SLTMOBITEL
The Connection

Voucher Management System

SOFTWARE REQUIREMENT SPECIFICATION

Document Number:

Issue No:

Issue Date:

Issued By:

Document Control Sheet

Contact for Inquiries & Proposed Changes

Name : Uyani Panagoda

Designation : Trainee IT business analyst

E-mail : uyaninpanagoda@gmail.com

Prepared By

Name	Designation	Date	Signature
Uyani Panagoda	Trainee IT business analyst		
Sithumi Vindy	Trainee IT business analyst		
Navoda Prabashini	Trainee IT business analyst		

Checked By

Name	Designation	Date	Signature
Nishantha Alwis	Manager(S/D)/Digital Platforms		

Approved By

Name	Designation	Date	Signature
Nishantha Alwis	Manager(S/D)/Digital Platforms		

Revision History

Issue Number	Issued Date	Issued By	Summary of Changes	Document Section

Contents

1. Introduction

1.1 Purpose

The purpose of this document is to define the procedure for developing a Voucher Management System to promote SLT services to customers, including FTTH services. This system is designed to streamline the creation, approval, printing, distribution, activation, and tracking of vouchers within the organization. This process involves various stakeholders, including project managers, operational and batch-level approvers, procurement staff, store staff, agents, and customers.

Currently, the whole process is handled manually, resulting in a lot of paperwork. As the volume of vouchers increases, managing the process manually has become difficult. Human errors often occur during the process, especially in data entry operations and calculations, causing the entire employee team to double-check tasks, leading to a huge waste of time. To view the sales history, staff must keep physical logs. This process also poses security risks as documents and reports can be easily lost, damaged, or destroyed, leading to potential financial issues. Moreover, generating reports manually can result in inconsistencies and inaccuracies, and it is very labor-intensive. Approval delays often happen because vouchers get stuck in the approval process due to the need for physical signatures and manual checks. Furthermore, the department has to bear extra costs related to paper, printing, storage, and many other expenses.

The purpose of this Software Requirement Specification (SRS) report is to present a detailed description of the Voucher Management System application. This contains an overview of the overall SRS, including the purpose, project scope, overall description, functional features, non-functional requirement references, and external interface requirements. The features were proposed after thoroughly understanding client requirements. The document will give an understanding of how this application will help to improve the customer and client base and make business operations more efficient and accurate.

1.2 Scope

The Voucher Management System aims to develop a fully functional, automated solution to the manual voucher management process and wishes to integrate with other SLT systems later. Project managers could create vouchers, each containing four coupons. These vouchers then undergo approval at both operational and batch levels before being sent to procurement for printing. Once printed, the vouchers are distributed to stores and then to agents for sale. Upon sale, agents are responsible for activating the vouchers, making them usable by customers. Throughout this process, the system tracks the status of the vouchers, from creation to redemption, at SLT workshops, ensuring that all coupons are used. The admin portal plays a crucial role in this process, as admins manage agent registrations and oversee the system's operation to ensure smooth functionality. This involves creating a user-friendly web application with advanced features to facilitate voucher management.

1.3 Definitions, Acronyms, and Abbreviations

ERP – Enterprise Resource Planning

CRM – Customer Relationship Management

FTTH – Fiber to the Home

VMS: Voucher Management System

MFA: Multi-Factor Authentication

RBAC: Role-Based Access Control

1.4 References

2. Project Overview

2.1 Project Objectives

The proposed system aims to streamline several key processes. It will enable project managers to efficiently create vouchers and batches while implementing a multi-level approval process to ensure compliance and accuracy. The system will also facilitate the distribution of approved voucher batches to stores and agents, simplifying the activation process for agents upon sale. Additionally, it will provide an admin portal for managing agent registrations and overseeing system operations. Furthermore, the system will enhance tracking and reporting capabilities by updating the status of vouchers such as inactive, reserved, active, etc.

2.2 Background and Context

The organization has identified a need to improve the management of its voucher system. The existing process is manual and prone to errors, leading to inefficiencies and potential losses. The new Voucher Management System aims to address these issues by providing an automated, streamlined solution by minimizes human interference. This system will not only enhance operational efficiency but also provide better control and visibility over the voucher lifecycle. It will support compliance with organizational policies and improve customer satisfaction by ensuring vouchers are handled accurately and promptly.

2.3 Stakeholders

1. Project Manager- Create vouchers and batches and initiate the approval process.
2. Operational Level Approvers- Review and approve vouchers at the operational level.
3. Batch Level Approvers- Review and approve voucher batches for procurement.
4. The Procurement Department- Print approves voucher batches and sends them to stores.
5. Stores Department- Receive printed voucher batches and distribute them to agents.
6. Admin- Register and manage agents, oversee the system operations, and generate reports.
7. Agents- Sell vouchers, activate them upon sale, and ensure customers can use them.
8. Customers- Purchase and use activated vouchers at SLT workshops.

3. Risk Management

Enabling risk management for the voucher management system focuses on identifying potential risks throughout the project life cycle, analyzing identified risks and threats and finally mitigating and monitoring them. Through this risk assessment the team aims to meet the client expectation and minimize financial costs to deliver the project successfully to meet the client expectation. This includes assessing and controlling financial risks, market risks, technical risks and security risks.

In the risk plan the identified risks are listed with the mitigation plan and to better assess the risks, the impact is categorized as high, moderate, low and the severity has determined for each risk. The purpose of this is to implement effective mitigation strategies, prioritize risks and allocate the resources effectively to ensure the success of the VMS application.

The implemented risk management strategy will be as following,

- Risk Comment (CO) – description or an explanation of the identified risk or threat.
- Mitigation (MT) – actions to reduce probability of risk.
- Amelioration (AM) – safeguard to decrease severity.
- Opportunity (OP) – positive outcomes.
- Fallback (FB) – backup plan or often mention as "plan B".

Ref	Risk	Risk Management Strategy
01	Create fake OAuth/login pages.	**CO:** Unauthorized personals can create replicas of Google or Facebook login pages and when users attempt to log in, login credentials are stolen by unauthorized personals. **MT:** Educate users about the security threats using workshops to provide awareness. Implement and update security software that can detect and block phishing websites **AM:** Conduct regular security awareness campaigns Establish a clear incident response plan to quickly address phishing incidents. **OP:** Enhance user trust and loyalty. **FB:** Implement **account recovery** options. Maintain **regular backups** of critical data like financial data.
02	Phishing attacks via email	**CO:** Unauthorized, external people can send emails pretending to be

	to steal Service ID and Passwords.	from a inside department (IT Department/DP) asking internal users to login to fake portals using service ID and passwords. Such pages can be used to **steal sensitive information**, and **financial details**. **MT:** Educate users about the security threats using workshops to provide awareness. Implement and update security software that can detect and block phishing websites **AM:** Conduct regular security awareness campaigns Establish a clear incident response plan to quickly address phishing incidents. **OP:** Enhance user trust and loyalty. **FB:** Implement **account recovery** options. Maintain **regular backups** of critical data like financial data.
03	System downtime.	**CO:** Unexpected crashes or failures can make the voucher management system unavailable to users temporarily or permanently. **MT:** design and implement a **Disaster Recovery** plan including regular backups and a secondary system that can be activated in case of primary system failure. **OP:** increasing the **reliability** of the VMS.
05	Challenges in integrating the application with existing systems that are ERP and CRM.	**CO:** If a failure happens when integrating with existing systems it can lead to data silos and operational inefficiencies. **MT:** Conduct a thorough integration testing. **AM:** Use “Agile” development methodology so that can adapt to changes quickly and can do necessary enhancements and improvements. Implement user feedback mechanisms. **OP:** Increase client and end-user satisfaction. Increase the project quality since the product will have innovative features and a strong value. **FB:** Introduce alternative revenue streams as offering premium features.
06	Frauds	**CO:** Fraudulent activities can happen such as multiple redemptions of the same voucher/coupon.

