ADDIS ABABA UNIVERISITY

ADDIS ABABA INSTITUTE OF TECHNOLOGY

SCHOOL OF INFORMATION TECHNOLOGY AND ENGINEERING

CAR RENTAL PLATFORM-- Project Proposal

Section 1 Group 6

ADVISOR: Nuniyat Kifle

PREPARED BY: -

Name	*ID no*
BEKA BIRHANU ATOMSAU	UGR/ 3402/14
BETHEL DEREJE TEFERI	UGR/1397/14
DAWIT BELAY YIBELTAL	UGR/8622/14
LEMI DINKU GILO	UGR/3860/14
SAMRAWIT DAWIT MENGSTU	UGR/0221/14
SIMON DEREJE WOLDEAREGAY	UGR/0952/14

Table of Contents

ACRONYMS

UI – User Interface

API – Application Program Interface

Baas – Backend as a Service

SDK – Software Development Kit

IDE – Integrated Development Environment

JS - JavaScript

HTML– Hypertext Markup Language

CSS – Cascading Style Sheets

ABSTRACT

This project aims to create an online platform connecting car owners with potential renters, offering a convenient and trustworthy solution for car rentals. The system prioritizes building trust through verification and includes a review system for users to make informed decisions. The primary goal is to enhance the car rental experience for both car owners and renters by providing an easy-to-use platform.

1. INTRODUCTION

1.1 Background

Our project, the Car Rental Platform, is a response to a pressing need in the Ethiopian market. The current car rental process in Ethiopia relies heavily on traditional, time-consuming methods or intermediaries like brokers. This project aims to digitalize and streamline car rentals by connecting car owners directly with renters. We intend to create a user-friendly online platform, eliminating the need for intermediaries and enhancing the overall experience for both parties.

1.1.1 Customer Base

Our target audience is composed of individuals in Ethiopia who require access to rental cars, whether for personal or business purposes. The project is designed to cater to the diverse needs of customers in search of rental vehicles while saving them valuable time in the process. Simultaneously, it offers car owners an opportunity to efficiently manage their vehicles and maximize their earnings.

1.1.2 Nature of the Project

The Car Rental Platform represents a groundbreaking product that introduces an entirely new approach to car rentals in Ethiopia. It is an innovative solution designed to revolutionize the car rental industry. The platform serves as a comprehensive online marketplace that provides an extensive range of vehicle options to customers while affording car owners the opportunity to optimize their earnings through effective vehicle management.

1.1.3 Project Timeline and Cost

In the academic context, we anticipate completing this project within the current semester, spanning approximately three months. Specific costs related to the project will be determined during subsequent project planning phases, covering development resources, marketing, and maintenance expenses.

1.2 The Existing System

The traditional car rental system in Ethiopia has been characterized by inefficiency, inconvenience, and the involvement of costly intermediaries. Customers faced burdensome manual searches for suitable vehicles, while brokers added complexity and undisclosed fees. The Car Rental Platform intends to bridge this gap by providing a streamlined, transparent, and cost-effective solution.

1.3 Statement of the Problem

Ethiopia's car rental industry has long suffered from outdated, inefficient methods, impacting both customers and car owners. Lengthy and manual searches, coupled with a lack of pricing transparency, have led to increased costs and decreased trust in car rental services. This problem extends beyond the target population, affecting the broader Ethiopian economy by impeding efficient transportation.

The proposed solution, the "Car Rental Platform," aims to address these issues by streamlining the rental process, reducing costs, and fostering trust between car owners and renters. This can stimulate economic growth and make vehicles more accessible for various purposes.

1.4 Objective of the project

1.4.1 General objectives

The general objective of the Car Rental Platform is to create a reliable and efficient platform connecting customers with car rental providers, simplifying the process, enhancing convenience, and facilitating seamless transactions.

1.4.2 Specific objectives

- Simplify the car rental process for customers and car rental providers.
- Enhance the overall user experience by providing a user-friendly interface.
- Offer a wide range of rental options for customers to choose from.
- Facilitate direct communication and coordination between customers and car rental providers.
- Ensure transparency and reliability in the car rental process.

