
SOFTWARE REQUIREMENTS SPECIFICATION

for

Digital Payments Book

Prepared by :

- 1. Arpita Dubey (19ESKIT017)**
- 2. Aryan Sharma (19ESKIT018)**
- 3. Atishay Harsola (19ESKIT019)**
- 4. Gaurav Jangid (19ESKIT026)**

Submitted to : Mrs. Shalini Singhal
(Assistant Prof.)

May 30, 2023

Contents

Revision History	1
1 Introduction	2
1.1 Purpose	2
1.2 Scope	2
1.3 Technologies to be used	2
1.4 Overview	3
2 Overall Description	4
2.1 Product Perspective	4
2.2 Product Functions	4
2.3 User Classes and Characteristics	4
2.4 Operating Environment	5
2.5 Constraints	5
2.6 Assumptions and Dependencies	5
3 External Interface Requirements	6
3.1 User Interfaces	6
3.2 Hardware Interfaces	6
3.3 Software Interfaces	6
4 Functional Requirements	8
4.1 Register	8
4.2 Manage details by customer (user)	8
4.3 Manage details by retailer(admin)	9
5 Nonfunctional Requirements	10
5.1 Performance Requirements	10
5.2 Safety Requirements	10
5.3 Security Requirements	10
5.4 Error Handling	10

5.5 Software System Attributes	10
6 Other Requirements	11
6.1 Use Case Diagram	11
6.2 Entity Relationship Diagram	12
6.3 Activity Diagram	13
6.4 Sequence Diagram	14
6.5 Class Diagram	15
6.6 Component Diagram	16
6.7 Deployment Diagram	17
7 References	18

Revision History

Revision	Date	Author(s)	Description
1.0			
2.0			
3.0			
4.0			

Chapter 1

Introduction

1.1 Purpose

The purpose of the project is to maintain customers, payments and their purchases. A retailer will be an admin of the application and each customer of the retailer's shop will be the user. Customers can create their account in the payment book app by reading and agree to the terms and conditions of the shop.

1.2 Scope

This project Digital Payment Book Application has been developed in such a manner, that the future requirements of the user are met. The project is flexible to adapt the changes efficiently without affecting the present system. We are also planning to implement more features on the app. This is the future scope of our project.

1.3 Technologies to be used

1. **ReactJS** : ReactJS is a declarative, efficient, and flexible JavaScript library for building reusable UI components. It is an open-source, component-based frontend library which is responsible only for the view layer of the application.
2. **NodeJS** : NodeJs is a cross platform runtime environment and library for running JavaScript applications outside the browser. It is used for creating server-side and networking web applications. It is open source and free to use.
3. **ExpressJS** : Express is a fast, assertive, essential and moderate web framework of NodeJS. Assume Express as a layer built on top of the NodeJS that helps manage a server and routes, It provides a robust set of features to develop web and mobile applications.
4. **MongoDB** : MongoDB is an open source document database that provides high performance, high availability, and automatic scaling.
5. **Cascading Style Sheets** : Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

1.4 Overview

The implementation of Digital Payments Book starts with entering and updating master records like customer details, payment information. Any further transaction like pending payments, purchase history will automatically update the current customer details.

Chapter 2

Overall Description

2.1 Product Perspective

The proposed Digital Payments Book will take care of the each customer of the retailer's shop at any point of time. The pending payments, purchase history will update the current customer details automatically so that admin will get the update current customer details.

2.2 Product Functions

1. Administrator (Retailer) -

- Admin should be able to maintain data about customers purchases.
- Can send alerts to the customers if there is any due for payment or payment is pending for a long time.
- Increase the period for payment.
- Can get the information (status report) of any customer who has pending payment.
- Add and edit customer details and arrange customers by amount of pending payment.
- Can see payment details and pending payment of the customers.
- Can send lateness warnings to people who have exceeded deadline date.
- Can record customer details who have paid all pending payments.

2. Users (Customer) -

- Customers can create their account in the payment book app by reading and agree to the terms and conditions of the shop.
- Customers are given a provision to check their account's information.
- Once a user created their account they can login to their account by using their own credentials.
- Users will be able to see their purchase history, pending payments.
- If the user is having any doubt or complaints they can contact the retailer by using contact us service.

2.3 User Classes and Characteristics

We have 2 levels of users :

1. **User module:** In the user module, user will create and login to their accounts.

- Agree to terms and conditions
- See purchase history
- See pending payment
- Contact

2. **Administration module:** The following are the sub module in the administration module :

- Register user
- Maintain data
- Send alerts

2.4 Operating Environment

This software will work on all operating systems with a web browser with active internet connection.

