

Work Breakdown Structure (WBS)

(INVOICE PAYMENT AUTOMATION SYSTEM)



March 18, 2025

Table of Contents

[1. Project Overview 2](#_Toc193208730)

[2. Work Breakdown Structure (WBS) 3](#_Toc193208731)

[1. Planning Phase 3](#_Toc193208732)

[2. Design Phase 3](#_Toc193208733)

[3. Development Phase 3](#_Toc193208734)

[4. Testing Phase 3](#_Toc193208735)

[5. Deployment Phase 3](#_Toc193208736)

[6. Maintenance & Updates 4](#_Toc193208737)

[3. Technologies and Tools Used 5](#_Toc193208738)

# 1. Project Overview

The Payment Automation System aims to streamline financial transactions by automatically retrieving payments from Safaricom M-Pesa, recording them against existing invoices, updating the invoice balance, generating a new invoice, and providing options for downloading or displaying the updated invoice.

# 2. Work Breakdown Structure (WBS)

## 1. Planning Phase

1.1 Define system requirements (SRS document)  
1.2 Identify key stakeholders  
1.3 Research existing solutions and market needs  
1.4 Establish development and deployment timelines

## 2. Design Phase

2.1 Create UML diagrams (Use Case, Sequence, and Class Diagrams)  
2.2 Design database schema and entity relationships  
2.3 Design UI wireframes and user flow  
2.4 Define API architecture and integrations (M-Pesa API)

## 3. Development Phase

3.1 Backend Development (Django / Node.js)  
3.2 Frontend Development (React.js / Vue.js)  
3.3 Database Setup and Optimization (PostgreSQL / MySQL)  
3.4 Payment Gateway Integration (Safaricom M-Pesa API)  
3.5 PDF Invoice Generation Feature  
3.6 Implement User Authentication and Role-Based Access

## 4. Testing Phase

4.1 Unit Testing for individual components  
4.2 Integration Testing to ensure API connectivity  
4.3 Security Testing (data encryption, secure API calls)  
4.4 User Acceptance Testing (UAT) with sample transactions

## 5. Deployment Phase

5.1 Deploy Backend Services (AWS, Digital Ocean, or Heroku)  
5.2 Deploy Frontend Application (Netlify, Vercel)  
5.3 Connect Domain and Secure with SSL  
5.4 Monitor Initial Deployment and Fix Bugs

## 6. Maintenance & Updates

6.1 Monitor system performance and logs  
6.2 Regular security updates and patches  
6.3 Improve system efficiency based on user feedback  
6.4 Add new features and scalability improvements

# 3. Technologies and Tools Used

* **Backend Development:** Django (Python)/ flask or Node.js (Express.js) or Laravel (php)
* **Frontend Development:** React.js or Vue.js
* **Database:** PostgreSQL / MySQL /MongoDB
* **API Integrations:** Safaricom M-Pesa API, RESTful API for transactions(Daraja)
* **Hosting:** AWS, Orange host
* **Security:** OAuth, JWT Authentication, SSL Encryption
* **Testing Frameworks:** Jest (for frontend), PyTest (for backend)
* **PDF Generation:** Report Lab (Python) or jsPDF (JavaScript)

