

SpartanRideSRS

SpartanRide

<https://github.com/pyramus02/SpartanRide>

12/10/2024

Final

Aidan Quinn_(Driver), **Elijah Carpenter**_(Admin),

Kannan Thilak_(Rider)

Table of Contents:

1. Project General Description.....	4
2. Product Features.....	4
3. Functional Requirements.....	6
a. Actor 1: Rider.....	6
b. Actor 2: Driver.....	6
c. Actor 3: Administrator.....	7
4. Nonfunctional Requirements.....	8
a. Actor 1: Rider.....	8
b. Actor 2: Driver.....	8
c. Actor 3: Administrator.....	8
5. Scenarios.....	9
Actor 1: Rider.....	9
SCENARIO: Create/Modify Customer Profile.....	9
SCENARIO: View available services.....	10
SCENARIO: Subscribe to available services.....	11
SCENARIO: Write review.....	12
Actor 2: Driver.....	13
SCENARIO: Create/modify/withdraw provider profile.....	13
SCENARIO: Create services.....	14
SCENARIO: View Customer Statistics.....	15
SCENARIO: Reply to Review.....	16
Actor 3: Administrator.....	17
SCENARIO: Manage User Access.....	17
SCENARIO: Moderate services.....	18
SCENARIO: Moderate reviews.....	19
SCENARIO: View usage statistics.....	20
Scenarios with Screenshots.....	22
Driver Scenario(Aidan):.....	22
Rider Scenario(Kannan):.....	23
Admin Scenario(Elijah):.....	24
Design Document:.....	25
1. Project Overview.....	25
2. Use-Case Model.....	25
3. State Machine Diagrams.....	26
a. User(Rider) – Kannan Thilak.....	26
b. Provider(Driver) – Aidan Quinn.....	27

c. Admin – Elijah Carpenter.....	28
----------------------------------	----

1. Project General Description

The goal of SpartanRide is to provide a traveling service to college students. It is a carpooling service wherein a student can set a destination, either as the driver or rider, and find other's going towards the same destination. This will not only allow students to save money on gas, but to assist the environment, help students meet new people, and to build a larger community. The users that will be interacting with the application would be under these three categories of actors: Rider, Driver, and Administrator.

2. Product Features

The SpartanRide application is designed to allow users to interact and share a travel experience together. Below are the key features that the application contains:

Driver Search: Riders within the application will be able to see and locate drivers who are headed towards the location Riders have set. This will allow them to select the Driver, given there is space available, and request a pickup to ride to said location.

Identification Verification: Within the app, both types of users will be able to verify their status as a student. Not only that, but depending on the way a particular student signed up, they will either verify their student ID with a driver's license, and/or a license plate as well. All with the goal of ensuring Rider and Driver safety.

Profile Modification: After signing up for the service, the user will be able to edit their profile with things that are specific to the user type. Where Drivers would be able to add their car type and seating allotment, Riders can set a pickup spot. Both accounts could also have a biographical section, profile picture, and Student ID/Year.

Review System: Riders will be able to review Drivers, and vice versa. They will also be able to reply to these reviews.

3. Functional Requirements

a. Actor 1: Rider

- fr. i: Riders will be able to sign up for the service.
- fr. ii: Riders will be able to verify as a student.
- fr. iii: Riders will be able to set a destination, and pick a driver going that route based on said destination.
- fr. iv: Riders will be able to cancel the ride prior should they feel the need.
- fr. v: Riders will be able to respond to reviews and review Drivers.
- fr. vi: Riders will be set a profile picture and modify their profile.
- fr. vii: Riders will be able to switch to Drivers if desired.