		MT: Using unique voucher codes to prevent multiple redemptions.
07	Revenue loss.	**CO:** Potential impact on revenue if vouchers are misused or not tracked correctly and can cause financial loss. **MT:** Implement strict validation and tracking mechanisms for when issuing vouchers and redemption. **FB:** Have a secondary validation process to verify voucher transactions.

4. Requirements

4.1 Functional Requirements

The functional requirements of the VMS has demonstrated trough the detailed use case descriptions as below.

3.1.1 Internal user login.

User case ID	UC01	
Use case name	Internal user login	
Actors	System Admin, Project Manager, Approval Manager	
Description	This use case describes the process for an internal user (a SLT employee) to log into the system.	
Pre-conditions	- There should be a user who needs to do a task in voucher creation and managing process. - Should be a SLT employee and use service number as username and password use in slt systems. - The user should be registered.	
Post-conditions	- User should be logged into the system.	
Pre status	User – ***not allocated***	
Post status	User – ***not allocated***	
	Action	**System Response**
Success path	**1**. User directed to the login page. **3.** User enters and submits the login details.	**2.** System request for login details. **3.** System validates the details. **4.** Redirects to the relevant page based on role.
Alternate path	If user credentials are wrong, should display an error massage.	

3.1.2 External user login.

User case ID	UC02	
Use case name	External user login	
Actors	Agent, Sub-agent , User	
Description	This use case describes the process for an external user to log into the system using Google or Facebook authentication	
Pre-conditions	- There should be a user who needs to sell and verify a voucher and see the sales history. - The external user should be registered.	
Post-conditions	- The external user should be logged into the system.	
Pre status	User – ***not allocated***	
Post status	User – ***not allocated***	
	Action	**System Response**
Success path	**1.** The external user clicks “Sign up with Google” or “Sign up with Facebook”. **3.** The external user clicks “Sign up with Google” or “Sign up with Facebook”.	**2.** System redirects to Google or Facebook login page. **4.** System authenticates and logs in the user.
Alternate path	The external user cancels the login process on Google/Facebook.	
Exceptional path	The external user enters incorrect credentials.	The system displays an error message.
Special requirements	Secure authentication integration with Google and Facebook	

3.1.3 Create a new project.

User case ID	UC03	
Use case name	Create a new project	
Actors	Project Manager	
Description	This use case describes the process for a project manager to create a new project in the system.	
Pre-conditions	- The user needs to create a new project. - The user is logged in. - The user should be registered and active. - The user has project manager privileges. - The status of the batch and project should be “pending”.	
Post-conditions	- A new project should be created.	
Pre status	Project – ***not allocated*** , User – ***active***	
Post status	Project – ***inactive*** , User – ***active***	
	Action	**System Response**
Success path	**1.** User navigates to the “Project” page and clicks “Create Project”. **3.** The user fills out project details and clicks "Save".	**2.** The system displays the "Create New Project" form. **4.** The system saves and displays the new project in the list.
Alternate path	The user clicks "Cancel" while creating a project.	
Exceptional path	The user enters invalid data.	The system displays an error message and prompts for correction.
Special requirements	The project code should be an alphanumeric value.	

3.1.4 Approve the newly created project

User case ID	UC04	
Use case name	Approve the created project	
Actors	Operational-level approval manager	
Description	This use case describes the process for an approval manager to approve a created project in the system. If the user decides to reject the project or does not have the necessary privileges, the system handles these scenarios accordingly.	
Pre-conditions	- The project has been created and is pending approval. - The user should be registered and active. - The user is logged in. - The user has approval manager privileges	
Post-conditions	- The project's status should be updated to "approved".	
Pre status	Project – ***inactive*** , User – ***active***	
Post status	Project – ***active*** , User – ***active***	
	Action	**System Response**
Success path	**1** User navigates to the “new project” page and clicks “Create project”. **3.** The user clicks the “Details” icon. **5.** The user fills out the form and clicks the “Approve” button.	**2.** The system displays the created project. **4.** System redirects the user to the “Approve project” form. **6.** The system updates the project status to "approved"
Alternate path	The user decides to reject the project instead of approving it	
Exceptional path	The user fills form and clicks “Reject” button.	The system returns to the “created project”.
Special requirements	The approval action should be logged for auditing purposes	

3.1.5 Create a new batch.

User case ID	UC05	
Use case name	Create a batch.	
Actors	Project Manager	
Description	This use case describes the process for a project manager to create a new batch of vouchers in the system. **“Batch” is a collection of vouchers**.	
Pre-conditions	- The user needs to create a new batch. - The user is logged in. - The user should be registered and active. - The user has project manager privileges	
Post-conditions	- A new batch of vouchers should be created. - The status of the created batch should be “pending”.	
Pre status	Batch – ***not allocated*** , User – ***active*** , Project – ***active***	
Post status	Batch – ***pending*** , User – ***active*** , Project – ***active***	
	Action	**System Response**
Success path	**1.** User navigates to the “Create Batch” page and clicks “Create Batch”. **3.** The user fills out batch details and clicks "CREATE BATCH".	**2.** The system displays the "Create Batch" form. **4.** The system saves the new batch and displays it in the list.
Exceptional path	The user enters an invalid voucher count.	The system displays an error.
Special requirements	The voucher count should be less than the existing count.	

3.1.6 Approve a newly created batch.

User case ID	UC06	
Use case name	Approve the created batch.	
Actors	Approval manager	
Description	This use case describes the process for an approval manager to approve a batch created by a project manager.	
Pre-conditions	- User needs to approve a batch created by project manager. - User is logged in. - User is registered and active. - User has approval manager permissions.	
Post-conditions	- A task should be created in the back-end accordingly. - Status of the batch changes into “Approved”. - Redirects the user to the “Create Print” form.	
Pre status	Batch – ***pending*** , User – ***active*** , Project – ***active***	
Post status	Batch – ***approved*** , User – ***active*** , Project – ***active***	
	Action	**System Response**
Success path	**1.** User navigates to the “List of Batch” page and clicks “Create Batch”. **3.** User clicks “Details” icon. **5.** User fills form and click “Approve” button. **7.** User enters data and submits.	**2.** System displays the all the batch lists created for different projects. **4.** System redirects user to “Approve Batch” form. **6.** System redirects user to “Create Print” form. **8.** System changes the status to “Approve” and the “**Voucher Creation**” task will be created.
Exceptional path	The user fills form and clicks “Reject” button.	The system returns to the “List of Batch” without changing the batch status.

3.1.7 View project details.

User case ID	UC08	
Use case name	View project details.	
Actors	Project Manager, approval managers, admin	
Description	This use case describes the process for a user to view the details of a specific project. The user must be logged in, registered, and have the necessary privileges to view project details. The user navigates to the projects page, selects a project from the list, and the system displays the project details. If the user does not have the required permissions, an error message is displayed.	
Pre-conditions	- The user is logged in. - The user I registered. - The user has the necessary privileges to view project details	
Post-conditions	- Project details are displayed to the user	
Pre status	–	
Post status	–	
	Action	**System Response**
Success path	**1.** User navigates to the “Project List” page **3.** User selects a project from the list.	**2.** The system displays the list of projects. **4.** The system displays the details of the selected project.
Exceptional path	If the user does not have the necessary privileges.	The system displays an error message indicating insufficient permissions.

