

Gonona - Inventory Management System for Small Businesses

By
MD. Rifat Hossain
212-15-4123

FINAL YEAR DESIGN PROJECT REPORT

This Report Presented in Partial Fulfillment of the Requirements
for the **Degree of Bachelor of Science in Computer Science and
Engineering**

Supervised by

Dr. Arif Mahmud
Associate Professor
Department of Computer Science and
Engineering Daffodil International University

Co-Supervised by

Dr. Abdus Sattar
Associate Professor
& Director M.Sc
Department of Computer Science and
Engineering Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY
Dhaka, Bangladesh

September 17, 2025

APPROVAL

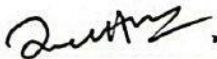
This Project titled “Gonona - Inventory Management System for Small Businesses”, submitted by MD. Rifat Hossain, ID No: 212-15-4123 to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 17 September, 2025.

BOARD OF EXAMINERS



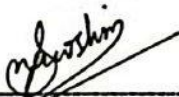
Dr. Sheak Rashed Haider Noori
Professor and Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Chairman



Dr. Zahid Hasan
Associate Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Samia Nawshin
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Dr. Md. Arshad Ali
Professor
Department of Computer Science and Engineering
Hajee Mohammad Danesh Science & Technology
University

External Examiner

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Dr. Arif Mahmud, Associate Professor**, Department of Computer Science and Engineering, Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:

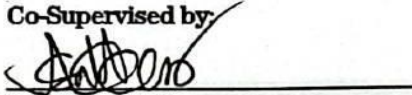


Dr. Arif Mahmud

Associate Professor

Department of Computer Science and
Engineering Daffodil International University

Co-Supervised by:

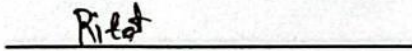


Dr. Abdus Sattar

Associate Professor & Director M.Sc

Department of Computer Science and
Engineering Daffodil International University

Submitted by:



MD. Rifat Hossain

Student ID: 212-15-4123

Department of Computer Science and
Engineering Daffodil International University

ACKNOWLEDGEMENTS

This work would not have been possible without the support and contributions of many individuals over the past two semesters. We are deeply grateful to everyone who has

assisted us in one way or another.

First, we express our heartfelt thanks and gratefulness to the almighty for His divine blessing making it possible for us to complete the **Final Year Design Project(FYDP)** successfully.

We are grateful and wish our profound indebtedness to **Dr. Arif Mahmud, Associate Professor**, Department of Computer Science and Engineering, Daffodil International University, Dhaka, Bangladesh. Deep knowledge and keen interest of our supervisor in the field of **Computer Science** to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

We would like to express our heartfelt gratitude to the Head of the Department of Computer Science and Engineering, for his kind help in finishing our project and also to other faculty members and the staff of the Department of Computer Science and Engineering, Daffodil International University.

We would like to thank our entire course-mates at Daffodil International University, who took part in this discussion while completing the coursework.

Finally, we must acknowledge with due respect the constant support and patience of our parents

ABSTRACT

Inventory management is a critical component of any retail or product-based business, directly impacting profitability, customer satisfaction, and operational efficiency. This project introduces a Software-as-a-Service (SaaS) based Inventory Management System that combines core inventory and sales tracking functionalities with an integrated Point of Sale (POS) system and SMS marketing capabilities, designed to serve small and medium-sized businesses (SMBs) that require a scalable, easy-to-use, and cost-effective solution to manage their day-to-day operations from a centralized online platform. The software allows business owners to maintain product inventory records and manage them thoroughly. Sales transactions which includes realtime monitoring of stock levels and low-stock warnings are all features of the vendor billing and customer service module, which also allows for the swift charging of clients as well as the capturing of sales history. A module of the software capable of SMS marketing is unique for its ability to communicate with clients through automated systems that send personalized promotional or order update messages. This also aids in customer retention which increases repeat sales. The system is software as a service, with the system being fully cloud hosted and having no local installation requirements. Each tier provides a unique pricing plan without being too expensive to stay within budget, which helps fulfill various corporate needs. Modular and scalable systems allow for future updates, while the multi-user access permission and role based access guarantees effective and safe use of the software. The system demonstrates dependable performance and usability based on the actual performance and effectiveness measures, and fully tested data, to provide a comprehensive system. This also improves sales through faster inventory control and SMS communications.

Table of Contents

Approval	i
Declaration	iii
Acknowledgements	iv
Abstract	v
List of Figures	vii
List of Tables	viii
1 Introduction	1
1.1 Introduction.....	1
1.2 Motivation.....	2
1.3 Objectives.....	2
1.4 Methodology.....	3
1.5 Project Outcome.....	4
1.6 Organization of the Report.....	5
2 Background	7
2.1 Introduction.....	7
2.2 Literature Review.....	8
2.2.1 Similar Applications.....	9
2.3 Gap Analysis.....	11
2.4 Summary.....	12
3 Research Methodology	13
3.1 Methodology/Requirement Analysis & Design Specification.....	13
3.1.1 Overview.....	13
3.1.2 System Design.....	13
3.1.3 Functional and Nonfunctional Requirements.....	14
3.1.4 UI Design.....	14
3.2 Detailed Methodology and Design.....	15
3.3 Project Plan.....	16
3.4 Task Allocation.....	18
3.5 Summary.....	21
4 Implementation and Results	22
4.1 Environment Setup.....	22

4.2	Testing and Evaluation/Performance/ Comparative Analysis.....	25
4.3	Results and Discussion.....	27
4.4	Summary.....	29
5	Engineering Standards and Design Challenges	30
5.1	Compliance with the Standards.....	30
5.1.1	Software Standards.....	30
5.1.2	Hardware Standards.....	32
5.1.3	Communication Standards.....	33
5.2	Impact on Society, Environment and Sustainability.....	34
5.2.1	Impact on Life.....	34
5.2.2	Impact on Society & Environment.....	34
5.2.3	Ethical Aspects.....	36
5.2.4	Sustainability Plan.....	37
5.3	Project Management and Financial Analysis.....	39
5.4	Complex Engineering Problem.....	41
5.5	Summary.....	44
6	Conclusion	45
6.1	Summary.....	45
6.2	Limitation.....	46
6.3	Future Work.....	47
	References	48

List of Tables

2.2	Summary of Literature Reviewed.	8
5.1	Mapping with complex problem solving.	41
5.2	Mapping with knowledge Profile.	42
5.3	Mapping with complex engineering activities.	43

Chapter 1

Introduction

This chapter outlines the project by presenting its problem, motivation, objectives, methodology, expected outcomes, and report structure.

1.1 Introduction

If you run a small or mid-sized business, keeping tabs on your inventory can feel a bit like herding cats. It's one of those things that'll either help you rake in cash or leave you stuck with empty shelves and no idea where your profits evaporated to. And honestly? In a ton of developing countries, shop owners are still stuck doing all this with little more than a pen, paper, and maybe the world's oldest Excel spreadsheet. No fancy tools, just messy notebooks and headaches. That's how stuff gets missed, shelves go bare, and opportunities go right out the window.

Now, there's Gonona coming in like a superhero for small shops. Picture this: real-time updates on what you've got in stock (or, let's be honest, what you've already run out of), sales that don't take ages to process, automatic "hey, you're running low!" heads-ups, and even those cheeky promo texts that have everyone rolling their eyes—and then actually buying something. It's all tied together in a tidy platform you don't need a computer science degree to use. If you've got the internet, you're good to go.

Oh, and Gonona's built as a SaaS app. So, no installing weird software or paying an arm and a leg up front. It's subscription-based—easy for shops that don't have piles of cash lying around and need something that'll just work, no fuss. They really thought about the folks who'd use this thing: easy setup, affordable, and it won't turn into another tech nightmare you have to babysit.

