Dan's Jivin' Car Barn

Project Overview

This project aims to build an online car rental service, including functionality for customers and employees alike. The website will include authentication capabilities for its users, including a built-in system that allows users to keep track of their money. Different kinds of cars can be reserved by users for different periods of time, provided that they have the money to continue renting the car. There will also be an option for customers to purchase insurance for their rentals.

Team Organization

We are adopting the Scrum philosophy in our group, with the role of Scrum Master and Product Owner rotating through team members after every Sprint in the following order:

1. Hyrum
2. Shawn
3. Max
4. Stockton
5. Braden

The Developers are responsible for delivering the work in the current Sprint. In Sprint meetings and standups, Developers should report their progress honestly, including any blocking issues that arise. Developers will adhere to Definition of Done when completing a task to ensure it is completed correctly. It is also the role of the Developer to create a plan for each Sprint, including the Sprint Backlog.

The Product Owner is responsible for setting the direction of the project to ensure that the project is on track to meet the requirements. This involves prioritizing tasks and selecting the tasks of the current Sprint. Other team members may push back on decisions, but the Product Owner has final say on which tasks are selected each Sprint.

The Scrum Master is responsible for ensuring that each team member is supported in their role. In Sprint meetings, they ensure that value is clearly described and direction is clearly set by the Product Owner. They help remove obstacles that impede development and ensure that the expected amount of work is being done. Additionally, the Scrum Master will take responsibility for blockers outside the team's ability to resolve.

Software Development Process

The development will be broken up into five phases. Each phase will be a little like a Sprint in an Agile method and a little like an iteration in a Spiral process. Specifically, each phase will be like a Sprint, in that work to be done will be organized into small tasks, placed into a "backlog", and prioritized. Then, using on time-box scheduling, the team will decide which tasks the phase (Sprint) will address. The team

will use JIRA to keep track of tasks in the backlog, those that will be part of the current Sprint, those in progress, those that are in code review and those that are done. Code that has been written must be reviewed by at least one other team member before it is merged into the master branch.

Each phase will also be a little like an iteration in a Spiral process, in that each phase will include some risk analysis and that any development activity (requirements capture, analysis, design, implementation, etc.) can be done during any phase. Early phases will focus on understanding (requirements capture and analysis) and subsequent phases will focus on design and implementation. Each phase will include a retrospective.

Phase	Iteration
1.	Phase 1 - Requirements Capture
2.	Phase 2 - Analysis, Architectural, UI, and DB Design
3	Phase 3 - Implementation, and Unit Testing
4	Phase 4 - More Implementation and Testing

We will use Unified Modeling Language (UML) to document user goals, structural concepts, component interactions, and behaviors.

Communication policies, procedures, and tools

Communication Procedures: We will be doing daily, asynchronous chat standup meetings via Discord. We will also be engaging in weekly in-person meetings, either truly in-person or via Zoom.

Tools: We will be using Discord for chat communications and voice chats, Zoom for video calls, Jira for project tracking.

Communication Policies: Daily asynchronous standup meetings in Discord chat, this is a daily check in on what people are working on, what people need help with, and possible concerns that have come up. Weekly standup meetings in-person every Tuesday, Wednesday, and Thursday at 4:00pm. These in-person meetings will involve more structure and depth to go over what people are working on and possible issues that have arisen.

Configuration Management

See the README.md in the Git repository.

Risk Management

The largest risk that has been noted is that if a user changes their contact information while they have an active reservation, making it hard to track them if a user drives off with a vehicle. This is mitigated by

attaching a current user to a reservation along with attaching the user's information at the time of reservation.

Users can possibly get locked out of their profile.

The manager could softlock the whole app by accidently changing their own permissions making it impossible for any changes to be made in the app. This is mitigated by requiring at least one manager in the application.