

# **MINI PROJECT – II**

**(2018-'19)**

## **“Recommendation Engine”**

**Software Requirements Specification (SRS)**



**Institute of Engineering & Technology**

### **Team Members**

Devanshu Jindal (161500193)  
Gagan Gaur (161500208)  
Harsh Agarwal (161300029)  
Ishan Kulshrestha (161500246)  
Vishnu Agarwal (161500628)

### **Supervised By**

Mr. Piyush Khandelwal  
Asstt. Professor  
Department of Computer Engineering & Applications

## Table of Contents

<b>1. INTRODUCTION</b>	1
1.1 PURPOSE	1
1.2 SCOPE	1
1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS	2
1.4 REFERENCES	2
1.5 OVERVIEW	2
<b>2. GENERAL DESCRIPTION</b>	4
2.1 PRODUCT PERSPECTIVE	4
2.2 PRODUCT FUNCTIONS	4
2.3 USER CHARACTERISTICS	4
2.4 ASSUMPTIONS AND DEPENDENCIES	4
<b>3. SPECIFIC REQUIREMENTS</b>	5
3.1 EXTERNAL INTERFACE REQUIREMENTS	5
3.1.1 User Interfaces	5
3.1.2 Communication Interfaces	5
3.2 USE CASES	6
3.3 NON-FUNCTIONAL REQUIREMENTS	7
3.3.1 Performance	7
3.3.2 Reliability	7
3.3.3 Availability	7
3.3.4 Security	7
3.3.5 Maintainability	7
3.3.6 Portability	7
3.4 INVERSE REQUIREMENTS	7
3.5 DESIGN CONSTRAINTS	7
<b>4. ANALYSIS MODELS</b>	9
4.1 SEQUENCEDIAGRAMS	9
4.2 ACTIVITY DIAGRAM	10

## **1. Introduction**

Recommending the questions that a programmer should solve given his/her current expertise is a big challenge for Online Judge Platforms but is an essential task to keep a programmer engaged on their platform.

In this project, we are given the data of programmers and questions that they have previously solved along with the time that they took to solve that particular question. Our aim is to build a model that can predict the time taken to solve a problem given the user current status. This model will help online judges to decide the next level of questions to recommend to a user.

This project will be based on the concept of machine learning and to be more specific in this project we will be testing, training and predicting data. After creating the model and getting an acceptable training and testing accuracy the model will be embedded on a website.

### **1.1 Purpose**

The purpose of this SRS is to create a detailed description of how the project/software is to be developed with its functional and non-functional requirements. This SRS contains all the necessary requirements required for our project.

### **1.2 Scope**

- This project is intended for predicting the time taken to solve a problem given the user current status.
- This model will help online judges to decide the next level of questions to recommend to a user.

### **1.3 Definitions, Acronyms, and Abbreviations**

- **Recommendation Engine:** A recommendation engine, also known as a recommender system, is software that analyzes available data to make suggestions for something that a website user might be interested in,
- **Machine Learning:** Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed.
- **Collaborative Filtering:** Collaborative Filtering is a method of making automatic predictions (filtering) about the interests of a user by collecting preferences or taste information from many users (collaborating).
- **Dataset:** A data set is a collection of data. Most commonly a data set corresponds to the contents of a single database table, or a single statistical data matrix, where every column of the table represents a particular variable.
- **Preprocessing:** Data preprocessing describes any type of processing performed on raw data to prepare it for another processing procedure. Commonly used as a preliminary data mining practice.
- **XGBoost:** XGBoost is an open-source software library which provides a gradient boosting framework for C++, Java, Python, R, and Julia. It works on Linux, Windows, and macOS.
- **Mean Squared Error:** The mean squared error or mean squared deviation of an estimator measures the average of the squares of the errors.
- **Tags:** They are used in context to define the types of algorithms.
- **Google Colab:** Colaboratory is a free Jupyter notebook environment that requires no setup and runs entirely in the cloud. With Colaboratory you can write and execute code, save and share your analyses, and access powerful computing resources.

### **1.4 References**

All the diagrams in this SRS are made by the author from scratch. References are provided in every section wherever needed.

### **1.5 Overview**

A recommendation engine, also known as a recommender system, is software that analyzes available data to make suggestions for something that a website user might be interested in, such as a book, a video or a job, among other possibilities.

