**Software Engineering**

**Software Requirements Specification**

**(SRS) Document**

**< Project Name>**

**<Date>**

**<Version>**

**<By: Team Member names>**

**<Honor Code>**

|  |
| --- |
| **Table of Contents** |

[1. Introduction 2](#_Toc113453891)

[2. General Description 3](#_Toc113453892)

[3. Functional Requirements 3](#_Toc113453893)

[4. Technical Requirements 4](#_Toc113453894)

[4.1 Operating System & Compatibility 4](#_Toc113453895)

[4.2 Interface requirements 4](#_Toc113453896)

[4.2.1 User Interfaces 4](#_Toc113453897)

[4.2.2 Hardware Interfaces 4](#_Toc113453898)

[4.2.3 Communications Interfaces 4](#_Toc113453899)

[4.2.4 Software Interfaces 4](#_Toc113453900)

[5. Non-Functional Requirements 4](#_Toc113453901)

1. Introduction
   1. **Purpose:** The goal of your project and the objectives it aims to accomplish
   2. **Document conventions:** The typographical methodologies followed within the document.

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 . . ..”

For example:

“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.”

* 1. **Definitions, Acronyms, and Abbreviations**

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

For example:

|  |  |
| --- | --- |
| **Term** | **Definition. Acronym, Abbreviation** |
| .Net | A set of software technologies from Microsoft for connection information, people, and computer systems. |
| ATPS | An abbreviation for Automated Police Ticketing System. This is the name of the system that is being built. |
| C# | A programming language created by Microsoft. We will be using this language to build the ATPS. |
| DB | An abbreviation for Database. |
| MS | An abbreviation for Microsoft. Microsoft is a large software company which produces the software that will be used to implement ATPS. |
| Microsoft Access | A database software created by Microsoft. The campus police vehicle violation database was created using Microsoft Access. |

* 1. **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.
  2. **Project Scope:** Specify how the software goals align with the overall business goals and outline the benefits of the project to business.
  3. **Technology Challenges:**
  4. **References:** Mention books, articles, web sites, worksheets, people who are sources of information about the application domain, etc. Use proper and complete reference notation. Give links to documents as appropriate. You should use the APA Documentation model (Alred, 2003, p. 144).

Alred, F., Brusaw, C., and Oliu, W. (2003). *Handbook of Technical Writing* (7th ed.)*.* Boston: Bedford/St. Martin’s.

## General Description

* 1. **Product perspective:** Describe the context and origin of the product.
  2. **Product features:** A high level summary of the functions the software would perform and the features to be included.
  3. **User class and characteristics:** A categorization and profiling of the users the software is intended for and their classification into different user classes

For Example:

Our website application does not expect our users to have any prior knowledge of a computer, apart from using a web browser, or knowledge of astronomy. Our website application has removed the need for them to have astronomy, math, or science knowledge and allows the user to focus on exploring the night sky.

* 1. **Operating environment:** Specification of the environment the software is being designed to operate in.
  2. **Constraints:** Any limiting factors that would pose challenge to the development of the software. These include both design as well as implementation constraints.

For example:

Due to the use of a 3d engine, we had to limit the web browsers supported. To limit user error when entering the user’s address, we implemented a drop-down AJAX country, state, and city selection.

* 1. **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

## Functional Requirements

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.

* 1. **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.

For Example:

* 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.
* FR3: The system will automatically fill in data fields using vehicle owner information should a ticket need to be issued.
* FR4: The system will allow the user to update a ticket after the ticket has been issued.
* FR5: The system will allow the user to delete a ticket after the ticket has been issued.
* FR6: The system will keep the user’s ticket information and the server’s vehicle violation database synchronized to within 24 hours.
  1. **Secondary:** Some functions that are used to support the primary requirements.

## Technical Requirements

* 1. Operating System & Compatibility
  2. Interface requirements
     1. User Interfaces

The logic behind the interactions between the users and the software. This includes the sample screen layout, buttons and functions that would appear on every screen, messages to be displayed on each screen and the style guides to be used.

* + 1. 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.

* + 1. Communications Interfaces

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

* + 1. 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.

## 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.

* 1. **Performance requirements**

For Example:

The performance requirements need to be specified for every functional requirement. The rationale behind it also needs to be elaborated upon.

* NFR2(R): The local copy of the vehicle violation database will consume less than 20 MB of memory
* NFR3(R): The system (including the local copy of the vehicle violation database) will consume less than 50MB of memory
* NFR5(R): The novice user will be able to create and print a ticket in less than 5 minutes.
* NFR6(R): The expert user will be able to create and print a ticket in less than 1 minute.
  1. **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.

* 1. **Security requirements**

Privacy and data protection regulations that need to be adhered to while designing of the product. For Example:

* NFR7(R): The system will only be usable by authorized users.
  1. **Software quality attributes**
     1. Availability
     2. Correctness
     3. Maintainability
     4. Reusability
     5. Portability

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

* 1. **Process Requirements**
     1. Development Process Used
     2. Time Constraints
     3. Cost and Delivery Date
  2. **Other requirements**

These may include the legal requirements, resource utilizations, future updates etc.

* NFR4(R): The system will conform to FERPA guidelines to maintain student privacy.

All SRS/SRD should be:

* **Correct:** A method of analysis that ensures that the software meets the requirements identified.
* **Unambiguous:** There is only one interpretation of what the software will be used for and it is communicated in a common language.
* **Complete:** There is a representation for all requirements for functionality, performance, design constraints, attributes, or external interfaces.
* **Consistent:** Must be in agreement with other documentation, including a systems requirements specification and other documents.
* **Ranked for Importance and/or Stability:** Since all requirements are not of equal weight, you should employ a method to appropriately rank requirements.
* **Verifiable:** Use measurable elements and defined terminology to avoid ambiguity.
* **Modifiable:** A well-defined organizational structure of the SRS document that avoids redundancies can allow easy adaptation.
* **Traceable:** Ability to trace back to the origin of development and move forward to the documents produced from the SRS.
* **Legible and Professionally Presented**: Must use a consistent font and style. Must have proper formatting of tables and charts. Must be grammatically correct. Use active tense and concise sentences.