***Software Engineering***

***Software Requirements Specification***

***(SRS) Document***

**[Project Name]**

**[9/26/2023]**

**[Version 1]**

**By: [Christopher Hochrein, Philip Sijerkovic, Amin Abdelrahim]**

**[Honor Code]**

**Table of Contents**

[1. Introduction 3](#_Toc126197756)

[1.1. Purpose 3](#_Toc126197757)

[1.2. Document Conventions 3](#_Toc126197758)

[1.3. Definitions, Acronyms, and Abbreviations 3](#_Toc126197759)

[1.4. Intended Audience 4](#_Toc126197760)

[1.5. Project Scope 4](#_Toc126197761)

[1.6. Technology Challenges 4](#_Toc126197762)

[1.7. References 4](#_Toc126197763)

[2. General Description 4](#_Toc126197764)

[2.1. Product Perspective 4](#_Toc126197765)

[2.2. Product Features 4](#_Toc126197766)

[2.3. User Class and Characteristics 5](#_Toc126197767)

[2.4. Operating Environment 5](#_Toc126197768)

[2.5. Constraints 5](#_Toc126197769)

[2.6. Assumptions and Dependencies 5](#_Toc126197770)

[3. Functional Requirements 5](#_Toc126197771)

[3.1. Primary 5](#_Toc126197772)

[3.2. Secondary 5](#_Toc126197773)

[4. Technical Requirements 6](#_Toc126197774)

[4.1. Operating System and Compatibility 6](#_Toc126197779)

[4.2. Interface Requirements 6](#_Toc126197780)

[4.2.1. User Interfaces 6](#_Toc126197781)

[4.2.2. Hardware Interfaces 6](#_Toc126197782)

[4.2.3. Communications Interfaces 6](#_Toc126197783)

[4.2.4. Software Interfaces 6](#_Toc126197784)

[5. Non-Functional Requirements 6](#_Toc126197785)

[5.1. Performance Requirements 6](#_Toc126197786)

[5.2. Safety Requirements 7](#_Toc126197787)

[5.3. Security Requirements 7](#_Toc126197788)

[5.4. Software Quality Attributes 7](#_Toc126197789)

[5.4.1. Availability 7](#_Toc126197790)

[5.4.2. Correctness 7](#_Toc126197791)

[5.4.3. Maintainability 7](#_Toc126197792)

[5.4.4. Reusability 7](#_Toc126197793)

[5.4.5. Portability 7](#_Toc126197794)

[5.5. Process Requirements 7](#_Toc126197795)

[5.5.1. Development Process Used 7](#_Toc126197796)

[5.5.2. Time Constraints 7](#_Toc126197797)

[5.5.3. Cost and Delivery Date 7](#_Toc126197798)

[5.6. Other Requirements 7](#_Toc126197799)

[5.7. Use-Case Model Diagram 8](#_Toc126197800)

[5.8. Use-Case Model Descriptions 8](#_Toc126197801)

[5.8.1. Actor: Editor (Responsible Team Member) 8](#_Toc126197802)

[5.8.2. Actor: Writer (Responsible Team Member) 8](#_Toc126197803)

[5.8.3. Actor: Photographer (Chris) 8](#_Toc126197804)

[5.9. Use-Case Model Scenarios 8](#_Toc126197805)

[5.9.1. Actor: Editor (Responsible Team Member) 8](#_Toc126197806)

[5.9.2. Actor: Writer (Responsible Team Member) 9](#_Toc126197807)

[5.9.3. Actor: Photographer (Chris) 9](#_Toc126197808)

# Introduction

## Purpose

[The goal of your project and the objectives it wishes to accomplish]

The goal of APPLICATION NAME is to make it easy for a newspaper team to complete projects by enabling effective communication between project members and allowing management of project materials.

## Document Conventions

[Full description of the main objectives of this document in the context of your project.

Here’s how you should begin this section:

The purpose of this Software Requirements Document (SRD) is to

“In it, we will . . ., . . ., and . . ..”]

The purpose of this Software Requirements Document (SRD) is to describe the client-view and developer-view requirements for the Automated Police Ticketing System (APTS). Client-oriented requirements describe the system from the client’s perspective. These requirements include a description of the different types of users served by the system. Developer-oriented requirements describe the system from a software developer’s perspective. These requirements include a detailed description of functional, data, performance, and other important requirements.

## Definitions, Acronyms, and Abbreviations

[Include any specialized terminology dictated by the application area or the product area.

For example:]

|  |  |
| --- | --- |
| Java | A programming language originally developed by James Gosling at Sun Microsystems. We will be using this language to build the Restaurant Manager. |
| MySQL | Open-source relational database management system. |
| .HTML | Hypertext Markup Language. This is the code that will be used to structure and design the web application and its content. |
| SpringBoot | An open-source Java-based framework used to create a micro Service. This will be used to create and run our application. |
| MVC | Model-View-Controller. This is the architectural pattern that will be used to implement our system. |
| Spring Web | Will be used to build our web application by using Spring MVC. This is one of the dependencies of our system. |
| Thymeleaf | A modern server-side Java template engine for our web environment. This is one of the dependencies of our system. |
| NetBeans | An integrated development environment (IDE) for Java. This is where our system will be created. |
| API | Application Programming Interface. This will be used to implement a function within the software where the current date and time is displayed on the homepage. |

## Intended Audience

[Describe which part of the SRS document is intended for which reader. Include a list of all stakeholders of the project, developers, project managers, and users for better clarity.]

