**Trace & GigIt**

**Software Requirements Specification**

**Version 2.5**

**12/02/2019**

Deepak Sai Krishna Jayanthi,

Subramanya Sai Bharadwaj Gandrakota,

Sowmya Reddy Kumbham,

Nikitha Lakmarapu,

Nandini Yadav Janga,

Leela Krishna Kosaraju

**Submitted in partial fulfillment**

**Of the requirements of**

**CSIS 44-691 Graduate Directed Project 1**

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| **Date** | **Description** | **Author** | **Comments** |
| 9/4/2019 | Version 1.0 | Deepak Sai Krishna Jayanthi | First Draft |
| 9/5/2019 | Version 1.1 | Nikitha Lakmarapu | Introduction |
| 9/7/2019 | Version 1.2 | Nandini Yadav Janga | General Description |
| 9/8/2019 | Version 1.3 | Sowmya Reddy Kumbham,Leela Krishna Kosaraju | Requirement specifications |
| 9/13/2019 | Version 1.4 | Nandini Yadav Janga, Sowmya Reddy Kumbham | Updated new requirements |
| 9/14/2019 | Version 1.5 | Nikitha Lakmarapu, Leela Krishna Kosaraju | UML diagram |
| 9/20/2019 | Version 1.6 | Leela krishna,  Nikitha lakmarapu and Nandhini Yadav Janga | ER diagram |
| 10/7/2019 | Version 2.0 | Sowmya Reddy Kumbham | Product Perspective,User Characteristics |
| 10/10/2019 | Version 2.1 | Sowmya Reddy Kumbham  Leela Krishna Kosaraju | General Constraints, Assumptions and Dependencies,  Added customer prototypes and owner prototypes |
| 11/10/2019 | Version 2.2 | Leela Krishna Kosaraju, Sowmya Reddy Kumbham | Added Use Case Diagram, Functional and Non-functional Requirements. |
| 11/10/2019 | Version 2.3 | Nikitha Lakmarapu | Added Design constraints and Non-Functional requirements. |
| 11/10/2019 | Version 2.4 | Nandini Yadav Janga,  Subramanya Sai Bharadwaj Gandrakota | Technical Manual |
| 12/02/2019 | Version 2.5 | Leela Krishna Kosaraju, Nikitha Lakmarapu | End User Manual |

# Document Approval

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

**Table of Contents Page No**

1. Introduction
   1. Purpose
   2. Scope
   3. Definitions, Acronyms, and Abbreviations -- TO DO--
   4. References
   5. Overview -- TO DO--
2. General Description
   1. Product Perspective
   2. Product Functions
   3. User Characteristics
   4. General Constraints
   5. Assumptions and Dependencies
3. Specific Requirements
   1. **External Interface Requirements** 
      1. **User Interfaces**
      2. **Hardware Interfaces**
      3. **Software Interfaces**
      4. **Communications Interface**
   2. Functional Requirements
   3. **Use Cases**
   4. **Class/Objects**
   5. Non-Functional Requirements

3.5.1. Performance

3.5.2. Reliability

* + 1. Availability
    2. Security
    3. Portability
  1. Design Constraints
  2. **Logical Database Requirements**
  3. **Other Requirements**
  4. Prototypes (for complete project)
  5. Use Case Diagrams

4. Design

4.1. ER diagram

**4.2. GUI**

**5. Analysis Models**

**5.1. Data Flow Diagram**

**5.2. Sequence Diagram**

6. Technical Manual

7. End User Manual

## Introduction

**1.1 Purpose:**

To book an appointment in any business areas, we usually contact them by phone or go there in person. In our everyday life, we often come across such issues. So, our main aim of this project is to make appointments online based on the location and available time slots of those business areas. Application has both perspectives like user can place an appointment and the service provider can accept or reject the appointment request based on their availability.

**1.2 Scope:**

New user opens the application and register using his/her details. After registering, the user needs to give permission to location access. If the user is already registered, he/she needs to login using his/her credentials. Based on the location, for all users i.e. existing and new users the application displays nearby salons where an appointment can be made online by selecting particular saloon. The appointment is fixed once the user clicks the submit button. Later he/she can view the appointments which were booked by him/her. These appointments can be edited or deleted.The application displays the salons which were recently and mostly visited. User can go through his/her profile in settings where he/she can edit his/her credentials including his/her mobile number. If he/she wants to change password he/she gets a confirmation message to his/her registered mail or mobile number and then after confirmation the password is updated.

Service provider starts with registering himself/herself with his/her details and exact location and he/she needs to give permission to location access. Once he/she is registered, the business is visible to the nearest users to book the appointments. Once the user appointment is submitted then service provider can accept based on his/her availability. He/She can edit saloon profile and employee details and his/her availabilities. He/She has the ability to lock the appointment timings where there will be no possibility to book an appointment at that particular mentioned time. So that user can only select the appointment timings which were not locked by the service provider.

User can pay through online after the appointment is confirmed..He/She has option to save his/her card details during payment for future use.

The application can be used within six miles of radius around the globe for users.It displays saloons to the user based on the radius.It can be used within a city with utmost fifty users.

**1.4 References:**

1. <https://squareup.com/us/en/appointments>

## 2. General Description

**2.1 Product Perspective:**

The software app Trace & GigIt is a mobile application for the users who maybe the customers or the owners of the salon.