A recommendation engine, also known as a recommender system, is software that analyzes available data to make suggestions for something that a website user might be interested in, such as a book, a video or a job, among other possibilities.

In our case our model will predict the time taken to solve a problem given the user current status. This model will help online judges to decide the next level of questions to recommend to a user.

### **2. General Description**

The project is solely based on two things:

- Database (queries) – they are the intents that will be saved in the API and according to the queries fired, the API will revert the output.
- Machine Learning Models – We have trained different models using the data in different ways which has provided us with diversified insights which are very helpful for us to suggest new problems to the users efficiently.

#### **2.1 Product Perspective**

There are various projects available in the market which works on the same concept and their utility is widely appreciated in the market. For ex: Recommender systems are utilized in a variety of areas including movies, music, news, books, research articles, search queries, social tags, and products in general. There are also recommender systems for experts, collaborators, jokes, restaurants, garments, financial services, life insurance, romantic partners (online dating), and Twitter pages.

## **2.2 Product Functions**

The project will use the segregated data and after merging the data, it will be preprocessed, and then the required models will be trained on the same.

## **2.3 User Characteristics**

The end user of this project will be the online platforms to decide the next level of questions to recommend to a user by integrating it into their online platforms.

## **2.4 Assumptions and Dependencies**

It is to be assumed that the new data gathered after using our recommendation engine will be processed again to increase the efficiency of the engine.

There are 3 training data files.

- train\_submissions.csv - This contains 1,55,295 submissions which are selected randomly from 2,21,850 submissions. Contains 3 columns ('user\_id', 'problem\_id', 'attempts\_range'). The variable 'attempts\_range' denoted the range no. in which attempts the user made to get the solution accepted lies.
- user\_data.csv - This is the file containing data of users. It contains the following features :-  
user\_id - unique ID assigned to each user  
submission\_count - total number of user submissions  
problem\_solved - total number of accepted user submissions  
contribution - user contribution to the judge  
country - location of user  
follower\_count - amount of users who have this user in followers  
last\_online\_time\_seconds - time when user was last seen online  
max\_rating - maximum rating of user  
rating - rating of user  
rank - can be one of 'beginner', 'intermediate', 'advanced', 'expert'  
registration\_time\_seconds - time when user was registered
- problem\_data.csv - This is the file containing data of the problems. It contains the following features :-  
problem\_id - unique ID assigned to each problem  
level\_id - the difficulty level of the problem between 'A' to 'N'  
points - amount of points for the problem  
tags - problem tag(s) like greedy, graphs, DFS etc.

