1. Submit a single written report in WORD or PDF format (single-spaced, font size=12) that concisely and accurately describes your work in terms of requirements analysis, design, implementation, test and any remaining issues. The report should include the following content:

a. An introduction section that briefly introduces project

Our project is MovieDB, We intend to create a database that include all movies released from 2007 to 2011. The movies are classified by 5 criterias, which are Genre, CriticalRating, AudienceRating, ReleaseDate, and Budget. Public users are able to search movies by one or more criterias and registered users are able to add, delete, or modify the movie database. Registered users need to signup with username, password, and email address, their information and behaviors will stored in user table to help us analyze client preference in the future.

b. An initial estimate of effort for completing the project

Our initial plan is to separate project by two part: Database Development and Query & Front-end development. In the database development part, we collect all of movies released from 2007 to 2011 and search movies’ detail information online, which include Genre, CriticalRating, AudienceRating, ReleaseDate, and Budget. We finished the database with 562 records and use MySQL to develop query language on our database. In the front-end development part, we plan to use NetBeans for the IDE development, use HTML for the UI design, use PHP for server side design, and use Apache as the server. The expected result is to query a film depends on movies’ criteria, delete, modify, and add a current movie record.

c. A project management plan describing the tasks assigned to each team member

In the whole project, Wenyu He is responsible for database development and management, which include collect movies, classify movies by each criteria, create SQL language on database - users could directly search movie name, genre, or set the searching range in CriticalRating, AudienceRating, ReleaseDate, and Budget. Users can search the movies that satisfy one or more criterias;

Xujian Zhang is responsible for UI design by using HTML which include homepage, sign-up page, registered-users page, modify page, and delete page. Users could search movies, sign-up or sign-in to their account on homepage, site will redirect to sign-up page when users click sign-up bottom. Registered users could search, add, modify, and delete movies on registered users page, site will redirect to modify and delete pages when registered users click modify and delete bottoms.

Xiang Liu is responsible for developing service side module by implementing PHP, including creating search module to manage database and SQL language; creating insert, modify, and delete modules that allow registered users to insert, modify, and delete movies; creating sign-in module to compare the received data with the data recorded in the user table; creating authentication module to check the username & password in the front-side cookies with the username & password in the server-side user table.

d. A section on requirements analysis: describing functional requirements, non-functional requirements, and constraints

**Functional Requirements of *Movie DB:***

For public users:

Public users can search movie records based on Movie titles, genre, rotten tomatoes rating, audience rating, budget, year of release.

Public users can register an account with username, password, and email address.

For registered users：

Registered users can use the search function.

Registered users can add, modify, delete movie records.

**NON-Functional Requirements of *Movie DB:***

Easy to Use. Users can get results without any tutorials.

Security. Authentication is required for any potential changes in the movie records.

**Constraints:**

If a user tries to modify a current movie, he or she could just modify other information instead of the movie name.

