Problem statement:

Since more and more recreational facilities are available to us now, people become confused about how to spend their spare time during their weekends and vocations. For example, one may not know what movies are now showing and what movies will be matched to him. It takes time on searching movies, finding comments, finding category and finding cinemas. There may be a much convenient method to search the results.

We are trying to develop the website that focus on providing tips for people based on their preferences.

In addition, we also provide the movie recommendations, the location of the theatres as well as the timetables and tickets links.

Proposed solution:

We divide our website into two sections. The first part is called movie recommendation We will provide the movies to the users according to their profiles which contains the movie type and the available locations. The second part is about the movie details. Once the user selects a certain movie, the page will show the description, rating and the link to the official website, after clicking the movie’s image, some more details will come out such as the theatre names and the show times. We divide our website into two sections.

The first part is called movie recommendation We will provide the movies to the users according to their profiles which contains the movie type and the available locations.

The second part is about the movie details. Once the user selects a certain movie, the page will show the description, rating and the link to the official website, after clicking the movie’s image, some more details will come out such as the theatre names and the show times.

Architecture:

First, we create the pages for the login and register. Users can choose their preferences during the register. After that, they will enter the recommendation page and we will make use of our APIs to get the information from other websites.

There is a strong relationship between the profile page and the recommendation page and our project can send the user's profile information to the recommendation page to show the correct lists of the movies. The detail page also depends on the former pages and we will provide more information about them.

APIs Discuss and APIs we used:

Movie:

Rotten Tomatoes API v1.0

Key: vxmeppxp23dqznz7ydwxt466

docs: http://developer.rottentomatoes.com/docs

func: http://developer.rottentomatoes.com/docs/read/JSON

http://api.rottentomatoes.com/api/public/v1.0/lists.json?apikey=vxmeppxp23dqznz7ydwxt466

Cinema and theatres:

gracenote

Key: 9gnk9dw7rbpzg55wmk8qamn9

docs: http://developer.tmsapi.com/docs/data_v1/

zipcode: //www.zipcodeapi.com

Use cases:

1.Use Case: create account

Description: the users have to build their own account in order to make the function of the website

available to them including the user name and the password

Precondition: the account is built and identity is validated

Steps:

[user] – create account request

[application] – get prepared for the new record and show the sign in page

[user] – set the user name and the password as well as the E-mail address

[application] – create the account and send the confirm email to user

[user] – confirm via email and get the user name and password.

[application] – activate the account for the user.

Post condition: the record for the login users increased

Alternate paths: user provides the password or the email is invalid.the application should send the

message that ask the user reset the illegal information

.

2.Use Case: update preference

Description:after logging in the website,the user will be asked to finish his preference selection which

will help him in the future.At the same time,the user may change his interesting or other

activities are needed.

Precondition: preference exists and user authenticated

Steps:

[user] – update the preference based on their interests

[application] – replace the former record with the new one

Post condition:record is update after the operation

3.Use Case: search location

Description:the user enters cities,towns,states,address,zip codes for searching information

Precondition:the location is legal

Steps:

[user] – enters a location as zip code

[application] – responds with a map of location of interest

[user] – select a location of interest

[application] – display details about the location of interest

[user] – confirms selection or dismisses selection

4. Use Case: choose cinema

Description:there will be many cinemas that match user the interests,the user can decide one of them
as the best choice for him

Precondition:one or more cinemas exist

Steps:

[user] – search the matched cinemas

[application] – lists all the matched cinemas

[user] – select a cinema as the best match

[application] – display all the details of the cinema including the open hour and the films

[user] – confirms selection or dismiss selection

5.Use Case: Auto recommendation

Description: After the user log-in and get the home page, the system automatically lists some items of
entertainment (sports matches, new films, etc.) based on the user’s information (interests, location,
etc.) and the current date.

Precondition: users successfully log in.