1.2 Motivation

Small businesses form the backbone of many economies around the world, including Bangladesh. Yet, it's still incredibly common to find these enterprises relying on outdated methods—think handwritten ledgers, clunky spreadsheets, or expensive, not-so-user-friendly software—to manage sales, inventory, and basic operations. The result? Missed chances, errors that just don't need to happen, and a level of inefficiency that holds them back from reaching their potential.

Introducing an SMS marketing module into their toolkit changes the game. Direct communication with consumers not only streamlines engagement but also opens avenues for growth—particularly crucial in regions where mobile phone use is widespread, yet reliable internet remains a challenge.

My goal with this project is to help bridge the digital divide experienced by these businesses. I want to use what I've learned in school to help small businesses, boost economic growth in the area, and come up with a solution that will have a big and lasting impact.

1.3 Objectives

The main goal of the proposed project is to make a cloud-based inventory management system just for small and medium-sized businesses. A niche specific Point of Sale (POS) system with inventory tracking capabilities. You can track transactions, process sales and purchases and bill customers in real time. Additionally, the system features an SMS marketing feature which assist outreach by allowing companies to send promotional messages. Role-based permissions helps administrators and staff to use the platform independently and safely.

Since the system is a Software-as-a-Service (SaaS) solution, any internet-connected device can access it. To preserve data integrity, promote system scalability, and guarantee that the system is consistently dependable and operates effectively, the architectural approach also calls for the use of contemporary development methodologies and technologies. These objectives, which seek to provide a comprehensive digital solution that enhances customer service, sales, and productivity, are in line with what small businesses actually need.

1.4 Methodology

The Gonona Inventory Management System was developed using a structured software development methodology that prioritized user-centric design, scalability, and flexibility. The project adopted the Agile development process to offer continuous feedback and iterative development at every stage of the project lifecycle.

The methodology included the following essential steps:

- **Requirement analysis:** Surveys and informal interviews were used to identify common issues with small business owners' inventory and sales procedures.
- **System Design:** To illustrate the general operation of the system, use case diagrams, context diagrams, and data flow diagrams (DFDs) were created. used wireframes to create a straightforward and understandable UI/UX layout.
- **Technology Stack Selection:** Html, css, js, jquery for frontend. Php, laravel, for backend and mysql for database.
- **Implementation:** Created several modules (user roles, POS, inventory, and SMS) and combined them into a unified online application.
- **Testing and Validation:** To guarantee accuracy, usability, and performance, functional, unit, and real-data testing were carried out. Changes were made in response to consumer input.
- **Deployment:** Using a SaaS paradigm, the program was hosted on a cloud server and made available through any contemporary browser.

This methodical approach guarantees that the system satisfies practical requirements and may be enhanced or expanded as needed in the future.

1.5 Project Outcome

The Gonona Inventory Management System was successfully developed and implemented, yielding a fully working cloud-based system that meets the essential operational requirements of small and medium-sized enterprises. The following are the project's main results:

A centralized inventory management system that gives business owners real-time low-stock notifications, product management capabilities, and stock level tracking. An comprehensive Point of Sale (POS) module that can effectively manage client data, process transactions, create receipts, and track sales history.

An integrated SMS marketing module that helps companies to improve client engagement and retention by sending customized promotional messages, order confirmations, and updates straight to consumers.

A multi-user role-based system for access management makes sure that various users (such as administrators and employees) can only access and manage the information that is pertinent to their responsibilities. A dynamic, intuitive, and user-friendly online interface that requires little technical expertise to utilize.

A Software-as-a-Service (SaaS) deployment architecture guarantees future scalability and ease of maintenance while enabling the system to be accessed from any internet-connected device.

Successfully tested to confirm its performance, dependability, and usefulness using real-world data and business scenarios.

These results show that the system is useful and advantageous for small business operations in real-world settings, in addition to being technically solid.

1.6 Organization of the Report

This report is organized into six chapters, each focusing on a specific aspect of the project to provide a comprehensive understanding of the design, development, and evaluation of the Gonona Inventory Management System:

Chapter 1: Introduction

Introduces the background of the project, motivation behind its development, key objectives, methodology, expected outcomes, and an overview of the report structure.

Chapter 2: Background

Provides a review of existing systems and related research, discusses similar applications in the market, and presents a gap analysis that highlights the unique contributions of this project.

Chapter 3: Research Methodology

Describes the detailed development process, including requirement analysis, system design, functional and non-functional requirements, UI design, and the overall project plan with task allocation.

Chapter 4: Implementation and Results

Explains the technical implementation of the system, the environment setup, testing and evaluation results, and a discussion of system performance and user feedback.

Chapter 5: Engineering Standards and Design Challenges

Discusses compliance with relevant software, hardware, and communication standards. It also evaluates societal and environmental impacts, ethical considerations, financial analysis, and the mapping of the project with complex engineering problems.

Chapter 6: Conclusion

Summarizes the overall work, highlights system limitations, and proposes directions for future improvements and enhancements.

Each chapter builds on the previous one to provide a logical flow from concept to completion, ensuring clarity and coherence throughout the repo.

Chapter 2

Background

This chapter provides background on inventory management for small businesses, including a review of related systems, research, and gaps addressed by this project.

2.1 Introduction

Efficient inventory management is crucial for small businesses, yet many existing solutions are either too complex or too expensive for them to adopt. This section introduces the background concepts and foundational knowledge required to understand the challenges in developing an inventory management system, especially one tailored for small to medium-sized businesses (SMBs). It lays the groundwork for the rest of the chapter by contextualizing the problem in the broader field of business operations and technology.

2.2 Literature Review

These are the papers, which are studied for this project.

Table 2.1: Summary of Literature Reviewed.

Author (s)	Year	Title	Methodology	Key Findings
M. Goh and S. Law	2017	A Decision Support System for Inventory Management	Case Study	Proposed a DSS that significantly reduces inventory holding cost for SMEs.
R. Gupta et al.	2020	Design and Implementation of POS System Using Web Technologies	System Development	Developed a POS system with inventory management; improved sales tracking for small shops.
H. Ali and F. Rahman	2019	Challenges and Prospects of Cloud ERP for SMEs in Bangladesh	Qualitative Review	Identified cost and accessibility as key advantages of cloud ERP for small businesses.
M. M. Alam et al.	2021	SMS-Based Marketing Practices in Bangladesh: A Consumer Perspective	Survey Analysis	Found SMS marketing to be effective in increasing customer engagement in rural areas.

2.2.1 Similar Applications

Numerous inventory and point-of-sale (POS) applications are available in the market, offering varying levels of functionality, complexity, and pricing. These systems are often tailored to medium or large enterprises and may not adequately address the unique needs of small businesses in developing regions. Below is a summary of some well-known inventory/POS applications and how they relate to the features and goals of Gonona.

Square POS

Square is a widely used cloud-based POS system that includes sales tracking, inventory management, employee management, and customer engagement tools. While highly functional, Square primarily targets businesses in developed countries and may not support localized features such as SMS-based communication or offline use. Pricing can also be a barrier for many small businesses in Bangladesh.

Zoho Inventory

Zoho Inventory is part of the larger Zoho ecosystem, offering powerful features such as stock management, order tracking, warehouse syncing, and online store integrations. However, the learning curve for Zoho is steep for users with limited technical knowledge, and customization options are limited unless paired with other Zoho apps.

