Vanier College
Faculty of Science and Technology
System Development
420-436-VA (Sect. 00001)

Deliverable 04

Orange Team

Trevor Obodoechina

Janna Lomibao

Sen Bellotiny Donissaint

Maxime Mir-Orefice

Signatures

I, Maxime Mir-Orefice (2319522), certify that I contributed to this deliverable. M.M-O

I, Janna Lomibao(2322746), certify that I contributed to this deliverable. J.A.L

I, Sen bellotiny Donissaint (2255525), certify that I contributed to this deliverable. S.B.D

I, Trevor Obodoechina (2149315), certify that I contributed to this deliverable. T.O

Table of Contents

Signatures **1**
Table of Contents **2**
Executive Overview **3**
Summary Description of the Client **4**
Business Problem **5**
Narrative Description **6**
User Stories Map **7**
Appendix 1 - Flowcharts **11**
Admin 11
Registered 12
Unregistered 13
Contact Us 14
Booking 15
Account 16
Appendix 2 - Use Cases **17**
Unregistered Client 17
Registered Client 18
Admin 19
Appendix 3 - Sequence Diagrams **20**
Login 20
Register 21
Booking 22
Review 23
Inventory 24
Contact 25
Appendix 4 - Class Diagrams **26**
Controllers 26
Models 27
Appendix 5 ..... **28**
Project Plan **29**
Bibliography **30**

Executive Overview

In Delivery 4, we made UML diagrams that would be used to represent the actors and actions that they could take. The UML diagrams also show 3rd parties involved in certain processes that happen in the back-end. Many classes had to be made as the project or website will incorporate many things. The actions revolving around those classes also made use of APIs to help with actions such as payment, photos, location, etc. While making those diagrams we became aware of certain problems or questions we might encounter later on. Some of the solutions to those problems were simple while others remain to be discussed with the client.

Summary Description of the Client

Madeleine is proficient in computer software for organizing like excel. She is quick to understand technology and has a very positive attitude towards it.

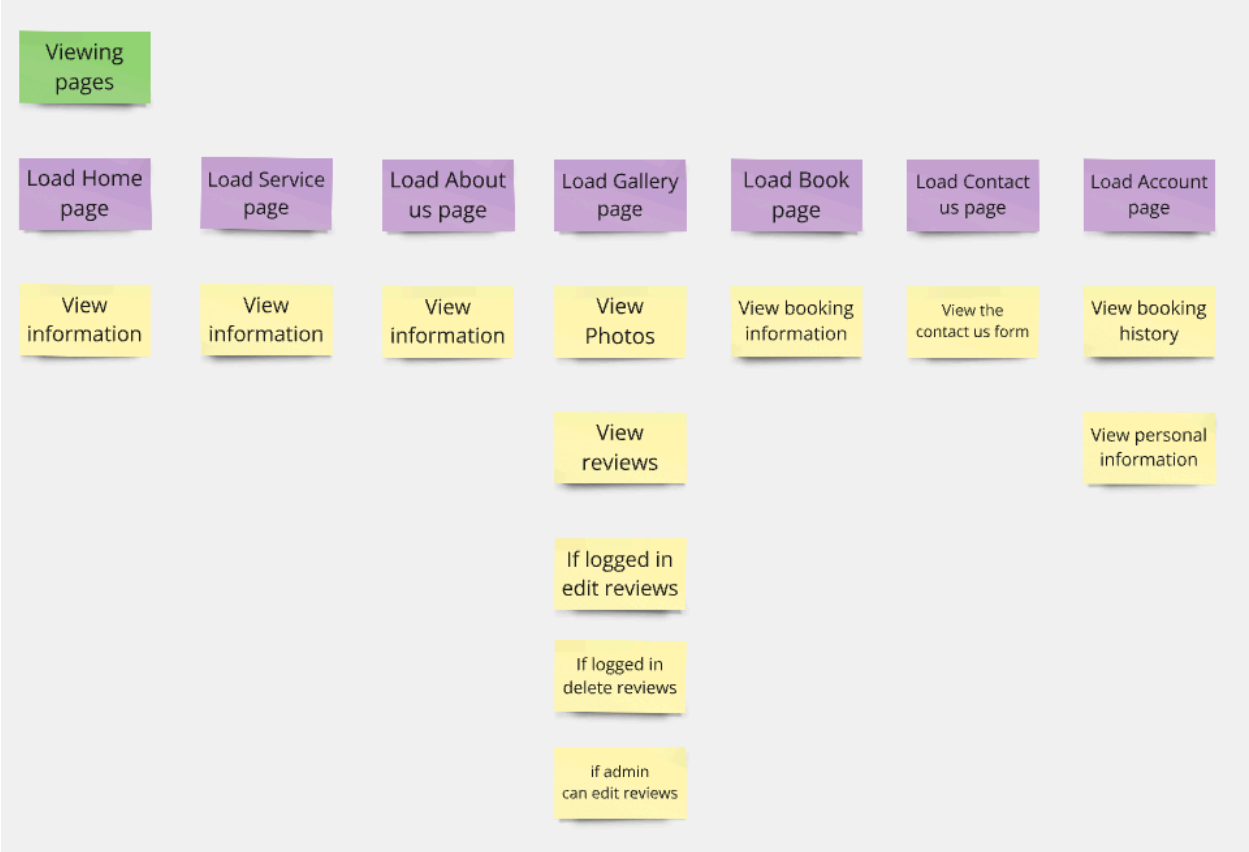
Business Problem

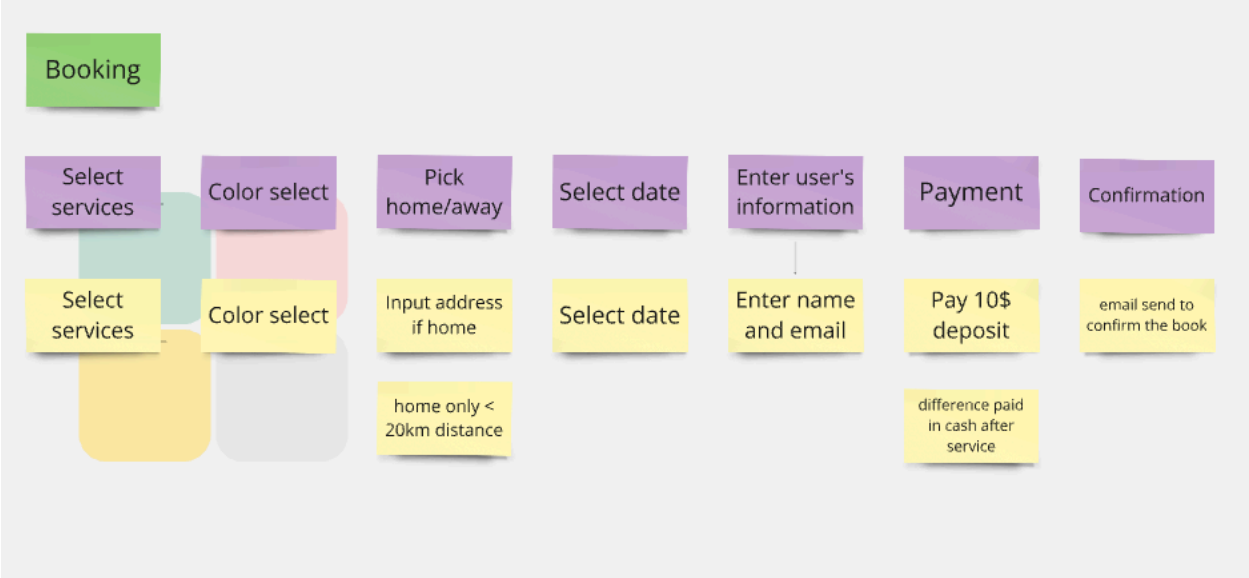
A nail salon can face several challenges such as appointment scheduling issues, lack of reviews, and more. These problems can be resolved by creating a website that is well-defined. This will improve the connection to clients and provide a chance for business growth. One of the most common problems that a nail salon business encounters is lack of exposure. A website could open a wider area to reach new clients. This will not only increase profit but also can bring stability to the business. Another challenge is appointment scheduling issues. Relying on manual booking , such as phone and direct messages, this can cause overbooking or missed appointments that can cause trouble not only for the owner but also the clients. A website can also allow clients to change or cancel appointments without taking any time from the client.

