Contents

[Introduction 2](#_Toc167877628)

[Purpose 2](#_Toc167877629)

[Scope 2](#_Toc167877630)

[Definitions, Acronyms, and Abbreviations 2](#_Toc167877631)

[References 2](#_Toc167877632)

[Project Overview 2](#_Toc167877633)

[Project Objectives 2](#_Toc167877634)

[Background and Context 3](#_Toc167877635)

[Stakeholders 3](#_Toc167877636)

[Project Management 3](#_Toc167877637)

[Project Plan 3](#_Toc167877638)

[Schedule 3](#_Toc167877639)

[Monitoring and Control 4](#_Toc167877640)

[Resource Management 4](#_Toc167877641)

[Risk Management 4](#_Toc167877642)

[Requirements 4](#_Toc167877643)

[Functional Requirements 4](#_Toc167877644)

[Non-functional Requirements 4](#_Toc167877645)

[Design 5](#_Toc167877646)

[System Architecture 5](#_Toc167877647)

[Detailed Design 5](#_Toc167877648)

[User Interface Design 5](#_Toc167877649)

[Implementation 5](#_Toc167877650)

[Development Environment 5](#_Toc167877651)

[Coding Standards 5](#_Toc167877652)

[Testing 5](#_Toc167877653)

[Test Plan 5](#_Toc167877654)

[Objectives 5](#_Toc167877655)

[Scope 6](#_Toc167877656)

[Test Cases 6](#_Toc167877657)

[Test Environment 6](#_Toc167877658)

[Execution of the Test Plan 6](#_Toc167877659)

[Maintenance and Support 7](#_Toc167877660)

[Maintenance Plan 7](#_Toc167877661)

[Regular Updates 7](#_Toc167877662)

[Monitoring 7](#_Toc167877663)

[Appendices 7](#_Toc167877664)

[Glossary 7](#_Toc167877665)

[Additional Resources 7](#_Toc167877666)

**Voucher Management System (VMS)**

# Introduction

## Purpose

The functional requirements for the Voucher Management System (VMS) are being discussed with project stakeholders including the Product Development team, IT department, and other relevant personnel involved in its development, implementation, and maintenance.

## Scope

The VMS project focuses on managing and redeeming vouchers for FTTH new connections, but does not cover marketing strategies, customer feedback mechanisms, or non-technical aspects of voucher promotions.

## Definitions, Acronyms, and Abbreviations

FRD: Functional Requirements Document

FTTH: Fiber-to-the-Home

VMS: Voucher Management System

ERP: Enterprise Resource Planning

SOA: Service-Oriented Architecture

API: Application Programming Interface

SME: Small and Medium Enterprises

## References

- IEEE Standard for Software Project Management Plans

- Agile Manifesto

- ISO/IEC 25010:2011 (System and Software Quality Models)

# Project Overview

## Project Objectives

- Increase promotion and uptake of FTTH services through targeted coupon selling strategies.

- Improve operational efficiency in voucher distribution and redemption processes.

- Enhance customer experience by providing a seamless and user-friendly coupon redemption process.

- Optimize resource utilization by repurposing existing IT solutions for voucher management.

## Background and Context

The VMS project aims to improve voucher management and customer satisfaction by leveraging an existing Voucher Management System developed for SME businesses, addressing the current ERP system's lack of flexibility and scalability for effective voucher management.

## Stakeholders

Project Sponsor:

Project Manager:

Development Team: Software Engineers, UI/UX Designers

QA Team: Quality Assurance Engineers

End Users: Customer Service Representatives, Sales Team

Compliance Team: Legal and Regulatory Advisors

# Project Management

## Project Plan

The Voucher Management System (VMS) project will be implemented in four phases for systematic and efficient development and implementation.

Phases

|  |  |
| --- | --- |
| Initiation | The phase focuses on outlining the project scope and objectives to ensure all stakeholders have a clear understanding of the project's objectives. |
| Planning | During this phase, detailed project plans and schedules will be created to outline the project's roadmap and ensure proper resource allocation. |
| Execution | This phase involves executing project tasks, such as developing and testing the VMS. |
| Closure | The final phase focuses on completing the project, delivering the product, and obtaining stakeholder approval. |

## Schedule

The project schedule is divided into four phases, each with specific tasks and deadlines, and a detailed Gantt chart will visually represent the timeline and key milestones.

|  |  |
| --- | --- |
| Initiation | Define project scope and objectives.  Identify stakeholders and their roles.  Conduct initial risk assessment.  Prepare project charter. |
| Planning | Develop detailed project plans.  Create project schedule and Gantt chart.  Allocate resources.  Develop risk management plan.  Prepare budget estimates.  Conduct project kickoff meeting. |
| Execution | Develop software according to requirements.  Conduct regular progress reviews.  Perform system integration and testing.  Implement coding standards and source code management.  Ensure data synchronization between ERP and VMS.  Conduct user training and prepare documentation. |
| Closure | Finalize and deliver the VMS product.  Obtain stakeholder approval.  Conduct post-implementation review.  Close project and release resources.  Archive project documentation and lessons learned. |

## Monitoring and Control

Weekly status meetings are held to discuss progress, issues, and next steps.

Monthly progress reports provide detailed information on project status, risks, and budget updates.

The Change Management Process is designed to manage any modifications to the project scope or requirements.