2.5 Constraints

- GUI is only in English.
- Login and password is used for the identification of users.
- Limited to HTTP/HTTPS or locally during development.
- This system is working for a single server.

2.6 Assumptions and Dependencies

The success of this system depends on

1. Existence of an Internet service.
2. Admins (retailers) and Users (customers) must be comfortable with computers and have enough ability to work with the product.
3. Website interface must be friendly and easy-to-use.

Chapter 3

External Interface Requirements

Requirements refers to the needs of fabricated software to work efficiently and effectively, some of the requirements of this software are as follows:

3.1 User Interfaces

For the efficient working of the User Interface, i.e. the Front End of the system, the OS must be having at least Internet Explorer 8 installed. To login to the website.

3.2 Hardware Interfaces

For the hardware requirements, the SRS specifies the logical characteristics of each interface b/w the software product and the hardware components. It specifies the hardware requirements like memory restriction, cache size, processor, RAM etc. those are required for software to run.

1. Minimum Hardware Requirements

- Hard Disk: 20GB and Above
- RAM: 512MB and Above
- Processor: Pentium III and Above

2. Referred Hardware Requirements

- HDD 80 GB
- RAM: 512 MB
- Cache: 1 MB L1
- Cache 512 KB L2

3.3 Software Interfaces

1. **For Hosting** - Any Windows Operations System with DOS Support and Virtual Studio for development. Primarily Windows 8 having Dream Weaver Installed with a working LAN connection to mandatory.
2. **For Using** - Any type of operating system with a Least Internet Explorer Installed and having minimum of 521 kbps working LAN compulsorily.

3. **Web Server** - Operating System (Windows)
4. **Data Base Server** - MongoDB, Operating System (Windows)
5. **Referred Software Requirements** -
 - Front End: ReactJS
 - Back End: NodeJS, MongoDB

Chapter 4

Functional Requirements

4.1 Register

1. **Description :** First the user will have to register/sign up. There are two different type of users.
2. **Retailer/Admin :** The admin have to provide details about the name of shop ,address, phone number, email id.
3. **Customer/User :** The user have to provide details about his/her name, address, phone number, email id.
4. **Sign up**
 - Input: Detail about the user as mentioned in the description.
 - Output: Confirmation of registration status and a membership number and password will be generated and mailed to the user.
 - Processing: All details will be checked and if any error are found then an error message is displayed else a membership number and password will be generated.
5. **Login**
 - Input: Enter the membership number and password provided.
 - Output : User will be able to use the features of software.

4.2 Manage details by customer (user)

1. **Pending Payment**
 - Description : List of pending payment will be displaced along with data of last date.
2. **Search**
 - Input : Enter the retailer's name or shop name.
 - Output : List of payment will appear if any.
3. **Purchase History**
 - State : All the purchases of user will be displayed.
 - Input : Click on purchase history user wants.
 - Output : Lists of all purchase items along with amount paid.

4. Fine

- Input : Check for the fines.
- Output : Details about fines on different pending payments.
- Processing : The fine will be calculated, if it crossed the last date and the user did not pay if then fine will be applied by Rs 100 per day.

4.3 Manage details by retailer(admin)

1. Maintain data about purchases**2. Payment detail**

- Input : Enter the details of the customers such as names.
- Output : List of pending payments by particular customer if any.

3. Send Alerts

- Input : Enter the name of the customer whom you want to send alert.
- Output : Sending alert confirmation.

Chapter 5

Nonfunctional Requirements

Non-functional requirements make up a significant part of the specification. They are important as the client and user may well judge the product on its non-functional properties. Provided the product meets its required amount of functionality, the non-functional properties – how usable, convenient, inviting and secure it is – may be the difference between an accepted, well-liked product, and an unused one.

5.1 Performance Requirements

The system shall accommodate high number of retailers and customers without any fault. Responses to view information shall take no longer than 5 seconds to appear on the screen.

5.2 Safety Requirements

System use shall not cause any harm to human users.

5.3 Security Requirements

System will use secured database. Normal users can just read information but they cannot edit or modify anything except their personal and some other information. System will have different types of users and every user has access constraints.

5.4 Error Handling

Digital Payments Book product shall handle expected and non-expected errors in ways that prevent loss in information and long downtime period.

