SOFTWARE DESIGN DOCUMENT

Recommender System

Group Members

IRM2016002

IIM2016003

ITM2016004

ISM2016006

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Purpose

The purpose of this document is to build a recommender system for movies which incorporates collaborative and content based filtering.

1.2 Scope

The scope of this project is to provide an easy to use software which recommends movies to user as per his watch list or enables him/her to be suggested movies as per users with similar tastes.

1.3 Overview

This document provides an overview of design and architecture of the project. The document contains all the functionalities of the software which we intend to prepare and some screenshots of the accuracy graph which we plotted with sample dataset. The dataset used is from Movie lens.

1.4. References

[1] Greg Linden, Brent Smith and Jeremy York Amazon.com Recommendations: Item-to-Item Collaborative Filtering

[2] Jianming He and Wesley W. Chu A Social Network Based Recommender System (SNRS)

[3] Gediminas Adomavicius and Alexander Tuzhilin Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions

[4] Thanisa Numnonda A real-time recommendation engine using lambda Architecture

[5] Nikos Manouselis, Hendrik Drachsler, R iina Vuorikari, Hans Hummel, Rob Koper Recommender Systems in Technology Enhanced Learning

[6] Emanuel Lacic,Dominik Kowald and Elisabeth Lex Neighborhood Troubles:
On the Value of User Pre-Filtering To Speed Up and Enhance Recommendations

[7] Yanxiang Huang, Bin Cui,Wenyu Zhang, Jie Jiang, Ying Xu TencentRec: Real-time Stream Recommendation in Practice

[8] Raymond J. Mooney, Loriene Roy Content-Based Book Recommending Using Learning for Text Categorization

[9] Marko Balabanovic and Yoav ShohamContent-Based, Collaborative Recommendation

[10] Badrul Sarwar, George Karypis, Joseph Konstan, and John RiedlAnalysis of Recommendation Algorithms for E-Commerce

[11] Chhavi Rana and Sanjay Kumar JainA study of the dynamic features of recommender systems

[12] Ago Luberg, Tanel Tammet and Priit JrvSmart City: A Rule-based Tourist Recommendation System

[13] Francesco Ricci, Lior Rokach, and Bracha ShapiraRecommender Systems:
Introduction and Challenges

[14] Zi-Ke Zhang, Chuang Liu, Yi-Cheng Zhang and Tao Zhou Solving the cold-start problem in recommender systems with social tags.

2. SYSTEM OVERVIEW

It is a basic system to recommend to recommend top movies based on the calculated scores using a weighted rating formula. We could have simply used movie rating as our recommendation metric. However this has a few caveats.

1) It doesn't take the popularity of a movie into consideration. For example, a recently released local movie with a relatively small size (e.g. 30) of audience and voters (who voted this movie 4.0) would be considered better than a movie with a rating of 3.5 with voters over 100,000.

2) It will in general always prefer movies with extremely high ratings and will not wait for the ratings to regularize after a sufficient viewership (votes) have been achieved.

Taking this into consideration, a weighted rating system that takes into account both the number of voters as well as the respective ratings of a movie would prove to be better. An example of this would be IMDB's Top 250 Chart.

3. SYSTEM CHARACTERISTICS

The System characteristics are namely:-

- ➢ to be scalable and easily maintainable in the future
- ➢ to provide security features to protect data
- ➢ a large number of concurrent users
- ➢ the nature of the interface to the users of the system
- ➢ To view the previous run datasets
- ➢ To download previous run datasets.

4. SYSTEM ARCHITECTURE

4.1 UML Diagrams

There are mainly 9 UML diagrams which are found to be sufficient in order to describe software's architecture and design.

So, following are the various diagrams related to the project :

Use Case View :

Fig. 4.1 Use Case Diagram

Activity Diagram :

