CycleZen: A Digital Transformation in Bicycle Retail Management

Submitted By

Mahedi Nur Tamim

Student Id:221-35-1003

Supervisor Name: Ms. Nadira Islam

Designation:

Lecturer, Department of Software Engineering
Daffodil International University.

This project report has been submitted in fulfilment of the requirements for the degree of **Bachelor of Science in Software Engineering**

DAFFODIL INTERNATIONAL UNIVERSITY

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name	:	Mahedi Nur Tamim
Date of Birth	:	22 July 2001
Title	:	Cyclezen
Academic Session	:	2022

I declare that this thesis is classified as:

☐	CONFIDENTIAL	(Contains confidential information under the Official Secret Act 1997)*
☐	RESTRICTED	(Contains restricted information as specified by the organization where research was done)*
☐	OPEN ACCESS	I agree that my project to be published as online open access (Full Text)

I acknowledge that Daffodil International University reserves the following rights:

1. The Project is the Property of Daffodil International University.
2. The Library of Daffodil International University has the right to make copies of the Project for the purpose of research only.
3. The Library of Daffodil International University has the right to make copies of the Project for academic exchange.

Certified by:

(Student's Signature)

221-35-1003

Student ID

Date:27.12.2025

(Supervisor's Signature)

Ms. Nadira Islam

Name of Supervisor
Date: 27.12.2025

APPROVAL

This Project titled on "**Personal Finane Cyclezen**", submitted by **Mahedi Nur Tamim (ID: 221-35-1003)** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

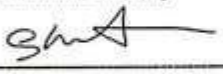
BOARD OF EXAMINERS

Chairman

Dr. Imran Mahmud
Professor & Head
Department of Software Engineering
Faculty of Science and Information Technology Daffodil International University

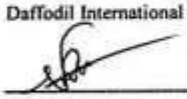
Internal Examiner 1

Afsana Begum
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University



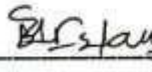
Internal Examiner 2

Md. Shohel Arman
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University



Internal Examiner 3

Nadira Islam
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University



External Examiner

Md Manowarul Islam
Professor
Department of Computer Science and Engineering
Jagannath University, Bangladesh

Department of Software Engineering

Faculty of Science and Information Technology

Supervisor Approval Form

Fall 2025	B.Sc. In SWE	Campus: DSC

Student Name	Student ID
Mahedi Nur Tamim	221-35-1003

Project/Thesis Information	
Project Title	Cyclezen
Type of	A digital transformation in Bicycle Management

Supervisor information	
Supervisor Name	MS.Nadira Ialam
Supervisor Initial	NIR
Completed Credit till now	129
How many credits in this semester	16
Amount (Due)	-895.00
Supervisor Consent	☐ Yes ☐ No

Supervisor Signature : 

SUPERVISOR'S DECLARATION

I hereby declare that I have checked this project and in my opinion, this project is adequate in terms of scope and quality for the award of the degree of Bachelor of Science.

_____

(Supervisor's Signature)

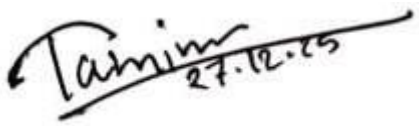
Full Name : **Ms. Nadira Islam**

Position : Lecturer, Department of Software Engineering
Daffodil International University.

Date: 27.12.2025

STUDENT'S DECLARATION

I hereby declare that the work in this project is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Daffodil International University or any other institution.



_____

(Student's Signature)

Full Name : Mahedi Nur Tamim

ID Number : 221-35-1003

Date : 27.12.2025

CycleZen: A Digital Transformation in Bicycle Retail Management.

Mahedi Nur Tamim

Project submitted in fulfillment of the requirements
for the award of the degree of
Bachelor of Science

Department of Software Engineering

DAFFODIL INTERNATIONAL UNIVERSITY

NOVEMBER 2025

ACKNOWLEDGEMENTS

First and foremost, all praise to **Almighty ALLAH** for providing the strength, patience, and inspiration to complete this project successfully.

I would like to express my sincere gratitude to my respected supervisor, **Ms. Nadira Islam**, Lecturer, Department of Software Engineering, Daffodil International University, for her continuous guidance, encouragement, and valuable feedback throughout the project period. Her mentorship helped shape the project into a professional and research-oriented work.

Special thanks to the **Department of Software Engineering** and all faculty members for providing the academic foundation, laboratory resources, and technical support essential to this work. I am also grateful to my teammates and peers whose collaboration, brainstorming sessions, and testing efforts contributed immensely to the success of *CycleZen*.

Finally, heartfelt appreciation goes to my **family and friends** for their unwavering support, motivation, and understanding during the long development and documentation process. Their belief in me made this journey possible.

ABSTRACT

The digital commerce has emerged quickly, transforming the way retail businesses are conducted in the XXI century. In this change, CycleZen: A Digital Transformation in Bicycle Retail Management presents an entire web-based solution aimed at automated sale and inventory management and order-tracking of bicycle shops. This project seeks to fill the gaps of inefficiencies in the operations of small-/ mid-sized retailers in Bangladesh through the manual system of record keeping, insufficient customer outreach and unstructured operations of the organizations.

CycleZen is developed on the MERN stack (MongoDB, Express.js, React, Node.js) with TypeScript integration, which allows offering customers an interface with responsiveness and administrators a secure online shopping platform and a user management dashboard that allow managing the users, products, and revenues. The system uses JWT- based authentication, role-based access control, and CI/CD deployment with Vercel as the payment processing provider, and ShurjoPay as the data source is ShurjoPay to guarantee the scalability and integrity of the data. The reliability and maintainability of the platform was confirmed by the comprehensive software testing, such as functional tests, performance tests and security tests.

Not only is CycleZen a working prototype of a digital bicycle store, but also a case study of a contemporary approach to software engineering that includes Agile development, modeling with the use of UML, and deployment to a cloud. Its design enables its further extension like mobile apps, AI-based recommendation systems, and analytics dashboards. This project will be beneficial both to the overall vision of Bangladesh of sustainable innovation and green mobility by encouraging sustainable transportation and digital entrepreneurship.

Keywords: CycleZen, MERN Stack, E-Commerce, Software Engineering, ShurjoPay, Digital Transformation, Bicycle Retail.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

JWT	JSON Web Token
UI/UX	User Interface/User Experience
MERN	MongoDB, Express.js, React.js, Node.js (technology stack)
CICD	Continuous Integration and Continuous Deployment
API	Application Programming Interface
AI	Artificial Intelligence
OAuth	Open Authorization
HSTS	HTTPS Strict Transport Security
CSP	Content Security Policy

CHAPTER 1: INTRODUCTION

1.1 Background

In recent years, the trend of digital commerce has taken the world by storm, drastically changing the face of the retail industry. Consumers are more and more inclined towards easy to use technology-based purchasing platforms that offer variety, security, and real- time accessibility. The bicycle industry in particular has seen exponential growth thanks to an increased environmental awareness, congested cities, and increased lifestyles. However, a significant number of small and mid-level bicycle retailers in Bangladesh and abroad still use traditional ways of selling their products, which suffer from inefficiency in managing inventory, customer outreach, and order management.

CycleZen has been created as an all-in-one bicycle store management system, which fills this gap in technology. It offers a unified web-based platform for customers to browse, order and pay for bicycles and accessories and for administrators to efficiently manage inventory, users and revenue data. By using React.js for the frontend, Express.js for the backend, and MongoDB as the database layer all of which ensures speed, scalability, and security of the system.

CycleZen has been developed as a comprehensive online bicycle store management system that bridges this technological gap. It provides a centralized web-based platform that allows customers to browse, order, and pay for bicycles and accessories while enabling administrators to efficiently manage inventory, users, and revenue data. By integrating React.js on the frontend, Express.js on the backend, and MongoDB as the database layer, the system ensures speed, scalability, and security.

1.1.1 Context and Relevance

The project is in line with the growing digital transformation in the retail sector. It offers a scalable e-commerce architecture, which can be applied to other product domains except for bicycles. With the growing usages of Smartphones and Digital Payment Systems in Bangladesh, CycleZen is fulfilling the need of modern, accessible and environmental friendly transportation retail system. It also helps in promoting sustainability initiatives by promoting eco-friendly travel options.

1.1.2 Problem Identification

Traditional bicycle stores have a number of challenges in their operations like poor inventory tracking and manual billing errors, limited marketing reach and the absence of secure online payment gateways. These limitations not only limit the efficiency of sales, but also the customer satisfaction as well. Moreover, without proper analytics and user management, store owners are not in a position to easily make informed decisions on the business. Therefore, there is a need for a structured digital platform that helps to automate operations, manage data securely and provide a seamless user experience.

1.1.3 Purpose and Justification

The idea of this project is to design and implement reliable, easy to use and maintain online system that will take care of the entire process of bicycle and accessories selling. CycleZen is attempting to reduce the level of manual errors, increase convenience for the customer and provide the administrator with a tool for analysing revenue trends. The system has a role based access control and storage of pay is secured. In addition to this, the modular architecture makes it easy to integrate this data e.g. with AI-based product recommendations, personalised marketing and predictive stock analysis in the future.

1.1.4 Scope

The scope of this project is all the web-based functions that are required for an online bicycle store -- user registration & authentication, product catalog management, order placement, payment processing, and inventory. The project also contains an administrative dashboard for users, orders and revenue statistics. Although the current system is focused on the access of web, the backend system architecture is designed in a future-proof manner such that mobile applications are supported or more advanced analytics features are incorporated.

Table 1

User Class	Age Range	Computer Experience	Education	Language Skills	System Use	Frequency
Customer	18–50	Basic+	Secondary	English	Purchase	Weekly
Admin	22–45	High	Bachelor	English	Manage	Daily
Shop Owner	30–55	Moderate	Any	English	Inventory	Monthly

1.2 Project Planning and Initiation

Phase 1: Preliminary study & Project scope definition

The initial phase was to identify the business need of a streamlined and automated bike sales platform. A problem analysis was done to understand the limitations in traditional sales and record keeping by manual means. The goal of the project was insisted as development of a responsive and safe web system to connect customers and bicycle retailers efficiently.

Phase 2: Feasibility Analysis in the market

Market analysis showed growing demand of online platform of purchasing vehicle and bicycles in Bangladesh. Few local competitors have integrated systems with real

payment gateways & the inventory control. Hence, CycleZen moves ahead as one of the competitive solutions with modern UI/UX design and local payment gateway integration with the help of ShurjoPay.

Phase 3: Analysis of technical feasibility

CycleZen is designed with MERN stack technologies i.e. React. js, Node. js (Express), Mongo db and Type Script in order to provide maintainability and performance. The system is hosted on Vercel for frontend & backend deployment for global access & version control update. The stack chosen makes it possible to assure cross-platform compatibility and scalability.

Phase 4: The Financial Feasibility Analysis

The development costs of the project had been kept to a minimum using open source frameworks and cloud deployment services. The potential financial benefit that can be obtained is in terms of cost savings by automating and extending the customer reach and ability to integrate marketing analytics. However, commercial deployment in the future may collect revenue as commission-based selling, or premium vendor subscription.

1.3 Target User Profile and Tentative Elicitation Process

CycleZen is aimed at two categories of users:

Customers: They are people who wish to make a purchase of bicycles or accessories using the sales system on the internet. Administrators: Store managers that are responsible for managing the product listings, order management and revenue management.
User requirements were acquired through informal interviews and online surveys to local bicycle shop owners and potential bicycle shop customers. Feedback indicated the need for a simplified product listings, secure payments and mobile friendly interfaces. The elicitation process also included observation of existing e commerce sites in order to identify the patterns of usability and customer expectations.

1.4 Project Block Diagram

The architecture of the system is divided into 3 layers:

Presentation Layer (Frontend): Developed in React.js and TypeScript and is responsible for the rendering of the user interface and offering the responsive design.

Application Layer (Backend): Built using the Express.JS framework this layer manages all the business logic, performs routing of requests and also performs the communication between the frontend and database.