## Project Scope

[Specify how the software goals align with the overall business goals and outline the benefits of the project to business.]

The goal of the software is to provide an easy-to-use interface for all customers, employees, and managers of a restaurant, as well as provide customers with flexibility to meet their needs. This aligns with the overall business goals of a restaurant as a restaurant requires fast and efficient service in order to fulfill the needs of its customers.

The benefits of the project to business include:

* Relieving stress and pressure from employees and managers as customers are given the opportunity to request services when needed.
* Increasing pleasure to customers as they are given more power when they want to order rather than having to wait for an employee to ask for their order.
* Reducing the amount of time that a customer needs to wait; therefore, increasing the amount of customers that are able to be served in the restaurant within a day.

The goal of this software it to provide an easy-to-use interface for writers, editors, and

## Technology Challenge

*Leave 1.6 blank for now.*

## References

*Leave 1.7 blank for now.*

# General Description

## Product Perspective

PRODUCT NAME was originally created out of the desire to make it easier to work on group projects together.

## Product Features

The application will allow employees to create different accounts depending on their role in the team. Writers are able to upload drafts for editors to view and edit. Once editors have finished editing a draft, they can then send it back to writers for revision, or they can mark the draft as completed. Writers and editors can also make requests for photos that will be uploaded to a bulletin that all photographers in the system can see and access. Photographers can select and respond to requests by uploading photos, which can then be accessed by the parties that made the requests.

## User Class and Characteristics

Our application is designed for use by a newspaper writing team. Users should know the basics of using a computer, but the system should be simple and intuitive. The system should allow them to fulfill tasks they were already doing in a more streamlined and accessible way.

## Operating Environment

The application will be designed to operate via the web on desktop operating systems.

## Constraints

*Leave 2.5 blank for now.*

## Assumptions and Dependencies

[A list of all assumptions that you have made regarding the software product and the environment along with any external dependencies which may affect the project]

The software will be dependent on Spring Web and Thymeleaf in order to create and execute the MVC architecture that will be developed within NetBeans. The application will also use the World Time API (http://worldtimeapi.org/) that will display the current date and time on the home dashboard for everyone to see.

# Functional Requirements

[Statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations.]

## Primary

[All the requirements within the system or sub-system in order to determine the output that the software is expected to give in relation to the given input. These consist of the design requirements, graphics requirements, operating system requirements and constraints if any.]

* FR0: The system will allow users to make accounts and log in.
* FR1: The system will allow writers to send drafts to editors.
* FR2: The system will allow editors to edit drafts and add notes for changes to be made. These can then be sent back to writers to produce the next draft.
* FR3: The system will allow writers and editors to post requests for photos to a bulletin viewable by photographers. These requests can contain a written description of the photo they are looking for.
* FR4: The system will allow writers and editors to access photos that are submitted in response to these requests and allow them to close requests.
* FR5: The system will allow photographers to view photo requests on the bulletin, choose requests to accept, and then upload a photo in response to a specific request.
* FR0: The system will allow the user to lookup of vehicle owner information based on license plate number. This information will contain owner’s permit number, assigned lot, and previous violations including tow history.
* FR1: The system will allow the user to enter a new vehicle into the vehicle violation database.
* FR2: The system will allow the user to issue a ticket. The ticket information will be issued in electronic and paper form.

## Secondary

[Some functions that are used to support the primary requirements.]

* Authorization scheme so that only relevant parties can access documents and photo requests.
* Account creation and password protection so that each employee can be recognized within the system.

# Technical Requirements



## Operating System and Compatibility

[The environments that will be needed to operate the system]

The application will be compatible with desktop operating systems.

## Interface Requirements

### User Interfaces

Each user will have a separate interface, different users will be able to send different files to each other to collaborate on their projects. There will be a login feature for a persons profile. On your profile you will be able to add your name, and a description about yourself.

### Hardware Interfaces

[All the hardware-software interactions with the list of supported devices on which the software is intended to run on, the network requirements along with the list of communication protocols to be used.]

The web app will run on any device that can access the internet and access/interact with web pages.

### Communications Interfaces

[Determination of all the communication standards to be utilized by the software as a part of the project]

It must be able to connect to the internet as well as the local database on phpMyAdmin.

The communication protocol, HTTP, must be able to connect to the World Time API and return the current date and time.

It must be able to connect to the internet, as well as the database used for logins [TBD]. Utilizes HTTP, must be able to connect to the [TBD] API and return a value.

### Software Interfaces

[The interaction of the software to be developed with other software components such as frontend and the backend framework to the used, the database management system and libraries describing the need and the purpose behind each of them.]

We will use React and Spring Boot ThymeLeaf to help build the frontend, as well as JPA for the backend database functionality. We will also use Spring Boot with Java to connect the frontend to the backend.