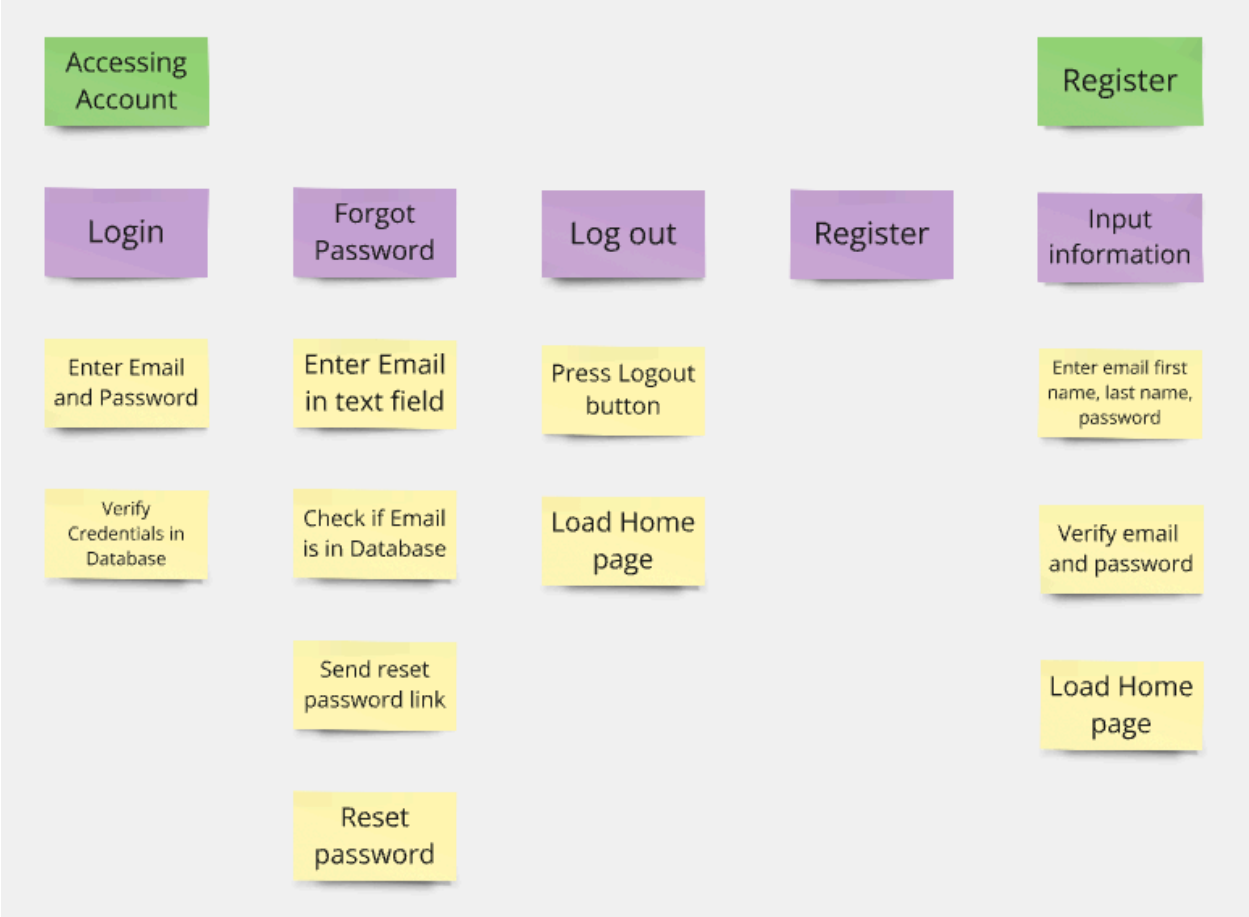
Narrative Description

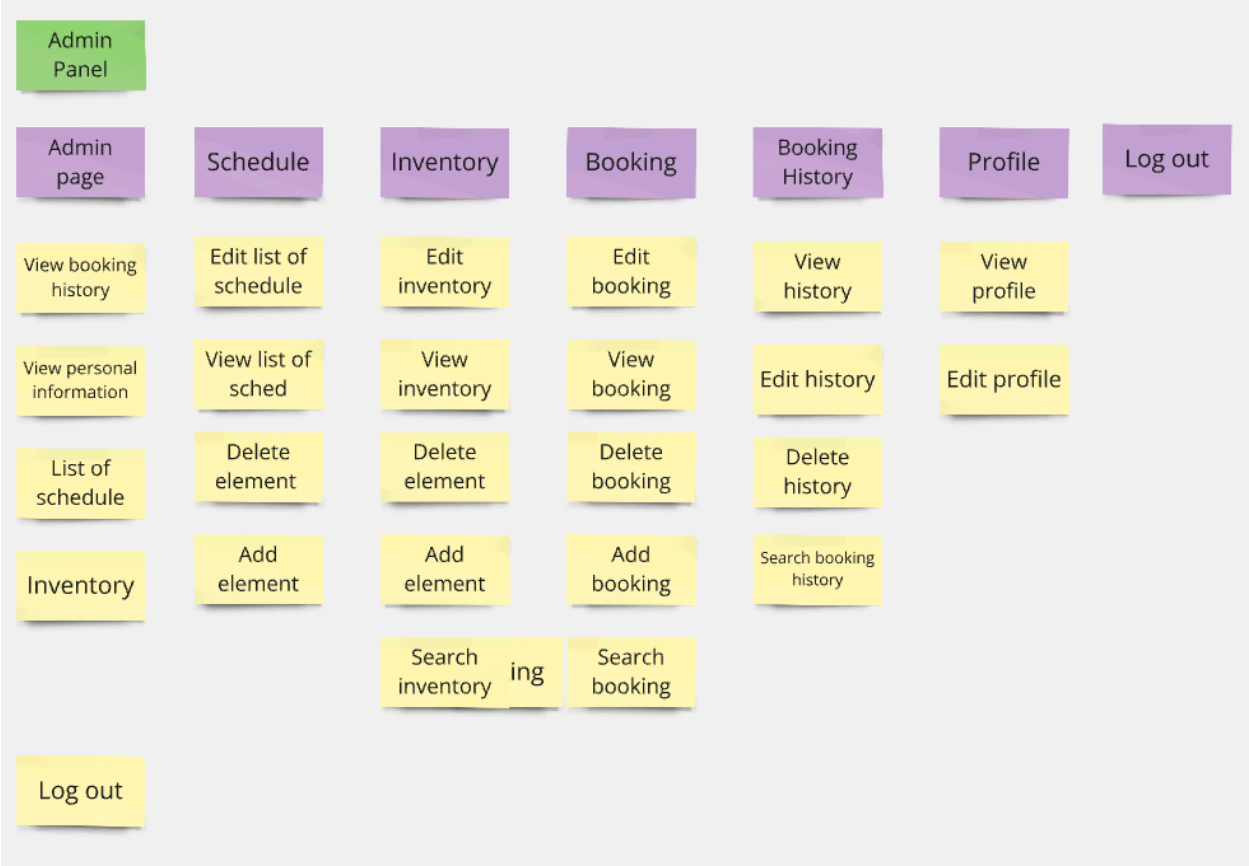
The future prototype will have 3 primary roles, being **owner/admin**, **registered client** and **unregistered client** . **Unregistered clients** will be able to view the gallery, see information on the owner and her information, book appointments on the website and contact the owner through the **Contact Us** page. The main difference between a registered client and someone who hasn't yet is that a registered client can view and access their history of booking to facilitate booking again. An unregistered client is also given the option of booking an appointment at their home, **if** it is in the 20km radius of the owner's home. Next is the role of the **registered client** who has the ability to do everything an unregistered client can do. They can also write reviews and view their booking history. The client will be able to get a discount if it's their birthday as a benefit of registering. The client is able to bring their own nail polish if they would like, but as there is a limit of three different colored nail polish, their home brought nail polish would count as one of three. The last main role is the **owner/admin**, which in this case would be our current chosen client (Madeleine Snook). Other than all the freedom a registered user has, an owner is able to manipulate the whole website's system, including being able to manipulate reference photos, edit reviews and upload new swatches to the website. The owner can also add, edit or remove prices of the services. To help with encouraging registering, the owner will upload discounts depending on a client's birthday. Other types of discounts are also possible; E.g. holidays and special events. The owner/admin and registered client must login successfully to access or interact with the system. To successfully login a user must enter an email and password matching the one found in the database. In the case that the user wants to change their password at a later time, they will have to use their current credentials (email). In the separate case of the user having forgotten their password, they will go through almost the same process. Processes such as logging in, registering and password configuration will require 2FA after successfully entering credentials.

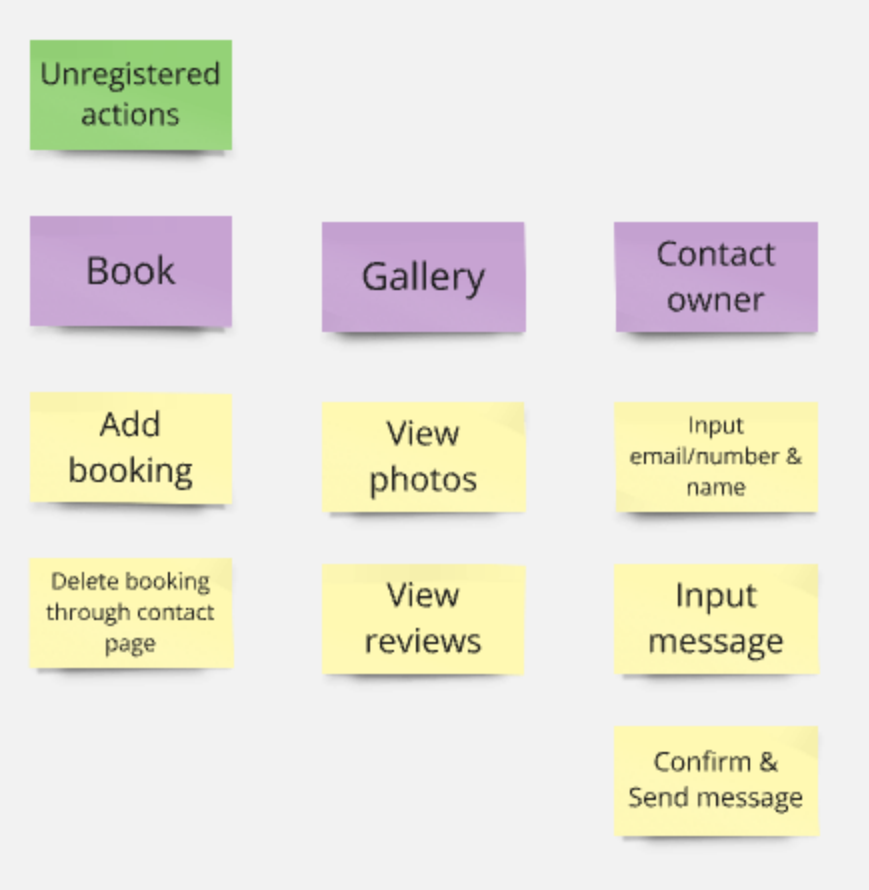
User Stories Map

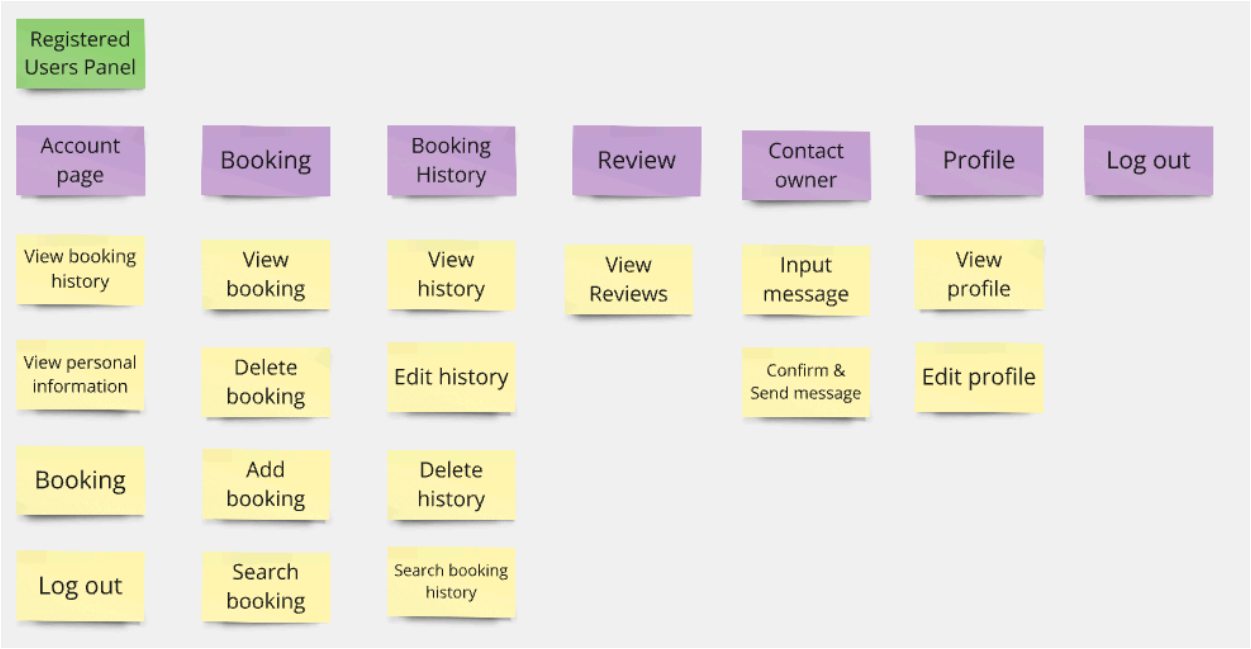





Accessing Account
Login
Enter Email and Password
Verify Credentials in Database
Forgot Password
Enter Email in text field
Check if Email is in Database
Send reset password link
Reset password
Log out
Press Logout button
Load Home page
Register
Register
Input information
Enter email first name, last name, password
Verify email and password
Load Home page