The customers can book their appointments online and pay for their service. They can search for the nearest locations where they can get their work done. They can also view the costs and make their decision based on their budget. They can also get the availability of the salon and book as per their convenience. The customer get their booking confirmation once owner accept the booking request. Customer will get the notification of their booking and booking confirmation mail.

The application is not only for the customers, it is also for salon owners.

Salon owners can register their salones and provide their service appointments online. Owner can add available timings based on the workers he have in the salon. He can also publish his service cost online. He can accept or reject the appointment based on the walkins he have on his salon.

**2.2 Product functions:**

Initially to use the app user should signup and create an account using email id or phone number. Then user has to login to the application using the credentials. While installing the application user has to give permissions to access the location. Based on the location, all the nearby salons will be displayed.

By clicking on the interested salon, user can see all the details of that particular salon including the address, contact details, working hours, available slots to book an appointment. Unavailable time slots will be disabled and no action will be performed when the user clicks on those slots. User can select a particular time slot and request the booking. Then the request will be sent to the salon where they can either accept or reject the request. Once the request is accepted , user’s booking is confirmed. Once your booking is confirmed or had your hair service done, you can pay the amount through the application. For this you give your card details and has an option to save the payment for future uses for easy payment.

**2.3 User Characteristics:**

The user characteristics are the abilities and the characteristics that are affecting user performance are called user characteristics.

|  |  |  |
| --- | --- | --- |
| **User Characteristics** | **Description** | **Values** |
| Role & Task | Tasks by both customers and owners like login, signup, verification | Commander |
| Age | Users of different ages have different abilities and different levels of memory, comprehension. | Preferably above 15 |
| Gender | Theme of the application. | Male & Female |
| Education Level | Knowledge acquired and reasoning abilities. | High school and above |
| Culture background | Culture related to application content like vocabulary and signs. | Good |
| Interest | Interest related to physiological preference. | Business and service |
| Profession level | Comprehensibility and operant level of operation platform PT | Good |
| Computer skills | Effective use of system. | Very good |

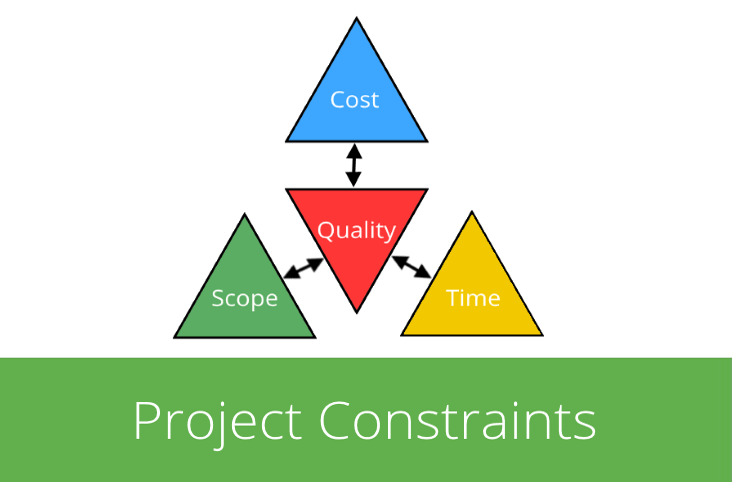
**2.4 General Constraints**

Constraints are limitations to the project, such as the limited availability of resources like time and funds. The project is dependent on various internal and external aspects.

The basic constraints for the project are Scope, Schedule, Budget/Costs, Quality, Resources, Risks and these are the external aspects of constraints.

The internal aspects of design constraints are like internet, location, payments and notifications enable.

Both external and internal constraints of the product will affect the product life cycle.



**2.5. Assumptions and Dependencies**

Assumptions:

* Location must be on to know the present location and nearest salons.
* Internet is required for the application.
* The time taken for the task completion like OTP and verification mail.
* System constraints like the mobile version and RAM.
* app will be available in the google store.

Dependencies:

* Server and user must be in the same network.
* Second 10000 requests can be handled.
* iP address must be given in the application.
* Location must be on to know the present location and nearest salons.
* Internet is required for the application.
* Only registered salones can be viewed in the application.
* Application is installed through APK file.

**3. Specific Requirements**

**3.2 Functional Requirements**

Trace & GigIt is an application where the users can get appointments from the salons online. Here, saloons are the clients and application is the link to make the appointments for users. Presently there is no existing platform to make the appointments and to give these services to assist the clients business. So, the main purpose of the application is to make the client business easy.

Application is build according to progressive enhancement principles. Responsive design for varying screen size, Version compatibility for devices, more touch flexibility and less keyboard usage in the application are the features of application. The other main feature is application and data security. In app payments are also available in the application user can pay after confirming the appointments or after having the service.

**Requirements:**

* User must sign up with email or phone number.
* To use the application, one must login with appropriate details.
* If user forgot password then he can reset it with phone number or email.
* Salon must register and sign in to get the appointment requests.
* Salon manager must update the employee availability.
* Salon manager must post their prices for the different services they offer.
* The user can able to find the nearby saloons with the help of mobile gps.
* User can view the saloon details and the prices of the services they provide.
* The user can book an appointment by just clicking on appointment time slot in their interested saloon.
* Salon manager must accept the request and allot the user to a certain employee.
* The appointment must be updated in the user calendar and must get the notification on the day of appointment.
* User must book their appointment one day prior.
* If the user does not show up for the appointment, then the app waits for at most 15 minutes and then cancels the appointment.
* We should also store client details such as saloon address, prices of their services, employee details and their available slots in our database.
* User can do payments in the application once he book the appointment.
* There is a payment option in the application from where he can do payments.
* All the types of payment methods are available in the application and user can even save the payment methods.

