CEBU INSTITUTE OF TECHNOLOGY
UNIVERSITY

IT342-Section SYSTEMS INTEGRATION AND ARCHITECTURE 1

FUNCTIONAL REQUIREMENTS SPECIFICATION (FRS)

Project Title:

Prepared By: Ryan Noay

Date of Submission:

Version: 1.0

Table of Contents

Introduction

Purpose

Shopify is an e-commerce web-based application developed to support local businesses by providing a centralized digital platform for online marketing and transactions. The system integrates secure electronic payment processing to ensure efficient, reliable, and seamless financial transactions between buyers and sellers. It features core modules such as product catalog management, order processing, and user account administration to enhance operational efficiency and user experience. The application aims to promote digital transformation among local sellers while providing customers with a convenient, secure, and accessible online shopping environment.

Scope

The Shopify e-commerce application is a web-based platform focusing on integrating multiple system components into a unified online marketplace. The system combines a backend service, frontend interface, and electronic payment integration to enable seamless communication between buyers, sellers, and administrators. Core functionalities include user authentication, product management, shopping cart operations, order processing, and secure online payment transactions. The application follows a client-server architecture, integrating different modules to demonstrate efficient system interaction, data exchange, and overall architectural design.

Definitions, Acronyms, and Abbreviations

Definitions

- **E-commerce:** The buying and selling of goods or services over the internet.
- **Electronic Payment (E-payment):** A method of paying for goods or services electronically through payment gateways or online banking.
- **Client-Server Architecture:** A system design where the client (frontend) sends requests to a server (backend), which processes and returns the results.
- **RESTful API:** A web service that allows communication between frontend and backend systems using HTTP requests and responses.
- **Product Catalog:** A collection of product information including names, descriptions, images, and prices displayed to customers.
- **Shopping Cart:** A feature that allows users to select and temporarily store products before completing a purchase.
- **Transaction Processing:** The execution and recording of a purchase or payment in the system.

- **User Authentication:** A security process that verifies the identity of a user before granting access to the system.

Acronyms

- **API:** Application Programming Interface
- **UI:** User Interface
- **UX:** User Experience
- **DBMS:** Database Management System
- **HTTPS:** HyperText Transfer Protocol Secure

Abbreviations

- **Admin:** Administrator – manages system operations and monitors users and transactions.
- **Seller / Vendor:** Local business or individual selling products in the system.
- **Customer / Buyer:** End-user purchasing products through the system.
- **NFR:** Non-Functional Requirements

Overall Description

System Perspective

The Shopify e-commerce application is a web-based system designed as part of a larger digital commerce environment that supports local businesses and online buyers. The system follows a client-server architecture where the frontend interface communicates with a backend service through RESTful APIs. It integrates multiple components including user management, product management, order processing, and electronic payment systems to provide a unified platform for online transactions. The application interacts with external services such as payment gateways to ensure secure and efficient financial transactions.

User Classes and Characteristics

Administrator (Admin):
Responsible for managing system operations, monitoring transactions, managing user accounts, and maintaining product listings. Admin users have advanced access privileges and basic technical knowledge.

Seller (Vendor):
Local business owners or individual sellers who manage product listings, update inventory, and process customer orders. Sellers require a simple and user-friendly interface with minimal technical complexity.

Customer (Buyer):
General users who browse products, add items to cart, and complete purchases using electronic payment methods. Customers are expected to have basic knowledge of web navigation and online shopping.

Operating Environment

The system operates within a web-based environment accessible through modern web browsers such as Google Chrome, Mozilla Firefox, or Microsoft Edge. The backend may be developed using Spring Boot (Java-based framework) providing RESTful API services, while the frontend may use React.js for interactive user interfaces. The system requires internet connectivity and may utilize a relational database management system (e.g., MySQL or PostgreSQL) for data storage. Deployment can be performed on cloud servers or local hosting environments.

Assumptions and Dependencies

Users have stable internet access and compatible web browsers.

The system depends on external electronic payment gateways for transaction processing.