[application] – list several recommendations, and advance search selection

[user] – choose one selection

[application] – turn to the web page showing the details

[user] – make an advance selection: fill in location, date and items(films, sports) and submit

[application] – show new list of items

6. Use Case: check reminder

Description: user may be confused if they have several schedule in a time,so the application provides

the reminder function to prevent users from miss the date

Precondition: user already login

[user] – start the reminder function

[application] – responds with a schedule of the user

[user] – check the schedule recently

[application] – provides the reminder 1 or 2 days before the activities

7. Use Case:get useful link

Description: for some activities,user need some special links to ensure their safety

Precondition: user confirms the service

[user] – confirm the service

[application] – display all the details about the services

[user] – select one of them and confirm

[application] – show all the useful tips

8. Use Case: Add to my schedule

Description: After the user choose the time, location and movie name, the information can be added to my schedule.

[application] – list several array of records about times, locations and movies.

[user] – choose one certain record.

[application] – turn to the web page showing the details

9. Use Case: Filter search result

Description: set options and filter search result.

[user] – reset search references and submit the changes.

[application] – re-search and present movies based on user’s selection.

10. Use Case: Search single movie based on movie name

Description: search single movie by its name.

[user] – input movie name.

[application] – show the results and details.

11. Use Case: view details

Description: view movie details

[user] – select single movie on the page.

[application] – show movie details.

12. Use Case: Go official website

Description: provide a link to official website of movies.

[user] – click on the button.

[application] –link the user to the website.

13. Use Case: See tickets

Description: provide a link to see tickets.

[user] – click the button.

[application] – turn to the next page to show the available theatres and show times that near the user's location.

14. Use Case: Logout

Description: Different users can access our website using the function logout.

[user] – click the button.

[application] – return to the login page.

15. Use Case: List all the available theatres

Description: After the user select the certain movie name, we will provide all the theatres that are available. The user can then choose the proprite one.

[user] – User clicks the button to access the detail page

[application] – List all the available theatres.

16. Use Case: List all the available times

Description: After the user select the certain theatre name, we will provide all the times that are available. The user can then choose the proprite one.

[user] – User clicks the button to access the detail page

[application] – List all the available times.

17. Use Case: show user's schedule

Description: show all of the events in schedule in order of date.

[user] – click show schedule button.

[application] – Provide all the schedule the user has.

18. Use Case: Add comments and remarks

Description: The user can add comments and remarks in this page for future use.

[user] – type the comments and remarks in the text area.

[application] – reset the details the user provides.

19. Use Case: remove the schedule event

Description: For some reasons,the user has t change their schedule and our application provides the functions to remove the items from the user’s schedule.

[user] – choose the item he wants to remove and clicks the remove button.

[application] – delete the item from the schedule list.

20.Use Case:get useful link

Description: for some activities,user need some special links to ensure their safety

