SYMPHONY: As You Like It!

Music Recommendation System

Submitted by:
(101503004) ABHIMANYU SHARMA
(101503023) AMANDEEP SINGH
(101503086) HARNOOR SINGH BEDI
BE Third Year, CSE
Project Team No. CPG-142

Under the mentorship of:
Ms. Vineeta Bassi (Assistant Professor)

Mr. Sanjeev Rao (Lecturer)

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Computer Science and Engineering Department
Thapar Institute of Engineering and Technology, Patiala

May 8, 2018

1. Working Principles

The explosive growth in the amount of available digital information and the number of visitors to the Internet have created a potential challenge of information overload which hinders timely access to items of interest on the Internet. Information retrieval systems, such as Google have partially solved this problem, but prioritization and personalization of information were absent. This has increased the demand for recommender systems more than ever before. Recommender systems are information filtering systems that deal with the problem of information overload by filtering vital information fragment out of large amount of dynamically generated information according to user's preferences, interest, or observed behavior about item. Recommender system could predict whether a particular user would prefer an item or not based on the user's profile.

Phases of recommendation process

1. **Information collection phase**

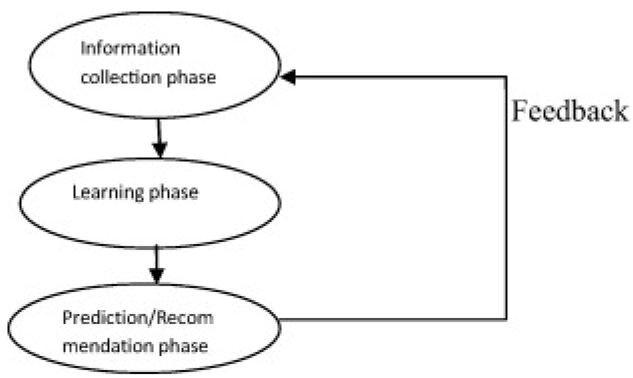
This collects relevant information of users to generate a user profile or model for the prediction tasks including user's attribute, behaviors or content of the resources the user accesses. A recommendation agent cannot function accurately until the user profile/model has been well constructed. The system needs to know as much as possible from the user in order to provide reasonable recommendation right from the onset. Recommender systems rely on different types of input such as the most convenient high quality explicit feedback, which includes explicit input by users regarding their interest in item or implicit feedback by inferring user preferences indirectly through observing user behavior. Hybrid feedback can also be obtained through the combination of both explicit and implicit feedback.

2. **Learning phase**

It applies a learning algorithm to filter and exploit the user's features from the feedback gathered in information collection phase.

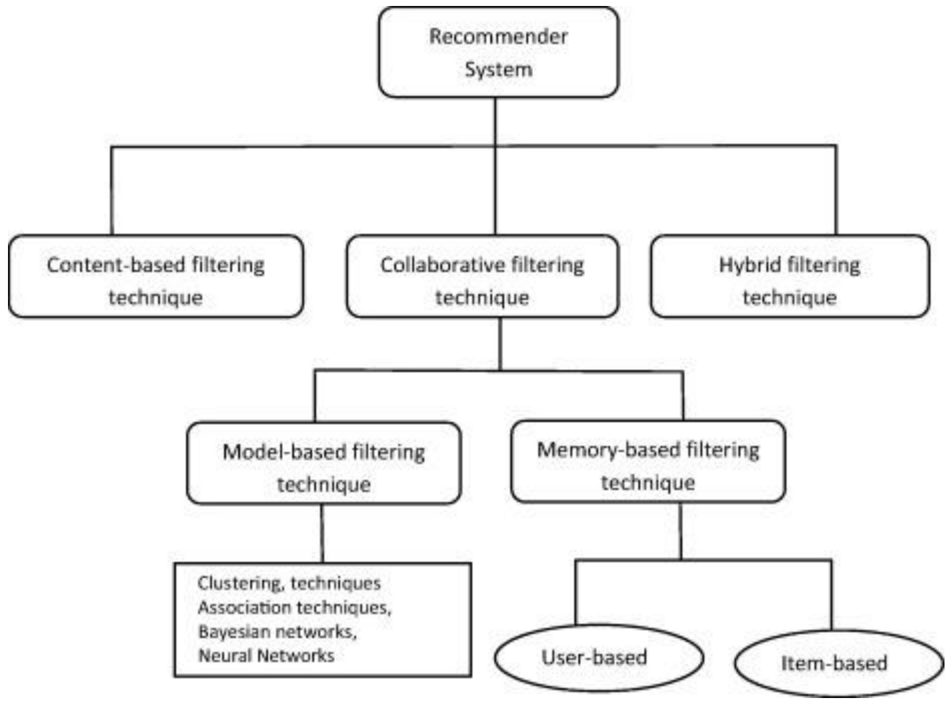
3. **Prediction/recommendation phase**

It recommends or predicts what kind of items the user may prefer. This can be made either directly based on the dataset collected in information collection phase which could be memory based or model based or through the system's observed activities of the user.



