Dans Car Barn Rent a Car

# Project Overview

This project aims to build a car rental web app that will allow users to rent a car with a few clicks of a button.

The project will allow users to make an account, transfer funds, and select certain cars for specific days. The system will have employee logins, as well as customer logins.

# Team Organization

Project Manager: Josh Hatch

Designers and Developers: Michael Hanks, Kaiden McMillen, Ryan Gubler

# 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 a Scrum Board to keep track of tasks in the backlog, those that will be part of the current Sprint, those in progress, and those that are done.

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

GitHub – Our main resource for combining everyone’s work is by using GitHub. Our entire project is stored in a repository. This is so we can track data, submissions, communicating with graders, etc.

Discord – Our main way of communicating is through discord. We can have calls, chats, etc in order to maintain clear and concise communication. Most of our team meetings will take place in discord calls. Discord also has the ability to screen share.

In person team meetings – The agreement is to meet in person before each Sprint is close to being finished. This is so we can make sure we are all on the same page with everything and we all have a clear understanding with what is going on.

# Risk Analysis

* Logging in
  + Likelihood – Low
  + Severity – Medium
  + Consequences – Customer will not be happy with situation, considering they cannot really do anything specific unless logged in. System should not crash if logging in has failures.
  + Workarounds – Make sure there are limited errors logging in, also we could use ID or other form of identification in person. Also could send verification email to show us when they arrive.
* Database
  + Likelihood – Low
  + Severity – Maximum
  + Consequences – The entire web app will not work properly, entire scheme is based upon full functionality of the database.
  + Workaround – Currently none. The program relies on the database.
* User Interface
  + Likelihood – Low
  + Severity – Maximum
  + Consequences – Users cannot do anything on the website.
  + Workaround – Currently None. The system does not work properly and will lose functionality.
* Manager paying employees
  + Likelihood – Low
  + Severity – Low
  + Consequences – Employees do not get paid, resulting in people quitting.
  + Workaround – Boss keeps record on paper as well to be able to distribute cash.
* Manager adding new cars to site
  + Likelihood – Low
  + Severity – Low
  + Consequences – Less cars means less ability to make money.
  + Workaround – None

# Configuration Management

See the README.md in the Git repository.