**INTERSWITCH ACADEMY PROJECT PROPOSAL**

**Title**

Online Carpooling Application

**Trainee Name**

Adegboyega Olorunfemi

**ISW ID**

AOBCSO07619

Table of Contents

[1. BACKGROUND STUDY AND GENERAL OVERVIEW 3](#_Toc120276099)

[2. THE BENEFITS OF THIS PROJECT 3](#_Toc120276100)

[3. REQUIREMENTS/OBJECTIVES 3](#_Toc120276101)

[4. SCOPE 4](#_Toc120276102)

[5. CHALLENGES 4](#_Toc120276103)

[6. SDLC (AGILE METHODOLOGY) 4](#_Toc120276104)

[6.1 Definition 4](#_Toc120276105)

[6.2 Benefits of Agile Methodologies 5](#_Toc120276106)

[7. SCHEDULE 5](#_Toc120276107)

[8. LIST OF TECHNOLOGIES USED 6](#_Toc120276108)

[8.1 Explanation of the technologies above 6](#_Toc120276109)

[9. DESIGN DIAGRAMS (Schema and E-R Respectively) 7](#_Toc120276110)

[10. CODING 8](#_Toc120276111)

[11. TESTING 11](#_Toc120276112)

[11.1. Test Script For DeePee (Manual Testing) 11](#_Toc120276113)

[12. IMPLEMENTATION 12](#_Toc120276114)

[12.1 Login 12](#_Toc120276115)

[12.2 Sign up as a customer/host 12](#_Toc120276116)

[12.3 Upload a trip page 12](#_Toc120276117)

[12.4 Find a ride page 12](#_Toc120276118)

[12.5 Landing page 12](#_Toc120276119)

[12.6 The home page 12](#_Toc120276120)

[12.7 The about page 12](#_Toc120276121)

[12.8 The contact page 12](#_Toc120276122)

[12.9 Active search tab 13](#_Toc120276123)

[12.10 Customer’s trip request tab 13](#_Toc120276124)

[12.11 Host trip upload tab 13](#_Toc120276125)

[12.12 Host’s trip requests tab 13](#_Toc120276126)

[13. FUTURE ENHANCEMENT 13](#_Toc120276127)

[14. CONCLUSION 14](#_Toc120276128)

[15. REFERENCES 15](#_Toc120276129)

# BACKGROUND STUDY AND GENERAL OVERVIEW

Considering the serious challenges in transportation e.g., inadequate road network, high traffic, air pollution, security risks, the rising cost of vehicle purchase and maintenance, economic recession, and inflation, etc. Humanity is faced with grave problems when it comes to commuting in major cities, especially at peak hours.

# THE BENEFITS OF THIS PROJECT

A major solution to the problems listed above is in utilizing the current resources available rather than trying to get more vehicles into the already congested road networks. Hence, this application would make the following possible:

- Vehicle owners in Interswitch would be able to make more money to conveniently settle their vehicle expenses.

- There would be less pressure on the road because many vehicle owners would not have to drive and yet would still get an affordable lift in a comfortable and decent vehicle.

- Air pollution would be greatly reduced since there would be fewer vehicles on the road.

- Traffic on the road would also be greatly reduced as there would be fewer vehicles on the road.

- Road users would also spend less to commute since the transport fares would be cheaper than Uber and sharing of rides (vehicle pooling) is in the picture.

- There would no longer be the possibility of getting into a one-chance vehicle or getting into vehicles driven by notorious and careless drivers.

- This service even beats the shuttle service as less time is spent on the road since the host has fewer stops and branching to do. Also, the delay that comes from waiting for many different people to fill up the shuttle is greatly reduced.

- Interswitch staff would also get to network and bond across buildings and departments.

# REQUIREMENTS/OBJECTIVES

The Requirements for this application include:

* Functional homepage for all users to see what the webpage is about and how to contact the business administrators.
* Simple signup form for both vehicle hosts and customers.
* Secure login for both vehicle hosts and customers
* Platform for vehicle hosts to input their trip details.
* Platform for customers to search out trips and then select their desired host.
* Dashboard to display contact details of the host to the customer for further discussions like the cost of the trip.
* Avenue for customers to upload agreed price and for host to approve the uploaded price.