**3.5 Non-Functional Requirements:**

* This application must be able to install in any android device with version greater than or equal to 5.0
* Android devices must have active GPS and Internet services.
* The user should give access to gps once the app gets installed in his/her android device.
* This app should be user-friendly
* This app should have less response time
* All the sensitive information will be encrypted.
* The user can access only a few saloons that are registered with this app.
* User can even pay manually at the saloon.
* we should have a database to store all the data such as user information and all their appointment details.
* This app should find all nearby salons.

**3.5.1. Performance:**

* Less battery consumption while using app.
* Managing memory by keeping track of each memory allocation by Dalvik Virtual Machine.

**3.5.2. Reliability:**

* 10000 requests can be handled per second.
* App takes less than 100ms API response time.

**3.5.3. Availability:**

* It displays the saloons within 6 miles radius from the current location.
* User is displayed with a week schedule to make appointments.

**3.5.4. Security:**

* Strong security that balances great encryption and good performance.
* Restricting access to sensitive permissions making app less vulnerable for attackers.

**3.5.5. Portability:**

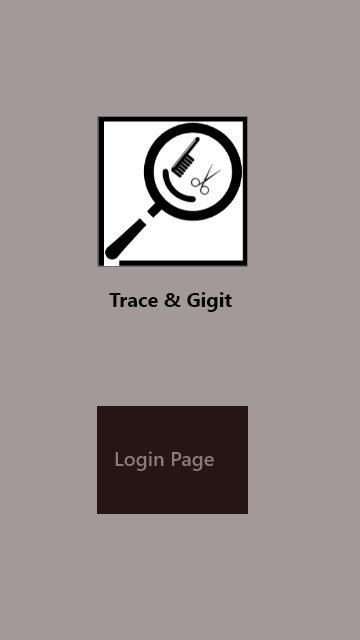
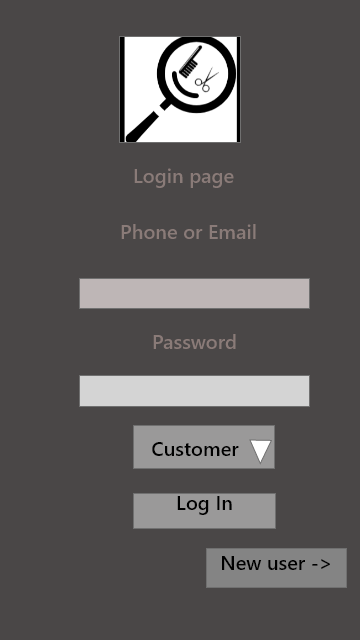
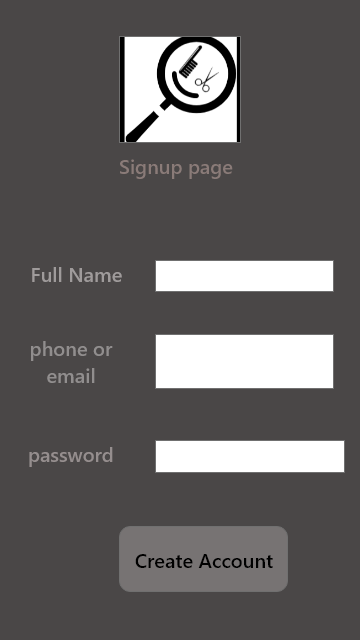
* App can be used in mobile devices and tablets.
* It works only in android devices.

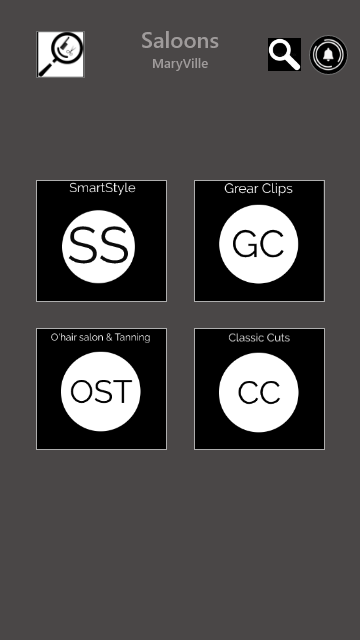
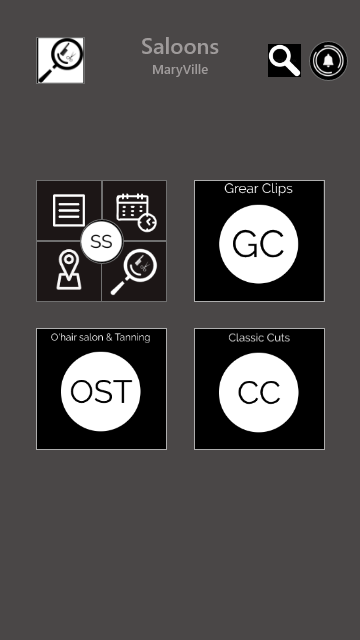
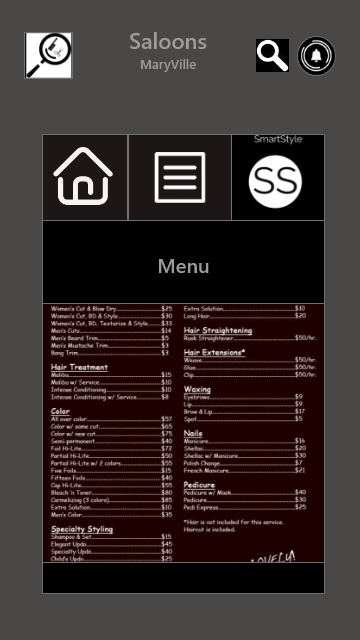
**3.7. Design Constraints:**