b. Actor 2: Driver

- fr. i: Drivers will be able to sign up for the service.
- fr. ii: Drivers will be able to verify as a student.
- fr. iii: Drivers will be able to register their car as the vehicle they will be driving.
- fr. iv: Drivers will be able to set a destination.
- fr. v: Drivers will be able to agree to riders joining them depending on the amount of space in the car.
- fr. vi: Drivers will be able to start a trip when they want or if their seats are full.
- fr. vii: Drivers will be able to cancel the trip should they feel the need.
- fr. viii: Drivers will be able to respond to reviews and review Riders.
- fr. ix: Drivers will be able to set a profile picture and modify their profile.
- fr. x: Drivers will be able to switch to Riders if they want to drive.

c. Actor 3: Administrator

- fr. i: Administrators will be able to login with administrator privileges.
- fr. ii: Administrators will be able to approve of rider verifications.
- fr. iii: Administrators will be able to approve of driver verifications.
- fr. iv: Administrators will be able to ban riders for unsafe or harmful actions.
- fr. v: Administrators will be able to ban drivers for unsafe or harmful actions.
- fr. vi: Administrators will be able to modify and delete reviews for moderation purposes.
- fr. vii: Administrators will be able to see system statistics, which shall encompass both user usage statistics (for example, peak operating hours) and technical statistics (for example, the amount of time the system was down during a server outage).
- fr. viii: To support administrative tasks, Administrators will be able to interface directly with the software product's database server.

4. Nonfunctional Requirements

a. Actor 1: Rider

- nfr. i: The location data of riders must be used to calculate their relative position to drivers with a high confidence interval.
- nfr. ii: Profiles should display information with high speed and high fidelity.
- nfr. iii: All data fields concerning riders must be easily stored in the system's database.

b. Actor 2: Driver

- nfr. i: Driver information should be stored and compared with a database that we set.
- nfr. ii: GPS Location should be very accurate so Drivers don't make mistakes.
- nfr. iii: Profiles should load fast as they are important in seeing who the Driver picks up.

c. Actor 3: Administrator

- nfr. i: The software service should maintain industry standard 99.9% uptime yearly.
- nfr. ii: The data of users must be secure and protected from hacks and accidental data leaks.
- nfr. iii: Clearly malicious behavior from users should be met with administrative action no later than 24 hours after the fact.

5. Scenarios

Actor 1: Rider

SCENARIO: Create/Modify Customer Profile

- I. Initial Assumption: The rider has successfully accessed the system through a web browser and is currently interacting with the graphical user interface.
- II. Normal: If the rider has never made an account before, they will be prompted to set an email address, password, and username, as well as to verify their status as a UNCG student. Their account will then be activated pending moderator approval. If the rider has already made an account, they will input their identifying information to login, and then be displayed a dashboard screen, with buttons prompting them to edit or add relevant information to their profile.
- III. What can go wrong: A rider can lose access to their account, either because of an error within the system, a malicious actor, or simply forgetting their identifying information.
- IV. Other activities: This is also the area where a rider will be prompted to allow access to their location data if they have not already done so.
- V. System state upon completion: The rider has successfully created an account or has successfully logged into their account. The changes they have made to their profile are reflected in the system database and to other users.

SCENARIO: View available services

- I. Initial Assumption: The rider has successfully logged into the system, and is currently being displayed their account dashboard.
- II. Normal: The rider will click on a labeled button which will display a new page showing all nearby rides, as well as their respective drivers and destinations within the vicinity of the rider.
- III. What can go wrong: The rider's location data may be corrupted or missing, preventing the system from accurately displaying nearby rides.
- IV. Other activities: This is the same area where a rider is able to accept rides.
- V. System state upon completion: The rides nearest to the rider and most relevant to their destination have been found, and the page has been displayed to the user successfully.

SCENARIO: Subscribe to available services

- I. Initial Assumption: The rider has successfully logged into the system, and is currently being displayed their account dashboard.
- II. Normal: The rider will click on a labeled button which will display a new page showing all nearby rides, as well as their respective drivers and destinations within the vicinity of the rider. The rider will then select the ride most relevant to their desired destination using a button displayed on the web page.
- III. What can go wrong: The system may not register the selected ride as actually having been accepted, or may mark a ride which wasn't selected as accepted.
- IV. Other activities: The rider may also browse rides here without accepting any.
- V. System state upon completion: The user has successfully accepted a ride and this is reflected both in the system database and to the driver.