5.5 Software System Attributes

- Reliable
- Secure
- Available

Chapter 6

Other Requirements

6.1 Use Case Diagram

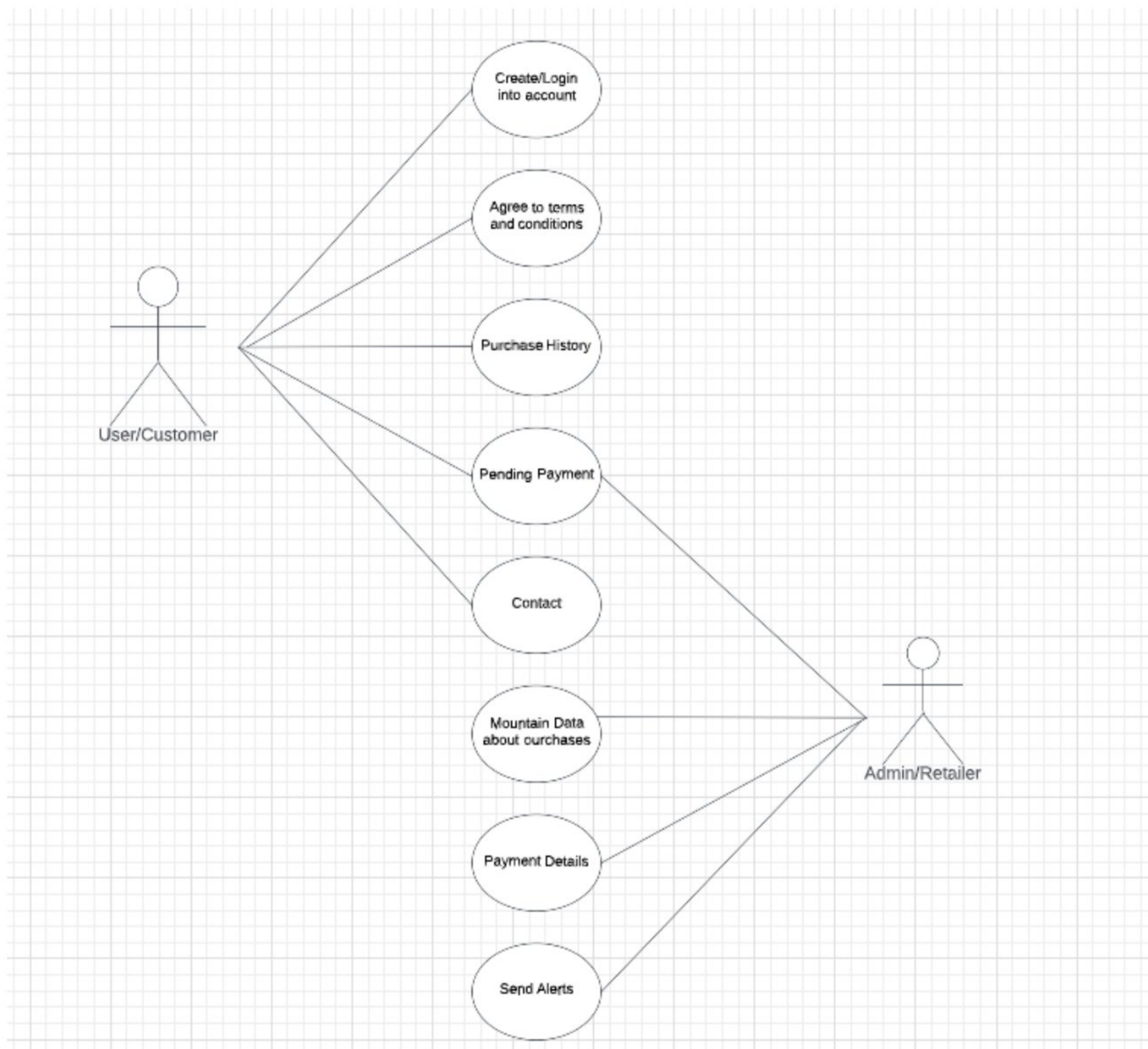


Figure 6.1: UseCase - Digital Payments Book