# SCOPE

The scope of this project is within the Interswitch head office in Victoria Island Lagos and her employees’ resident in Lagos State.

# CHALLENGES

* Scaling the project down was a big challenge but with the help of the facilitators, this was possible.
* Knowing where to start and the step-by-step procedure to build the Web Application with ease was a big concern but the facilitators guide helped through this.
* Getting ideas for the project’s business logic and web page design was a challenge but some similar web applications came in very handy. These Web applications include (<https://www.uber.com/ng/en/>, <https://www.lyft.com/>, <https://lagosride.com/>, <https://www.blablacar.co.uk/>, and <https://www.bolt.eu/> ).
* Being my first project in Web application development, adopting every technology used was a big challenge and this made the application building much more time-consuming and hectic. This led to series of personal engagements with colleagues and online materials, and this pushed the project far.

# SDLC (AGILE METHODOLOGY)

The agile framework was used for the development of the application.

## 6.1 Definition

Agile SDLC methodology is based on collaborative decision-making between requirements and solutions teams, and a cyclical, iterative progression of producing working software. Work is done in regularly iterated cycles, known as sprints, that usually last two to four weeks.

Every task in this project was developed as a sprint to ensure that there were presentable deliverables per time. This also ensured that the project adapted quickly to different requests from the instructors per time.

For each iteration, there was a concurrent functioning of various areas of product development such as:

1. Planning
2. Requirements Analysis
3. Design
4. Development
5. Unit Testing
6. Deployment

Diagram

Description automatically generated

## 6.2 Benefits of Agile Methodologies

* Involves pair programming which reduces the number of errors in the development or coding phase and is better than a single programmer doing all the hard part.
* This model trims down the entire development time of any project.
* After each iteration, customers and stakeholders of the project can get a fair idea of the updated software that is being developed by the agile model. So, any change in the system parts can be addressed at any iteration.

# SCHEDULE

The project development and implementation spanned three months.

The project topic, scope, requirements, and design diagrams were developed in the first month,

The front-end design was a major task up to the end of the second month and then the database implementation alongside the backend development was done before the third month elapsed. Testing and pushing to Git-hub were likewise done in the same period.

**The time allocated for the total system development life cycle is 3 months.**

Tasks done each week in the three months

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Week  Month | Week 1 | Week 2 | Week 3 | Week 4 |
| September | Discussion and approval of project topic and problem statement with trainers | Discussion and approval of project topic and problem statement with trainers | Preparation and approval of design diagrams by trainers | Development of project front-end design plans and user journey using hand drafts and Microsoft word. |
| October | Implementation of project front end designs using HTML, CSS, JavaScript, and the Visual Studio IDE. | Implementation of project front end designs using HTML, CSS, JavaScript, and the Visual Studio IDE. | Implementation of project front end designs using HTML, CSS, JavaScript, and the Visual Studio IDE. | Development of backend service and database entities and attributes using Spring boot, JPA and Microsoft SQL Server on IntelliJ |
| November | Development of backend service and database entities and attributes using Spring boot, JPA and Microsoft SQL Server on IntelliJ | Rounded up development of backend service and project documentation draft submission | Project testing, demo presentation and project documents final submissions | Project final presentation |

# LIST OF TECHNOLOGIES USED

**Er Diagram Design:** Two online designers were utilised.https://app.diagrams.net/#G1csr15vq\_OTii9lF-2uSEgD-m1CABYcH- AND <https://app.quickdatabasediagrams.com/#/d/tdqVfX>

**Frontend Designs:** HTML, CSS, and JavaScript were used with the Visual Studio IDE.

**Database Development:** Microsoft SQL Server was used.

**Backend Design:** Java and Spring Boot were used with the IntelliJ IDE.

**API Testing:** Postman was used for API testing by calling the RESTAPIs.

**Project Cloning:** Git hub was used to clone the project files.

## Explanation of the technologies above

**HTML:** Hypertext Markup Language, this is a standardized system for tagging text files to achieve font, color, graphic, and hyperlink effects on World Wide Web pages using Web browsers.

**CSS:** CSS stands for Cascading Style Sheets · CSS describes how HTML elements are to be displayed on screen, paper, or in other media

**JavaScript:** This is an Object-oriented computer programming language commonly used with HTML and CSS to create interactive web pages.

