**CAB SERVICE MANAGEMENT**

**SRS (Software Requirement Specification)**

**Introduction:**

The Cab Service Management system is the online service which will automate the process of booking a cab and will facilitate both the user and the driver with reduced time and efforts. Both driver and user can register into the system with several information's. The user can login to the system and request for booking a cab on their required date and time, providing all necessary information. The fare will be calculated and user should confirm it. The rider enters their pickup location and destination into the search box then confirm the booking. A nearby driver can view and choose to accept the request. Then the driver can provide service to the user on the specific date and time. Finally, the user will have an opportunity to give feedback for the service they got. The driver can check it and take appropriate action for the future improvements.

**Purpose:**

The Cab service system is a mobile application that facilitates cab service reservations through online. Users may reserve cabs in advance online, streamlining operations and ensuring drivers and passengers are always on the same page. The purpose of this SRS document is to specify software requirements of the Cab Service Management System. It is intended to be a complete specification of what functionality the system provides. The main purpose of the system is to automate the process of booking a cab.

**Scope:**

The global demand for online cabs is currently at an all-time high. Rather than catching a running cab service, people prefer to travel via pre-booked cabs. This is a profitable industry. The online cab booking system has truly struck a chord with smartphone users all around the world. Here we will be able to combine the greatest travel app features into the final product.

**Modules:**

* **Admin:**

Administrator can login to the system using username and password. They can approve the drivers after verifying their details and review the cab details anytime. They can verify the payments and can provide invoice to the customers. Admin can view feedbacks given by customers to the drivers and ratings to the app from the customers.

* **Customer:**

Customer can register to the system and log in to it to search for nearby cab. After reviewing cab details, they can book for the cab. They can provide feedback to the drivers for their service and ratings to the app.

* **Driver:**

Driver can register to the system and after logging in they can view requests from customers for cab service. They can accept the requests and can provide the service. Feedbacks from customers can also viewed.

**Functional Requirements:**

**1. Registration:**

The customer and driver should create their own username and password. The data that should be collected from both are different. In this case we need to create separate registration for driver and customer.

Driver registration [Button] {Redirects to driver registration page}

Customer registration [Button] {Redirects to customer registration page}

**1.1 Driver Registration**

The driver needs to upload many details and generate a username and password.

* First name [minlength(1) and maxlength(30)]
* Last name [maxlength(30)]
* Date of birth [date] {required}
* Gender [drop down options (male, female, others)] {required}
* Mobile number [phone] {required}
* Email [varchar] {required} (primary key)
* License number [number] {required}
* License [File upload] {required}
* Cab registration number [File upload] {required}
* Registration details [File upload] {required}
* Create password [password] {required}
* Confirm password [password] {required} (should match the create password option)
* Submit [button] {Redirected to login page}

**1.2 Customer Registration**

The customer should register the first time they enter the application. They are required to provide their name, mobile number, email and create a password.

* First name [minlength(1) and maxlength(30)]
* Last name [maxlength(30)]
* Date of birth [date] {required}
* Gender [drop down options (male, female, others)] {required}
* Mobile number [phone] {required}
* Email [varchar] {required} (primary key)
* Create password [password] {required}
* Confirm password [password] {required} (should match the create password option)
* Submit [button] {Redirected to login page}

**2.Login:**

The customer can log in to the application by providing their email and password. There will be a forgot password link where the user can reset the password.

* Email [varchar] {required} (primary key)
* Password [password] {required}
* Forgot password [link] {It will help the users to recover or change their passwords. This link will redirect into password reset page.}
* Submit [button] {If the role is user redirect to home page of user, if it is driver redirect to driver’s home page}
* Not Registered[label]
* Register [link] {If you are not a registered user this link will redirect to the registration page}

**2.2 Admin login:**

The admin can log in to the application using username and password.

* Username [username] {required}
* Password [password] {required}
* Submit [button] {Redirected to admin home page}

**3.Password reset:**

Help the user to reset the password.

* Email [varchar] {required} (primary key)
* Submit [button] {After the user click on the submit button, they will get a password recovery mail which redirect into password reset page 2 which contains new password [password] {required} and Confirm password [password] {required} (should match the create password option).}

**4.Customer Home:**

Once a customer is logged in, the customer will land on to the home page. The home page has many different functionalities. Search feature for pickup and destination locations. While searching suggested locations will show up and user can pick up from it. After selecting the locations estimated fair will be shown up. The user can confirm after verifying the fair for booking. After booking a status page will show up.

**5.Driver Home:**

Once the driver got approval from admin they can log in to the system. When a nearby user requests a ride, the request will show up in the driver interface. The driver may choose to accept or reject the request. If the driver does neither, the request will stay there until another driver accepts the ride or the ride gets cancelled.

After the driver accepts the user request, they can view user details such as their name, contact number, pickup location and destination.

**6.Admin Home:**

Once the driver logs in, they will move to the home page.

* Admin can view the details of drivers who register in the application and can verify them. After verification they can approve them.
* Admin have access to all the cab details.
* Once the customer books a cab admin can view its booking details and also which driver accepted the order.
* Admin can manage the available routes.
* Admin can view the feedbacks given by customers to drivers.
* Admin can view ratings given by customers for the application.

**7.Status:**

In the status page the customer can view informing you that your booking is confirmed along with the cab details.

**8.Payment:**

The user may choose among the available payment options including direct cash payment. If the user chooses online payment the site connects the with the bank API and redirects to the payment gateway. For payment using card access collect information's like card number, card holder’s name, and mailing address, expiration date. A transaction record in the form of the mail.

**9.Driver status:**

After dropping the customer on their destination, the status page in driver side shows the message “Your order is completed successfully”

**10.Feedback & Rating:**

After the ride the customer can provide feedback to the drivers and they can also provide ratings to the application.

**11.Invoice:**

After the payment an invoice should be generated including order id, name, fare and other tax details.

**Non-Functional Requirements:**

* Performance Requirements: The response time should not vary with increasing the size of the data storage.
* Security Requirements: The application should not modify any system files.