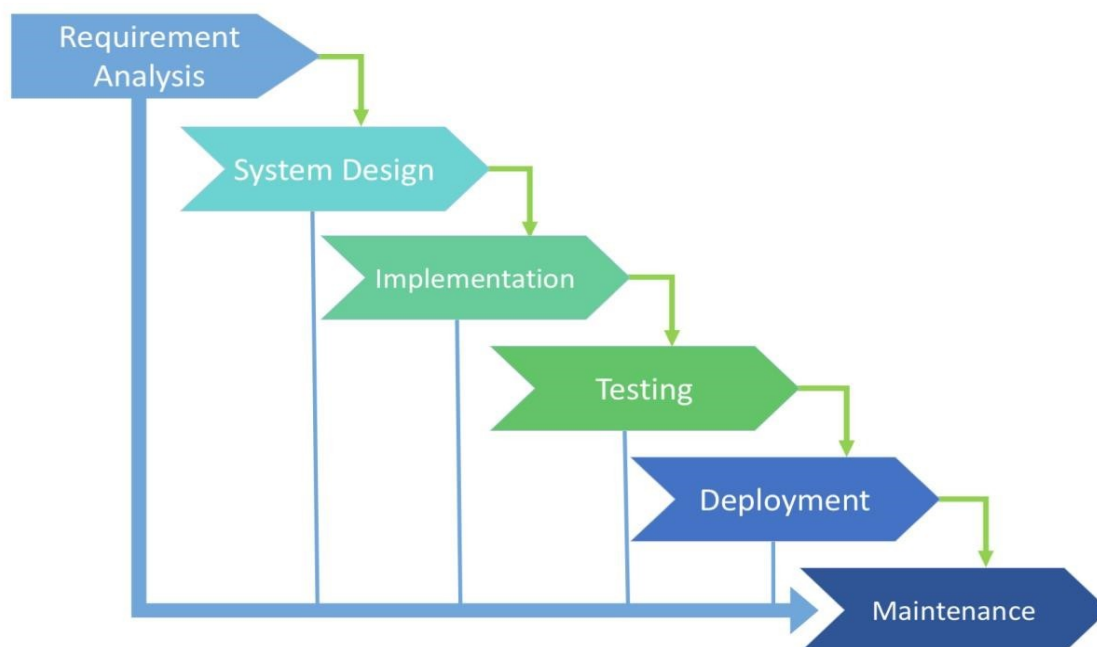
## Recommendation Engine

- test\_submissions.csv - This contains the remaining 66,555 submissions from total 2,21,850 submissions. Contains 1 column (ID). The 'attempts\_range' column is to be predicted.

### **3. Specific Requirements**

This software development life cycle model used in this project will be Iterative Enhancement Model. This model has the same phases as the waterfall mode, but with fewer restrictions.

Generally the phase occurs in the same order as in the waterfall model, but they may be conducted in several cycles.



### **3.1 External Interface Requirements**

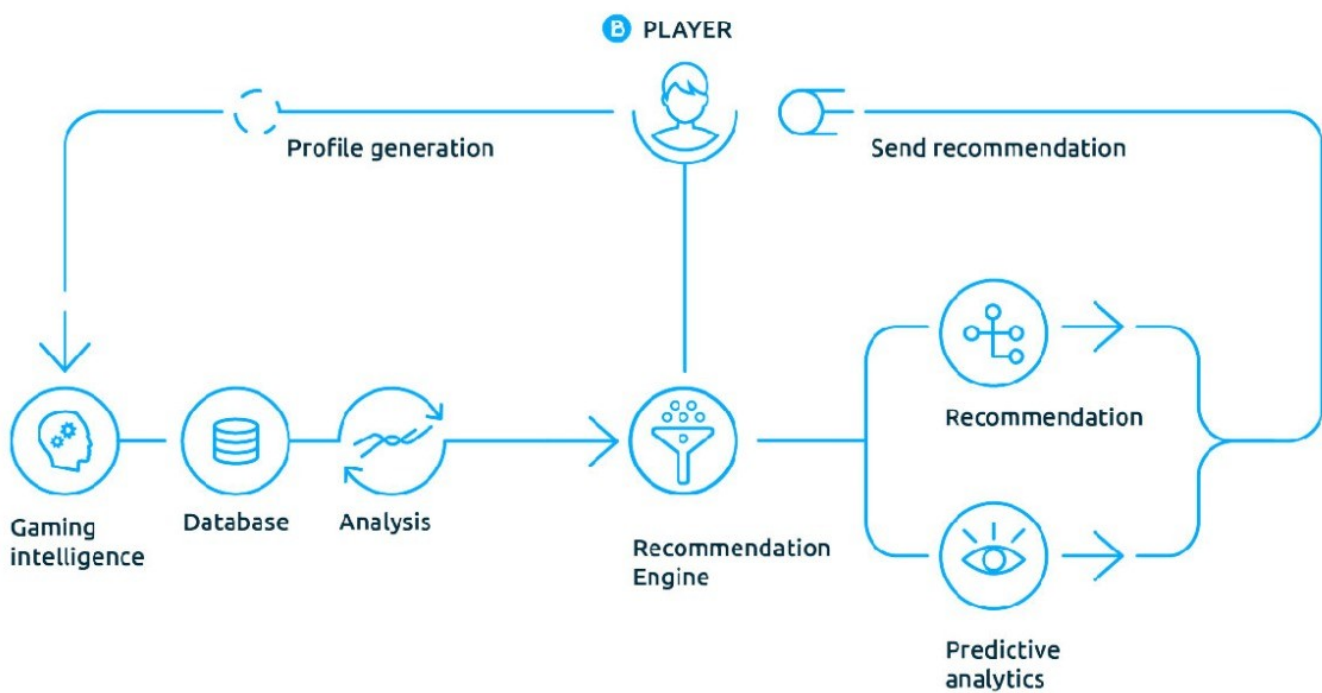
#### **3.1.1 User Interfaces**

The user will be using our recommendation Engine which will be integrated in their website.