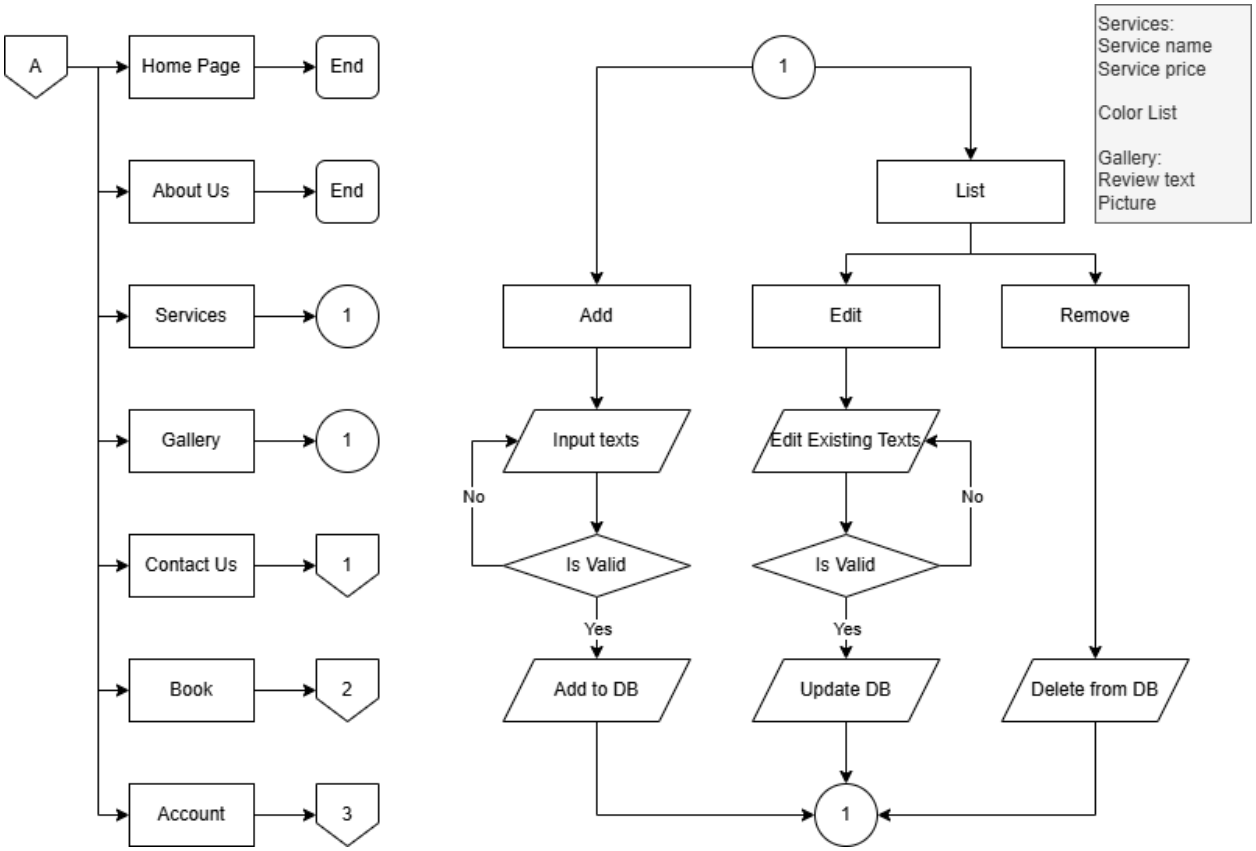

Admin Panel
Admin page
Schedule
Inventory
Booking
Booking History
Profile
Log out
View booking history
View personal information
List of schedule
Inventory
Log out
Edit list of schedule
View list of sched
Delete element
Add element
Edit inventory
View inventory
Delete element
Add element
Search inventory
ing
Edit booking
View booking
Delete booking
Add booking
Search booking
View history
Edit history
Delete history
Search booking history
View profile
Edit profile

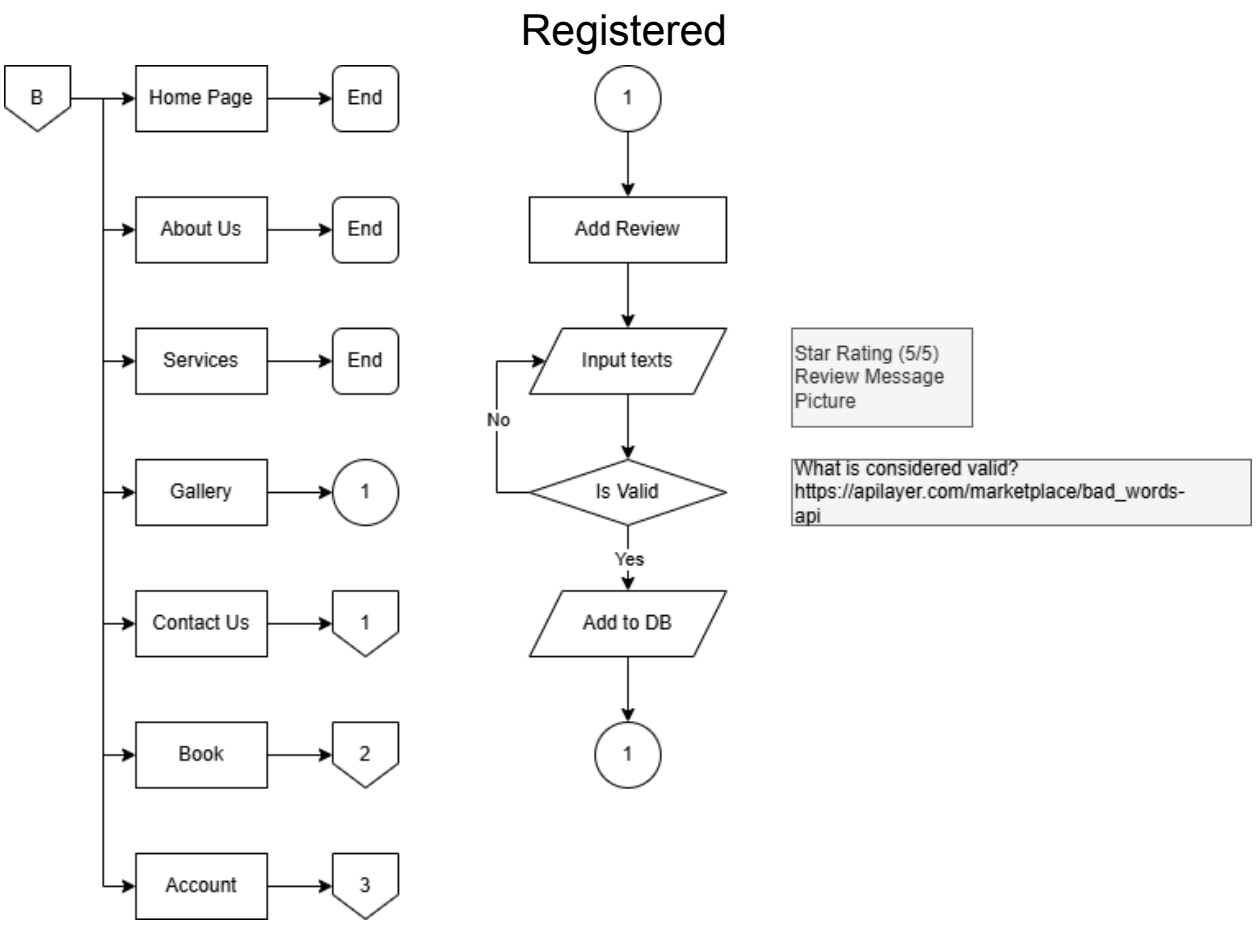

Unregistered actions
Book
Gallery
Contact owner
Add booking
View photos
Input email/number & name
Delete booking through contact page
View reviews
Input message
Confirm & Send message


Registered Users Panel
Account page
Booking
Booking History
Review
Contact owner
Profile
Log out
View booking history
View booking
View history
View Reviews
Input message
View profile
View personal information
Delete booking
Edit history
Confirm & Send message
Edit profile
Booking
Add booking
Delete history
Log out
Search booking
Search booking history