SCENARIO: Write review

- I. Initial Assumption: The rider has successfully logged into the system, and is currently being displayed their account dashboard. The rider has also accepted and completed a ride.
- II. Normal: The rider will receive a labeled prompt asking them to write a review for their most recent ride, which they can either accept or decline. If they accept, they will be sent to a new page to write a text review within a given character limit.
- III. What can go wrong: Riders may write irrelevant or inappropriate reviews which require moderation.
- IV. Other activities: The rider is also free to decline to write a review at all.
- V. System state upon completion: The review is represented as a data field of the driver it was made about in the system database, and can be seen by all other users.

Actor 2: Driver

SCENARIO: Create/modify/withdraw provider profile

- I. Initial Assumption: The driver has successfully logged in and is currently being displayed on their driver dashboard.
- II. Normal: If the driver has never made an account before, they will be prompted to set an email address, password, and username, as well as to verify their status as a UNCG student. Their account will then be activated pending moderator approval. If the driver has already made an account, they will input their identifying information to login, as well as information relevant to their offered rides, and then be displayed a dashboard screen, with buttons prompting them to edit or add relevant information to their profile. If a driver wishes to withdraw their driver account, they can request this action from an administrator by pressing a labeled button on their dashboard.
- III. What can go wrong: A driver can lose access to their account, either because of an error within the system, a malicious actor, or simply forgetting their identifying information.

- IV. Other activities: This is also the driver where a driver will be prompted to allow access to their location data if they have not already done so.
- V. System state upon completion: The rider has successfully created an account or has successfully logged into their account. The changes they have made to their profile are reflected in the system database and to other users.

SCENARIO: Create services

- I. Initial Assumption: The driver has successfully logged in and is currently being displayed on their driver dashboard.
- II. Normal: The driver will press a labeled button prompting relevant information regarding the ride they are offering, including vehicle information and destination, among other data fields. The ride will then be displayed to riders for them to accept.
- III. What can go wrong: This is open to abuse by malicious actors, and should be moderated by administrators. A database error occurring here could prevent drivers from reaching riders with their services.
- IV. Other activities: This is also the driver where a driver will be prompted to allow access to their location data if they have not already done so.
- V. System state upon completion: The system database has been updated to reflect the existence of the new ride and this change is visible to riders.

SCENARIO: View Customer Statistics

- I. Initial Assumption: The driver has successfully logged in and is currently being displayed on their driver dashboard. The driver also has completed enough rides to have generated relevant and useful information regarding their customers for personalized statistics.
- II. Normal: The driver will press a labeled button which will display a new page showing a variety of statistics for the driver. There will be various input buttons available for the driver to specify what kind of statistics they want, which will then be displayed in a window for them, They might for example want to know the most popular destination over a given time frame, and contrast that with a previous time frame.
- III. What can go wrong: Either statistics are not displayed at all due to a breakdown in the system relevant monitoring, or else false or incomplete statistics are displayed successfully for the same reason.
- IV. Other activities: The driver may also query for general statistics about all the rides being offered and accepted on the system.

- V. System state upon completion: The queried statistics have successfully been displayed to the driver, who can use them when planning the rides they offer.

SCENARIO: Reply to Review

- I. Initial Assumption: Driver has had a review left on their profile and is currently logged into the service.
- II. Normal: Users will see the relevant replies underneath their profile's review page, and see a button under each review reading "reply".
 - A. If the button is pressed, a text box will appear allowing the Driver to reply to the Rider's review.
- III. What Can Go Wrong:
 - A. Users can accidentally make a typo after submitting, so there will be an edit function.
 - B. Users may abuse this function so reviews will be reviewed prior to posting.
- IV. Other Activities: Users may delete the comment that they write, or like or dislike comments.
- V. System State on Completion: Reply will be sent to the original reviewer and notify them. They will be able to view it and reply to that message as well.