**Microsoft SQL Server:** Microsoft SQL Server is a relational database management system (RDBMS) that supports a wide variety of transaction processing, business intelligence and analytics applications in corporate IT environments.

**Java:** Java is an object-oriented programming language that produces software for multiple platforms. One of the reasons Java is widely used is that when a programmer writes a Java application, the compiled code can run on most operating systems (OS), including Windows, Linux and Mac OS

**Spring Boot:** Java Spring Boot (Spring Boot) is a tool that makes developing web application and microservices with Spring Framework faster and easier through three core capabilities: Autoconfiguration. An opinionated approach to configuration and the ability to create standalone applications

**Postman:** Postman is an API platform for developers to design, build, test and iterate their APIs. Postman simplifies each step of the API lifecycle and streamlines collaboration between developers while creating the APIs.

**Git hub:** GitHub is a web-based version-control and collaboration platform for software developers. It is used to share codes and to build projects collaboratively in teams.

**IDE:** An integrated development environment (IDE) is a software application that helps programmers develop software code efficiently. It increases developer productivity by combining capabilities such as software editing, building, testing, and packaging in an easy-to-use application.

# DESIGN DIAGRAMS (Schema and E-R Respectively)

Graphical user interface, text, application

Description automatically generated

Diagram

Description automatically generated

# CODING

**The presentations were done using HTML, CSS, and JavaScript.**

**HTML CODES ON VISUAL STUDIO CODE IDE**

**A computer screen capture

Description automatically generated with medium confidence**

**CSS CODES ON VISUAL STUDIO CODE IDE**

**A computer screen capture

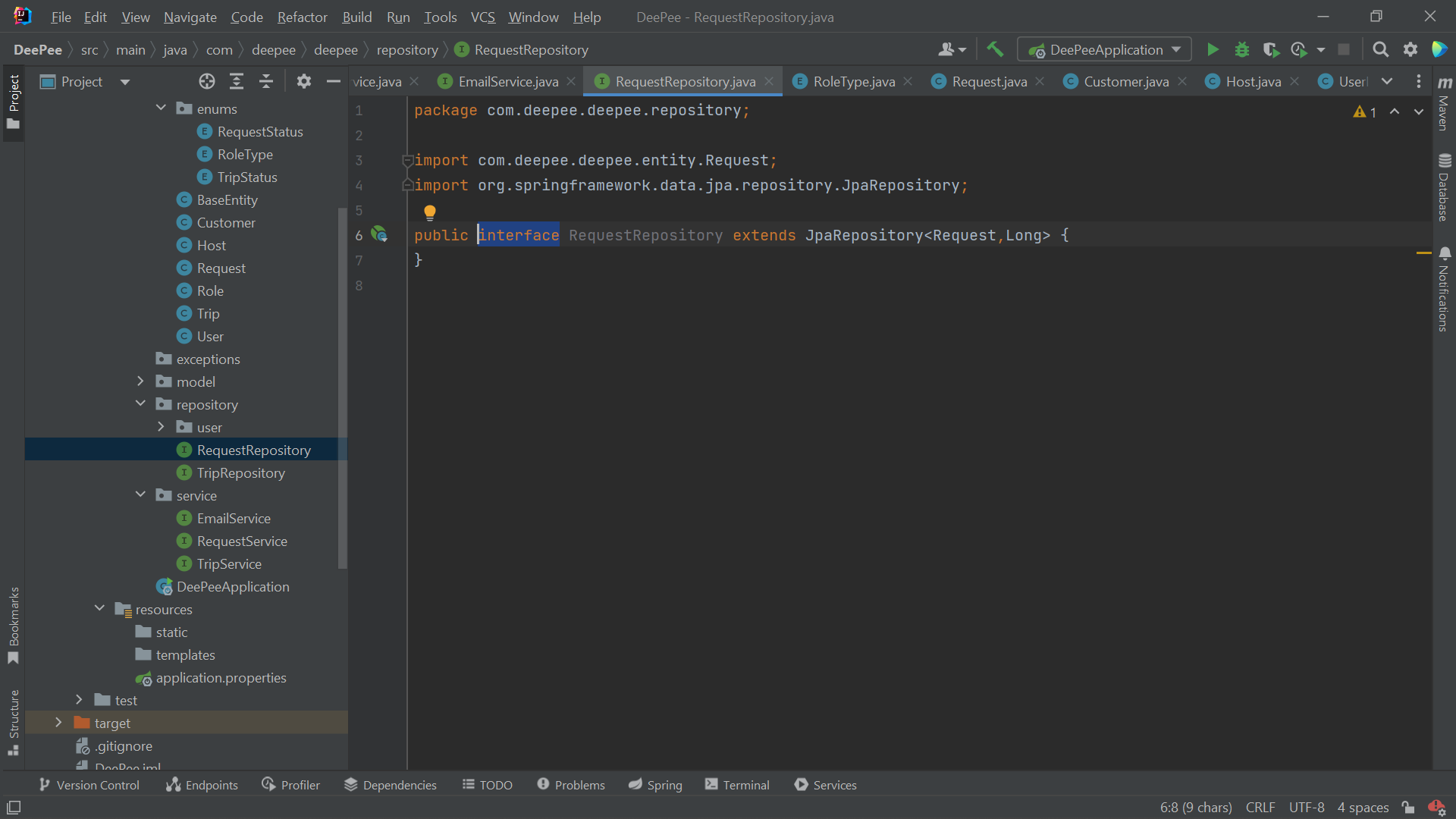
Description automatically generated with medium confidence**

