Design Document

Taxi-Service

10/31/2024

Version 1

Avery Killian, Carter Gibb, Gia Nguyen

Taxi Service Application

GitHub Repository: https://github.com/AveryKillian99/342-Group6/tree/final

SRS Document

1. General Description

This software will provide a taxi-like service, similar to apps such as UBER and LYFT, to allow users to request pickup and drop off at specific locations and have drivers deliver them to that location.

2. Features

This software is designed to allow customers to easily and safely get a ride to a location, with a driver they can feel comfortable with, while providing providers the ability to make some money by accepting a customers request, picking them up, dropping them off, and getting paid based on their speed, safety, and professionalism.

Two way Rating and Review: A rating and review section applicable to both customers and providers so that both parties feel safe and comfortable interacting with the other.

Report System: A system dedicated to handling reports of extreme measures for either customers or providers to add another level of safety and security for the parties.

Interactive Map and Location Service: The customers and providers who use our map will have access to a built in location and gps service to make pickup and drop off as easy as possible for all parties.

3. Functional Requirements

-FR0: Allow users to create profiles as customers or providers

-FR1: Allow users to see currently active customers/providers

-FR2: Allow customers to request a ride to a location

-FR3: Allow drivers to receive ride requests

-FR4: Allow drivers to accept or deny requests

-FR5: Allow users to view respond and create ratings/reviews for themselves and others

-FR6: Allow customers and providers to make reports regarding company policy

-FR7: Allow System Admins to view reports

-FR8: Allow System Admins to punish appropriate parties

-FR9: Accounts are protected with logins

-FR10: Providers will be verified by System Admins

-FR11: Accounts will be created separate for Admins

4. Non-Functional Requirements

-NFR0: Current and accurate locations of users will need to be kept up to date as they move and change locations.

-NFR1: User reviews and ratings of others

-NFR2: Users will only be shown services available in their immediate area by default

-NFR3: Relevant information about the service a user has selected will be shown in less than 10 seconds

-NFR4: Users will be prompted to ask if they want to write a review for a driver after the ride happens

5. Scenarios
    a. Customers: Carter Gibb
        i. Location Service:
        - **Initial Assumption:** The customer is logged in and can interact with the site. Home page is a local map displaying themself and nearby active drivers
        - **Normal:** The user will look for an active driver near them and select them, moving to making a request
        - **What Can Go Wrong:** The user could select the wrong driver or decide against the one they selected, so a back button will be provided to return to home page
        - **Other Activities:** The user can press refresh to update the locations of them and any nearby drivers
        - **System State on Completion:** There will be a driver's information displayed and the user can decide whether or not to request a ride

        ii. Ratings and Review
        - **Initial Assumption:** The customers has logged in and can view their profile and others profiles and request service
        - **Normal:** The users can view ratings and create ratings/reviews of their own and respond to others reviews
        - **What Can Go Wrong:** The users could leave a rating/review for the wrong person without meaning to, a user can delete a comment or rating of theirs if needed
        - **Other Activities:** A user can look at extended reviews such as viewing a response made to another comment and the response to that etc..
        - **System State on Completion:** A new review will be left and added to the database for that account

        iii. Report System
        - **Initial Assumption:** A user is logged in and can interact with the page and view ratings/reviews and request services
        - **Normal:** The user can interact with a report button for drivers/customers and be given a small popup allowing further details of the report and the report will then be sent
        - **What Can Go Wrong:** Users could report the wrong person or the wrong comment/review or send it too early without providing reason/details. Reports will be stored and can be viewed/updated/or deleted
        - **Other Activities:** Users can view the current stage of the report (needs to be reviewed, being reviewed, report complete, etc)

- **System State on Completion:** User report will be stored and sent and user will be returned to subsequent page before report

b. Providers: Avery Killian
   i. Location Service
   - **Initial Assumption:** User can log in to the providers account and interact with the page
   - **Normal:** User will be able to see active requests near them along with set their account as active or inactive
   - **What Can Go Wrong:** Users can accept a request they can't do or something can come up that makes them unable to handle the accepted request. Cancellations can be made within reason and excess cancellations will be reviewed
   - **Other Activities:** Users can see other active providers nearby to ensure no two people accept the same job
   - **System State on Completion:** Provider will be given location of customer and sent to a separate page with more details