#### **3.1.2 Communications Interfaces**

You can access our recommendation engine through internet

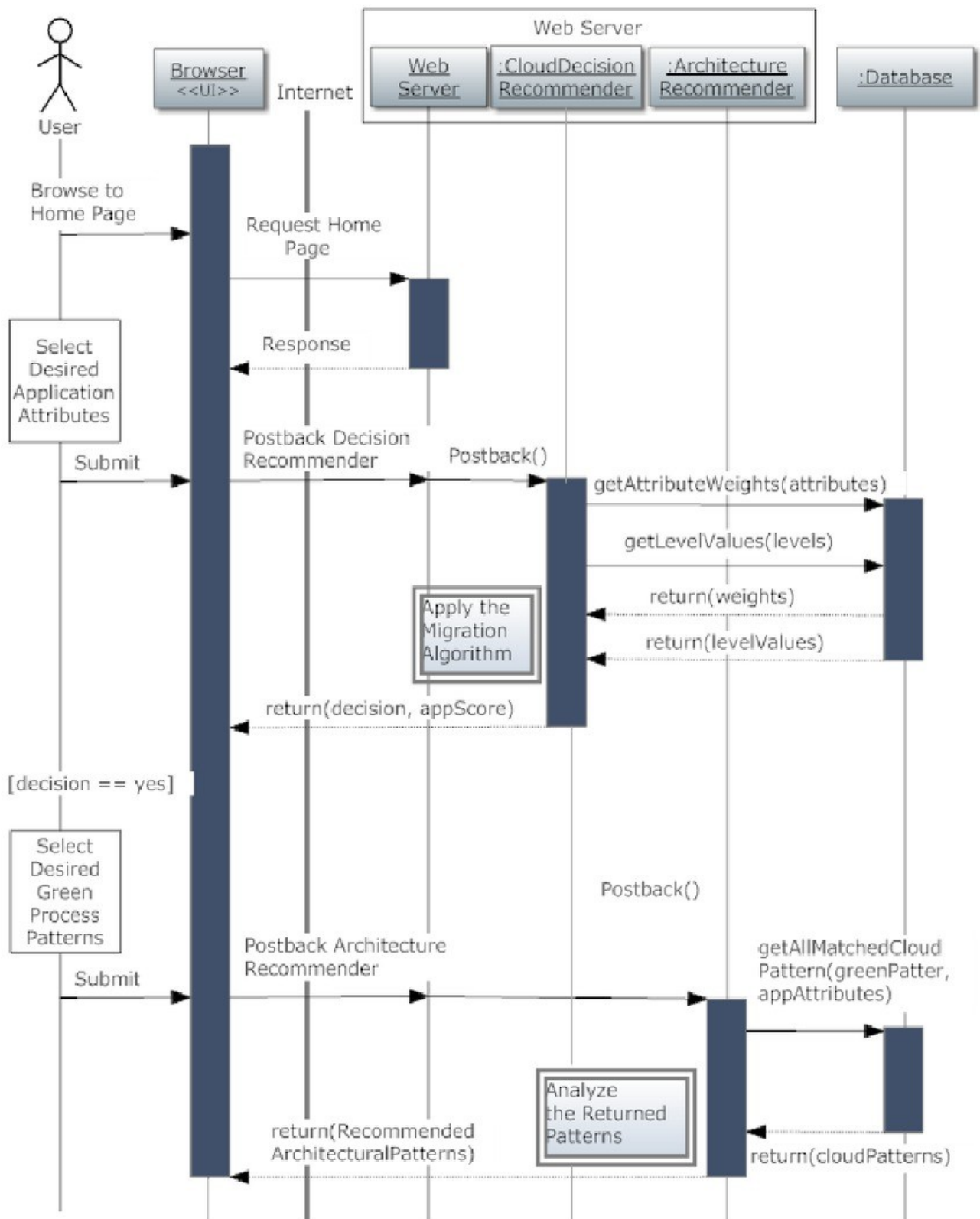
### **3.2 Use Cases**





## 4. Analysis Models

### 4.1 Sequence Diagrams



## 4.2 Activity Diagram

