INDUS UNIVERSITY
LEARN WELL TO LIVE WELL

FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY

(F.C.I.T)

RM
RENTMENT

Software Requirement Specification

Software Requirement Specifications

Project Code	Rentment Application
Supervisor	*Sir Zulfiqar*
Co Supervisor	*Sir Mushtahir*
Project Team	*AdeelMuhammad,Irfanuallah,Farah*
Submission Date	

Table of Contents

1. Introduction

This document is created for the **Rentment Application** where each aspect in related to the software are covered either it is the stakeholder or the Functional requirement, non -functional requirement and the operating environment The system constraints which is very important and also the assumption and dependencies can be describe in this document

1.1. Purpose of Document

This document purpose is related to the Rentment Application which refers to the how the Application work will have gone. Which model and the external entity and internal entity are required. How to design the application architecture. What is the functional and the non-functional requirement of system.

The documentation of rental service software serves several important purposes:

- **User Guidance:** Documentation provides user guides and manuals to help users understand how to navigate and use the rental service software. This includes step-by-step instructions, screenshots, and explanations of various features and functionalities.
- **Easily get Solution:** In the event of issues or errors, documentation provides troubleshooting guides. This helps users identify and resolve common problems without requiring direct assistance from support teams.
- **Security Information:** Documentation often includes information about security measures and best practices to ensure that users understand how to keep their data secure when using the rental service software.
- **Updates and Changes:** Software is regularly updated to fix bugs, introduce new features, or enhance security. Documentation informs users about these updates, including any changes to existing features or the addition of new functionalities.
- **API Documentation:** If the rental service software provides an API (Application Programming Interface) for developers, documentation is essential to guide them on how to integrate and interact with the software programmatically.
- **Legal Information:** Documentation may include terms of service, end-user license agreements, and other legal information that users need to be aware of when using the rental service software.

In conclusion, documentation plays a crucial role in ensuring that users can effectively use, troubleshoot, and maintain the rental service software. It enhances the overall user experience and helps both end-users and developers get the most out of the software.

1.2. *Intended Audience*

- **Vendors:** These are individuals or businesses looking to rent products or services through the application. They are the primary users and may include a wide range of people with varying levels of technological proficiency.
- **Service Providers (Clients):** These are individuals or business analyst have managed application to create listings, manage rental requests, and communicate with renters.
- **New Users:** People who are new to the platform and need guidance and information on how to use it effectively.
- **Third-Party Partners:** If the application collaborates with third-party service providers (e.g., for NIC verification), these partners may also be part of the audience.

1.3 *Abbreviations*

In the context of a rental application, there are several common abbreviations and acronyms that may be used in documentation, communication, or within the application itself. Here are some relevant abbreviations:

- **NIC:** National Identity Card
- **ID:** Identification
- **IDV:** Identity Verification
- **KYC:** Know Your Customer
- **KYV:** Know Your Vendor
- **UX:** User Experience
- **UI:** User Interface
- **FAQ:** Frequently Asked Questions
- **T&C:** Terms and Conditions
- **PP:** Privacy Policy
- **TOS:** Terms of Service
- **VRM:** Vendor Relationship Management
- **ML:** Machine Learning
- **AI:** Artificial Intelligence
- **IOT:** Internet of Things
- **NLP:** Natural Language Processing
- **OCR:** Optical Character Recognition
- **RPA:** Robotic Process Automation
- **HTTPS:** Hypertext Transfer Protocol Secure
- **API:** Application Programming Interface

1.4 Document Convention

- **Document Titles and Headings:**

Use clear and descriptive titles and headings for different sections of your documentation, such as "User Registration," "Product Listings," "NIC Certification," "Return Policy," and "Product Inspection." • **Consistent Formatting:**

Establish a consistent formatting style for your documentation, including fonts, font sizes, headings, and text formatting. This ensures a professional and organized appearance.

- **Abbreviations and Acronyms:**

Define and list any abbreviations or acronyms used in your documentation at the beginning or in a glossary section for easy reference. For example, "NIC" for "National Identity Card" should be defined.

- **Use of Sections and Subsections:**

Divide your documentation into logical sections and subsections. For instance, under "Product Inspection," you can have subsections like "Pre-Verification" and "Post Verification."

- **Numbered Lists and Bullet Points:**

Use numbered lists and bullet points for step-by-step instructions, guidelines, or policy points. This makes it easy to follow and understand.