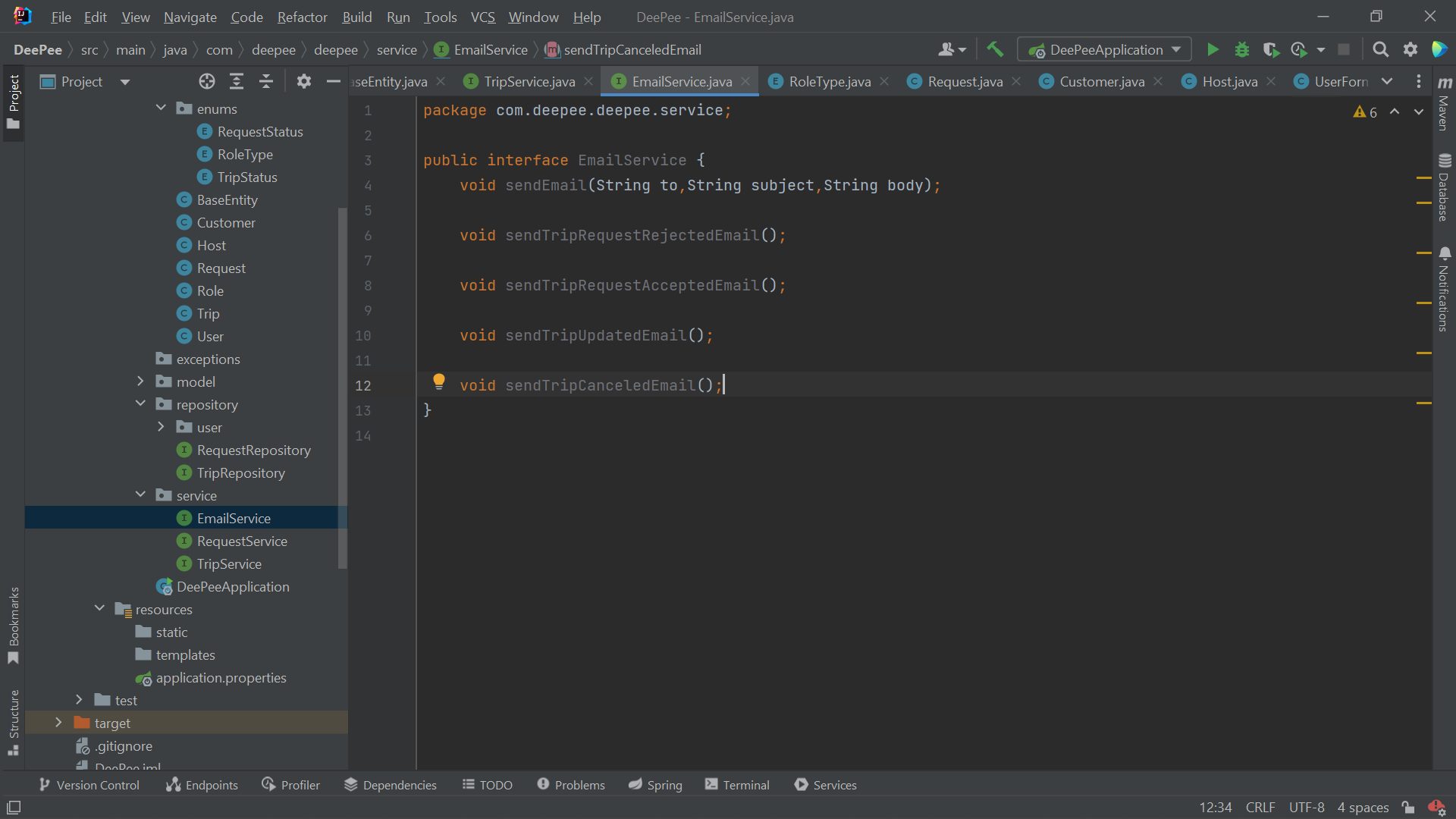
**JAVASCRIPT CODES ON VISUAL STUDIO CODE IDE**