3.1.8 Search coupon number.

User case ID	UC09	
Use case name	Search coupon Number.	
Actors	Project manager, Agent	
Description	This use case describes the process for a user (project manager and agent) to search for and view details of a specific voucher or coupon. The user must be logged in and have the appropriate permissions. The user navigates to the search page, enters the voucher or coupon number, and the system retrieves and displays the relevant details. If the entered number is invalid or not found, the system will display an error message. **Please note that the Agent has not sold the voucher to customer yet.**	
Pre-conditions	- User needs to find the details about a particular voucher/coupon. - User has Project Manager Permissions or Agent Permissions.	
Post-conditions	- User is logged in.	
Pre status	–	
Post status	–	
	Action	**System Response**
Success path	**1.** User navigates to the “Search Coupon/Voucher Number” page and clicks “Create Batch”. **3.** User enters voucher/coupon number.	**2.** System displays a search box. **4.** System displays the voucher/coupon details.

3.1.9 Register an External User (Agent and Sub-agent)

User case ID	UC10	
Use case name	Register an external user.	
Actors	Admin.	
Description	This use case describes the process for an admin to register a new agent and a sub-agent.	
Pre-conditions	- Admin needs to register a new user.	
Post-conditions	- The user is logged in. - The user has Admin permissions.	
Pre status	External user – ***not allocated*** , Admin - ***active***	
Post status	External user – ***not allocated*** , Admin - ***active***	
	Action	**System Response**
Success path	**1.** The user navigates to the "User List" and clicks the "Register User" button. **3.** User selects the "User Type" as external. **5.** The user enters valid inputs for agent and sub-agent and clicks the "Register" button. .	**2.** The system navigates the user to the "User Registration" page and displays a drop-down field. **4.** The system redirects the user to the user registration form. **6.** System validates the input data and adds to the new users to the database.

3.1.10 Register an Internal User (SLT Employee)

User case ID	UC12	
Use case name	Register an internal user.	
Actors	Admin.	
Description	This use case describes the admin register a SLT employee into the system as an internal user.	
Pre-conditions	- Admin needs to register a new internal user.	
Post-conditions	- The user is logged in. - The user has Admin permissions.	
Pre status	External user – ***not allocated*** , Admin - ***active***	
Post status	External user – ***not allocated*** , Admin - ***active***	
	Action	**System Response**
Success path	**1.** User navigates to the "User Registration". **3.** User selects internal. **5.** The user enters the Service ID and clicks the "Search" button. **7.** User selects the "Role" from the drop-down list and clicks "Register".	**2.** The system navigates the user to the "User Registration" page and displays a drop-down field. . **4.** The system displays a search field. **6.** System displays the information of the employee whose service ID matches the service ID. **8.** System validates the input data and adds to the new user to the database.

3.1.11 Issue a batch to print

User case ID	UC07	
Use case name	Issues a batch to print.	
Actors	Project Manager	
Description	This use case describes the process for a project manager to issue an approved batch of vouchers for printing. The user selects the batch to print, and proceeds with the batch operation. After entering the necessary data and submitting the form, the system sends a read protected Excel file and to open it a key will be sent via SMS. The details (email, contact details) should be sent in read mode. If there is any *delivery failure* the system should allow user to reattempt and a log for reattempts should be maintain accordingly.	
Pre-conditions	- The user needs to send the approved batch of vouchers to print. - The user is logged in. - The user should be registered. - The user has project manager privileges and should be active.	
Post-conditions	- The batch status should be updated to " ready to print "	
Pre status	Batch – ***approved*** , User – ***active*** , Project – ***active*** , Voucher – ***inactive*** , Coupon – ***inactive***	
Post status	Batch – ***ready to print*** , voucher - ***inactive*** , coupon- ***inactive***	
	Action	**System Response**
Success path	**1.** User navigates to the “Batch list” page and clicks “Create Batch”. **3.** User selects the batch to be printed and clicks the “Batch Operation” **5.** The user clicks the “Update” button. **7.** User enters data and submits.	**2.** The system displays the batch list created for different projects. **4.** System redirects the user to the “Batch Operation” page. **6.** System redirects the user to the “Create Print” form. **8.** System sends an Excel file to the procurement with the voucher numbers.
Alternate path	The user cancels the operation before submitting the form	
Exceptional path	The system fails to send the Excel file or alert message.	The system displays an error message and prompts the user to retry.

3.1.12 Loading printed vouchers to stores

User case ID	UC14	
Use case name	Loading printed vouchers to stores	
Actors	Print vendor, Procurement Officer, Store Manager	
Description	This use case describes the process of loading the printed vouchers into the store. The store manager receives the printed vouchers. The store manager accepts the printed vouchers and updates the status of the vouchers “**inactive**”. The printed voucher sets will be received in iterations, not as a single bulk delivery. To identify which list has been received, the vendor should send the voucher codes in a printed list along with the printed voucher sets. A back-end task should be created to this process accordingly. When agent requested vouchers, the store releases existing vouchers which are in “active” status.	
Pre-conditions	- Printed vouchers are received from the print vendor. - The batch status should be "Print-ready" before send the batch to print vendors.	
Post-conditions	- The voucher status of the printed batch should be updated to "inactive". - A back-end task should be created to the iterations and voucher codes. - The store inventory is updated with the received vouchers.	
Pre – voucher status	Batch – ***ready to print*** , Voucher – ***inactive*** , Coupon – ***inactive***, Project – ***active***	
Post –voucher status	Batch – ***goods received*** , Voucher – ***active*** , Coupon - , Project **-** ***active***	
	Action	**System Response**
Success path	**1.** User (Approval manager) approved the batch. **3.** User (print vendor) sends	**2.** System changes the status of batch as “**Approved**” and the “ready to print”.

	the printed list of vouchers and voucher codes to the store.	**4.** Status of the vouchers change to “**Inactive**” and a backend task will be created accordingly.
Exceptional path	The print vendor cancels the receipt operation before confirming.	The system fails to reflect the vouchers in the store inventory.
Special requirements	Partial deliveries must be handled, with the system allowing for partial status updates.	

3.1.13 Issue an voucher

User case ID	UC010	
Use case name	Log an issued voucher.	
Actors	Agent	
Description	This use case describes the process for an agent to log an issued voucher when a sale is made. The agent must be logged in and have the necessary permissions. The agent navigates to the "Issue Voucher" page, enters the sold voucher number, and submits it. The system logs the voucher issuance and confirms the action. If the entered voucher number is invalid, an error message is displayed.	
Pre-conditions	- The user needs to issue a voucher when sells. - The user has Agent permissions.	
Post-conditions	- The user is logged in.	
Pre status	Voucher **-** ***active*** , Coupon – ***active*** , User – ***active*** , Project **-** ***active***	
Post status	Voucher **-** ***reserved*** , Coupon – ***reserved*** , User – ***active*** , Project – ***active***	
	Action	**System Response**
Success path	**1.** The user navigates to the "Issue Voucher". **3.** The user views the details and clicks the "Issue Voucher" button and enter the issued date. .	**2.** The system displays the details of the voucher searched. **4.** System counts the validity period will be calculated using expire date and issued date. **5.** The status of the voucher should be change from "Active" to "Issued". **6.** When agent submits the voucher number, it will be tracked by the system.