We will use HTML, CSS, and Javascript to build the frontend, as well as Spring Boot with Java to connect the frontend and backend. The backend will be done with java.

# Non-Functional Requirements

[Constraints on the services or functions offered by the system (e.g., timing constraints, constraints on the development process, standards, etc.). Often apply to the system as a whole rather than individual features or services.]

## Performance Requirements

[The performance requirements need to be specified for all the functional requirements.]

* NFR0(R):
* NFR0(R): The local copy of the vehicle violation database will consume less than 20 MB of memory
* NFR1(R): The system (including the local copy of the vehicle violation database) will consume less than 50MB of memory
* NFR2(R): The novice user will be able to create and print a ticket in less than 5 minutes.
* NFR3(R): The expert user will be able to create and print a ticket in less than 1 minute.

## Safety Requirements

[List out any safeguards that need to be incorporated as a measure against any possible harm the use of the software application may cause.]

## Security Requirements

[Privacy and data protection regulations that need to be adhered to while designing of the product.]

* NFR#(R): The system will only be accessible by authorized users.

## Software Quality Attributes

[Detailing on the additional qualities that need to be incorporated within the software like maintainability, adaptability, flexibility, usability, reliability, portability etc.]

### Availability

[Details]

### Correctness

[Details]

### Maintainability

[Details]

### Reusability

[Details]

### Portability

[Details]

## Process Requirements

### Development Process Used

[Software Process Model]

[Waterfall Model]

### Time Constraints

### Cost and Delivery Date

## Other Requirements

TBD

## Use-Case Model Diagram

A diagram of a diagram

Description automatically generated

## Use-Case Model Descriptions

### Actor: Editor (Responsible Team Member)

* **Use-Case Name:** Login
* **Use-Case Name**: Access drafts
* **Use-Case Name**: Send drafts to writers
* **Use-Case Name**: Create photo requests
* **Use-Case Name:** View photo requests and access photos

### Actor: Writer (Responsible Team Member)

* **Use-Case Name:** Login
* **Use-Case Name**: Access drafts
* **Use-Case Name**: Send drafts to editors
* **Use-Case Name**: Create photo requests
* **Use-Case Name**: View photo requests and access photos
* **Use-Case Name**: [Brief Use-Case Description]

### Actor: Photographer (Chris)

* **Use-Case Name**: Login
* **Use-Case Name**: View photo requests and access photos
* **Use-Case Name**: Upload Photos

## Use-Case Model Scenarios

### Actor: Editor (Responsible Team Member)

* **Use-Case Name**: Login
  + **Initial Assumption**:
  + **Normal**:
  + **What Can Go Wrong**
  + **Other Activities**:
  + **System State on Completion**:
* **Use-Case Name**: Access drafts
  + **Initial Assumption**:
  + **Normal**:
  + **What Can Go Wrong**
  + **Other Activities**:
  + **System State on Completion**:
* **Use-Case Name**: Send drafts to writers
  + **Initial Assumption**:
  + **Normal**:
  + **What Can Go Wrong**:
  + **Other Activities**:
  + **System State on Completion**:
* **Use-Case Name:** Create photo requests
  + **Initial Assumption:**
  + **Normal:**
  + **What Can Go Wrong**
  + **Other Activities:**
  + **System State on Completion:**
* **Use-Case Name**: View photo requests and access photos
  + **Initial Assumption**:
  + **Normal**:
  + **What Can Go Wrong**
  + **Other Activities**:
  + **System State on Completion**:

### Actor: Writer (Responsible Team Member)

* **Use-Case Name**: Login
  + **Initial Assumption**:
  + **Normal**:
  + **What Can Go Wrong**
  + **Other Activities**:
  + **System State on Completion**:
* **Use-Case Name**: Access drafts
  + **Initial Assumption**:
  + **Normal**:
  + **What Can Go Wrong**
  + **Other Activities**:
  + **System State on Completion**:
* **Use-Case Name**: Send drafts to editors
  + **Initial Assumption**:
  + **Normal**:
  + **What Can Go Wrong**:
  + **Other Activities**:
  + **System State on Completion**:
* **Use-Case Name:** Create photo requests
  + **Initial Assumption:**
  + **Normal:**
  + **What Can Go Wrong**
  + **Other Activities:**
  + **System State on Completion:**
* **Use-Case Name**: View photo requests and access photos
  + **Initial Assumption**:
  + **Normal**:
  + **What Can Go Wrong**
  + **Other Activities**:
  + **System State on Completion**:

### Actor: Photographer (Chris)

* **Use-Case Name**: Login
  + **Initial Assumption**:
  + **Normal**:
  + **What Can Go Wrong**:
  + **Other Activities**:
  + **System State on Completion**:
* **Use-Case Name**: View photo requests and access photos
  + **Initial Assumption**:
  + **Normal**:
  + **What Can Go Wrong**:
  + **Other Activities**:
  + **System State on Completion**:
* **Use-Case Name:** Upload photos
  + **Initial Assumption:**
  + **Normal:**
  + **What Can Go Wrong:**
  + **Other Activities:**
  + **System State on Completion:**