1.5 Proposed system

Overview:

The proposed system is a web-based car rental platform designed to streamline the process of renting vehicles by serving as an intermediary between customers and car rental providers. It aims to simplify the search, booking, and coordination process for users.

Key Features:

1. **User-Friendly Interface:**
    - The system will offer a user-friendly interface for customers to search for available rental cars based on their preferences, including location, dates, vehicle type, and other relevant criteria.
    - The goal is to provide a seamless and intuitive user experience, making it easy for customers to find the right rental option.
2. **Comprehensive Listings:**
    - The system will aggregate a comprehensive list of car rental options from various car rental providers.
    - This feature ensures that customers have a wide array of choices, making it convenient for them to select a suitable vehicle for their needs.
3. **Facilitated Booking Process:**
    - After selecting a rental option, the system will facilitate the booking process by providing essential information and contact details of the car rental provider (carter).
    - This enables customers to directly communicate and coordinate with the car rental provider for tasks such as payment processing, rental agreement, and any specific requirements.
4. **Reliability and Transparency:**
    - The web-based rental system is built to be a reliable platform, ensuring transparency and efficiency in connecting customers with car rental providers.
    - The system's purpose is to simplify the rental process, offering a centralized platform for customers to find and book rental cars.

1.6 Feasibility Study

1.6.1 Economic feasibility

As the project is for educational purposes and no funding is available, the economic feasibility of deploying the car rental platform is assessed without any cost. To cut costs, open-source technologies and free resources will be used.

- Developmental costs
  - An IDE that is available free of cost, without any associated licensing fee will be used. The frameworks to be used are also freely available. The version control system is also free to use and there are platforms that offer free hosting services for its repositories. For testing and deployment, free testing and deployment tools will be used.
  - The team can utilize open-source community which offer tutorials, documentation, and free web tools. This makes learning and addressing problems more effective and eliminates the need for costly support services or training.
- Operational Costs
  - Free cloud hosting offers reliable performance and scalability for website hosting without significant operational costs. If finding a free service is challenging, local hosting on local servers can be used.
  - The team can handle upgrades and maintenance on a regular basis, negating the need to hire dedicated operational workers or outsource the work. This strategy gives us better control over the maintenance process while reducing continuing operating costs.

1.6.2 Technical Feasibility

The technical resources that the team has access to are the main focus of this examination. It aids in the assessment of technical team's ability to translate concepts into functional systems and the capacity of its technical resources. Here is a list of the technological feasibility studies we've done.

- The system can be updated effectively and frequently because of the continuous integration and deployment capabilities of the development tools. This guarantees that the platform can easily integrate new features and enhancements and adjust to changing requirements. The selected development tools are appropriate for the project because they are widely used, offer an abundance of documentation, and offer **strong community support**.The Cloud servers and cloud database management tools that are going to be used can handle the required data storage needs.

- Secure authentication mechanisms such as are multi-factor authentication, encryption, and secure protocols, will be implemented throughout the platform. This ensures that user data and interactions are protected, minimizing the risk of unauthorized access or data breaches.
- While the initial version of the system may have limitations due to the use of open-source services, with sufficient funding, the platform will expand in the future. A larger user base and better performance can be achieved by upgrading to paid subscription plans or using dedicated hosting services, which can offer expanded storage and database management capabilities.

1.6.3 Schedule feasibility

- **Project management**
  - Project Approach: The chosen project management approach, which follows a sequential and non-iterative model, is well-suited to the project's goals. It significantly increases the likelihood of completing the project within the specified time frame.
  - Task Breakdown: The project tasks are clearly defined, and a realistic timeline is provided. This ensures a smooth project flow and on-time task completion.
  - Team Bandwidth: Team members are allocated tasks based on their expertise, ensuring that they have the necessary bandwidth to fulfill their responsibilities within the specified timeline.

- Communication and Collaboration: Weekly stand-up meetings, frequent project reviews, and status updates enable early identification of delays and allow for timely adjustments and corrective actions.
- Documentation: Requirements and design specifications are included in the project documentation, which is continually updated and maintained. This reduces the risk of misunderstandings and scope creep by ensuring clarity and alignment among team members and stakeholders.