3.1.14 Redeem coupon

User case ID	UC13	
Use case name	Redeem coupon	
Actors	Agent, cashier, Customer	
Description	This use case describes the process for an agent to redeem a sold voucher. If already redeemed, an error message is displayed. Updates the voucher status and confirms the redemption. If the voucher number is invalid or expired, an error message is displayed.	
Pre-conditions	- The customer must have purchased a voucher. - Status of coupon should be “**Active**”. - The validity period should **not be expired**. - The agent must be registered and logged into the system.	
Post-conditions	- The voucher is marked as redeemed in the system. - The customer receives confirmation of redemption. - The redeemed voucher is recorded in the system for tracking and reporting. - The user has Admin permissions.	
Pre status	Coupon – ***active*** , Voucher – ***active*** , Project - ***active***	
Post status	Coupon – ***redeemed*** , Voucher – ***active*** (if all coupons are redeemed then Voucher = “ ***redeemed***”) , Project – ***active***	
	Action	**System Response.**
Success path	**1.** The user navigates to the “Redeem Voucher” page. **3.** The user enters the voucher number and clicks the “Submit” button.	**2.** The system displays an input field for entering the voucher number. . **4.** The system validates the voucher number and updates the voucher status.
Alternate path	The user cancels the redemption process before submitting the voucher number	
Exceptional path	The voucher number is invalid or expired	System displays an error message and prompts the user to correct the input.
Special requirements	Ensure that the redemption process is secure, and the data is correctly validated and updated in real-time.	

4.2 Non-functional Requirements

- **Performance -** There will be no bulk upload only 1000 vouchers will be uploaded to the system at a time.
- **Reliability -** The system should have an uptime of 99.9% to ensure it is available most of the time. The system should recover from failures within 5 minutes to minimize downtime. The system should ensure that voucher data is accurate and consistent at all times.
- **Security -** The system should support multi-factor authentication. The system should implement role-based access control to ensure that users can only access functions they are permitted to use. All sensitive data, including voucher codes and user information, should be encrypted both at rest and in transit.
- **Usability** - The system should have an intuitive and user-friendly interface for both administrators and end-users. Comprehensive user manuals and help guides should be available for system users.
- **Maintainability -** The system should be designed with a modular architecture to facilitate easy updates and maintenance. The system code should follow best practices and coding standards to ensure readability and ease of maintenance.
- **Compliance -** The system should maintain comprehensive audit logs of all transactions and changes for compliance and review purposes.
- **Interoperability -** The system should support integration with third-party applications and services via APIs. The system should support standard data formats for easy data exchange.
- **Portability -** The system should be deployable on various platforms, including on- premises servers and cloud environments. The system should be developed using platform-independent technologies to ensure it can run on different operating systems.
- **Efficiency -** The system should optimize the use of resources (CPU, memory, etc.) to minimize operational costs. The system should be designed to minimize energy consumption, especially if deployed on physical servers.

4.3 Software Quality Attributes

- **Speed:** Utilizing best coding practices, the web system will exhibit efficient speed, facilitating seamless usage across various devices. Rigorous testing conducted by Quality Assurance Engineers ensures optimal performance and responsiveness.

- **User-friendly Interface:** The system will feature a user-friendly interface, enabling easy navigation and comprehension for all users.

- **Security:** Robust security measures will safeguard the web system against potential cyber threats, ensuring the confidentiality and integrity of user data.

- **Flexibility:** The system's architecture allows for the addition of new features without introducing defects, ensuring adaptability to evolving requirements.

- **Maintainability:** The web system will be designed for ease of maintenance and updates, enabling modifications based on client preferences and needs.

- **Testability:** The system will be equipped with comprehensive testing capabilities, allowing for seamless testing of functions at any time as required.

5. Design

5.1 System Architecture

The purpose of designing system architecture of the proposed VMS system is to provide a blueprint or a comprehensive framework for the entire system, components and their interactions and to provide the guidance to developers, designers and stakeholders on how the system should be built.

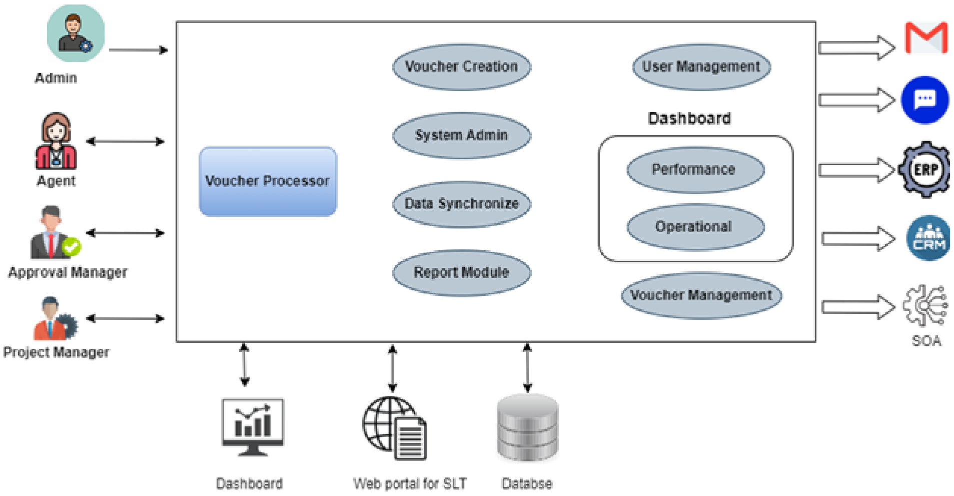


Figure: System architecture of VMS.

5.2 Detailed Design

This section will provide the documentation completed before the design phase, including database design, application design using class diagrams, user interface design, external system interfaces that describe how the VMS will interact with other external systems, API design that includes the API endpoints and request/response formats, and finally, the data flow, which describes the movement of data within the system. Furthermore the Functional Requirement Document is created to clearly define the functionalities that the software must deliver.

5.2.1 Documentation

During the initiation phase the goals and deliverables are finalized and the Software Requirement Specification is created and it was reviewed and it approved by the manager of Software Development, Digital Platforms. All the functional, non-functional requirements and a high level overview of the project are discussed in the SRS document and it shared with the development team to ensure that the team understands the project’s requirement and scope.

After the SRS was approved and finalized the Functional Requirement Document (FRD) was created by the business analysts during the beginning of design phase and the detailed functional requirements are defined and documented to provide the guide the design and development of the application. The FRD is reviewed and approved by the manager of Software Development, Digital Platforms. The mentioned documents will be maintained during the entire project life cycle.