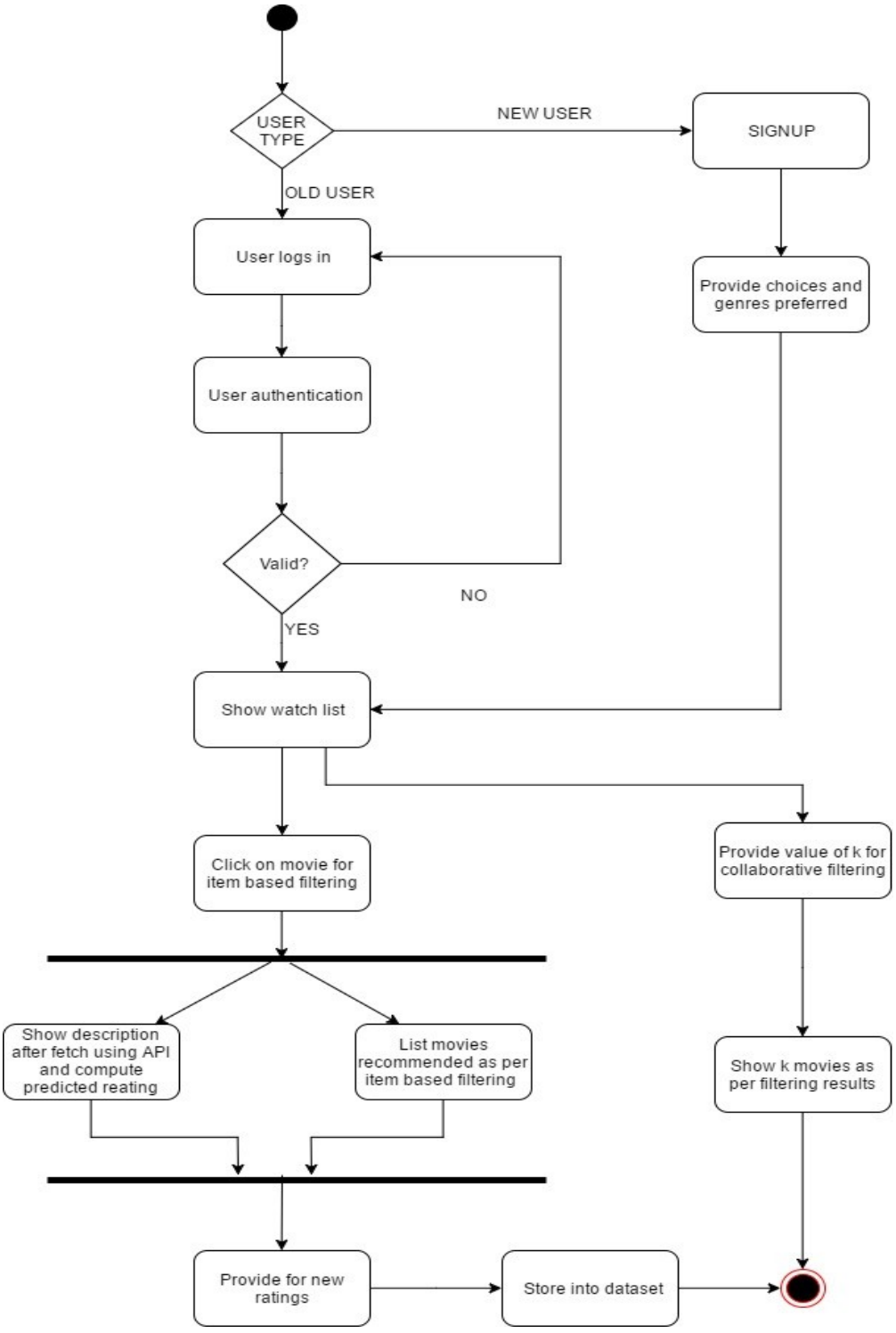

Fig. 4.2 Activity Diagram

Sequence Diagram :