The backend and frontend services must maintain proper API communication for system functionality.

Security measures such as authentication and data protection are assumed to be implemented properly.

The application is designed primarily for local marketing and may require future enhancements for scalability and additional integrations.

System Features and Functional Requirements

Feature 1: User Registration and Authentication

Description:

This feature allows users to create accounts, log in securely, and manage authentication credentials. It ensures that only authorized users can access specific system functionalities based on their roles (Admin, Seller, or Customer).

Functional Requirements:

- The system shall allow users to register using required personal information (e.g., username, email, password).
- The system shall authenticate users through secure login mechanisms.
- The system shall support role-based access control for Admin, Seller, and Customer users.
- The system shall allow users to log out securely from the application.

○ Feature 2: Product Management

Description:

This feature enables sellers and administrators to create, update, delete, and manage product listings available in the marketplace.

Functional Requirements:

- The system shall allow sellers to add new products with details such as name, description, price, and images.
- The system shall allow authorized users to update product information.
- The system shall allow deletion or deactivation of products.
- The system shall display product listings to customers in a structured format.

○ Feature 3: Shopping Cart and Order Processing

Description:

This feature allows customers to select products, add them to a shopping cart, and process orders for purchase.

Functional Requirements:

- The system shall allow customers to add or remove items from the shopping cart.
- The system shall calculate total order cost automatically.
- The system shall allow customers to place orders and confirm purchases.
- The system shall update order status during processing.

○ Feature 4: Electronic Payment Integration

Description:

This feature integrates electronic payment systems to allow secure and efficient online transactions.

Functional Requirements:

- The system shall connect to external payment gateways.
- The system shall process payments securely.
- The system shall confirm payment status and update order records.
- The system shall notify users of successful or failed transactions.

○ Feature 5: Administrative Management

Description:

This feature provides administrative tools for monitoring system activities, managing users, and overseeing transactions.

Functional Requirements:

- The system shall allow administrators to view system reports and transaction history.
- The system shall allow administrators to manage user accounts.
- The system shall allow monitoring of product listings and orders.

● Non-Functional Requirements

Performance

- The system shall handle at least 100 concurrent users without significant performance degradation.
- The system shall respond to user requests within 3 seconds under normal load conditions.
- The system shall process payments and update order statuses in real-time.

Reliability

- The system shall maintain 99% uptime, excluding scheduled maintenance.
- The system shall provide backup and recovery mechanisms to prevent data loss.
- The system shall ensure accurate and consistent order and payment processing.

Security

- The system shall implement secure authentication and role-based access control for all users.
- The system shall encrypt sensitive data, including passwords and payment details.
- The system shall protect against common security threats, such as SQL injection and cross-site scripting (XSS).

Usability

- The system shall provide a user-friendly and intuitive interface for Admins, Sellers, and Customers.
- The system shall support navigation through clear menus, buttons, and labels.
- The system shall provide error messages and guidance to assist users when needed.

Maintainability

- The system shall be modular to allow future updates or enhancements without affecting existing functionality.
- The system shall use standard coding practices for readability and maintainability.
- The system shall log system errors for easier debugging and maintenance.

Portability

- The system shall operate on major web browsers (Chrome, Firefox, Edge).
- The system shall be deployable on cloud servers or local hosting environments.