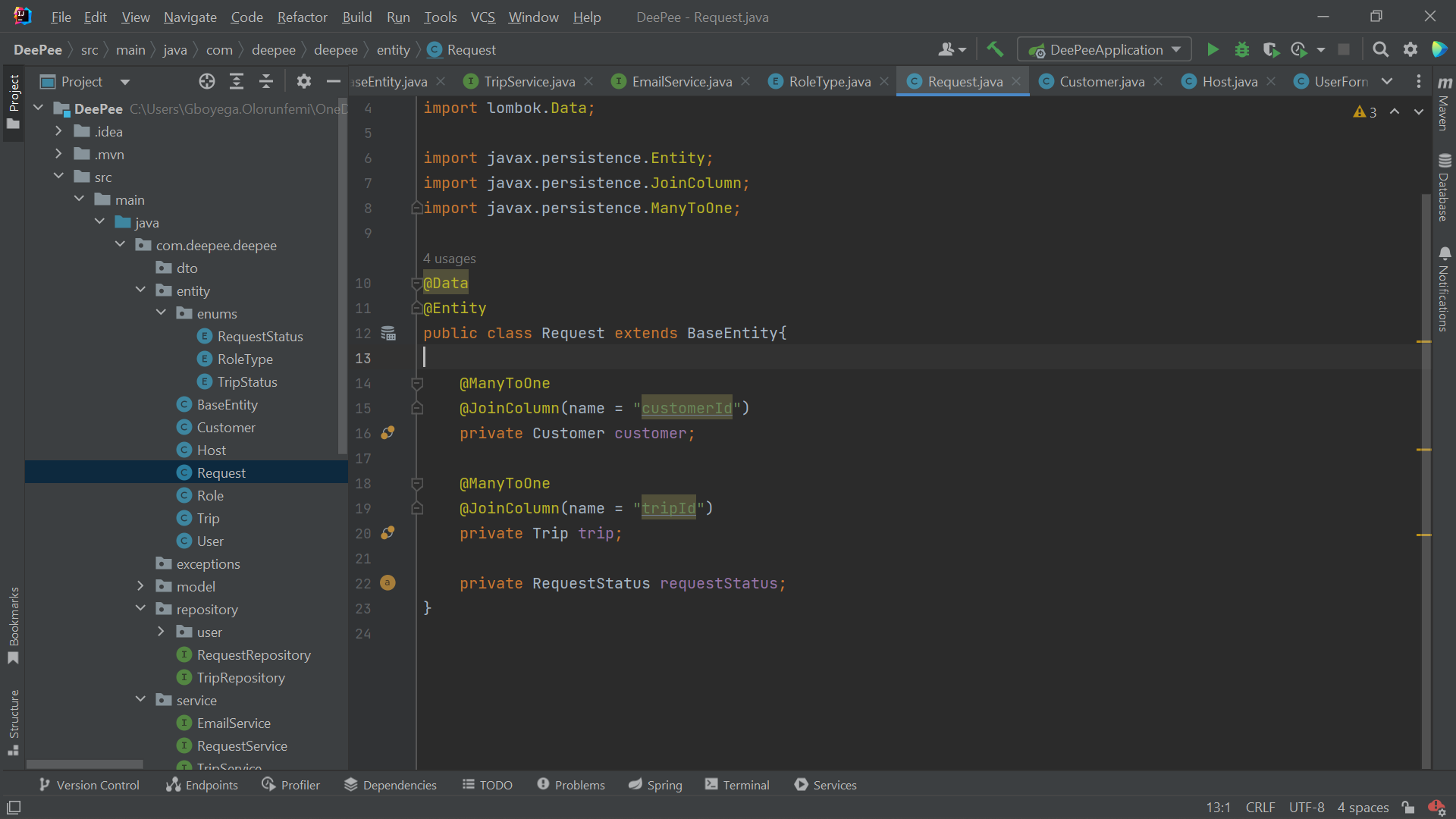
**Text

Description automatically generated**

**The Server-Side Scripting was done using Java 17 and the Framework used was Spring Boot.**







# TESTING

11.1. Test Script For DeePee (Manual Testing) **Version: 1.0**

|  |  |  |  |
| --- | --- | --- | --- |
| **Functionality** | **Test Scenario** | **Expected Result** | **Pass/Fail** |
| User Signup | Valid and Invalid/null input types | Successful and Unsuccessful Signup respectively | Pass |
| User Signup | Repeated email input | Unsuccessful Signup | Pass |
| User Login | Valid and Invalid input details | Successful and Unsuccessful Login respectively | Pass |
| Host Upload Trip | Attempt to fill just one field and Filling all input fields | Unsuccessful and Successful Upload respectively | Pass |
| Customer Search Trip | Attempt to not fill in required field and Filling the required field | Unsuccessful and Successful Search respectively | Pass |
| Customer Request Trip | After searching and selecting a trip | The Customer should see all pending requests and should be able to make a request | Pass |
| Host Accept Trip | After getting a trip request | The Host should see all requests for his/her trip and should be able to approve the request | Pass |

# IMPLEMENTATION

<https://github.com/gboye-g/Coding-Academy-Project-Frontend.git>  
  
<https://github.com/gboye-g/Coding-Academy-Project-Frontend.git>

12.1 Login: Users get to log in to access the defined features based on their role as either a customer or a host.

12.2 Sign up as a customer/host: Users create an account based on the role they want perform.

12.3 Upload a trip page: The Host enters the trip detail and then clicks the upload button for the detail to be added as a record to his/her Uploaded Trips Tab on his/her dashboard