6.2 Entity Relationship Diagram

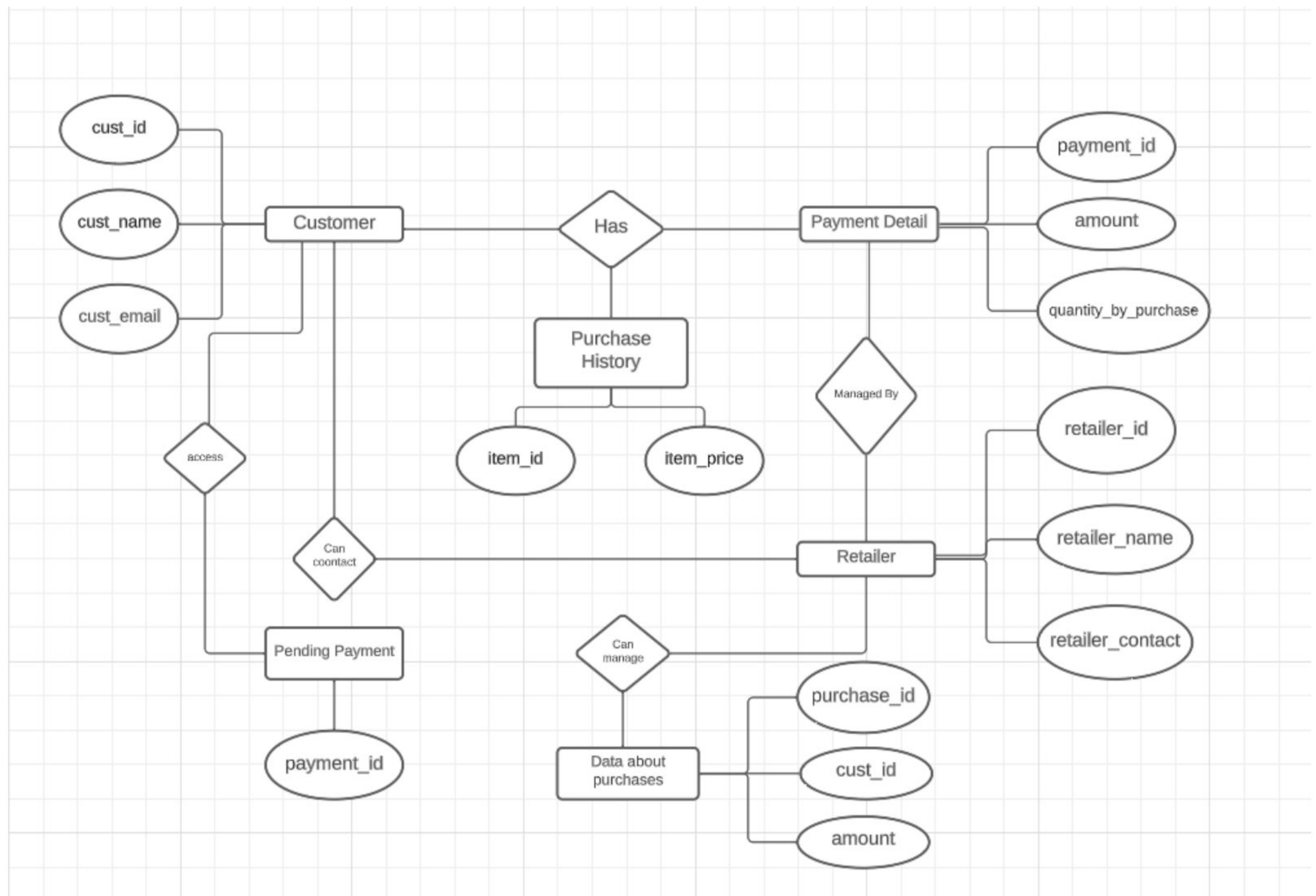


Figure 6.2: ER Diagram - Digital Payments Book

6.3 Activity Diagram

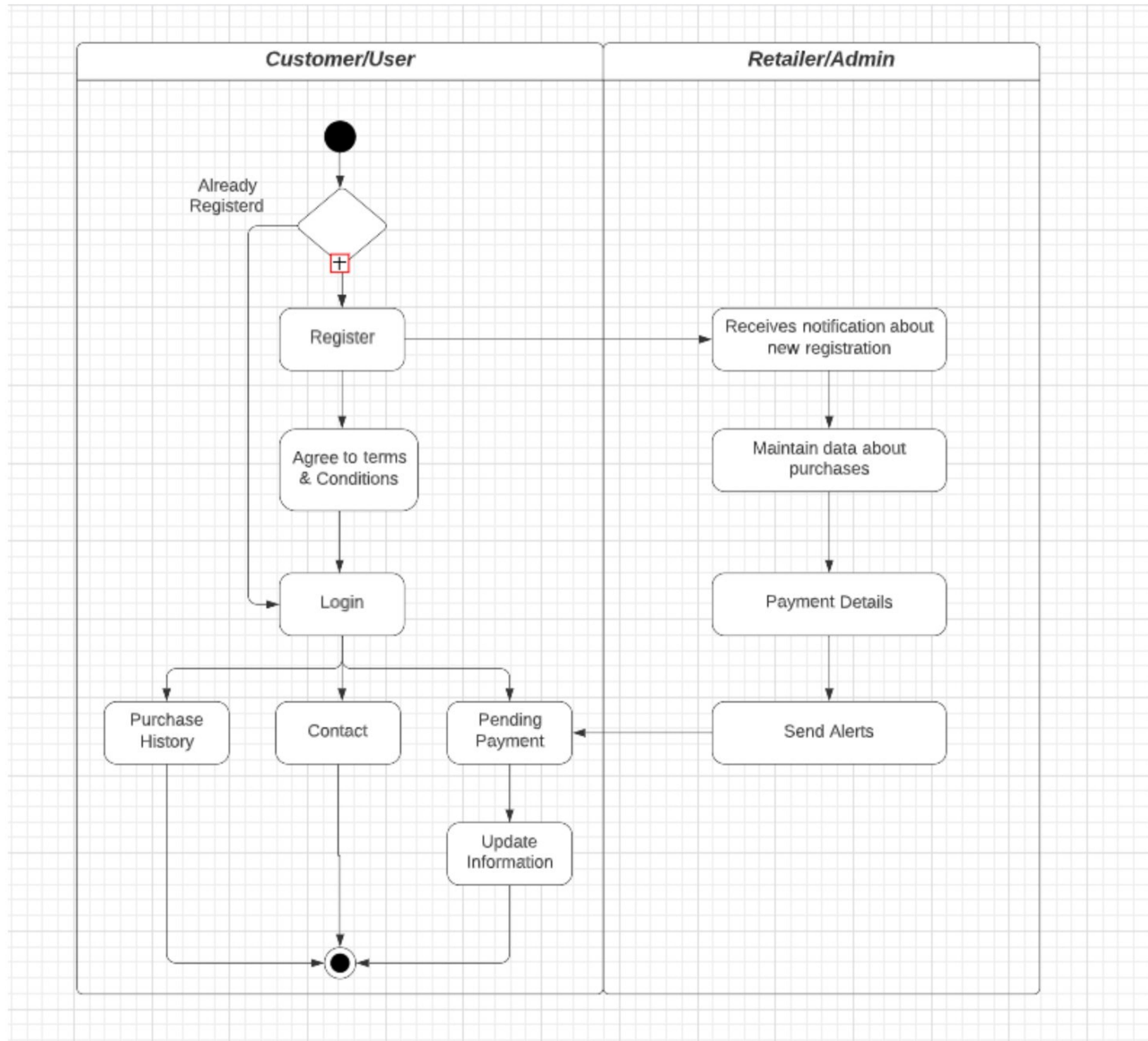


Figure 6.3: Activity Diagram - Digital Payments Book