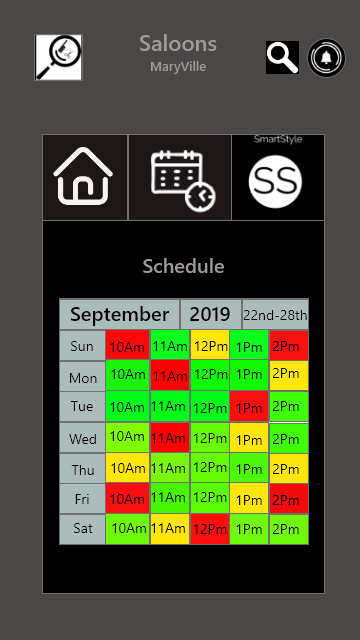
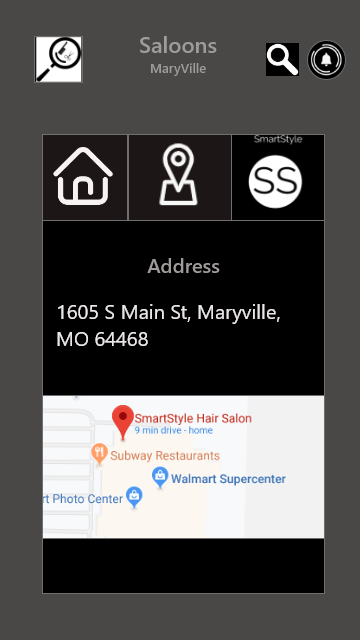
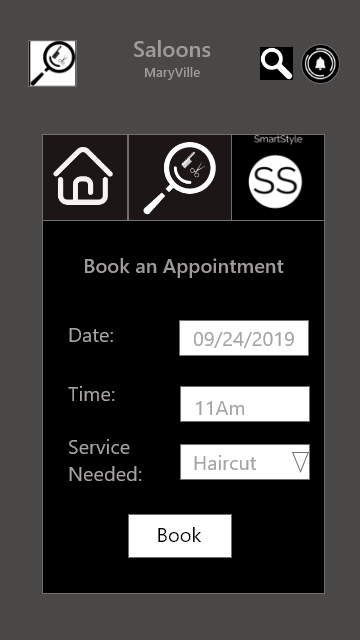
* Server and user must be in the same network.It makes users difficult to use application.
* GPS is required to know the current location of user.
* Internet is required for the application.
* Right now application can only be installed through APK file.

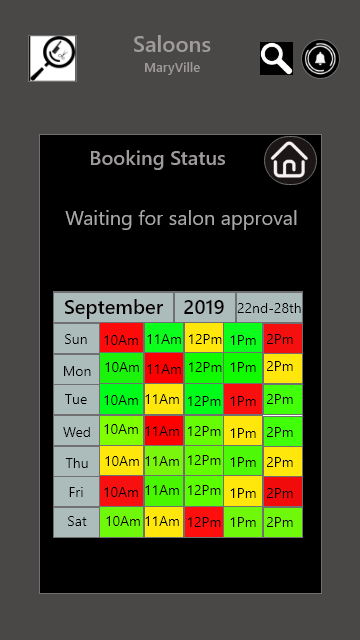
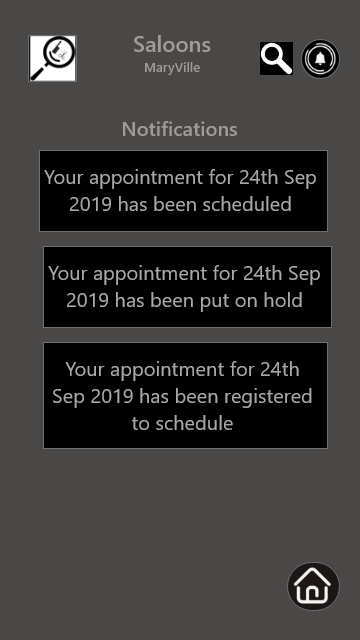
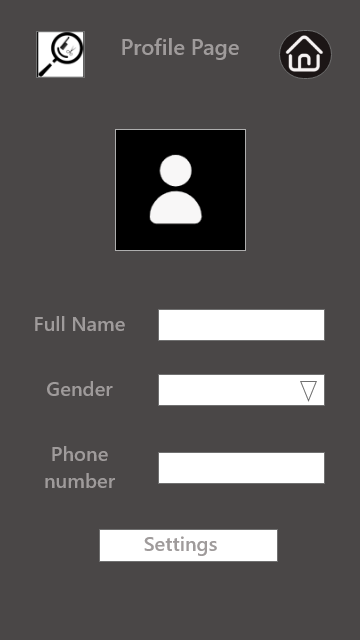
**3.10. Prototypes:**