- **Tables and Charts:**

When presenting data or complex information, use tables and charts for clarity and visual appeal.

- **Links and Cross-References:**

Include hyperlinks or cross-references to related sections within your documentation. This helps users navigate and find information efficiently.

- **Consistent Terminology:**

Use consistent and standardized terminology throughout your documentation. For instance, use "renter" instead of switching between "user" and "customer."

- **Legalese and Contracts:**

If your documentation includes legal agreements or contracts, ensure they are drafted by legal professionals and adhere to the relevant legal conventions in your jurisdiction.

- **Date and Version Control:**

Include the document's creation date and version number in the header or footer. Update the version number when you make significant revisions.

- **Images and Screenshots:**

Use clear and relevant images and screenshots to illustrate key points, especially when explaining how to use the application.

- **Consistent References:**

Use consistent references to specific features or elements of your application, such as buttons, menus, or user interface elements.

- **User Feedback:**

If possible, include sections for user feedback, questions, or comments to encourage users to provide input or seek clarification.

- **Review and Update:**

Regularly review and update your documentation to keep it accurate and relevant to the application's current state.

- **Table of Contents and Index:**

Include a table of contents and an index to help users quickly locate specific information.

- **Accessibility Considerations:**

Ensure that your documentation is accessible to all users, including those with disabilities. Provide alternatives for non-text content, such as image descriptions.

- **User-Friendly Language:**

Write your documentation in clear, user-friendly language, avoiding jargon and technical terms whenever possible. Use plain language to ensure comprehension.

2. Overall System Description

2.1. Project Background

The purpose of the Rentment Application is to provide a platform for users to easily rent products, connecting renters and customers in a streamlined manner.

2.2. Project Scope

The Rentment Application will encompass features for both renters and customers, facilitating product discovery, booking, and management of rental transactions.

2.3. Out of Scope

The scope of a Rental Application is defined by the features and functionalities it includes. However, it's also crucial to identify what is explicitly excluded from the scope to set clear expectations. Here are some aspects that might be not in the scope of a Rental Application:

- If the primary focus is on rentals, the application might exclude features related to buying or selling properties.
- Features related to property auctions or bidding, which are more common in sales-oriented platforms, might be excluded.
- If the application is not intended for real estate investors, features related to investment analytics or tools may be excluded.
- Depending on the target audience, the application might focus on short-term rentals, excluding features specific to long-term leasing.
- Food,Make-up,Drinks like things are not included in the Rentment Application

2.4. Project Objectives

The primary objective of a Rentment Application is to provide a user-friendly platform that facilitates the rental process for both property owners (landlords or renters) and individuals seeking to rent properties or items. Here are some key objectives that a Rentment Application project might aim to achieve:

- Provide the different rent products category
- Provide the product is available for the Rent on current time or not.
- Developed the agreement (Damage Policy)between the Vendors (product owner) and the Customers (renter)
- Develop an intuitive and user-friendly interface that simplifies the search, booking, and communication processes for both renters (customer) and property owners (vendors)
- Provide pre and post product verification

- Provide Secure System only the Authorize user can Access to product
- Enable secure and seamless payment transactions, supporting various payment methods for rental transactions.
- Implement a notification system to keep users informed about booking requests, confirmations, messages, and other relevant updates.
- Implement a system that allows users to leave and view reviews and ratings for properties and users, contributing to a trustworthy community.

2.5. Stakeholders

Some potential stakeholders for a Rentment Applications are Elucidate below

- **Users:**
  The individual who used the application to take their choice of product for the rent easily and in secure manner. In other hand the Application owner who managed the all the aspect that required to satisfy the customer.
- **Vendors:**
  Those individual who have need to rent their product on the Application. Where they upload the product and also provide the availability of product and related aspect.
- **Development Team:**
  The development team can be illustrated in the form of the table

work	Development Team		
1. SRS	Farah		
2. SDS	Farah		
3. Front End	Adeel	Irfanullah	
4. Back End	Adeel	Irfanullah	
5. Testing	Farah	Adeel	Irfanullah

- **External Service Providers:**
  If third-party services are integrated (e.g., payment gateways), those providers become stakeholders.
- **Technology Partners:**
  If the Rentment Application relies on specific technologies or platforms, those technology providers become stakeholders.
- **Supervisor and Co Supervisor (Experts)**