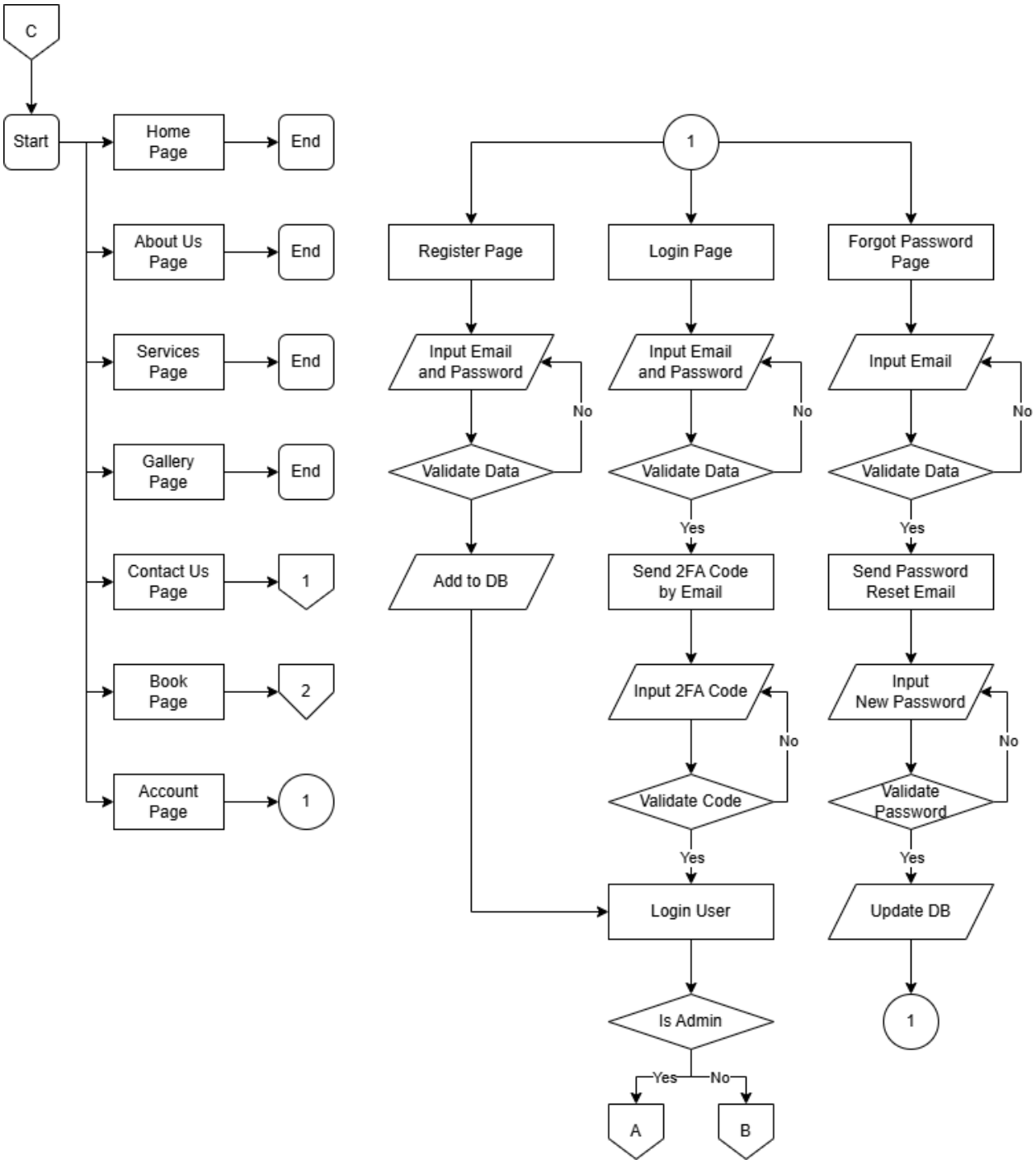
Appendix 1 - Flowcharts

Admin

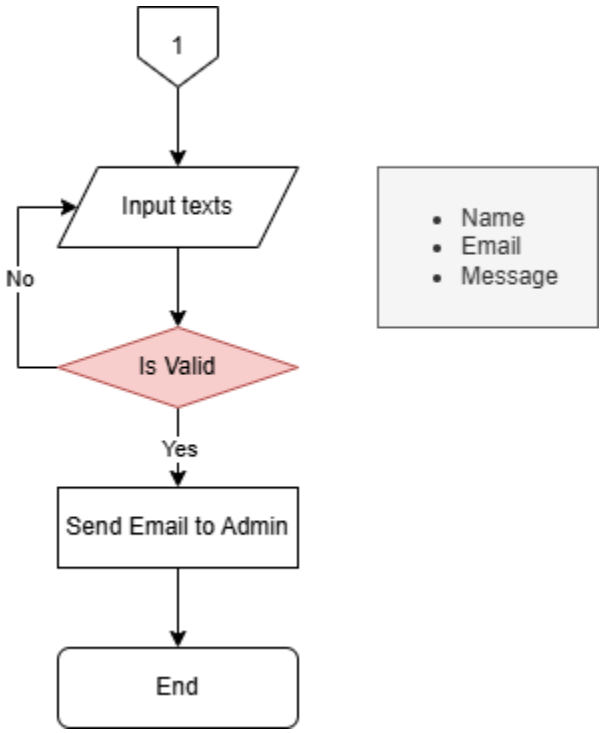



Registered
B
Home Page
End
About Us
End
Services
End
Gallery
1
Contact Us
1
Book
2
Account
3
1
Add Review
Input texts
No
Is Valid
Yes
Add to DB
1
Star Rating (5/5)
Review Message
Picture
What is considered valid?
https://apilayer.com/marketplace/bad_words-api