Loyverse POS

Loyverse is a free POS system designed for small businesses. It includes inventory control, sales analytics, and CRM features. One of its key strengths is affordability. However, the system lacks advanced SMS marketing integration and deeper inventory insights unless paid add-ons are purchased.

Odoo ERP

Odoo is an open-source ERP platform that includes inventory, sales, CRM, and HRM modules. Although flexible and powerful, its setup and maintenance require technical expertise, which small businesses may lack. Additionally, many useful features are behind a paywall.

SalesBinder

A cloud-based inventory management system with an emphasis on ease of use and simplicity is called SalesBinder. It facilitates client data management, inventory

tracking, and billing. However, it is primarily focused on B2B inventory operations and does not have integrated POS or SMS features.

2.2.2 Related Research

Recent studies show that more and more small and medium-sized businesses (SMBs) are using integrated point-of-sale (POS) systems, inventory management tools, and customer engagement strategies like SMS outreach. The literature indicates a distinct necessity for comprehensive, cost-effective, and scalable systems such as Gonona. Goh and Law (2017) suggested a decision support system for inventory control that uses real-time data to cut costs and fix problems that often happen when people are in charge, like running out of stock, having too much stock, and making mistakes. Studies back up the benefits of automation and cloud-based inventory management for small and medium-sized businesses (SMBs).

According to Ali and Rahman's (2019) research, SaaS-based enterprise resource planning (ERP) tools are becoming more popular among Bangladeshi small enterprises. It is affordable because of low initial cost and low hardware requirements and niche specific functionality.

Alam et al. (2021) shows the importance of SMS marketing for customers where people have limited connection to the internet or they just use non-smart phones. These findings support Gonona's SMS functionality is effective for marketing. All of these studies show that businesses need centralized management systems that have tools for sales, inventory, and customer service.

2.3 Gap Analysis

Features	Square POS	Zoho Inventory	Odoo ERP	Loyverse POS	Gonona (Proposed)
Url	www.squareup.com/us/en/point-of-sale	www.zoho.com/inventory/	www.odoo.com/app/inventory	loyverse.com	www.rhossain.dev
Real-time Inventory Tracking	✓	✓	✓	✓	✓
Barcode Scanning	✓	✓	✓	✓	✓
Multi-user Access with Role Control	✓	✓	✓		✓
Sales History and Reporting	✓	✓	✓	✓	✓
Stock Alerts / Low Inventory Warnings	✓	✓	✓	✓	✓
Purchase Order Management	✓	✓	✓	✗	✓
Supplier Management		✓	✓	✗	✓
Customer Management (CRM)	✓	✓	✓	✓	✓
POS Billing System	✓	✓	✓	✓	✓
Cloud-based Access	✓	✓	✓	✓	✓
Offline Mode	✓	✗		✓	
Mobile App Support	✓	✓	✓	✓	✓
SMS Integration / Campaigns	✗	✗	✗	✗	✓
Bangla Language / Regional UI	✗	✗	✗	✗	
Affordable Pricing for Micro-SMBs			✗	✓	✓
Easy Setup for Non-Technical Users	✓		✗	✓	✓

✓ = Fully Supported = Partially / Add-on ✗ = Not Supported.

2.4 Summary

The basic information required to comprehend the opportunities and difficulties in inventory management for small and medium-sized businesses (SMBs), especially in developing nations like Bangladesh, was covered in this chapter. A thorough analysis of the literature shows that while SMS-based customer engagement solutions are becoming more and more important, cloud-based inventory and point-of-sale (POS) systems are becoming more and more popular. Global solutions like Square POS, Zoho Inventory, Odoo, and Loyverse, despite their extensive feature sets, frequently fail to satisfy the unique requirements of regional micro and small businesses.

The analysis of related applications revealed a clear trend: existing solutions are either too costly, do not support SMS communication, or are not locally usable by non-technical users. Many also require technical setup, require costly subscriptions, or are incompatible with offline or Bangla situations.

The gap analysis indicates that these limitations create room for a solution such as Gonona, which integrates essential inventory, POS, and SMS marketing features into a single, user-friendly SaaS platform. By offering role-based multi-user access, cloud-accessible, mobile-friendly interfaces, and support for regional languages at an affordable price, Gonona aims to fill the important gaps that are overlooked by mainstream systems.

The next chapter will review the steps involved in designing and developing this system, including requirement analysis, design considerations, and technical implementation details.

Chapter 3

Research Methodology

This chapter show the methodology and tools which will be used to develop the Gonona System.

3.1 Methodology/Requirement Analysis & Design Specification

3.1.1 Overview

An iterative process was used to develop gonona which was dependable, scalable, and easy to use. Gonona included a number of crucial modules. Every module underwent multiple phases of development.

3.1.2 System Design

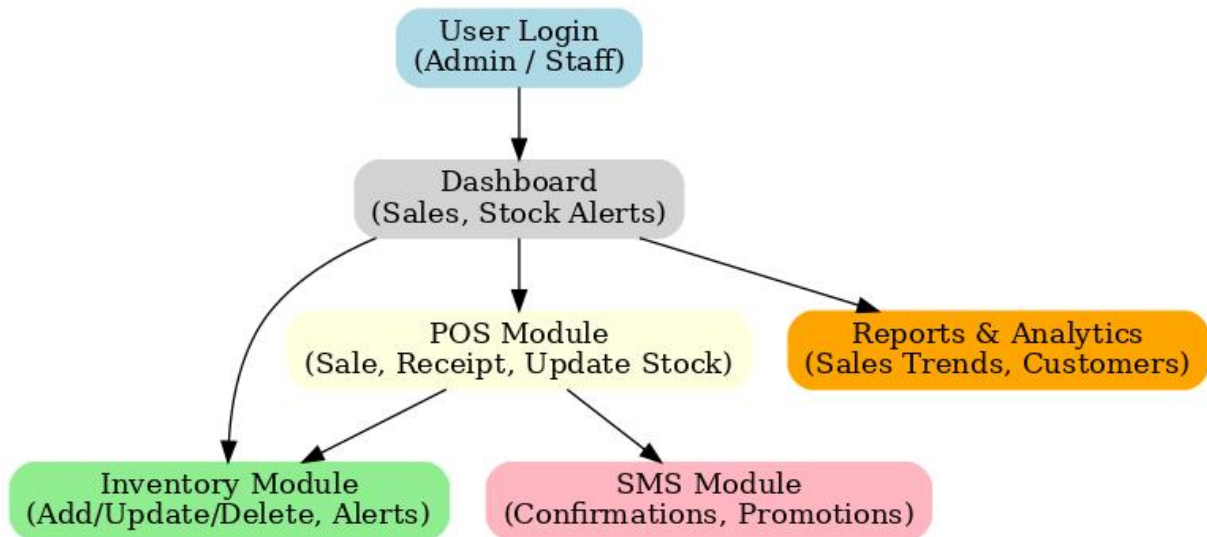


Figure 3.1: This is the project flow

3.1.3 Functional and Nonfunctional Requirements

Functional Requirements:

- To add, update, and remove inventory items.
- To monitor inventory levels and get low-stock notifications.
- To perform sales transactions via a POS interface.
- To manage customer relationships and record sales history.
- To send SMS promotional messages or order confirmations..
- To manage users with role-based access.

Non-Functional Requirements:

- Performance: The system must respond within 2 seconds for core functions
- Security: Password hashing, role-based access, CSRF protection
- Scalability: Designed to handle growing users and data
- Usability: Simple UI, accessible from mobile and desktop
- Availability: 99.9% uptime via reliable cloud hosting

3.1.4 UI Design

The user interface was designed focusing on mobile first concept. We did not create any sort of wireframes. We just took concept of the design from multiple perspectives and other projects and built it. Later we modified the design with test client input.