Actor 3: Administrator

SCENARIO: Manage User Access

- I. Initial Assumption: The system administrator is logged in as an administrator and needs to take a user access action (either approving an account verification or banning an account.)
- II. Normal: The administrator will click on a dashboard button which will display the user database management console. The administrator will query the account by name, and press a button which enforces the administrative action.
- III. What can go wrong: The action is performed on the wrong account, or cannot be performed on the correct account. In both cases, this is likely an issue with either the database or command being sent to the database.
- IV. Other activities: The administrator can directly message the user in question to inform them of the pending user access action.
- V. System state on completion: The database is updated successfully and the access abilities of the given user have changed.

SCENARIO: Moderate services

- I. Initial Assumption: The system administrator is logged in as an administrator and needs to take a moderation action regarding posted rides which are inappropriate for whatever reason.
- II. Normal: The administrator will click on a dashboard button which will display the ride management console. The administrator will query the ride by the username of the driver, and take the necessary action, either deleting it or modifying it.
- III. What can go wrong: This power is open to abuse, so administrators should be well-known and trusted actors. Additionally, steps should be taken to prevent malicious hackers from gaining administrator access to the system.
- IV. Other activities: The system should automatically message the driver who posted the ride as well as any riders who signed up for it to inform them of the change.
- V. System state on completion: The ride is successfully modified or deleted and this is reflected both in the database and to users on the system's graphical user interface.

SCENARIO: Moderate reviews

- I. Initial Assumption: The system administrator is logged in as an administrator and needs to take a moderation action regarding posted reviews which are inappropriate for one reason or another.
- II. Normal: The administrator will click on a dashboard button which will display the review management console. The administrator will be able to query the review either by the user who made the review or the user who was the subject of the review. The administrator will press a button to enable modification of the review, either changing it or deleting it outright.
- III. What can go wrong: This power is open to abuse, so administrators should be well-known and trusted actors. Additionally, steps should be taken to prevent malicious hackers from gaining administrative access to the system.

- IV. Other activities: The system should automatically message the user who posted the review as well as the user who was the subject of it to inform them of the change.
- V. System state on completion: The review is successfully modified or deleted and this is reflected both in the database and to users on the system's graphical user interface.

SCENARIO: View usage statistics

- I. Initial Assumption: The system administrator is logged in as an administrator and wants to see statistics regarding the system.
- II. Normal: The administrator will click on a dashboard button which will display the system statistics page, showing a variety of relevant metrics of system performance, concerning both technical topics (such as system downtime) and topics related to administrative tasks (such as number of moderation requests completed).
- III. What can go wrong: Either statistics are not displayed at all due to a breakdown in the system relevant monitoring, or else false or incomplete statistics are displayed successfully for the same reason.
- IV. Other activities: The administrator can query different time frames for the given metrics to compare current performance to past performance.

- V. System state on completion: The desired statistics have been successfully displayed to the administrator for use on relevant system tasks.

Scenarios with Screenshots

Driver Scenario(Aidan):

- John Doe registers as a Driver using an email, name, password, and car model. He then sets a destination and start area on his driver page. Then Andrew Sam, who is a registered rider will also select John Doe as his driver going to a similar location, where he can also leave a review on John's Page.

The screenshot displays the SpartanRide website interface. The header is dark purple with the 'SpartanRide' logo in yellow on the left and 'Sign Up' and 'Log In' buttons in yellow on the right. The main content area has a light blue background. On the left, there is a promotional text block. On the right, there is a 'Driver Sign Up' form with a blue border and a drop shadow. The form contains five input fields: Name, Email, Password, Car Model, and Plate Number, each with a placeholder label. A blue 'Submit' button is located at the bottom of the form. Below the form, there is a dark purple footer with the 'SpartanRide' logo in yellow.