Recommendation techniques

The use of efficient and accurate recommendation techniques is very important for a system that will provide good and useful recommendation to its individual users. This explains the importance of understanding the features and potentials of different recommendation techniques.



1. *Content-based filtering*

Content-based technique is a domain-dependent algorithm and it emphasizes more on the analysis of the attributes of items in order to generate predictions. When documents such as web pages, publications and news are to be recommended, content-based filtering technique is the most successful. In content-based filtering technique, recommendation is made based on the user profiles using features extracted from the content of the items the user has evaluated in the past. items that are mostly related to the positively rated items are recommended to the user. CBF uses different types of models to find similarity between documents in order to generate meaningful recommendations. It could use Vector Space Model such as Term Frequency Inverse Document Frequency (TF/IDF) or Probabilistic models such as Naïve Bayes Classifier, Decision Trees or Neural Networks to model the relationship between different documents within a corpus. Content-based filtering technique does not need the profile of other users since they do not influence recommendation. Also, if the user profile changes, CBF technique still has the potential to adjust its recommendations within a very short period of time. The major disadvantage of this technique is the need to have an in-depth knowledge and description of the features of the items in the profile.

2. *Collaborative filtering*

Collaborative filtering is a domain-independent prediction technique for content that cannot easily and adequately be described by metadata such as movies and music. Collaborative filtering technique works by building a database (user-item matrix) of preferences for items by users. It then matches users with relevant interest and preferences by calculating similarities between their profiles to make recommendations. Such users build a group called neighborhood. A user gets recommendations to those items that he has not rated before but that were already positively rated by users in his neighborhood. Recommendations that are produced by CF can be of either prediction or recommendation. Prediction is a numerical value, R_{ij}, expressing the predicted score of item j for the user i, while Recommendation is a list of top N items that the user will like the most. The technique of collaborative filtering can be divided into two categories: memory-based and model-based.

a. Memory Based techniques

The items that were already rated by the user before play a relevant role in searching for a neighbor that shares appreciation with him. Once a neighbor of a user is found, different algorithms can be used to combine the preferences of neighbors to generate recommendations. Due to the effectiveness of these techniques, they have achieved widespread success in real life applications. Memory-based CF can be achieved in two ways through user-based and item-based techniques. User based collaborative filtering technique calculates similarity between users by comparing their ratings on the same item, and it then computes the predicted rating for an item by the active user as a weighted average of the ratings of the item by users similar to the active user where weights are the similarities of these users with the target item. Item-based filtering techniques compute predictions using the similarity between items and not the similarity between users. It builds a model of item similarities by retrieving all items rated by an active user from the user-item matrix, it determines how similar the retrieved items are to the target item, then it selects the *k* most similar items and their corresponding similarities are also determined. Prediction is made by taking a weighted average of the active users rating on the similar items *k*.

b. Model-based techniques

This technique employs the previous ratings to learn a model in order to improve the performance of Collaborative filtering Technique. The model building process can be done using machine learning or data mining techniques. These techniques can quickly recommend a set of items for the fact that they use pre-computed model and they have proved to produce recommendation results that are similar to neighborhood-based recommender techniques.

3. *Hybrid filtering*

Hybrid filtering technique combines different recommendation techniques in order to gain better system optimization to avoid some limitations and problems of pure recommendation systems. The idea behind hybrid techniques is that a combination of algorithms will provide more accurate and effective recommendations than a single algorithm as the disadvantages of one algorithm can be overcome by another algorithm. Using multiple recommendation techniques can suppress the weaknesses of an individual technique in a combined model. The combination of approaches can be done in any of the following ways: separate implementation of algorithms and combining the result, utilizing some content-based filtering in collaborative approach, utilizing some collaborative filtering in content-based approach, creating a unified recommendation system that brings together both approaches. Weighted hybridization combines the results of different recommenders to generate a recommendation list or prediction by integrating the scores from each of the techniques in use by a linear formula.

Evaluation metrics for recommendation algorithms

The quality of a recommendation algorithm can be evaluated using different types of measurements.

Statistical accuracy metrics evaluate accuracy of a filtering technique by comparing the predicted ratings directly with the actual user rating. Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and Correlation are usually used as statistical accuracy metrics. MAE is the most popular and commonly used; it is a measure of deviation of recommendation from user's specific value. Root Mean Square Error (RMSE) puts more emphasis on larger absolute error and the lower the RMSE is, the better the recommendation accuracy.

$$\text{MAE} = \frac{1}{N}\left(\sum_{u,i} |p(u,i) - r(u,i)|\right)$$

where p(u, i) is the predicted rating for user *u* on item *i*, r(u,i) is the actual rating and *N* is the total number of ratings on the item set.