12.4 Find a ride page: The Customer enters the number of seats and/or any other search parameter of his/her choice and then clicks the Search button to see a tabular display of the Contact details of the Hosts that conform to the Search. This table is displayed once the customer clicks the Search button as the Customer is redirected to his/her Active Search Tab on his/her Dashboard.

Login, Signup and Trip Entry forms

**A screenshot of a computer

Description automatically generated with medium confidence**

12.5 Landing page: Users see an overview of what the web application is about.

12.6 The home page: summarizes the purpose of the application.

12.7 The about page: elaborates on the strengths of the business to convince users to use the application.

12.8 The contact page: tells users how to reach the business.

A screenshot of a video game

Description automatically generated

**AFTER BOTH THE HOST AND THE CUSTOMER SIGN UP AND LOGIN, THE USERS** **DASHBOARDS EXPERIENCE IS AS FOLLOWS**

12.9 Active search tab: The Active Search Tab holds only the last search result with an additional column titled "Add to requests tab". This column has the + button to add a record to the Customer’s Trip Request page.

12.10 Customer’s trip request tab: This Trip Request Tab is a tabular representation of the record of all trips added by the Customer from the Active Search Tab. The table records contain the Trip’s Host details and are arranged from the closest start time to the farthest.   
The table has the following additional columns:  
One column, with the – button  
Another column, showing the Trip status. The trip status is dynamically updated as either pending request, pending acceptance, accepted, timed out, ongoing, or completed, depending on the action performed.  
Another making provision for the customer to input the agreed trip price  
Lastly, an additional column to request any of the trips.

