1. Data Design

The design of the application can be broadly categorized into 3 sections – the frontend, backend and the recommender system. The frontend uses popular web development languages like HTML, CSS, JavaScript and PHP. Bootstrap has also been used to design the front end and make it responsive to devices of different dimensions. The backend runs on MySQL provided by XAMPP – this is required as the data is relational. This allows for convenient database integration that allows maintenance of a large database of information that is required for the website. The recommender itself is coded in Python and is based on a Collaborative filtering algorithm. It uses a large dataset (which is stored and accessed from a file, since the database cannot support a huge amount of data) generated by the members of this project team for the purpose of implementing this project at a professional scale.

The system required that recommendations, regarding movies the customer may want to watch, be made on the basis of how the Customer rated a particular movie. From his/her ratings, a similarity score could be calculated using the ratings of other customers. The recommendation part of this project is coded entirely in python as its rich set of libraries allow smooth flow of large volumes of data and also provide extremely useful functions for performing operations on the data. For effective and accurate recommendations, the dataset must be largely sufficient and hence we make use of file storage methods as well as relational data stored in the database.

The database as well as the file containing customer activity is accessible through-out the span of the website. Though these are used by specific functionalities in the website, they are global data structures. Temporary data storage is required while registering a new customer and this is carried out using php objects/variables.

1.1. Schema Diagram

The transformation from the entity-relationship model to the relational model is very straightforward. A feasible set of relational schemas is as follows.

Admin:

Admin_ID	AdminName	AdminEmail	AdminPassword

Customer:

Customer_ID	Username	Email	Password

Query :

Query_ID	Question	Answer	Ad_ID	C_ID

Movies:

movieid	title	genre

Ratings:

userid	movieid	rating	timestamp

2. Architectural and Component Level Design

2.1 System Architecture

The constructed system will have three-tier architecture. Three-tier architecture allows any one of the three tiers to be upgraded or replaced independently. The user interface is implemented on a desktop PC and uses a standard graphical user interface with different modules running on the application server. The relational database management system on the database server contains the computer data storage logic. The middle tiers are usually multitiered.

The three tiers(Figure 3.1) in a three-tier architecture are:

1. Presentation Tier: Occupies the top level and displays information related to services available on a website. This tier communicates with other tiers by sending results to the browser and other tiers in the network.
2. Application Tier: Also called the middle tier, logic tier, business logic or logic tier, this tier is pulled from the presentation tier. It controls application functionality by performing detailed processing.

3. Data Tier: Houses database servers where information is stored and retrieved. Data in this tier is kept independent of application servers or business logic.

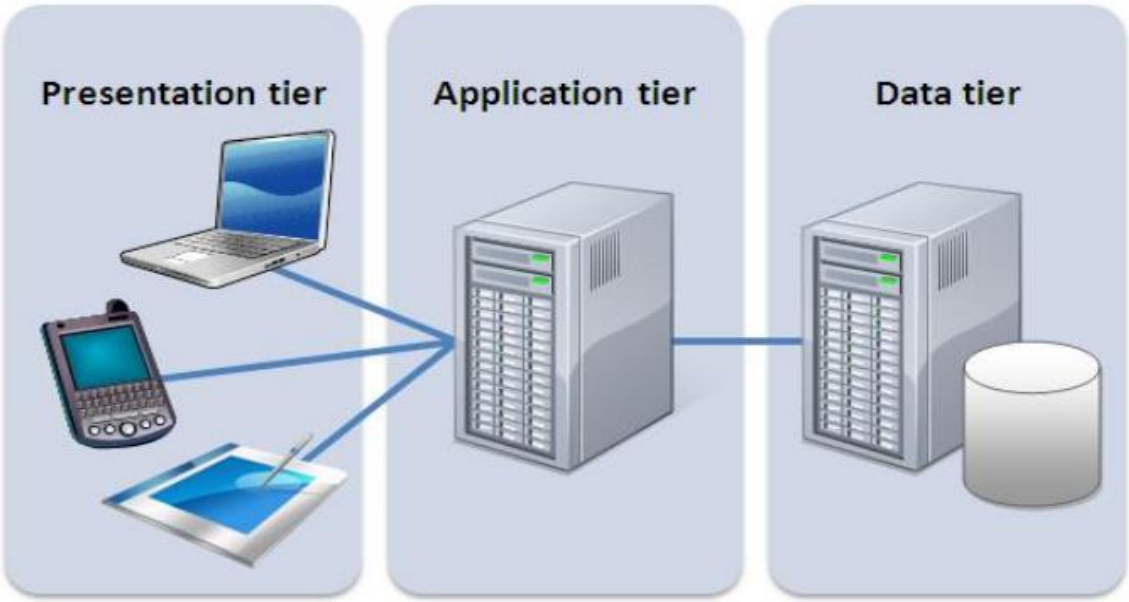

Figure 1: Three tier architecture

3. Diagrams

3.1. Scenario based elements

A) Activity Diagram:

An activity diagram represents the actions and decisions that occur as some function is performed. Activity diagram adds additional detail not directly mentioned by the use-case. It supplements the use-case by providing a graphical representation of the flow of interaction within a specific scenario. Figure 3.2 shows all the activities performed. Activity diagram forms a part of the interface design in the design model.