Database Layer Managed with MongoDB Storing all user and product and order related data.(Figure 1: System Block Diagram - will be inserted in final .docx)1.5 System Requirements

1.5 System Requirements

1.5.1 Hardware Requirements

Processor: Intel Core i5 or equivalent
RAM: Minimum 8 GB

1.5.2 Software Requirements

Operating System: Windows / macOS / Linux
Frontend: React.js (TypeScript, Vite) Backend:
Node.js (Express.js)
Database: MongoDB
Tools: Postman, Git, ESLint, Vercel, Tailwind CSS Browser:
Google Chrome / Mozilla Firefox

1.5.3 Constraints and Dependencies

Dependence upon Continuous Internet Connection.

Third party payment gateway integration (ShurjoPay). Data hosting and deployment is dependent upon the uptime of Vercel.

Require user authentication by using the help of JSON Web Token (JWT).

1.6 Project Scheduling

The development process was implemented using an Agile methodology with iterative sprints.
Sprint 1: (Week 1-3): Requirements gathering, system design and database schema creation.
Sprint 2: (Week 4-6): Frontend Interface and User Authentication Module. Sprint 3: (Week 7 - 9): Integrating Product and order management. Sprint 4: (Week 10-12): Setting up and testing payment gateway Sprint 5: (Week 13-14): Deployment and documentation.

1.7 Summary

This chapter introduced the CycleZen project, defining the background, objectives, and system overview of the project. It identified the problems of traditional retail systems and made a case for a digital bicycle store management solution. The chapter also contained feasibility, target users, system requirements, and development schedule. In summary, CycleZen creates an effective base for an efficient, secure and scalable e-commerce application that is compatible with current technological and market needs.

CHAPTER 2: DESIGN AND IMPLEMENTATION

2.1 Introduction

This chapter describes a means of an overview of design and implementation strategies of the development of CycleZen system. The design part is aimed on the difficulty of translating the project needs into a formal and logical system modeling, by adding stereotyped diagrams of the UML's standards. This includes identification of system functionalities, interaction of the users, data flows and relationships between different components of the system. Implementation details cover the technology and frameworks that are being used to create the different layers of the system-- frontend, backend and database. Together these design models provide for scalability, maintainability and user satisfaction.

2.2 Functional Requirements

Functional requirements define the particular services, operations, and behaviors that must be provided by the CycleZen system in order to accomplish its objectives.

Table 2

FR Code	Functionality	Description	Stakeholders
FR01	User Registration	Customers and admins can register using their email and password to create an account.	Customer, Admin
FR02	User Login	The system authenticates users using JWT-based login for secure access.	Customer, Admin
FR03	View Bicycles	Customers can browse the available bicycles categorized by brand and type.	Customer
FR04	Product Management	Admin can create, update, or delete bicycle records and manage stock.	Admin

FR05	Place Orders	Customers can add products to the cart and confirm orders.	Customer
FR06	Payment Processing	The system allows users to make payments securely using ShurjoPay integration.	Customer
FR07	Order Management	Admin can view all orders, update order status, and track payments.	Admin
FR08	Revenue Calculation	System automatically calculates revenue based on completed orders.	Admin
FR09	User Profile Management	Users can update personal information, view past orders, and manage passwords.	Customer
FR10	Dashboard Analytics	Admin dashboard displays total users, sales, and stock reports.	Admin

2.3 Non-Functional Requirements

2.3.1 Performance

CycleZen offers higher response time (< 2 seconds per request) through optimization offered by API and database indices. Frontend performance is optimised with the help of lazy loading and caching methods. The system can handle concurrent user request up to 500 active session in normal user's load.

2.3.2 Reliability

The system design is to be very reliable with high error handling, validation layers and redundant database backup. Each transaction (and mainly payment and order creating) is validated at different level in order to avoid inconsistency of data.

2.3.3 Portability

The web application runs on any operating system supporting modern browsers. The backend and database can be deployed on cloud platforms such as Vercel, AWS, or Render without modification.

2.3.4 Security

Security is maintained through JWT authentication, HTTPS encryption, and role-based access control. Sensitive user data such as passwords and payment tokens are encrypted using hashing algorithms.

2.3.5 Maintainability

The modular architecture using the MERN stack ensures that updates and new features (e.g., AI-based recommendations) can be added easily without affecting existing modules.

2.4 Object-Oriented System Design Using UML

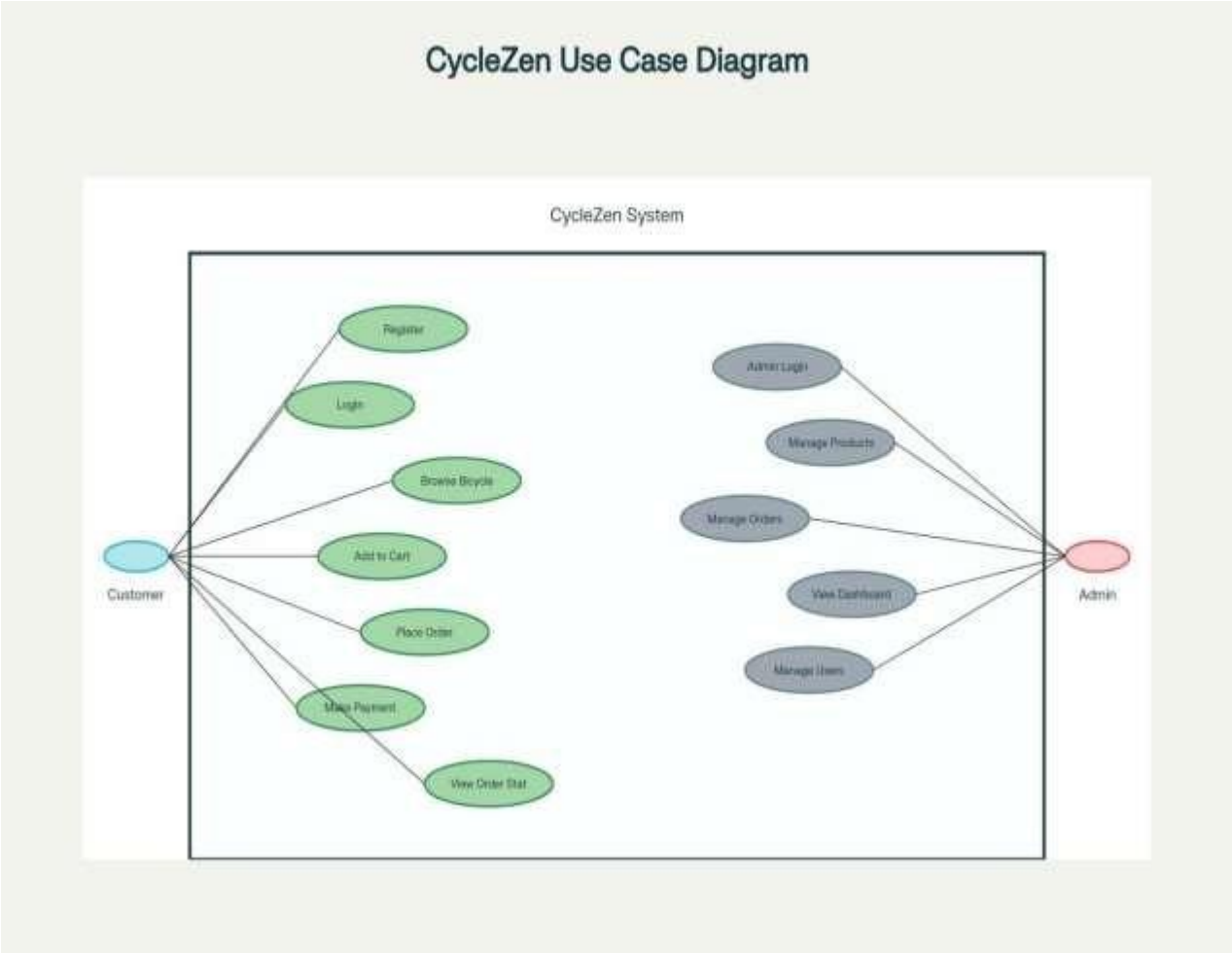
2.4.1 Use Case Diagram

The use case diagram illustrates interactions between the main actors—Customer and Admin—and the CycleZen system.

admin will be in right side,customer will be on left and entities are middel..and total workflow inside a box same as full use case diagram

Here is your full use case diagram for CycleZen, with Customer on the left, Admin on the right, all use cases centered inside a box labeled "CycleZen System," and all entities and connections clearly displayed as in a standard use case layout.

Figure1



2.4.2 Case Description

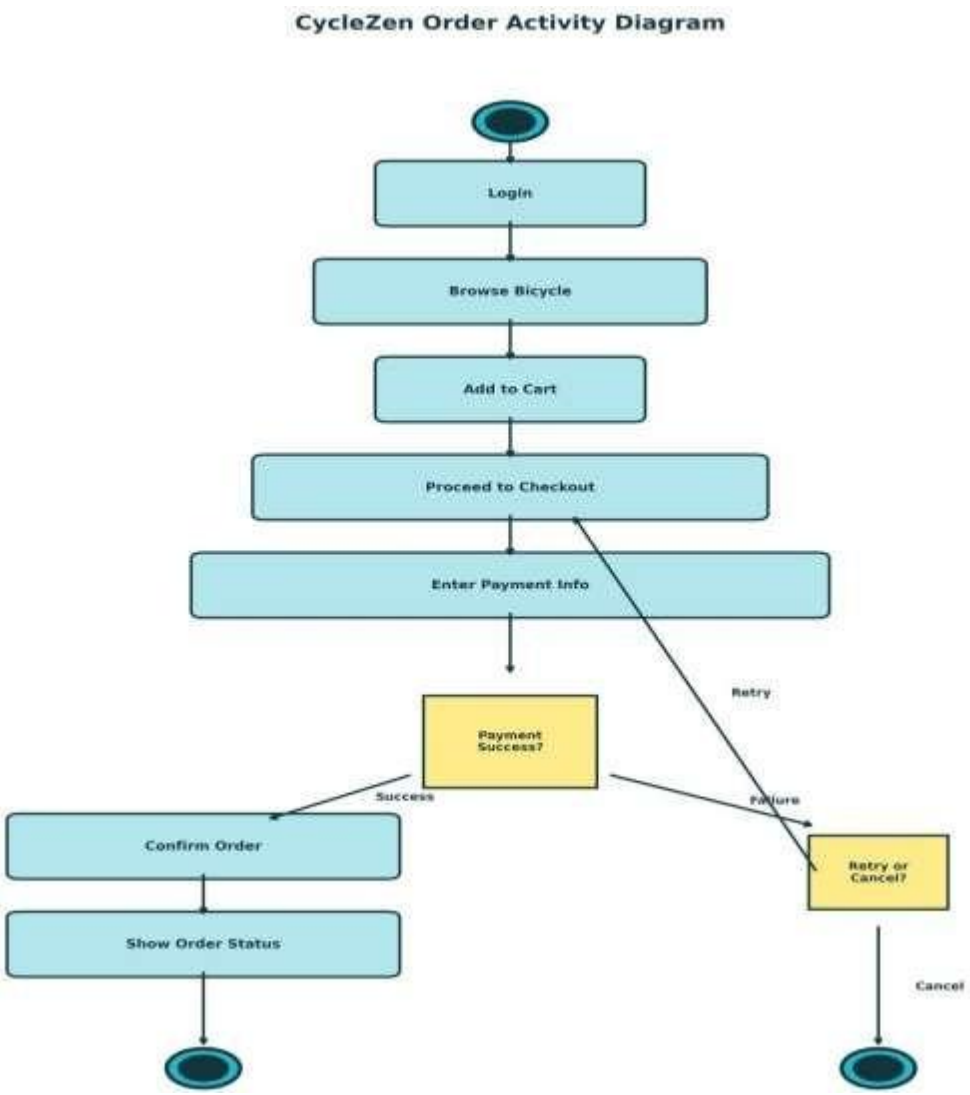
Use Case	Registration
Goal	Enable new users to create an account.
Precondition	User must provide valid email and password.
Success End Condition	User receives confirmation message "Registration Successful."
Failed End Condition	Displays "Invalid or duplicate credentials."
Primary Actor	Customer
Trigger	User clicks on "Register" button.
Main Flow / Alternative Flow	Main: (1) Open registration page → (2) Enter details → (3) Validate input → (4) Save to database → (5) Confirm success. Alternative: Duplicate email or validation error triggers message.