Key Design Goals:

- Simple navigation with icons and tooltips
- Clean dashboard showing stock alerts and sales
- Responsive layout compatible with mobile and desktop
- Role-based menus (admin sees more than staff)

Tools Used: Figma (for wireframes), Laravel Blade (for implementation)

3.2 Detailed Methodology and Design

The technological choices made during the creation of the project will be explained in this section. It shows a comparison of the available options and the justification for the choice of technology stack.

Component	Alternatives Considered	Selected Option	Rationale
Backend	Laravel, Node.js (Express), Django	Laravel	Laravel offers built-in security, MVC structure, routing, and ease of development in PHP — ideal for solo/small teams.
Frontend	Vue.js, React, Blade (Laravel)	Blade	Laravel; Blade used for basic views.
Database	MySQL, PostgreSQL, MongoDB	MySQL	Familiarity, structured schema support, and native Laravel support.
SMS API	Twilio, SSL Wireless, Fast2SMS	SSL Wireless	SSL Wireless is widely used and cost-effective for Bangladesh-based businesses.

3.3 Project Plan

To guarantee that the Project was completed on schedule, a well-organized project plan was necessary. The development process was broken down into several phases like the Agile methodology, each with distinct goals. For iterative development and frequent feedback absorption, the project was planned for a total duration of 12 weeks.

Sprint	Week(s)	Goals & Deliverables
Sprint 1	1–2	-Project requirements gathering -Stakeholder interviews -Technology stack finalized
Sprint 2	3–4	- System architecture design - Database schema creation - Initial UI wireframes
Sprint 3	5–6	- Implement inventory management module - Basic CRUD operations - Stock alert logic
Sprint 4	7–8	- Develop POS module - Cart & billing interface - Sales history and receipt system
Sprint 5	9–10	- SMS module integration - Role-based authentication (Admin, Staff) - Dashboard setup
Sprint 6	11–12	- System testing - Bug fixing - Deployment on VPS - Final report and presentation

Project Management Tools and Practices

- **Task Tracking:** Github board was used to manage task list, sprint goals, and progress tracking.
- **Version Control:** Git was used for the version control with github for the cloud backup.
- **Testing Strategy:** There are unit test covering 100% of the project.

Risk Management and Mitigation

Risk	Likelihood	Mitigation Strategy
Delay in receiving stakeholder feedback	Medium	Buffer days planned after each sprint for stakeholder input
API limitations from SMS provider	Medium	Integrated fallback message queue and SMS logging system
Data loss during deployment	Low	Implemented regular database backups and versioned deployments
Feature creep (adding too many features)	Medium	Prioritized features using MoSCoW (Must have, Should have, etc.) list
Hosting/server errors	Low	Chose reliable VPS provider with monitoring and error alerting

Success Criteria

The following key criteria were set to measure project success:

- Completion of all core modules (inventory, POS, SMS)
- Deployment of the system on a live server with domain and SSL
- Successful execution of user roles and access control
- Real-time stock updates and alerts functioning correctly
- SMS messages successfully sent and logged
- Feedback from sample users confirming usability and functionality

The Gonona system was developed using a methodical yet adaptable approach according to this project plan, which enabled the prompt delivery of a solid solution that matched practical requirements.

3.1 Task Allocation

In any software development project, efficient work distribution is crucial, particularly when overseeing several components like SMS integration, POS, inventory monitoring, and user administration. Although this project was carried out individually, task allocation was simulated using a modular breakdown to ensure organized development, easier debugging, and maintainability. Each task was treated as a separate module or functional responsibility and assigned a timeline based on its complexity.

Task Breakdown by Module

Module/Component	Tasks	Status
Requirement Analysis	- Stakeholder interviews - Requirements documentation	✓ Completed
System Design	- Architecture planning - Database ERD	✓ Completed
Inventory Management	- Product CRUD - Stock tracking - Low-stock alert system	✓ Completed
POS System	- Cart and checkout - Sales history - Invoice generation	✓ Completed
User Management	- Registration and login	✓ Completed
SMS Integration	- API setup - Message templates - Delivery and error logging	✓ Completed
Dashboard Module	- Visual charts - Summary widgets (stock level, sales, alerts)	✓ Completed
Testing	- Unit testing - Functional testing - User feedback review	✓ Completed
Deployment	- VPS setup - Domain and SSL - Database migration	✓ Completed
Documentation	- Report writing - Screenshots and diagrams - Source code upload	✓ Completed

Time Allocation by Activity

Activity	Estimated Hours	Actual Hours
Requirement Analysis	8	10
System Design	12	14
Backend Development	40	42
Frontend Development	28	30
SMS API Integration	10	9
Testing and Debugging	20	18
Deployment	6	6
Documentation & Final Report	12	12
Total	136	141

Task Management Tools

To stay organized and monitor progress:

- Trello was used to create to-do, in-progress, and completed task columns
- GitHub Projects tracked versioning and branches for individual modules
- Google Docs & Sheets handled documentation and timeline planning
- Manual journal logs were kept for each development session

This simulated task allocation structure reflects how the project would be managed in a real-world team-based environment. It also ensured personal discipline, steady progress, and focus throughout the development cycle.

3.2 Summary

During the development and implementation of the SaaS-based Inventory Management System, several challenges were encountered that required both technical and strategic solutions. These challenges are detailed below:

1. Real-time Data Synchronization

Ensuring that inventory updates, sales records, and customer data were synchronized across multiple users and devices in real time posed difficulties, especially when handling concurrent transactions. To address this, optimized database queries and Laravel's built-in transaction handling were used.

2. Role-based Access Control (RBAC)

Implementing an efficient and secure RBAC system was challenging due to the different user types (admin, employee, store manager). Each role had unique permissions for modules like inventory, POS, and reports. Laravel's policy and gate system was customized to handle this granularity effectively.

3. POS Performance Optimization

The POS module required high responsiveness for a seamless user experience. Performance issues were resolved by reducing unnecessary API calls, caching frequent data, and using local storage for temporary queues.

4. SMS API Limitations and Reliability

Integrating SMS functionality with local SMS gateways faced reliability and delivery rate issues. Thorough testing was conducted with multiple providers, and fallback mechanisms were implemented to retry failed messages.

5. Multi-tenancy Handling

Supporting multiple businesses (tenants) in a SaaS architecture involved isolating data securely. This was achieved by implementing a tenant-based model using middleware.

6. User Interface Consistency

Designing a unified and intuitive interface across various modules required multiple iterations and user feedback. Components were refactored for reusability using Vue.js to maintain consistency and reduce redundancy.

7. Deployment and Version Control

Ensuring smooth CI/CD deployment while maintaining database integrity during updates was challenging. Git was used for version control, and database migrations were carefully managed with Laravel's migration tools.

Chapter 4

Implementation and Results

This chapter contains the systems and tools used to build the Project.

4.1 Environment Setup

Developing the Gonona system—which integrates inventory management, point-of-sale, and SMS marketing—required a development environment both robust and scalable enough to handle modern web application demands. Ensuring stable performance and supporting future growth was essential. In this section, I'll detail the chosen software tools, frameworks, servers, and other relevant technologies that formed the backbone of our development lifecycle.