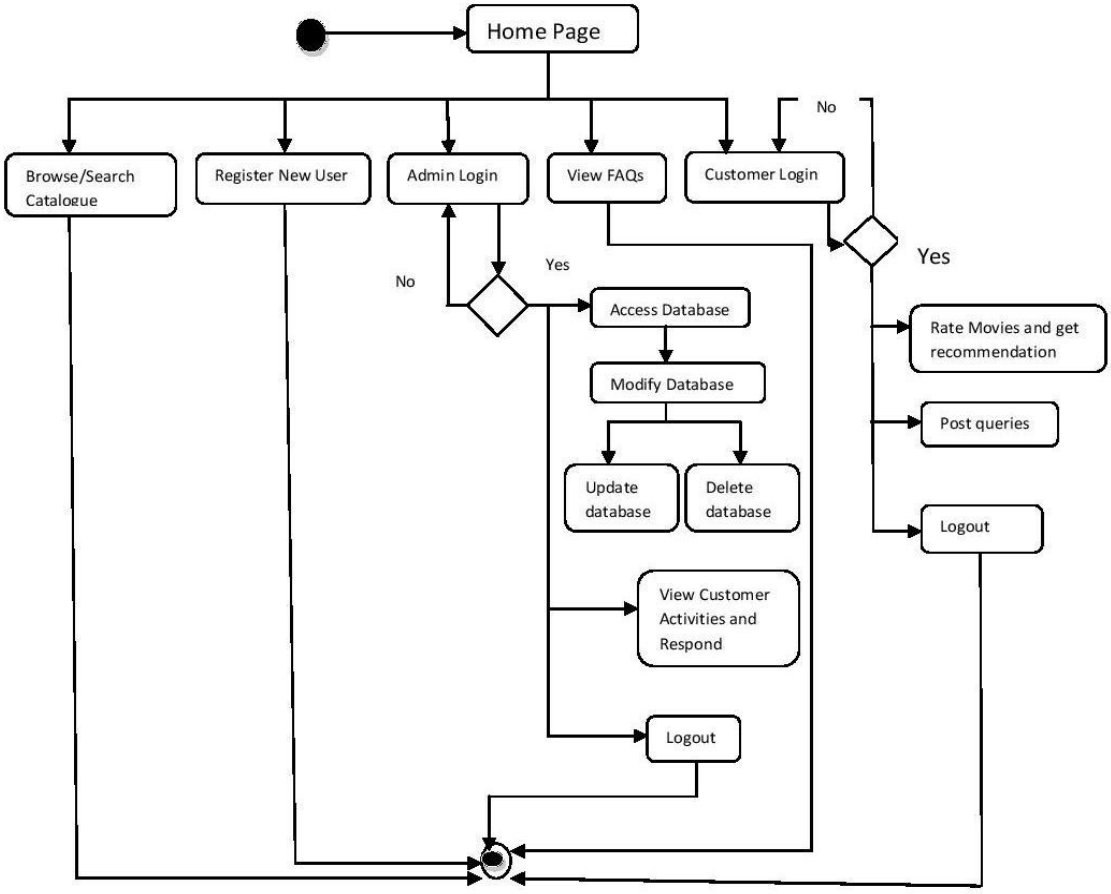

Figure 2: Activity diagram

B) Swimlane diagram:

The UML swimlane diagram is a useful variation of the activity diagram and it allows the modeller to represent the flow of activities by the use-case and at the same time indicate which actor or analysis class has responsibility for the action described by an activity rectangle. Responsibilities are required as parallel segments that divide the diagram vertically, like the lanes in a swimming pool. The activity diagram is rearranged so that the activities associated with a particular analysis class fall inside the swimlane of that class. The swimlane diagram also forms a part of the interface design in the design model.

The actors here are: customer, administrator and viewer and system is the analysis class. For instance, the customer can access the functionalities such as querying, rating the movies etc.