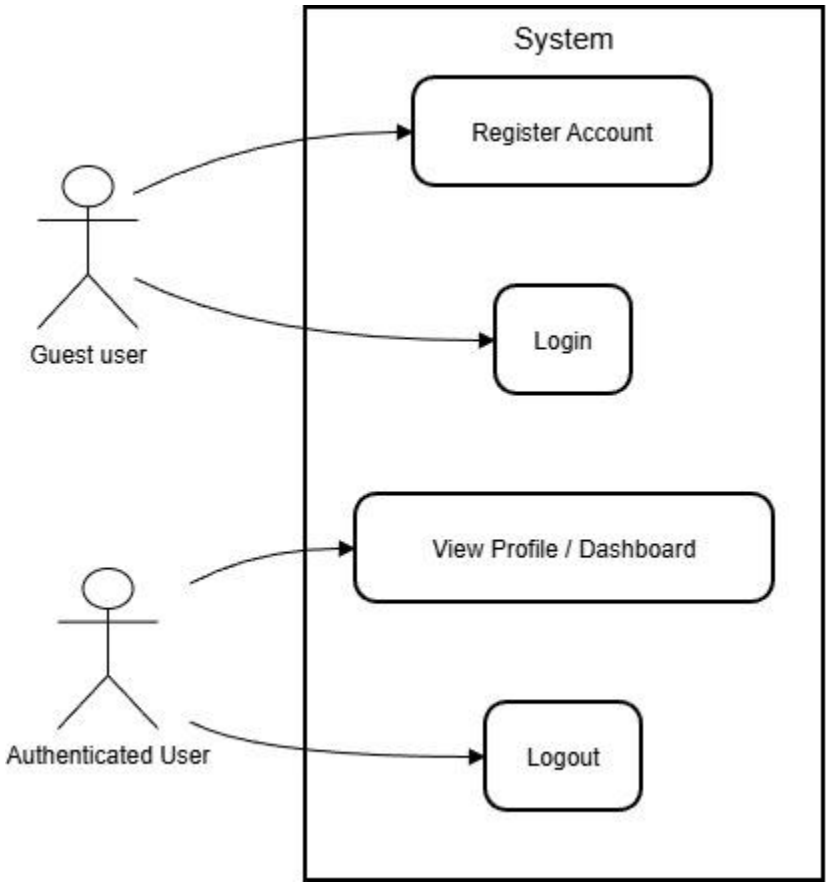
Scalability

- The system shall be designed to allow future integration with additional modules, such as mobile apps, AI recommendations, or third-party logistics.
- The system shall support increasing numbers of users and transactions with minimal system changes.

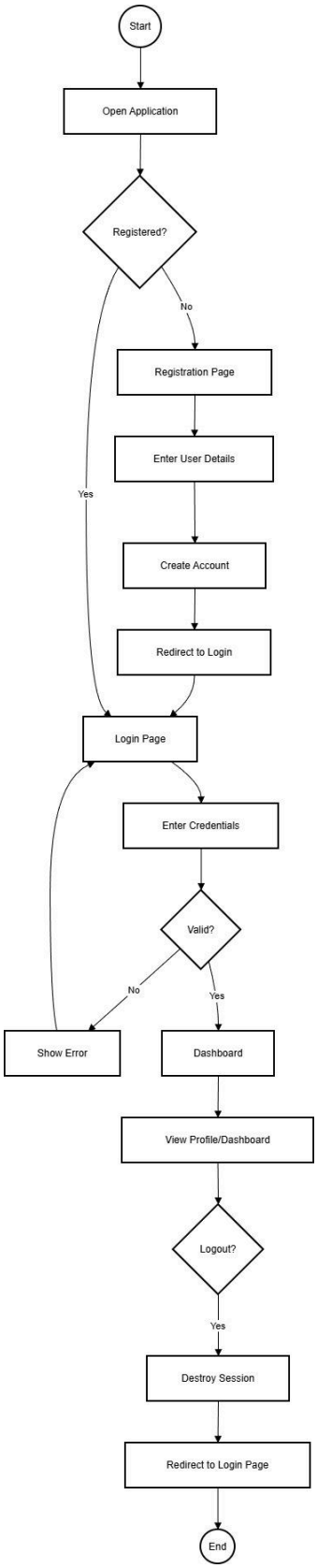
- ## System Models (Diagrams)
  - ### ERD

USER		
int	user_id	PK
varchar	username	
varchar	email	
varchar	password_hash	
datetime	created_at	
datetime	last_login	
enum	status	