6.4 Sequence Diagram

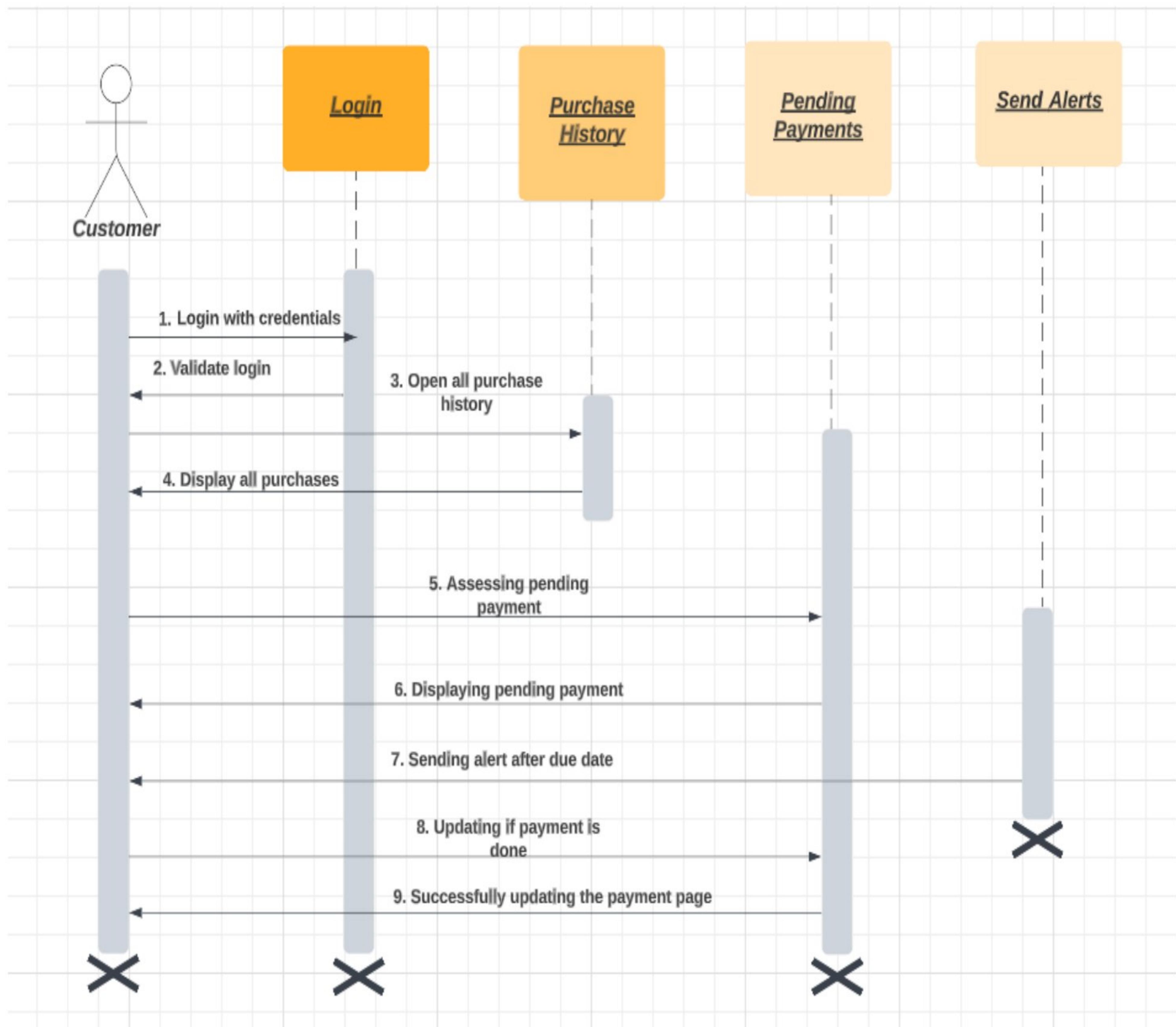


Figure 6.4: Sequence Diagram - Digital Payments Book

6.5 Class Diagram

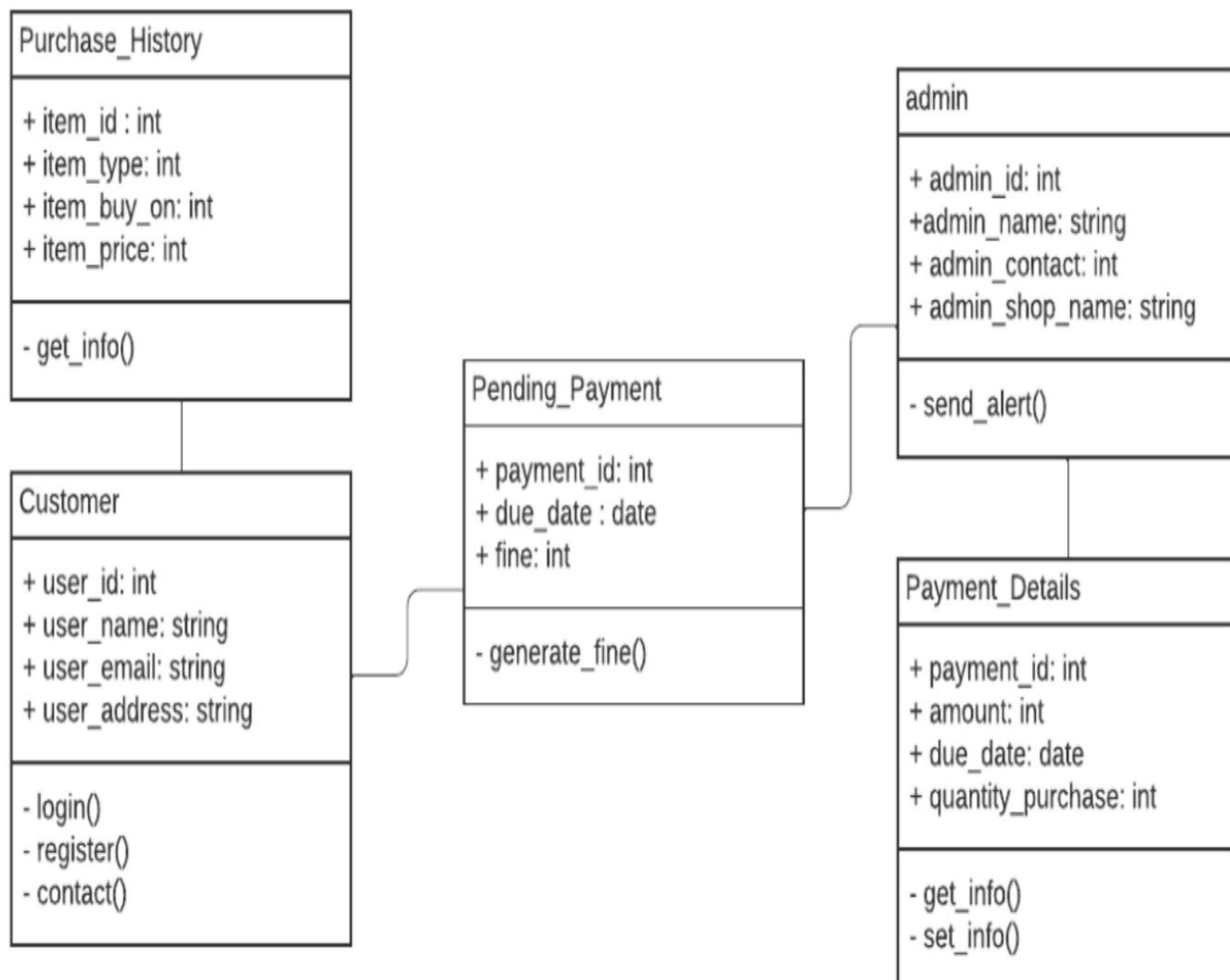