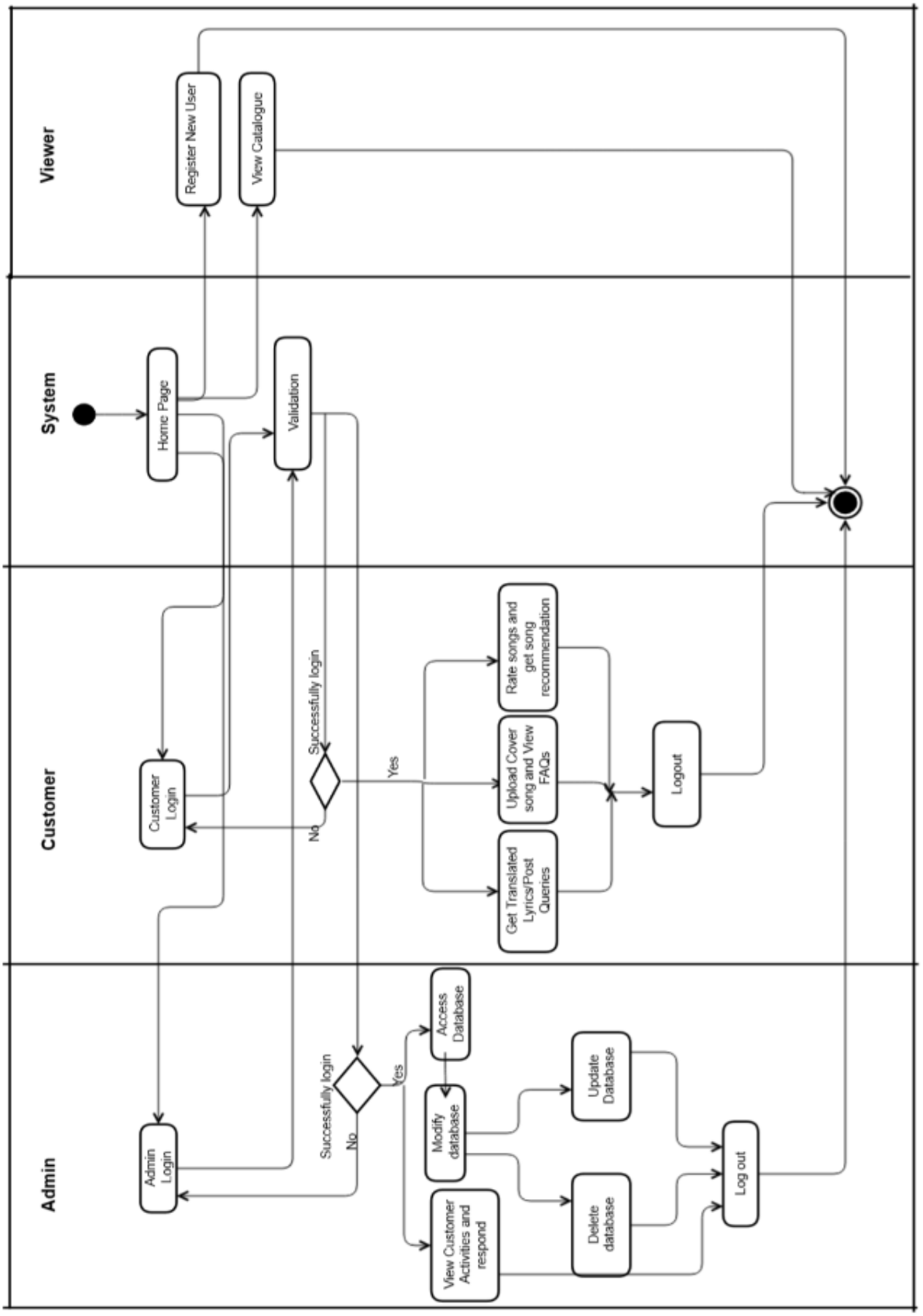

Figure 3: Swimlane diagram

C) Use Case Diagrams:

It graphically describes a specific usage scenario in straightforward language from the point of view of a defined actor. Here in our case, customer, viewer and admin are the actors. The login use case is used by activities such as rate movies, get recommendation etc. And in the admin use-case diagram, the modify use case is extended by “add data” and “delete data” use cases.

The “extends association” is used when we have one use case which is similar to another use case, but, does a bit more or is more specialized. And, the “uses association” occurs when we are describing the use cases and notice that some of them have subflows in common.

Use case diagrams form a part of the Interface Design of the Design Model.