- **Risk Assessment**
  - **Risk**: Unforeseen Absences and Skill Gaps: Potential delays in development and testing due to unexpected team member absences or skill gaps.
    - **Mitigation**: Mitigation strategies include balancing the workload, identifying skill gaps early and addressing them through online courses, and maintaining effective communication and collaboration within the team.
  - **Risk**: Delays or issues with the external software dependencies: can impact the development timeline.
    - **Mitigation**: Strategies include evaluating the reliability and compatibility of software technologies, establishing contingency plans, staying updated with the latest versions and documentation of dependencies, and addressing issues promptly..
  - **Risk**: Inefficient Usage of Version Control: Potential conflicts, code inconsistencies, or delays in code integration due to inefficient use of version control.
    - **Mitigation**: Mitigation strategies include ensuring that all team members are familiar with version control best practices, promoting regular communication and collaboration, conducting code reviews, and ensuring smooth code integration.
  - **Risk**: Project Scope Changes: Potential delays in the schedule due to project scope changes or additional requirements.
    - **Mitigation**: Defining and documenting the project scope clearly at the outset; establishing a formal change management process to evaluate and approve any scope changes.
  - **Risk**: User Acceptability Testing: The system may not meet end users' expectations or specifications if user acceptability testing is insufficient.

- **Mitigation**: Strategies include thorough user acceptability testing, early end-user involvement in the testing process, and collecting feedback to make necessary adjustments, ensuring the system aligns with user needs and offers a positive user experience.

1.7 Scope

The scope of our project is to develop a platform that facilitates connectivity between car owners and renters, ensuring reliability and trustworthiness for both parties. The system will include the following features:

1. **Connectivity**: The platform will enable car owners to list their vehicles available for rent, while renters can search and book the desired vehicles based on their preferences and requirements.
2. **Verification Measures**: Robust verification measures will be implemented during the registration process to ensure the authenticity and reliability of both car owners and renters. This will include identity verification.
3. **Review System**: The platform will incorporate a review system that allows users to provide feedback and ratings regarding their rental experiences. This feedback mechanism will help establish trust and enable users to make informed decisions when choosing whom to rent from or rent to.

The following services will not be provided by our platform:

1. **Legal Issues, Contracts, and Agreements**: Our platform will not be involved in any legal matters, contracts, or agreements between car owners and renters. All legal contracts and agreements will be conducted in-person between the involved parties, ensuring compliance with applicable laws and regulations.
2. **Transaction**: We will not handle any financial transactions related to car rentals. Payments and financial transactions will be solely conducted between the car owners and renters, utilizing their preferred payment methods.

1.8 Methodology

1.8.1 Developmental Methodology

After considering all the circumstances and carefully speculating, we have chosen to use **Waterfall Model** as a software development methodology. Among other models, we believe Waterfall Model is best suited to our project for the following reasons:

1. **Documentation Emphasis**: The Waterfall model places a strong emphasis on documentation. It requires comprehensive documentation at each stage, including requirements, design, and testing documents. This can be advantageous in projects where extensive documentation is necessary for compliance, regulatory purposes, or future maintenance.

2. **Resource Allocation**: It can be suitable when resources (human, financial, or infrastructure) need to be allocated and planned upfront. It allows for better resource management and allocation since the project requirements and scope are determined in the early stages. Given our limited human resource, it is better to focus on one stage of development at a time.
3. **Well-Defined Requirements**: When the project requirements are clear, stable, and unlikely to change significantly, the Waterfall model can be effective. It relies on upfront and detailed documentation, making it suitable for projects with well-defined requirements from the beginning.
4. **Sequential Progression**: The Waterfall model follows a sequential progression, with each phase building upon the previous one. This can be beneficial when there is a clear and logical flow of activities, and dependencies between phases are well-defined. It allows for a structured and systematic approach to development.