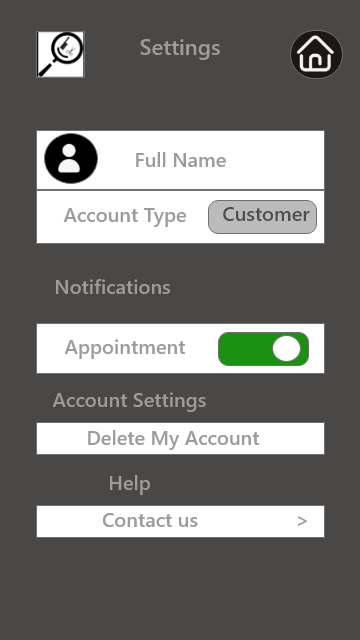
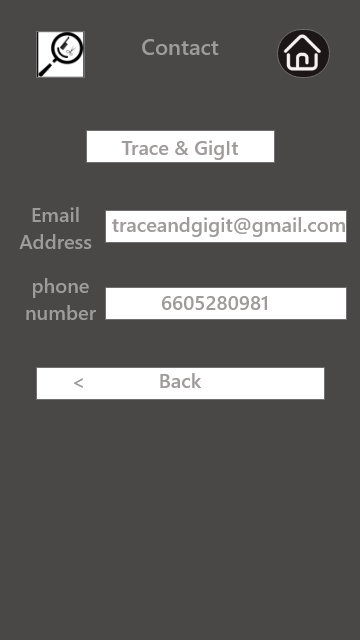
**Customer prototypes:**

**  **

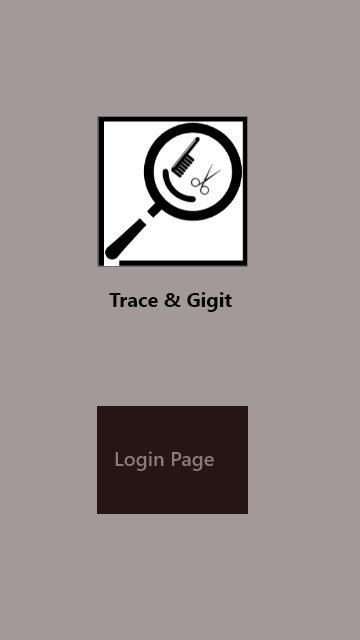
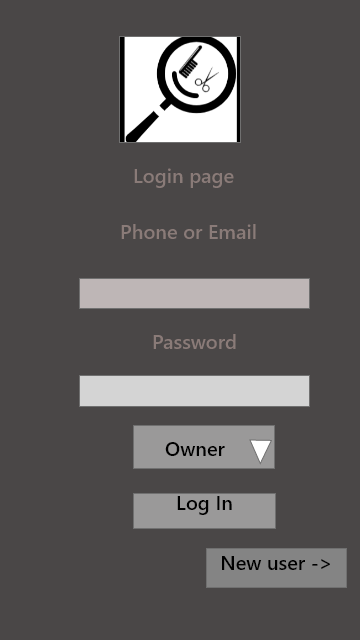
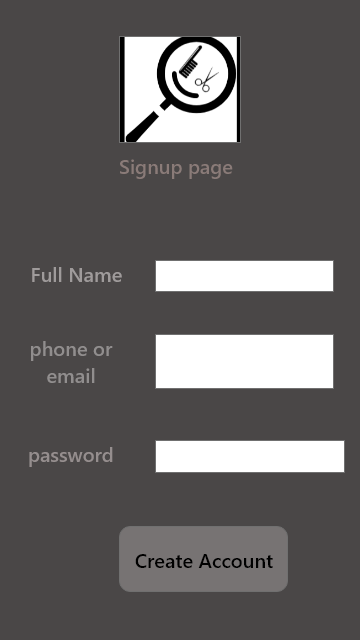
**  **

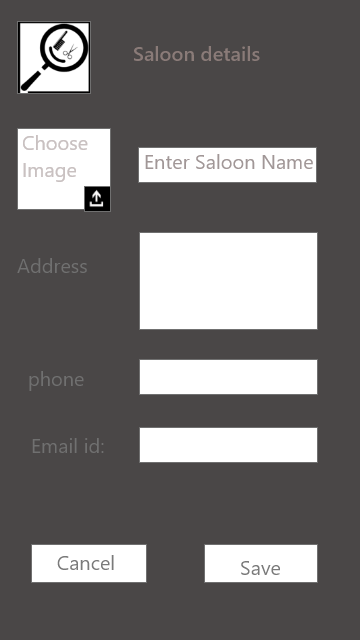
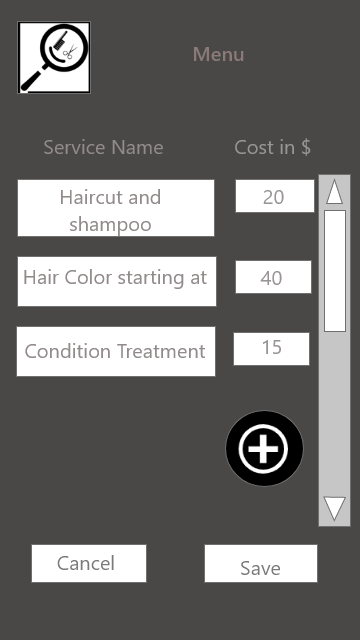
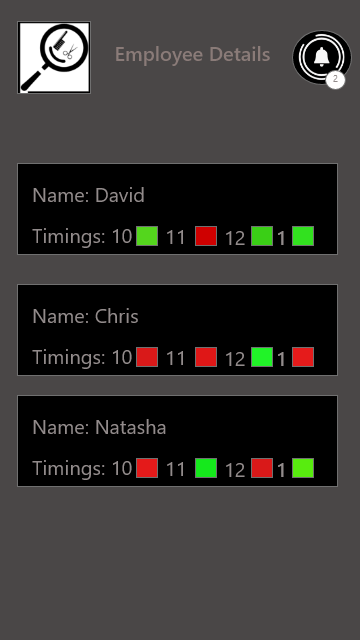
**  **