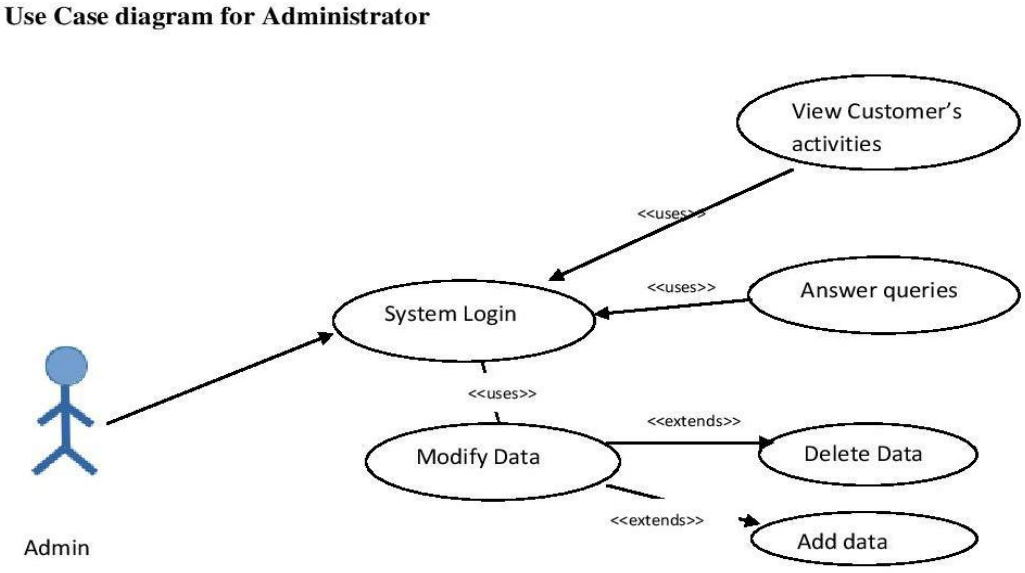

Figure 4: Use Case diagram for Administrator

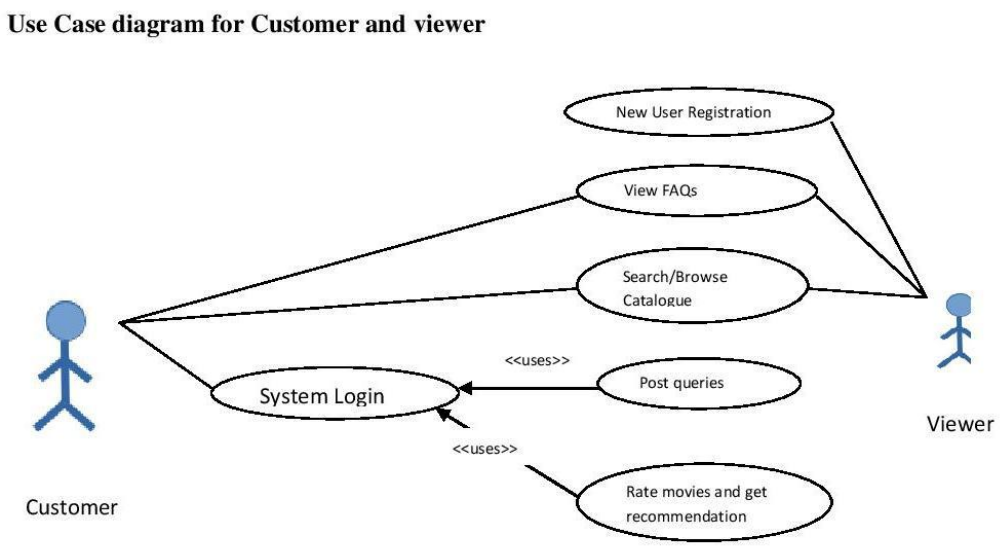

Figure 5: Use Case diagram for Customer and viewer

3.2 Flow oriented elements

A) Context – level Data Flow Diagram

The data flow diagram enables to develop models of the information domain and functional domain at the same time. A context level DFD for the online music recommender system is shown in figure 1. The primary external entities (boxes) produce information for use by the system and consume information generated by the system. A context level or level 0 DFD has a single process. The labelled arrows represent data flowing between the entities and the processes.

In this diagram, the main process is the system/website which gives recommendations. There are three entities: administrator, customer and viewer. The information flows between the entities and the process as shown in the diagram.

Data flow diagrams form a part of the architectural, interface and component level design in the design model.

