CMPE 451 Milestone 2 Report

CultureMania - Group 1

Table of Contents

1- Executive Summary

We have reached all of our milestone 2 objectives and are proudly running in cloud now.
http://ec2-18-196-2-56.eu-central-1.compute.amazonaws.com/

We have reached the milestone objectives which were:

- **Backend objectives:**
  - Vote Service.
  - Comment to comment Service
  - Recommendation using Semantic Api(DataMuse)
    - User Based (Recommended Feed)
    - Item Based (Right side of the each Heritage Item`s in detailed view.)
  - Search using Semantic Api(DataMuse)
    - Basic Search
    - Advanced Search

  - Furthermore, we added auto generated Avatar`s for each user to make our application more user friendly.

- **Android objectives:**
  - Heritage Item Creation
  - Heritage Item Detail Display
  - Heritage Item Image Display
  - Heritage Item Image Upload
  - Displaying User Avatars
  - Posting and Displaying Comments
  - Recommendation, New, Top and Trending Tabs
  - Searching Heritage Items
  - Voting Heritage Items

- **Frontend objectives**
  - Random - Best - Trended - Recommended - New Feedy
  - Recommended items to the detailed item page

- Comment to comment
- Vote and vote delete

There are still many small bugs and we intend to fix them during the week. Our search and recommendation is working satisfactorily but there is still room for improvement. It`s search space increases exponential and this causes problems in complicated requests. We plan to solve it.

We plan to implement add youtube embedded video to heritage items in the following weeks.

2- List and status of deliverables

- Milestone 2 report
- Project Plan
- Website itself.
  - http://ec2-18-196-2-56.eu-central-1.compute.amazonaws.com/
- Api documentation
  - https://github.com/bounswe/bounswe2017group1/wiki/API-Documentation
  - https://github.com/bounswe/bounswe2017group1/blob/web/web/api/README.md

3- Summary of coding work done by each team member

Mehmet Sefa Balık Team: Android	● Item creation implementation ● Search feature implementation	● Comment functionality implementation ● Leading Android Team
Taha Metin Bayi Team: Android	● Heritage item image upload implementation ● Heritage item image display implementation	● Heritage list swipe to refresh implementation ● Main page heritage list implementation
Yiğit Alperen Bildik Team: Backend	● Image upload	

Mustafa Feyzioğlu Team: Android	● Voting system implementation ● Avatar image implementation	● Tabular format (Recommended, new etc.) implementation ● Visual Polishing
Mehmet Gülşen Team: Backend	● Semantic API (formerly ConceptNet, now Datamuse) implementation	● Advanced Search implementation. ● Heritage search score improvements..
Abdullah Furkan Ilısu Team: Backend	● Basic Search Implementation ● Advanced Search implementation with Mehmet	● Heritage-based recommendation system implementation ● User-based recommendation system implementation
Ali Kireçligöl Team: Frontend	● 5 different feeds in website homepage ● Google Maps Api in item creation(autocompletion)	● Wikidata auto completion in item tag creation ● Avatar`s added to her. items
Hakan Şirin (Communicator) Team: Backend	● Vote Service ● Comment, Heritage item service expansions (is_owner, up/downvote count etc.)	● Item based recommendation review and bugfix. (malfunction in scoring) ● User Profile Avatar Service
Ceyhun Uzunoğlu Team: Backend	● Item based recommendation using DataMuse.	● Data Creation for Milestone 2 presentation.
Basri Yılmaztürk Team: Frontend	● Comment implementation ● Heritage item recommendation and missing data(creator and date)	● Heritage item missing token fix ● Top bar authorization problem fix

4- Project plan

Time Period	**Task Name**	**Assignee**
16.11.2017 - 19.11.2017	Basic search implementation for backend	Abdullah Furkan İlısu
16.11.2017 - 19.11.2017	ConceptNet Semantic API implementation	Mehmet Gülşen
16.11.2017 - 19.11.2017	Item Creation and item detail view implementation	Mehmet Sefa Balık
16.11.2017 - 19.11.2017	Action bar implementation	Mustafa Feyzioğlu

