**CS673 Software Engineering** 

**Team 5 - PixelPeak**

**Project Proposal and Planning**

| Team Member | Role(s) | Signature | Date |
| --- | --- | --- | --- |
| Kenny Light | Team Leader | *Kenny Light* | 9/8/2024 |
| Abdallah El Hamawi | Requirement Leader | *Abdallah El Hamawi* | 9/8/2024 |
| Richard Frechette | QA | *Richard Frechette* | 9/8/2024 |
| Mateus Silva | Design & Implementation | *Mateus Silva* | 9/8/2024 |
| Max Zink | Configuration Leader | *Max Zink* | 9/9/2024 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Revision history**

| **Version** | **Author** | **Date** | **Change** |
| --- | --- | --- | --- |
| **0** | **Abdallah, Kenny, Richard, Mateus** | 9/8/2024 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

[Overview](#_g6igqliy7rm)

[Related Work](#_bf21eadgjj29)

[Proposed High level Requirements](#_rgyo4hi9stmq)

[Management Plan](#_ts358bsdtbcv)

[Objectives and Priorities](#_nxeeppkjxgn4)

[Risk Management (need to be updated constantly)](#_tk7yixobah8p)

[Timeline (need to be updated at the end of each iteration)](#_iksrndohvx29)

[Configuration Management Plan](#_j5uvivmxqcsp)

[Tools](#_dzly5b9kz982)

[Deployment Plan if applicable](#_sd8zu6r3jisd)

[Quality Assurance Plan](#_vra5ptwu59qx)

[Metrics](#_vwjduhc9wuah)

[Code Review Process](#_hx3eaiwb8v3m)

[Testing](#_l9xnpmd6hh0y)

[Defect Management](#_5amsh8h9f0c7)

[References](#_pd9euov6m4du)

[Glossary](#_ty3i2nqffhtc)

# Overview

Our web and app development company aims to meet the growing demand for quality digital solutions in an expanding industry. As businesses increasingly rely on having a strong online presence, we provide custom-built websites and applications that are secure, scalable, and user-friendly. Our goal is to help businesses, startups, and individuals enhance their digital footprint through high-performance, tailored solutions. We will utilize the MERN stack (MongoDB, Express.js, React, Node.js) for development and Jira for efficient project management and collaboration. Our services cater to small to medium-sized businesses, startups, e-commerce platforms, non-profits, and personal brands.

# Proposed High level Requirements

* 1. Functional Requirements

**Team Leader**: As the Team Leader, I want to track overall project progress, so that the team stays on schedule and the project is completed on time.

**Requirements Leader**: As the Requirements Leader, I want to define what the app needs to do, so that the team knows what features to build.

**Configuration Leader**: As the Configuration Leader, I want to manage the code and deployments, so that the app is updated smoothly and without issues.

**Design Leader**: As the Design Leader, I want to create user-friendly and visually appealing pages, so that users enjoy using the app.

**QA Leader**: As the QA Leader, I want to test all features, so that we catch and fix bugs before users see them.

**Security Leader**: As the Security Leader, I want to ensure the app is secure, so that user data is protected and the app is safe from attacks.

* + 1. Essential Features (the core features that you definitely need to finish):

(For each essential features, please give a rough estimation in terms of person hours or an range of person hours)

**Page Layout (About Us, Reviews, Services, Contact Form)**: 10-15 person hours.

**Working Contact Form**: 10-15 person hours.

**Version Control Setup**: 3-5 person hours.

**Basic Security (e.g., secure form submissions)**: 8-12 person hours.

**Testing (basic functionality checks)**: 8-12 person hours.

* + 1. Desirable Features (the nice features that you really want to have too):

**Contact form Validation:** Check for correct input (e.g., valid email) before submitting the form.

**Basic Security**: Protect the contact form from spam (e.g., CAPTCHA), secure pages with HTTPS.

**Responsive Design**: Ensure the website works well on different devices (mobile, tablet, desktop).

* + 1. Optional Features (additional cool features that you want to have if there is time):

**Contact Form with File Uploads**: 5-8 person hours.

**Contact Form with calendar scheduling**: 5-8 person hours.

* 1. Nonfunctional Requirements
     1. **Security Requirements :**

Control who can edit or update the app.

Use HTTPS for secure communication.

Follow standard security guidelines to avoid common risks.