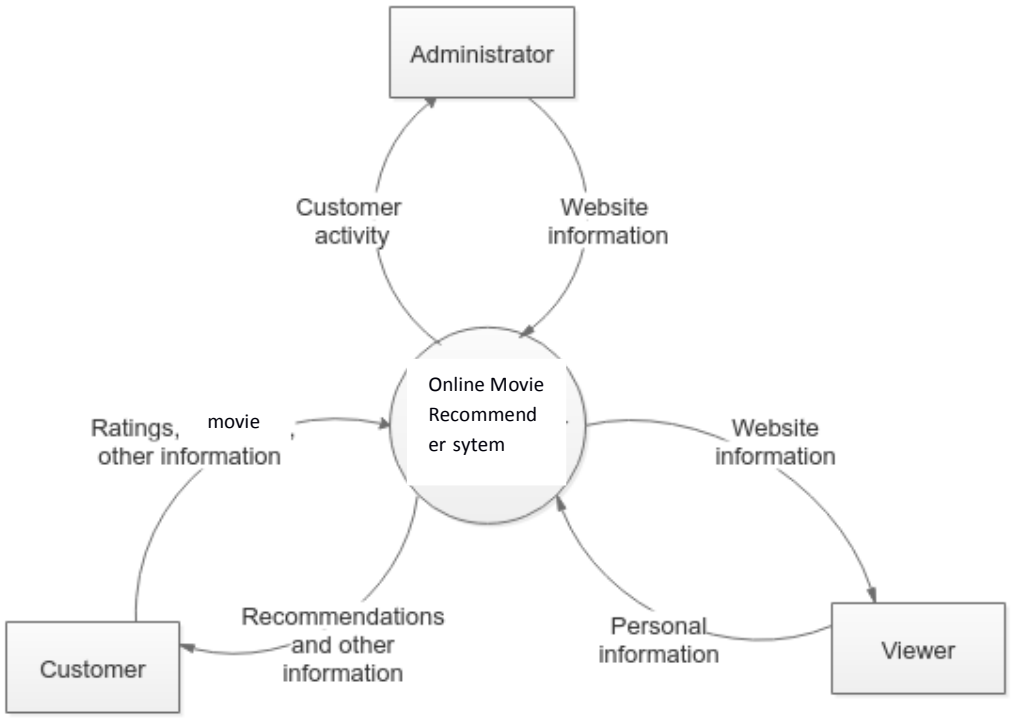

Figure 6: Level data flow diagram

3.3 Web ML Diagrams

A) Data Model (ER diagram)

The data model, also called the structural model, expresses the data content of the site in terms of the relevant entities and relationships. It also indicates the number of participants of each entity-type that are allowed to participate in the relationship. Entities are represented as rectangular boxes with their attributes listed within. Labelled lines indicate the relationships.

The diagram above shows the important entities present in the project and indicates relationships.

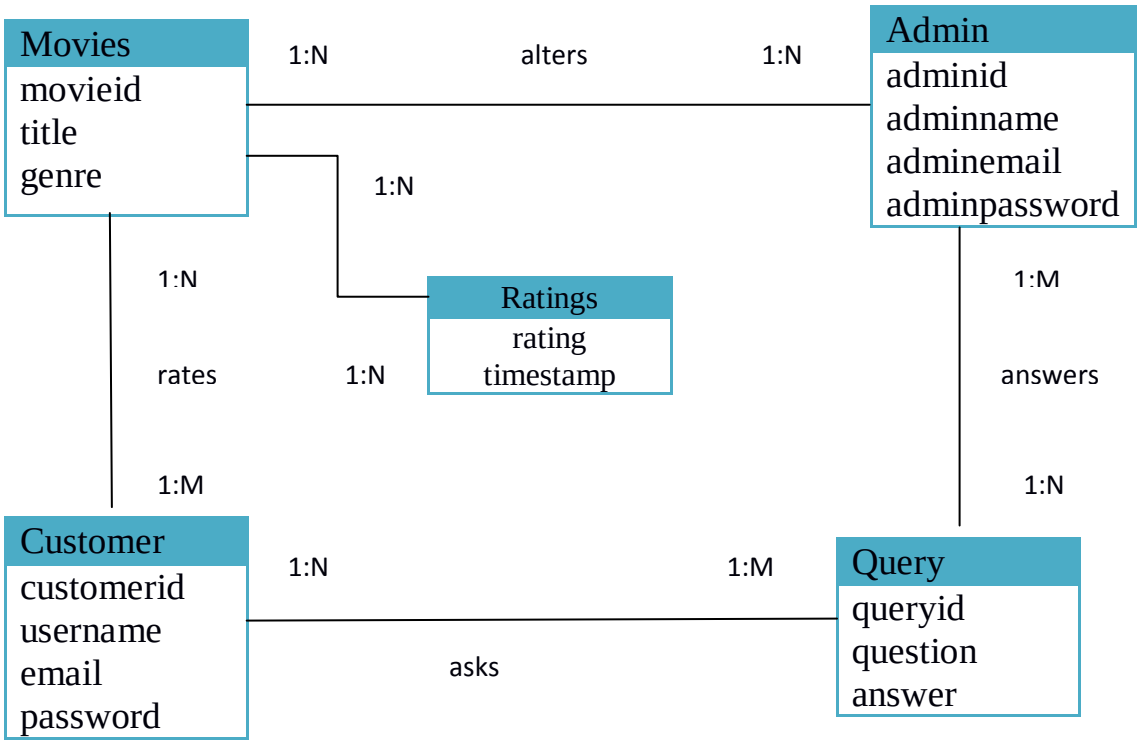

Figure 7: Data model

B) Hypertext model

Hypertext models represent the layout of the website in terms of its page composition. This includes both navigation (how pages are linked to one another) and composition (what is contained in those pages). Various symbols are used to represent the web page components - like the index icon for listing a set of records from the database or the entry icon for accepting input from the user. Direct icons link the current page to another depending on the functionality required and are represented by boxed arrows. A set of related pages are grouped together under a single hypertext model diagram and the extent of the complete website can be shown using multiple hypertext diagrams – as in the case for this project.