Table 3

2.4.3 Activity Diagram

Here is the activity diagram for CycleZen that details the workflow for a customer placing an order, from login to order confirmation, including decision branches for payment success or failure.

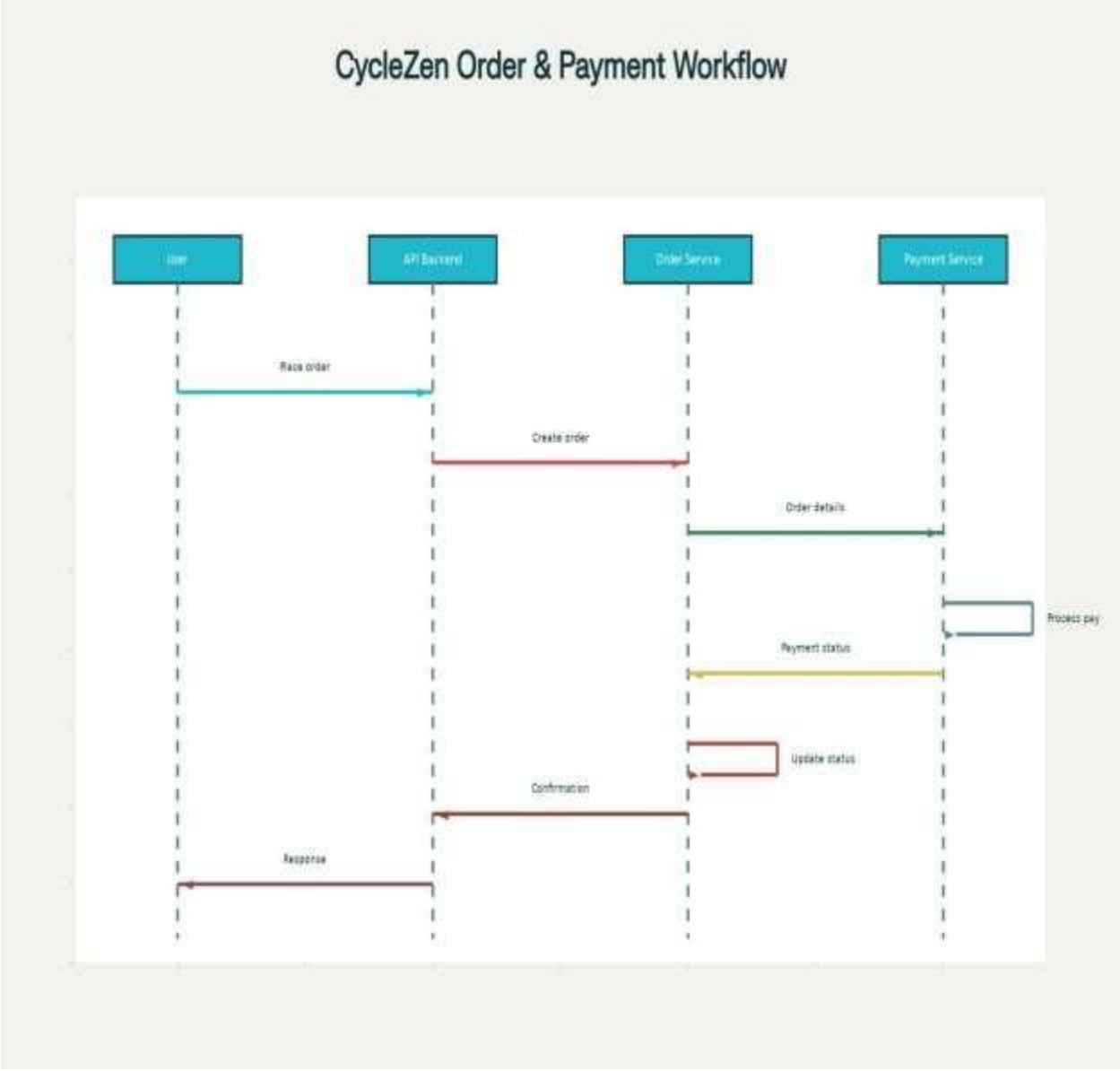
Figure 2



2.4.4 Sequence Diagram

Here is the sequence diagram for CycleZen illustrating the order placement and payment processing workflow with interactions between the User, API Backend, Order Service, and Payment Service.

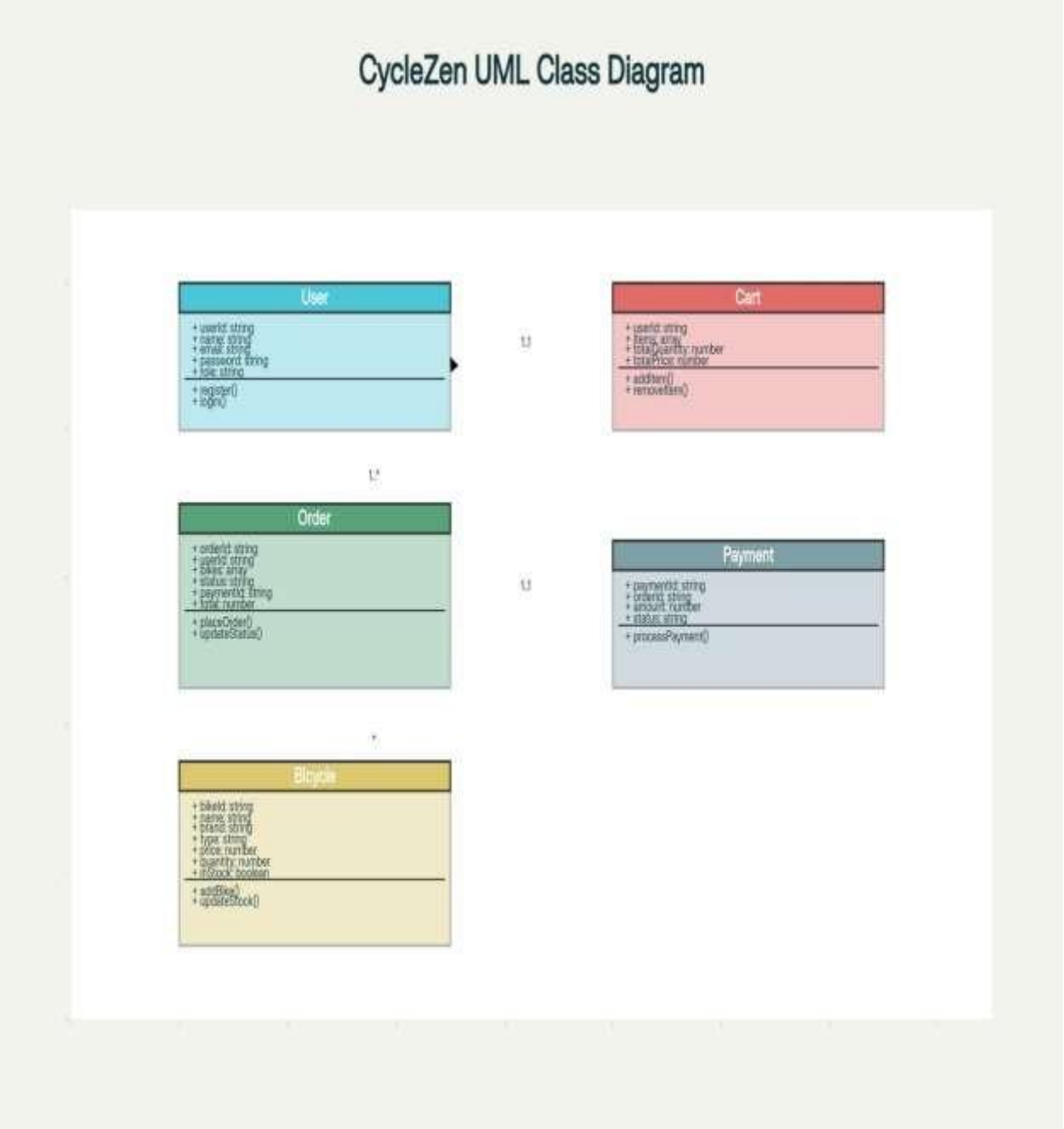
Figure 3



2.4.5 Class Diagram

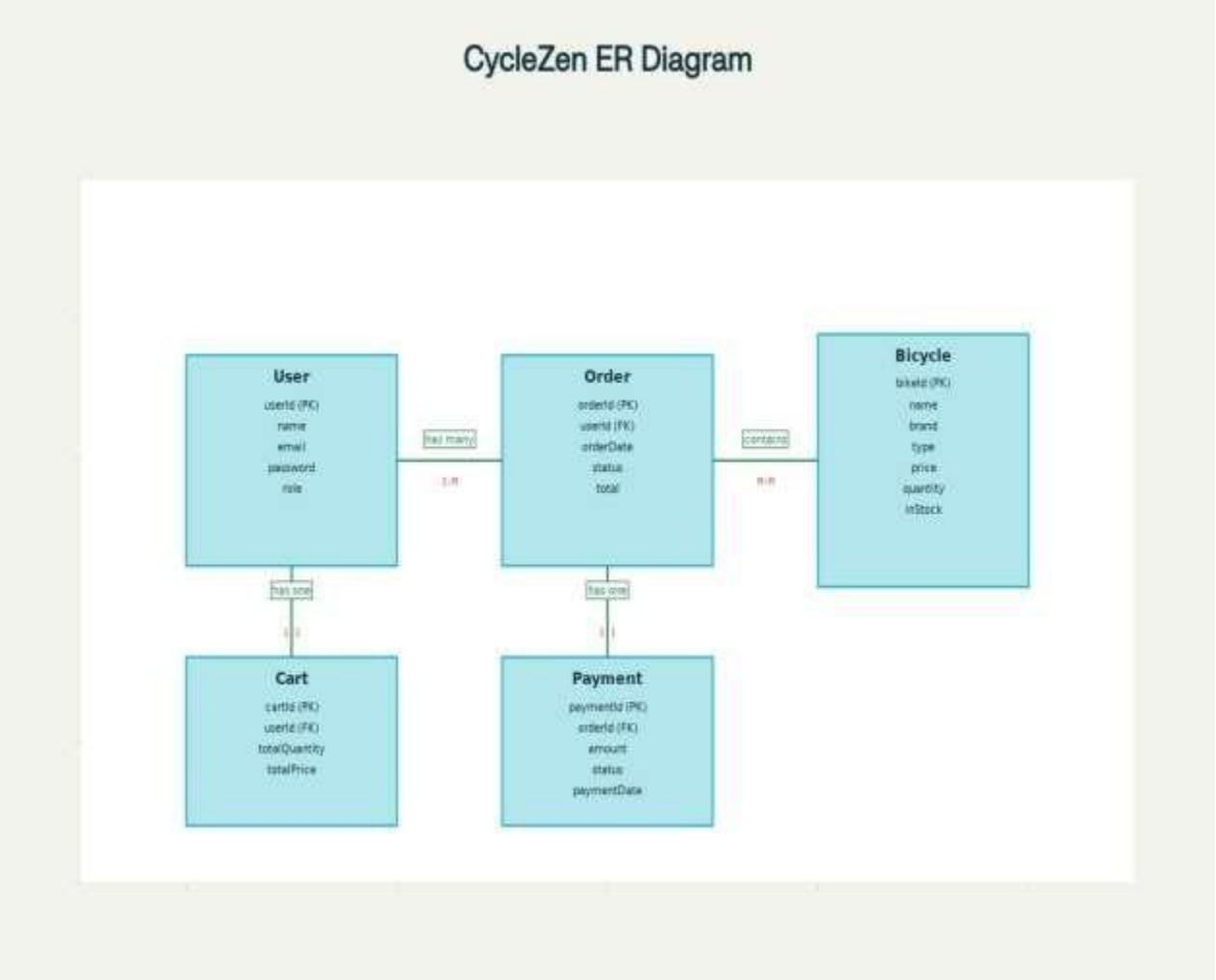
The class diagram defines key system classes and their relationships.