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{u,i}(p(u,i) - r(u,i))^2}$$

Decision support accuracy metrics that are popularly used are Precision Recall Curve (PRC), Precision, Recall and *F*-measure. These metrics help users in selecting items that are of very high quality out of the available set of items. The metrics view prediction procedure as a binary operation which distinguishes good items from those items that are not good. Precision is the fraction of recommended items that is actually relevant to the user, while recall can be defined as the fraction of relevant items that are also part of the set of recommended items.

$$Precision = \frac{Correctly\ Recommended\ items}{Total\ Recommended\ items}$$

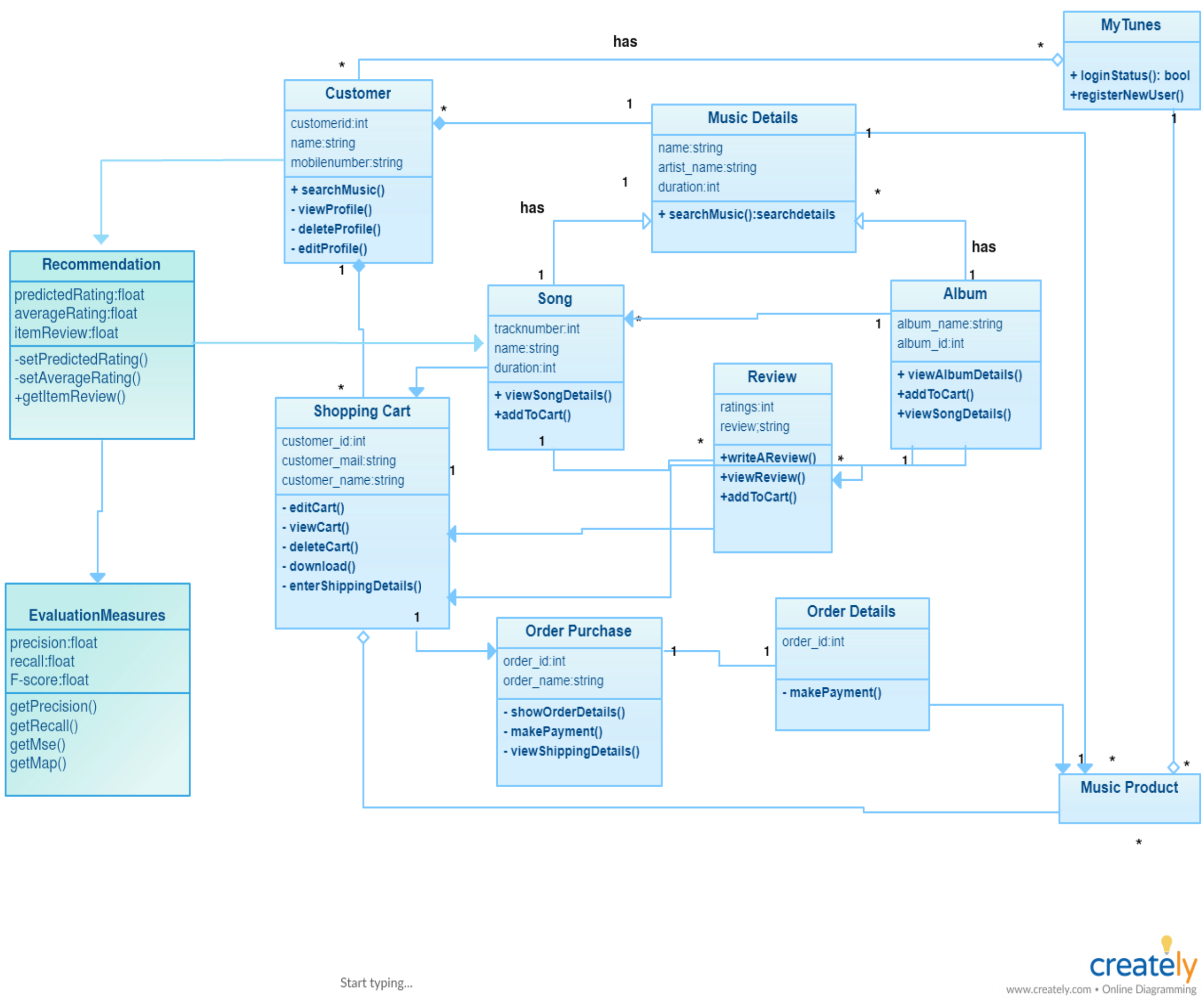
$$Recall = \frac{Correctly\ Recommended\ items}{Total\ useful\ recommended\ items}$$

F-measure defined below helps to simplify precision and recall into a single metric. The resulting value makes comparison between algorithms and across data sets very simple and straightforward.

