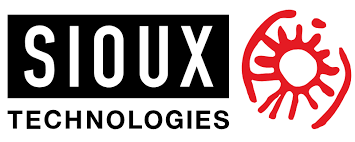
**Sioux Parking Management System**

Project Plan

**Date:** *21.06.2024 (last modified)*

**Authors:** *Desislav Hristov, Atanas Tsenov, Mihail Mihov, Stanislav Nikolov*

**Version:** *1.0*

**

Contents

[**1. Introduction** 2](#_Toc169886026)

[**2. Project Goal** 2](#_Toc169886027)

[**3. Client & Team** 3](#_Toc169886028)

[**5. Research questions and methodology** 3](#_Toc169886029)

[5.1 How can technology enhance the management of parking spaces? 3](#_Toc169886030)

[5.2 What are the best practices for secure and efficient data handling in parking management systems? 4](#_Toc169886031)

[**6. Deliverables** 4](#_Toc169886032)

[6.1 Documentation 4](#_Toc169886033)

[6.2 Front-End Application 4](#_Toc169886034)

[6.3 Back-End Infrastructure 4](#_Toc169886035)

[6.4 Database Schema 4](#_Toc169886036)

[**7. Non-Deliverables** 5](#_Toc169886037)

[**8. Constraints** 5](#_Toc169886038)

[8.1 Constraints Imposed by Semester Requirements 5](#_Toc169886039)

[8.2 Self-Imposed Constraints 5](#_Toc169886040)

[**9. Phasing Based on Sprints** 6](#_Toc169886041)

**1. Introduction**

In modern urban environments, effective management of parking spaces is a common challenge. Sioux aims to solve this problem by implementing a parking management system using modern technology. This project involves the development of an application that uses cameras for vehicle detection & license plate recognition, and a comprehensive front-end for managing client appointments and notifications.

**2. Project Goal**

The goal of the Sioux Parking Management System project is to create a user-friendly application that:

* Monitors parking space availability and detects vehicle license plates utilizing cameras.
* Allows secretaries to manage client details and appointments through an easy-to-use interface.
* Utilizes Google Calendar for scheduling and notifications.
* Sends email notifications for appointment confirmations and parking lot status updates.

**3. Client & Team**

* Client: Sioux
* Development Team: Stanislav, Desislav, Mihail, Atanas
* Assessors: Oliveira De Arruda Camara, Amália M.A.

**4. Strategy**

The project will adopt an Agile development approach, specifically Scrum, due to its flexibility, emphasis on customer feedback, and ability to adapt to changing requirements. This iterative methodology will facilitate regular evaluation of progress and ensure alignment with user needs and business goals.

**5. Research questions and methodology**

5.1 How can technology enhance the management of parking spaces?

* Approach: Evaluate existing parking management systems and conduct user interviews.
* Methodology: Iterative design, user testing, and feedback loops.

5.2 What are the best practices for secure and efficient data handling in parking management systems?

* Approach: Research industry standards.
* Methodology: Desk research and Expert Interview.

**As the project progresses, additional research questions may emerge, necessitating updates to the project deliverables and approach.**

**6. Deliverables**

6.1 Documentation

* Platform Design Documents
* Project Plan

6.2 Front-End Application

* A fully functional React-based web application featuring user-friendly navigation, interactive elements, and a visually appealing interface.

6.3 Back-End Infrastructure

* A robust Java Spring Boot RESTful API that handles all server-side logic, database interactions, and client-server communications securely and efficiently.
* Implementation of business logic.

6.4 Database Schema

* A well-structured MySQL database schema, optimized for performance and scalability, including all necessary tables, relationships, and indices.
* Integration of JPA/Hibernate for object-relational mapping to facilitate smooth and efficient data operations.

**7. Non-Deliverables**

What will not be delivered:

* Integration with third-party job portals.
* Maintenance of the application after delivery.
* User manual for the app.

**8. Constraints**

8.1 Constraints Imposed by Semester Requirements

**These constraints are defined by the academic framework, deadlines, and deliverables set by the educational institution for the semester.**

* Time: Fixed deadlines for project milestones and the final delivery date can limit the scope of work, necessitating prioritization of core features over nice-to-have additions.
* Technical: The choice of technology stack (React, Java Spring Boot, MySQL, etc.) may limit certain functionalities or integrations, especially if relying on legacy systems or specific third-party APIs.
* Resource: The project must be completed with the available resources within the academic setting, including access to software, hardware, and any institutional support services.

8.2 Self-Imposed Constraints

**These constraints are the additional limitations or challenges we've identified based on our project vision, personal goals, or perceived project needs.**

* Operational: The operational environment, including hosting servers, development and production environments, and deployment pipelines, can impose limitations on scalability and performance.
* Integration: Challenges in integrating with external systems or other third-party services due to API limitations or technical incompatibilities.
* Scalability: Potential limitations in the platform's ability to scale efficiently with increased user load or data volume, influenced by the initial architectural and infrastructure decisions.

**9. Phasing Based on Sprints**

Introduction to Phasing

In alignment with the Agile development approach, the Parking Management System project will be executed in phases, divided into multiple sprints. Each sprint is planned to last two weeks and aims to deliver specific, incremental functionalities of the platform. This phasing ensures continuous progress, frequent assessments, and the ability to incorporate feedback rapidly.