# **SYSTEM REQUIREMENTS SPECIFICATION**

## **DEFINITION OF SRS**

System Requirements Specification is a document that describes what the system will do and how it will be expected to perform. Furthermore, SRS is the document that that identifies the functionality that is needed by a system in order to satisfy the customer’s requirements. This show that SRS is the most effective way that meets user needs so as to accomplish stakeholder’s goals.

Also, System requirements Specification summarized into four Ds:

## **DEFINE THE PURPOSE WITH AN OUTLINE**

This system will be created from scratch and not by using any Templates. The following are the outline might look like:

### **PROBLEM IDENTIFICATION**

Our problem is concerned about Collection of Machinga’s Revenue. In our societies, there were problems about collecting Machinga’s Revenue due to lack of information in case of Validation, Payment records, Date issued on as well as Date expired on.

### **SCOPE**

Here, the purpose of creating this system is to record Machinga details concern with Revenue. Also, this system will generate Machinga’s details including on Address, Date issued on and expired on also generate Machinga’s report based on payment history.

Also, the scope area we are going to dealing with is City Center (DAR-ES-SALAAM). The deadline of design this system will be around 18th Feb, 2022.

### **SYSTEM REQUEST**

According to the problem introduced, we proposal to design system called Machinga Collection Revenue System that dealing with records Machinga details as well as generates Machinga payment history.

### **SYSTEM PROPOSAL**

System proposal is serves as a summary of the system analyst’s work in the business.

## **DEFINE THE PRODUCT’S PURPOSE**

Throughout this System Requirements Specification, we will meet the customers need. Meaning that, we are going to outline all user needs, functionality and non-functionality. When defining the product’s purpose, we include:

### **PRODUCT SCOPE**

The benefits of designing Machinga Collection Revenue System is to solve the problem concerned about collection of Machinga Revenues. Furthermore, this system will also help Admin to generate reports of all payments as well as Machinga details.

The goals of this system is to make easier on determine who is already registered, expired date and Machinga address

The goals of designing Machinga Collection Revenue System are to ensure that Admin have access to determine registered members and also to determine who is already paid or not

### **INTENDED AUDIENCE AND INTENDED USE**

Person who have access to use this SRS are the developers and testers because this SRS should have user needs, functionalities and non-functionalities that developers should follow and also guide in implementation in order to fulfil customer’s needs.

### **DEFINE RISKS**

## **DESCRIBE WHAT YOU WILL BUILD**

Here, we are going to give description of what we are going to build. Machinga Collection Revenue System is the new product in town, that help TRA to dealing with Machinga in details. Description based on:

### **USER NEEDS**

The person who going to use this product direct is ADMIN and not Machinga because we are going to manage Machinga details. The critical needed by user is to collect revenues from informal business such as Machinga.

### **ASSUMPTIONS AND DEPENDENCIES**

We assuming that ADMIN can accesses Machinga details as well as can generate reports that’s based on payments history will be true.

## **DETAILS YOUR SPECIFIC REQUIREMENTS**

### **FUNCTIONAL REQUIREMENTS**

These include:

* Allow only registered users to access the application
* Validate data entered in the application (only client-side validation is enough)
* Allow a registered user to view and update their profiles
* Allow a registered user to initiate a “request of a service” by completing an online form and uploading relevant supporting documents
* Provide implementation status of requested services
* Keep log of every activity performed in the application

### **EXTERNAL INTERFACE REQUIREMENTS**

There are several types of interfaces you may have requirements for, including:

* User
* Hardware
* Software
* Communication

hk

### **NON-FUNCTIONAL REQUIREMENTS**

These can be just as important as functional ones. These include:

* Performance
* Safety
* Security
* Quality

## **DELIVER FOR APPROVAL**