SpartanRide Sign Up Log In

Travel, Socialize, and be Productive with Spartan Ride!

Spartan Ride is a carpooling service for students attending UNCG! Allowing students to not only save money, but to build relationships as well.

Starting is easy! All you have to do is verify your status as a student at UNCG, make an account, and then you can start driving!

Driver Sign Up

Name

Email

We'll never share your email with anyone else.

Password

Car Model

Plate Number

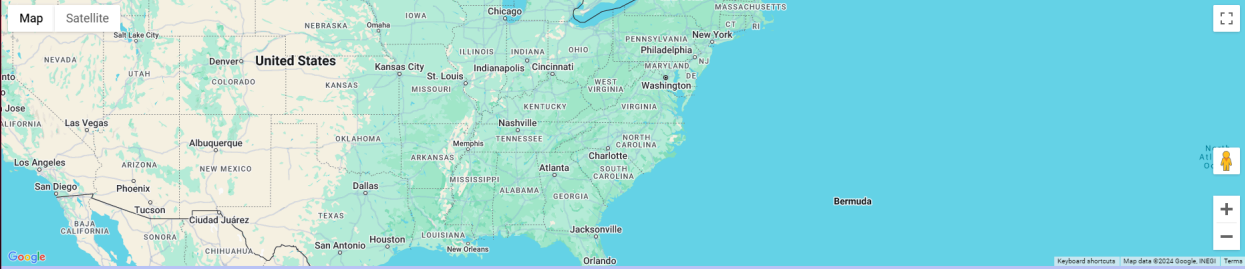
SpartanRide

Rider Scenario(Kannan):

- Andrew Sam registers as a Rider using email, name, password. He selects the drivers available which matches his preferred start area to his preferred destination. Then John Doe, who is a registered Driver will also have an option to either Accept or kick out the rider as his driver going to a similar location. Finally Based on the Ride the rider can write a review which will be posted in the driver page.

SpartanRide Logout

Explore the UNGC Area!



Map Satellite

United States

Keyboard shortcuts | Map data ©2024 Google, INEGI | Terms

Welcome to SpartanRide

Available Rides

ID	Name	Destination	Embarkment	Subscribe	Unsubscribe	Report
1	John Druve	North Green Street	West Salven Avenue	Sub	Unsub	Report
2	BOB			Sub	Unsub	Report
52	John Doe	1600 Spring Garden St.	UNCG Petty Building	Sub	Unsub	Report

Write a Review

From Ride ID:

Admin Scenario(Elijah):

- The admin views reported Drivers, and Riders, taking appropriate action and banning users where needed. These changes are reflected in the other use cases, with banned users unable to access the system. The admin views reported reviews, and deletes them where they are inappropriate. These reviews are no longer visible to Drivers. The admin queries for system statistics and these statistics are accurately displayed.

SpartanRide

Admin

User Access Actions:

ID	Name	Destination	Embarkment	Dismiss
1	John Druve	North Green Street	West Salven Avenue	<button>Dismiss</button>

ID	Name	Email	Dismiss
Ban Driver			
<button>Ban</button>			
Ban Rider			
<button>Ban</button>			