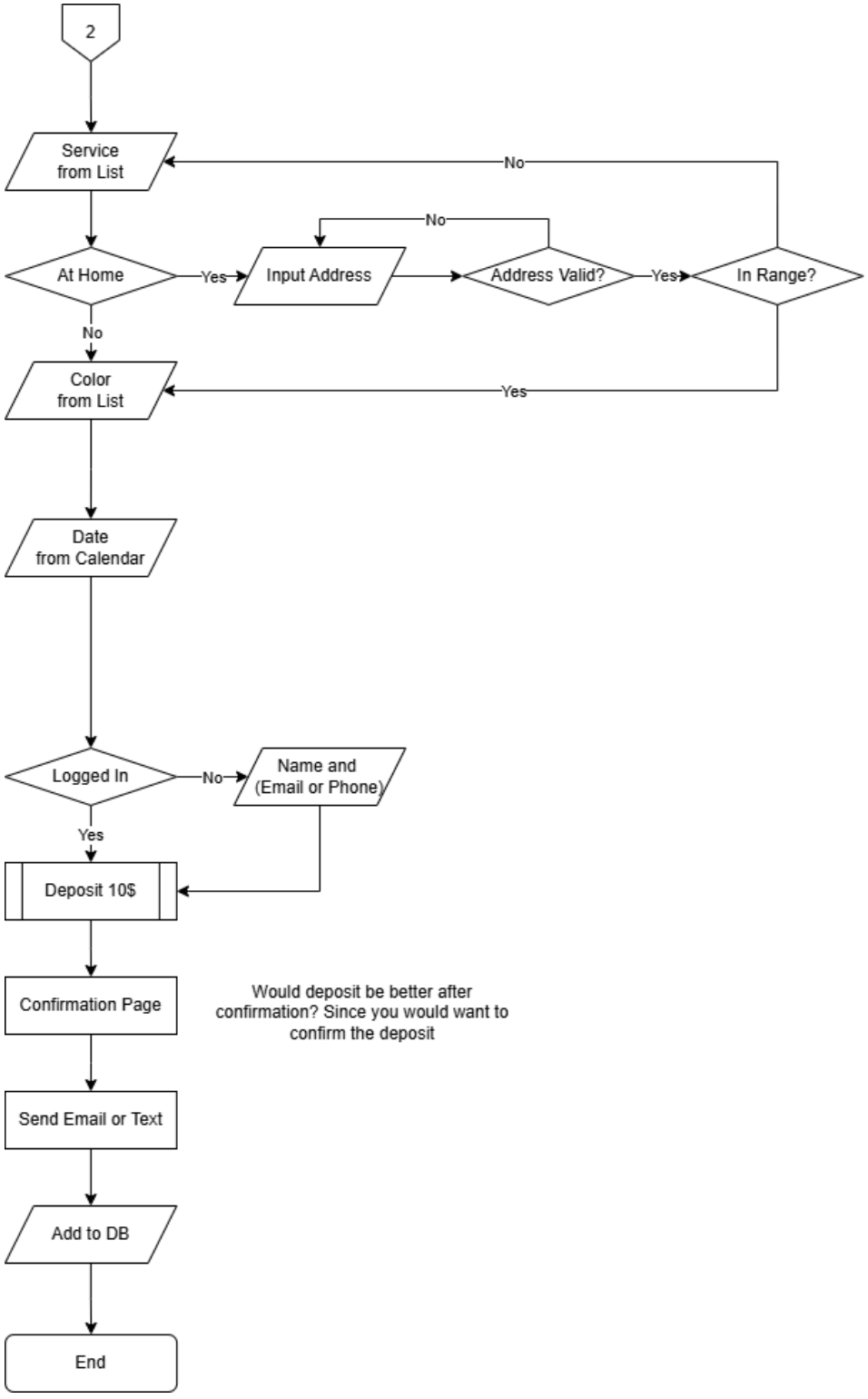
Unregistered



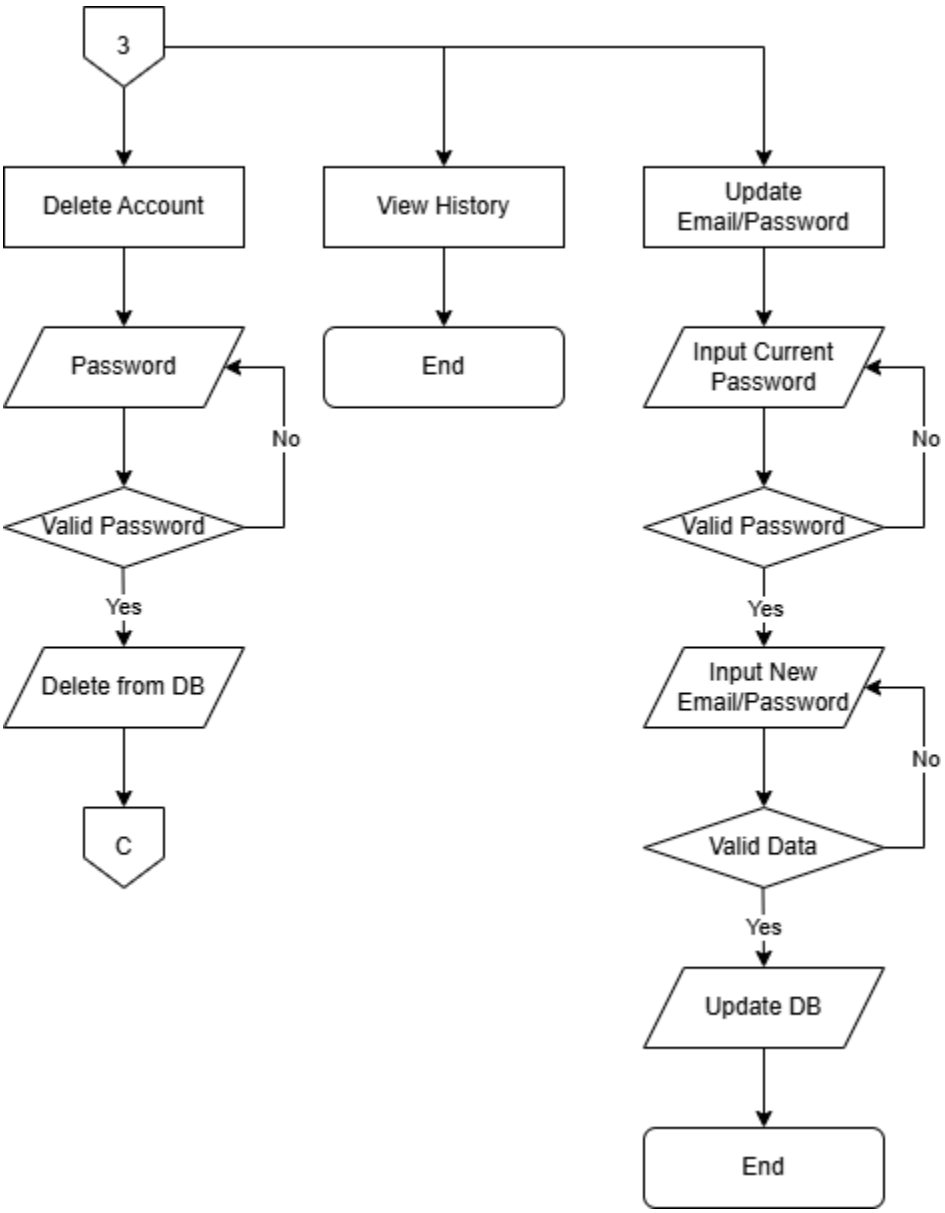
Contact Us


1
Input texts
No
Is Valid
Yes
Send Email to Admin
End
Name
Email
Message

Booking

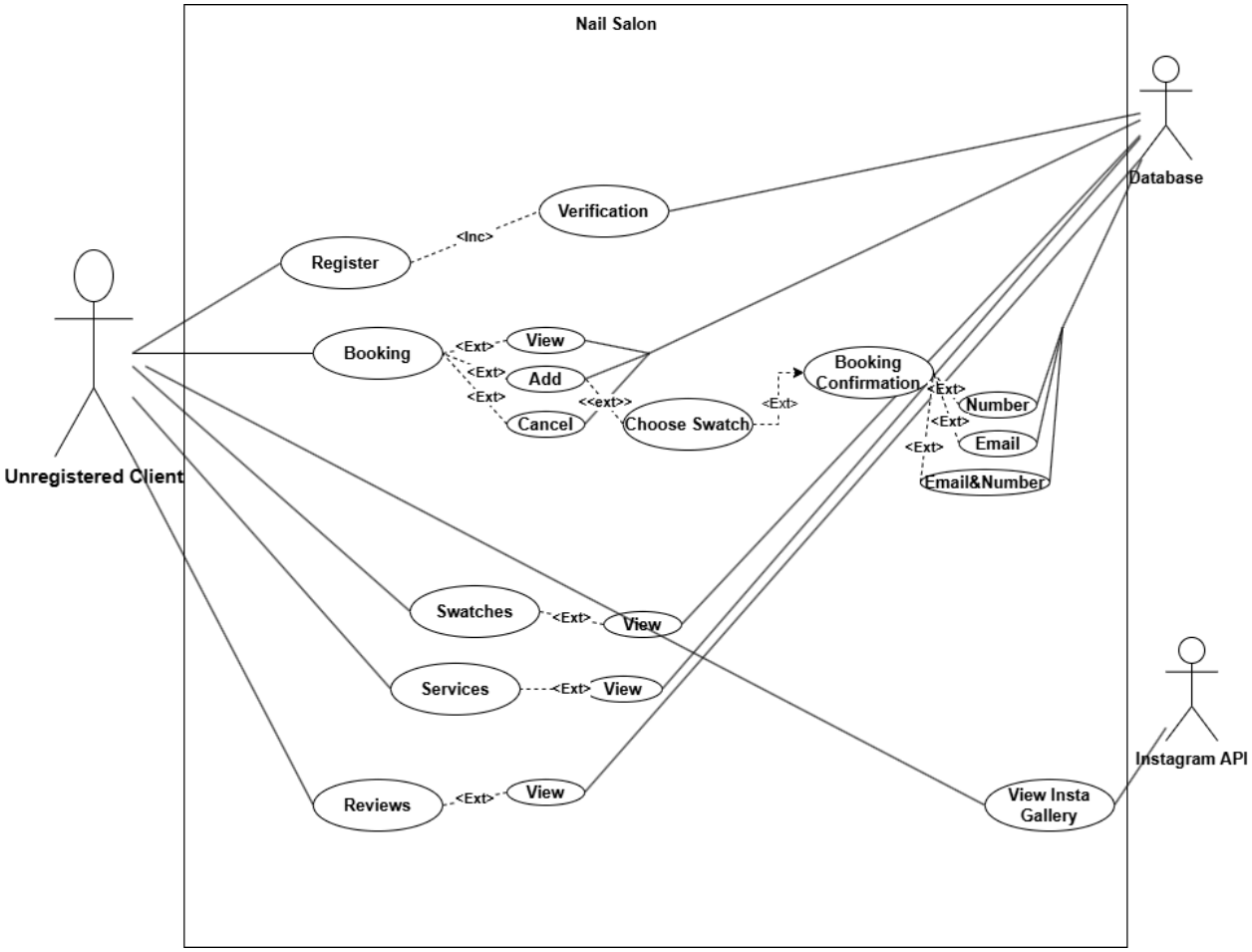

2
Service from List
No
No
At Home
Yes
Input Address
Address Valid?
Yes
In Range?
No
Color from List
Yes
Date from Calendar
Logged In
No
Name and (Email or Phone)
Yes
Deposit 10$
Confirmation Page
Would deposit be better after confirmation? Since you would want to confirm the deposit
Send Email or Text
Add to DB
End

Account

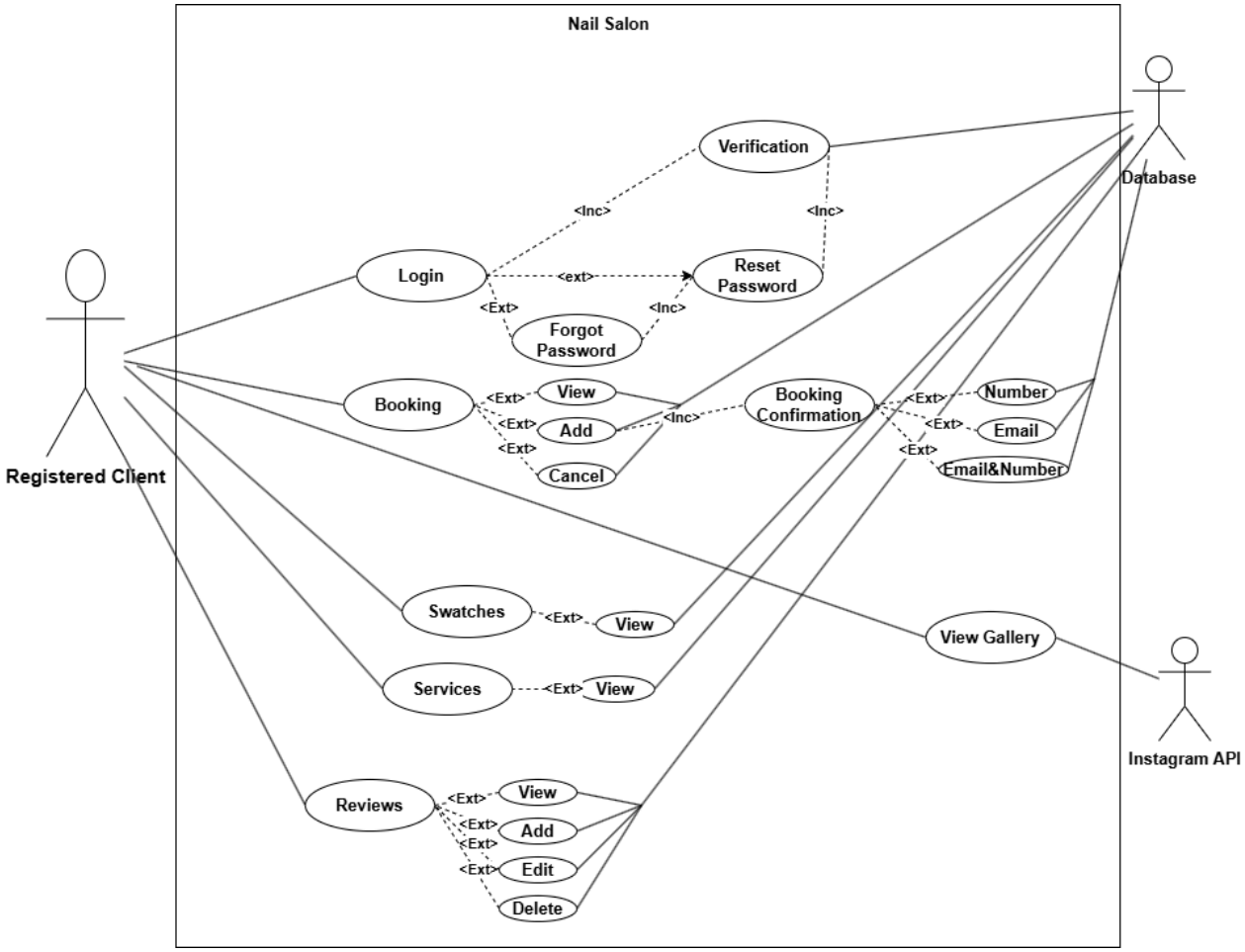

3
Delete Account
View History
Update Email/Password
Password
End
Input Current Password
No
No
Valid Password
Valid Password
Yes
Yes
Delete from DB
Input New Email/Password
No
C
Valid Data
Yes
Update DB
End