Consequently, since this project is well-defined with clear objectives and limited complexity, the Waterfall model can be a straightforward and efficient approach. It provides a linear path for development, which can be appropriate since the project scope is well understood and not likely to change significantly.

1.8.2 Project phases

As it uses Waterfall model, this project will have the following phases.

1. Project Planning
2. Requirement analysis
3. Design
4. Implementation and Unit Testing
5. Integration and Integration Testing
6. System Testing
7. Deployment and Maintenance

1.8.3 Requirement Collection and Analysis

Due to specific circumstances, we have decided not to conduct interviews or questionnaires as part of this process. Instead, we will rely on alternative methods to gather and analyze requirements. There are two primary reasons for this decision:

1**. Limited Budget:** As a constraint, we have a limited budget allocated for this project. Conducting interviews or questionnaires would require additional resources, such as hiring external consultants or dedicating significant time and effort from the project team. Considering our budget limitations, we have determined that alternative methods will be more feasible and cost-effective.

2. **Well-Defined Requirements**: Another key factor influencing our decision is that the project requirements are already well-defined. Through prior discussions and available documentation, we have gained a comprehensive understanding of the project's objectives, scope, and functionality. The requirements have been documented in sufficient detail, allowing us to proceed with the analysis phase without extensive data collection.

What do we use instead of Data Collection (Interviews and Questionnaires)?

Document Analysis: We will thoroughly analyze existing documentation related to the project, including business requirements documents, functional specifications, and any relevant technical documentation. By reviewing these materials, we can extract and consolidate the essential requirements. By employing this alternative method, we are confident that we can effectively collect and analyze the requirements within the given constraints. Our approach will enable us to proceed with the subsequent phases of the project.

1.8.4 Design

The design phase of a project is a critical stage where the project team defines the structure, components, and behavior of the system. In this phase we will give great emphasis to exploring different approaches to solve the problem. And try to pinpoint the best methods and designs to produce the optimal performance.

This phase will encompass 3 main components namely the user interface, system architecture, and database structure.

1. **User Interface Design**
   User Interface (UI) design involves creating the visual and interactive elements that users will interact with to accomplish tasks within the system. In this design aspect we will aim to design intuitive and user-friendly interface that enhances usability and user experience. Key activities in UI design include:
   - **Wireframe**: Creating general representations of the UI layout and structure, showcasing the placement of elements, navigation flow, and overall user interaction.
   - **Visual Design**: Defining the visual aesthetics, color schemes, typography, and graphical elements to create an appealing and consistent visual interface.

2. **System Design**
   System design focuses on defining the architecture, components, and interactions of the system. This phase determines how different modules, components, and subsystems will work together to fulfill the system requirements.

   - **Architectural Design**: Defining the overall system architecture, including the high-level structure, components, and their interactions. This involves choosing appropriate architectural patterns or frameworks.

   - **Component Design**: Designing individual software components or modules, specifying their functionality, interfaces, and interactions with other components. This includes defining APIs, data structures, and algorithms.

3. **Database Design**
   Database design involves designing the structure, relationships, and organization of the system's database. This phase focuses on creating an efficient and well-structured database schema that meets the system's data storage and retrieval needs.

We are strongly convinced that this emphasis on design phase will reduce critical issues that could emerge in the implementation phase.

1.8.5 Implementation Methodology

During the implementation phase, we will utilize wide range of software technologies (libraries, frameworks…). These are some of the software we plan to use to bring the design to life:

1. **HTML5, CSS3 and JavaScript(React.js)- for front-end**
   - **React.js** - To develop the UI components and states
   - **React router (third-party library)-** to provide routing functionality for the website
   - **CSS framework**- to build eye catching styles and animations

2. **Backend as a Service- for Back-end**

   Backend as a Service (BaaS) offers a range of features and services that simplify the development, deployment, and management of the backend infrastructure for applications.

   We opted to use BaaS for our backend needs for the following reasons:

   - **Rapid Application Development:** it offers pre-built backend services and APIs that enable developers to quickly integrate common functionalities into their applications without having to build everything from scratch.
   - **Scalability and Flexibility:** it is designed to handle scalability and can automatically scale resources based on application needs. This eliminates the need for developers to manage server provisioning and scaling manually.
   - **Cost Efficiency:** Since we intend to use **open source BaaS**. This eliminates the need for infrastructure investments and reduces operational costs associated with managing and maintaining backend infrastructure.