  Who give the guidance and guide throughout the project Development

2.6. Operating Environment

The hardware and software infrastructure on which the application will run. It includes the required platforms which the Application run, operating systems, browsers, and other technical requirements that are illustrated below

- **Required Platforms:**
  Web: web browsers that the application supports are (e.g., Google Chrome, Mozilla Firefox, Safari, Microsoft Edge).
  Mobile: our application is designed for specific mobile platforms are Android and versions
- **Operating System**
  The operating system which the our Application is work is the Android for the Mobile phone and the for the websites Access the our Application work on the windows Operating System
- **Development Tool**
  The IDE (Integerated Development Enviroment) include the Visual studio and the where different Libraries and the Framework are used to developed the Rentment Application
- **Third Party Services**
  List any third-party services or APIs integrated into the application and their requirements.
- **Front End Development**
  We used the Framework for the frontend is the React and the React native which is based on the java script programming language and also used some other important libraries and UI/UX design like the Ant Design
- **Database**
  Our Application Database operation are control and store in the firebase database
- **Backend Development**
  We used the Backend Framework for the server side operation are the nodes which is based on the java script and also many other technologies for Backend Functionalities

2.7. System Constraints

The system constraint refers to that required when using the Application either it is the specific platform which the Application run or the system which is required to run the application. If the system Constraint are not meet they affect the many factor of the rentment application various aspects of the system, including performance, security, scalability, and compatibility. Here are some common system constraints for a rental application:

- **Required Hardware:** The rentment application must perform their operation within the constraints of the hardware it runs on, including processing power, memory, storage capacity, and network bandwidth.

- **Platform compatibility:** The application may be constrained by compatibility with specific operating systems. It must be designed and tested to run on the targeted operating systems and versions.

- **Required Network Bandwidth:** The application may need to operate under specific network conditions, such as bandwidth limitations, latency constraints, and the ability to handle intermittent connectivity.

- **Require CIA of Data:** Security constraints include the need for data encryption, secure user authentication, and authorization mechanisms to protect sensitive information.

- **Performance Requirements:** The application must meet certain performance standards, including response time, throughput, and resource utilization.

- **UI Constraints:** Design and usability constraints may include adherence to specific user interface guidelines, accessibility requirements, and compatibility with different devices and screen sizes.

- **Data Storage Constraints:** Limitations on data storage capacity, database performance, and data retention policies may impact the design and functionality of the rental application.

2.8. *Assumptions & Dependencies*

Assumptions and dependencies are key elements to consider when designing and developing a system. They help stakeholders understand the context in which the system operates and identify potential risks. Here's a breakdown of assumptions and dependencies in the context of a system:

2.8.1 Assumptions:

- **User Behavior:** Assumptions about how users will interact with the system, including their preferences, expectations, and habits.
- **System Design:** Assumption about the how the application look like that enhanced the user experience
- **Data Quality**: Assumptions about the accuracy, completeness, and consistency of the data the system will handle.
- **Security:** Assumptions about the security measures in place, including encryption, authentication, and authorization mechanisms.
- **Performance**: Assumptions about the expected load on the system, concurrent users, and response times.

2.8.2 Dependencies:

- **Hardware Dependencies**: The system may depend on specific hardware components or configurations.
- **Software Dependencies**: Dependencies on third-party libraries, frameworks, or software components.
- **External Services:** Dependencies on external services, APIs, or web services that the system relies on for certain functionalities.
- **Data Dependencies:** Dependencies on external data sources or databases that the system needs to access.
- **Organizational Dependencies:** Dependencies on specific teams, individuals, or departments for collaboration, support, or decision-making.
- **Timeline Dependencies:** Dependencies on specific milestones, deadlines, or project schedules.
- **Communication:** Clear documentation helps in communicating expectations among team members and stakeholders, ensuring everyone is on the same page.
- **Resource Allocation**: Properly managing dependencies helps in allocating resources efficiently, ensuring that the necessary components are available when needed.

3. External Interface Requirements

3.1. Hardware Interfaces

The hardware interfaces of a Rentment Application refer to the physical connections and requirements related to the hardware components that the application interacts with. In modern web and mobile applications, hardware interfaces are often abstracted and standardized by the operating systems and platforms. However, it's still important to consider the hardware-related aspects that may impact the performance and functionality of the application. Here are some considerations for hardware interfaces:

- **Database Servers**
  Its depend the Data which we required

- **Load Balancers**:
  If load balancing is employed for distributing incoming traffic, specify the hardware requirements for load balancers.