Appendix 2 - Use Cases

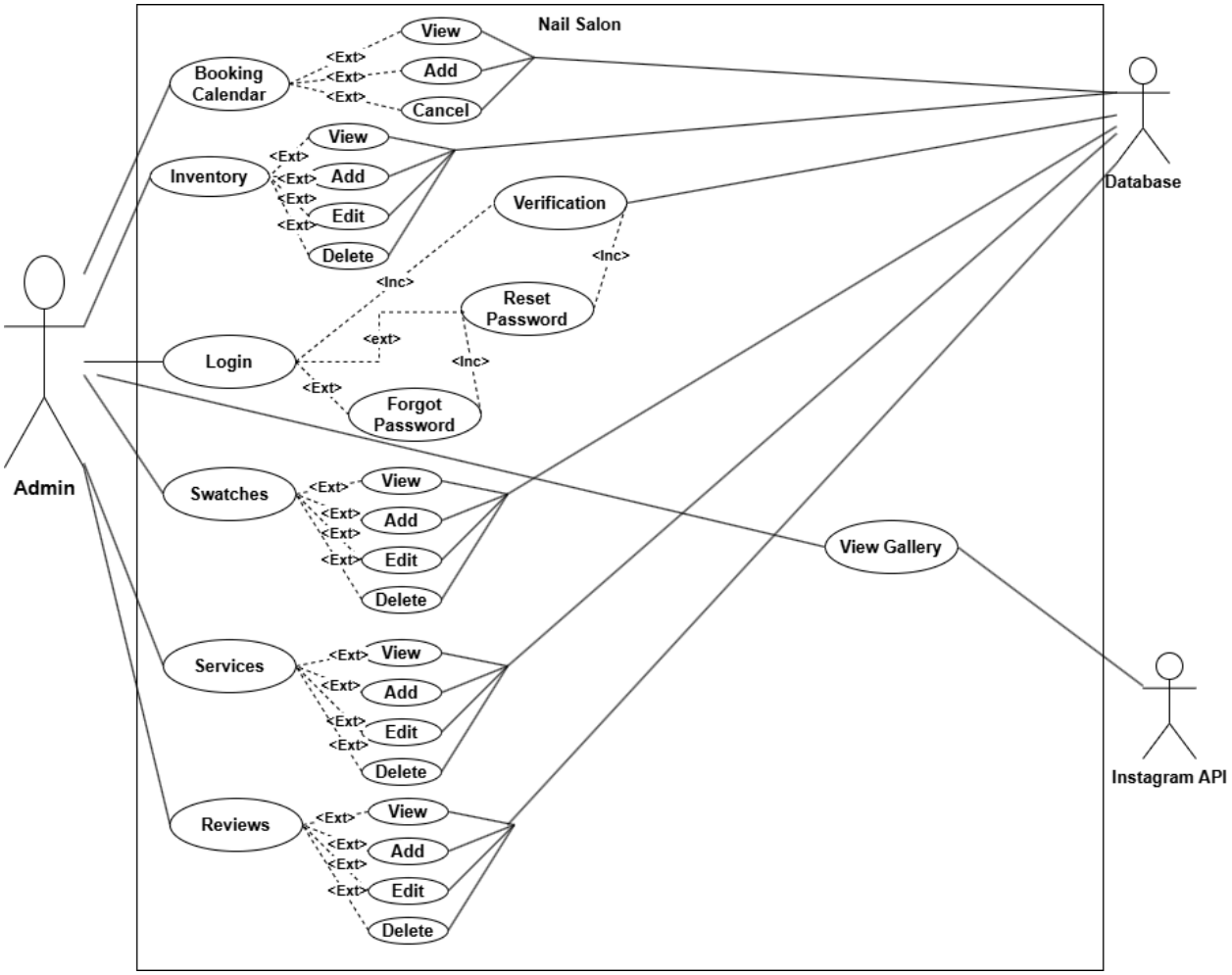
Unregistered Client



Registered Client

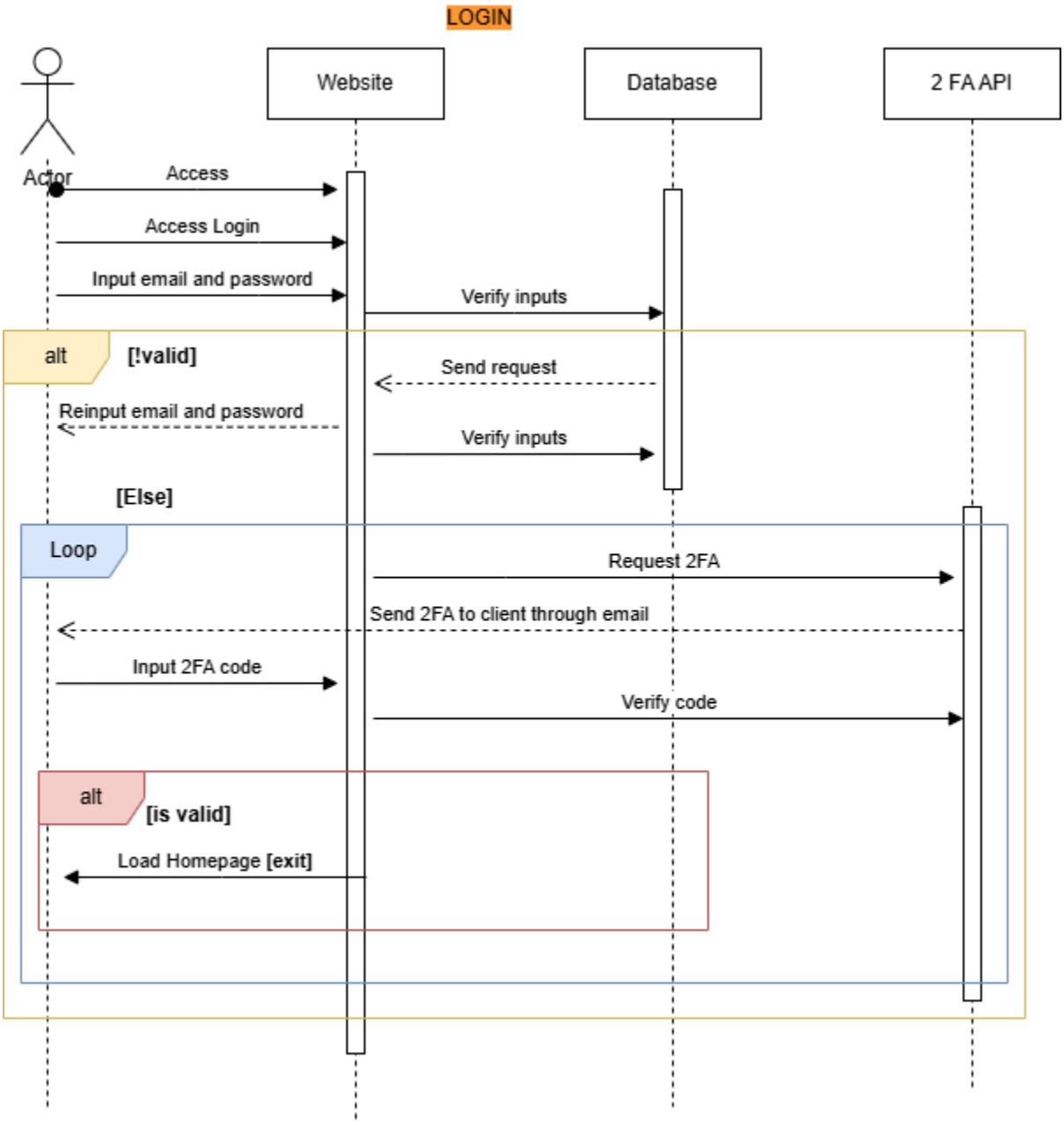


Admin

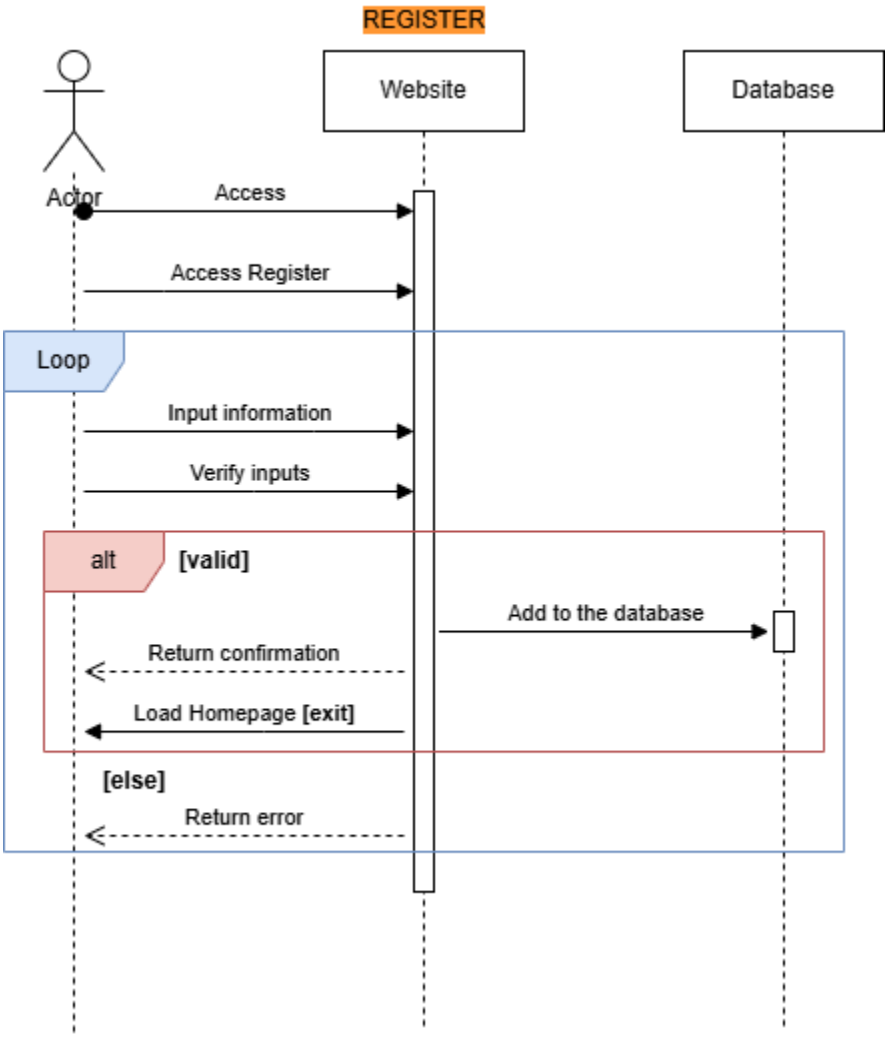

Nail Salon
Admin
Database
Instagram API
Booking Calendar
View
Add
Cancel
Inventory
View
Add
Edit
Delete
Verification
Reset Password
Login
Forgot Password
Swatches
View
Add
Edit
Delete
View Gallery
Services
View
Add
Edit
Delete
Reviews
View
Add
Edit
Delete
<Ext>
<Inc>
<ext>

Appendix 3 - Sequence Diagrams

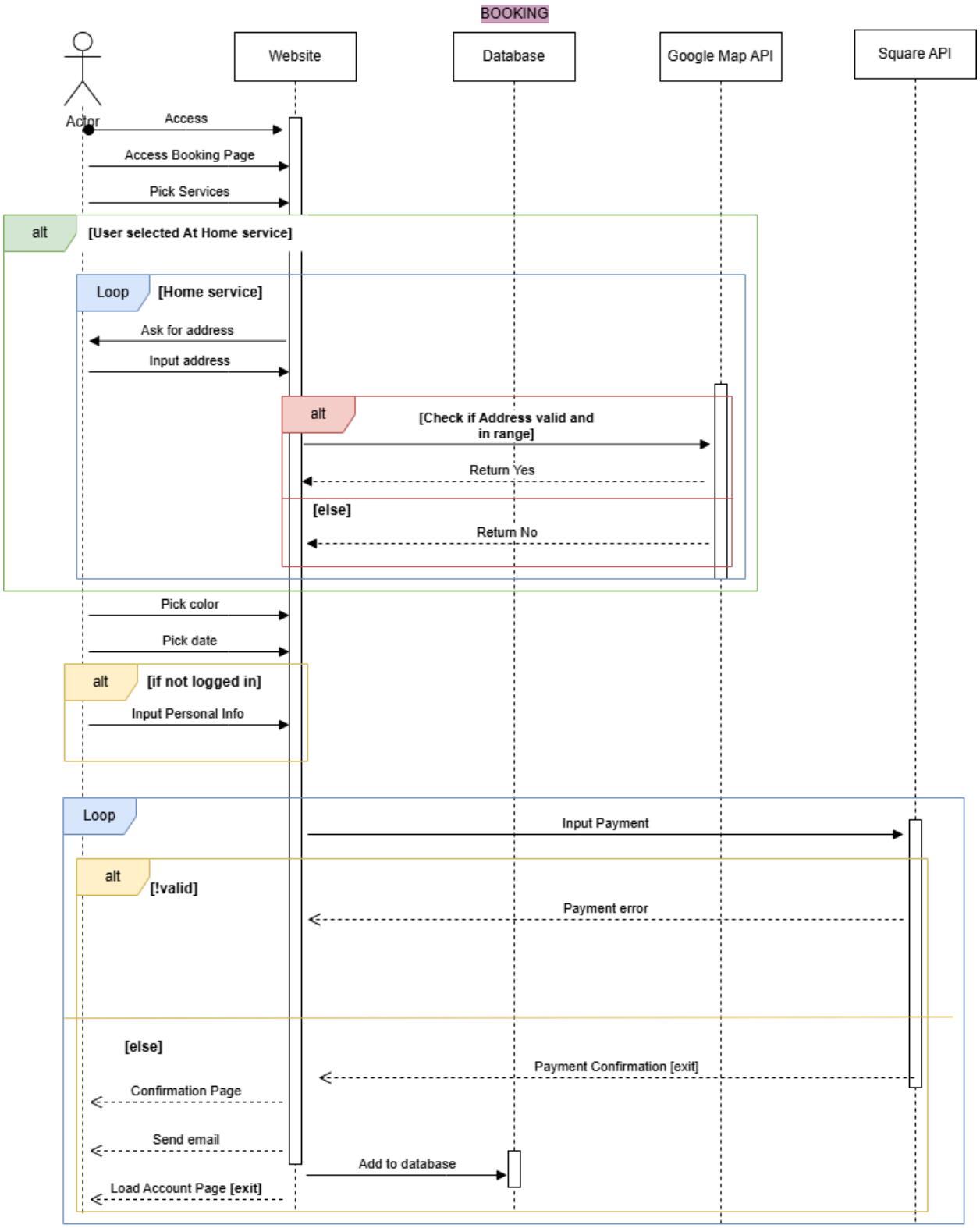
Login



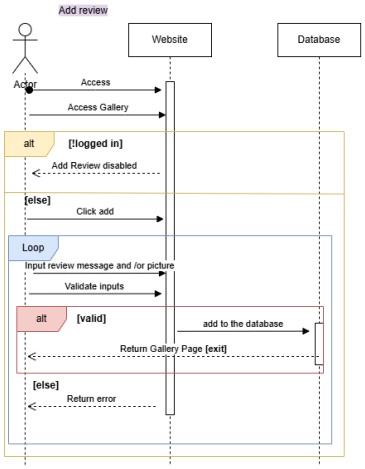
Register

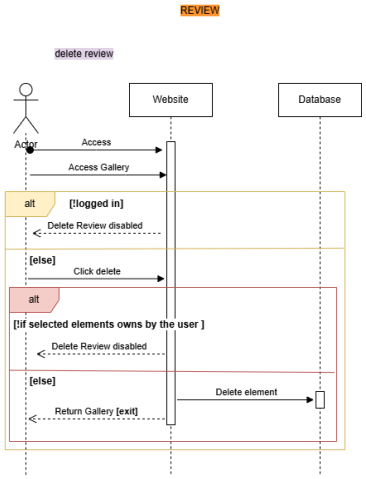

REGISTER
Website
Database
Actor
Access
Access Register
Loop
Input information
Verify inputs
alt
[valid]
Add to the database
Return confirmation
Load Homepage [exit]
[else]
Return error