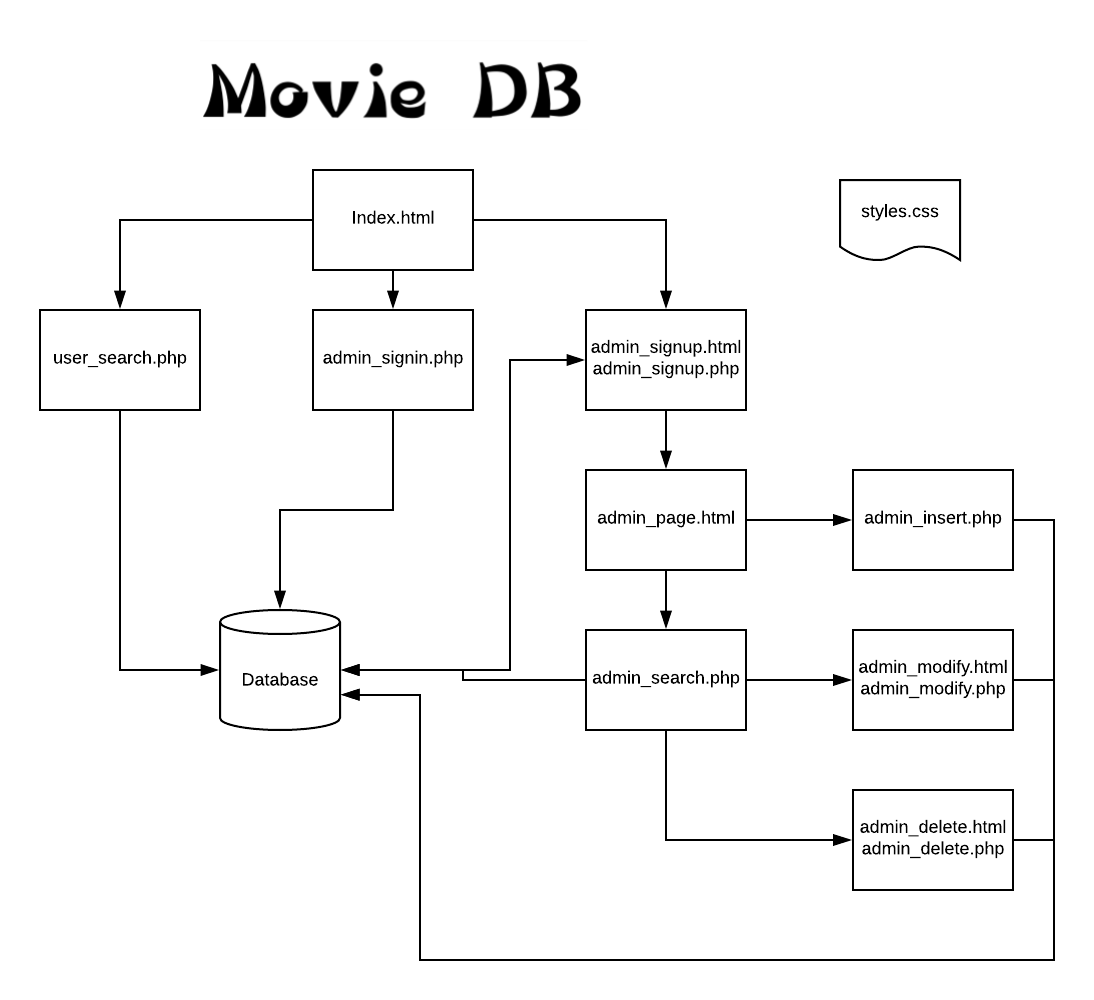
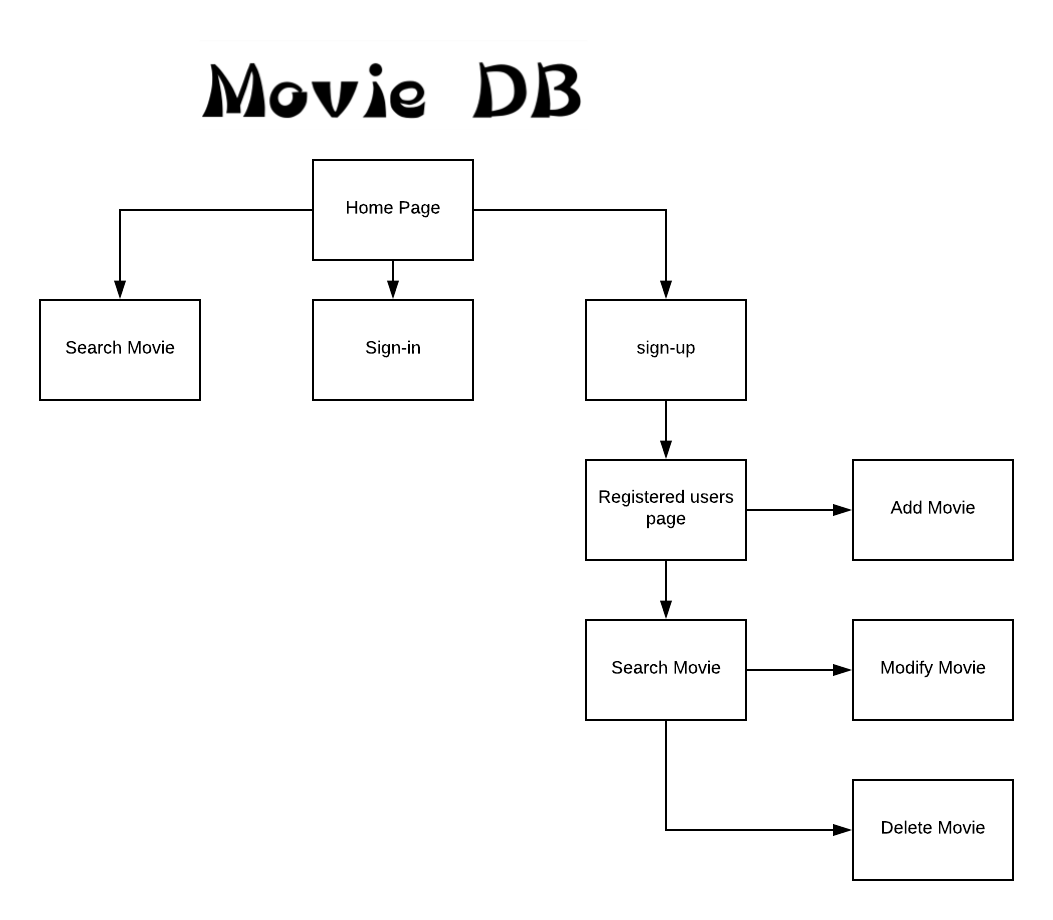
Depends on the data types of the information of a movie, users can’t add or modify an information with a wrong data type. For example, users can’t modify a year to any character.

