**WAJANE AUTOMATED RECORDING SYSTEM**

**System Requirement Specification (SRS)**

1. **Introduction**

The section introduces the system requirement specification (SRS) for the Wajane Automated Recording System (WARS).

* 1. **Purpose**

The purpose of this SRS is to come up with a web based application to ease records management for Wajane COSALO.

* 1. **Intended Audience And Reading Suggestions**

The document will come in handy for project manager, users, testers and any other party that might be interested in further developing or advancing the functionalities or capabilities of the system.

The intended audience or stakeholders are as listed below:

* **Wajane Automated Recording System (WARS) members:**
  + Chairperson
  + Secretary who will add or remove members using the new system
  + Accountant who will undertake accounting responsibility using WARS
* **The development team:**
* Programmer(s) who must ensure that the software components that they develop implements the specification in this SRS.
* Testers who must check that the system in valid and whose test must validate the requirements.

This specification is organized into the following sections:

* *Introduction*, which introduces the specification for the WARS to its readers.
* *Overall description*, which provides a brief, high level description of the WARS including its definition, business goals, business objectives, context, and capabilities.
* *Functional Requirements*, which specifies the functional system requirements in terms of a use case model consisting of each external’s use cases and use case paths.
* *Data Requirements*, which specifies the system data requirements in terms of required data components.
* *Quality Requirements*, which specifies the required system quality factors.
* *Constraints*, which documents required architecture, design, and implementation constraints on the WARS.
  1. **Project Scope**

Wajane Automated Recording System (WARS) is an automated, networked, cloud based web application that is intended to solve the perennial problems witnessed in record handling. The product is intended to serve a local COmmunity SAvings and LOans organization. Above all, we hope to solve the problem of records anomalies.

The objectives of this specification of the (WARS) are to:

* Provide a system overview of the (WARS) including definition, goals, objectives, context, and major capabilities.
* To formally specify its associated:
* Functional requirements.
* Data requirements.
* Quality requirements.
* Constraints.
  1. **References**

This SRS references from the following document(s):

* **OPEN Process Framework (OPF) Conventions:**
* Use Case Modeling Guidelines, which documents the guidelines used to develop the use case model specifying the functional requirements in this specification.
* System Requirements Specification Content and Format Standard, which specifies the content and format of this specification.
* System Requirements Specification Inspection Checklist, which is used during the inspection of this specification.
* System Requirements Specification Template, which provides the skeleton of this specification.
* <https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database>
* SYSTEM REQUIREMENT SPECIFICATION DOCUMENT template by KCA University.

1. **Overall Description**

This section provides a high level description of the Wajane Automated Recording System (WARS) system including its product perspective, product features, user classes and characteristics, operating environment, design and implementation constraint, user documentation and assumptions and dependencies.

* 1. **Product Perspective**

WARS will be an automated web based cloud application that will be highly efficient for records management for the Wajane COSALO.

* 1. **Product Features**

WAR system will take advantage of ubiquity of the ubiquity of the internet to implement the system in such a way that the application will be:

1. a web based application
2. automated to reflect changes whenever they are made(so long as one has access to the internet)
3. a cloud application to enable virtualization
4. will be android based( owing to android dominance)
5. having the online version (webpage) to carter for non-android users.
   1. **User Classes And Characteristics**

The user of the system with the administrator permission should be able to able to generate a unique key or password with which he/ she will be able to admit other members. He /she will also be in a position to remove member(s) when the latter quits. Each and every member will be in a position to see any changes made and when they were made (regarding admission of new members, dismissal of members, changes made to individual member records etc.) once they are connected to the internet. The application will support two types of user privileges: member and administrator. Members will enjoy member functions as the administrator will have both member functions and administrator functions.

The member should be able to perform the following functionalities:

* Save money
* Borrow loans
* View changes made
* Send messages
* Receive messages
* Request dismissal
* Recommend new member(s)

The administrator should be able to perform the following functions:

* **Member functions**
* Save money
* Borrow loans
* View changes made
* Sand message
* **Administrator functions**
* Effect transactions
* Admit new members
* Dismiss member(s)
* Make changes to the records

Effecting transactions by administrator(s) does not involve real monetary transactions but just changes in the records. The changes made will be kept in a log.

* 1. **Operating Environment**

Listed below is the operating environment for the Wajane Automated Reflection System:

* Operating system: Android OS
* Database: SQLite
* Cloud
* Platform: Java/PHP
* Client/server system
  1. **Design and Implementation Constraint**

1. Implement the database at least using a local database.
2. At the initial stages, usage of emulators.
3. Use of a local machines with xampp as an alternative to client server system.
   1. **User documentation**

Will be availed once the system is up and running.

* 1. **Assumptions and Dependencies**

One assumption is that all members have android devices or any other device that can access the internet via web browser. Another assumption is that all the members have the ability to operate either of these devices that can be used for the application. Lack of the devices and knowledge to operate the devices can incapacitate one to access the services availed by the application.

The members will also have to part with some amounts to cater for the hosting fee by the cloud vendors.

1. **System Features**

* **Description and priority.**

The WARS system will ensure data integrity, privacy and confidentiality, together with elimination of any anomalies related to record keeping. This system will come in handy in ensuring proper management of records by the organization.

* **Stimulus/ Response sequence**
* Click a button for respective operation
* Prompt for input(only if there is any needed)
* Display information dependent on button pressed
* Commit and exit
  1. **System Feature 1**
     1. **Functional requirements**

The other application features include:

**Cloud Services**

For the cloud service, **Software As A Service(SAAS)**  will be adopted so that the web application is hosted on the cloud so that changes made by respective members and their logs are stored in the cloud. Such changes will reflect on every member device once they are connected to the internet. The model is as shown below:

Client side server side

(members) (cloud app)

Web pages pages

server

cloud

Android App

Storage

**Client/Server System**

The term client/server refers primarily to an architecture or logical division of responsibilities, the client is the application (also known as the front-end), and the server is the DBMS (also known as the back-end). It is also demonstrated above.

A client/server system is a distributed system in which,

* Some sites are client sites and others are server sites.
* All the data resides at the server sites.
* All applications execute at the client sites.

1. **Other nonfunctional requirements**
   1. **Performance requirements**

The implementation stages involved in implementation of the WARS system are as listed below:

1. **E-R Diagram**

The E-R diagram constitutes a technique for representing the logical structure of a database in a pictorial manner. The resulting analysis is then used to organize data as a relation, normalizing relation and finally obtaining a relation database.

* **ENTITIES:**Which specify distinct real-world items in an application.
* **PROPERTIES/ATTRIBUTES:** Which specify properties of an entity and relationships.
* **RELATIONSHIPS:** Which connect entities and represent meaningful dependencies between them.

Member

Borrows

Saves

Money

1. **Other requirements**
   1. **Safety requirements**

For cloud applications, data safety is guaranteed but should there be any failure , the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure.

* 1. **Security Requirements**

The cloud vendor has to be chosen with care. The administrators have to secure their phones by means such as use of passwords, not exposing their phones to other people carelessly etc.

* 1. **Software Quality Attributes**
* **Integrity:** the changes made by members should reflect as expected without any alteration.
* **Usability:**  the software should be easy to learn and use.
* **Correctness:** the members should be able to use the system without any flaws.