   Here are some common offerings provided by BaaS platforms:

   1. **User Management and Authentication:** BaaS platforms often provide built-in user management systems that handle user registration, login, and authentication. These services typically include features such as user profile management, password reset, email verification, and social media login integration.

   2. **Data Storage and Database Management:** BaaS platforms offer scalable and managed data storage solutions. They provide APIs and SDKs to interact with databases. This allows developers to store and retrieve data efficiently without worrying about database administration tasks.

3. **File Storage and Content Delivery:** BaaS platforms provide storage solutions to handle file uploads, storage, and retrieval. This allows developers to store user-generated content, media files, documents, and other assets.

4. **Push Notifications and Messaging:** BaaS platforms often include services for sending push notifications and in-app messaging. These services enable developers to send real-time notifications to users on various platforms (web, mobile) and engage with them effectively**.**

5. **Serverless Computing and Functions as a Service (FaaS):** Some BaaS platforms offer serverless computing capabilities, allowing developers to deploy functions or code snippets that run in response to events or specific triggers.

Therefore, to utilize features listed above we will be using an **Open Source BaaS** like **Supabase**.

1.8.6 Testing

Testing will encompass two major subcomponents, unit testing and system testing.

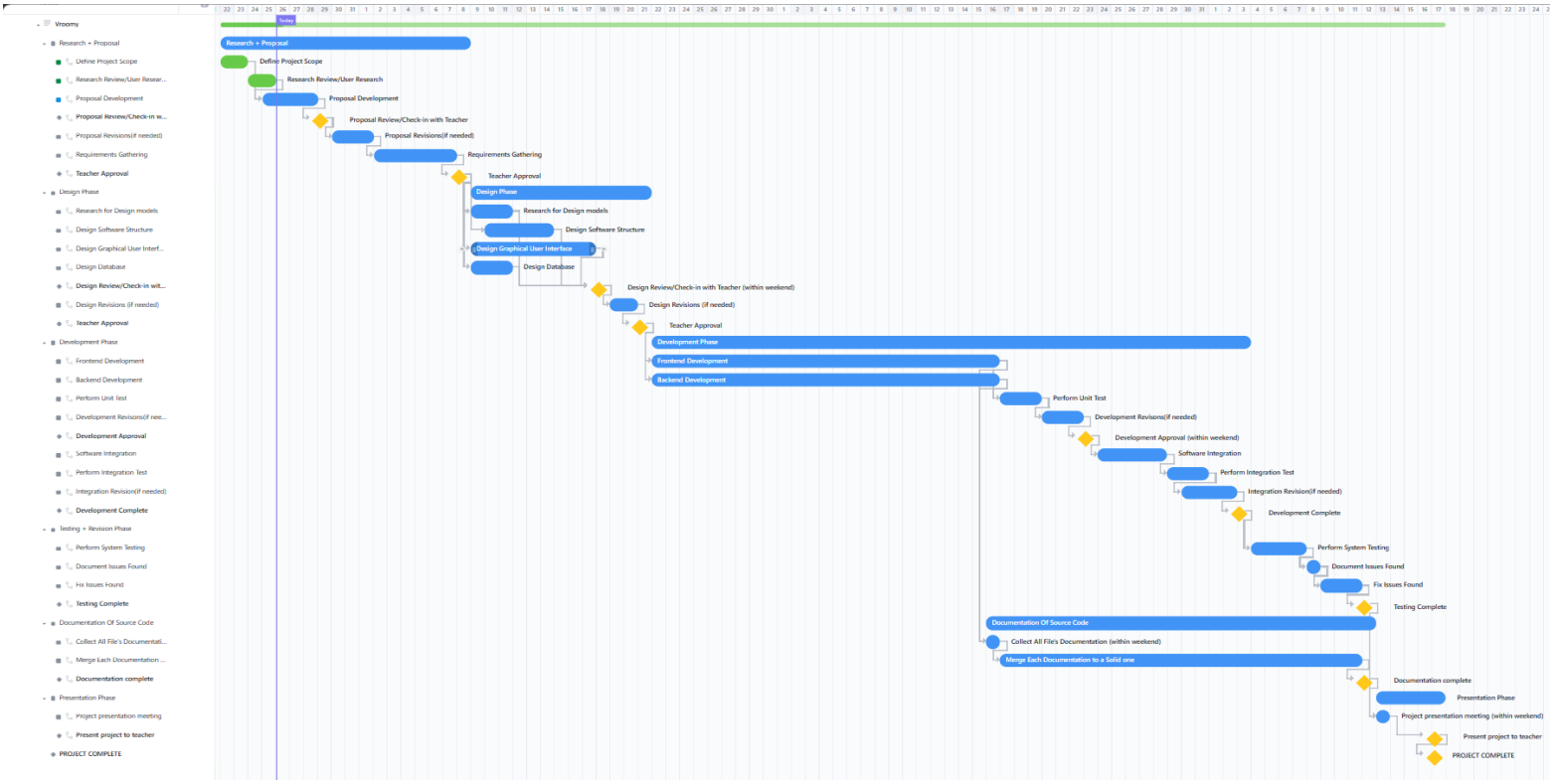
1. Unit Testing – This will be done alongside the implementation phase. A component is implemented using the previously defined design. At this time the component has to be tested whether can accomplish what it is set out to do. Issues found in component testing will be handled before proceeding to the next stage (system integration).