Public user couldn’t use the function of adding, modifying or deleting a movie.

When a public user signed up, he or she could use the function of adding, modifying or deleting a movie.

e. A section about the design including system architecture, Web technologies, any graphical diagrams if applicable

**system architecture:**



**Technology used:**

Use NetBeans as the IDE of development.

Use Html for UI design and PHP for server-side design.

Use MySQL as database

Use Apache as the server.

f. A section on user interfaces describing any human-computer interaction principles considered in the UI design

The main purpose for designing Movie DB website is giving users a chance to search movie information easily. Not only they can search a movie, but also they can add, modify, delete movie information easily.

For affordance, we put all of our input diagram in the middle of pages. Users don’t need learn any manual and use the website directly. They don’t need to find any input diagram, button because Movie DB show them obviously.

For signifier, we put explanation before or after any input diagram. Users could easily know the function or the part they try to use. We put simple but direct name on the buttons such as “search”, “sign in”, “sign up”, “add”, “modify”, and “delete”. After any operation occurs, users could always see a button called “go back”. If users press it, it will easily lead users to home page or registered user page.

We don’t hide any navigation. So when a user tries to sign up, he or she could see the differences between public user page and registered user page: another section of the page will appear. Also, when registered users try to search any movie, they will find two new buttons which are “modify” and “delete”. User could easily know the function of all the buttons. If users want to add a new movie, Movie DB will provide a new page to guide users fill all the information of the movie, and the same as when users want to modify a movie.

g. A section on implementation. This section should describe the implementation techniques, test cases, and any remaining issues.

In implementation, at first we transferred our dataset from .csv file to .sql file by using MySQL. We created several .php and .html files to make sure the connection between Movie DB and MySQL is established and stable. After that, we discussed how our website should be looked like. When finish designing the UI, for front-side which is .html designing, we developed all pages based on functional requirement such as home page, sign-in page, sign-up page and so on by using NetBeans as IDE. Then we tested all the pages to make sure all function has set. Then depends on the .html, we developed various .php module to initiate all methods and functions on every .html file. Thus, we successfully connected the database and Movie DB. We used apache as our server which is provided by XAMPP. Last we tested all function to make sure the whole website could work well.

The remain issues for the Movie DB: 1) we need to try to make our website look more beautiful; 2) we also need develop our cookie function and to identity any user from any device.

h. A section on results describing the final results of the project using relevant screenshots

i. A conclusion section that briefly summarizes actual effort spent and lessons learned from doing the project.

In this project, we learned the whole process of web system and architecture, we understand how to integrate database management skill, IDE development, UI design, and server side design to develop a exciting website. These skills would extremely useful for our future career development. In our team cooperation, we breakdown each part to team member who is specialized in and we are willing to help others when someone encounter the problem. we not only improved our professional skills, but also developed our team working and communicating skills that make us to be a outstanding team player to handle the complicated project in the future. Overall, we believed that we did an outstanding job.