## Resource Management

|  |  |
| --- | --- |
| Human Resources | Project Manager, Developers, QA Engineers, UI/UX Designers |
| Technical Resources | Development servers, workstations, software licenses |
| Materials | Documentation, training materials |

## Risk Management

|  |  |
| --- | --- |
| Technical Risks | Integration issues and performance bottlenecks are significant challenges in the current system. |
| Management Risks | Schedule delays and budget overruns are common issues that need to be addressed. |
| Mitigation Strategies | Regular risk assessments and contingency planning are essential for managing potential risks and ensuring smooth operations. |

# Requirements

## Functional Requirements

The ERP system is used for managing the sale and distribution of vouchers, with necessary configuration adjustments.

Utilize the existing VMS for coupon redemption for new FTTH connections, ensuring seamless customer onboarding.

## Non-functional Requirements

|  |  |
| --- | --- |
| Performance | The system is designed to accommodate 1000 concurrent users. |
| Security | This involves the use of data encryption, user authentication, and authorization. |
| Usability | The user interface (UI) is designed with a responsive and intuitive approach. |

# Design

## System Architecture

The system will utilize a multi-tier architecture consisting of a presentation, business logic, and data layer, with components interacting via RESTful APIs.

## Detailed Design

|  |  |
| --- | --- |
| Data Structures | Customer, Order, Product |
| Algorithms | Data validation, report generation |
| Interfaces | API endpoints for CRUD operations |

## User Interface Design

The mockups and prototypes will consist of:

Login Screen

Dashboard

Customer Profile Management

# Implementation

## Development Environment

|  |  |
| --- | --- |
| Programming Languages | JavaScript, Python |
| Tools | Visual Studio Code, Git, Docker |
| Platforms | AWS for hosting, Jenkins for CI/CD |

## Coding Standards

|  |  |
| --- | --- |
| Code Readability | The text emphasizes the importance of clear and concise naming conventions. |
| Documentation | Inline comments and API documentation are essential tools for enhancing user experience and ensuring smooth functionality. |
| Testing | Unit tests and code reviews are essential tools for evaluating and improving software development processes. |

Source Code Management

|  |  |
| --- | --- |
| Version Control | The Git with GitFlow branching strategy is a popular and effective tool for managing and executing code in Git. |
| Repository | The central repository on GitHub is a central platform for managing and organizing various projects and files. |
| Branching | The main branch is responsible for managing the development branch. |

# Testing

## Test Plan

The test plan outlines a strategy for evaluating the Voucher Management System (VMS) to ensure it meets all requirements and is free of defects, thereby validating its functionality, performance, and reliability.

### Objectives

The task involves ensuring that the software meets all functional and non-functional requirements.

The task involves identifying and documenting defects, focusing on critical issues that could potentially affect the system's functionality.

The quality assurance process involves implementing comprehensive testing strategies to ensure the overall quality of the software.

### Scope

The functional testing process ensures that the VMS accurately performs all specified functions.

Non-functional testing involves evaluating the system's performance, security, usability, and other non-functional attributes.

Regression testing is a crucial process to ensure that new code modifications do not negatively impact existing functionalities.

### Test Cases

|  |  |
| --- | --- |
| Input Data | The specific data sets will be utilized for functionality testing. |
| Expected Results | The system should produce the correct outputs when processing input data. |
| Acceptance Criteria | The test must meet certain conditions for it to be considered successful. |

### Test Environment

The test environment should closely resemble the production environment for accurate and reliable results, including hardware, software, and network conditions similar to production conditions.

|  |  |
| --- | --- |
| Hardware | Test Servers are servers with similar configurations to production servers to ensure consistent system behavior. |
| Software | Test all dependencies, including databases, ERP systems, CRM systems, and other integrated components, including their versions. |
| Network | Simulated Network Conditions\*\* are network configurations that simulate production conditions, ensuring the system performs well under expected load and latency. |

### Execution of the Test Plan

|  |  |
| --- | --- |
| Preparation | Configure the test environment, including hardware, software, and network configurations. |
| Test Execution | The test cases are conducted using the prepared input data, capturing the actual results and comparing them to the expected outcomes. |
| Defect Logging | The task involves identifying and documenting any discrepancies between the actual and expected results as defects, outlining their severity and impact. |
| Regression Testing | After addressing defects, it is crucial to re-run the test cases to ensure that the fixes did not introduce any new issues. |
| Reporting | The task involves preparing and presenting test results, including defect reports, and evaluating the overall quality of the system. |

# Maintenance and Support

## Maintenance Plan

The Voucher Management System (VMS) will undergo a comprehensive maintenance plan, ensuring its reliability and performance through scheduled updates, continuous monitoring, and proactive issue resolution.

### Regular Updates

Regularly scheduled patches and updates will be applied to the system to address security vulnerabilities, enhance performance, and introduce new features.

The process involves monthly minor updates and quarterly major updates, with updates tested in a staging environment before production deployment to ensure stability and compatibility.

### Monitoring

The system will undergo continuous system health checks to proactively detect and resolve any issues.

Monitoring tools and dashboards are utilized to track system performance and key metrics, while automated alerts are provided for anomalies or performance issues, ensuring timely resolution.

# Appendices

## Glossary

|  |  |
| --- | --- |
| API | The Application Programming Interface (API) is a set of tools and protocols used for creating and interacting with software applications. |
| CI/CD | Continuous Integration/Continuous Deployment is a method that involves integrating automation into the development stages of an app to deliver it frequently to customers. |

## Additional Resources

Templates are project templates and checklists that offer a standardized method for organizing and documenting project activities and procedures.

Supplementary Materials provide in-depth information on topics related to VMS and project management through additional reading and reference materials.