5.2.2 Database design

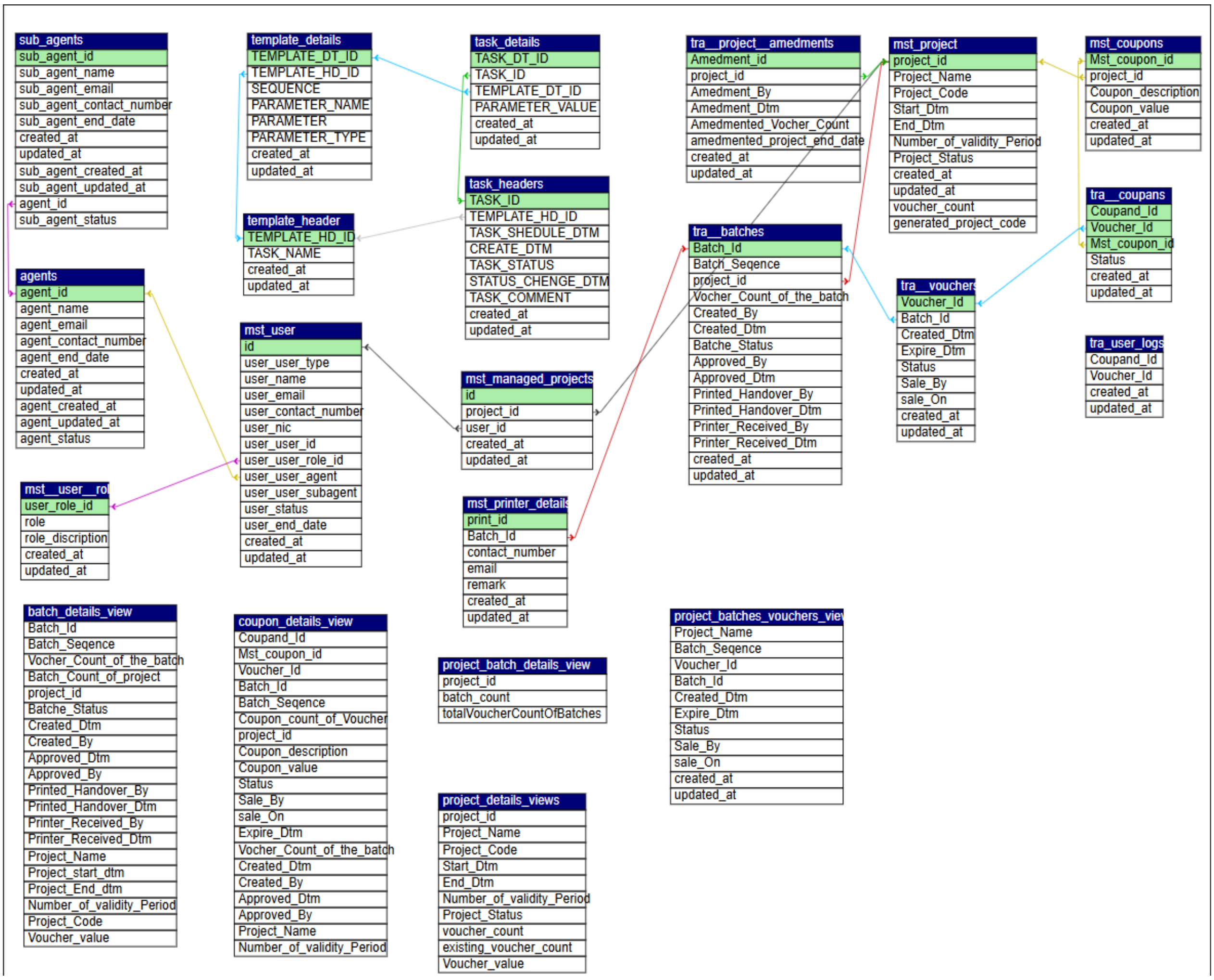


Figure: Class diagram for VMS.

5.2.3 Application design

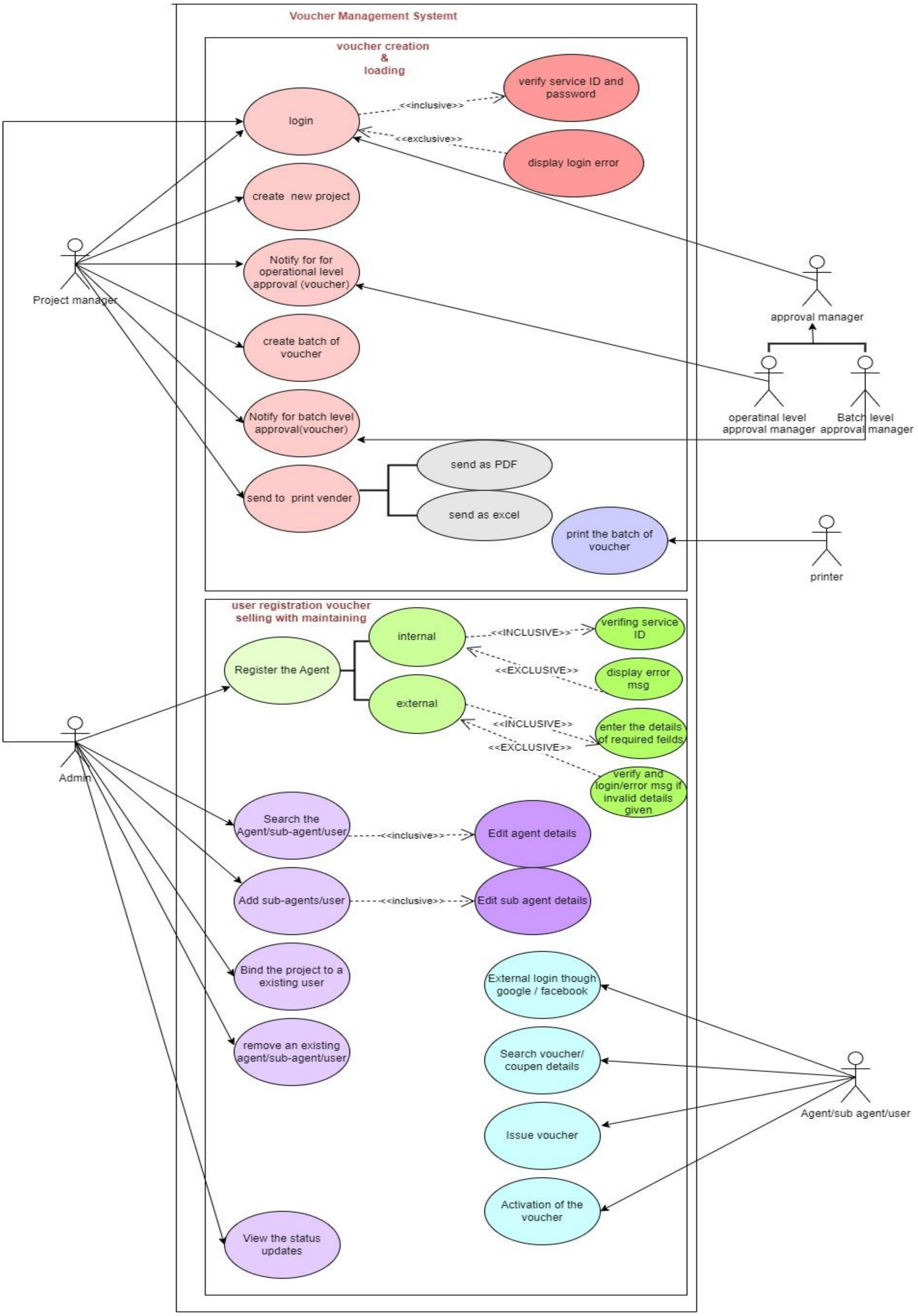
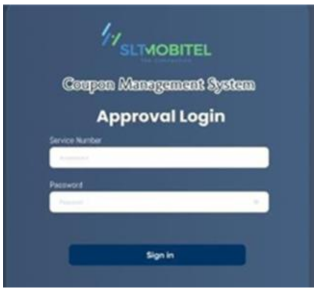

Voucher Management Systemt
voucher creation & loading
login
<<inclusive>>
verify service ID and password
<<exclusive>>
display login error
create new project
Notify for for operational level approval (voucher)
create batch of voucher
Notify for batch level approval(voucher)
send to print vender
send as PDF
send as excel
print the batch of voucher
Project manager
approval manager
operatinal level approval manager
Batch level approval manager
printer
user registration voucher selling with maintaining
Register the Agent
internal
external
<<INCLUSIVE>>
verifing service ID
<<EXCLUSIVE>>
display error msg
<<INCLUSIVE>>
enter the details of required feilds
<<EXCLUSIVE>>
verify and login/error msg if invalid details given
Admin
Search the Agent/sub-agent/user
<<inclusive>>
Edit agent details
Add sub-agents/user
<<inclusive>>
Edit sub agent details
Bind the project to a existing user
remove an existing agent/sub-agent/user
View the status updates
External login though google / facebook
Search voucher/ coupen details
Issue voucher
Activation of the voucher
Agent/sub agent/user

Figure: Use case diagram for VMS.

