

Software Requirements Specification

for

Video Rental System (VRS)

Version 1.0 Approved

Prepared By:

20CS10076 Vishal Ravipati
20CS30033 Monish Natarajan
20CS30066 Rohit Ranjan

Indian Institute of Technology, Kharagpur
16.03.2022

Table of Contents

Table of Contents	1
1. Introduction	2
1.1 Purpose.....	2
1.2 Project Scope.....	2
1.3 Acronyms and Abbreviations.....	2
2. Overall Description	2
2.1 Product Perspective.....	2
2.2 Product Features.....	3
2.3 User Classes and Characteristics.....	3
2.4 Design and Implementation Constraints.....	3
2.5 User Documentation.....	4
2.6 Assumptions and Dependencies.....	4
3. System Features	4
3.1 Secure Login.....	4
3.2 Searching of Titles.....	5
3.3 Recommendation System using Data Science.....	5
4. External Interface Requirements	6
4.1 User Interfaces.....	6
4.2 Software Interfaces.....	7
5. Other Nonfunctional Requirements	7
5.1 Performance Requirements.....	7
5.2 Safety and Security Requirements.....	7
5.3 Software Quality Attributes.....	7

1. Introduction

1.1 Purpose

This SRS specifies the software requirements of Video Rental System, henceforth referred to as VRS. This is the first release of the software and the SRS provides the full description of the product.

1.2 Project Scope

The purpose of the VRS is to provide a platform for customers to buy and rent movies, and also provides backend support to staff to maintain the platform. The product is expected to utilize the interests and trends of the customers to suggest appropriate movies to customers, provide a payment platform and maintenance functionality. The product will incorporate data science to make personalized suggestions to customers and predictions about stock of movies in the catalog.

1.3 Acronyms and Abbreviations

VRS: Video Rental System

SRS: Software Requirements Specification

SQL: Structured Query Language

K-NN: K Nearest Neighbors

2. Overall Description

2.1 Product Perspective

The SRS specifies the features of the VRS, which is a new, self-contained product. The system hopes to create a smooth platform for customers to buy and rent movies, staff to maintain the platform. The product will also incorporate data science to streamline the process of movie suggestions and catalog maintenance.

2.2 Product Features

- The product provides customers with a means to buy or rent movies of their choice.

- It allows customers to surf through a catalog of movies, providing information about the movies(basic plot, genre), allowing them to sort by ratings, genre, etc and search for movies based on keywords.
- It provides a platform for purchase of movies, generating invoices for reference.
- It provides backend access to staff and managers to maintain the catalog of movies, updating records, stocking up movies, and view information about renting of movies.
- The product uses data science to provide suggestions to customers based on their view/purchase history and trends in general popularity.
- Data science is also used to make predictions about the duration of current inventory.

2.3 User Classes and Characteristics

There are three broad user classes for VRS. These are :-

1. **Customers:**

These will have access to the frontend. They are the focus of the product and they are expected to buy/rent movies. The product is also expected to facilitate the smooth purchase and returning of movies. The user interface is to be easy to use and understand for the convenience of the customers who might not have any experience. They have no access to the backend.

2. **Staff:**

These will have restricted access to the backend. They will be given the permission to maintain the catalog of movies, which involves checking stock and order durations, clearing dues and stocking up on movies in low quantities. The product will be expected to provide alerts to the staff in case of defaults or movies running out. As this involves a lot of manual work, the product is expected to provide as much automation as possible through alerts, notifications and predictions about lifetime of stock.

3. **Managers:**

Managers will have full and unrestricted access to the backend. They will have the power to overlook the inventory, change the movie catalog, including removing movies, changing genres and also to perform audits of the finances of the company using the product. Since this class has immense power, it will also have significant security protection.

2.4 Design and Implementation Constraints

The software's efficiency is dependent on the manual work of maintaining the catalog of movies to be done by the staff. The software is also constrained by the limited working scale, hence requiring smaller databases of movies.

2.5 User Documentation

The software will come with a user manual that will guide users in using the website in a proper manner.

2.6 Assumptions and Dependencies

The recommender system based on data science may not be entirely accurate or relevant to the customers' interests, nor can staff get exact predictions of the number of days for the movies to remain in stock. These systems are meant to be for convenience and not to be strictly adhered to.

3. System Features

3.1 Secure Login

Our program provides a secure password protected login for both the User and Company Employees. We implement the same using SQL database management and secure password hashing techniques.

3.1.1 Description and Priority

Secure Login feature is of high priority as a user's account may contain sensitive personal information and on the other hand the employee's access provides backend access to change features of the software.