   ii. Ratings and Review
   - **Initial Assumption:** User is logged in, can interact with page, and request services
   - **Normal:** User can view their own ratings/reviews and others and create ratings/reviews
   - The users could leave a rating/review for the wrong person without meaning to, a user can delete a comment or rating of theirs if needed
   - **Other Activities:** A user can look at extended reviews such as viewing a response made to another comment and the response to that etc..
   - **System State on Completion:** A new review will be left and added to the database for that account

   iii. Report System
   - **Initial Assumption:** A user is logged in and can interact with the page and view ratings/reviews and request services
   - **Normal:** The user can interact with a report button for drivers/customers and be given a small popup allowing further details of the report and the report will then be sent
   - **What Can Go Wrong:** Users could report the wrong person or the wrong comment/review or send it too early without providing reason/details. Reports will be stored and can be viewed/updated/or deleted
   - **Other Activities:** Users can view the current stage of the report (needs to be reviewed, being reviewed, report complete, etc)

- **System State on Completion:** User report will be stored and sent and user will be returned to subsequent page before report

c. SysAdmin: Gia Nguyen
   i. Location Service
   - **Initial Assumption:** User will be able to see active drivers and their locations
   - **Normal:** The user will be able to select a driver and view their profile from this menu
   - **What Can Go Wrong:** A different driver is selected from the intended one, so a button will be available to go back to the location page
   - **Other Activities:** Navigate to other pages to review driver and customer accounts
   - **System State on Completion:** A driver or page will be selected to view

   ii. Ratings and Review
   - **Initial Assumption:** Users will be able to view ratings on both customers and drivers and remove them as necessary
   - **Normal:** A user will remove reviews that are problematic for any reason
   - **What Can Go Wrong:** A review may be accidentally removed, a confirmation box will be added to make sure the decision to remove it was intentional
   - **Other Activities:** Navigate to another customer or provider to read the reviews associated with them
   - **System State on Completion:** Selected review will be removed and other reviews will again be available to view

   iii. Report System
   - **Initial Assumption:** Admin can view active reports and get notifications for new reports coming in
   - **Normal:** Admin will view reports and details provided, and view history/account of the reported user
   - **What Can Go Wrong:** Admin could dismiss reports by accident or punish the wrong accounts. History will be kept and database of reports can be adjusted within certain amount of time
   - **Other Activities:** Navigate to accounts associated with reports or move to other pages to review services
   - **System State on Completion:** Report will be completed and added to a separate temporary database as completed and listing the punishments applied.