$$F - Measure = \frac{2 * Precision * Recall}{(Precision + Recall)}$$

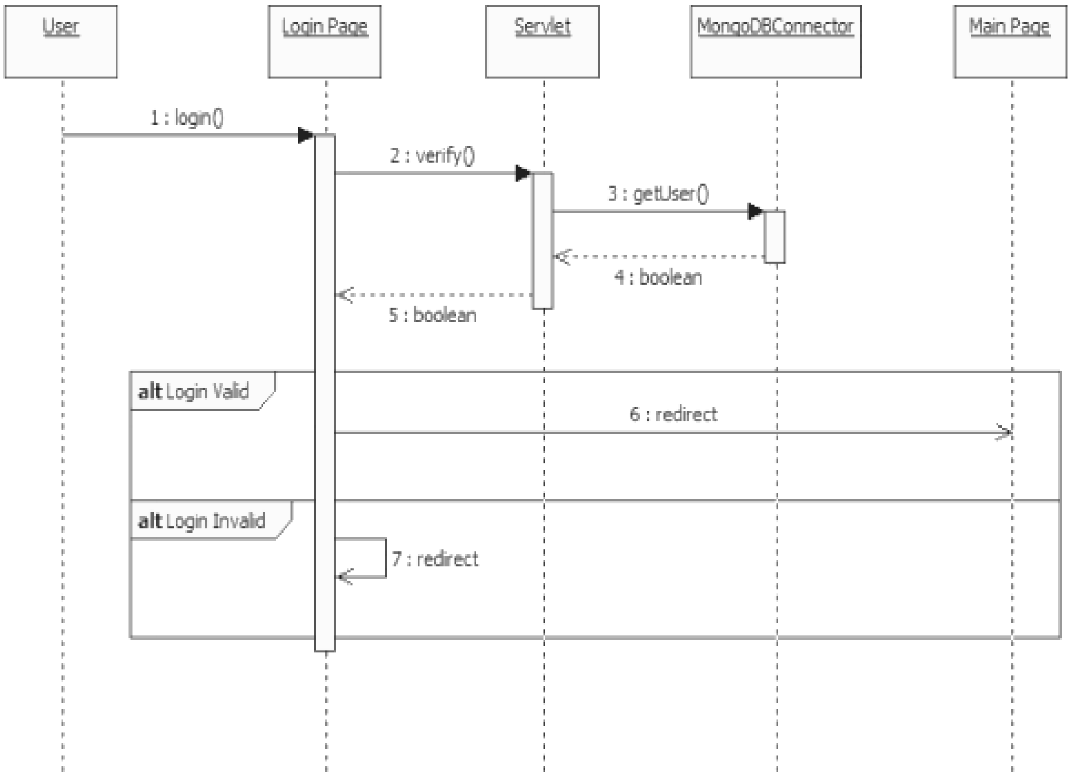
2. SRS with Finalized Analysis Model:

1. Class Diagram:

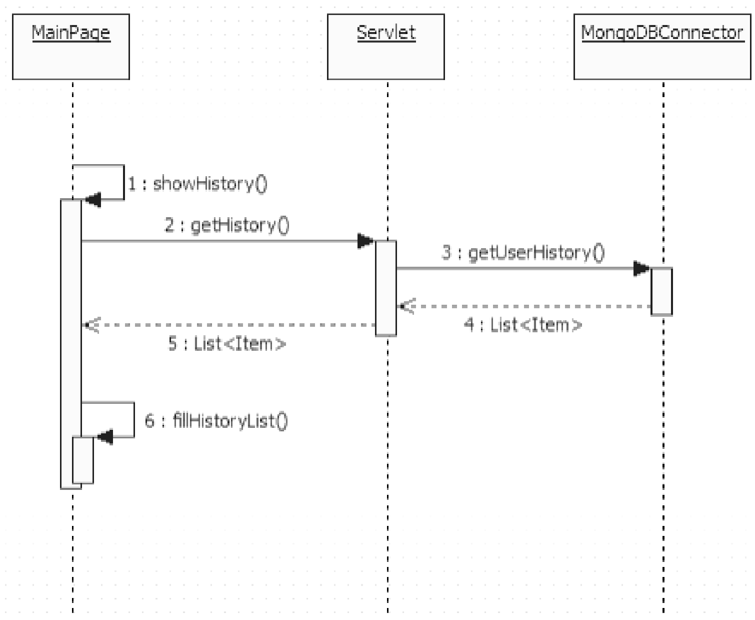


2. <u>Sequence diagram</u>:

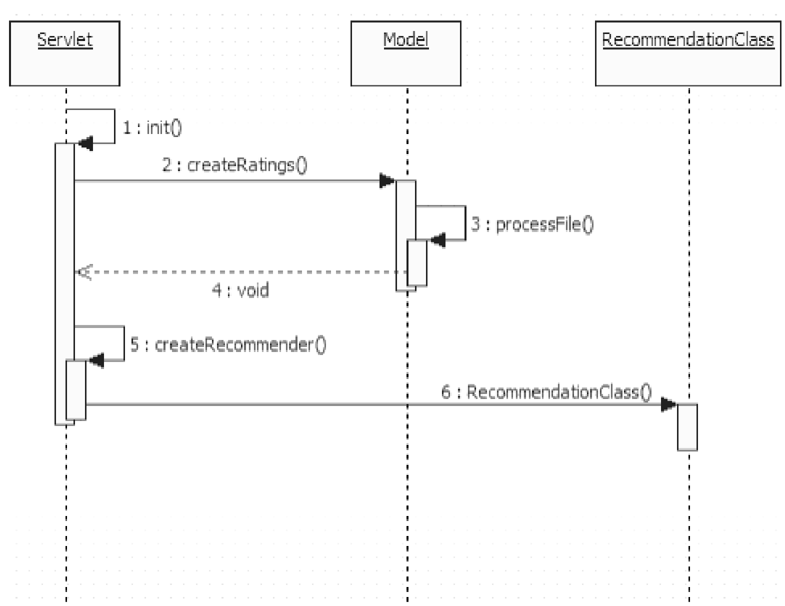
2.1 Login Sequence Diagram:



