Software Requirements Specification

Video Rental System

Owais Ahmad Lone (21CS10048)

Vinayak Gupta (21CS10077)

Ishaan Sinha (21CS30064)

Contents

Chapter 1

Introduction

1.1 Purpose

The purpose of this document is to define the requirements for the Video Rental System (VRS). The intended audience of this document includes the owner of the video store, the staffs of the video store, and the customers of the VRS. It is also intended to provide guidance to the requirements team, requirements analyst, design team, and other members of the developing organization.

1.2 Product Scope

The owner of a local movie store wanted to create a new business plan where everything about renting or buying a movie was done online. Therefore, the VRS will allow the following functionality online: to search for movies, to add to cart a movie which can be later purchased or taken for rent, to either rent or buy movies. The store personnel may use the VRS to process the rented or returned movies, to add or remove movies to/from his store; video inventory and to update movie information.

1.3 Intended Audience and Reading Suggestions

- Staff: A person who is working in a video store.
- Customer: Anyone who interacts with the VRS with the potential of signing up to create a new account if already not a user or logging in if already a user.
- Functional requirement: A service provided by the software system.
- Non-functional requirements: A constraint on the system or how the system is developed.

- Owner: The person with administrative power in the VRS.
- SRS: This Software Requirements Specification document.
- Timer: An event that causes an action to occur at a specified time.
- Movie ID: Unique number given to each movie unit.
- Statement: Printout showing an individual rental by a customer. For each rented movie, it displays the movie ID, movie name, and due date. It also shows rental charges.
- VRS: Video Rental System (the system under development)

1.4 Overview of rest of SRS

Chapter 2 of the SRS describes the product in more detail. Chapter 3 provides a complete list of the functional requirements of the intended system. Chapter 4 shows the class diagram, and Chapter 5 the use case diagram. Chapter 6 provides the non-functional requirements. The appendices appear next.

1.5 References

To be added

Chapter 2

Overall Description

2.1 Product Perspective

The **VRS** is a web-based system. The overall system consists of two interfaces, one of the owner's email system, and the other browsers used by VRS customers and staffs with staffs having some additional permission and accessibilities. The system provides a secure environment for all financial transactions and for the storing and retrieving of confidential user information.

2.2 Product Functions

The VRS allows customers to search the movie inventory provided by this store. To rent movies through the VRS, one must register as a user and create an account using the VRS. Once after creating an account and logging into the VRS, the VRS provides the functionality for renting or buying movies and even adding or removing any movie to/from cart which can be purchased at some later stage of time.

The staffs of the store use VRS to process the timeline of rented movies as to when was it rented and till which date it is available to that customer. The owner of the store uses VRS to add new movies into the system, remove movies from the system, and modify movie information.

2.3 User Classes and Characteristics

The two main groups of VRS users are customers and store personnel.

A customer is someone who has registered and created an account or who is about to create one after signing up with VRS. A customer can rent or buy movies and pay fees online. As with a customer, these activities require no product training since the level of technical expertise and educational background

of a member is unknown. The only skill needed by a member is the ability to browse a website.

The store personnel are divided into two groups: the staff-level personnel and owner-level personnel. Their educational level is unknown and both group needs little to no training.

2.4 Operating Environment

The project was developed using **django 4.1.8** which has a pre-developed prototype of a online site written in **Python 3.9, HTML5, CSS5** and **Javascript**. The movies database was populated using **SQLite**

2.5 Design and Implementation Constraints

This system provides web access for all customer and staff functions. The user interface will be intuitive enough so that no training is required by customers or store personnel. All online financial transactions and the storage of confidential user information will be done in a secure environment. Persistent storage for rental, purchase and movie inventory information will be maintained.

2.6 Assumptions and Dependencies

No assumptions and dependencies affect this system at this time.

Chapter 3

External Interface Requirements

3.1 User Interfaces

This Rental Site will first ask the user whether he/she is an admin, a customer or a staff. This page looks like as follows:

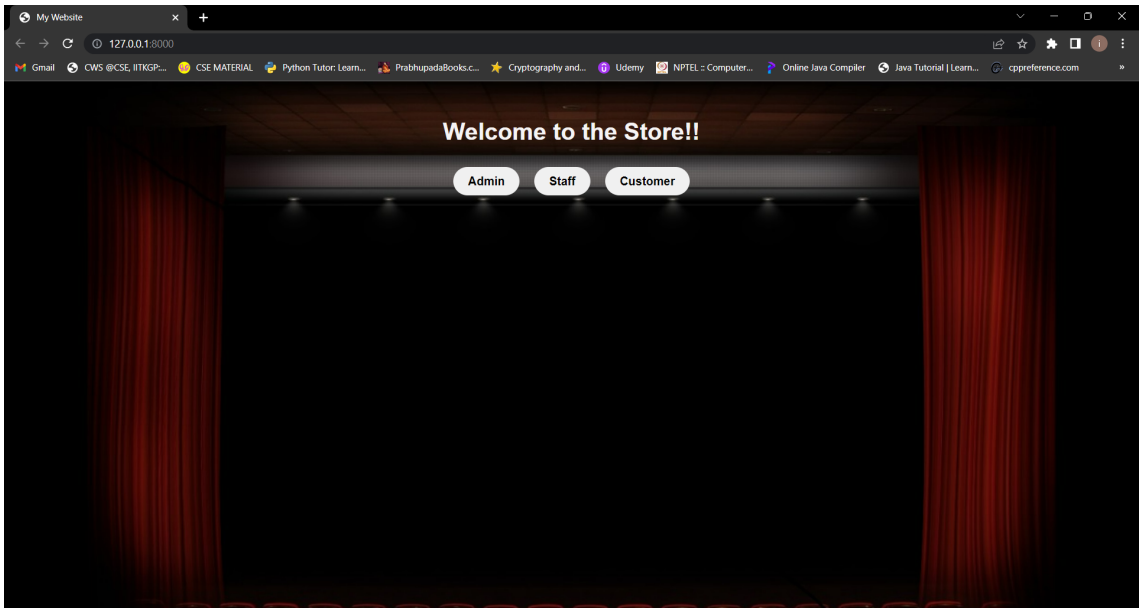

Figure 3.1: Opening Page

Based upon the response the site will redirect the user to the respective login

pages. Now specifically in the case of customer login there would be one option for the user if he or she is visiting the site for the first time, to sign up. The page look like as follows: After clicking the link highlighted in the box it will

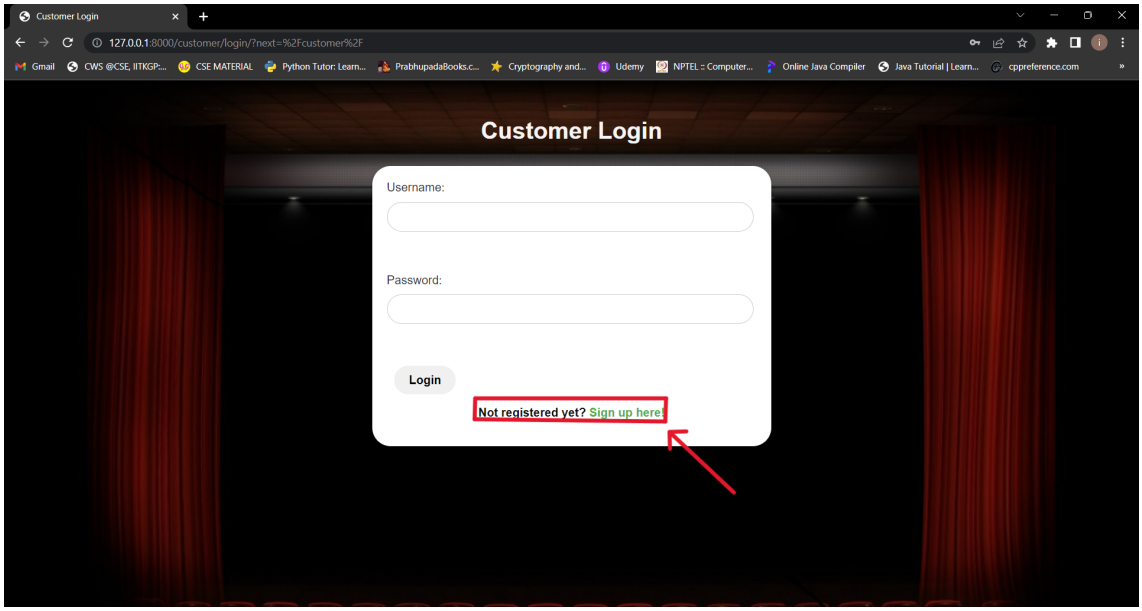

Figure 3.2: Login

redirect to sign-up page for the user to sign-up and create new account.

Basically the system shall register the user after he or she has filled the fields correctly and this credentials of user will get saved to the database. This page also generates corresponding error messages.

3.2 Hardware Interfaces

- The system allows the customer or member to search movie information.
- The system allows customers to rent movies.
- The system also allows customers to purchase movies
- The system allows a staff to process returned movies.
- The system allows the owner to add new movies to the movie inventory.
- The system allows the owner to remove movies from the movie inventory.

3.3 Software Interfaces

For Movies Poster and Description we used **CSS** and **HTML**, and for the list of movies to get added to the cart of any customer we used **Javascript** and **Django**. Even for customer database **Javascript** and **Django** are used. For modelling real time renting and buying transactions we have used **PayPal sandbox accounts.**

Chapter 4

Other Non-functional Requirements

4.1 Description

- The system shall provide web-access for all customer, staff and owner functions.
- The system shall provide persistent storage for rental, purchase and movie inventory information.
- The system shall provide an intuitive user interface that requires no training.

Chapter 5

System Features

5.1 Renting/Buying

- **User Class- Customer**
- **Object Class- Movies**
- **Functionalities**

  Customer: Add, Remove, view and get details.

  Movies: Add, Remove , view purchased history of the movie

5.2 Browsing through the list of movies

- **User Class**- Customer, Staff and Admin
- **Object Class**- Movies
- **Functionalities**- The description and poster of the movie would be visible along with the prices and option to add to cart.