2. System Testing—after the components are implemented and tested, the components will be integrated to give the whole system. During, this integration the inter-communication among the components will be tested and ensure that the system can deliver what it is planned to.

During both unit testing and system testing, we will be using testing tools, which can significantly simplify the testing process, enhance efficiency, and improve the overall quality of the software.

1.9 Project Management Plan

1.9.1 Time Management Plan



1.9.2 Quality Management Plan

Quality Risks: Potential risks related to software quality may include coding errors, incomplete testing, and inadequate documentation.

Project Quality Management Plan:

1. Quality Objectives:
   Ensure that the software meets industry standards and is bug-free.

2. Quality Assurance:
   Regular code reviews and testing will be conducted throughout the project.

3. Testing Considerations:
   Unit testing, integration testing, system testing, and user acceptance testing will be conducted at various project stages.

4. Usability and Acceptance Testing (Optional):
   Usability testing will be conducted to ensure the software is user-friendly.

1.9.3 Communication Management Plan

Principles for Reporting and Distributing Information:
Utilize Clickup for milestone tracking and task allocation and git/github for version control.

I) Internal Communication and Reporting:

a) Daily Status Reports:

- **Type of Meeting:** Telegram Group Message
- **Participants**: Project Team
- **Purpose**: Provide a daily report on the current status, including any errors, challenges, or assistance required.
- **Schedule**: Every morning.

b) In-Person Weekly Meeting:

- **Type of Meeting**: In-Person
- **Participants**: Project Team
- **Purpose**: Discuss the current status, address errors, and provide assistance.
- **Schedule**: Every Wednesday morning.

c) Weekly Google Meeting:

- **Type of Meeting**: Google Meeting
- **Participants**: Project Team
- **Purpose**: Review weekly progress and plan for the next week.
- **Schedule**: Every Saturday.

d) Millstone's meeting:

- **Type of Meeting**: Google Meeting
- **Participants**: Project Team
- **Purpose**: Discuss milestone achievements, provide feedback, and address any issues.
- **Schedule**: upon Millstones.

e) Final Meeting (Before Presentation):

Type of Meeting: In-Person
Participants: Project Team
Purpose: Prepare for the final project presentation and address any remaining issues.
Schedule: Prior to the final project presentation.

II) External Communication and Reporting:

Teacher Reports upon Request:

- **Type of Report:** As requested by the teacher
- **Participants:** Project Team
- **Purpose:** Provide detailed reports to the teacher as requested for project updates, status, and any specific information required.