3.1.2 Stimulus/Response Sequences

On opening the program, the user will be prompted to select the service interface from between Consumer Access and Employee Login. Employee login is further divided into Staff Access and Manager Access. Once the interface is selected the user will be prompted to enter their username and password. Upon successful login, the user will be prompted to the respective interface.

3.1.3 Functional Requirements

Secure Login feature is implemented by matching the entered username and password combination with the records in the database. In case of a mismatch of the same, the user will be prompted to select a forgot, username/password option wherein a user upon entering his/her email-id can send a reset login details link to their respective emails.

REQ : SQL Database Management Language

A suitable SQL database management framework will be required for storing a users login and other account details

3.2 Searching of titles

Our program would provide a robust interface to search for titles based on name, genre, and movie description.

3.2.1 Description and Priority

Title search is a key feature with medium to high priority. Such a feature is necessary for ease of selection of titles and provide a great user-program interaction.

3.2.2 Stimulus/Response Sequences

After login, the user interface will contain a search bar to find movies name, genre, and movie description.

3.2.3 Functional Requirements

This feature is implemented using appropriate search algorithms and movie data which is stored in the database.

REQ 1: SQL Database Management Language

A suitable SQL database management framework will be required for a user's past purchase history and database of movies.

3.3 Recommendation System using Data Science

Our program would provide consumers movie title recommendations based on data science analysis techniques such as clustering and K-NN.

3.3.1 Description and Priority

Recommendation Systems is a key feature with medium to high priority. Such a feature is necessary to retain existing customers and to improve the overall user experience of the platform.

3.3.2 Stimulus/Response Sequences

Once a user has added a given title to their 'cart', they user will be prompted to select more titles from a list of recommendations obtained on applying K-NN algorithm to show similar titles to the one selected. A user could also be recommended based on his/her past movie selections which are stored in the SQL database.

3.3.3 Functional Requirements

Recommendation is implemented by using Clustering and KNN techniques on user data which is stored in the database.

REQ 1: SQL Database Management Language

A suitable SQL database management framework will be required for a users past purchase history and database of movies

REQ 2: Data Science Library

scikit-learn 1.0.2 library in python will be used to implement k-means clustering algorithm

4. External Interface Requirements

4.1 User Interfaces

Our User Interface will be implemented using Flask 2.0.3 . Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. Our backend will be implemented in Python 3.7 and the front-end and back-end will communicate seamlessly over a local server that Flask helps us to host.

The user interface can be divided into 3 major sections:

1. Landing Page :

The landing page is the first page that any stranger on the internet lands at for the first time. Here the netizen will be given options to login as different stakeholder classes or one can choose to browse the landing page without logging in. The landing page contains browsing controls through the entire movie library available on the platform. Further there will be sections for displaying the top movies being bought and rated in real time. There will also be a blog section where users can post blogs where they curate must-watch movies of different genres.

2. Login/Sign-Up Page :

Users on clicking the login button on the landing page are brought here. This page is implemented using security enabled forms that Flask provides using its sister extensions. Users

will be given the option to sign in as one of the stakeholder classes. User ID and Passwords will be asked for. Further, the interface will allow users to reset passwords in case they are forgotten and verification of authenticity will take place using a set of security questions. New users will be allowed to sign up as well.

3. Personal Dashboard :

Once a user logs in successfully, he is redirected to this page. The personal dashboard again has different variants for the different stakeholder classes. A particular stakeholder will only be able to view and take actions from the dashboard depending on the class rights. Each of these particular actions/rights will be displayed in separate rectangular panels which will allow us to just hide the non-permitted panels for a particular stakeholder.

4.2 Software Interfaces

Our backend software will run on Python 3.7 . We will use an object oriented programming paradigm as much as possible. Further, our backend will communicate with both the Flask server for frontend and SQL server for databases. Whenever a user logs in, an object of the particular class is instantiated with required data-fields fetched from the SQL databases.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Our VRS is built up on 3 major modules namely the backend, the frontend and the databases. The frontend is chosen to be hosted as a web application because it allows seamless cross-platform usage. Further the entire premise of a Video Rental System depends on it being online and accessible from anywhere and anytime. Python and SQL are chosen to work with given their proven performance benefits and seamless compatibility mutually.

5.2 Safety and Security Requirements

Databases are really crucial for our application. We implement all the security standards both for our databases by encrypting them and our frontend forms using Flask's inbuilt form encryption. Further details can be found in the Secure Login System Feature section.

5.3 Software Quality Attributes

Our VRS promises utmost quality and robustness. Both ease of usage and functional completeness have been ensured at each step. There is a clear distinction between rights of the

different stakeholder classes which ensures no misuse/loss of business and data. The separate hosting of the movies and auditing allows for seamless changes in the library. Further, the admin is given rights to delete entries and change rights of other classes as well. All these features combined make our VRS complete on all fronts of quality.