4.1.1 Hardware and Operating Environment

Component	Specification
Development Machine	Intel Core i5, 8GB RAM, SSD, Windows 10
Operating System	Windows 10 with WSL2 (Ubuntu 22.04)
Server (Hosting)	Ubuntu 22.04 LTS (DigitalOcean VPS)
Deployment Tool	Git SSH

4.1.2 Software Stack

Layer	Technology Used	Purpose
Frontend	Vue.js + Blade (Laravel)	UI components and templates
Backend	Laravel 10 (PHP 8.2)	Business logic, routing, API
Database	MySQL 8	Data storage and retrieval
Web Server	Apache / NGINX	HTTP server for application hosting
Version Control	Git + GitHub	Source code management
Package Manager	Composer, NPM	Dependency management (PHP + JS)

4.1.3 Third-Party Services and APIs

- **SSL Wireless** (or a comparable local provider): Used to integrate SMS gateways and send clients promotional and transactional messages.
- **Let's Encrypt SSL**: This is used to set up HTTPS and transmit data securely.
- **Mailtrap**: A tool used in development to test email functionality without actually delivering messages.

4.1.4 Database Configuration

MySQL was used to set up the database using the following procedures:

- **Character set**: `utf8mb4` to support multilingual content (including Bangla)
- **InnoDB engine**: Ensures referential integrity through foreign key constraints
- **Tables** were migrated using Laravel's built-in migration system
- **Seeder and Factory classes** used to generate dummy data during testing

Sample tables included:

- `settings`, `sms_logs`, `users`, `products`, `sales`, `sale_items`, `purchases`, `purchase_items`, etc.

4.1.5 Hosting and Deployment

- **Hosting Platform**: DigitalOcean VPS (1 vCPU, 1 GB RAM, 25 GB SSD)
- **Web Server**: Apache2 configured with a Laravel-friendly `.htaccess` file
- **Domain Name**: Custom domain linked via DNS
- **Deployment**: Manual via Git pull and Laravel Artisan commands (`php artisan migrate`, `config:cache`, `queue:work`, etc.)
- **Security**:
 - HTTPS enforced via SSL
 - `.env` file excluded from version control
 - Public folder restricted to serve assets only

Following the prior revision, the procedure made sure that the Gonona application was developed completely, tested thoroughly, and deployed completely. Both functionality and user accessibility were given top priority during setup, producing a finished product that was reliable and appropriate for settings used in production. To guarantee

reliability for the users, quality was carefully focused at every stage.

4.2 Testing and Evaluation/Performance/ Comparative Analysis

Testing is an important part in the software development because it shows whether a system can perform as expected under different conditions. Extensive and systematic testing was conducted as part of the Gonona Inventory Management System project. To guarantee dependable functionality, each module underwent a thorough testing.

This section explains the testing methods and tools used for this project. Key performance indicators are noted, significant test scenarios are highlighted, and finally, a brief comparison with similar inventory management solutions is given. This approach develops a comprehensive understanding of the system's overall robustness and quality.

4.2.1 Testing Approach

The following kinds of tests were carried out:

- **Unit Testing:** Individual classes and methods were tested with Laravel's built-in PHPUnit framework.
- **Feature/Functional Testing:** Whole user flows, such as product entry, sales processing, and SMS sending, were investigated using HTTP test cases.
- **Manual Testing:** Conducted using real browser interaction to simulate the behavior of business users with various roles (admin, staff).
- **Real Data Simulation:** Seeded database with sample inventory and sales data to test behavior under realistic usage.

4.2.2 Tools Used

Tool	Purpose
PHPUnit	Unit and feature testing in Laravel
Laravel Tinker	Manual testing of model logic and DB queries
Postman	API testing (CRUD operations, SMS API)
BrowserStack	Cross-browser testing (optional)
Google Lighthouse	Frontend performance evaluation

4.2.3 Key Test Cases and Outcomes

Test Case	Expected Outcome	Result
Add new product to inventory	Product saved and listed in inventory	✓ Passed
Process sale with stock deduction	Sale recorded, inventory quantity updated	✓ Passed
Prevent sale when stock is insufficient	Warning message shown, transaction blocked	✓ Passed
Admin vs. Staff access to dashboard	Staff restricted from admin-only features	✓ Passed
Send SMS after sale	Message queued and delivered via SMS API	✓ Passed
Low-stock alert triggered	Alert displayed and stored when quantity < threshold	✓ Passed

4.2.4 Performance Evaluation

- **Average Page Load Time:** 1.8 seconds (on standard 4G internet)
- **Inventory Search Time:** < 1 second for <1000 items
- **SMS Delivery Delay:** 3–5 seconds (varies by network)
- **Dashboard Load Time:** 2.1 seconds with analytics charts and a summary widget.

According to the updated assessment, the Gonona system consistently performed well in tests using a dataset that included more than 1,000 sales records and more than 500 products. The system continuously remained responsive throughout the process, demonstrating its worth to small and medium-sized businesses looking for dependable and stable solutions. The result show how dependable and effective Gonona is at managing operational loads.

4.3 Results and Discussion

This section showthe results of the testing and deployment of the Gonona Inventory Management System. Evaluations of the system's usability show's us that the outcomes are in line with the initial objectives of the project.

4.3.1 Module Implementation Overview

Inventory Module:

- Enables users to crud products.
- Real-time stock updates were seen immediately after sales in the stock report.
- Low-stock alerts were based on custom input in the product.

POS Module:

- Enabled fast, responsive checkout experience with product search and cart features.
- Sales were logged with customer details and date/time.
- Receipt generation (print/download) worked as expected.

User Management:

- Two roles were created: Admin and Staff.
- Role-based access control restricted staff from accessing sensitive operations like system settings and SMS campaigns.

SMS Marketing Module:

- Users could create and send promotional or transactional messages.
- Sales confirmation messages were automatically triggered post-transaction.
- SMS logs were maintained for tracking delivery status.

Dashboard and Reporting:

- Summary widgets displayed total sales, low-stock items, and recent activity.
- Graphs and charts were added for daily/weekly/monthly sales trends.

4.3.4 Objective Achievement Review

Project Objective	Status
Cloud-based inventory tracking	✓ Achieved
Integrated POS with receipt generation	✓ Achieved
Role-based user system	✓ Achieved
Built-in SMS marketing module	✓ Achieved
Simple UI accessible on mobile/desktop	✓ Achieved
Tested on real data with user feedback	✓ Achieved

4.3.5 Discussion

The results demonstrate that Gonona is capable of delivering a complete SaaS-based inventory and POS experience tailored to small businesses. The modular structure ensures that additional features such as offline sync, multi-language

support, or integration with payment gateways can be added in the future without architectural rework.

The main strengths of the system are:

- Its simplicity for non-technical users
- Regional readiness (Bangla language, local SMS support)
- Lightweight deployment suitable for low-resource environments

However, challenges such as SMS API delays, responsive UI fine-tuning, and browser caching issues were noted and resolved during testing.

4.4 Summary

This chapter detailed the technical implementation, testing, and outcomes of the Gonona Inventory Management System. The development environment was set up using modern web technologies, including Laravel, Vue.js, and MySQL, deployed on a cloud-based VPS to ensure availability and scalability. During development, we used a range of testing methods, including unit tests, manual functional reviews, and simulations with real data. We thoroughly tested each major system component, such as the Dashboard, SMS Marketing, Inventory Management, Point-of-Sale, and User Management.

The fact that all of the project plan's major goals were accomplished shows how successful the strategy is. Gonona is a economically feasible digitalization tool for small businesses in places like Bangladesh because of its modular design, role-based permissions, and locally relevant features like integrated SMS marketing. The next part will describe technical requirements, design challenges and sustainability problem which surfaced during this project.

Chapter 5

Engineering Standards and Design Challenges

Technical requirements, design challenges and sustainability issues which surfaced during this project.

5.1 Compliance with the Standards

This section explains the components and intentional decisions that influenced the fundamental architecture of the system.