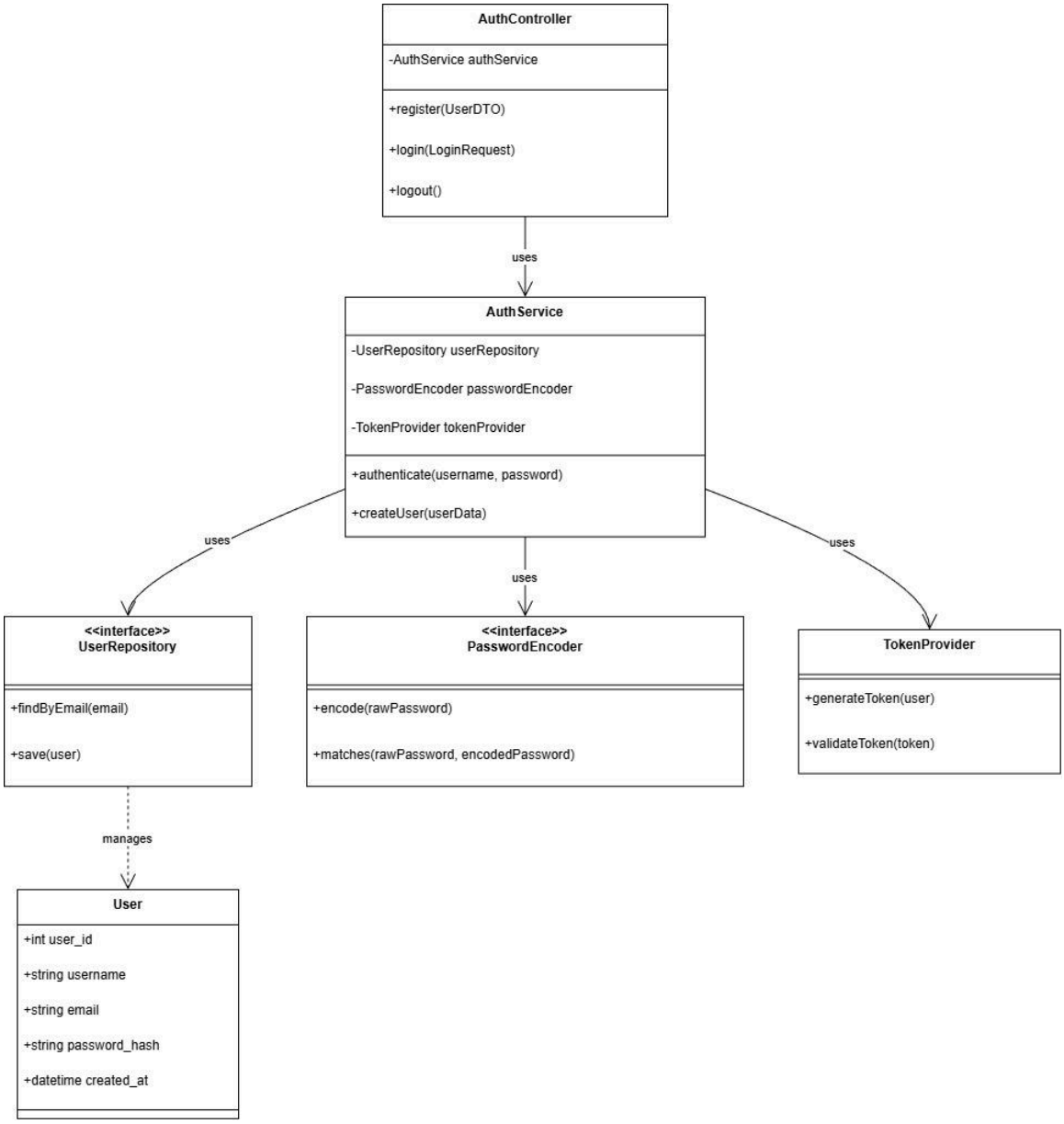
- **Use Case Diagram**



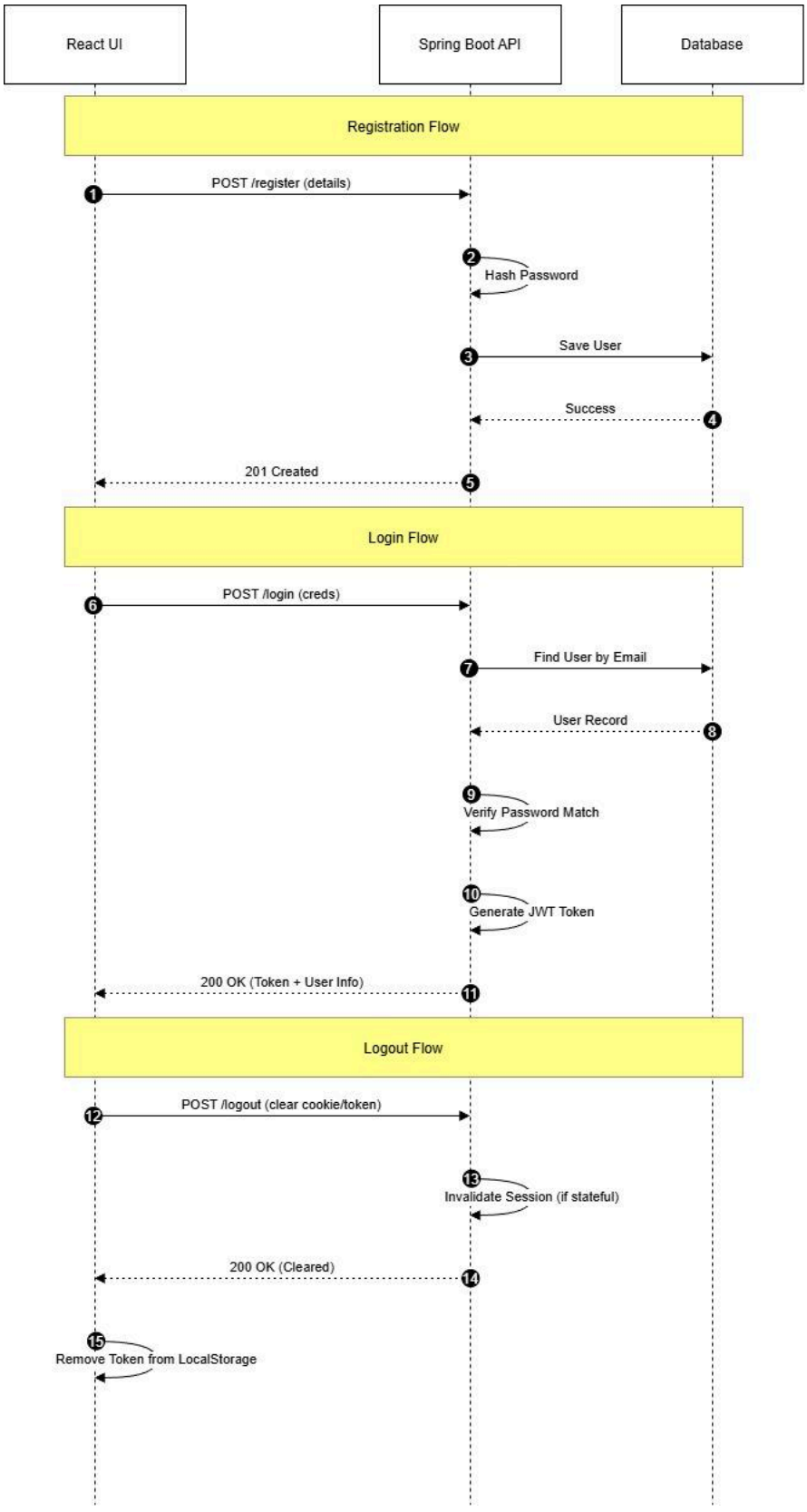
- ## Activity Diagram



- **Class Diagram**



- ## Sequence Diagram



- **Appendices**
  Include any additional information, references, or support materials.