Precondition: user confirms the service

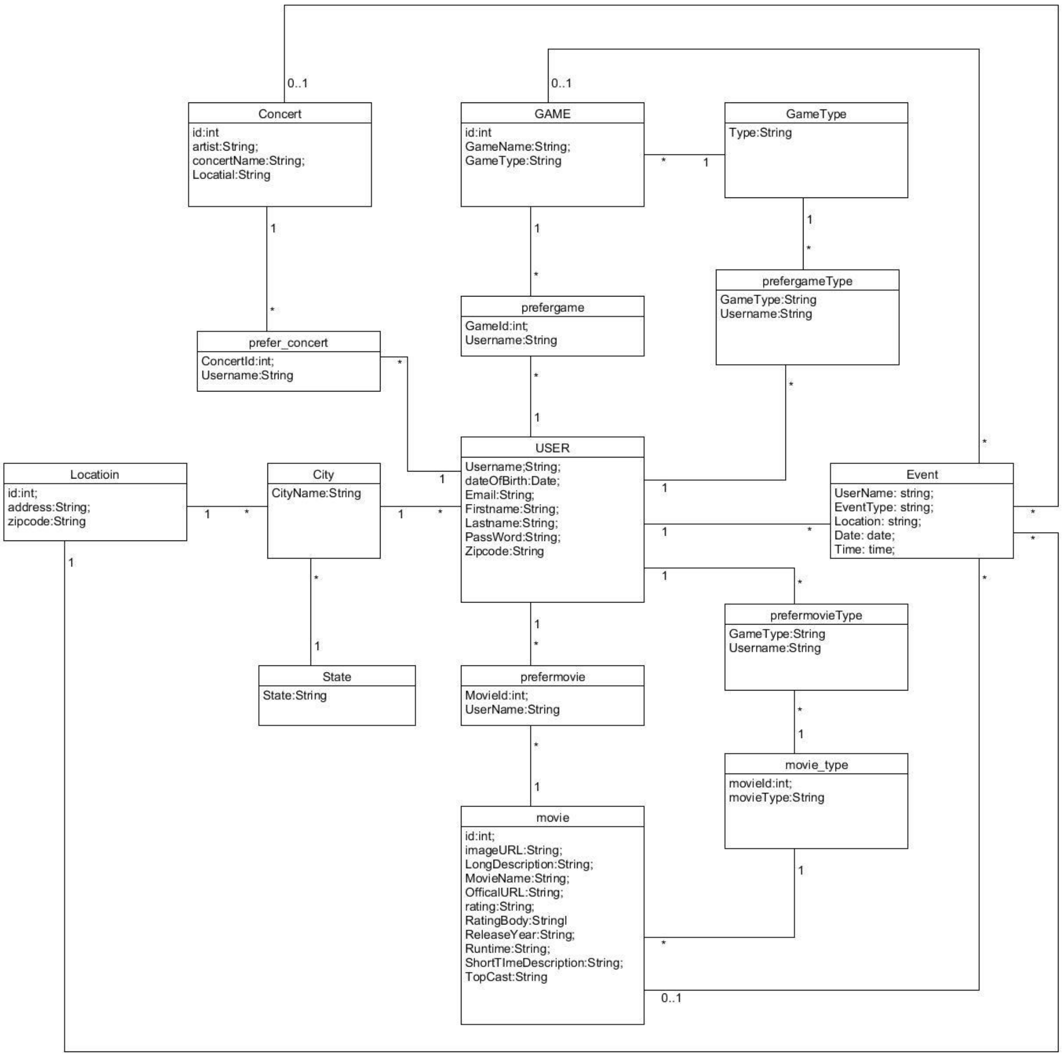
[user] – confirm the service

[application] – display all the details about the services

[user] – select one of them and confirm

[application] – show all the useful tips

Data model:


0..1
0..1
Concert
id:int
artist:String;
concertName:String;
Locatial:String
GAME
id:int
GameName:String;
GameType:String
*
1
GameType
Type:String
1
1
1
*
*
prefergameType
GameType:String
Username:String
*
prefergame
GameId:int;
Username:String
prefer_concert
ConcertId:int;
Username:String
*
*
*
1
*
USER
Username;String;
dateOfBirth:Date;
Email:String;
Firstname:String;
Lastname:String;
PassWord:String;
Zipcode:String
Locatioin
id:int;
address:String;
zipcode:String
City
CityName:String
1
1
*
1
*
1
*
Event
UserName: string;
EventType: string;
Location: string;
Date: date;
Time: time;
*
*
1
1
*
*
*
1
*
1
State
State:String
prefermovieType
GameType:String
Username:String
prefermovie
MovieId:int;
UserName:String
*
1
*
1
movie_type
movieId:int;
movieType:String
1
movie
id:int;
imageURL:String;
LongDescription:String;
MovieName:String;
OfficalURL:String;
rating:String;
RatingBody:Stringl
ReleaseYear:String;
Runtime:String;
ShortTImeDescription:String;
TopCast:String
*
0..1