16.11.2017 - 19.11.2017	Main page heritage list implementation	Taha Metin Bayi
16.11.2017 - 19.11.2017	Improvements in API service for comment and heritage item responses	Hakan Şirin
16.11.2017 - 23.11.2017	Image upload backend	Hakan Şirin, Yiğit Alperen Bildik
18.11.2017 - 20.11.2017	Feeds in frontend Comments in heritage	Ali Kireçligöl Basri Yılmaztürk
20.11.2017 - 23.11.2017	Search new Semantic API alternative to ConceptNet	Abdullah Furkan İlısu, Mehmet Gülşen
20.11.2017 - 23.11.2017	Advanced search implementation for backend	Abdullah Furkan İlısu, Mehmet Gülşen
24.11.2017 - 26.11.2017	Backend endpoints for Trending, Top and New feeds	Mehmet Gülşen
24.11.2017 - 30.11.2017	Comment functionality implementation	Mehmet Sefa Balık
24.11.2017 - 27.11.2017	Heritage list swipe to refresh implementation	Taha Metin Bayi
24.11.2017 - 30.11.2017	Heritage based recommendation for backend	Abdullah Furkan İlısu
24.11.2017 - 30.11.2017	Item based recommendation for backend	Abdullah Furkan İlısu Ceyhun Uzunoglu
24.11.2017 - 30.11.2017	Vote deletion endpoint and service	Hakan Şirin
24.11.2017 - 30.11.2017	Voting system implementation	Mustafa Feyzioğlu
26.11.2017 - 30.11.2017	Semantic API's integration into the search system	Mehmet Gülşen
1.12.2017 - 5.12.2017	Auto generated Avatar service is implemented. Owner Avatar image url`s are added to items.	Hakan Sirin
1.12.2017 - 7.12.2017	Semantic API changed from ConceptNet to Datamuse API	Mehmet Gülşen
1.12.2017- 7.12.2017	Recommendation System Bugfixes	Abdullah Furkan İlısu, Ceyhun Uzunoglu, Mehmet Gülşen

1.12.2017- 7.12.2017	Vote API service improved	Hakan Sirin
1.12.2017 - 7.12.2017	Creation of meaningful data and testing	Everyone, Special thanks to Ceyhun
1.12.2017- 7.12.2017	Tab System (Recommended, New,Top,Trending) and avatar image implementation	Mustafa Feyzioğlu
1.12.2017- 7.12.2017	Search Feature implementation	Mehmet Sefa Balık
1.12.2017- 5.12.2017	Heritage item image display implementation	Taha Metin Bayi
1.12.2017- 7.12.2017	Heritage item image upload implementation	Taha Metin Bayi
1.12.2017 - 7.12.2017	Google Maps, WikiData api Heritage recommendation Search	Ali Kireçligöl Basri Yılmaztürk

5- Code Structure

We have three different teams: Backend, frontend and Android. Backend team's code is in "/web/api" , the frontend team's code is in "/web/web", and the Android team's code is in "/android" .
Backend and frontend are working in the "web" branch of git while Android team is working in the "Android" branch.

Whenever we want to implement a new feature, we are creating branches from "web" or "Android" and work on new branches. When the work is done, pull requests into "web" or "Android" is generated. Team members review the code and merge them. If it is something trivial and/or too simple to waste time by waiting for a review, sometimes the pull request owners also merge their own pull requests.

Periodically, the "web" and "android" branches are merged into the "master" branch, which can also be considered as the deployment branch.

6- Evaluation of tools and managing the project

At the project development process we used hangouts for team meetings. For code reviews and code related communications we used github, and also whatsapp for general communication. Hangout meetings were productive and very useful for solving problems that relates all team members. Because of the fact that the problem of being in school and having no course at that hour at the same is so difficult for all of us, hangout was a useful solution for us. We use Github as much as possible. This project is first team project of most of us. For that reason, we were struggled with the tools of Github and Git (pull requests, issues, comments, merges, branches, lots of branches) at the beginning, but we eventually overcome this hard period with little flesh wounds. And also, although it is an unprofessional way of software development communication, we used Whatsapp a lot. We will work on that in the following weeks.

For coding purposes, the backend team is using Python IDE PyCharm by JetBrains. It is easy to use and the built-in terminal makes it easier to run the code and to use git without opening multiple terminals.

For coding purposes, the Android team, we are using Android Studio because of these reasons: Android Studio uses the quick growing Gradle build system that is so integrated, and Gradle is really a great tool, Android Studio features the typical Java code auto completion, the tools and menu items in Android Studio tend to get us where we want to be a little more promptly and effortlessly than other IDEs, Android Studio uses modules to manage and organize your code modules have their own Gradle build files which mean it can state their own dependencies, Android Studio is now released with very less bugs, and provides a more stable performance guarantee than other IDEs and the system needs are lower too and Android Studio has GUI (Graphical User Interface).

In terms of branch usage in git, what we did was creating branches based on features from the web branch and then creating pull request returning to once again web branch when the feature is completed. Adding db to git caused great trouble since it is impossible to merge binary SQLite objects but lesson learned there. We are surely not gonna add db files to git in the future.

Another lesson learned is never skip weekly meetings. It was a really challenging Milestone week because we skipped the meeting in the previous week.