Team name: *Leorik*

Team members: *Dao Quoc Cuong & Pham Vu Hung*

Proposed Level of Achievement : Gemini

Target Audience: Manga readers, whether novice or veteran

Motivation & Aim:

If you are a **regular manga reader**, sooner or later the time will come when you start to get bored of reading the same motif over and over again. You'd start to wander around the forums, looking for good recommendations for **new materials**, but sometimes it's such a chore.

Or if you are interested in **picking up manga** as a new hobby, finding **where to start** is the first problem. Friends are the first ones you turn to, as they are probably also the one who introduced you to the manga world. But more often than not, **they don't know** what kind of materials you'd be into.

The world of mangas is **complex and expansive**, even for experienced readers. Whether you want to start getting into mangas but don't know where to begin, or you're a veteran reader who just finished a series and want to try something different and fresh next, it's always a **hassle** to look for a new manga to pick up, especially one of a new genre.

Comparison to other implementations:

At this moments, there are a few websites that recommend new manga for their users; however, most of these just require us to provide them with a list of our read mangas, and from there they will give us recommendations consist of mangas with similar style and genre. Hence, our app aims to assist a completely new reader or someone who wants to explore new materials, to cater them with interesting mangas that they might have thought that they wouldn't be able to enjoy (but actually do).

Project Scope:

A Manga recommender web application which helps reader choose new materials from unorganized public source, including:

A **webpage** which acts as the main user interface where users can input reading preference, access reading profile and receive recommendations. Here the user first takes a quiz which aims to determine their preference and then receives back a customized reading profile along with list of Manga recommendations. The profile is editable by the user as well as sharable.

A **database** to store reading profile information of users for re-access, edit and share. It will be used to aid their future seeking of recommendations.

Description of system:

The system consists of 3 main components: user interface, recommendation algorithm and database.

The user interface is the website, hosted at mangarecommender.000webhostapp.com, built using HTML, CSS, JavaScript and PHP. W3.CSS framework is also employed to ease the design process. The website has a main index page where an overview of the project/application is shown; a quiz which users can participate in to get the reading habits profiled and to be given gitting recommendations; individual manga pages with relevant information and a search function to find them; a user dashboard where users can register and sign-in to save, access, and modify their quiz results/recommendations and reading list.

The recommendation algorithm is built into the quiz page, using entirely JavaScript. The major genres as well as content categories are used as scoring indexes. Each question pertains to a set of indexes, and the user's answer will determine which index(es) to add/minus points to. The final score will determine which genres and categories to be used to search for recommendation.

The app communicate with our database to display useful information and relevant external links every time the user access an individual manga page entry, which can happen via the recommendation list, the read list, or when user manually searches for a manga.

The app manga search function is an implementation of MyAnimeList API, of which can only be used (backend) with a MyAnimeList account and password set.

Lastly, the database is built using VertrigoServ as the local server, with mySQL to handle queries to the server and PHP as the language to send request (from the website). The database consists of 2 tables, one stores the user's username, ID, email, hashed password, quiz's results/recommendation, already-read list; and the other stores a list of manga and their respective information for ease of access and faster display. The 2 way communication between the website UI and backend database is the foundation of the app usability. .

Testing

Speed: In order to ensure a base usability, our system needs to operate in a reasonable time for each features. The slowest part of the program will be the runtime bottleneck for the whole app.

Our first implementation incorporated an API of a database, MyAnimeList (myanimelist.net/api/), to get manga information. However, in testing, the API of this particular site performed very slowly, as it can only accept one query at a time: The API takes 8 second on average to process a query of 5 mangas.

Due to this inflexibility, we switched to another database, MangaUpdates (https://www.mangaupdates.com/). This one however has no functional API so we scrape data directly from its main displaying website via HTML objects resulted from PHP: Document Object Model parser. The data then get collected inside our own database for easier and quicker handling.

This method run time bulk is at the data collection step initially, which only took around 6 minutes for 5000 queries.

However, 5000 manga series might be insufficient to recommend a wide range of mangas, especially for

veteran readers, so we also implemented a method where we scrape data directly from the second database mentioned above.