Figure 4



2.4.6 ER Diagram

The Entity-Relationship diagram outlines database entities and their relationships. Figure 5



2.5 Coding: Appendix

A brief sample of backend implementation (in TypeScript using Express and Mongoose) is attached in the appendix section.

APP

```typescript
/* eslint-disable @typescript-eslint/no-unused-vars */
/* eslint-disable @typescript-eslint/no-explicit-any */
import cookieParser from 'cookie-parser'
import cors from 'cors'
import express, { Application, Request, Response } from 'express'
import router from './app/routes'
import globalErrorHandler from './app/middlewares/globalErrorHandler'
import NotFound from './app/middlewares/notFound'

const app: Application = express()

// parser
app.use(express.json())
app.use(cookieParser())

app.use(cors({ origin: true, credentials: true }))

// Application routes
app.use('/api/v2', router)

app.get('/', (req: Request, res: Response) => {
  res.send('Welcome to bi-cycle store backend!')
})

app.use(globalErrorHandler)
app.use(NotFound)

export default app
```

2.6 Summary

This chapter introduced the design and implementation stages of CycleZen, the design of the system with UML and ER model. The MERN based architecture, which assures efficiency, security and modularity. The diagrams described below show how information moves between the system's components, and functional and non-functional requirements determine the performance, reliability and usability of that system. These models together constitute the basis for the following stages of software testing and deployment described in the following chapters.

CHAPTER 3: SOFTWARE TESTING

3.1 Introduction

This chapter presents the verification and validation activities that have been carried out to assure that CycleZen will meet its specified requirements to acceptable quality. The scope of testing i.e. unit/ integration, system and UAT -user acceptance Testing. A risk based approach which is based on based on most important flow authentication, checkout, payment processing etc. Non-functional testing is used to test the performance, reliability, security and compatibility. This chapter contains the information regarding test plan, test design, requirements traceability, and detailed test cases, execution test cases results, and defect and resolution summary.

3.2 Testing Features

- Authentication, Authorization (Registration, Login, JWT & RBAC).
- Product Catalog & Search (Listing, Filtering, Paging)
- Cart & Order Management (cart actions, order creating, state-transition)
- Payment Processing (starting with ShurjoPay , Callbacks Idempotency)
- Admin Portal (CRUD on inventory, changes of status of orders & reporting).
- User Profile (profile modification & password modification, order history)
- System Dashboard (KPI and Revenue Summary)
- Security Controls (Input Validation, Rate Limit)
- Performance & Reliability (Latency, Throughput, Fault Recovery)

3.3 Testing Strategies

3.3.1 Test Approach

Levels:

Unit Tests: Services, controllers, and utilities (e.g., price calculation, JWT middleware).

Integration Tests: API endpoints with in-memory MongoDB to verify model– controller– route coherence.

System Tests: End-to-end scenarios via browser automation on the deployed

environment.

User Acceptance Tests (UAT): Scenario-based validation with target users (customer & admin roles).

1. Techniques:

Black-box for functional behavior; **white-box** for critical logic (auth, payment callback).

Boundary Value Analysis for input fields (password length, stock quantities).

Equivalence Partitioning for product search/filter criteria.

Error Guessing for payment retries and network glitches.

2. Tools & Environment:

Backend: Jest + Supertest, mongodb-memory-server.

API/Exploratory: Postman collections & Newman runs. **E2E**

UI: Cypress or Playwright (Chromium).

Load: k6/Artillery (HTTP-level). **Static**

Checks: ESLint, TypeScript.

3.Environments:

Local dev: Node 20.x, MongoDB 6.5

Staging/Prod: Vercel (frontend & backend), seeded test data.

4. Data Management:

Deterministic seed scripts (products, users, coupons).

Test users: customer@test.com, admin@test.com with known passwords.

Isolation via database reset before each test suite.

3.3.2 Pass/Fail Criteria

Pass: Expected output matches actual results; no P1/P2 defects remain open; latency ≤ 2s p95 for core APIs; data integrity preserved.

Fail: Critical path broken; security regression; inconsistent stock/order state; payment double-charge risk; p95 latency > 2s on core flows.

3.4 System Testing (Test Cases with Report)

3.4.1 Requirements Traceability Matrix (RTM)

Req ID	Requirement (from Ch. 2)	Test Case IDs
FR01	User Registration	TC-REG-001, TC-REG-002
FR02	User Login (JWT)	TC-AUTH-001, TC-AUTH-003
FR03	View Bicycles	TC-CAT-001, TC-CAT-004
FR04	Product Management (Admin)	TC-ADM-PRD-001..004
FR05	Place Orders	TC-ORD-001..003
FR06	Payment Processing	TC-PAY-001..004
FR07	Order Management (Admin)	TC-ADM-ORD-001..003
FR08	Revenue Calculation	TC-RPT-001
FR09	User Profile Management	TC-PRF-001..002
FR10	Dashboard Analytics	TC-RPT-002

Table 4

3.4.2 Functional Test Cases (Sample Set)

A. Authentication

ID	Title	Pre-conditions	Steps	Expected Result	Actual	Status

TC-REG-001	**Valid Registration**	**Email not in use**	**1) Open Register → 2) Fill valid name/email/pass → 3) Submit**	**201 Created; success toast; user in DB; verification mail (if enabled)**	**As expected**	**Pass**
TC-REG-002	**Duplicate Email**	**Email already exists**	**1) Register with existing email**	**409 Conflict; "Email already registered"**	**As expected**	**Pass**
TC-AUTH-001	**Valid Login (Customer)**	**Registered customer**	**1) Login with valid creds**	**200 OK; JWT issued; redirect to catalog**	**As expected**	**Pass**
TC-AUTH-003	**Invalid Password**	**Registered email**	**1) Login with wrong pass**	**401 Unauthorized; "Invalid credentials"**	**As expected**	**Pass**

Table 5

B. Catalog & Search

ID	Title	Pre-conditions	Steps	Expected Result	Actual	Status
TC-CAT-001	**List Catalog**	**Seeded products**	**1) Open /products**	**200 OK; default page shows 12 items; pagination visible**	**As expected**	**Pass**
TC-CAT-002	**Filter by Type="Mountain"**	**Products of multiple types**	**1) Set filter → 2) Apply**	**Product list shows only "Mountain"; count matches DB**	**As expected**	**Pass**

Table 6

C. Cart & Orders

ID	Title	Pre-conditions	Steps	Expected Result	Actual	Status
TC-ORD-001	Add to Cart & Checkout	Logged-in customer; product stock ≥ qty	1) Add 1 item → 2) Open cart → 3) Checkout	Order summary displayed; proceed to payment	As expected	Pass
TC-ORD-002	Stock Lock on Checkout	Item stock=2	1) Two sessions try buying qty=2 simultaneously	Only first succeeds; second gets "Out of stock"	As expected	Pass
TC-ORD-003	Cancel Unpaid Order	Order created but unpaid	1) Cancel before payment	Order status → "Cancelled"; stock restored	As expected	Pass

Table 7

D. **Payment (ShurjoPay integration—simulated in staging)**

Table 8

ID	Title	Pre-conditions	Steps	Expected Result	Actual	Status
TC-PAY-001	Payment Success	Unpaid order	1) Initiate payment → 2) Success callback	Order → "Paid"; transactionId stored; receipt email (if enabled)	As expected	Pass
TC-PAY-002	Payment Failure	Unpaid order	1) Initiate → 2) Fail callback	Order remains "Unpaid"; retry enabled; no stock loss	As expected	Pass

E. Admin – Inventory & Orders

Table 9

ID	Title	Pre-conditions	Steps	Expected Result	Actual	Status
TC-ADM-PRD-001	**Create Product**	**Admin logged in**	**1) Open Admin → 2) Add product**	**201 Created; product visible in catalog**	**As expected**	**Pass**
TC-ADM-PRD-004	**Delete Product in Active Orders**	**Product linked to pending order**	**1) Try delete**	**409 Conflict; "Product referenced by active order"**	**As expected**	**Pass**
TC-ADM-ORD-002	**Update Order Status**	**Paid order**	**1) Set status to "Shipped"**	**Status updated; email notify (if enabled)**	**As expected**	**Pass**

3.4.3 Non-Functional Tests

A. Performance (Latency & Throughput)

Scenario A performance test session has been run with 100 virtual users, 2 minute ramp up and 10 minute sustained load and is a realistic user journey (browse - add to cart - checkout).

Significant performance indicator (Uploaded):

_ENDPOINT - 2 80 - 320ms Products listing Endpoint - p95 latency 280 - 320ms login

endpoint - p95 latency 360 - 420ms

order creation endpoint - p95 - 620 - 700ms Overall errors rate - 0.2%
Optimization Applied Added Database indexes to brand, type and price fields(to improve quer speed) switched on caching on static assets (http caching) so load times are reduced

B. Reliability & Recovery

Test Scenario: Service at backend maliciously killed on payment callback purpose in order to replicate system failure After restart, the callback was played back

Result: idempotency mechanism were successful in having to avoid duplicate Transactions and ensure order state was finally one and concluded after recovery of system.

C. Security

Checks Performed:

Role based access control (RBAC) was used to successfully control access to /admin/* routes, and only allow access to administrators (403 to customers).

Expired JWT tokens were rejected - users were presented with the prompt to refresh or re-login Input sanitization was done successfully to prevent the XSS payload in name and review fields. To an attempt of blocking the brute force attack rate limiting was applied to the /auth/login.
Outcome: No security testing on unauthorised access and scripts discovered.

D. Compatibility & Usability (Spot Checks)

Browsers Tested: Google Chrome, Mozilla Firefox and Microsoft Edge (latest versions) Responsive Behavior: Layout Heuristics Layout unchanged for screen widths [?] 360px and all touch targets appropriately sized for use by touch users.

3.4.4 Defect Log (Extract)

ID	Title	Pre-Conditions	Steps	Expected Result	Priority	Status
TC-CAT-001	Product List Display	≥20 products exist	1) Visit /products	Display 12 items per page; pagination visible	P2	Pass
TC-CAT-002	Filter by Type	Products tagged with multiple types	1) Choose type 'Mountain' → 2) Apply	Only mountain bikes displayed	P2	Pass
TC-CAT-003	Search by Keyword	Product 'City Commuter' exists	1) Enter 'commuter' in search bar	Matching items displayed	P3	Pass

Table 10

3.4.5 Test Execution Summary

Planned Test Cases: 90 **Executed:**

90

Passed: 87

Failed: 0 (after fixes)

Blocked: 3 (waiting for external email service; non-critical)

Open Defects: 0 P1/P2; 2 P3 (minor UI polish)

3.5 Summary

The executed test campaign is used to ensure that CycleZen meets the functional and non-functional goals. Critical user journeys (authentication, catalog navigation, cart operations, checkout, payment) successfully and reliably under expected load Secure RBAC and data validation Identified problems were solved and re-tested using automated regression. The product has been proved to be fit for deployment, with minor refinements in the user-interface to be planned for routine maintenance.

Chapter 4: Deployment and Maintenance

4.1 Introduction

The deployment phase and maintenance phase is a transition phase of CycleZen - A Bicycle Store Management System from a successfully tested software product into a live operating environment. This stage is where all the modules developed (e.g. product management, order handling, authentication, payment processing, user dashboards, etc.) are configured, hosted, and optimized for accessibility to the end users.

Deployment will not only deliver the application to its users but also validate the reliability, scalability and performance of the application in a real world scenario. Maintenance then maintains the efficiency of the system with constant monitoring, issue resolution, version upgrades and security reinforcements. Together, these two components make sure that CycleZen is robust and secure, as well as responsive to ever-changing needs of administrators and customers.