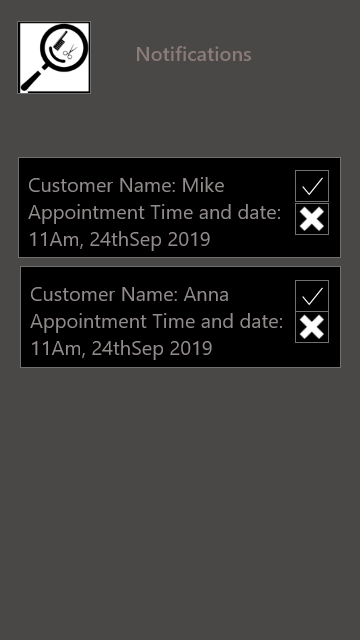
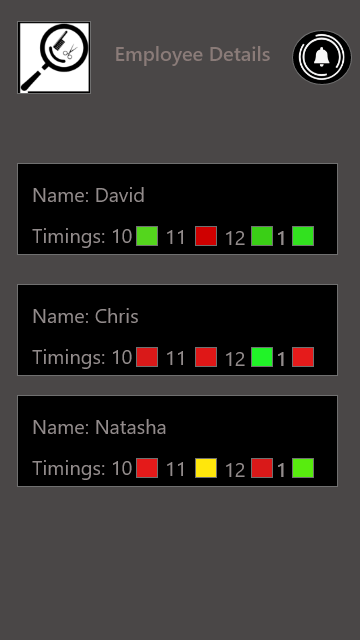
**  **