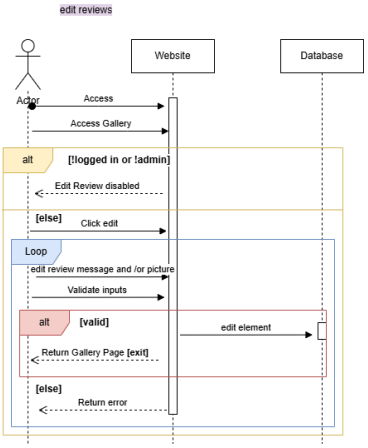
Booking


BOOKING
Website
Database
Google Map API
Square API
Actor
Access
Access Booking Page
Pick Services
alt
[User selected At Home service]
Loop
[Home service]
Ask for address
Input address
alt
[Check if Address valid and in range]
Return Yes
[else]
Return No
Pick color
Pick date
alt
[if not logged in]
Input Personal Info
Loop
Input Payment
alt
[!valid]
Payment error
[else]
Payment Confirmation [exit]
Confirmation Page
Send email
Add to database
Load Account Page [exit]

Review


Add review
Website
Database
Actor
Access
Access Gallery
alt
[!logged in]
Add Review disabled
[else]
Click add
Loop
Input review message and /or picture
Validate inputs
alt
[valid]
add to the database
Return Gallery Page [exit]
[else]
Return error

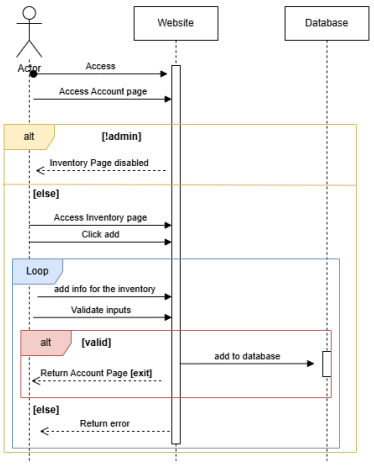

REVIEW
delete review
Website
Database
Actor
Access
Access Gallery
alt
[!logged in]
Delete Review disabled
[else]
Click delete
alt
[!if selected elements owns by the user]
Delete Review disabled
[else]
Delete element
Return Gallery [exit]


edit reviews
Website
Database
Actor
Access
Access Gallery
alt
[!logged in or !admin]
Edit Review disabled
[else]
Click edit
Loop
edit review message and /or picture
Validate inputs
alt
[valid]
edit element
Return Gallery Page [exit]
[else]
Return error

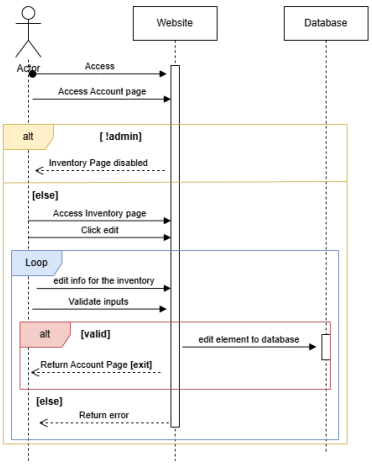
Inventory

INVENTORY [ADMIN]