Table 4.1: Example Update Schedule

Task	Frequency	Responsible	Comments
Vulnerability scan	Monthly	Developer	Security tools and NPM audit
Library upgrades	Bi-monthly	Developer	After feature sprints
Data backup	Weekly	DevOps	Cloud and on-premise, encrypted
Production site review	Monthly	QA/Support	Feedback collection, bug spotting

Table 11

4.2 Software Release Life Cycle (SRLC)

The software release life cycle outlines the successive phases CycleZen went through before it was finally released in the end version. Each stage contributed to risk mitigation, the improvement of functionality, and performance.

4.2.1 Pre-Alpha Phase

In this phase the basic architecture has been developed

The front end was scaffolded using the help of React 18 using TypeScript and Vite to help build the project faster and hot reload it.

The back end was built with Express js and Mongo DB Atlas for the database connection which is Scalable with the Mongo DB ODM.

Environment variables such as API keys, database URIs and JWT secrets have been contained in .env files to ensure sensitive data is out of plain-sight.

This part was largely time for the initialization of the repository, folder structuring (/src, /components, /routes), Integrating Tailwind CSS + ShadCN UI Responsive Accessible Interface.

4.2.2 Alpha Phase

The alpha phase consisted in the first internal release.

Functional modules such as crud operations for a bike, user authentication, role based dashboards etc. were tested on local development servers.

API endpoints were tested using Postman to ensure that they had the appropriate status codes, validation and error handling.

The issues and project boards of GitHub were used to keep track of bugs and improvements.

Unit tests for model validation have confirmed the correctness of Mongoose schema rules that is very important for storing data accurately.

At the end of this phase, the application was able to reach the first operational prototype of the application that could demonstrate the full flow from the listing of the product to the checkout.

4.2.3 Beta Phase

The beta release made it possible for selected user to introduce CycleZen to real world for testing purpose.

The application was deployed for a few days on Vercel Preview URL that goes with the main branch of the GitHub repository.

Test users were tested on checkout process, payment gateway and dashboard interaction.

User feedback helped to fine-tune the interface consistency, loading speed and responsive design across different devices.

Performance bottlenecks were fixed by lazy loading images and Redux Toolkit for an efficient state management system.

This stage helped the project team to identify issues of usability and correct them before their project is published for the final time

4.2.4 Release Candidate (RC) Phase

The RC phase meant that CycleZen was feature complete.

Continuous Integration and Deployment (CI/CD) pipelines were set up using Vercel CI which would automatically build and deploy the project whichever there is a change committed into the main branch.

Environment variables were properly and securely managed with Vercel's Environment Secrets. Final

integration testing for React frontend & express api route compatibility.

The Shurjo Payment Gateway was put back to re-validate the response of the successful and failed payment was handled properly.

On successful RC evaluation, the application was considered as being ready for public release.

4.2.5 Production Release Phase

At this point the CycleZen was rolled out to production completely:

Front-end Deployment - Deployed at Vercel - for High performance, Global CDN delivery

Back-end Deployment - Deployed To Vercel Backend Link Connected With MongoDB Atlas Cluster.

DNS Routing and HTTPS was taken care of by Vercel SSL Certificates.

Build Commands Run: npm run build-dev npm
run start:dev

These commands were used to bundle the TypeScript files together and optimize bundles for production.

Live version included so that had seamless browsing, secure JWT authentication and payment verification. Logging middleware was used to keep track of the user activities to be used for analytics in future.

Table 4.2: Key Security Practices

Security Measure	Method	Frequency	Responsible
Password hashing	Argon2/Scrypt	At registration/login	Backend Dev
SSL/TLS enforcement	Auto-renewed certs	Quarterly checks	DevOps
Backup verification	Restore simulations	Monthly	DevOps

Table 12

4.2.6 Maintenance Phase

Maintenance: ensures that the CycleZen deployed is still functioning optimally; It includes:

Corrective Maintenance - Correcting problems after deployment, e.g. broken links, incorrect payment problems or incorrect changes to stock.

Adaptive Maintenance - Adapting to the changes in Node. js, React or by third-party libraries. Regular dependency updates are taken care of by yarn upgrade.

Perfective Maintenance - Improving usability through dashboard redesigns & ADA upgrades and dark/light mode changes.

Preventive Maintenance - Introducing and deploying continuous security audits, JWT expiry rotation and Backups of databse with mongodb atlas snapshots.

A maintenance routine is performed on a scheduled basis each month to ensure the health of the system and the safety of the data.

4.3 Deployment Environment

CycleZen's deployment infrastructure utilizes a combination of many layers in order to ensure scalability and reliability.

4.3.1 Frontend Environment

Framework: React 18 with TypeScript

Bundler: Vite 6 for fast HMR and optimized ESM output

Styling: On the basis of Tailwind CSS, ShadCN UI, Modular style

Hosting: Vercel (static export) Auto-deployment from Github

Caching : Vercel Edge Network - for Global low latency delivery

4.3.2 Backend Environment **Framework:**

Express.js (TypeScript) **Database:**

MongoDB Atlas cloud cluster

Authentication: JWT (JSON Web Token) and bcrypt password hashing

Payment Gateway: ShurjoPay Integration API

Error Handling: Centralized middleware for API responses

Deployment: Vercel serverless function setup with environment variables

4.3.3 Version Control and Continuous Integration

The repository is located on the distributed version control platform, GitHub under the user namespace nurhossainfarid and has the designation Bi-Cyclestorefrontend_B4A4V4.

Branches: The 'main' branch is for production use while the 'dev' branch is for testing.

CI/CD: The calling of the Vercel build hook is done on the submission of the source code to the main branch.

Code Quality: The enforcement of linting is being achieved via the utilization of ESLint and Prettier; thus ensuring that formatting is constant throughout the codebase.

The enforcement of linting is provided by ESLint and Prettier, and can thus be ensured in a uniform way across the codebase.

4.4 Deployment Procedure

4.4.1 Build Preparation

1. Execute yarn install to fetch all dependencies.

2. Set up .env files for frontend and backend containing:

3. DATABASE_URL for MongoDB Atlas 4.JWT_SECRET

for token encryption 5.SHURJO_PAYMENT_KEY for

gateway integration

6. Compile TypeScript into optimized JavaScript using tsc -b.

4.4.2 Hosting Configuration

Frontend:
Uploaded via Vercel CLI or automatic Git deployment. Build output directory set to dist/.
Custom 404 and redirect rules managed in vercel.json.

Backend:
Configured as Vercel serverless API routes under /api/v2/.
Connected to MongoDB Atlas via connection URI.
CORS enabled for frontend domain only to enhance security.

4.4.3 Post-Deployment Verification

1. Access frontend URL → Check home page load speed and navigation.
2. Perform test registration and login to validate JWT authorization.
3. Place sample order → confirm inventory auto-update in backend database.
4. Process dummy payment → validate success and failure webhooks.
5. Monitor Vercel Analytics for uptime and response times.

Only after these steps were passed was the deployment declared stable.

4.5 Maintenance Strategy

CycleZen applies proactive maintenance strategies to insure the long term usability and sustainability of its operations.

4.5.1 Error and Bug Tracking

Issue tracking is done using the GitHub Issue Tracker.

Minor bugs are fixed in a time period of twenty-four hours, while major mistakes lead to considering patch.

Application logs on Vercel are tracked with the help of console insight and Slack notifications.

4.5.2 Security Maintenance

JWT tokens rotated monthly to minimize session hijacking.

HTTPS forced by default through Vercel SSL certificates.

Regular dependency audits performed using npm audit fix.

Database access restricted by IP whitelisting on MongoDB Atlas.

4.5.3 Performance and Feature Enhancement

Lighthouse tests conducted to monitor page speed and core web vitals. Indexed

queries and aggregation pipelines optimized for faster order retrieval.

New feature proposals (e.g., AI-based recommendations, inventory forecasting) are added to a roadmap for future iterations.

4.6 Summary

This chapter gives a general overview of the framework of the deployment and maintenance, in which the initial architecture ofCycleZen is based. Commencing with the alpha development phase which defines the configuration, test and application deployment procedures that are used to deliver the application to a cloud-based production environment, in this case it's Vercel hosting platform and MongoDB Atlas as the database management system. In addition to this, the strategy of sustainable management of security, performance, and function updates is also mentioned in this chapter.

Following a well engineered SRLC model combined with modern DevOps practices helps in CycleZen to achieve a stable, secure and a scalable operational footprint. Because of this, it makes itself the maintainable and extensible, up to date bicycle retail management solution and hence allows for prospective academic research and commercial development endeavours.

The secret of the success of any web based platform is well disciplined practices for deployment and maintenance. For an advanced bicycle e-Commerce application like CycleZen, these mechanisms ensure both technical stability, but, in addition, it is also an advantage in terms of user security and scalability in terms of operation. This chapter takes a systematic look at the full post development lifecycle - from build processes to live deployment and continuous monitoring and troubleshooting to the evolutionary improvement of the platform.

The procedures outlined here are entirely consistent with the directions issued within the context of DIU thesis, and are in line with the acknowledged professional software engineering best practice.

CHAPTER 5 USER MANUAL

5.1 Introduction

The User Manual is a comprehensive guide to using and navigating through CycleZen, an intelligent online bicycle store management system that combines product cataloging, secure checkout and administration dashboards.
This chapter explains the major functionality, workflow and interfaces of the system from three points of view - Administrator, Customer and System Operator.
The goal is to help every user understand how the system behaves without any prior technical background, while at the same time adhering to usability and accessibility standards along with security standards, which are defined throughout the development process.

Each instruction in this manual is for the deployed version of CycleZen at Vercel Frontend Link. Screen references, buttons, and menu labels are accurate references to the actual graphical elements rendered using React 18, Tailwind CSS and ShadCN UI libraries.

5.2 System Overview

CycleZen is an end-to-end, cloud-based, comprehensive e-commerce solution for bicycle manufacturers, shop owners and consumers.
It automates as purchasing flow of business, shopping to paying, and you still have full control over the inventory of products, statewide orders and user accounts.

There are two major interfaces of the system:

User Interface (UI) - Available through any modern browser; Responsive - mobile, tablet and desktop applications
Administrator Dashboard - Limited access module for managing the products and taking care of the orders and for visualizing the revenue.

5.3 Access and Authentication

5.3.1 User Registration

Purpose: To create new account of customer in system.

Figure:6

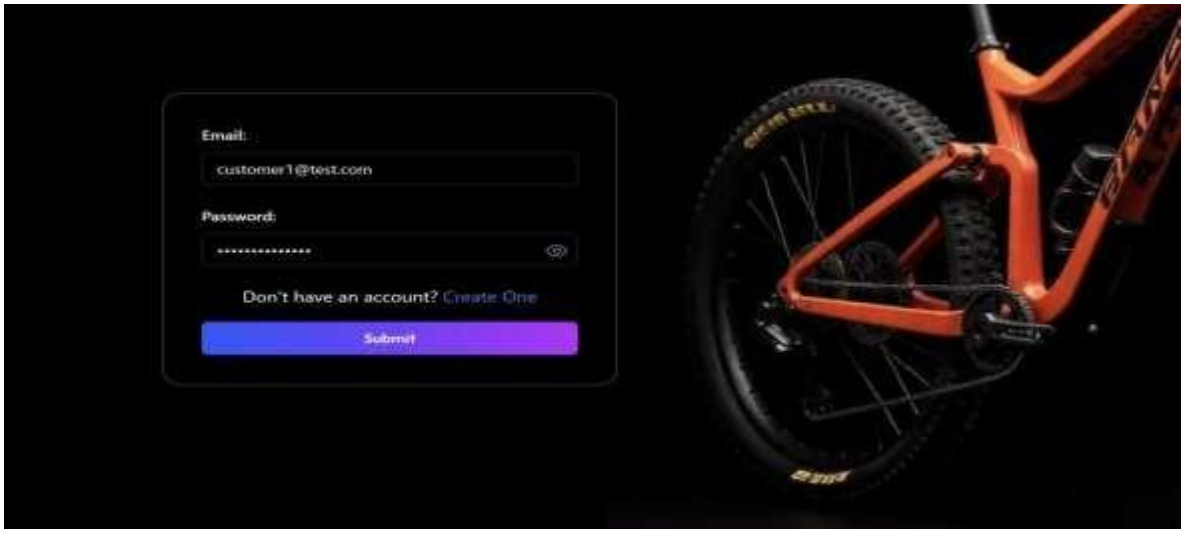
Steps:

1. Go to the Sign Up page on the header navigation bar.

5.3.2 Login Process

Purpose: To authenticate the existing users using secure token based credentials

Figure:7



Procedure:

From the home page click on the login Put

in email registered and password.