Figure 6.5: Class Diagram - Digital Payments Book

6.6 Component Diagram

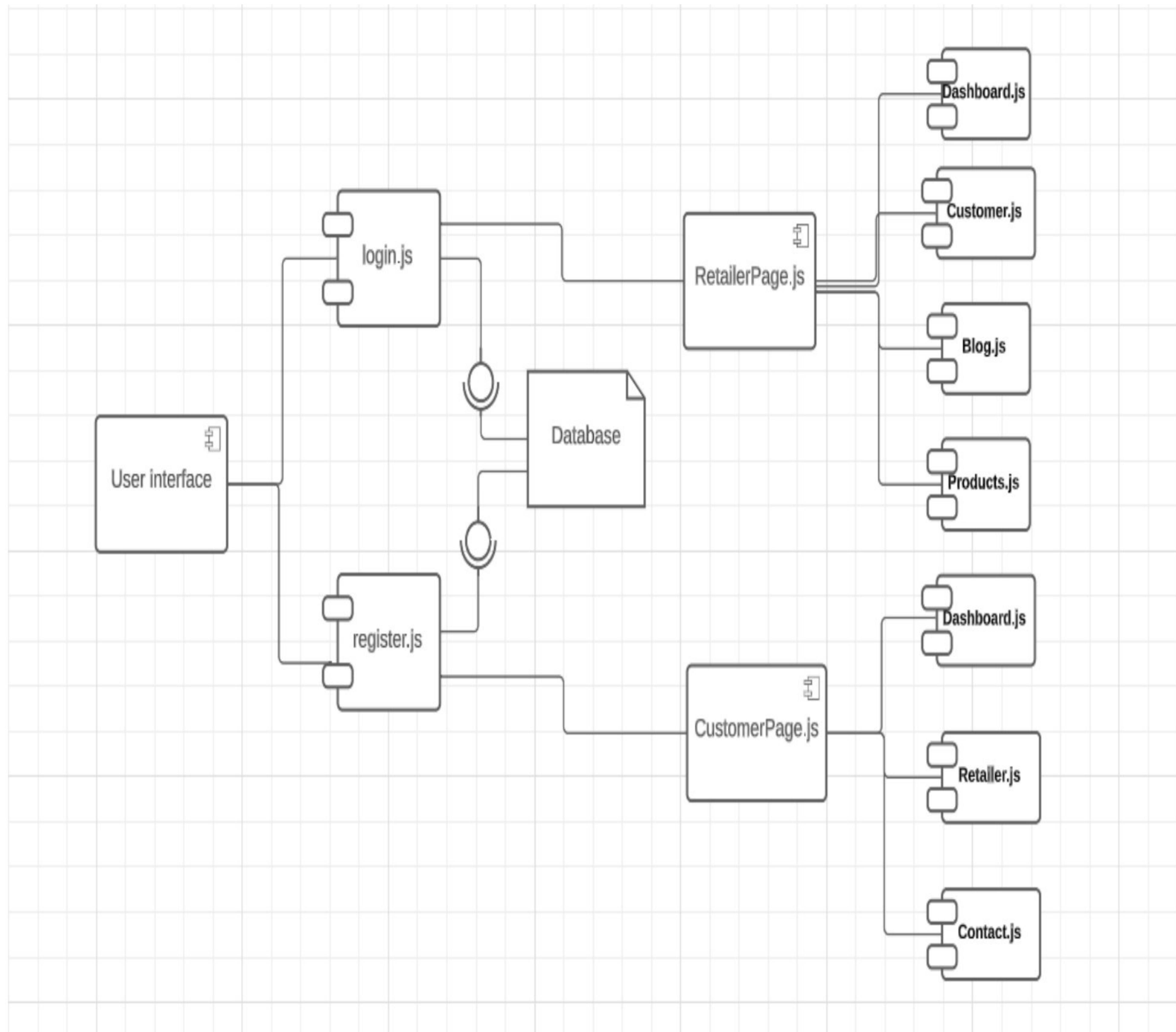


Figure 6.6: Component Diagram - Digital Payments Book

6.7 Deployment Diagram

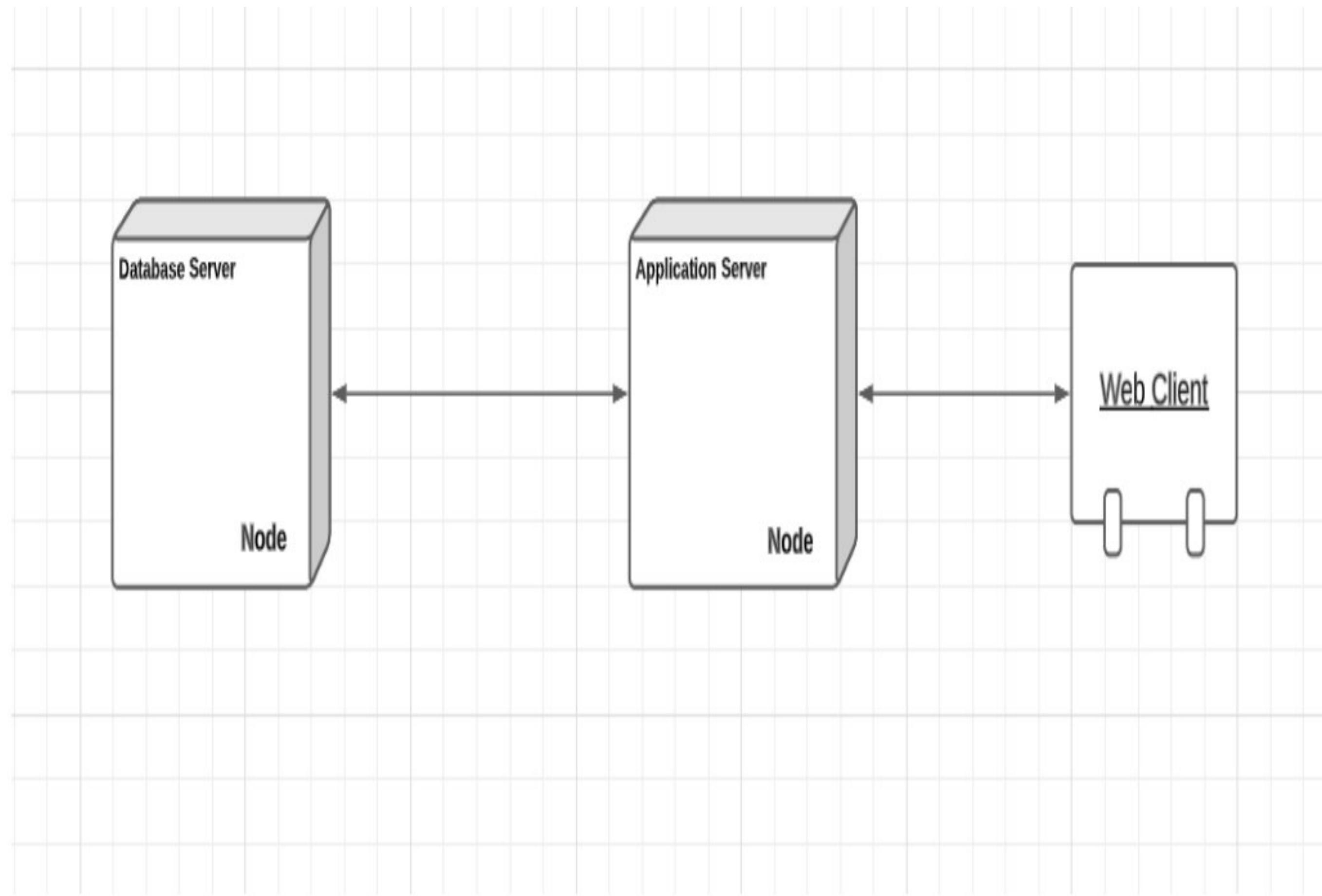


Figure 6.7: Deployment Diagram - Digital Payments Book

Chapter 7

References

1. Object Oriented Modeling and Design with UML-Michael Blaha, James Rumbaugh.
2. Software Engineering, Seventh Edition, Ian Sommerville.
3. IBM Red Books.
4. IBM TGMC Sample Synopsis.
5. IBM – www.ibm.in/developerworks.
6. Java - www.sun.com
7. Wikipedia - www.wikipedia.com
8. Database Management Systems - Navathe.
9. Complete Reference - J2EE - Keogh.