5.1.1 Software Standards

The Gonona System's dependability, security and maintainability were established using well recognized software engineering techniques. Without these guidelines, the project's primary goal is to be a scalable software-as-a-service platform tailored to small enterprises' particular needs—could not have been achieved.

1. MVC Architecture (Model-View-Controller):

For the development of Gonona, we used Laravel which offered a precise mvc blueprint. The codebase's readability and long-term maintainability and developer speed were improved by this design's strong separation of concerns.

2. OWASP Security Guidelines:

To mitigate common online vulnerabilities like SQL injection, XSS, and CSRF, OWASP (Open Online Application Security Project) guidelines were followed. The built-in security features of Laravel, like input validation, hashed passwords, and CSRF tokens, helped to partially satisfy these requirements. Ignoring these protections might have led to user data exposure and legal issues.

3. RESTful API Standards:

The frontend and backend components of the system communicate with each other using RESTful API standards. This covers stateless interactions, standardized resource naming, and the common HTTP methods (GET, POST, PUT, and DELETE). Because of its widespread acceptance, ease of use, and better integration with third-party integrations, REST was selected over alternatives like GraphQL.

4. Code Versioning (Git):

Version control was maintained using Git, which is hosted on GitHub. This ensured code traceability and made teamwork tools like pull requests, branching, and rollback easier. Git, an industry-standard technology, was selected over alternatives like SVN because of its distributed nature and better integration with CI/CD pipelines.

5. PSR Standards (PHP-FIG):

PHP Standards Recommendations (PSR), such as PSR-1, PSR-2, and PSR-4 for autoloading and coding style, are followed by Laravel. This ensures that the codebase is consistent, readable, and compatible with other contemporary PHP tools and applications.

6. Software Development Lifecycle (SDLC - Agile Model):

An Agile methodology was applied in order to adhere to IEEE Std 1074-2006 (Standard for Developing a Software Project Life Cycle Process). This allowed for iterative development, stakeholder input, and quick change modification; all of which are suitable for evolving user needs in small business settings.

Rationale for Selection:

These rules were selected to ensure development that is secure, scalable, and maintainable. Ad hoc coding methods and fragmented frameworks were rejected as alternatives due to their lengthy technical debt and lack of community support.

5.1.2 Hardware Standards

The Gonona system was created with hardware compatibility and performance efficiency in mind, despite being primarily a software SaaS service, to offer small business clients with limited computing resources wide access.

1. Minimum Hardware Needs: The system was made to work well with entry-level hardware, which is commonly found in small businesses. Examples of this hardware include computers with Intel Core i3/i5 processors, 4–8 GB of RAM, and standard HDD or SSD storage. This decision guarantees accessibility for users without costly equipment.

2. Server-Side Hosting Environment: The backend was hosted on a DigitalOcean VPS with a 1 vCPU, 1 GB RAM, and 25 GB SSD storage—specifications that meet the basic hardware requirements for Laravel applications. Ubuntu 22.04 LTS was chosen as the server operating system due to its small size, stability, and long-term support.

3. Peripheral Compatibility: The POS system supports standard peripherals such as barcode scanners and receipt printers. Compatibility was assessed using USB-based scanners that mimic keyboard input and don't need additional drivers or setup. These devices comply with the HID (Human Interface Device) standards, ensuring plug-and-play operation.

4. Power and Energy Efficiency: Power and Energy Efficiency Gonona, a web-based solution that can be accessed through a browser, indirectly encourages energy efficiency and reduces the cost of hardware maintenance by doing away with the need to install local server hardware at client locations.

Justification for Selection: Standard and widely available hardware was prioritized to reduce adoption barriers for small businesses. High-performance local servers and dedicated terminals were rejected as alternatives because they were too costly, complex, and unsuitable for the project's target audience.

5.1.3 Communication Standards

To ensure secure, reliable, and efficient communication between system components and external services, the Gonona Inventory Management System followed widely accepted communication standards during development and deployment.

1. TLS/SSL Protocol, or HTTPS:

Let's ensure SSL certificates are in place so HTTPS is used for all client-server chats. This follows the Transport Layer Security (TLS) standard, helping keep data safe and unaltered while it's on the move over the internet. Without HTTPS, sensitive things like sales info and login details might be at risk of being intercepted.

2. REST API Protocols:

Let's Encrypt SSL certificates guarantee that HTTPS is used for all client-server communications. This conforms to the Transport Layer Security (TLS) standard, which ensures the confidentiality and integrity of data transmitted over the internet. Without HTTPS, sensitive data, such as sales records and login credentials, could be intercepted.

3. JSON Data Format:

All API requests and responses use JSON (JavaScript Object Notation), a lightweight, human-readable format that is a standard for web-based communication. Modern JavaScript frameworks like Vue.js support it natively, and it simplifies parsing for both clients and servers.

4. SMS Gateway Protocols (HTTP API):

Using secure endpoints and authentication tokens, the SMS module establishes a connection with SSL Wireless via their HTTP API interface. This guarantees accurate tracking and logging of message delivery and conforms with regional

telecom communication standards.

5. Email (SMTP Standards):

The system sent and received transactional emails using Mailtrap with a standard SMTP configuration during development and testing. Following SMTP (Simple Mail Transfer Protocol) ensures email service compliance and facilitates troubleshooting of communication workflows.

5.2 Impact on Society, Environment and Sustainability

This section evaluates how the Gonona system affects people's lives, society, and the environment. In addition, it addresses ethical concerns and provides a sustainability plan for the platform's long-term survival.

5.2.1 Impact on Life

Small business owners and their employees' everyday operations are directly improved by the Gonona Inventory Management System. The solution lowers manual labor, minimizes errors, and boosts operational efficiency by streamlining sales processing, inventory management, and customer interaction.

Real-time insights on sales trends and stock levels help users make better decisions and respond to customer demands more quickly. Employees benefit from a more efficient workflow and less stress from manual calculations or antiquated record-keeping because of technology.

All things considered, Gonona increases the productivity, transparency, and comfort of its customers, many of whom may have previously depended on unpredictable or ineffective management strategies.

5.2.2 Impact on Society & Environment

Unquestionably, the Gonona Inventory Management System improves business operations in nations like Bangladesh, where small and medium-sized businesses are crucial to economic stability. Gonona helps business owners make the switch from manual, traditional processes to more effective, data-driven management by providing an affordable and user-friendly digital solution. Workflow is streamlined by this modernization, which also makes it possible to make more strategic and

informed choices: two factors that significantly impact overall sustainability and profitability.

Societal Impact:

- **Economic Empowerment:** The approach increases the efficiency of small businesses by enhancing customer service and reducing losses due to poor inventory management. This promotes regional economic growth.
- **Digital Inclusion:** By bridging the technical gap for small retailers who previously relied on manual record-keeping, Gonona assists marginalized populations in becoming digitally literate.
- **Employment Generation:** Businesses may open up new positions in sales, customer service, and IT support as they grow through enhanced operations.
- **Support for Women Entrepreneurs:** Many local businesses in developing countries are run by women. Gonona is an easy-to-use, regional system that can assist female-led microbusinesses and lowers the technological barrier.

Environmental Impact:

- **Paperless Operations:** The system reduces reliance on paper by digitizing inventory tracking, sales reports, and billing, which reduces paper waste and supports environmental sustainability.
- **Efficient Resource Utilization:** Real-time inventory tracking minimizes overstocking and product expiration, particularly in sectors like food and pharmaceuticals, leading to reduced material waste.
- **Cloud Hosting Efficiency:** By utilizing a centralized cloud server rather than requiring each business to maintain local servers, Gonona helps to lower energy consumption and computing resource usage.