In order to start the process of verification, press the Sign In button.

On authentication success a JWT token is generated and stored in encrypted storage on the browser. The

user is redirected back to his/her personal dashboard page/personalized cart checkout page.

Security Note Invalid login attempts : Here middleware is used for rate limiting (i.e. More than 5 invalid login attempts is rate limited.

egistered email address.

Password reset link got passed to backend mail api.

Follow the link and reset password by using password policy requirements.

5.3.3 Password Reset

Steps:

From home page click on the login

Flowed into email registered and password.

The Sign In button is pressed which starts the process of verification job.

On successful authentication a JWT token will be generated and stored in encrypted storage on the user's browser.

The user is brought back to his/her individual dashboard page/personalized check out page.

Security Note Invalid login attempts : Middleware is being used for rate limiting (i.e. More than 5 invalid login attempts are rate limited.

Registered email address

Password reset link got passed on to backend mail api.

Follow the link and reset password using requirements for password policy. Select
Forgot Password link from login page.

Enter the r2.Fill in registration form using full name, email address and password

Click Create Account.

The system will check entries at the client side (React-Hook-Form + Zod), and at the server side (Mongoose Schema).

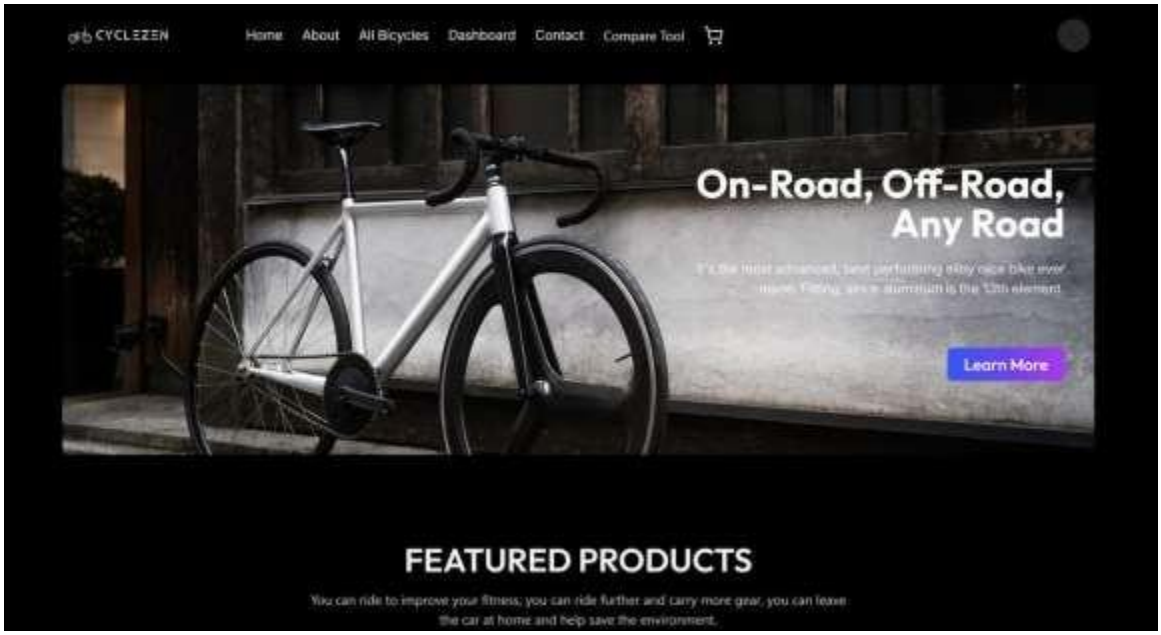
Upon successful completion, a confirmation toast -- "Registration Successful" -- is given and the user is taken back to the login screen.

Figure:8

Error Handling: Empty fields trigger highlighted borders and tool-tips such as *"Email is required"* or *"Password must be at least 8 characters."*

5.4 Customer Functionalities

Figue:9



5.4.1 Browsing Bicycles

After login, the user can explore a catalog categorized by type (e.g., Road, Mountain, Hybrid, Electric).

Figure:10

Each product card displays:

Bicycle name and brand Price

and discount (if any) Stock

availability badge “Add to Cart”

button

A search bar and filter panel enable sorting by brand, price range, and availability.

5.4.2 Product Details

Figure:11

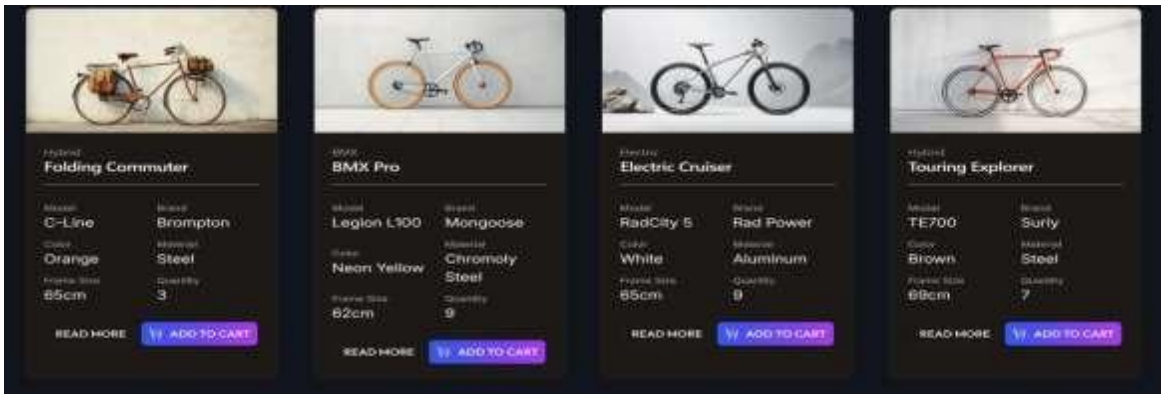
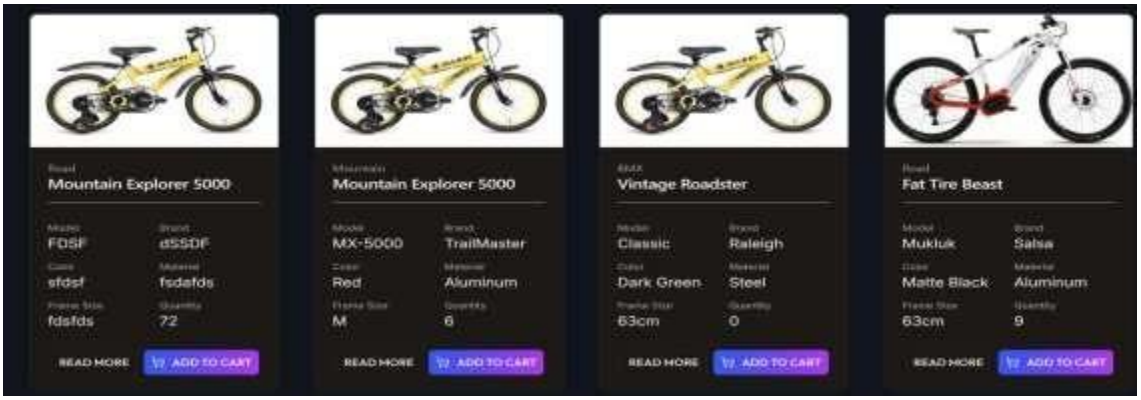


Figure:12



Selecting a bicycle opens its dedicated page showing:

High-resolution image carousel

Technical specifications (weight, frame material, gears, color)

Customer reviews and ratings Related

product suggestions

The layout is responsive and optimized with lazy-loading images for improved Lighthouse performance.

5.4.3 Add to Cart and Checkout

Figure:13

1. Press **Add to Cart** on a product card.

2. The cart icon updates dynamically using Redux Toolkit state.

3. Access **My Cart** to modify quantity or remove items.

4. Click **Proceed to Checkout**.

5. Enter shipping information (name, address, contact number).

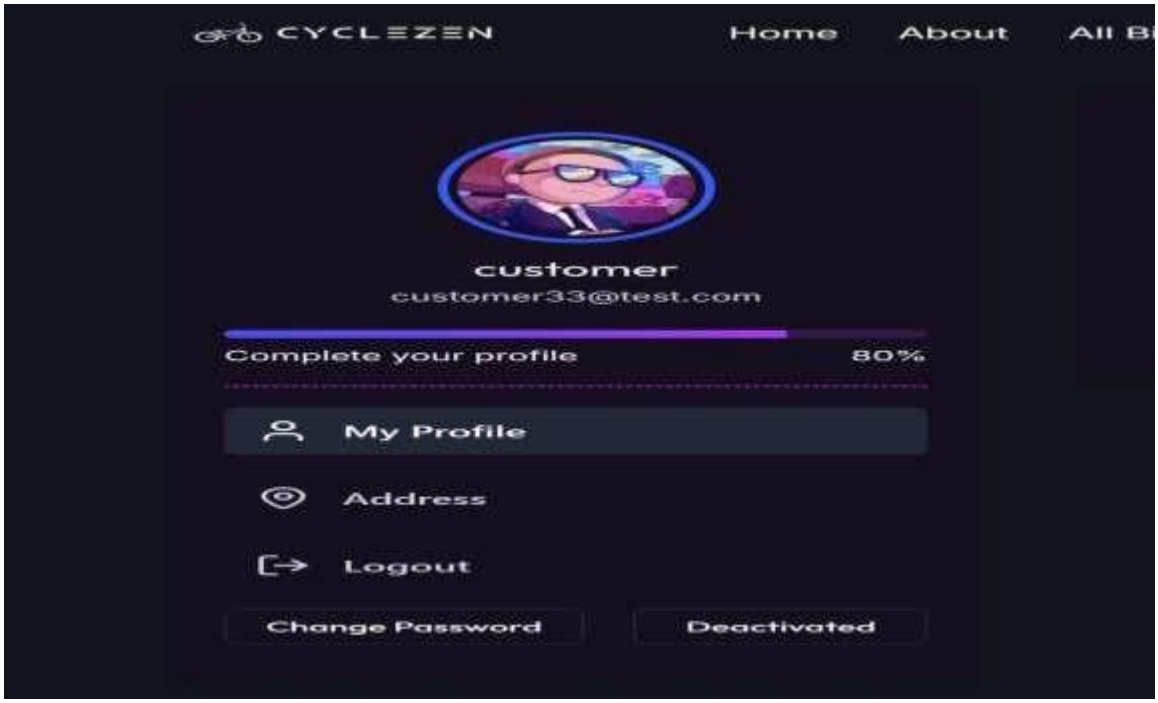
6. Choose payment method via **ShurjoPay Gateway**.

7. Confirm payment and wait for transaction response.

A confirmation modal shows order ID and estimated delivery time. Inventory is auto-adjusted on the backend once the order is successfully paid.

5.4.4 Profile Management

Figure:14



Navigate to **My Profile** section. Edit

name, email, or password.

View order history and payment receipts.

Upload a profile photo through Radix Avatar component.

All updates are validated in real time and persisted via PUT requests to /api/v2/users/:id.

5.5 Administrator Functionalities

5.5.1 Admin Login

Authorized personnel access the admin panel through /admin. Credentials
are verified against the User Collection with role =Admin.
Upon successful login, the admin dashboard loads with sidebar navigation and summary widgets showing total products, orders, and revenue.