5.2.4 User Interface Design

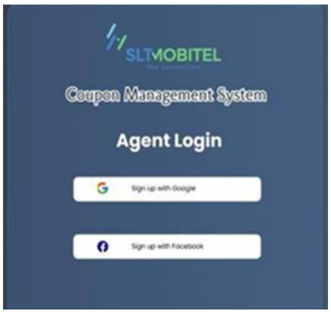
User interfaces of VMS are designed by the UI/UX team by using Figma in the design phase after gathering requirements from potential user to inform the design according to the UI principles.

1. Internal user login



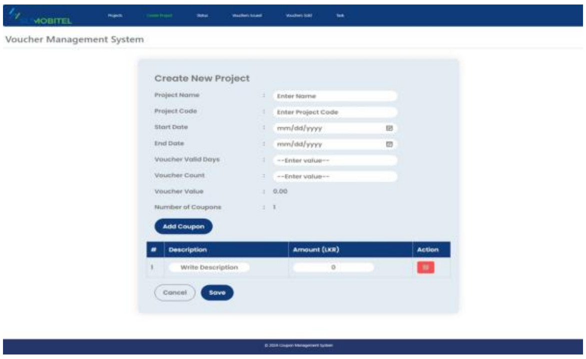
Internal users (SLT employees) who are project manager, approval manager and admin can login to their account using the login page where they enter service number as username and password to authenticate their login. After they logged in, users will be redirected to relevant pages based on their roles.

2. External user login



External users, who are agents, sub –agents can login to their account via sign in through there Google or Facebook account login. After they logged in, users will be redirected to relevant page.

3. Create project



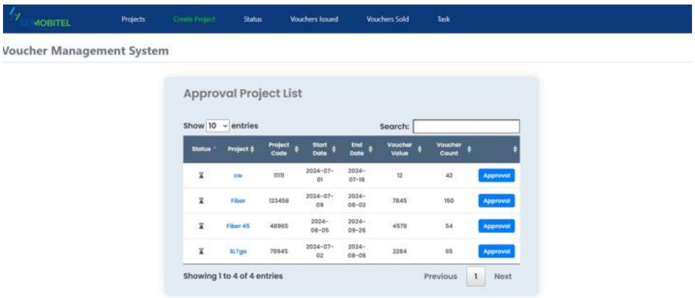
Create New Project

Project Name	:	Enter Name
Project Code	:	Enter Project Code
Start Date	:	mm/dd/yyyy
End Date	:	mm/dd/yyyy
Voucher Valid Days	:	--Enter value--
Voucher Count	:	--Enter value--
Voucher Value	:	0.00
Number of Coupons	:	1

#	Description	Amount (LKR)	Action
1	Write Description	0	

The user (project manager) will be able to create a new voucher by including four customized coupons name of the project, a unique project code, the create date of the voucher, and expiration date value of the voucher is auto-filled.

4. Approve project



Approval Project List

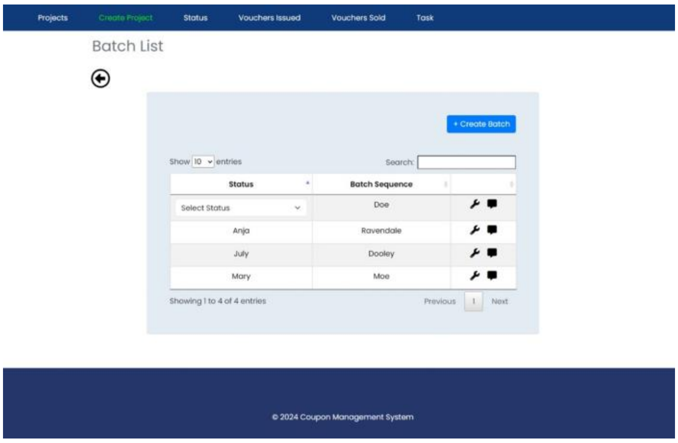
Show 10 entries

Status	Project	Project Code	Start Date	End Date	Voucher Value	Voucher Count	
⌛	cw	11111	2024-07-01	2024-07-18	12	42	Approval
⌛	Fiber	123456	2024-07-09	2024-08-02	7845	150	Approval
⌛	Fiber 45	48965	2024-08-05	2024-09-26	4578	54	Approval
⌛	SLTgo	78945	2024-07-02	2024-08-08	2284	65	Approval

Showing 1 to 4 of 4 entries

This interface allows users to approve or reject a batch of vouchers. The user (approval manager) can give approval to a created batch (a list of vouchers) by visiting the page “Approve Batch” and user can fill the “Remark” field and can approve the particular batch. Users can enter remarks in a provided text field and then click either the "Approve" or "Reject" button to finalize the decision.

5. Create batch

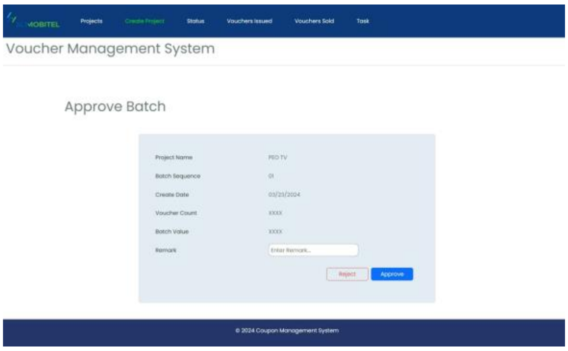


Status	Batch Sequence	
Select Status	Doe	
Anja	Ravendale	
July	Dooley	
Mary	Moe	

Showing 1 to 4 of 4 entries

This interface is a "Batch List" screen from a Voucher Management System. It features a table listing batches with columns for "Status" and "Batch Sequence." The table includes options to select a status and edit or delete entries. There are navigation controls to show entries and search functionality. A "Create Batch" button allows user to create new batches. The navigation bar at the top includes links to "Projects," "Create Project," "Status," "Vouchers Issued," "Vouchers Sold," and "Task."

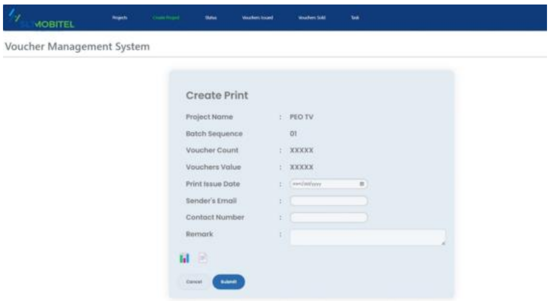
6. Approve batch



Project Name	PEO TV
Batch Sequence	01
Create Date	03/23/2024
Voucher Count	XXXX
Batch Value	XXXX
Remark	Enter Remark...

This interface is an "Approve Batch" screen from the Voucher Management System. It displays details of a batch including "Project Name," "Batch Sequence," "Create Date," "Voucher Count," and "Batch Value." There is a field to enter remarks. The screen has two buttons, "Reject" and "Approve," for *decision-making on the batch*. The navigation bar at the top includes links to "Projects," "Create Project," "Status," "Vouchers Issued," "Vouchers Sold," and "Task."

