition	Post-Condition
getLongitude	Public	double	none	getLocation method is called	The address of the hotel in decimals degree will be returned

getLatitude	Public	double	none	getLocation method is called	The address of the hotel in decimals degree will be returned
-------------	--------	--------	------	------------------------------------	--

**Figure 4.4 Service class**



**Table 4.7 Attribute Description for Service**

Attribute	Type	Visibility	Invariant
unitPrice	float	private	✓ Must be > 0
serviceName	String	private	✓ name <> NULL ✓ service name must be a string

availableUnits	int	private	✓ Must be $\geq 0$ .
----------------	-----	---------	----------------------

**Table 4.8 Operation Description for Service**

Operation	Visibility	Return type	Argument	Pre-Condition	Post-Condition
getUnitPrice	public	float	none	none	The price will be returned
getName	public	String	none	none	The name of the service will be returned
avaialableUnits	Public	int	none	none	The number of available units of the hotel object will be sent to the callers

**Figure 4.5 Order Class**





**Table 4.9 Attribute Description for Order**

Attribute	Type	Visibility	Invariant
orderId	int	private	<ul style="list-style-type: none"> <li>✓ Must be Unique</li> <li>✓ Must be randomly generated</li> <li>✓ Must be at least 10 digits characters (letters and numbers)</li> </ul>
datetime	datetime	private	<ul style="list-style-type: none"> <li>✓ must be in this format <b>yyyy-MM-dd</b></li> </ul>
token	String	private	<ul style="list-style-type: none"> <li>✓ must 256 hexadecimal characters</li> </ul>

**Table 4.10 Operation Description for Order Class**

Operation	Visibility	Return type	Argument	Pre-Condition	Post-Condition
-----------	------------	-------------	----------	---------------	----------------

getTotalPrice	Public	float	none	Services must be selected or add to the cart	- The Total price of the services will be returned to the caller
getToken	Public	String	none	A random token must be generated for a specific user	- The Token of the services will be returned to the caller
getDateTime	Public	Datetime	none	Order must be made for the particular user	The Date and time of the services will be returned to the caller

**Figure 4.6 CartItem class**



**Table 4.11 Attribute Description for CartItem**

Attribute	Type	Visibility	Invariant
quantity	int	private	✓ Must be > 1

Table 4.12 Operation Description for Cart Item

Operation	Visibility	Return type	Argument	Pre-Condition	Post-Condition
updateQuantity	Public	none	int	A service must be selected	Quantity of the selected item will be updated
getTotalPrice	Public	Float	none	Services must be selected or add to the cart	- The Total price of the services will be returned to the caller

Figure 4.7 Cart class



Table 4.13 Attribute Description for Cart

Attribute	Type	Visibility	Invariant
cartId	int	private	✓ cartId $\neq$ NULL ✓ Must be randomly generated and unique
cartItems	List	private	✓ List of cart items objects

**Table 4.14 Operation Description for Cart Class**

Operation	Visibility	Return type	Argument	Pre-Condition	Post-Condition
getTotalPrice	Public	float	none	Services must be selected or add to the cart	- The Total price of the services will be returned to the caller
clear	Public	void	none	Must be on the Service Cart page and Some service must be there	All Cart Items referring the cart should be deleted.
addToCart	public	none	CartItem	Service of a hotel must be selected	Items that are given through the parameter will be add to the specified cart for the user

createOrders	Public	List<Order>	none	Services must be successfully added to the cart	Services in the cart shall be ordered with respect to their hotels to form Orders.
--------------	--------	-------------	------	---	--

**Figure 4.8 Customer Model Class**



**Table 4.15 Attribute Description for CustomerModel Class**

Attribute	Type	Visibility	Invariant
customerName	String	private	✓ Must contain 1 “ “ (space) character. ✓ Must be a string

Table 4.16 Operation Description for Customer Class

Operation	Visibility	Return type	Argument	Pre-Condition	Post-Condition
Customer	public	Customer Object	String, String	none	The method is a constructor and a customer object will be created
register	public	none	none	Signup button is clicked and the appropriate fields to sign up are filled	The customer will be registered
addReservations	public	none	Service	A service must be selected	A reservation will be added to the Cart

Figure 4.9 Admin Model Class



**Table 4.17 Attribute Description for Admin Class**

Attribute	Type	Visibility	Invariant
name	String	private	<ul style="list-style-type: none"><li>✓ Must contain exactly 1 “ ” (space) character.</li><li>✓ Must be a String</li></ul>

**Table 4.18 Operation Description for Admin Class**

Operation	Visibility	Return type	Argument	Pre-Condition	Post-Condition
Admin	public	Admin Object	String, int	none	The method is a constructor and an Admin object will be created

## References

- Sommerville, Ian (2016). *Software engineering. 9th Edition*, Boston: Pearson
- A.Hunt and David Thomas (1999) *Pragmatic Programmer* USA
- S. Faulk (1995), University of Oregon [Software Requirements: a Tutorial](#)
- Pressman, Roger (2015). *Software engineering: a practitioner's approach*. New York, NY: McGraw-Hill Education,
- Ivan Marsic (2009), *Software engineering*, Rutgers University