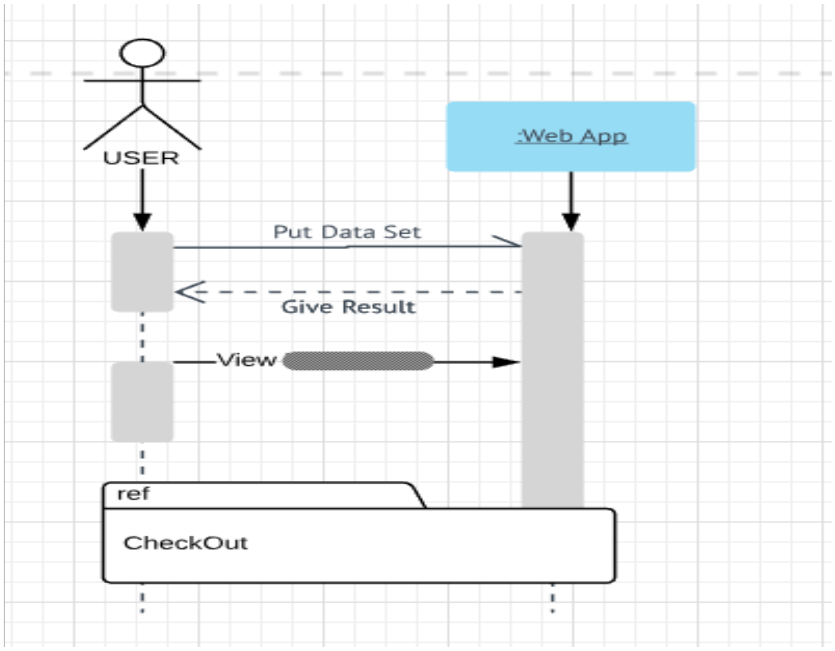

Fig 4.3 Sequence Diagram

ARCHITECTURAL DIAGRAM :

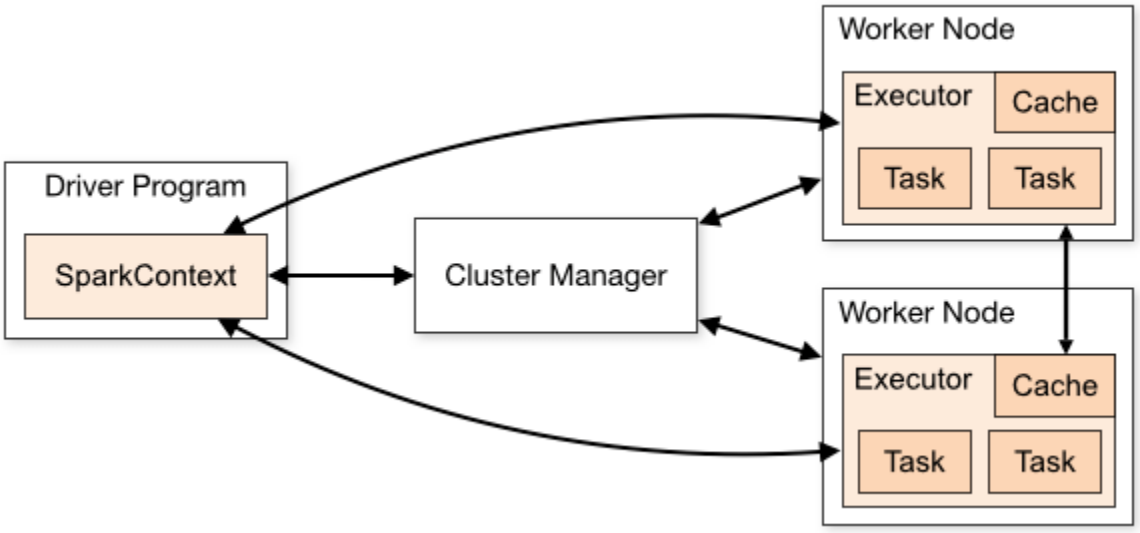

Fig 4.4 Architectural Diagram

State Diagram :