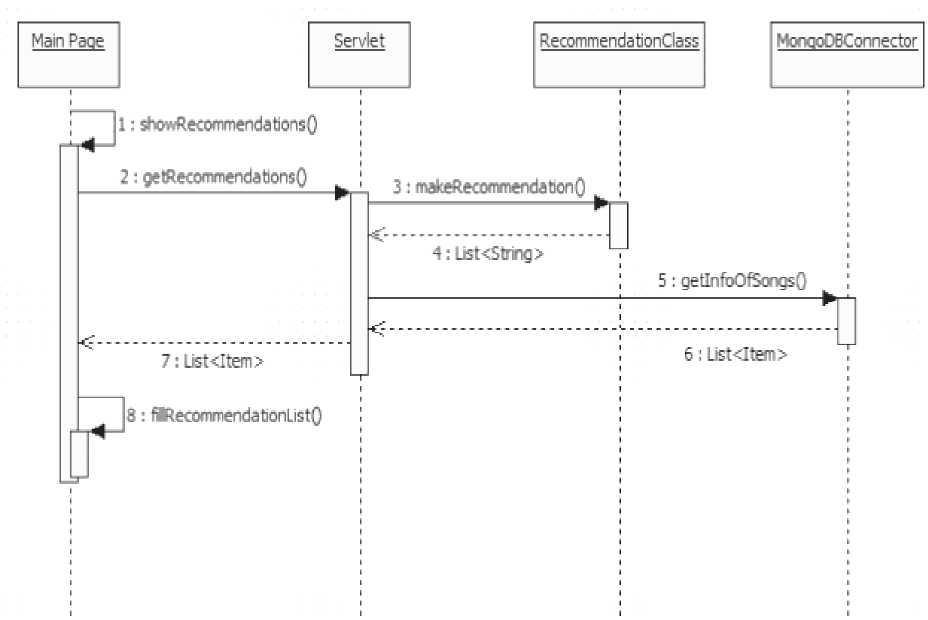
2.2 View History Sequence Diagram:



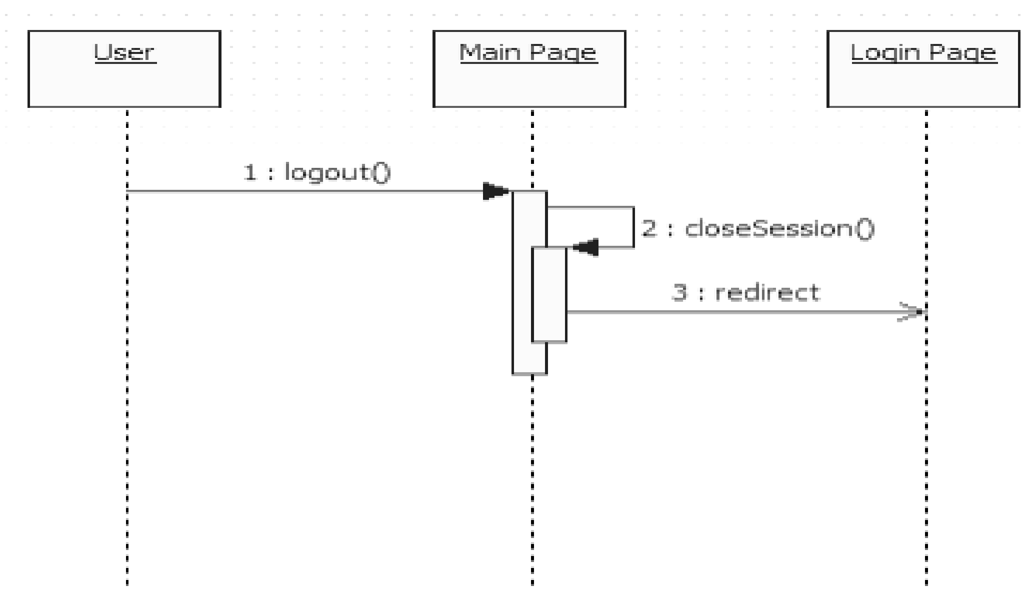
2.3 Generate Recommendations Sequence Diagram:



2.4 View Recommendations Sequence Diagram:

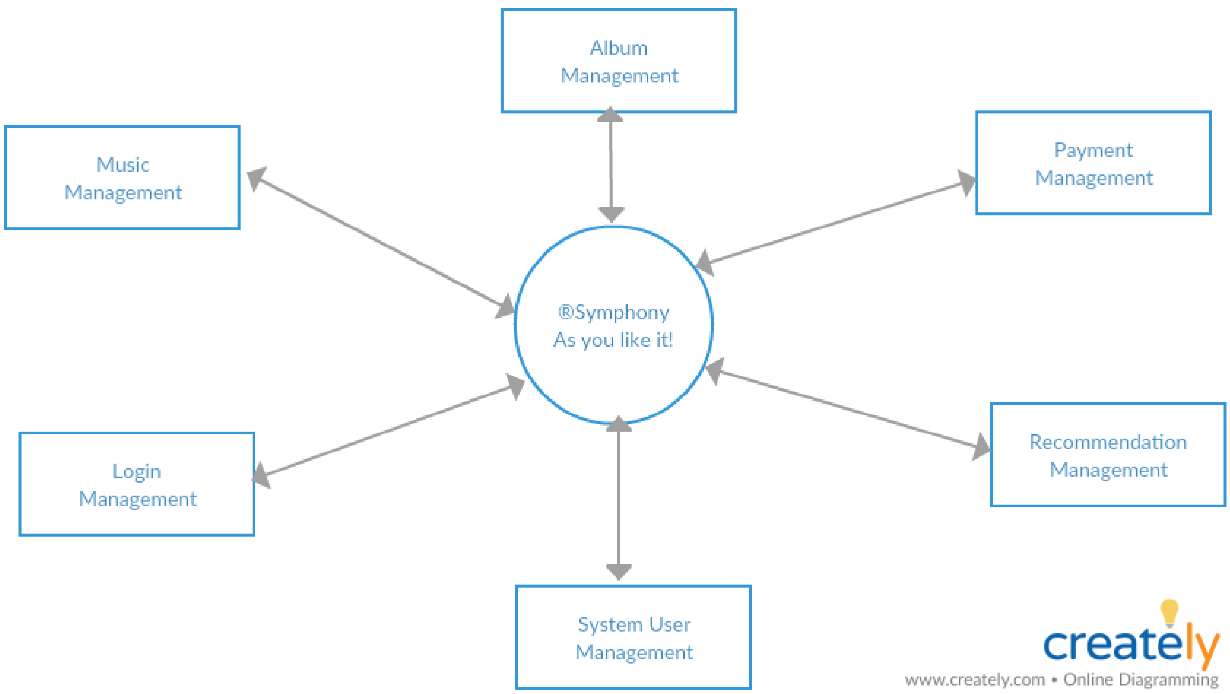


2.5 Logout Sequence Diagram:

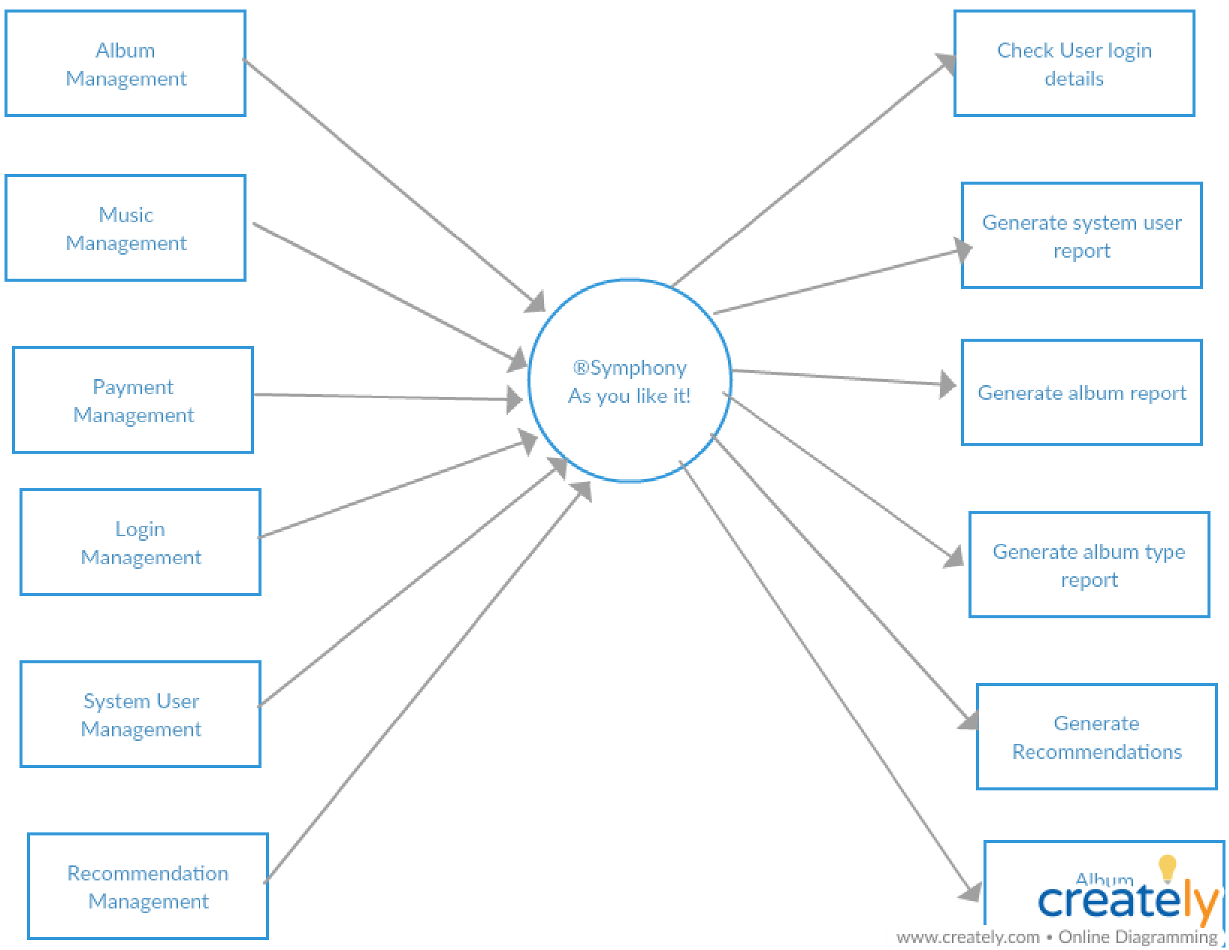


3. <u>Data Flow Diagrams:</u>

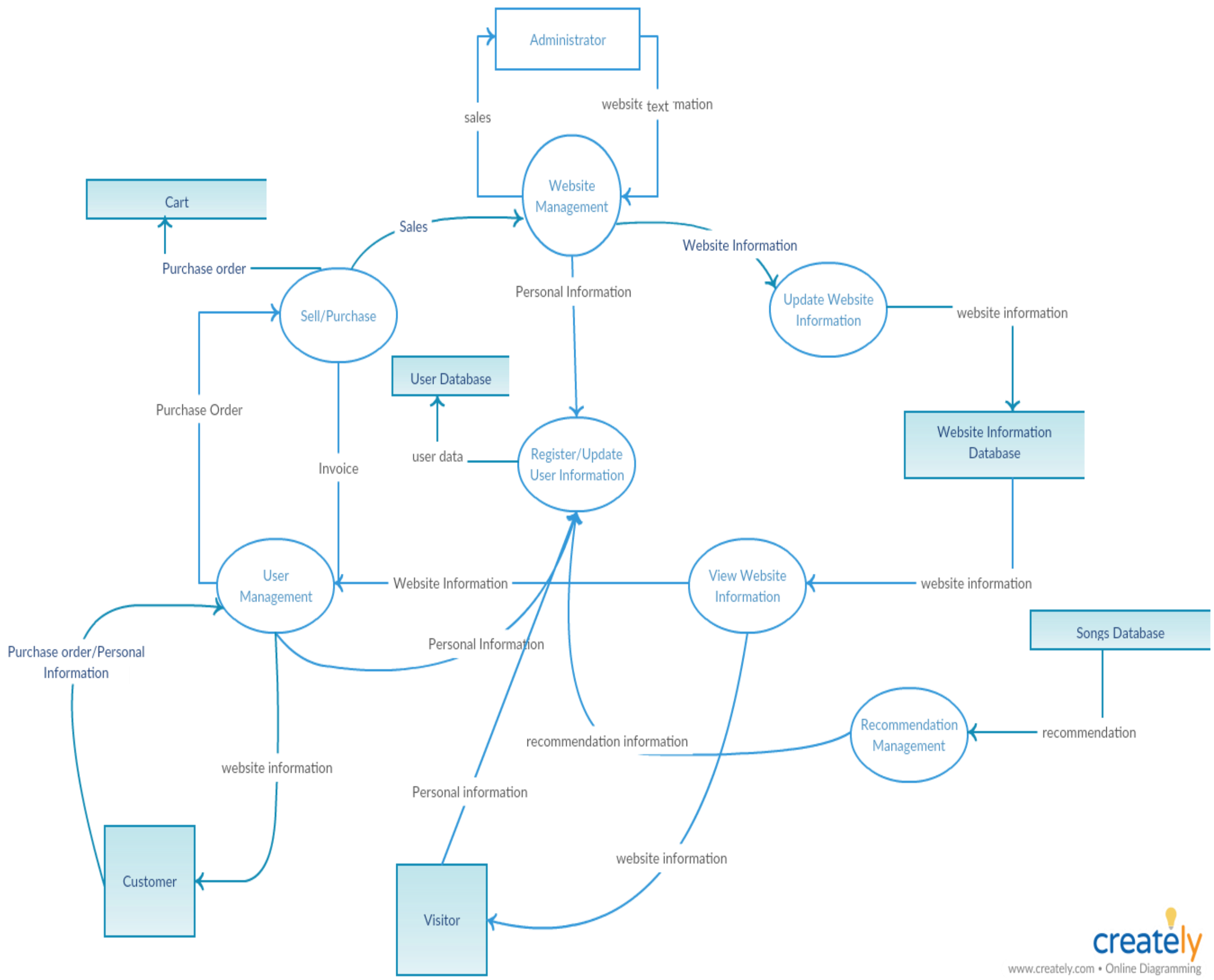
3.1 Zero level DFD:



3.2 First level DFD:



3.3 Second level DFD:



3.Cost analysis

The development tools that we will use for this project are open source. We will use Python, PHP, JavaScript, MYSQL, MongoDB and Apache Spark for our project. Since all these are open source, we will not incur any cost for the development of this project.