- **Client Devices:**
  Define the minimum hardware specifications for devices used by users (renters and property owners).
  Consider compatibility with various device types, including desktops, laptops, tablets, and smartphones.

- **Cameras:**
  Camera are required when the pre and post product verification are captured

- **Payment Processing Hardware:**
  If the application supports in-person transactions (e.g., property viewings with on-site payments), specify any hardware requirements for point-of-sale (POS) devices.

- **Networking Equipment**:
  Specify the hardware requirements for networking equipment, including routers, switches, and firewalls.

3.2. Software Interfaces

Software interfaces for a Rentment Application refer to the points of interaction between the application and other software components, systems, or services. These interfaces ensure seamless communication and data exchange. Here are some key software interfaces to consider for a Rentment Application:

- **Mobile Device Interfaces:**
  Specify the compatibility and supported versions of mobile operating systems (e.g., iOS, Android) for the Rentment mobile application.
  Define the minimum required specifications for mobile devices (e.g., screen size, resolution).

- **Application Programming Interfaces (APIs):**
  Define APIs for communication between different components of the Rentment Application.
  Document the request and response formats, authentication mechanisms, and endpoints.
  Specify any third-party APIs integrated into the application.

- **Database Interfaces:**
  Specify the type of database system used (e.g., MySQL, PostgreSQL, MongoDB) and outline the database schema.
  Define data access and manipulation methods.

- **External Service Interfaces:**
  Document interfaces with external services, such as payment gateways, mapping services, or social media platforms.
  Specify the communication protocols and data formats use

- **Authentication Interfaces:**
  Specify the authentication mechanisms used for user access.
  Document how the application interfaces with identity providers if using third-party authentication services.

- **Payment Gateway Interfaces:**
  If the application involves financial transactions, specify the interfaces with payment gateways.
  Document supported payment methods and currencies.

3.3. Communications Interfaces

Communication interfaces in a Rentment Application involve the protocols, methods, and mechanisms used for communication between different components of the system, as well as between the application and external entities. Here are key communication interfaces to consider:

- **API Communication:**
  RESTful API:
  Define a RESTful API for communication between different components of the Rentment Application.
  Specify resource endpoints, request methods, and response formats.
  Graph QL:
  If applicable, define a Graph QL API with appropriate query and mutation endpoints.
- **Real-Time Communication:**
  Web Sockets:
  Specify the use of Web Sockets for real-time communication between the client and server.
  Define the Web Socket endpoints and messaging formats.
- **Database Communication:**
  Database Connectivity:
  Specify the communication protocols and methods for interactions between the application and the database.
  Define how the application retrieves, updates, and manipulates data in the database.
- **External Service Communication**:
  RESTful APIs:
  Define communication interfaces with external services, such as payment gateways, mapping services, or social media platforms.
  Document authentication mechanisms and data formats.
- **Authentication Communication:**
  OAuth, OpenID Connect:
  If using third-party authentication services, define the communication interfaces with OAuth or OpenID Connect.
  Specify how the application exchanges authentication toke

4. Functional Requirements

4.1. Functional Hierarchy

- **Login to user:**

  The first step of the functional requirement is Login step. In this function user login with user name and password to authorization. All users need to make authorized to log in their account

- **Customer Registration**

  The customer give their credentials to take rental services from the Rentment Application if they verified. With the help of that functionality user can easily perform different operation Application

  The customer Registration include

✓ Customer Name
✓ Customer Email
✓ Customer NIC
✓ Customer Password
✓ Customer city
✓ Customer Mobile no
✓ Customer Credit no

- **Vendor Registration**

  The vendor give their credentials to provide rental services from the Rentment Application to customer with their specify product if they verified. With the help of that functionality vendor can easily perform different operation Application

  The Vendor Registration include

  ✓ Vendor Name
  ✓ Vendor Email
  ✓ Vendor NIC
  ✓ Vendor Password
  ✓ Vendor city
  ✓ Vendor Mobile no
  ✓ vendor Credit no
  ✓ vendor Product category

- **Vendor Provide Product**

After Enter the vendors the all their credential so they proceed into their dashboard/Homepage vendors will provide their product filling add product form by providing sufficient detail about product such as

- ✓ Product id
- ✓ Available Date
- ✓ Cost according to duration

It consist of following sub Functionalities

- ✓ Add Product
- ✓ Update Product
- ✓ Delete Product

I. Add Product

Vendor can Add product according to category which it depend on .