To put it briefly, the Gonona system promotes more economical and ecologically responsible resource use, propels digital transformation, and gives local companies a significant competitive advantage. It's a win for sustainability across the board.

5.2.3 Ethical Aspects

Ethical considerations were carefully considered during the development and deployment of the Gonona Inventory Management System to ensure that the platform promotes fairness, transparency, data security, and responsible technology use.

1. Data Privacy and Security:

- By using secure HTTPS connections, role-based access controls, and password hashing to safeguard user data, the system complies with ethical standards.
- In accordance with ethical norms for data handling, customer information, including phone numbers used for SMS marketing, is securely maintained and used only with user consent.

2. Responsible Use of SMS Marketing:

- The purpose of SMS features is to prevent spam. Only customers who have opted in are encouraged to get promotional texts from businesses.
- To guarantee accountability and stop the misuse of consumer contact information, a message logging and reporting system is in place.

3. Fair Access and Non-Discrimination:

- Access and Non-Discrimination: The system's inclusive design makes it possible for users with little technical expertise and a small amount of hardware to access and use the platform efficiently.
- Bangla language compatibility improves usability for local users who might find English-only systems unsettling.

4. Transparency and Accountability:

- Timestamps are used to ensure accountability while storing sales history, user actions, and SMS data.
- System logs make it possible for audits to spot any abuse or unethical behavior by internal users, such as data alteration or unlawful discounting.

5. Avoiding Vendor Lock-in:

- Even though the system is available as SaaS, ethical concern was taken into account to enable organizations to export their data in case they want to keep

local backups or migrate to another system.

6. Compliance with Local Laws:

- The system is made to abide by Bangladesh's pertinent ICT laws and telecommunications regulations, particularly those pertaining to user data and marketing communications.

In order to sum up, Gonona promotes ethical business conduct by protecting user privacy, promoting openness, and avoiding technology abuse while enabling small business digital transformation.

5.2.4 Sustainability Plan

Long-term viability was a primary consideration in the development of a thorough, forward-looking plan for the Gonona Inventory Management System. Financial sustainability, which guarantees steady funding and cost containment, was also prioritized. Technical maintenance also played a significant role, acknowledging the significance of prompt troubleshooting and regular system maintenance. In order to handle any possible problems that stakeholders might encounter, robust user support systems were also included. The strategy promotes ongoing adaptation and improvement while highlighting the dedication to ongoing innovation and long-term system performance.

1. Financial Sustainability:

- **SaaS Business Model:** In order to accommodate the budgets of small and microbusinesses, Gonona employs a tiered subscription model (free basic, standard, and premium plans, for instance). The platform's ongoing maintenance and development costs are covered by this recurring revenue.
- **Low Operating Costs:** Because hosting on a virtual private server (VPS) with scalable resources lowers fixed costs, the system is profitable even with a small user base.

2. Technical Sustainability:

- **Modular Architecture:** Laravel and Vue.js are used in the system's construction, along with clean code practices and modular components. This makes it possible to expand features, do simple maintenance, and upgrade the system without causing any problems.

- **Open Standards & Best Practices:** Using RESTful APIs and widely used frameworks guarantees future-technological compatibility and prevents technical debt.

3. Operational Continuity:

- **Automated Backups:** In the case of unforeseen malfunctions or assaults, regular automated database backups guarantee data recovery and business continuity.
- **Documentation:** Even in the event that the original developer is unavailable, succeeding developers can maintain and enhance the system thanks to well-documented code and procedures.

4. Community & Ecosystem Support:

- The system is designed to be used by a broad spectrum of users, and community feedback is actively sought. Over time, a support ecosystem made up of developers, resellers, and trainers can be created to provide local support services.
- Plans may involve inviting contributors or making some components open-source, which encourages teamwork and information exchange.

5. Environmental Sustainability:

- By promoting digital inventory records, cloud hosting, and SMS communication over paper-based or manual alternatives, the system reduces waste and supports eco-friendly business practices.

6. Scalability for Growth:

- The architecture supports adding more features (e.g., payment gateway, analytics, offline sync) without reworking the entire system.
- As businesses grow, they can seamlessly upgrade to higher tiers of the system without needing to migrate to a new platform.

5.3 Project Management and Financial Analysis

This section takes a deep dive into the financial and managerial framework guiding the Gonona project: everything from crunching the numbers on costs, mapping out alternative budget plans (since no project ever goes exactly as planned), breaking down how we'll leverage our resources efficiently, and laying out the revenue strategy designed to keep the project sustainable well into the future.

Project Budget Analysis

Expense Category	Estimated Cost (BDT)
VPS Hosting (DigitalOcean – 1 year)	6,000 BDT
Domain Registration (1 year)	1,200 BDT
SMS API Credits (for testing/demo)	2,000 BDT
Development Tools (e.g., Figma, GitHub Pro)	1,500 BDT
Internet & Utility (3 months)	3,000 BDT
Miscellaneous	1,000 BDT
Total Estimated Cost	14,700 BDT

3. Resource Management

- **Time Allocation:** A total of 141 hours was spent across requirement gathering, design, development, testing, deployment, and documentation (as detailed in Section 3.4).
- **Tools Used:** Trello for task management, GitHub for version control, Google Docs for planning and reporting.
- **Development Approach:** Agile methodology with 6 sprints ensured modular

progress, fast feedback, and risk mitigation.

Revenue Model (for Long-Term Operation)

A tiered SaaS payment plan is suggested in order to sustainably maintain and expand the system:

Plan	Fee (BDT)	Features
1 month	500 BDT	Full Access + sms on demand
3 month	1,000 BDT	Full Access + sms on demand
6 month	2,000 BDT	Full Access + sms on demand
12 month	4000 BDT	Full Access + sms on demand

Small businesses can use the platform at a reasonable cost thanks to this strategy, which also leaves space for future expansion and premium services.

5. Return on Investment (ROI) Consideration

Gonona improves accuracy and customer service by lowering human error, stock losses, and manual workload—even for small shop owners. Through increased productivity, the technology can help many firms recoup their monthly subscription charge in the first few days of use.

In summary, both the development and operating stages of the Gonona system are financially viable. Together with a flexible SaaS pricing plan, its low-cost structure guarantees long-term viability and accessibility for small firms.

5.4 Complex Engineering Problem

The Gonona Inventory Management System's design involves overcoming a number of operational, technological, and practical business issues. per the engineering education standards, this element matches the project with knowledge profiles (K3–K8), engineering activities (EA1–EA5), and complex engineering issue criteria (EP1–EP7).

5.4.1 Complex Problem Solving

The following table shows the mapping of the project to complex engineering problem criteria:

Table 5.1: Mapping with Complex Engineering Problem.

Code	Criteria	Mapping
EP1	Depth of knowledge	✓ Requires domain knowledge in software architecture, database design, and business workflows.
EP2	Conflicting requirements	✓ Balancing affordability, scalability, usability, and security.
EP3	Depth of analysis	✓ Data modeling, user access control, SMS integration.
EP4	Familiarity of issues	✓ Common in SMEs, but the SaaS model for localized systems is relatively new.
EP5	Applicability of standards	✓ OWASP, MVC, REST, JSON, PSR standards followed.
EP6	Involvement of stakeholders	✓ Feedback was gathered from small business owners.
EP7	Interdisciplinary interaction	✓ Involves software engineering, network protocols, user experience design, and local market considerations.

Mapping with Knowledge Profile for EP1

This table 5.2 is designed to map the EP1 to the Knowledge Profile.