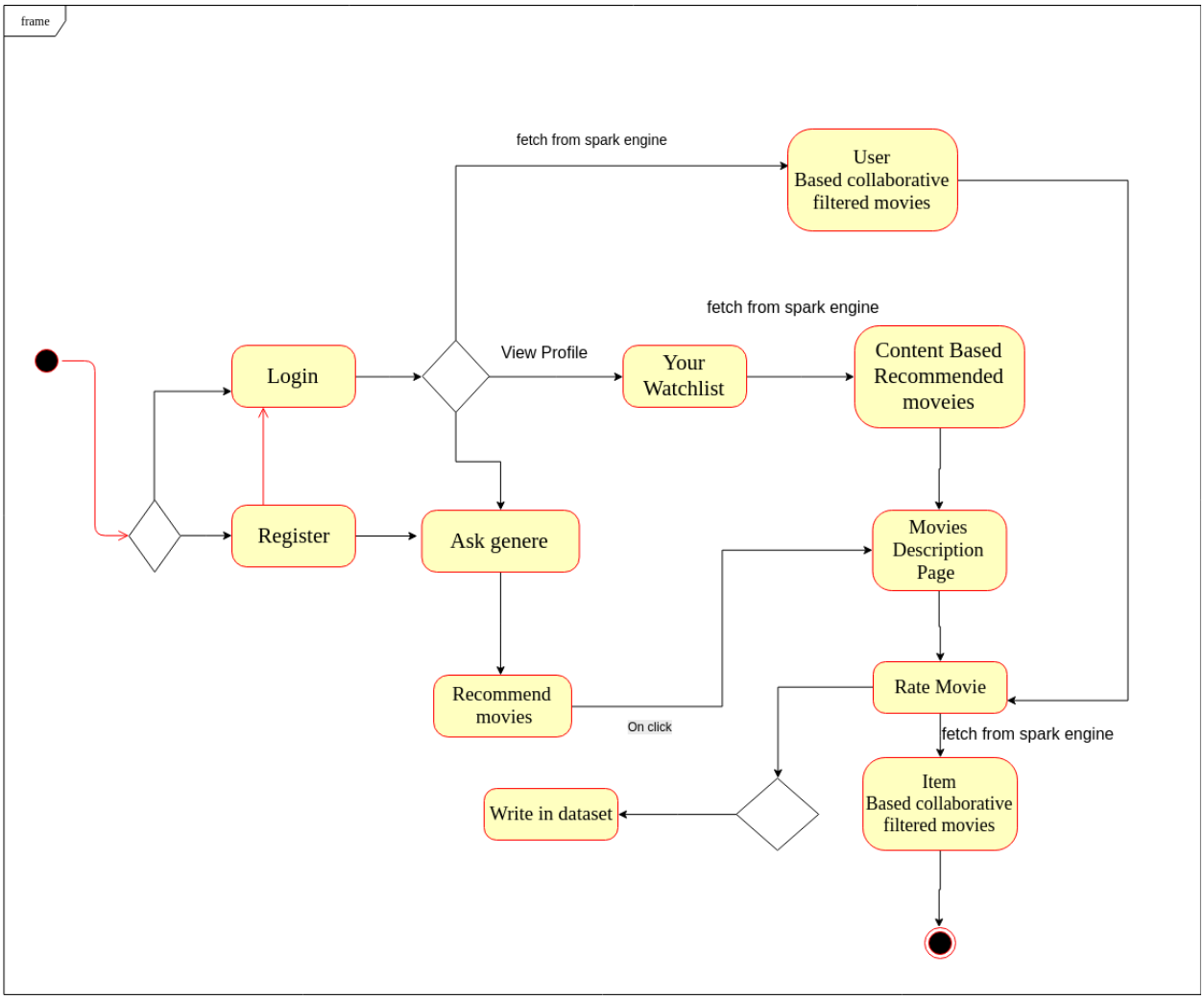

Fig 4.5 State Diagram

COMPONENT DIAGRAM :

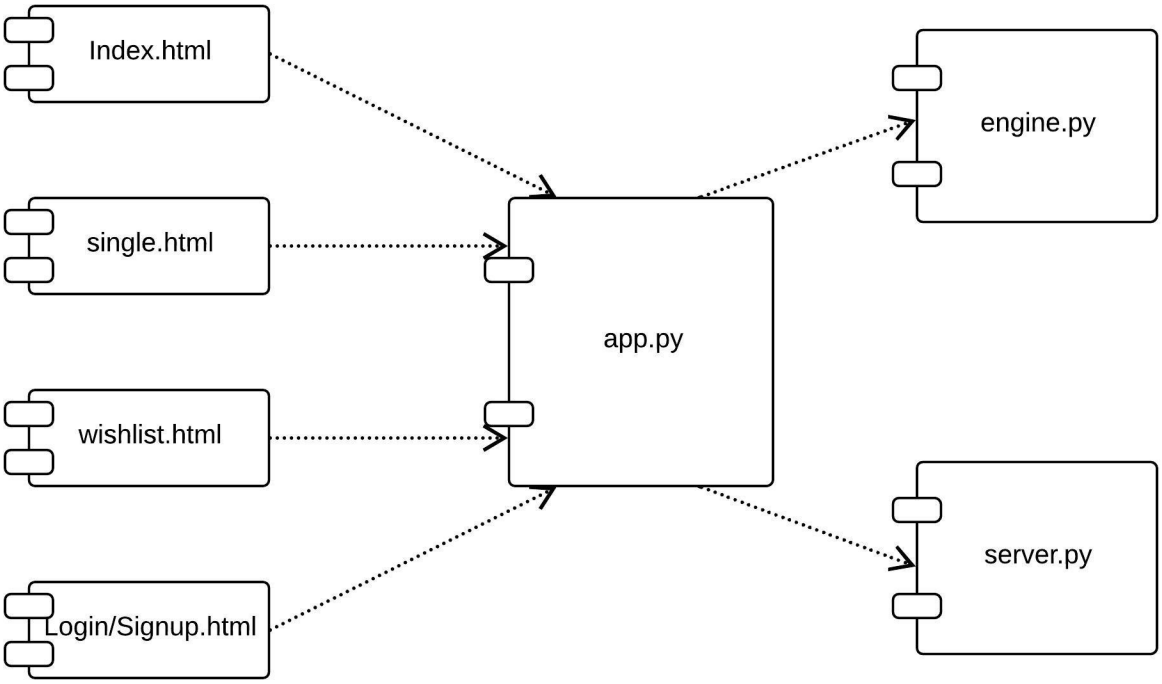

Fig 4.6 Component Diagram

COLLABORATION DIAGRAM :