Table of Contents

---

1. Introduction

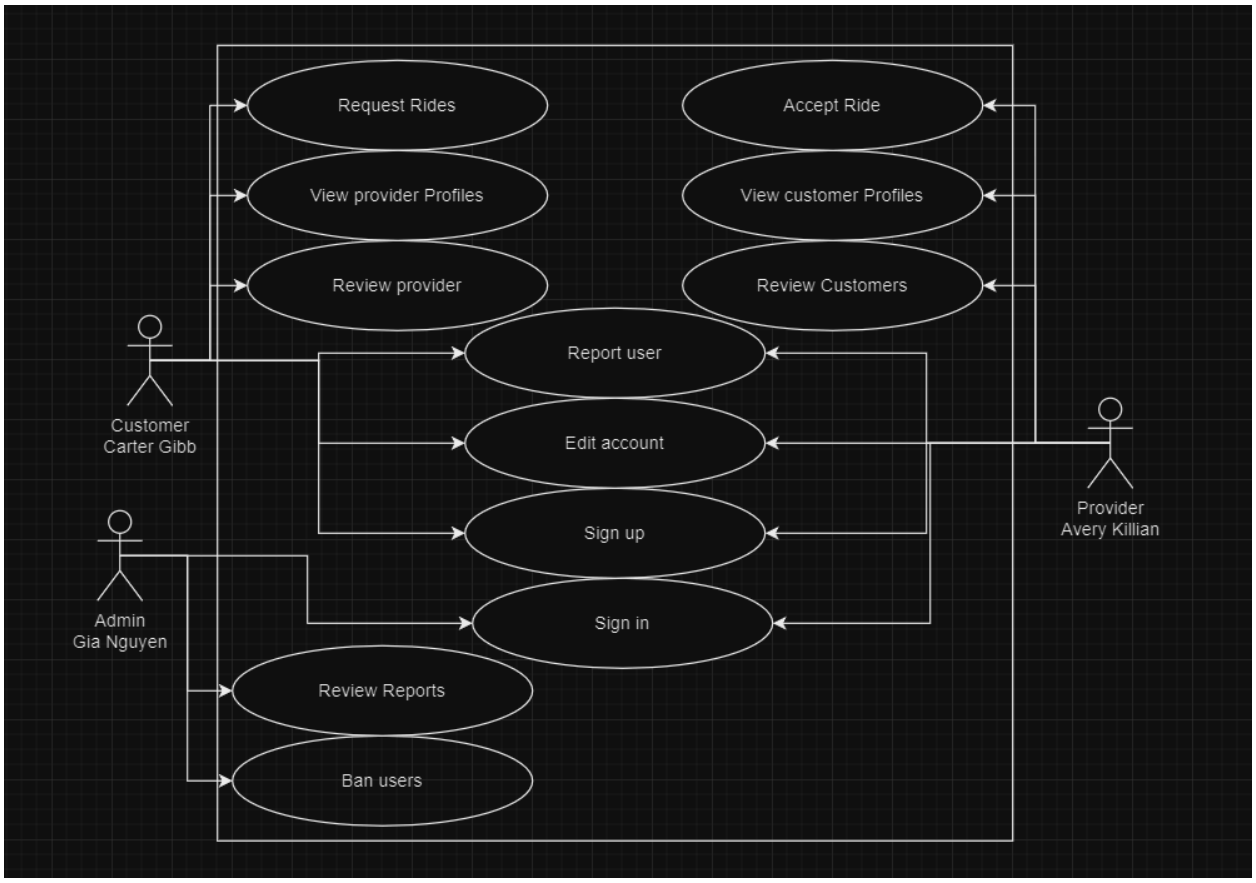
Project Overview

The Taxi Service Application is designed to connect users, providers, and admins. Users can request rides, providers can offer rides, and admins can manage users and services.

Objectives