7. Create print



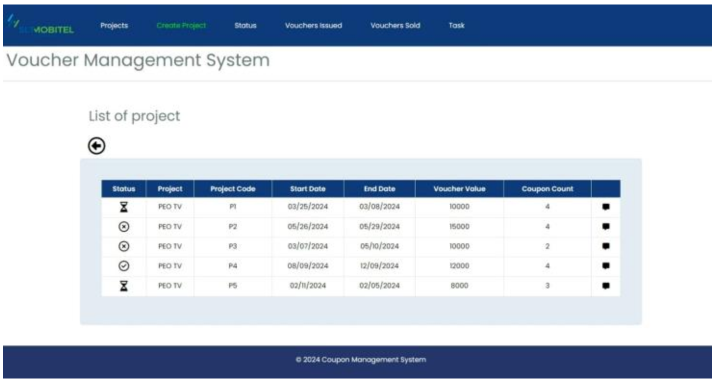
Voucher Management System

Create Print

Project Name	:	PEO TV
Batch Sequence		01
Voucher Count	:	XXXXX
Vouchers Value	:	XXXXX
Print Issue Date	:	mm/dd/yyyy
Sender's Email	:	
Contact Number	:	
Remark	:	

"Create Print" screen allows users to input details for printing vouchers, including "Project Name," "Batch Sequence," "Voucher Count," "Vouchers Value," "Print Issue Date," "Sender's Email," "Contact Number," and "Remark." The "Cancel" and "Submit," buttons are for managing the print creation process. The navigation bar at the top includes links to "Projects," "Create Project," "Status," "Vouchers Issued," "Vouchers Sold," and "Task." The User can send the created batch to print as Excel or PDF form after entering the remark.

8. View project details



Voucher Management System

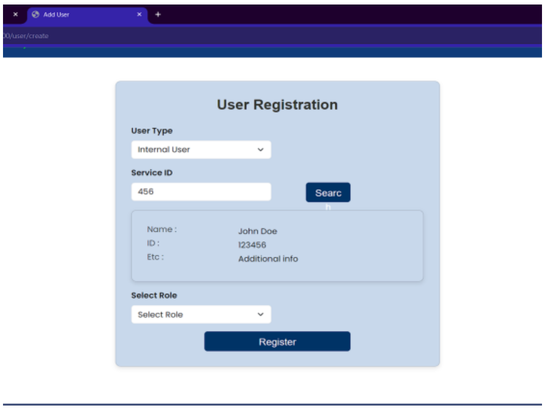
List of project

Status	Project	Project Code	Start Date	End Date	Voucher Value	Coupon Count	
⧗	PEO TV	P1	03/25/2024	03/08/2024	10000	4	💬
⊗	PEO TV	P2	05/26/2024	05/29/2024	15000	4	💬
⊗	PEO TV	P3	03/07/2024	05/10/2024	10000	2	💬
✓	PEO TV	P4	08/09/2024	12/09/2024	12000	4	💬
⧗	PEO TV	P5	02/11/2024	02/05/2024	8000	3	💬

© 2024 Coupon Management System

The interface is a Voucher Management System displaying a table of projects where user can view the project details. The table includes columns for Status, Project, Project Code, Start Date, End Date, Voucher Value, and Coupon Count. The top menu has options for Projects, Create Project, Status, Vouchers Issued, Vouchers Sold, and Task.

9. Register an internal user



User Registration

User Type: Internal User

Service ID: 456

Searc

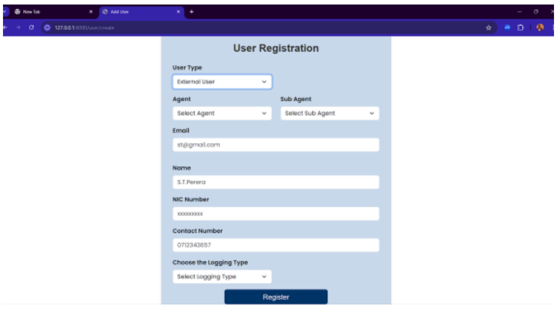
Name : John Doe
ID : 123456
Etc : Additional info

Select Role: Select Role

Register

In this user registration interface user can fill necessary information regarding to internal users and can register them to the system to complete their registration.

10. Register external user



User Registration

User Type: External User

Agent: Select Agent

Sub Agent: Select Sub Agent

Email: st@gmail.com

Name: S.T.Perera

NIC Number: xxxxxxxxx

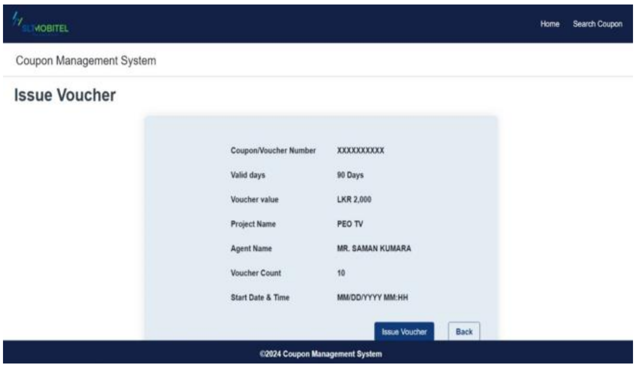
Contact Number: 0712343657

Choose the Logging Type: Select Logging Type

Register

In this user registration interface user can fill necessary information regarding to external users and can register them to the system to complete their registration.

11. Issue a voucher



Coupon Management System

Issue Voucher

Coupon/Voucher Number	XXXXXXXXXX
Valid days	90 Days
Voucher value	LKR 2,000
Project Name	PEO TV
Agent Name	MR. SAMAN KUMARA
Voucher Count	10
Start Date & Time	MM/DD/YYYY MM:HH

©2024 Coupon Management System

It includes fields for entering the voucher number, coupon/voucher number, valid days, voucher value, project name, agent name, voucher count, and start date/time. Users can fill in the required information and then either submit the form to issue the voucher or reset the form to clear all fields.

6. Implementation

In the implementation phase of the Voucher Management System Software Requirements Specification (SRS), the focus shifts from planning and design to actual development and deployment. This section introduces how the system will be built and put into operation.

6.1 Development Environment

Describes the purpose of the Voucher Management System (VMS) and its scope within the development phase.

- Programming Languages - HTML, PHP, Python
- Development frameworks - Bootstrap, Laravel
- Database - MySQL
- Tools - Visual Studio Code, Bitbucket

6.2 Coding Standards

In coding standards for a Voucher Management System SRS, would typically include this guideline.

Naming Conventions	Standards for naming variables, functions, classes, and other entities in the codebase to ensure clarity and consistency.
Formatting Rules	Guidelines for code indentation, line length, spacing, and use of comments to enhance readability.
Documentation Requirements	Requirements for documenting code, including inline comments, function/method headers, and overall code structure documentation.
Error Handling Practices	Standards for handling errors and exceptions to ensure robustness and reliability of the system.
Code Review Process	Procedures and criteria for conducting code reviews to maintain code quality and adherence to coding standards.
Version Control Guidelines	Practices for using version control systems (Git) effectively, including branching strategies and commit message conventions.
Security Guidelines	Guidelines for implementing secure coding practices to protect against vulnerabilities such as SQL and data breaches.
Performance Considerations	Considerations and guidelines for writing efficient code to optimize system performance, including resource management and algorithmic efficiency.

These standards ensure that the codebase is maintainable, scalable, and secure throughout the development and maintenance phases of the Voucher Management System.

6.3 Source Code Management (SCM)

Source code management ensures that all code changes are tracked, versioned and maintained efficiently. This allows team to manage code versions history and integrity. The source code management plan for the Voucher Management System has mentioned below.

- Bitbucket is chosen as a repository hosting service to host the code.
- The repository is private and has restricted access.
- Repository has directories for source code, documentation and configuration files.
- The main branch / master branch is defined.
- Feature branches are defined for bug fixes and development.
- Using “pull requests” to review and merge code changes.
- Backup and recovery.
- Documentation

7. Testing

7.1 Test Plan

The purpose of project test plan is to define the approach and activities for ensuring that the project delivers the required level of quality. The key purpose of project test plan is to defining quality objectives, identifying quality responsibilities, maintaining logs, establishing quality standards, outlining quality management activities, identifying quality checkpoints, defining quality management tools and technologies and ensuring continues improvement.

