

1. Project Overview

1.1 Project Objectives

- The objective is to develop a prototype application that aims to create an easy to navigate rental process.

1.2 Scope

- Key Features and Functionalities:
 - Start a reservation
 - View/Modify/Cancel a reservation
 - Browse vehicles for rent
 - CRUD operations on users
 - CRUD operations on vehicles
 - CRUD operations on reservations

1.3 Target Audience

- Customers
- Customer service representatives (CSR)
- System administrator(s)

2. Project Approach

2.1 Development Methodology

- Agile Scrum methodology because it enables incremental deliveries and adaptability to changing requirements.

2.2 Project Timeline

Sprint 1	Sprint 2	Sprint 3	Sprint 4
Creating GitHub Repo			

Setting Up ReadME file			
Writing a Wiki Page			
User Stories			
Task Assignment			
Plan for next Sprint in Wiki			

2.3 Collaboration and Communication

- Communication Channels: Regular team meetings, private forum in Moodle, Concordia, Discord Calls.
- Collaboration Tools: GitHub for version control, bug tracking, task management.

Project Approach and Technology

3. Technology Stack

3.1 Backend Frameworks

3.1.1 Python

We will be using Python along with frameworks like Django or Flask, for backend development because of its simplicity and vast ecosystem, ensuring ease of use and extensive support for developers.

Despite its slower performance for some tasks compared to compiled languages, Python's readability and the strong community backing make it a favorable choice. This balance of features and the ability to rapidly prototype and deploy makes python a great choice for our backend development strategy.

We will use Flask for the framework because of its simplicity and flexibility, making it ideal for both small projects making it possible to expand as well. The minimalist approach allows for rapid project setup and prototyping. Flask avoids unnecessary bloat which can help speed up the process of finishing the project. In addition, Flask can be extended through a wide array of community-supported extensions to accommodate more complex functionalities like database integration for the cars and user authentication.

3.1.2 Node.js/Express.js

Node.js with Express.js will be considered for backend development. Node.js enables non-blocking, event-driven architecture suitable for I/O-heavy applications such as the car rental project that will be built. The main potential weakness with it will be the Callback hell (mitigated by async/await) which is less suitable for CPU-intensive tasks.

3.1.3 .NET Core

We're considering .NET Core for backend development because it works across different platforms which is perfect for our project. .NET Core is a solid framework known for its good performance and scalability. It can run on different platforms, performs well, scales easily, and has a wide range of built-in functions. The main downside is that it is a complex framework that might be challenging for developers who aren't familiar with the .NET ecosystem to learn.

3.2 Frontend Frameworks

3.2.1 React.js

React.js will be utilized to construct the frontend user interface due to its efficient virtual DOM, which ensures fast rendering and component reusability. Its extensive ecosystem further enhances the development process by offering a wide range of tools and libraries, enabling the creation of dynamic and responsive user interfaces.

3.2.2 Vue.js

- Vue.js is considered for frontend development because of its user-friendly syntax and easy integration with existing projects. Its strengths are its approachable syntax, clean documentation, and easy integration capabilities. Vue.js is

well-suited for lightweight applications and rapid prototyping, making it a strong contender for frontend development for our project.

3.2.3 Angular

Angular will be considered for frontend development, especially for enterprise-scale applications. Angular provides a comprehensive solution with strong TypeScript integration. Its strengths are the full-featured framework, strong TypeScript integration and robust testing tools. It is complex to set up and has a steep learning curve which might cause some issues in the process.

4. Integration and Interoperability

4.1 Backend-Frontend Integration

- We will use a custom API for communication between backend and frontend.

4.2 Third-Party Services

- We will use Google Maps API to derive a way from the customer to the nearest Agency

5. Security Considerations

- Overview: Implement secure authentication and authorization mechanisms.

6. Conclusion

In conclusion, our project aims to develop a prototype car rental application with a focus on essential features, agile development, and secure implementation. By leveraging a carefully chosen technology stack and emphasizing collaboration and communication, we are confident in our ability to deliver a robust solution that meets the needs of our target audience.