After rigorous testing, we've reduced the quiz scoring time from 60 seconds to 20 seconds.

Correctness:

At first, we tried to search for mangas by genre using Manga-Scraper API (https://market.mashape.com/doodle/manga-scraper). While we managed to obtain a list of fitting manga, we have no way to know which ones are good and which ones are not. Combining this with MyAnimeList API, which can return the community scoring of queried series, takes an astronomical amount of time to process all series, so it was discarded.

We then switched to MangaUpdates database, which scores the series using Bayesian Average, each series is rated by up to a few thousands people. This gives satisfactory responses, thus is ultimately used in our prototype.

In addition, to know whether or not the quiz is capable of profiling the user's reading habit, we asked a few of our friends, both experienced and not, to try the quiz. The common response is that the top genres the quiz decided that they like seem correct, but the recommended mangas are sometimes too well-known that many would have already read, and sometimes too unpopular that some would be unlikely to start reading them. In conclusion, we have quite a working algorithm by now, but we will have to figure out which mangas are the best to recommend.

Next, we did a few dozens of trials of the quiz to see which results they return and if they are correct. There are some combinations of genres that doesn't return any results. In those scenarios, we should use 4th best, 5th best scoring indexes for searching, instead of only using the top 3.

After we incorporated content categories into our recommendation quiz algorithm, we again ran selected combination of quiz answers and compare the resulted recommendations to those from the first algorithm iterations. We also survey from peers to ask them feedback on which recommendations set they prefer or find more suited. The feedback suggested the new algorithm is an improvement over the old one, as the recommendations are less generic and more targeted.

We will continue to improve the recommendation algorithm, UI design and test for live hosting bug if time permits.

Project Log:

S/N	Task	Cuong (hrs)	Hung (hours)	Remarks
May				
1	Meeting to brainstorm and formulate initial ideas	4	4	Decide on the type of application, the topic, how problems may be tackled.

2	Lift-off Day 1	9	9	
3	Lift-off Day 2	9	9	
4	Meeting to consolidate plans and ideas	4	4	Finalize on realistic targets/features the app might have, tools that can be utilized
5	Researching, familiarizing and experimenting with new languages and platforms (JavaScript, NodeJS, jQuery, HTML/CSS, PHP, Microsoft Azure)	15	15	Download, install, read tutorials and online lessons. ● Cuong focus on jQuery, PHP Ajax and Azure ● Hung focus on HTML/CSS handling and NodeJS
6	Meeting to improve feature implementation based on known tools.	4	4	Determine which features are best implemented using which tools
7	Meeting to discuss the submission of milestone 1	3	3	Online discussion
8	Designing poster	2	2	
9	Making project video	1	1	
June				
10	Meeting to share ideas and organize development plans	3	3	Offline discussion
11	Learning PHP, MySQL and their utilization.	15		How to implement forms, api, database communication (data insertion & collection)
12	Learning and practicing implementation of HTML, CSS, JavaScript		15	To build the website and user interface
13	Meeting to collate learnt knowledge and discuss website design	3	3	
14	Research on how to handle JSON/XML/HTML object	8		To scrape data from website/database without usable API
15	Coding data handling functions and app features	10	5	- Scrape data from online database - Utilize API for searches - Create local database to store users' info
16	Devise recommendation algorithm and its implementation		5	- Compose quiz's questions - Devise scoring system - Request online database for fitting recommendations
17	Develop user interface		10	
18	Meeting to discuss the submission of milestone 2	3	3	

July				
19	Research on PHP Ajax technique	15		
20	Meeting to divide work and consolidate development plan	3	3	
21	Picking relevant content categories for quiz algorithm	2	2	
22	Incorporate categories into recommendation quiz/ algorithm improvement	10	15	Add more questions to fit the categories/genre and restructure the scoring system. *Note:* Categories are different from genres, and are defined by MangaUpdates
23	Coding implementation	16	20	New features: - Manga individual page - Function to search for manga - User's read list and dashboard
24	Design, improve website		5	
25	Meeting to discuss submission of milestone 3	3	3	
Total		142	143	