II. Update Product

Vendor can update the existing product details such as rent available dates etc by entering the product id of that product

III. Delete Product

Vendor can delete the product by his own choice

- **Database Maintenance**
  The application Entity (customer or the vendor) such as product detail, personal detail, etc. and data provided by the customer such us booking detail their credential maintained in a data base by the application administrative

- **Finding and Booking the Products**
  The customer after accessing the application he searches the product if he they find the required product then he need to fill the booking form and submit to the database

- **Verify Booking and Intimate the vendors**
  The administrator after Login to the site they verifies the recent bookings from customer Then he need to forward all bookings detail to vendor

- **Authentication**
  Authentication is nothing but providing the security to the system here every one must enter their valid credentials into the system throw the login page. The login page will restrict unauthorized users. A users must provide the his credentials like user id and password for log into the system otherwise the request will throw back

- **Transaction Process**
  If they customer get their desired product they have the perform the transaction with the entering their account details and the communicate their product vendor in efficient manner

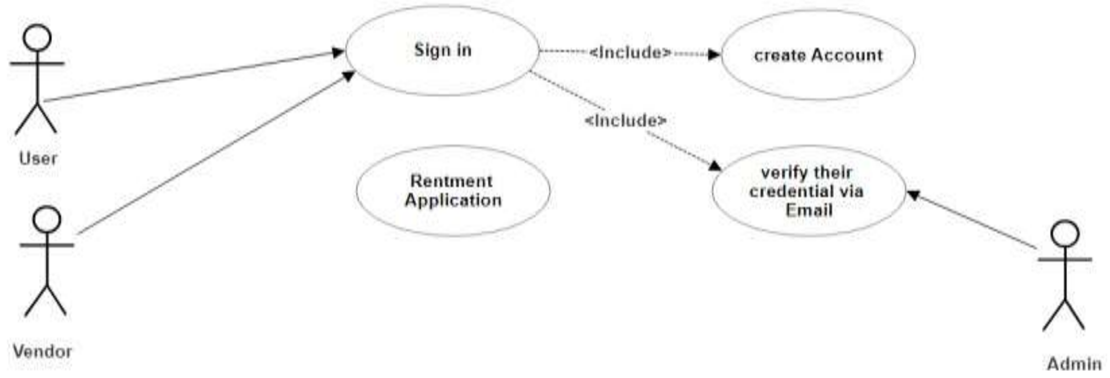
- **Accept Damage policy Agreement**
  Vendors have authority generate before they provide their product rent to the customers they create Damage policy agreement between them and the customers if there is any harm to the product

- **Pre and Post product verification**
  In there we verify the product using pre and post verification technique

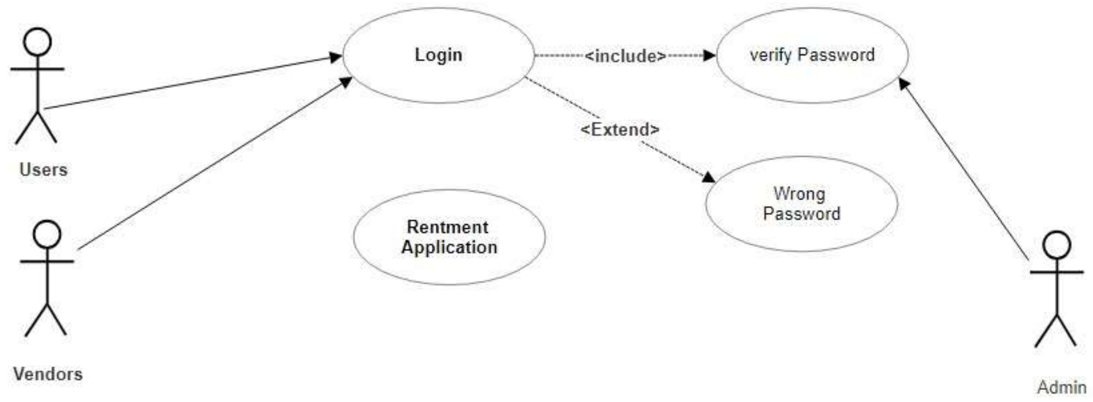
- **Log out**
  After using all operation user need to sign out to close the session