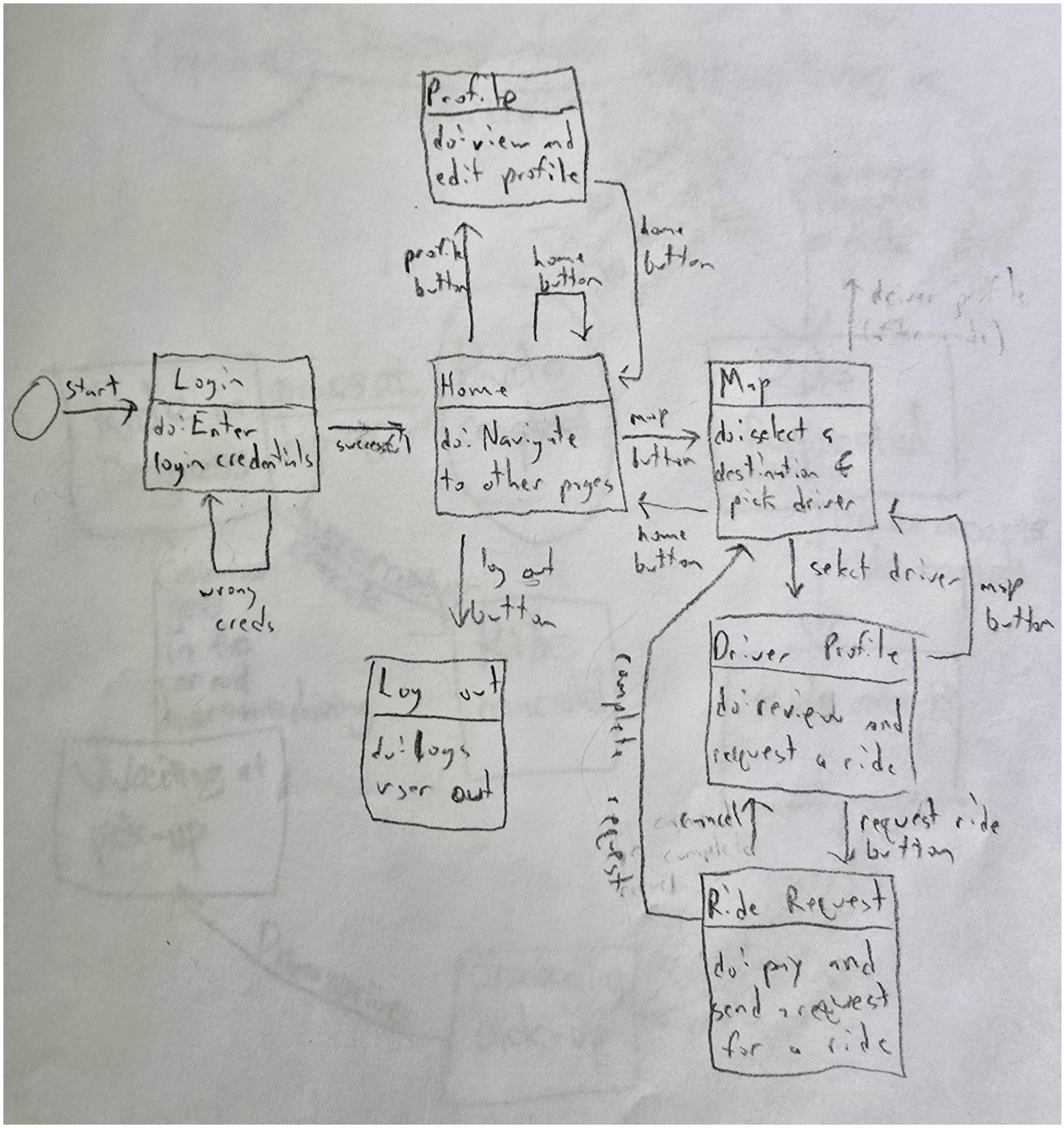
- Develop a web-based system with role-based access for users, admins, and providers.
- Implement CRUD functionalities for each actor:
    - **User**: Ride requests
    - **Admin**: User management, ride monitoring
    - **Provider**: Ride acceptance