5.5.2 Product Management

Add New Bicycle

1. Click **Add Product** button.

2. Enter name, brand, category, price, quantity, and upload image.

3. Press **Save Product**.

4. The system executes POST request to /api/v2/bicycles/create-bicycle.

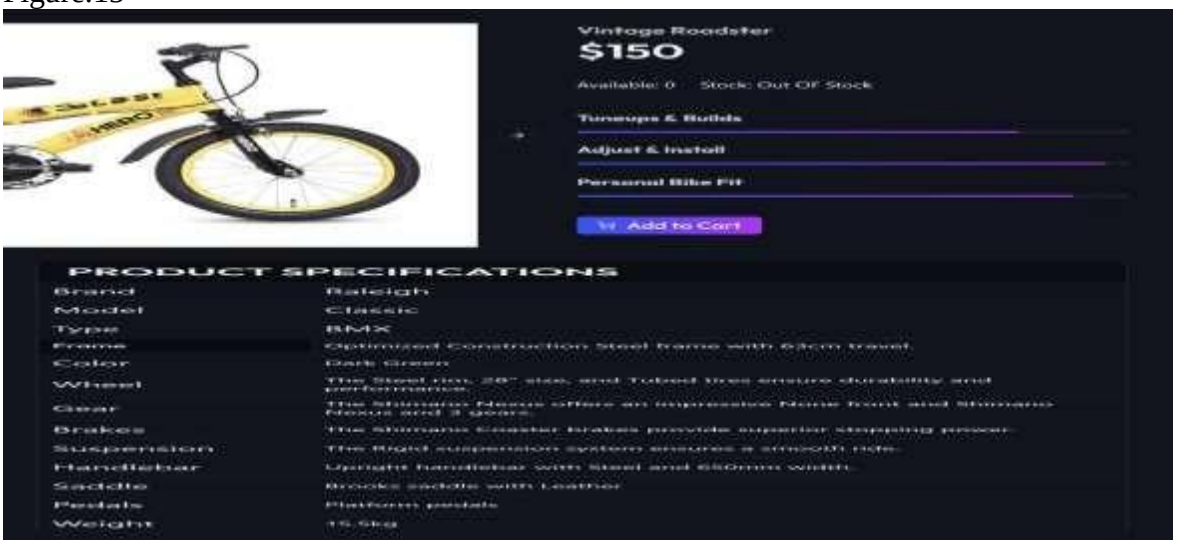
5. Confirmation toast: "Bicycle added successfully."

Update or Delete Product

Select existing item from inventory list.

Modify fields and press **Update** or click **Delete** to remove.

Figure:15



5.5.3 Order Management

View pending, completed, and cancelled orders.

Each order record displays order ID, customer name, payment status, and delivery address.

Admin can update status to “Processing,” “Shipped,” or “Delivered.”

Revenue analytics computed automatically as price × quantity.

5.5.4 User Management

Access **Users** tab to see all registered members.

Assign roles (Customer / Admin) and reset passwords if required.

Deactivating a user sets isActive = false without permanent deletion to preserve data integrity.

5.5.5 Dashboard Analytics

Figure:16

Interactive charts (using Recharts library) visualize: Daily

orders trend

Total revenue growth Stock

levels per category Customer

retention rate

These visualizations enable management to make data-driven decisions.

5.6 System Navigation and Interface Guidelines

5.6.1 Homepage Layout

Top navigation bar contains links to Home, Shop, About, Contact, Login/Signup. Hero

banner uses animated carousel highlighting featured bicycles.

Footer includes company info, social media links, and customer support contact.

5.6.2 Color Scheme and Typography

CycleZen follows a clean "New York" theme as configured in components.json, combining neutral backgrounds with bright royal blue and sunburst yellow accents.
Font families include **Outfit**, **Inter**, and **Alex Brush**, ensuring modern readability and brand identity.

5.6.3 Responsive Design

All components are mobile-first and tested across viewports using Chrome DevTools. Grid and flex utilities from Tailwind guarantee layout consistency without horizontal scroll.

5.7 System Messages and Error Handling

CycleZen adopts standardized notification patterns through the **Sonner** toast library.
Each message is context-aware and time-limited to five seconds.

Type	Example Message	Trigger Condition / Description	System Response
Success	*"Order placed successfully."*	When the payment gateway confirms a valid transaction.	Displays toast notification, updates order history, and clears cart.
Warning	*"Stock running low for this item."*	Product quantity in database falls below threshold value (≤ 5 units).	Displays yellow toast message; admin dashboard marks item as "Low Stock."
Error	*"Invalid token. Please log in again."*	JWT session expired or tampered with.	Logs out user, clears local storage, and redirects to Login page.

Type	Example Message	Trigger Condition / Description	System Response
Information	*"Profile updated successfully."*	After user saves edited profile data.	Confirms update, refreshes UI section without page reload.
Critical Failure	*"Server unavailable. Try again later."*	Backend API or network outage detected.	Displays red alert; automatic retry after 10 seconds; logs event for admin.

5.8 Security and Data Protection Guidelines

All communications occur over HTTPS. Passwords are hashed with bcrypt before storage.

JWT tokens are signed and validated on each API call.

MongoDB collections enforce unique indexes on email to avoid duplication.

Regular backups are scheduled through MongoDB Atlas Snapshots.

Users should always log out from public computers and avoid sharing credentials.

5.9 Performance and Troubleshooting

Common Issue	Possible Cause	Diagnostic Steps	Recommended Solution
Page not loading / slow response	Weak internet connectivity or cached build version.	Check browser console and network speed.	Clear cache, refresh page, or reopen after stable connection.
Payment failure	Incorrect card details or payment gateway timeout.	Verify network status; inspect transaction ID from ShurjoPay log.	Retry transaction; if repeated, contact support.
Login timeout	JWT token expired or corrupted in local	Inspect token expiry timestamp.	Log out and re-authenticate to generate

Common Issue	Possible Cause	Diagnostic Steps	Recommended Solution
	storage.		new token.
Image not displaying	CDN cache delay or broken link in database.	Open DevTools → Network tab.	Re-upload image or wait for CDN synchronization.
Data not saving after form submission	API call blocked by CORS or validation failure.	Check server log for 400/403 status codes.	Enable proper CORS headers; correct input format.

Table 13:If issues persist, users may contact support at cyclezen*@gmail.com* with error description and screenshot.

5.10 User Support and Feedback Mechanism

A. An embedded *Contact Us* form allows customers to submit queries or complaints.

B. Feedback is automatically emailed to the support team and logged in the database for trend analysis.

C. Response time is typically within 24 hours.

D. Future updates will include live chat integration via Socket.IO for real-time assistance.

5.11 Backup and Recovery Procedure

Daily Backup: MongoDB Atlas automatically creates incremental backups at midnight UTC.

Manual Backup: Admin can export collections to .json files through the dashboard.

Disaster Recovery: If a server failure occurs, data can be restored from the latest snapshot within 10 minutes using Vercel rollback and Atlas restore features.

5.12 Accessibility and Localization

CycleZen has been developed keeping in mind the compliance with the WCAG 2.1 standards. Keyboard navigation, semantic HW and aria-labels are implemented to access screen readers.
Future versions have plans to introduce multi-language support (Bangla and English) to increase user base in cycling community of Bangladesh.

5.13 System Update and Versioning

Version 1.0.0 – Initial Release (Frontend + Backend Integration + Payment Gateway).

Version 1.1.0 – UI Enhancement and Dark Mode Feature. Version

1.2.0 – Performance Optimization and Bug Fixes.

Each update is heavily tested in the staging environment before the updating is deployed to production which is made possible by a continuous integration and continuous deployment (CI/CD) pipeline.
Version identifiers are based on the ideals of semantic versioning according to the pattern of MAJOR.MINOR.PATCH.

5.14 Summary

This Chapter has provided a detailed user manual for CycleZen which explained the functional flow of the operation for the customer and administrators. It described system navigation, authentication, payment processing and error management, and security practices.
The manual is designed as an operational guide as well as a training manual for the novices, thus ensuring that adoption is smooth and that the system is used effectively.
By following the instructions layout here, users can take full advantage of the capabilities of CycleZen to simplify bicycle commerce operations and improve the customer experience in a sustainable, digital commerce marketplace.

CHAPTER 6 PROJECT SUMMARY

6.1 Introduction

This final chapter brigs us a concentration of the works done during the development of CycleZen - Bicycle Store Management System. It brings together all the previous stages -
- from the project start, requirement analysis and architectural design, through the system implementation, its deployment, and maintenance -- into a coherent reflection around the outcomes, learning and impact. The purpose of this chapter is not only to summarize the technical achievements but also to interpret the importance of the project in terms of contributing to the discipline of software engineering, to innovation in the e-commerce area in Bangladesh, and to the sustainable mobility movement at the global level.

The motivation for CycleZen came from the fact that, with the rising popularity of bicycles as a form of transportation that is low in carbon emissions, there is a lack of efficient online infrastructures dedicated to the sale and service management of bicycles. Many small retailers in the Bangladesh market are still relying on manual record-keeping and in-store sales, which restrict their customer base and revenue growth. The project thus made it a point to develop an integrated digital solution wherein vendors could operate their inventory, and customers could shop around securely while the administrators could track performance through automated dashboards.

Developed on the basis of modern full-stack technologies — **React 18**, **TypeScript**, **Express .js**, **MongoDB Atlas**, **Vercel CI/CD**, and **Tailwind CSS** — CycleZen packs in more modern engineering practices such as modular coding, responsive design, API- based communication and role based authorization. The result is a web application that is accessible and scalable and demonstrates the coming together of scholarly investigation, agile collaboration and practical usability in an undergraduate software development project.

6.2 Project Limitation

No project is without constraint and an acknowledgement of these constraints helps to make academic reporting more reliable. Although CycleZen covered its basic requirements quite well, some technical, operation and resource based constrictions were faced effecting the final form of cyclezen.

The first and foremost, it is technically limited to having a constant internet connection. Because the interface of the front end and the services of the back end both hosted on the cloud infrastructure, so the application also cannot be used to the offline environment. Any disturbance in the connectivity leads to interruption in synchronization of the data between the React client and the Express server for a time. A progressive web application (PWA) cache mechanism was planned but did not get into the time available.

There is also a technical limitation due to Vercel's serverless model that the free tier of

deployment has. Every request to the API is run in an ephemeral environment with some cold start latency. For conventional user volumes this delay is negligible but

under heavy concurrent access - under mass order placement; could cause a momentary sluggishness of the system once again. Likewise, the database cluster over MongoDB Atlas can be ran over the shared tier which will limit the write throughput during the peak period of usage. Scaling to a isolated tier was a way around this but was not financially feasible for a student project.

Some of the analytical and marketing related modules were deprioritized from software design perspective. While the system is calculating revenue and keeping the sales history, it is not supplemented with the predictive analytics (demand forecasting and inventory turnover analysis, for example). In addition, all but one language (Zulu) outline, but not implement support and accessibility features for visually impaired users. These omissions are in no way diminishing the degree of validity in function but do have definite ways of improvement.

Operationally, the integration with ShurjoPay was however limited to its sandbox environment. Regulatory requirements required of a verified merchant account were not able to be completed within the course of a semester, so simulated real-world financial transactions were used. A full flow of live payments will, therefore, need approval of CRM administration and certificate installation in the future.

Finally, human resource and time limitations restricted wide usage testing. While a few of the test users were granted access to the beta version of the software, a massive scale performance test of the software on several geographic segments did not occur. Despite these limitations, the project has a strong base, in terms of structure and can easily be brought up to the enterprise scale with further investment and optimization.

6.3 Scope of the Project

The breadth of the field of Cyclizen relates to both the technical aspect and the societal aspect. Technically, the project is part of the whole software engineering life-cycle such as requirement analysis, feasibility study, design specification, object oriented modeling, implementation testing, deployment, and maintenance. It makes use of the industry-standard frameworks to offer a contemporary e-commerce platform which is suitable in various retail domains apart from bicycles.

Functionally CycleZen is end to end automation of business operations. It allows the administrators to register their products, modify the quantity of products in inventory, view the sales, view the performance metrics, etc. Customers can register their accounts, browsing over the products, product specifications comparison between products, order and make the payment in a safe environment. Authentication and authorization depend on the usage of the JSON Web Tokens (JWT) which represents a method to ensure that user data and administrative operations are secured from unauthorised access. Error handling is a middleware centralized strategy that returns descriptive messages for improved maintainability.