**  **

**Owner Prototypes:**

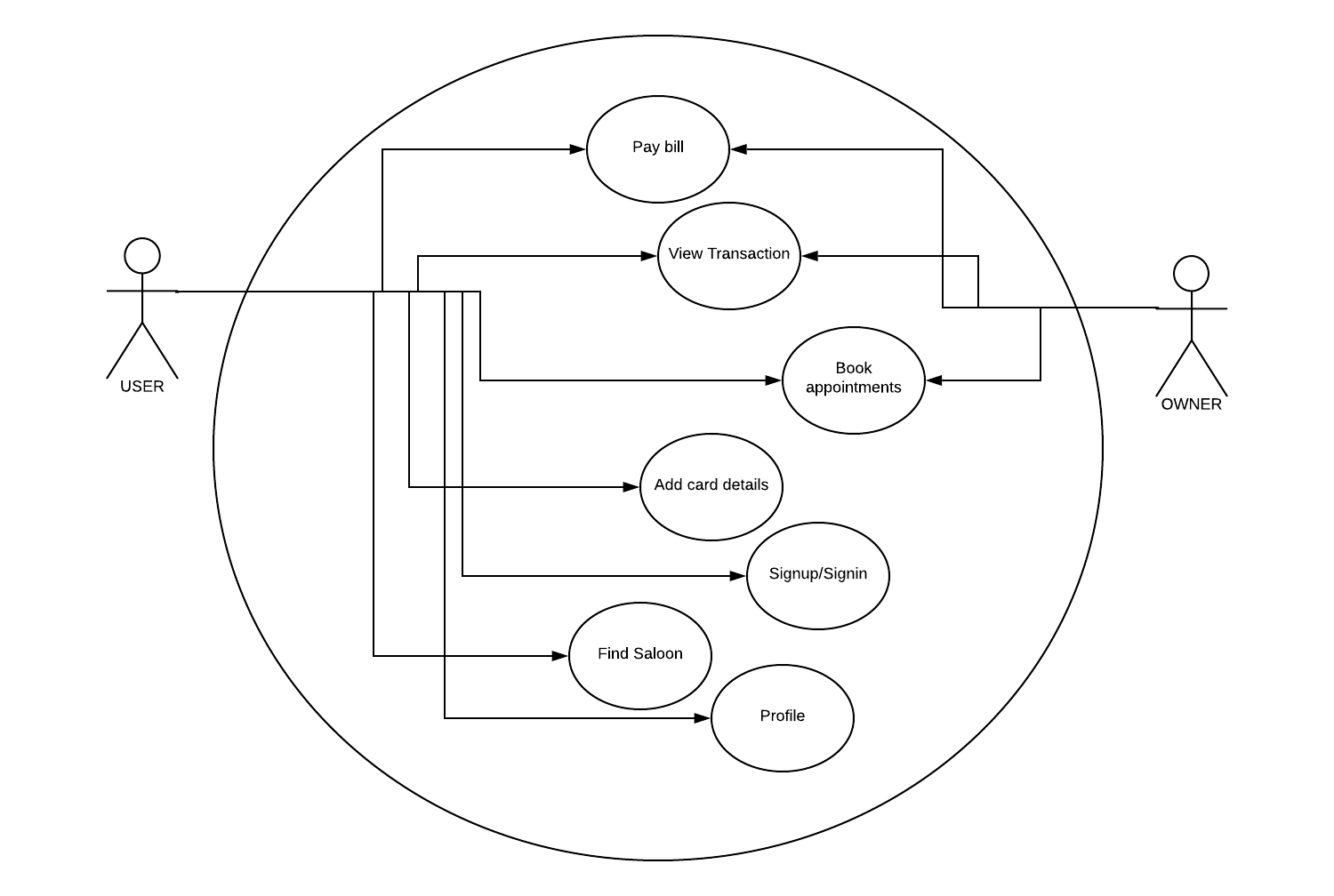
**  **

**  **

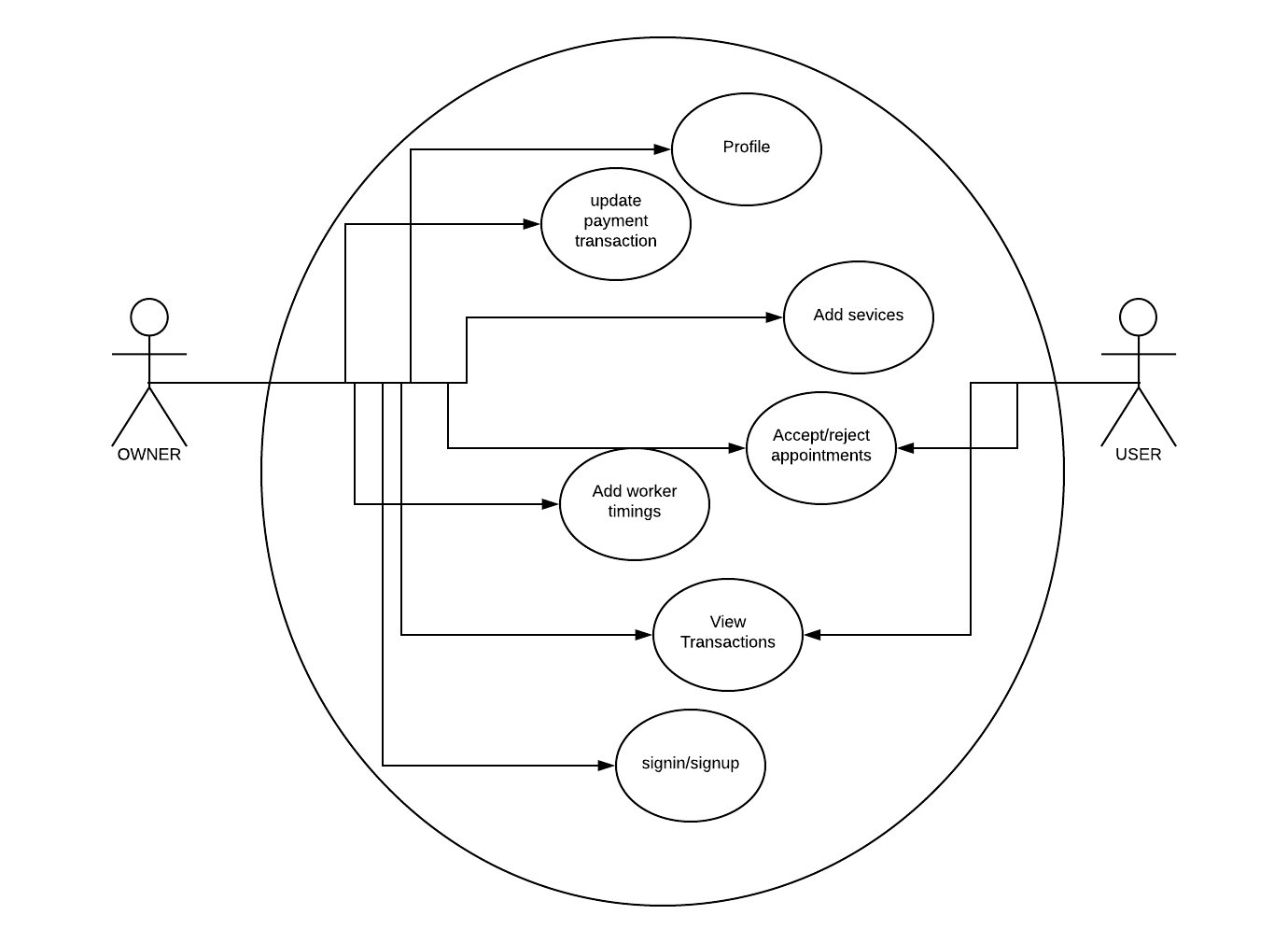
** **

**3.11 Use Case Diagrams**

1. **Customer Use Case diagram:**

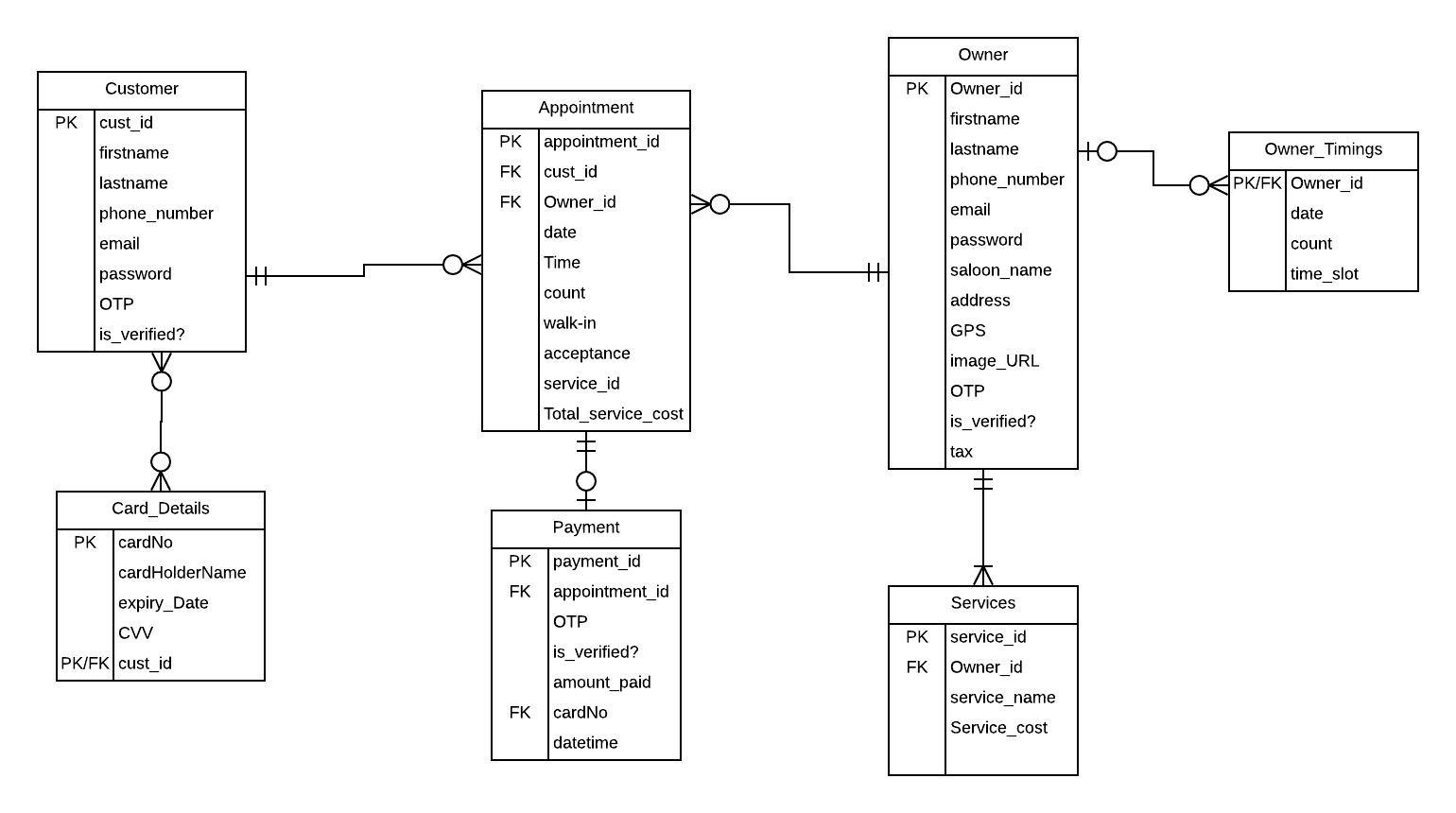
****

1. **Owner Use Case diagram:**

****

**4.Design**

**4.1. ER diagram:**



**6 Technical Manual:**

1. **On App launch :-** On the application launch, device details are retrieved and an API call is made to store to store the device details of the user on which the application is installed, an API call will be made to store them where it expects the device make and operating system and a client secret associated with the device os etc. Once these details are validated then these details are stored in the back end database for restricting the multiple logins of the user. then a session will be created for the device for further communication with APIS and created device is appended to respective sessions.
2. **SignUp**:-

If the user is the first time user, then he/she has to register to get the services by providing basic profile details like name, email, password etc. If the password and confirm password fields match then signup API is called. In the process of creating a user this API will expect a valid session which will be generated from the previous API, and user profile details based on these details, check the user is already existing or not if exist a user already registered response will be given and updates the session details or else a user password will be encrypted and saved along with user profile details and created user is appended with the session

1. **SignIn:-**   
    If the user is a returning user, they can directly login by entering the valid email id and password. To validate the credentials, sign in API is called once the user clicks on the signin button. If this API call is made we can assume that the device is already registered and user is already created it takes an input of email and password which should be get matched with details while creation of the user in previous API as when get log’s out from the application the session will updated with user as null, on successful sign in the session will appended with the logged in user if the user tries to login without registering the API response will be such that user should be created before trying for sign in. on successful sign in the user is owner/ customer is conwayed to front end in response