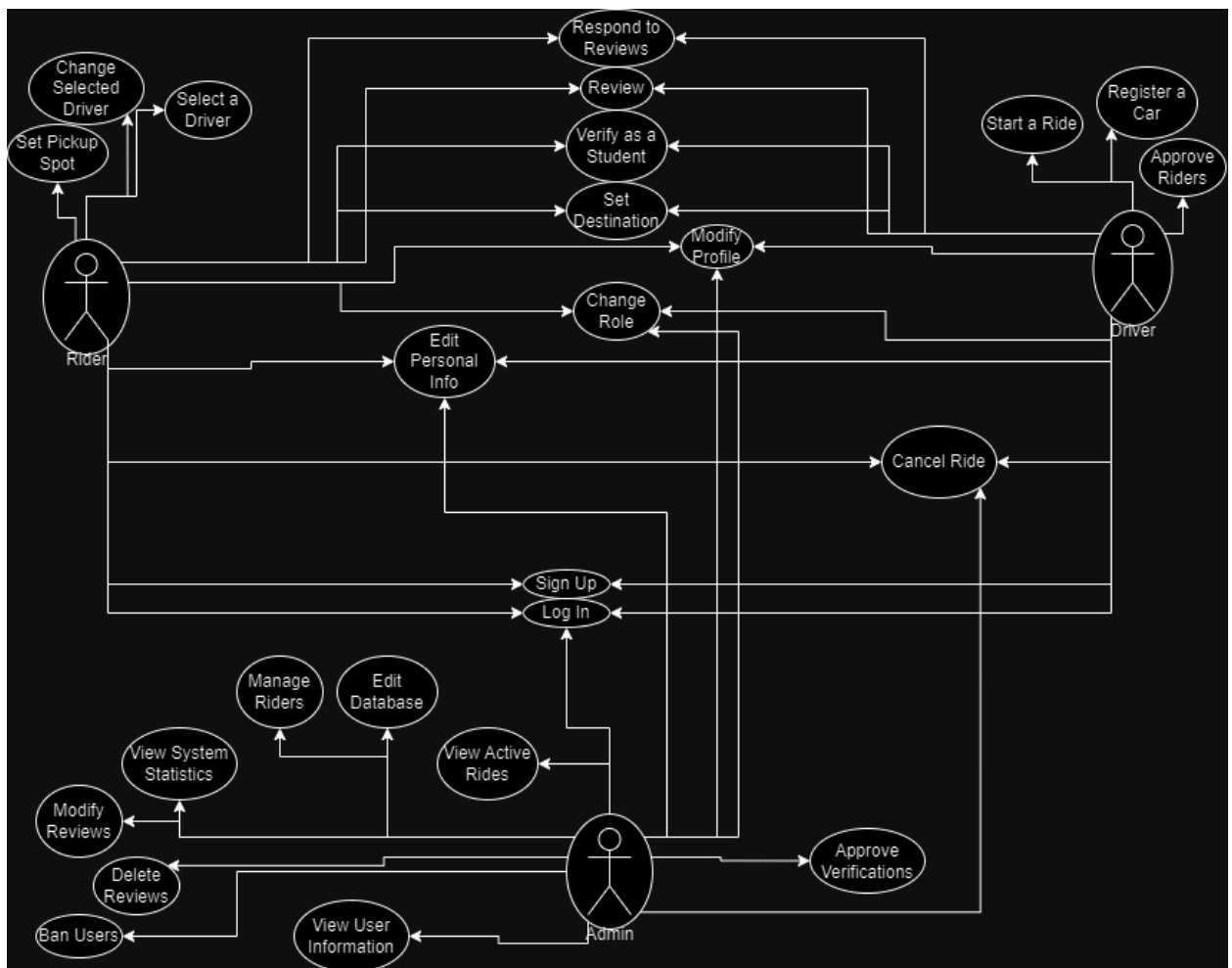
SpartanRide

Design Document:

1. Project Overview

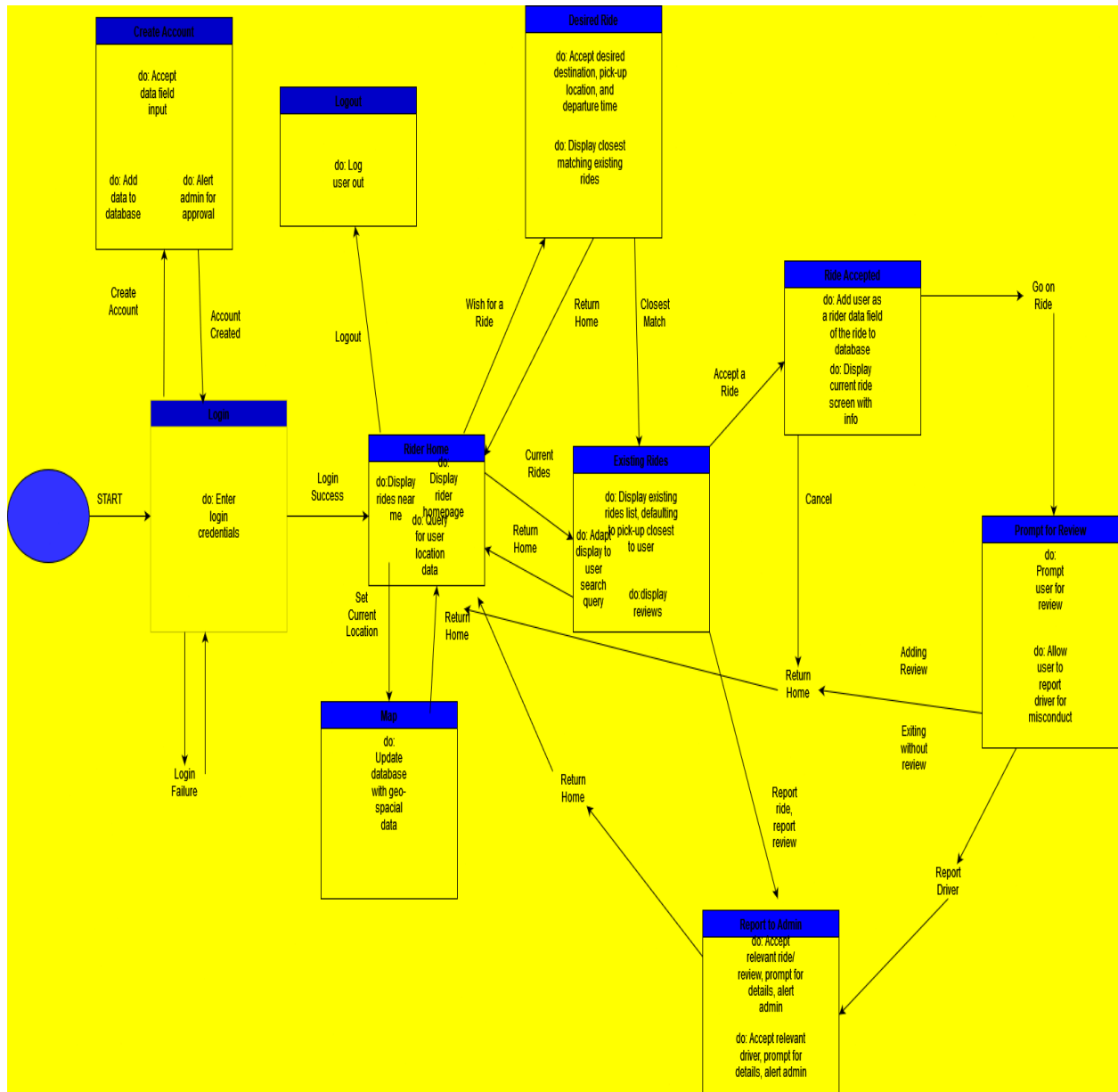
The goal of SpartanRide is to provide a traveling service to college students. It is a carpooling service wherein a student can set a destination, either as the driver or rider, and find others going towards the same destination. This will not only allow students to save money on gas, but to assist the environment, help students meet new people, and to build a larger community. The users that will be interacting with the application would be under these three categories of actors: Rider, Driver, and Administrator.