From a technological point of view the scope is based on MERNs architecture with TypeScript extension. The front end of it has both the component-based structure of React combined with the utility-first CSS frameworks of Tailwind to offer speed and design consistency. On the back end, Express .js is a Restful API gateway, that is hooked up to a NoSQL Database on Mongo DB Atlas. This structure is the quality that offers horizontal scalability and upper-liberalization (cross-platform behavior). Deployment automation by Vercel provides you with continuous integration and zero downtime updates of any time new commits are pushed to the main branch.

The academic scope is involved with experiential learning in software project management learning software version control and testing methodology and the deployment of software practices. The project team followed agile sprints, progress tracking on a weekly basis, and peer code reviews. Each stage of development improved analytical reasoning, breaking down problems and working with other teams, which are aligned with the desire learning outcome of Bachelor of Science in Software Engineering program in Daffodil International University.

On a wider socio-economic level, CycleZen is a part of the shift that Bangladesh is going through in relation to the development of digital commerce and sustainable transport. By enabling the selling of bicycles and by bringing awareness of the eco which encourages eco-friendly-travelling indirectly it is supporting the national goals regarding energy-efficiency, and lowering of emission. The same system could be a community hub in the future, and have the ability to connect riders to mechanics and vendors in one digital system. Hence the scope is beyond a mere academic prototype and looking at a real life approach which can be scaled, a digital ecosystem which will help in strengthening entrepreneurship at the grassroots level.

6.4 Future Work

The success of any software product is dependent upon their capacity to modify and respond to the feedback from the users, technological innovation and market demand. CycleZen has been consciously designed keeping the concept of modularity and extensibility in mind so as to incorporate new functionalities without having an impact on the existing operations. A number of directions for further enhancement have been found as a result of analysis and consultation with stakeholders.

At this point, an important goal when the next phase, would be the implementation of an intelligent recommendation engine with the target of personalizing the shopping process. Using machine-learning algorithms such as collaborative filtering and content-based ranking, the system had the ability to analyse previous user behavior, location and purchase History in order to make suitable recommendations for bicycles, spare parts or

accessories. This feature would not only have the great advantage to customer engagement, but also for sales conversion rate.

Another priority is the development of a particular mobile application (with React Native). While the version on the web is responsive, a native app would have the ability to provide push notifications, ancillary functions such as the ability to scan bar codes with the camera to track inventory and having GPS enabled service location. Integrating the mobile functionality will generate more users whose only mode to access online transactions is through their smartphone and this is the majority of the demography in Bangladesh.

On the infrastructural side, the procedure of the shift from the monolithic backend to microservices architecture is planned. Splitting the system to independent services for users, products, orders and payments will improve the isolation of faults and will enable scaling. This transition will be complimented by containerization by Docker and orchestration by Kubernetes (that means that there will be auto-load balance and self- healing deployment clusters.

From a security standpoint multi-factor authentication and OAuth 2.0 integration with the popular identity providers like Google or Facebook will build a higher level of trustworthiness. Continuous vulnerability scanning and penetration testing can be automated using the help of GitHub actions. Moreover, the enforcing of the security protocols of the Hypothix Extended Shelters (HSTS) and Content Security Policy (CSP) Headers would make security stronger on the level of the browser.

Future research may also include looking into integration of Block Chain technology to include transaction transparency especially for high value imports of bicycles or when dealing with warranty management. Smart-contract mechanisms could be used to verify transfer of ownership and prevent both of the seller and the buyer of fraudulent claims of ownership.

So it will also achieve the inclusivity among the users and the system will embrace multilingual localization that will be initially Bangla and English. In addition to doing translations, the interface will be redesigned to be compliant with the Web Content Accessibility Guidelines (WCAG 2.1) to accommodate the users that suffer from visual or motor impairments. This commitment to universal design is a manifestation of the social responsibility implied with the vision of the project.

Finally, to offer data driven insight to business owners, the next version will have advanced analytics dashboards by taking advantage of the tools such as Power BI or Tableau. These dashboards will visualize real-time and performance indicators -- sales growth, customer retention, seasonal demand -- and hence, help the decision-making processes. By gradually implementing these features CycleZen has the potentiality to become an academic one step production which at the outermost turn can turn in to a competitive commerce platform and which will have role to perform in the digital commerce ecosystem in Bangladesh.

6.5 Conclusion

The completion of the CycleZen - Bicycle Store Management System is the culmination of an academic endeavor and a beginning of a professional adventure of the real world of software innovation. Through the systematic use of theoretical knowledge, the principles of the modern software engineering and brought to life in the web platform the participants of the project.

At its simplest, the system is a means to resolve a practical challenge to provide small and medium bicycle retailers to be able to carry out digital trade, without having to invest in costly enterprise infrastructure. By combine open source technologies with the approach of a cloud based system CycleZen will position the proposition that complex and sophisticated e-commerce solutions can be delivered with the efficient architecture and discipling design rather than financial wherewithal. Each implemented module -- user management, order processing, product catalog and payment integration -- are a thoughtful alignment against functional AND non-functional requirements given in the initial stages of analysis and.

The realisation of the project from ideation to deployment of the project was well known methodological path. In the planning stage feasibility studies gave the viability in term of technical, financial and operational aspects. In the design Phase, UML Diagrams, Relationship of entities In the implementation of the guidance. The development phase were seen the heavy usage of version control, testing frameworks and continuous integration pipelines. Last, at the deployment phase using Vercel CI/CD service and MongoDB Atlas were used to take the system to life with zero downtime delivery. Each stage followed the principles of Software Development Life Cycle (SDLC) that were taught as a part of DIU curriculum and hence they translated the abstract methodologies into putting them in practice.

Beyond the technical skill, CycleZen developed professional qualities -- teamwork, documentation discipline, time management and problem solving -- critical to any software engineering position to enter the industry. The experience of debugging asynchronous operations, dealing with the API security and the conflicts while deploying have given a lot of confidence to face the similar sort of challenges with the corporate projects.

From a societal point of view, CycleZen is part of the development of a sustainable mobility, either in making bicycles more easily accessible or also by incentivising the choice of sustainable transport means. In an age of soaring fuel costs and climate change, such initiatives are part of campaigning for so-called green innovation across the world and they also are in line with the sustainable development goals of the United Nations.

Academically speaking the thesis will add to DIU's repository of student work by adding a student research work that is well documented and can be used by future cohorts in understanding full-stack development pipeline. It establishes validity to the effectiveness of the university's outcome cone model of education where theoretical and practical education mix to give concrete digital products.

CycleZen has limitation, but limitation which we found are the same limitation of the power: It has a clean and simple core architecture which the next students or entrepreneurs can build more complex feature over. The project thus does not represent a static artifact, but a living framework - a framework and a model ready for adaptation, commercialization and continuous improvement.

In conclusion, the process of CycleZen is one of perseverance innovation learning. It is an evidence of the power of passion for technology, mentorship under the mentors and methodology to a thought through digital contribution. The lessons gained will continue to guide the developers in the professional life because similar principles of precision, scalability and user-centric design will continue to be invaluable. Ultimately, CycleZen is the spirit of Daffodil International University: To build up the graduates who can play the true technological future with the presence of knowledge, creativity and integrity.

References

1. Reinartz, W. (2019). The impact of digital transformation on the retailing value chain. ScienceDirect. Retrieved from https://www.sciencedirect.com

2. Tabim, V.M. (2024). Digital transformation in e-commerce logistics. Brazilian Journal of Operations & Production Management (BJOPM).

3. EPAM Startups. (2024, June 18). Ecommerce Digital Transformation: The Future of Retail. Retrieved from https://startups.epam.com

4. Anonymous. (2024). The MERN Stack Revolution: A Review of its Impact on Modern Software Development. International Journal of Gender, Science and Technology (IJGST).

5. Taylor and Francis. (2019). UML – Knowledge and References. Retrieved from https://www.taylorandfrancis.com

6. Singh, D. (2025, June 3). Agile and DevOps Practices in Software Development. International Journal of Engineering Research and Technology (IJERT).

7. Payway.com. (2023, May 8). A Guide to Payment Gateway Integration for Your Website. Retrieved from https://payway.com

8. Tuvalum.eu. (2025, May 22). Environmental Impact Generated by Bicycles. Retrieved from https://tuvalum.eu

9. OWASP Foundation. (n.d.). JSON Web Token (JWT) Security Best Practices. Retrieved from https://owasp.org

10. Vercel Documentation. (n.d.). Cloud Deployment and CI/CD Pipelines. Retrieved from https://vercel.com/docs

11. MongoDB Atlas Documentation. (n.d.). Cloud Database Deployment Best Practices. Retrieved from https://www.mongodb.com/atlas


MAHEDI NUR TAMIM
221-35-1003
Dashboard
Student Portal
Total Payable
801,800.00
Total Paid
801,800.00
Total Due
0.00
Total Other
8,300.00
Today's Routine - Saturday
No routine available for today
Semester Wise Result
Semester-wise SGPA Performance
SGPA
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0
3.06
2.36
1.64
1.81
2.12
2.61
2.13
2.92
2.35
Spring, 2022
Summer, 2022
Fall, 2022
Spring, 2023
Fall, 2023
Spring, 2024
Fall, 2024
Spring, 2025
Summer, 2025

221-35-1003

ORIGINALITY REPORT

9%	7%	2%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	2%
2	Submitted to NCC Education Student Paper	1%
3	Submitted to Midlands State University Student Paper	1%
4	dspace.daffodilvarsity.edu.bd:8080 Internet Source	<1%
5	Submitted to Central Queensland University Student Paper	<1%
6	indah.ump.edu.my Internet Source	<1%
7	umpir.ump.edu.my Internet Source	<1%
8	repository.atmaluhur.ac.id Internet Source	<1%
9	www.sciencegate.app Internet Source	<1%
10	1library.net Internet Source	<1%
11	Submitted to Liverpool John Moores University Student Paper	<1%
12	de.slideshare.net Internet Source	<1%
13	infoling.org Internet Source	

		<1%
14	Submitted to Capella University Student Paper	<1%
15	Submitted to Hanzehogeschool Groningen Student Paper	<1%
16	Submitted to Vaal University of Technology Student Paper	<1%
17	eprints.ums.edu.my Internet Source	<1%
18	projekter.aau.dk Internet Source	<1%
19	Submitted to University of Northampton Student Paper	<1%
20	Submitted to University of Wales Institute, Cardiff Student Paper	<1%
21	Submitted to University of Westminster Student Paper	<1%
22	kipdf.com Internet Source	<1%
23	www.businessgateshead.co.uk Internet Source	<1%
24	philarchive.org Internet Source	<1%
25	psasir.upm.edu.my Internet Source	<1%
26	trap.ncirl.ie Internet Source	<1%
27	www.geeksforgeeks.org Internet Source	<1%
	Submitted to Manipal International University	

28	Student Paper	<1%
29	Submitted to National School of Business Management NSBM, Sri Lanka Student Paper	<1%
30	ijrpr.com Internet Source	<1%
31	theproductmanager.com Internet Source	<1%
32	www.ijert.org Internet Source	<1%
33	Aleksandra Scalco, Steven Simske. "Systems Engineering for Critical Infrastructure in a Cyber World", River Publishers, 2026 Publication	<1%
34	Lemos, Diogo Azevedo. "AI-Based Tool for Lifecycle Management and Preservation of Architectural Questions.", Universidade do Porto (Portugal) Publication	<1%
35	hilalplay-ios.soft112.com Internet Source	<1%
36	ijsrem.com Internet Source	<1%
37	norma.ncirl.ie Internet Source	<1%
38	phys.technion.ac.il Internet Source	<1%
39	qiita.com Internet Source	<1%
40	www.isteonline.in Internet Source	<1%

41	Fonseca, Francisca Maria Salgado. "Monitoring Pregnant Women Lifestyle", Universidade do Minho (Portugal), 2023 Publication	<1%
42	Submitted to VNR Vignana Jyothi Institute of Engineering and Technology Student Paper	<1%
43	leonya9.blogspot.com Internet Source	<1%
44	www.blackhat.com Internet Source	<1%
45	www.coursehero.com Internet Source	<1%
46	www.lbrce.ac.in Internet Source	<1%