2. Use Case Model

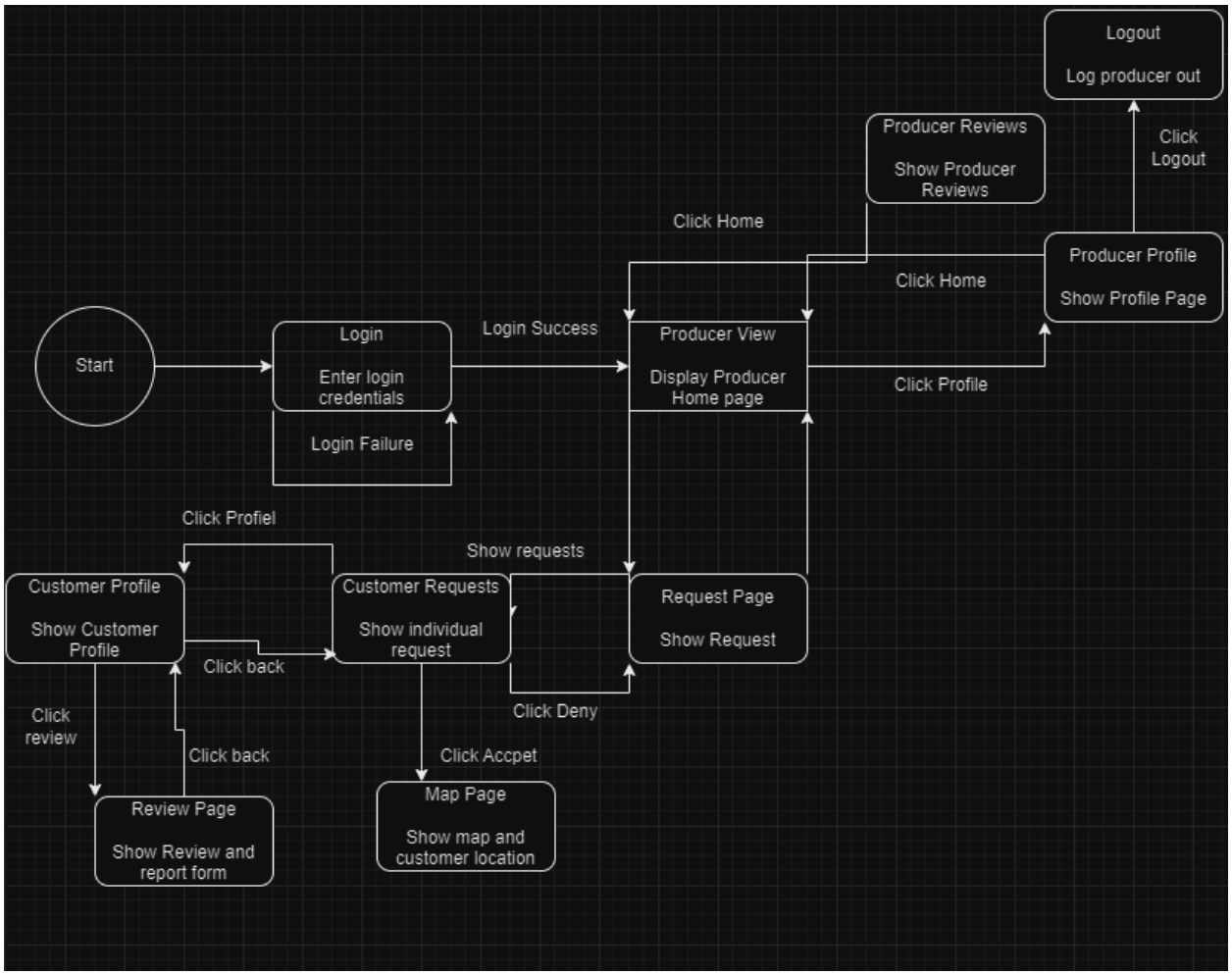


3. Design Document

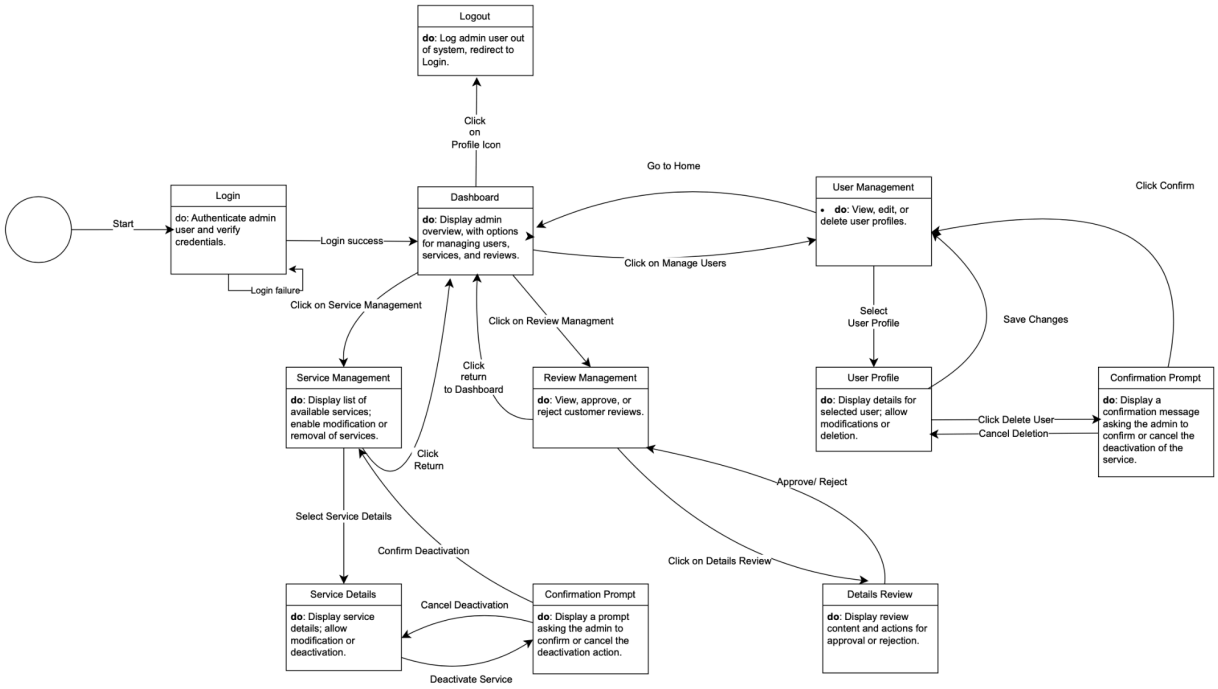
Customer: Carter Gibb



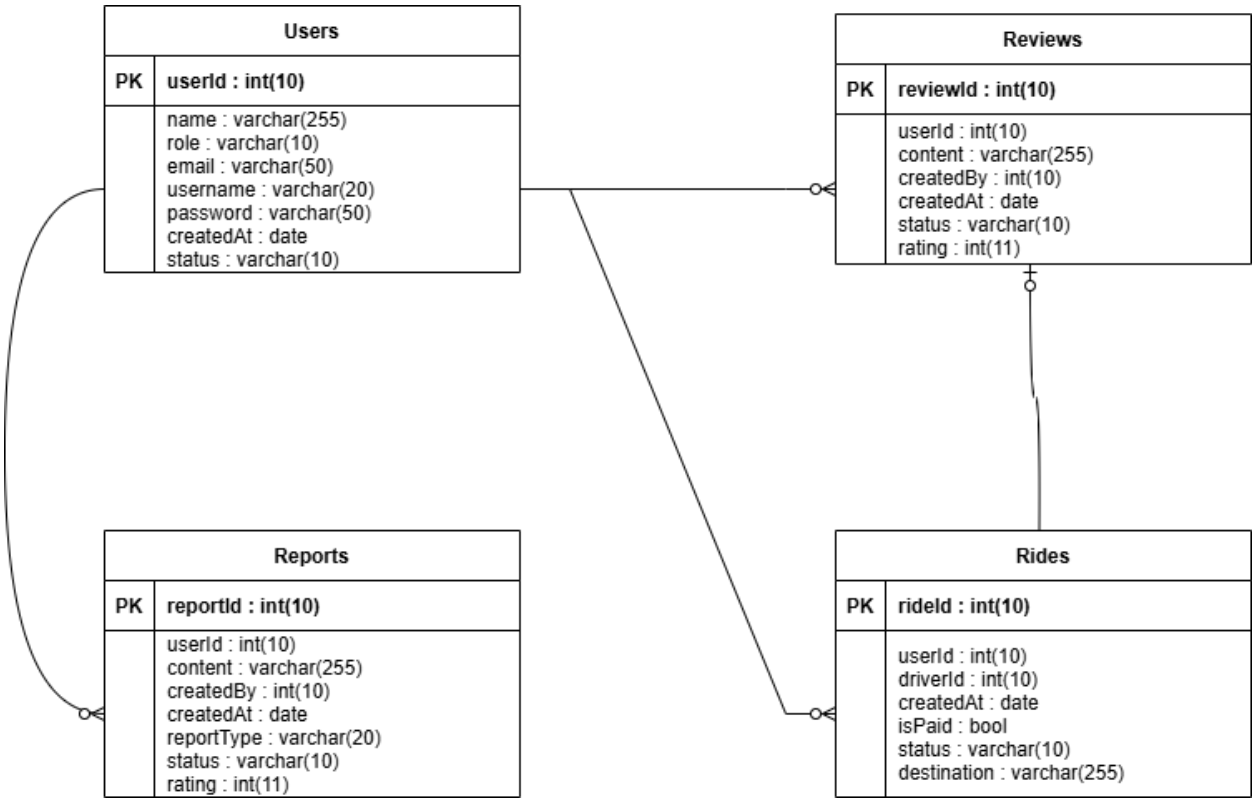
Provider: Avery Killian



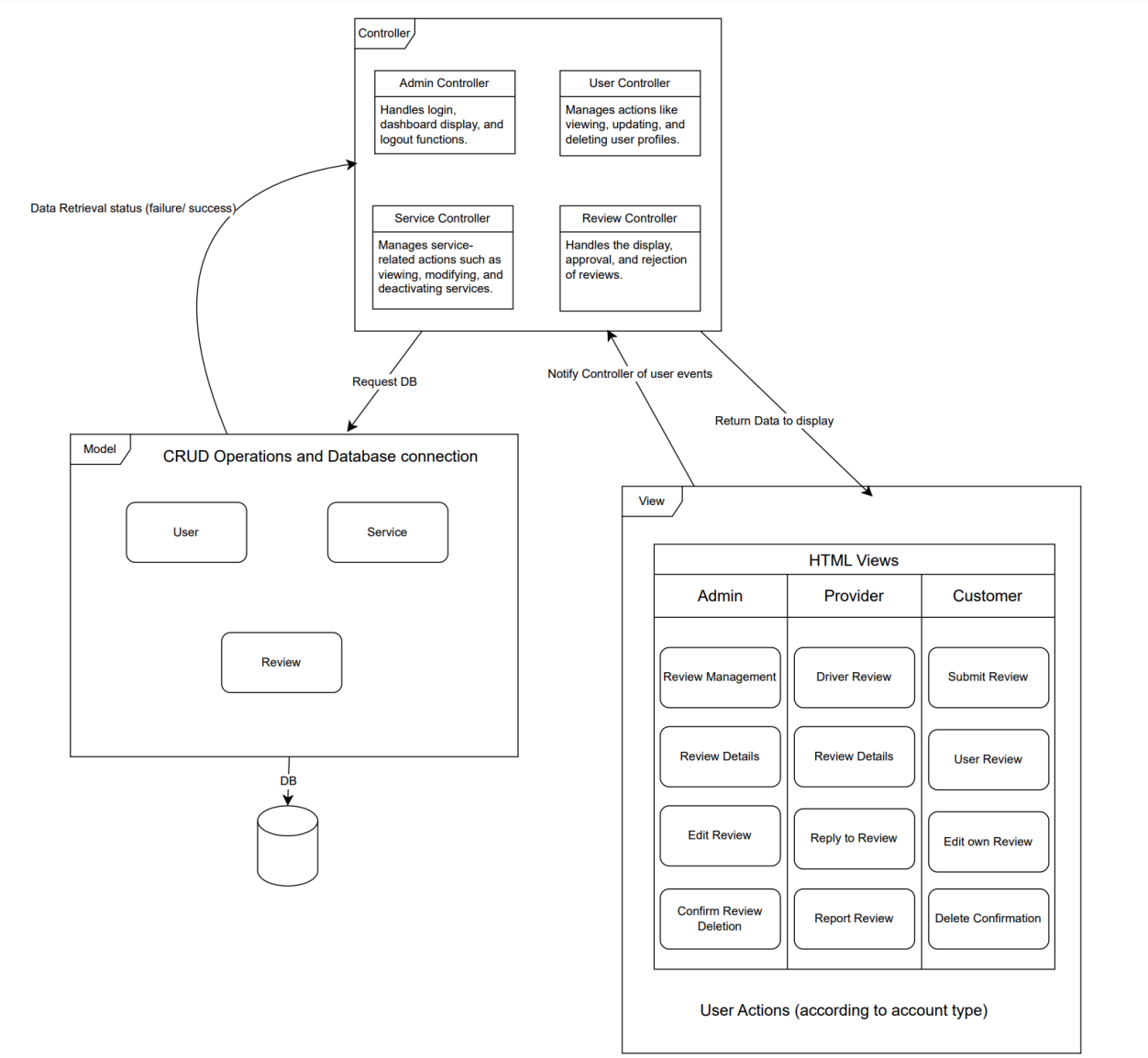
SysAdmin: Gia Nguyen



DataBase Schema



Software Architecture



---

5. Group Members and Responsibilities

- **Avery Killian (Provider Use Cases)**: Implemented the provider registration process and ride request functionalities.
- **Gia Nguyen (Admin Use Cases)**: Developed the admin dashboard and user management features.
- **Carter Gibb (Customer Use Cases)**: Created the user interface for accepting and managing ride requests.

---

6. Conclusion

Final Thoughts

The overall project did not go very well with many issues among our group and the final project. While each member did varying amounts of work when trying to combine everything the whole thing