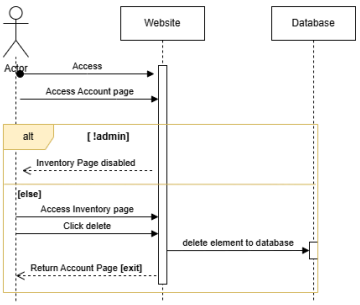
add inventory



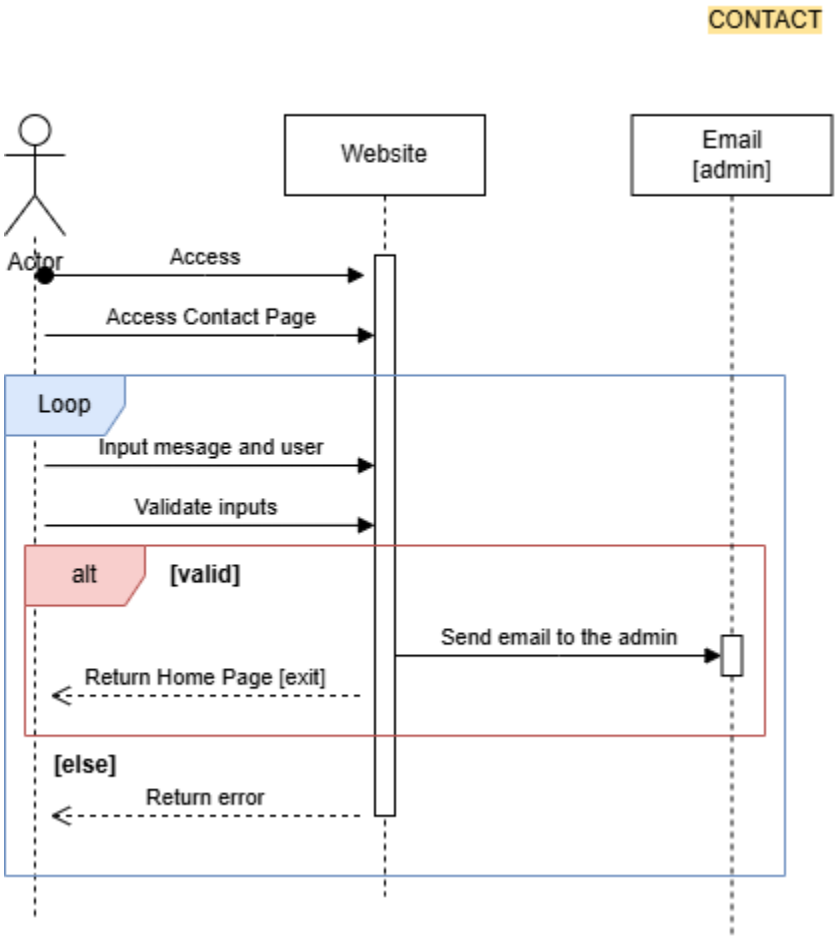
edit inventory



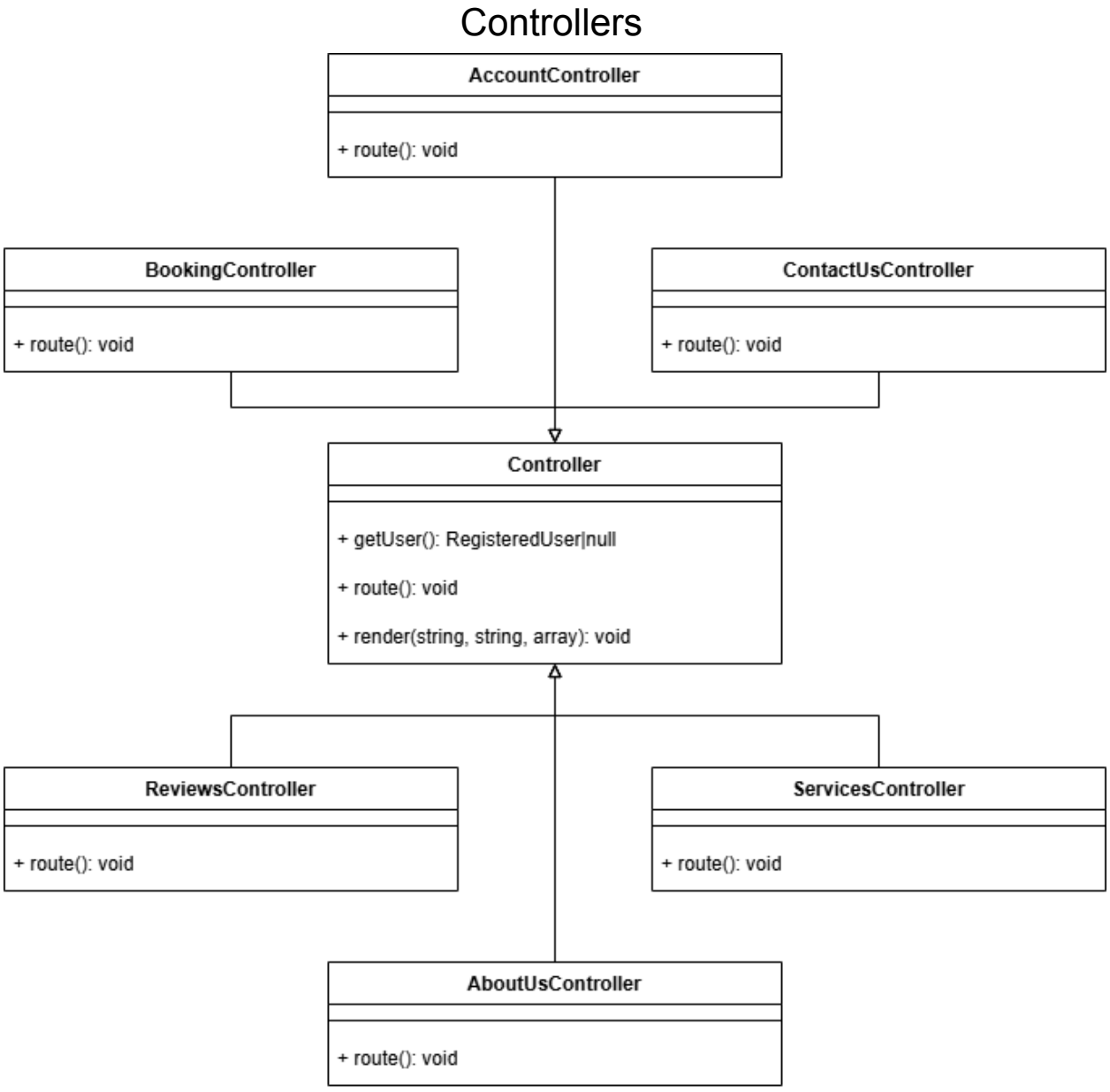
delete inventory



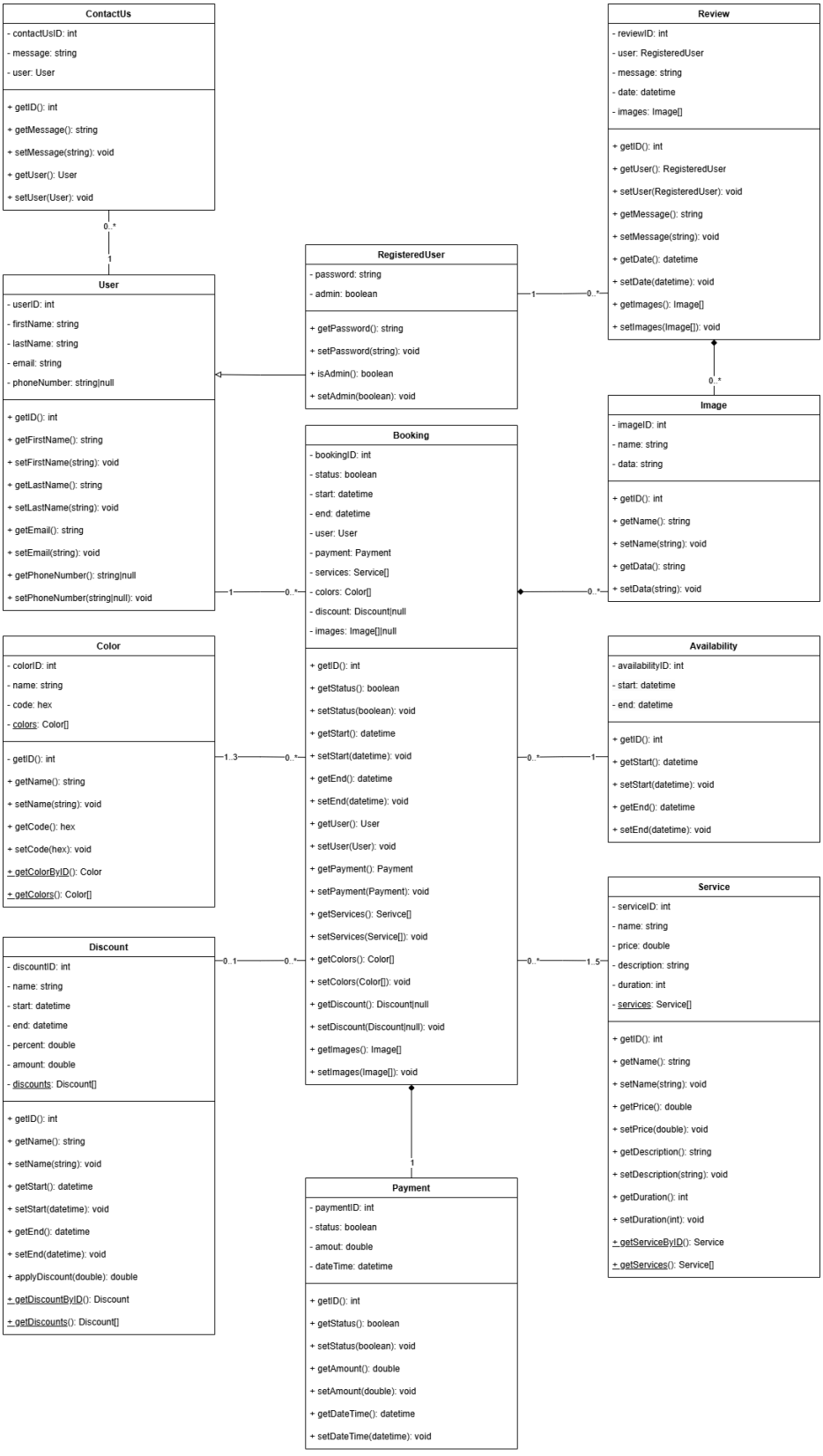
Contact


CONTACT
Website
Email
[admin]
Actor
Access
Access Contact Page
Loop
Input mesage and user
Validate inputs
alt
[valid]
Send email to the admin
Return Home Page [exit]
[else]
Return error

Appendix 4 - Class Diagrams



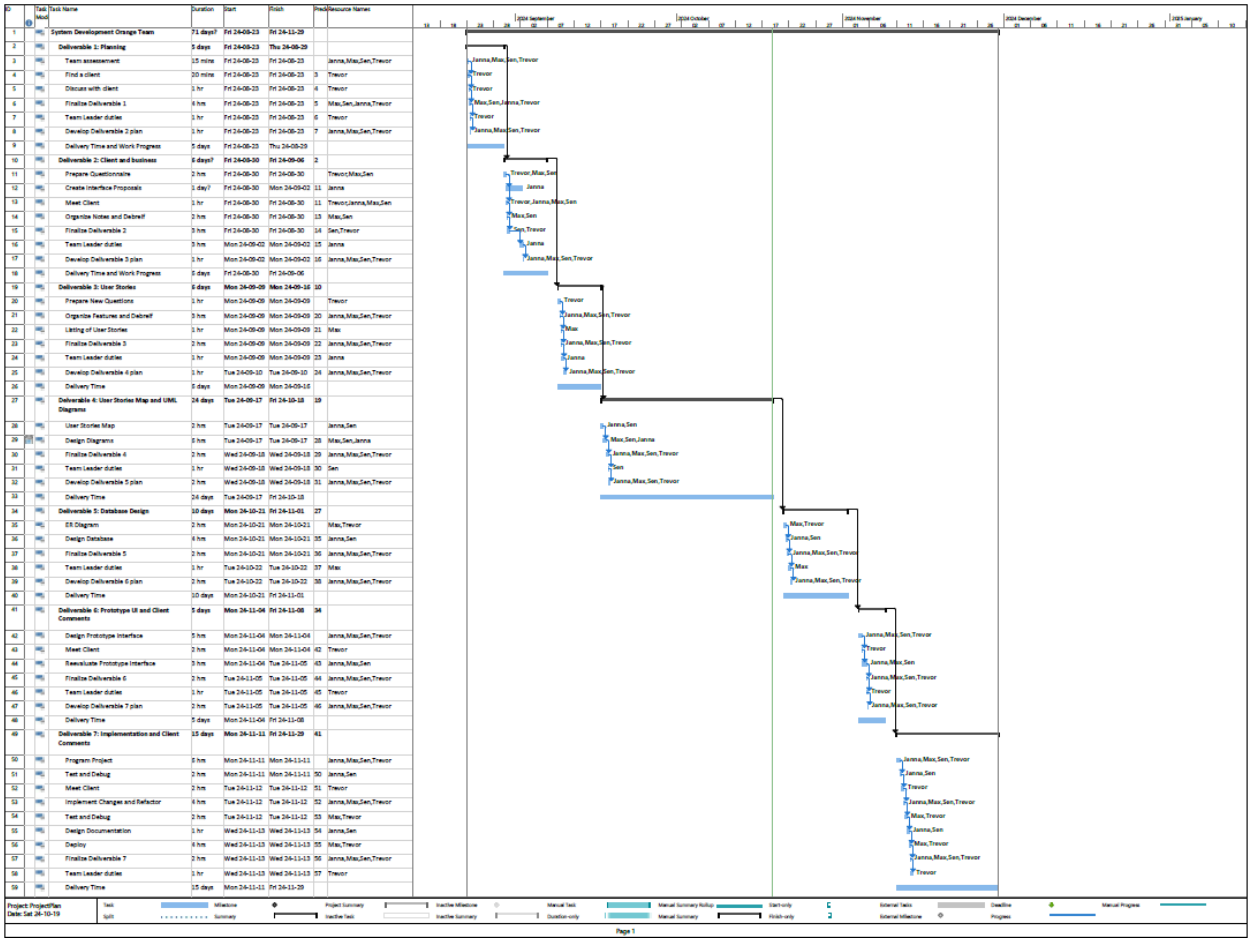
Models


ContactUs
- contactUsID: int
- message: string
- user: User
+ getID(): int
+ getMessage(): string
+ setMessage(string): void
+ getUser(): User
+ setUser(User): void
0..*
1
User
- userID: int
- firstName: string
- lastName: string
- email: string
- phoneNumber: string|null
+ getID(): int
+ getFirstName(): string
+ setFirstName(string): void
+ getLastName(): string
+ setLastName(string): void
+ getEmail(): string
+ setEmail(string): void
+ getPhoneNumber(): string|null
+ setPhoneNumber(string|null): void
RegisteredUser
- password: string
- admin: boolean
+ getPassword(): string
+ setPassword(string): void
+ isAdmin(): boolean
+ setAdmin(boolean): void
1
0..*
Review
- reviewID: int
- user: RegisteredUser
- message: string
- date: datetime
- images: Image[]
+ getID(): int
+ getUser(): RegisteredUser
+ setUser(RegisteredUser): void
+ getMessage(): string
+ setMessage(string): void
+ getDate(): datetime
+ setDate(datetime): void
+ getImages(): Image[]
+ setImages(Image[]): void
0..*
Image
- imageID: int
- name: string
- data: string
+ getID(): int
+ getName(): string
+ setName(string): void
+ getData(): string
+ setData(string): void
Booking
- bookingID: int
- status: boolean
- start: datetime
- end: datetime
- user: User
- payment: Payment
- services: Service[]
- colors: Color[]
- discount: Discount|null
- images: Image[]|null
+ getID(): int
+ getStatus(): boolean
+ setStatus(boolean): void
+ getStart(): datetime
+ setStart(datetime): void
+ getEnd(): datetime
+ setEnd(datetime): void
+ getUser(): User
+ setUser(User): void
+ getPayment(): Payment
+ setPayment(Payment): void
+ getServices(): Serivce[]
+ setServices(Service[]): void
+ getColors(): Color[]
+ setColors(Color[]): void
+ getDiscount(): Discount|null
+ setDiscount(Discount|null): void
+ getImages(): Image[]
+ setImages(Image[]): void
1
1..3
0..*
Color
- colorID: int
- name: string
- code: hex
- colors: Color[]
- getID(): int
+ getName(): string
+ setName(string): void
+ getCode(): hex
+ setCode(hex): void
+ getColorByID(): Color
+ getColors(): Color[]
Availability
- availabilityID: int
- start: datetime
- end: datetime
+ getID(): int
+ getStart(): datetime
+ setStart(datetime): void
+ getEnd(): datetime
+ setEnd(datetime): void
0..*
1
Discount
- discountID: int
- name: string
- start: datetime
- end: datetime
- percent: double
- amount: double
- discounts: Discount[]
+ getID(): int
+ getName(): string
+ setName(string): void
+ getStart(): datetime
+ setStart(datetime): void
+ getEnd(): datetime
+ setEnd(datetime): void
+ applyDiscount(double): double
+ getDiscountByID(): Discount
+ getDiscounts(): Discount[]
0..1
0..*
Service
- serviceID: int
- name: string
- price: double
- description: string
- duration: int
- services: Service[]
+ getID(): int
+ getName(): string
+ setName(string): void
+ getPrice(): double
+ setPrice(double): void
+ getDescription(): string
+ setDescription(string): void
+ getDuration(): int
+ setDuration(int): void
+ getServiceByID(): Service
+ getServices(): Service[]
1..5
Payment
- paymentID: int
- status: boolean
- amout: double
- dateTime: datetime
+ getID(): int
+ getStatus(): boolean
+ setStatus(boolean): void
+ getAmount(): double
+ setAmount(double): void
+ getDateTime(): datetime
+ setDateTime(datetime): void

Appendix 5

Client informed us that there weren’t any documents or forms of those kinds being used.

Project Plan



Bibliography