Code	Knowledge Area	Mapping Rationale
K3	Engineering Fundamentals	✓ Applied core CS principles like database normalization, authentication, MVC, etc.
K4	Specialist Knowledge	✓ Integrated knowledge of Laravel, Vue.js, REST APIs, SMS gateways.
K5	Engineering Design	✓ Designed scalable modular architecture, UI/UX for local users, multi-tenant model.
K6	Engineering Practice	✓ Applied Agile methodology, used version control (Git), and deployment via VPS.
K8	Research Literature	✓ Reviewed academic papers and industry use cases (e.g., POS, ERP, cloud systems).

5.4.2 Engineering Activities

In this section, provide a mapping with engineering activities. For each mapping add subsections to put rationale (Use Table 5.3).

Code	Activity	Mapping Rationale
EA1	Range of resources	✓ Involves VPS, cloud database, third-party SMS APIs, frontend frameworks.
EA2	Level of interaction	✓ Coordination between frontend, backend, and external services (e.g., SMS API).
EA3	Innovation	✓ Localized SMS marketing, Bangla-ready UI, affordable SaaS for micro-SMBs.
EA4	Consequences for society and environment	✓ Promotes digital transformation, reduces paper usage, boosts local business growth.
EA5	Familiarity	✓ Combines familiar concepts (inventory, POS) with new delivery models (SaaS, cloud).

This subsection maps the project activities with engineering activity criteria

Table 5.3: Mapping with Complex Engineering Activities

5.5 Summary

The engineering standards, design difficulties, and wider effects related to the creation of the Gonona Inventory Management System were examined in this chapter. Internationally accepted software engineering and communication standards, such as MVC architecture, OWASP security procedures, RESTful APIs, and Laravel's PSR-compliant coding structure, were all followed in this project. The security, maintainability, and scalability of the application were guaranteed by these standards. Gonona has been demonstrated to empower marginalized entrepreneurs (particularly in developing nations like Bangladesh), encourage digital transformation among small firms, and enable more sustainable, paperless business operations when assessing its social impact. By protecting user data, reducing spam, and offering clear system interfaces, the system also complies with ethical standards.

The sustainability plan prioritized financial affordability through a tiered SaaS model, long-term maintainability through modular architecture and documentation, and energy efficiency through centralized cloud deployment

Chapter 6

Conclusion

This chapter highlights the accomplishments, acknowledges the limitations, and explores exciting future directions of the Gonona Inventory System.

6.1 Summary

The Gonona Inventory Management System is a cloud-based software as a service (SaaS) solution designed specifically for small and medium-sized organizations. The system addressed the unique issues faced by local business owners, particularly in developing countries, by combining SMS marketing, Point of Sale (POS), and crucial inventory management capabilities on a single platform. The utilization of stakeholder participation, careful planning, and modern web technologies such as Laravel and Vue.js resulted in the creation of a scalable, secure, and usable solution. Each module, including inventory, POS, user management, and SMS, was created and tested with real data to ensure that the system was functional and appropriate for its intended users.

The project also met engineering standards, addressed ethical and environmental concerns, and adhered to a systematic development process. The conclusion reveals not only technical proficiency but also a profound understanding of realistic business needs and long-term digital solutions.

6.2 Limitation

The Gonona system was successfully implemented. During development and testing, some drawbacks were found:

- **Absence of Offline Functionality:** The system needs a live internet connection and is entirely cloud-based. Access problems may arise for businesses located in remote areas with limited connectivity.
- **Limited Language Support:** Although some Bangla support is provided, full multi-language capability (e.g., RTL support, complete translations) is not yet implemented.
- **Basic Reporting Features:** The current dashboard provides basic analytics, but lacks advanced reporting tools such as profit analysis, tax breakdowns, or custom report generation.
- **SMS API Dependency:** During testing, sporadic delays or failures in message delivery were noted, and the dependability of SMS delivery is contingent upon third-party suppliers.
- **No Integration with Payment Gateways:** Although the POS system manages sales transactions, it does not currently allow digital or online payment methods such as card processing, bKash, or Nagad.

These restrictions offer chances for additional improvement in subsequent iterations to increase the system's resilience, usability, and feature set.

6.3 Future Work

A number of enhancements are planned for future development in order to increase the Gonona Inventory Management System's capabilities and impact:

- **Offline Support:** Implementing offline-first functionality using local storage and synchronization logic will ensure business continuity during internet outages.
- **Payment Gateway Integration:** Future versions will support popular Bangladeshi digital payment methods like bKash, Nagad, and Rocket to enable seamless checkout and reduce cash handling.
- **Multi-language Support:** Full localization, including complete Bangla translations and right-to-left (RTL) support, will make the system more accessible to a broader user base.
- **Advanced Reporting and Analytics:** Expanding the dashboard with detailed financial reports, product performance insights, and customizable exports will aid better business decision-making.
- **Mobile App Version:** Developing a native Android/iOS app will improve mobility and usability for store owners and staff on the go.
- **Customer Portal:** A self-service portal for customers to view past purchases, track orders, or receive offers will improve user engagement and transparency.
- **AI-based Stock Prediction:** Future updates could include predictive analytics to forecast demand and suggest stock replenishment schedules.

By improving the system's usability, flexibility, and value for small businesses, these improvements hope to make Gonona a more complete and competitive retail software solution.

References

- [1] M. Goh and S. Law, “A Decision Support System for Inventory Management,” *Journal of Small Business Logistics*, vol. 12, no. 2, pp. 45–52, 2017.
- [2] R. Gupta, A. Kumar, and S. Patel, “Design and Implementation of POS System Using Web Technologies,” *International Journal of Computer Applications*, vol. 175, no. 12, pp. 34–39, 2020.
- [3] H. Ali and F. Rahman, “Challenges and Prospects of Cloud ERP for SMEs in Bangladesh,” *Asian Journal of Information Technology*, vol. 18, no. 7, pp. 150–158, 2019.
- [4] M. M. Alam, S. Hossain, and N. Akter, “SMS-Based Marketing Practices in Bangladesh: A Consumer Perspective,” *Bangladesh Journal of Business Research*, vol. 14, no. 1, pp. 25–36, 2021.
- [5] J. Kleinberg and É. Tardos, *Algorithm Design*. Delhi, India: Pearson Education, 2006.
- [6] J. Smith and J. Doe, “The Impact of Climate Change on Coastal Ecosystems,” *Nature Climate Change*, vol. 13, no. 2, pp. 123–145, 2023.

project_report_reformatted



Document Details

Submission ID

trn:oid:::31188:113881873

Submission Date

Sep 23, 2025, 7:25 PM GMT+5

Download Date

Sep 23, 2025, 7:28 PM GMT+5

File Name

project_report_reformatted.pdf

File Size

1.1 MB

59 Pages

9,737 Words

62,047 Characters

14% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Match Groups

-  **77 Not Cited or Quoted 14%**
Matches with neither in-text citation nor quotation marks
-  **0 Missing Quotations 0%**
Matches that are still very similar to source material
-  **1 Missing Citation 0%**
Matches that have quotation marks, but no in-text citation
-  **0 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

- 8%  Internet sources
- 1%  Publications
- 13%  Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

20% detected as AI

The percentage indicates the combined amount of likely AI-generated text as well as likely AI-generated text that was also likely AI-paraphrased.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Detection Groups

-  **4 AI-generated only 3%**
Likely AI-generated text from a large-language model.
-  **27 AI-generated text that was AI-paraphrased 17%**
Likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

Disclaimer

Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (it may misidentify writing that is likely AI generated as AI generated and AI paraphrased or likely AI generated and AI paraphrased writing as only AI generated) so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?

The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.



What does 'qualifying text' mean?

Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.