ended up crashing down around us leaving us a mess to show for our work. Learned much about the importance of communication and working with team members and beginning to realize just how difficult the real world jobsite will be.

7. Scenario Demonstration

Scenario:

Provider: Create Profile

Provider logs in for the first time and creates a profile. Inputs provider info such as names and phone numbers and car types.

Provider: Modify profile

Provider will click the profile tab, click adjust and then change info such as car type, phone number, etc.

Provider: View/Accept Rides and view customer profiles

Provider will click the requests page and view current ride requests, the provider will click the first request, view the customer profile, submit a review and report, then accept the ride.

Customer: Create Profile

Customer logs in for the first time and creates a profile. Inputs provider info such as names and phone numbers

Customer: Modify profile

Customers will click the profile tab, click adjust and then change info such as name, phone number, etc.

Customer: Request rides and leave reviews

Customers will click the request page and then select the location they desire to go along with relevant information. After customer will view the providers account, leave a review and a report

Admin: View reports

Admin will click the report page and see active reports, click the first report and review the reasoning before banning or forgiving the report. Admin will do this twice to show how each feature ban and forgive work.

Admin: Log In

Admin will display a log in scenario and the dashboard showing admin options.

Admin: delete comments and view activity

Admin will use the dashboard to show current rides and activity, will view a mean or threatening comment then choose to delete it to show database interaction.