Project Report

CS5200

Huazong Liu
Xingchen Cui
Xinmeng Lu

1. Problem statement. Tell us what were you trying to solve

Overview of MAGIC: THE GATHERING
There are about 50,000 cards, which have been released, sometimes you want to add a card to your deck list but you don't know the specific information about the card. Moreover, sometimes you want to know which card is better or you want to tell others this card is good, and many other functions. Our program can implement these functions.

Problem statement:
Magic: The Gathering (MTG; also known as Magic) is a published by Wizards of the Coast. Magic was the first trading card game and it's very popular in American. It has approximately twelve million players. Magic can be generally played by two players each using a deck of 60 printed cards. Each game represents a battle between wizards.

2. Proposed solution. Explain how you propose to solve the problem

Online API:
For Search Function, we need to implement typing one card's name and send request to online server to get data from public API, then we get the specific information, and display on our page. This search will generate a history to local database. You can check your entire search history if you want. For Browse Function and Random Card function is similar.

Local Database:
We need to implement login, registration and update user's information, rate cards, comment cards, etc. We need to implement check all the record (Search History,

Rate Record, Comment Record). So we should manage all the data in our database, and also need to set Primary Keys and Foreign Keys. Some of these functions need to use Mysql query.

3. Architecture: Discuss the various parts of your project and how the depend on each other.

Part of our project is depend on the information from the api website. The related functions are search card by name, generate a random card, get game rules and browse new released set.
And part of our project is managing database. These related functions are user registration, user log in, manage user profile, search card history, add card to favorite list, add card to shopping cart, user rate cards, get card's average rate and get top 10 high rated cards.
Those functions which related to user is depending on user data stored in database; while related to card information s depending on card data get from api website. Then use Servlet to make data connection between webpages.

4. APIs Discuss the APIs you used

urlApi = http://api.mtgdb.info/cards/[name]
urlApi =http://api.mtgdb.info/cards/random
urlApi = http://api.mtgdb.info/sets/[id]/cards/"
urlApi = http://api.mtgdb.info/sets/[id]/"

5. Technology stack that you used

Servlet

JPA

Tables

Fields

Composite Primary Key

Foreign Keys

Mapping Tables

Association Tables

Enumeration Tables

Entity Classes

Entity Fields

DAOs

Create Methods

Find One Methods

Find All Methods

Find Other Methods

Delete Methods

Update Methods

End Points

JSON Producers

JSON Consumers

GET Methods

POST Methods

PUT Methods

DELETE Methods

Server Web Service Client

Browser Web Service Client

Online Web Services

Online Web Services

User Interface

Pages

Input Fields

Links

Buttons

Data Tables

Data Lists

JavaScript Libraries

CSS Libraries

6. Use cases. List and discuss the use cases

1.

Use Case: Search a card by name (Exactly Search)

Search card from api.mtgdb.info by name. After search the name of card, it will return all information for this card. If the name of card is wrong, it will show the alert message ”card name not found!”

2.

Use case: Registration

Description: Users can create their account by username and password; they can also set their nickname, e-mail, etc., which are optional. If user name is used before, password does not match re-type password or other exception, the webpage will report the relative error message.

When user clicks “Register” button, then webpage turn to relevant page, if user submits register form, store user information in database.

After register, user can log in and manage their account.

3.

Use case: Log in

User must have an account to use this website. User needs to register first. After that, user can log in every time by using username and password.
When user log in, all actions is recorded with his or her name, such as searching history or writing comment.

4.
Use case: Update user profile
Description: Users can update or delete their information whenever they want (password, nick name, e-mail, etc.).
After click the button, Then turn to relevant page, if user submits update form, update user information in database.

5.
Use case: Random Card
Users can browse a random card. After click the button, user can get the information of a random card.

6.
Use case: Favorite List
After search card or get a random card, user can add card in his or her favorite list. One user cannot add the same card twice; in user's Favorite List, all cards only show one time and cannot be repeated.

7.
Use Case: Shopping Cart
User can add cards in shopping cart, and user can add several certain cards by typing in the number of cards.

8.
Use case: Shopping History

User has shopping history. When user order cards or these cards delivered there will be a record stored.

9.

Use case: Shopping Order Number

Every time user order cards may generate a record, and state status is order. And the order number for the card for the current user is recorded. If the same user order the same card next time, and the order number will be added to the original order number.

10.

Use case: Shopping Delivery Number

User ordered many cards, and all of these cards or part of them may be delivered. Every time cards be delivered may also generate a record, and the state status is delivery. The history number for this card for the current user is recorded; at the same time, the order number is detracted.

The sum of order number and delivery number is the total number user has bought that card.

11.

Use case: Search History

Description:

Every time user searches a card will be stored in local data base.

Users logged in to the system can browse all cards he or she searched before.

12.

Use case: Browse by set

Description:

Users can view all cards in a specific set, when user type a set name, it will shows all cards in the set which you request.

13.

Use case: Game Rules

Description: Users can browse the game rules by clicking the button named ‘Game Rules’.

Then the game rules will be displayed on the screen.

14.

Use case: User rate card

After search card or get a random card, user can rate this card from one to five stars. The rate of a certain card is the average number of all its rates getting from different users.

User can view the rates of cards he or she grades before.

15.

Use case: Average rate of card

All rates of a certain card will be recorded, and by clicking the button, it will return the average rate of this card graded by all users.

16.

Use case: Top 10 rate cards

Description: It is a list that shows users the top rated card, This function will be Real-time updates, When you rate a new card, the average rate of this card will update instantly, and it might affect the Top 10 List.

By clicking “Top 10 rate cards” button in homepage, it will show Top 10 rate cards are displayed.

17.

Use case: Comment

Description: Users can write comments about cards.

These comments are public. When other user search this card, he or she can view all comment created by different users.

In user's profile, he can view all the comments attached on different cards make by himself.

18.

Use case: Random card history

User can view there generating random cards history. Every time user log in can view his or her random generating cards history.

7. Data model. Discuss the UML class diagram