5.3 Purchasing and Renting History

- **User class**- Customer, Staff and Admin
- **Object class**- Movies

5.4 Payment and Billing

- **User class**- Merchant(The Site Owner)
- **Payer/buyer**- Customer

- **Functionalities**- All transactions happening with the help of **Paypal Sandbox Accounts.**

Chapter 6

Appendix

6.1 Class Diagram of the Project

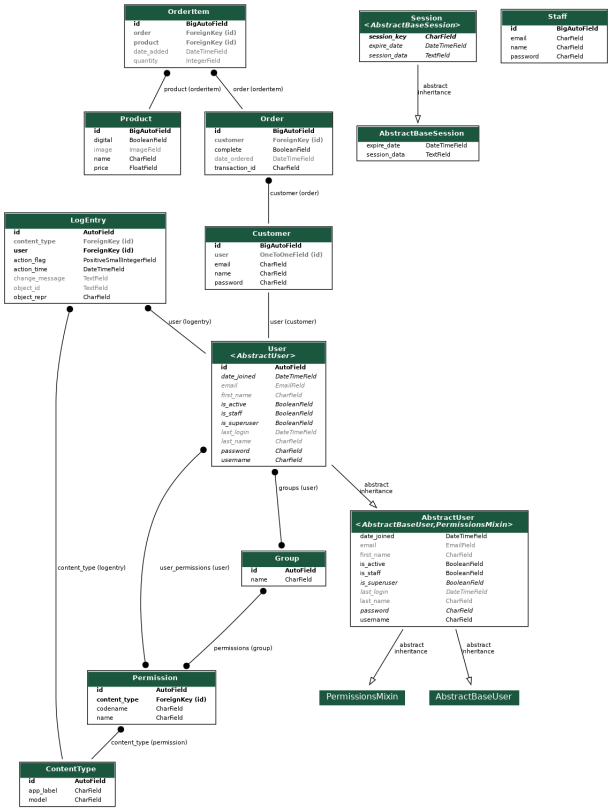

Figure 6.1: Class Diagram of the Project

6.2 Use Case Diagram

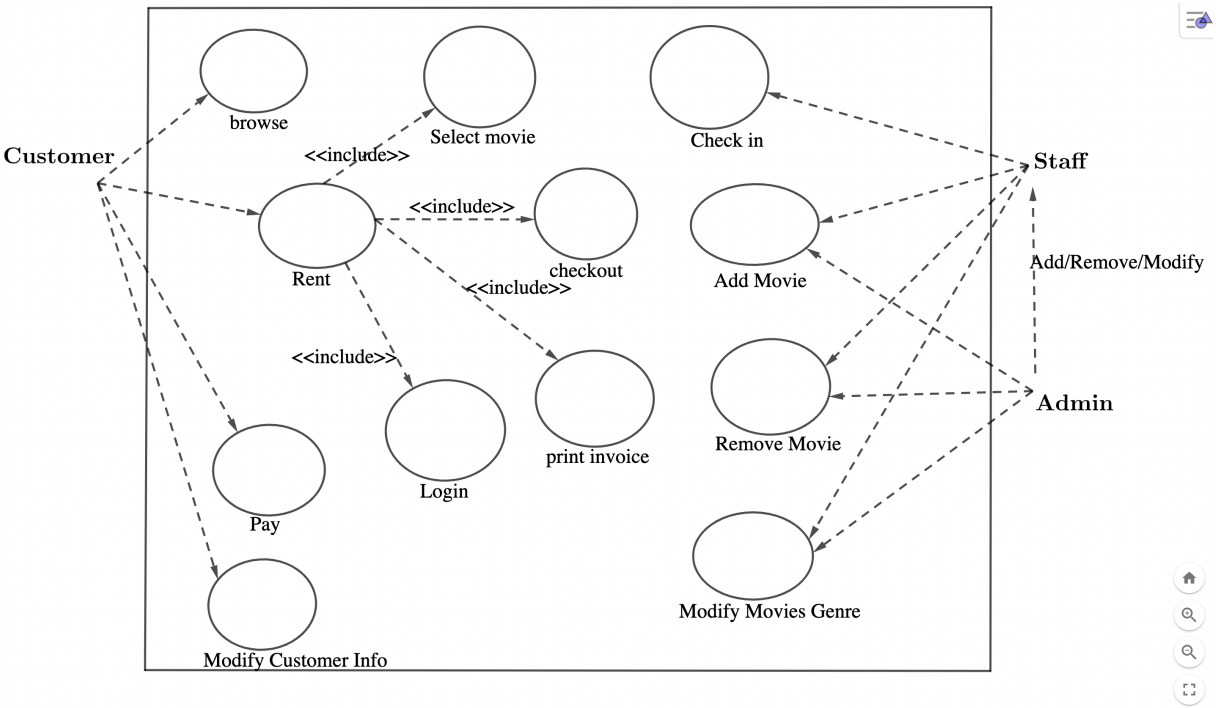

Figure 6.2: Use Case Diagram

********* END *********