2. **Home Screen :-** If the user is the customer who wants to get the service, then after signing in customer can see all the available salons nearby. If the user is the owner of the salon, he will see the schedule of his employees after signing in. Till now the owner and customer are treated as single user, from now the segregation takes place between owner and customer if the user is owner the list of employees for this owner is displayed along with their timings and availability from this screen an owner can edit employees details and their timings and accept or reject bookings if owner accepts the booking have to allocate it one of his employees if not there will be no change on the screen. if the user is a customer list of the nearby salons are displayed from this screen a user can request for an appointment and edit his profile and look for previous transactions and save his card details etc
3. **Forgot Password:-**   
    If a user forgets his password while logging in, then the application gives a chance to retrieve it. For this user has to click on the forgot password button which calls the forgot password API. In the API a third party is being used named send grid saying it is used to send OTP via email to the user if both the codes gets matched (user entered and generated code) the user will be allowed to update his password where after successful update then the session will get updated and password as well.

**7.End User Manual:**

1. Download the app apk file from google drive.
2. Allow permissions to install the app from unknown resources and Install it in your mobile phone.
3. Open the app and allow permissions to get the location of the user’s device.
4. If you are a new user, click on sign up button, enter all the details and click on submit button.
5. If you are an existing user, login with your credentials.
6. If you forget the password, click on “forgot password” button.
7. Enter the OTP that has been sent to your email address, reset your password and login.
8. Once you are logged in, home screen is displayed.