2. Use-Case Model

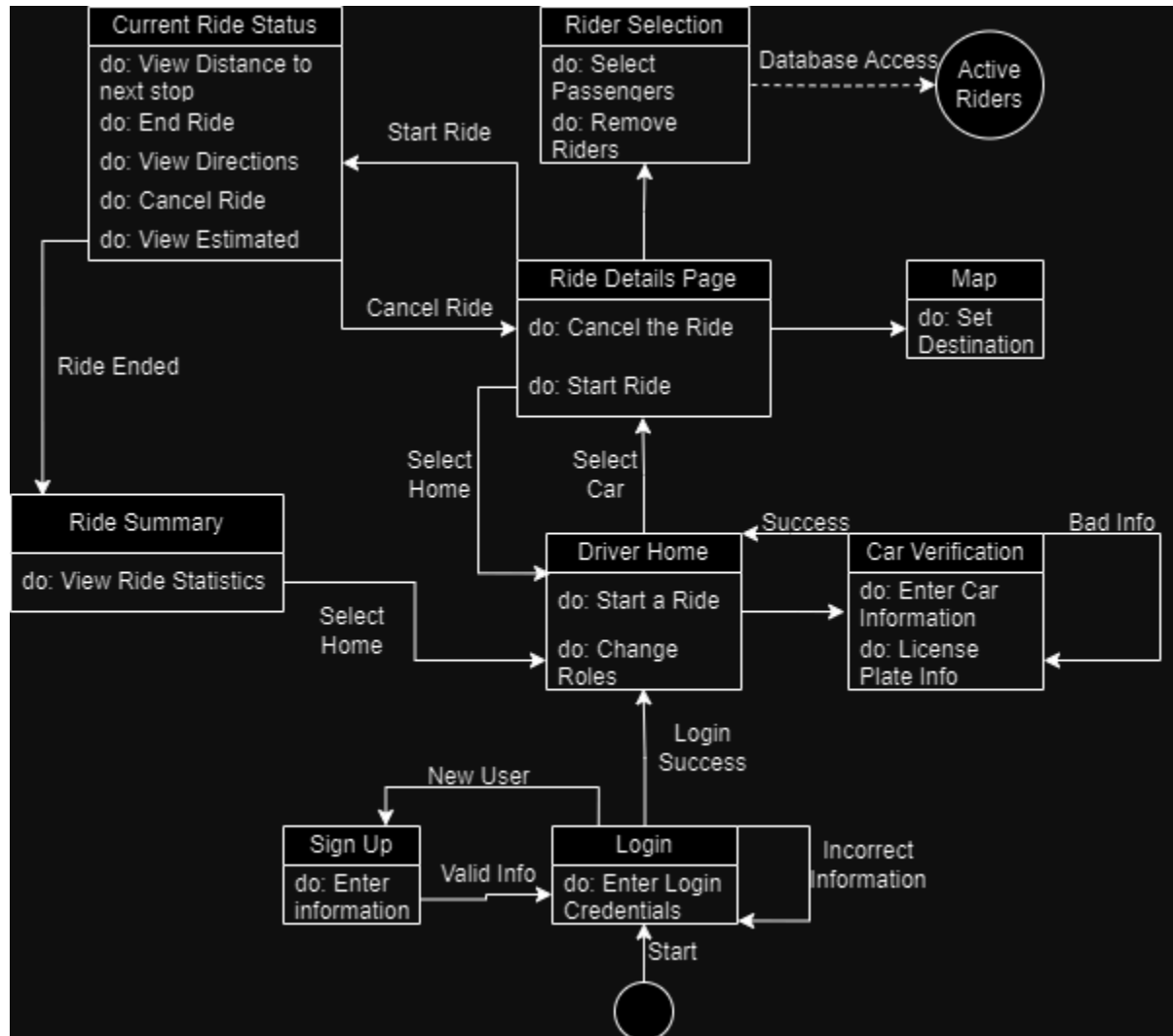


3. State Machine Diagrams

a. User(Rider) – Kannan Thilak



b. Provider(Driver) – Aidan Quinn



c. Admin – Elijah Carpenter