By implementing below mentioned quality control and audit processes, the Voucher Management System can be thoroughly tested against its quality specifications, ensuring a high-quality user experience and satisfaction.

- Code Review : Regular code reviews will be conducted by the development team to ensure adherence to coding standards, best practices, and quality guidelines.
- Automated Testing : Automated testing suites will be implemented to conduct unit tests, integration tests, and regression tests automatically.
- Manual Testing : Manual testing by QA engineers will complement automated testing efforts, allowing for comprehensive validation of the application's features, usability, and performance from a user perspective.
- Security testing : Conduct security testing to identify vulnerabilities and ensure the protection of user data, including personal information and payment details.
- Performance Monitoring: Continuous performance monitoring tools will be employed to track system performance metrics such as response times, latency, and resource utilization.

Maintaining detailed and organized logs for a Voucher Management System helps in troubleshooting, ensuring system integrity, preventing fraud, and meeting compliance requirements.

1) Define Key Events to Log - Log when the key events happen like voucher creation, voucher assignment, voucher redemption.
2) Implement log level for general operations
3) Use a structured format for logs.
4) Regular backups and Management
5) Periodically analyze logs and generate reports.

7.2 Test Cases

The purpose of writing test cases is to ensure the quality and functionality of the software and to identify defects early in the ongoing Voucher Management System (VMS). This process ensures that the VMS meets the requirements and business needs. The team aims to identify high-risk areas in the software, allowing them to focus on critical and important scenarios and issues immediately. The test cases cover all aspects of the VMS, including various user scenarios.

Test cases for all paths—including the project manager path, approval path, admin path, and agent path—have been written and reviewed by the manager. The manually tested functionalities include login, registration, creating projects and batches, approving projects and batches, creating tasks, executing voucher creation tasks, printing vendors, redeeming coupons, listing batches, listing sold vouchers, searching by coupon/voucher number, viewing voucher details of batches, viewing coupon details of vouchers, adding agents and sub-agents, viewing the user list, viewing the agent list, viewing project details, and viewing batch history (read-only UI).

8 Deployment

8.1 Deployment Plan

Deploying the Voucher Management System (VMS) is a crucial step that demands careful planning and precise execution to ensure a smooth transition from development to production. It’s broken down into several key phases, starting with pre-deployment activities. These include obtaining all necessary approvals, preparing the deployment environment, and double-checking the deployment checklist. During the actual deployment phase, we install and configure the VMS on our production servers, closely monitoring everything to ensure that all components are integrated and working properly. Post-installation, we run a series of validation tests to confirm that everything is functioning as expected and that all data has been correctly migrated.

The final phase is post-deployment, where we focus on training our users, providing ongoing support, and closely monitoring the system for any issues. The deployment plan for the VMS serves as our roadmap to a smooth and successful rollout. A detailed communication plan is also part of this strategy, keeping everyone informed about progress and any potential impacts on operations. This deployment plan ensures we handle the rollout systematically, minimizing risks and ensuring a smooth transition.

8.2 Installation Instructions

To install the VMS, follow a detailed, step-by-step guide to ensure a smooth process. Begin by preparing the installation environment and verifying that all system requirements are met, such as compatible operating systems, sufficient disk space, and necessary software dependencies. Once confirmed download the installation package and extract it to the designated directory.

The installation process involves running a setup and various configuration steps, including specifying database connections, setting up server parameters, and configuring network settings. Clear prompts and instructions at each step help ensure that the correct information is entered. After the initial setup, the script installs the necessary files and dependencies, including setting up the database schema and application components. Once installation is complete, run a series of validation checks to ensure everything is installed correctly and working as intended. This includes testing database connections, verifying web server configurations, and ensuring application services are running. The instructions also provide troubleshooting tips and support contact information in case any issues arise.

8.3 Rollback Plan

It's important to have a rollback plan for the VMS as a safety net in case anything goes wrong during deployment. This plan outlines the steps to restore the system to its previous state to maintain stability and integrity. Before starting the deployment, we take comprehensive backups of the current production environment, including databases, application files, and configuration settings. These backups act as our baseline and allow us to revert if necessary. If issues arise during deployment that requires a rollback, we first identify the root cause through detailed logging and monitoring.

Once the cause is determined, we stop all VMS services and restore the backed-up files and databases to their original locations, reversing any changes made to network and server configurations. After the restoration, we conduct validation tests to ensure everything is back to normal and all data is intact. The rollback plan also includes communication protocols to keep all stakeholders informed about the rollback decision and its implications. Having a well-documented and tested rollback plan ensures that we can quickly minimize downtime and maintain business continuity if anything goes wrong.

9. Maintenance and Support

9.1 Maintenance Plan

- **Releases:** The Voucher Management System (VMS) will have scheduled releases to introduce new features, enhancements, and security updates. Major releases will occur bi-annually, while minor updates and patches will be applied as needed to maintain system integrity.

- **Component Releases:** Component releases will focus on individual modules within the VMS, ensuring that each part remains current and optimized. These releases will be managed based on performance monitoring and user feedback, with updates applied as necessary.

- **Incidents, Problems, Changes:** All incidents, problems, and changes will be documented using a standardized process. Incidents will be categorized and prioritized to ensure effective resolution. Change requests will follow a formal process for assessment and implementation.

- **Priority – driven Development:** Development activities will be prioritized according to the severity and impact of reported issues. High-priority incidents will be addressed swiftly to minimize disruptions to SLT Telecom's operations and customer service.

- **Tracking of Change of Request:** Change requests will be tracked using a dedicated ticketing system. Each request will receive a unique identifier to facilitate tracking, ensuring transparency and accountability throughout the resolution process.

9.2 Roles and KPI

The maintenance and support team will consist of the following roles:

- **System Administrator**: Responsible for overseeing the overall health and performance of the VMS.
- **Support Engineers**: Handle user inquiries and manage incident resolutions.
- **Development Team**: Focus on implementing updates and system enhancements.

Key Performance Indicators (KPIs) will be established to assess the effectiveness of the maintenance and support processes, including:

- Average response time to incidents.
- Percentage of incidents resolved within the SLA.
- User satisfaction ratings.

9.3 Support Plan

9.3.1 Support Model

The support model for the VMS will follow a tiered structure, providing users with multiple levels of assistance based on the complexity of their requests. This approach ensures efficient handling of user inquiries and technical issues.

9.3.2 Support Units and the Generic Support Units

Support units will be established to address various user needs, including:

- **Technical Support Unit**: Focuses on technical inquiries and troubleshooting.
- **User Training Unit**: Responsible for conducting training sessions and creating educational resources.

Generic support units will provide foundational services, including general inquiries and resource availability. These units serve as the initial point of contact for all user support requests.

9.3.3 Incident Resolution

The incident resolution process will follow these steps:

1. Incident logging through the support ticketing system.
2. Initial assessment by the Technical Support Unit.
3. Escalation to the Development Team for unresolved issues.
4. User communication regarding resolution status and outcome.

9.3.4 Support Timeline

Support requests will adhere to a defined timeline:

- **Critical issues**: Response within 1 hour; resolution within 4 hours.
- **Non-critical issues**: Response within 24 hours; resolution within 72 hours.

KPIs for user support will include:

- First contact resolution rate.
- Average time to resolve incidents.
- User feedback and satisfaction scores.