12.11 Host trip upload tab: The Host is automatically directed to this page after each successful upload. This page is a tabular representation of the Host’s uploaded trips. Each record in the table would have 2 buttons. One to delete a trip from the page and another to start or end the trip.

12.12 Host’s trip requests tab: The Trip Request Tab is a tabular representation of the record of all trip requests sent to the Host by Customers. The table records contain the Customer’s contact and trip details.  
The table has the following additional columns:  
One column, with the – button  
Another column, showing the Trip status. The trip status is dynamically updated as either pending acceptance, accepted, timed out, ongoing, or completed, depending on the action performed.  
Lastly, another with a toggle button for the Host to accept any of the trips.

# FUTURE ENHANCEMENT

**This includes and is not limited to:**

* In-house vehicle inspection facility whereby an inspection validity code would be issued upon the host’s vehicle validation.
* Ability for customers to share the progress of their trips on WhatsApp and other messaging applications.
* An in-app chat box for easy communication between hosts and customers.
* Infusion of a goods delivery service alongside transportation of people.
* Addition of Google maps capabilities that allow choosing locations directly from the map as against typing in locations and tracking the real-time location and distance apart of the hosts and customers.
* A feature that allows customers and hosts to drop ratings, comments, and complaints about each other and concerning the web application.
* A feature that allows both the customer and the host to cancel the trip before it starts and the person(s) involved receives a notification that the trip has been canceled.
* In-app payment portal.
* There would also be a portal for all users whereby they can upload their profile picture, a means of identification, and then vehicle pictures for hosts. All these would be validated to ensure the safety of all users of the application.
* A sophisticated price estimation system for different trips.
* An extra feature whereby hosts can include the route they are traveling through so that customers can in turn search for hosts based on their preferred routes.
* Changing forms to a pop-up to reduce switching of pages.
* Ability for hosts to get other forms of alert for customer trip notifications like SMS and WhatsApp message.
* Ability for hosts and customers to have a tabular view of data for all kinds of queries related to their activities on the platform.
* Ability for customers to see other related search queries or suggestions in relation to their search pattern and history
* In-app storing of cookies and caches for input suggestions and other AI features for better user experience
* A tab/option for the customer to see an auto-updated list of all available drivers
* Ability of users to upload and book rides over a long-time frame.
* Restrictions on the conditions with which a customer can search for, request or cancel a trip and conditions for which a driver can upload, approve or cancel a trip. Users can also request or cancel multiple or all trips at once.
* Then there would be customer trips table whereby each customer’s trip has its unique, source and destination location, trip prize and source and destination time.
* On the search for trip page, the customer should be able to search using recent search history
* Change login to phone number
* Responses from backend validations (POPUPS, emails...)asper you have booked, you are logged in, you are signed out, wrong email, wrong password, email already exist

# CONCLUSION

The IntelliJ IDE was used and not Eclipse because it is the most Java-friendly IDE, with a lot of IntelliSense and error-catching capabilities. It is also more compatible with the Spring boot framework.

The Spring Boot framework was used because it caters to a lot of coding that is necessary for Spring. The framework also uses numerous annotations that efficiently handle the relationship between objects.

Although the REACT library provides a more organized and less cumbersome way to develop frontend codes and it also makes the connection to the Database and backend easy, It was not used at the time of the front end development due to a knowledge gap coupled with time constraints.

In summary, getting this application running to this point is a very memorable experience. Without the help of online resources, my colleagues, and my trainers the story would have been different. All Glory to God.

# REFERENCES

[What Is the Agile SDLC and How Does It Work? | Synopsys](https://www.synopsys.com/glossary/what-is-agile-sdlc.html#:~:text=Definition,last%20two%20to%20four%20weeks.)

[SDLC Agile Model (w3schools.in)](https://www.w3schools.in/sdlc/agile-model)

[Murugesh.viswanathan@latentbridge.com](mailto:Murugesh.viswanathan@latentbridge.com)

[Shanmugam.vengidusamy@latentbridge.com](mailto:Shanmugam.vengidusamy@latentbridge.com)

[saurav.ghosh@latentbridge.com](mailto:saurav.ghosh@latentbridge.com)