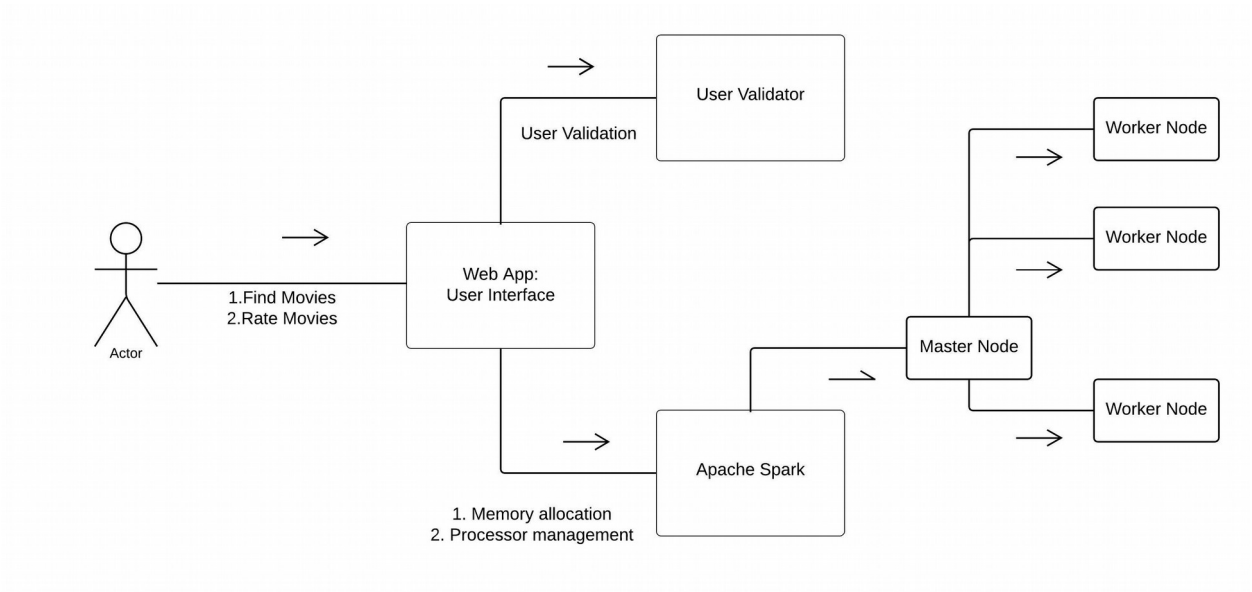

Fig. 4.8 Collaboration Diagram

DEPLOYMENT DIAGRAM :

