Assignment: Car Rental Service
Applying the SDLC as You Know How Today.

Overview

The purpose of this assignment is to provide students with practical exposure to the Software Development Life Cycle (SDLC), covering the phases of requirements gathering, design, and implementation. The model project selected is a Car Rental Service, where students will learn how to translate a real-world problem into structured requirements, design system architecture using UML modeling, apply design patterns, and implement core functionalities under an Agile Scrum framework.

The emphasis of this assignment is on demonstrating adherence to the SDLC process and the ability to break down an idea into clear requirements and architectural decisions, rather than solely on the final implementation quality. Students will also be expected to justify their choice of technology stack and approach to development.

Assignment Objectives

By completing this assignment, students should be able to:

- Understand and apply the phases of the SDLC in a practical project setting.
- Translate a problem statement into Requirements.
- Create plans, requirements, design and an implementation.
- Evaluate and justify technology stack choices (e.g., programming language, database, frameworks, tools).
- Deliver a working prototype of a Car Rental Service system that demonstrates the integration of requirements and design into implementation.

Project Description and Guidelines: Car Rental Service

You are required to design and implement a **Car Rental Service** that enables users to rent cars online. The system should support the following:

- Customer registration and login.
- Viewing available cars by category (e.g., sedan, SUV, economy, etc).
- Booking cars for a specified time period.
- Payment and confirmation process.
- Admin functionality for managing cars, availability, and bookings.

This project serves as the case study for you to apply SDLC concepts as you approach them today, and will be used to contrast how you develop your projects as learned in the class. Hopefully the gap will be a beneficial learning experience.

Assignment Tasks Breakdown

Phase 1: Planning (10)

- Work with your team to plan out your project:
    - **What** you are going to do and why.
    - **How** you are going to do it and why.

Phase 1: Requirements Gathering (20 marks)

- Identify **functional requirements** (e.g., booking, payment, car availability, administration, etc).
- [**OPTIONAL**] Identify **non-functional requirements** (e.g., performance, scalability, security).
- Document requirements in a structured form (e.g. Word, Visio, Visual Paradigm)

Phase 2: System Design (30 marks)

- Develop diagrams (Word, Visio, whatever you choose) that represent the following:
    - Architecture (block diagrams. Visio or Visual Paradigm work well)
    - Design (class diagrams. Visio or Visual Paradigm work well)
    - Dynamic Flows (diagrams that depict how the objects in the system will interact with one another. Visio is a good choice for simple diagrams of this nature. Later in the semester we will use UML Sequence Diagrams)

    - You may choose whatever tool for these diagrams that make you most efficient. In this class you will learn UML that will be used for all diagrams and information in your projects)

Phase 4: Technology Stack Justification (10 marks)

- Identify chosen programming language, frameworks, database, and tools.
- Justify why each choice is suitable for the system.

Phase 5: Prototype Implementation (15 marks)

- Implement core functionalities of the Car Rental Service (minimum: registration/login, viewing cars, booking functionality).
- Ensure the implementation aligns with requirements and design.
- Submit source code and demonstration to the GitHub that your team will use for both semesters.

Phase 6: Final Report and Reflection (10 marks)

- Submit a final report summarizing the SDLC process followed.
- Include challenges faced and lessons learned.
- Reflect on the process/approach you chose to complete this exercise, and WHY.

Grading Rubric (Total: 100 Marks)

Component	**Marks**	**Evaluation Criteria**
Planning	10	Quality and completeness of the initial plan.
Requirements Gathering/Analysis	20	Completeness, clarity, and structure of requirements, translating them into an architecture.

Component	**Marks**	**Evaluation Criteria**
Architecture	10	Correctness of the system architecture to support the requirements.
Design	20	Correctness and detail in diagrams that support the architecture and capabilities.
Technology Stack Justification	10	Logical reasoning behind choices; alignment with system requirements.
Prototype Implementation	20	Functionality, adherence to requirements/design, basic usability.
Final Report and Reflection	10	Clarity, completeness of write-up
Total	100	

Submission Guidelines

Deliverables must include:

- Requirement document.
- Architecture and design documents.
- Technology stack justification.
- Prototype code and demonstration video/screenshots.
- Final report and reflection.

All documents should be professionally formatted and submitted in PDF format along with source code in a compressed folder.