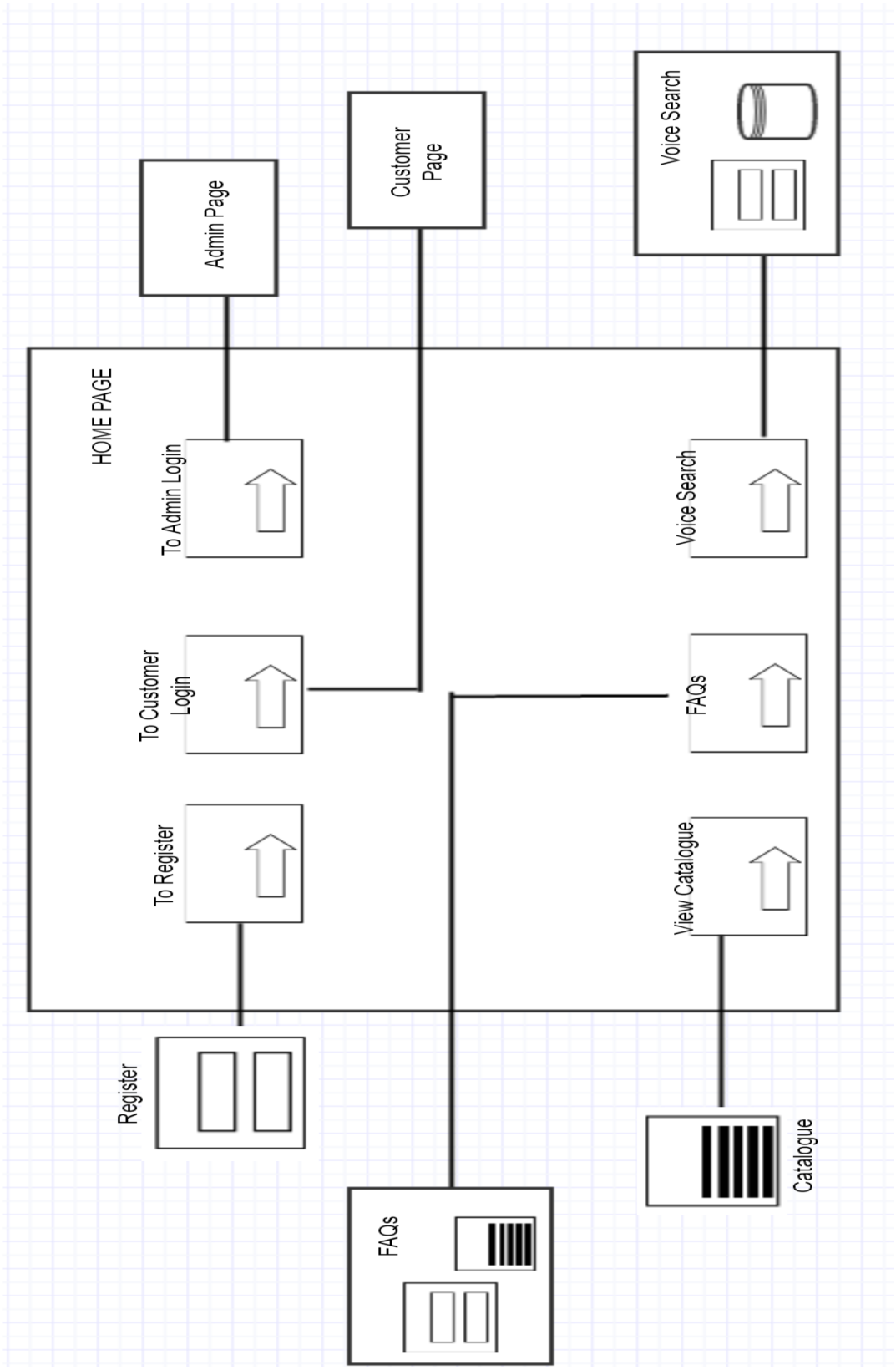

Figure 8: Hypertext model for homepage

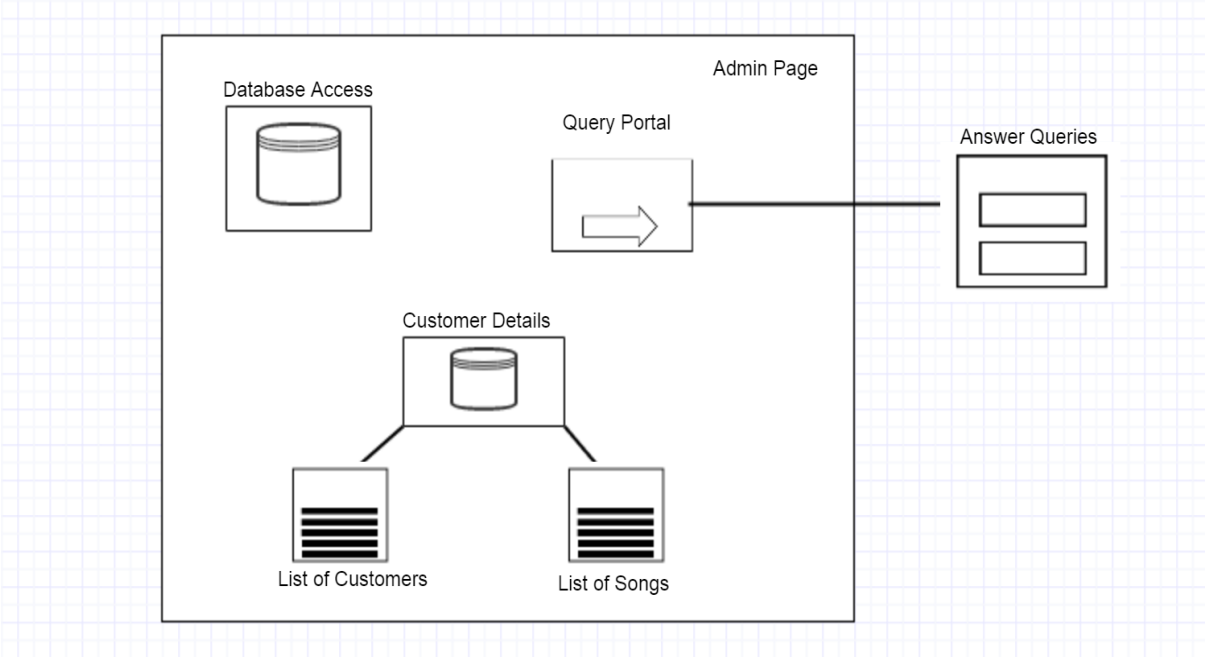

Figure 9: Hypertext model for Admin UI

3.4 Behavioural Diagrams

A) Sequence Diagrams

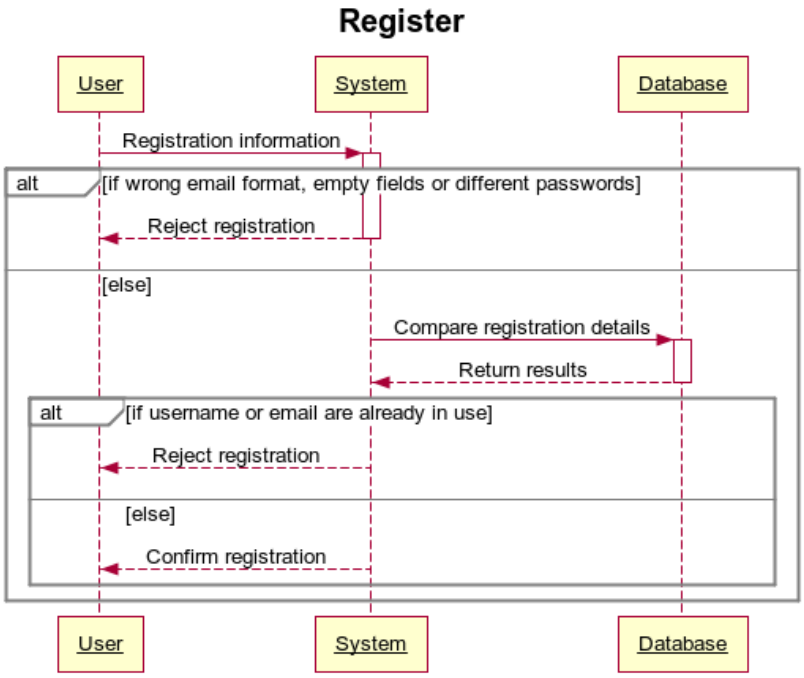

Figure 10: Sequence diagram for register

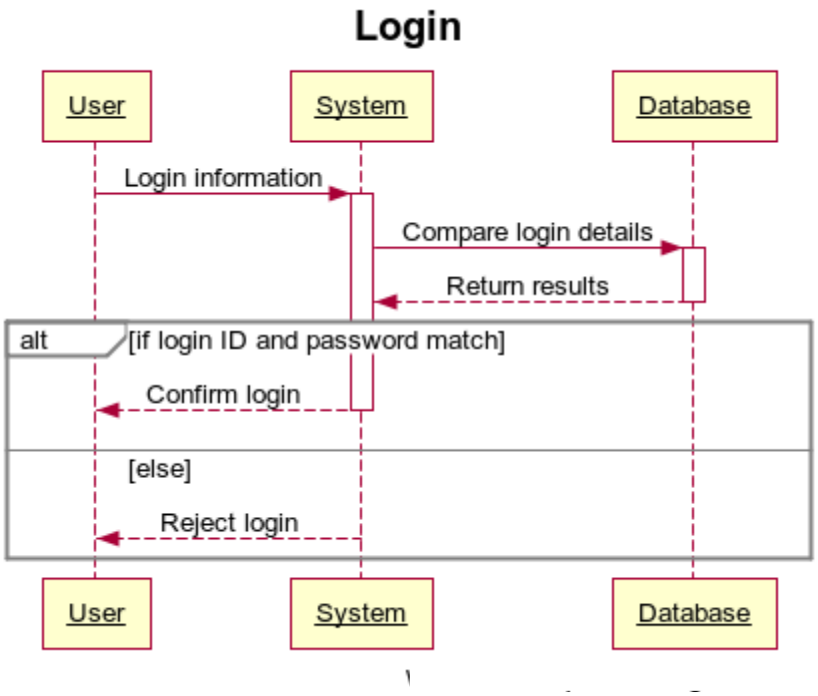

Figure 11: Sequence diagram for login

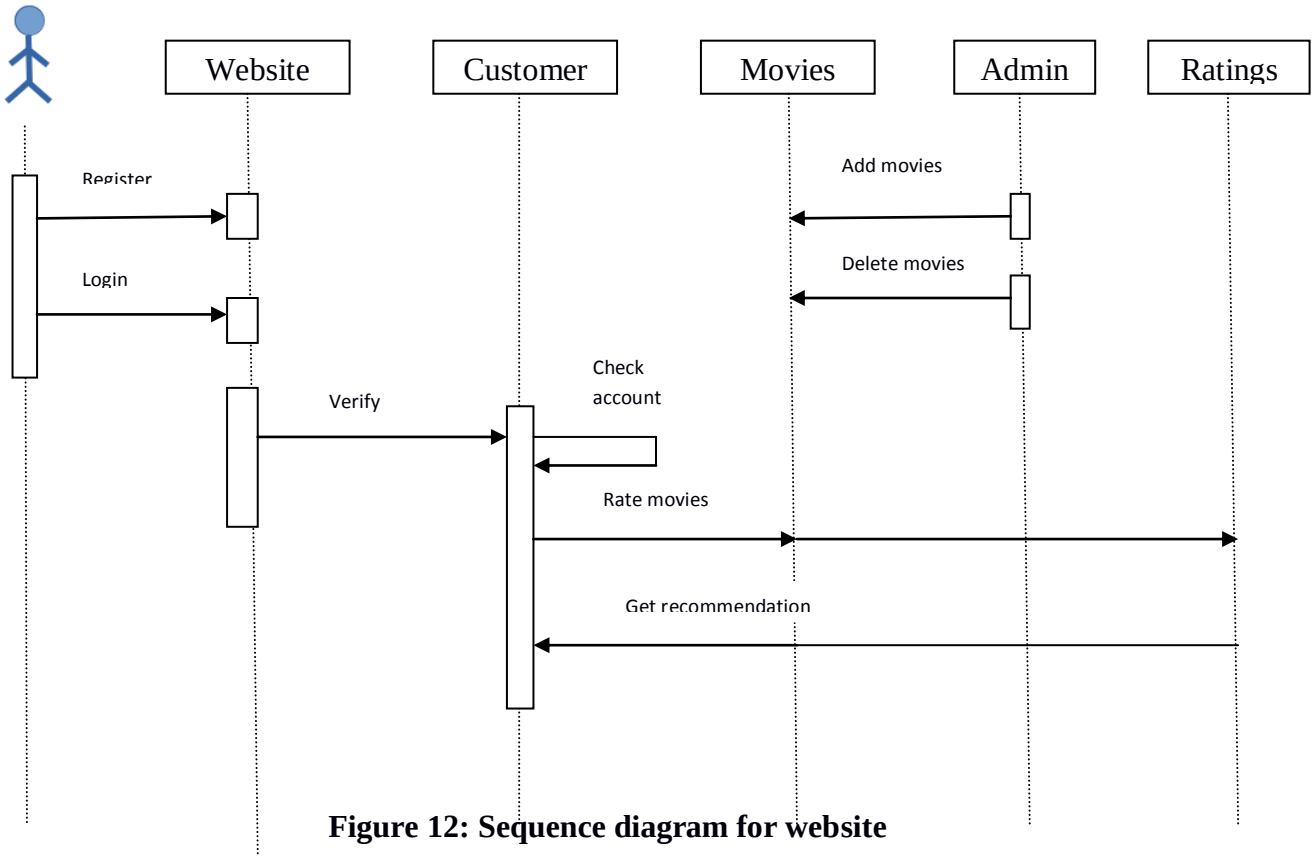

Figure 12: Sequence diagram for website

The above figure illustrates the sequence diagram of Online movie recommender system. Each of the arrows represents the event and indicates how the event channels behaviour between different entities like Customer, Admin, System and the Database. Time is measured vertically downwards and the narrow vertical rectangles represent time spent in processing an activity.

In the above sequence diagram we can see that, how different events viz. searching a movie, logging in, rating a movie, modifying database, etc. is handled by the system and the sequence in which each of these are processed. A sequence diagram forms a part of the interface and component level design in the design model.

4. Restrictions, Limitations and Constraints

1. It does not provide sharing option on other social media sites.
2. The only supported languages for using the website is English.
3. The customers cannot download movies.