# Management Plan

## Objectives and Priorities

(Please describe your project objectives with highest priority first. Project Goals can include but not limited to complete all proposed (essential) features, deploy the software successfully, the software has no known bugs, maintain high quality, etc )

## Risk Management (need to be updated constantly)

(Please write a summary paragraph about the main risks your group identified and how you plan to manage these risks. Then use the separate google sheet for detailed risk management. The template is provided in the same folder with this file. Please provide the link to the sheet.)

**Risk Management Sheet Link:**

## Timeline (this section should be filled in iteration 0 and updated at the end of each later iteration)

| Iteration | Functional Requirements(Essential/Disable/Option) | Tasks (Cross requirements tasks) | Estimated/real person hours |
| --- | --- | --- | --- |
| 1 | User Management | Register User, Login User, Get User Details, Update User, Delete User | 40 |
| 2 | Car Management | Add New Car, Get Car Details, Update Car Details, Delete Car, List Available Cars | 50 |
| 3 | Reservation Management | Create Reservation,  Get Reservation Details, Get Reservation List, Update Reservation, Delete Reservation | 60 |

# Configuration Management Plan

## Tools

Git. GitHub, Visual Studio Code, Postman, jira, GitHub CI/CD, Jest, Supertest, Chai, SAST, Heroku

* 1. Code Commit Guideline and Git Branching Strategy   
     We’re going to be using GitHub Flow, which involves the following:

**Create a Branch For New Feature**: Start by creating a new branch from the main branch, where we’ll make the changes.

**Make Changes**: On your branch, make the necessary changes or additions to the codebase.

**Start a Pull Request**: Once we’re done with the feature, open a pull request. This is where other team members can review your changes and approve everything.

**After Review and Approval, Merge**: After the pull request has been approved merge into the main branch.

**Deploy**: Once merged, the changes in the main branch are ready for deployment. In many workflows, this step is automated, allowing for continuous deployment to production or staging environments.

## Deployment Plan if applicable

Large-scale deployments are not expected to occur for this project due to its highly specific nature updating databases for car rentals. However, assuming this software was commissioned for a larger company, we would utilize a canary method for both release and deployment. This would mean we would provide the application to smaller subsets of the company for testing to gather feedback before releasing it fully. Then for any updates, these too would be provided to subsets of the company in rolling deployments for user’s thoughts before deployment.

# Quality Assurance Plan

## Metrics

The metrics we will use to measure the quality of software is code coverage, response time, code readability, defect rate, and cost. Code coverage will measure the percentage of the code that is executed during the test cases, more coverage meaning more code is tested. Response time measures the time the system takes to respond and complete tasks, lower response times are better for higher performance. Discuss code readability with team members so that changes are easy to implement and the code isn’t confusing. Defect rate is the number of defects found per line of code. Number of test cases is how many tests will be run on the software to confirm operation.

| Metric Name | Description |
| --- | --- |
| Code Coverage | Measuring the percentage of code executed during testing |
| Response Time | Amount of time the software takes to complete tasks |
| Code Readability | Readableness for someone unfamiliar with the software |
| Defect Rate | Amount of defects found per line of code |
| Number of test cases | Amount of scenarios that will be tested during the testing phase |

* 1. Coding Standard

camelCase will be used for variables, functions. Constants and global variables will be written in uppercase. File names will use either snake case or kebab case. Consistent error handling will catch error messages and can provide more descriptive outputs.

## Code Review Process

Team members will review all pull requests for approval before being merged into the main branch from the development branch for that additional feature. Reviews will be done individually with some group reviews for larger changes. Reviewers should think about code readability, errors, error handling, and integration into the existing features.

## Testing

Each developer should test their code for functionality and that it performs the intended task with reasonable error handling. The QA leader with the help of other team members will test integration of the new functionality and edge case conditions that may be overlooked by the developer. Testing will be performed at major integration steps by the QA leader and results will be reported to the team.

## Defect Management

Jira will be used to track defects and bugs. The defect types will be tested for are functional, performance, compatibility, and integration defects. When defects are discovered they will be reported to the Jira dashboard with a description of the defect. If the developer cannot solve the issue, then the QA leader or other team members will work with the developer to resolve the issue.

# References

(For more details, please refer to the encounter example in the book or the software version of the documents posted on blackboard. )

# Glossary

(Any acronym used in the document should be explained here)