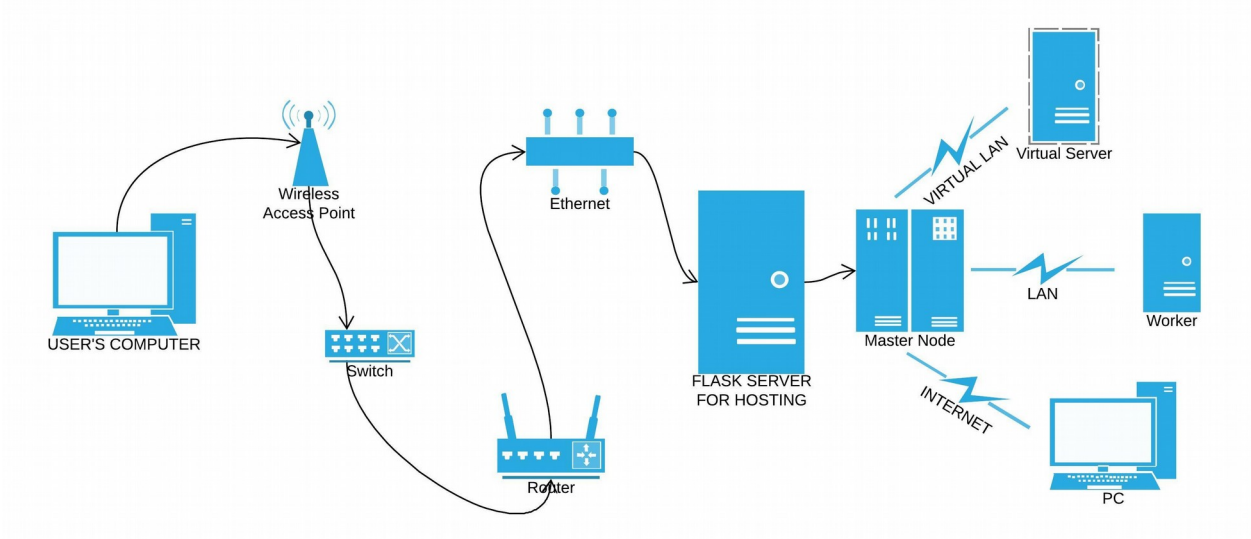

Fig. 4.7 Deployment Diagram

CLASS DIAGRAM :

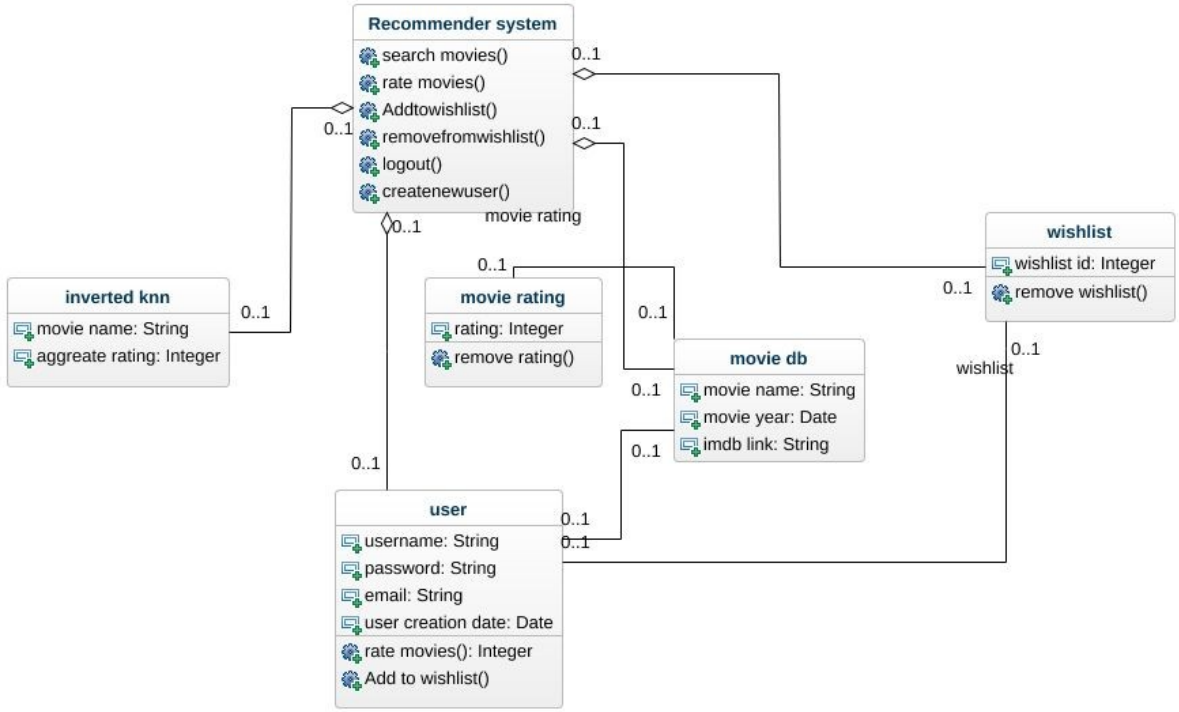

Fig. 4.9 Class Diagram

4.3 SOFTWARE DEVELOPMENT TOOLS

- **Text Editor** - Sublime Text/Visual Studio Code
- **Web Browsers** - Google Chrome + Mozilla Firefox
- **Browser developer tools** - Chrome/Firefox developer tools
- **StarUML**

4.4 LIBRARIES USED

- Matplotlib
- flask
- Numpy
- Pandas
- Sklearn
- Spark.mllib
- Flask_mysqldb