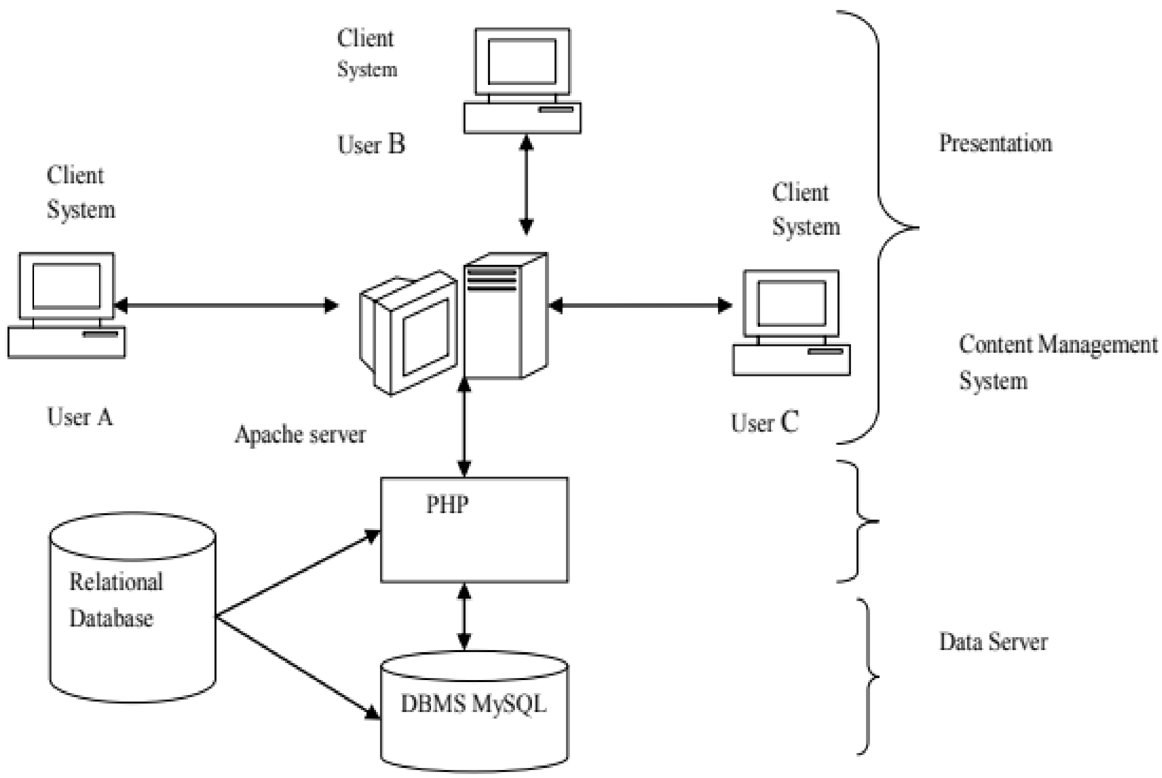
In addition to this if we want to make our project live, there will be cost of servers and domain names. Domain names cost around 1000-2000 rupees per year. Servers will cost around 700-800 rupees per year.

4.Constraints and assumptions

- The Internet connection is a constraint for the application. Since the application fetches data from the server over the Internet, it is crucial that there is an Internet connection for the application to function.
- Execution time for the algorithm should take no longer than one second.
- Music Recommender will be a sub component of Music website.
- Users shall be required to log in to the website to get recommendations.
- It also needs to give unique recommendations for each user.
- Another assumption is that the user has a web browser and a capable hardware in order to launch the website.
- It is also assumed that system is able to provide necessary requirements for the recommender system to run.

5. **Design Model:**

The online music portal system will be designed and implemented using MySQL as the database, Apache will be web server to provide basic functionality of the web services. PHP will be used as scripting language to program the server side that manipulates the knowledge in the database. The web architecture of our music portal is as follows:



Database design is as follows:

Table 1: Users

Field	**Type**
userid	int (25)
first_name	varchar (25)
last_name	varchar (25)
email_address	varchar (25)
username	varchar (25)
mobile_no	varchar (25)
sex	varchar (25)
marital_status	varchar (25)
nationality	varchar (25)
password	varchar (25)

Table 2: Track table

Field	**Type**
id	bigint (25)
performer_id	int (11)
album_id	int (11)
track_no	smallint (6)
name	varchar (20)
duration	varchar (6)
last_played	varchar (20)
time_played	int (11)
year	varchar (4)

Table 3: Performer Table

Field	**Type**
pid	int (11)
pname	varchar (20)
year	varchar (4)

Table 4: Favorites table

Field	**Type**
id	int (11)
track_id	int (11)
performer_id	int (11)
album_id	int (11)
name	varchar (50)
duration	varchar (6)
last_played	varchar (20)
time_played	int (11)
year	varchar (4)
user_id	int (11)

Table 5: Album table

Field	**Type**
a_id	int (11)
performer_id	int (11)
name	varchar (50)

Table 6: Cart table

Field	**Type**
ordernumber	int (11)
order_id	int (11)
userid	int (25)
payment	int (8)