4.2. Use Cases

4.2.1. Login/Signup use case use-case diagram



<table>
<tr><th colspan="3">UC-01: Sign Up on Mobile Device</th></tr>
<tr><td><b><i>Use case Id:</i></b></td><td colspan="2">UC-01</td></tr>
<tr><td><b><i>Actors:</i></b></td><td colspan="2">User ,Vendor</td></tr>
<tr><td><b><i>Feature:</i></b></td><td colspan="2">Actor wants to make account to use application</td></tr>
<tr><td><b><i>Pre-condition:</i></b></td><td colspan="2">Android Phone
Internal Connection</td></tr>
<tr><td colspan="3"><b><i>Scenarios</i></b></td></tr>
<tr><td><b><i>Step#</i></b></td><td><b><i>Action</i></b></td><td><b><i>Software Reaction</i></b></td></tr>
<tr><td><b><i>1.</i></b></td><td>User opens the application by tapping on application icon.</td><td><b>2.</b> System provides options to select between sign in and sign up</td></tr>
<tr><td><b><i>3.</i></b></td><td>User taps on Sign up button.</td><td><b>4.</b> System opens the sign up form.</td></tr>
<tr><td><b><i>5.</i></b></td><td>User fills the form fields</td><td></td></tr>
<tr><td><b><i>6.</i></b></td><td>User submit the form</td><td><b>7.</b> System check all the fields for errors and update database for new account</td></tr>
<tr><td colspan="3"><b><i>Alternate Scenarios:</i></b></td></tr>
<tr><td colspan="3"><b><i>5a:</i></b> User entered the wrong data.
<b>6a:</b> The error message will be displayed.
<b>7a:</b> Application will ask user to re-enter data on error fields.</td></tr>
<tr><td colspan="3"><b><i>Post Conditions</i></b></td></tr>
<tr><td><b><i>Step#</i></b></td><td colspan="2"><b><i>Description</i></b></td></tr>
<tr><td></td><td colspan="2">Upon completion of this use case, the user can login to use application feature</td></tr>
<tr><td><b><i>Use Case Cross referenced</i></b></td><td colspan="2">Login</td></tr>
</table>



UC-02: Login	
Use case Id:	UC-02
Actors:	User ,Vendor
Feature:	Actor wants to sign in for access the application features
Pre-condition:	Sign up

Page

Scenarios		
Step#	***Action***	***Software Reaction***
1.	User open the application	**2.** System provides options to select between sign in and sign up
3.	User tap on Login button	**4.** System open the login form
5.	User fill the login form by entering User ID and Password	**6.** System will verify and validate the given user ID and password.
7.	User gets access to the application.	
Alternate Scenarios:		
5a: User entered the wrong ID or password. **6a:** The error message will be displayed. **7a:** Application will ask user to re-enter ID and password.		
Post Conditions		
Step#	***Description***	
	Upon completion of this use case, the user's account will be displayed to the user.	
Use Case Cross referenced	Set desired Location, Set Electronic Appliance, Make Gesture	

5. Non-functional Requirements

5.1. Performance Requirements

- The system should handle a specified number of simultaneous users.
- Response time for critical actions should be within acceptable limits.
- Balance the outgoing and incoming traffic Rapidly
- Action on Component perform the Rapidly
- Notification ,transaction, add product get product like functionality like work the efficiently

5.2. Security Requirements

- Implement secure user authentication and data encryption.
- Authorize person can allow to access the credential
- Provide strong security features
- Ensure secure payment transactions.
-

5.3. Compatibility of Application

- Hardware Compatibility: Ensuring that the system can run on specified hardware configurations.
- Software Compatibility: Ensuring compatibility with specific operating systems, browsers, or other software.

5.4. Reliability

- The percentage of time the system should be operational and accessible to users.
- The system's ability to continue operating in the presence of hardware or software failures.
- The time it takes for the system to recover from a failure or downtime.

5.5. Usability:

- Guidelines for the user interface, including accessibility and user experience.
- The ease with which users can learn to use the system.
- The provision of clear and comprehensive user documentation.

6. References N/A

7. Appendices N/A

Page