Customer Demo 2 Milestone Group report

Prepared by:

- Bekir Tatar
- Emin Deniz
- Eren Altunoğlu
- Fikri Cem Yılmaz
- Hazar Çakır
- Kürşat Bakış
- Meltem Arslan
- Mehmet Yasin Şeremet
- Selman Berk Özkurt

Group number: 11

Group name: Renaissance

Sub-groups:

Backend:

- Eren Altunoğlu
- Mehmet Yasin Şeremet
- Selman Berk Özkurt

Frontend:

- Bekir Tatar
- Emin Deniz
- Meltem Arslan

Android:

- Fikri Cem Yılmaz
- Hazar Çakır
- Kürşat Bakış

Tag-id: v0.4(for frontend and android) , v0.4.0-backend

1. Executive Summary

As one may remember, our project is a self organized development platform called Renaissance. In milestone 1, we have done the register and login features and the general design and structure of the program. Milestone 2 mainly focuses on the features about “Paths”, where the core functionality lies in our project. According to the plan, users should be able to create paths, navigate to the path page, follow and give rating and effort values to the paths and search for a specific path. Also tags should have “Topics” in it which are fetched from the wikidata. In addition to those functionalities, users should be able to configure their profile.

As planned, nearly all of these functionalities are implemented in the project. The Frontend and Mobile part of the project is fully connected with the backend which means the backend provided all required endpoints. Currently, the user is able to edit its profile page and create and review a path which includes a picture, description, some topics related and several milestones in both web and mobile. Search for a topic, person or a tag is successfully implemented in the front-end and also one can navigate through to the topic page which includes corresponding paths, related topics and related paths parts. In mobile,“my paths” page is implemented and the users can see the paths that are created by them. The overall design is improved for the mobile application.

After the first milestone, we decided to focus on more functionality and less design. So each team member prioritized the coding part over design. Also after getting feedback about pull-requests, we decided to work in a more branched way and create more pull requests.

For the rest of the project, our primary goal is to create a working application. We are pleased with the work done in each sub-group and feedback based mechanism. We want to continue in this way.

2. Summary of work

Company	Contact
Kürşat	• My paths page is implemented • Edit profile page is implemented • Backend connections are made. • Respone decoding is done. • Data models are created. • Follower-followings list is implemented.
Hazar	• "Create path" feature is implemented • UX improvements are done. • "Search" feature is started to implementation but not finished. • Mobile part of the presentation is done by me and Fikri. • Presentation scenario were created with Hazar and Fikri Cem.
Fikri	• Path Page is implemented • Enroll a path feature is implemented.

	• Favorite a path feature is implemented. • Rate a path feature is implemented. • Rate effort for a path feature is impelented. • Path Container click navigation feature is implemented. • Bugfix on home page is made. • Backend connections are made. • UX improvements are done. • Topic page feature is started to implementation but not finished. • Mobile part of the presentation is done by me and Hazar.
Emin	• Setup Test Development Tool (Jest) for Frontend • Topic page • Search for topic and path • Edit profile • Show word cloud images with fallback
Meltem	• The first version of the web path page was designed. • The technologies used in front-end, back-end and android were added to the group wiki. • Presentation scenario were created with Hazar and Fikri Cem. • Web part of the Milestone 2 presentation was made. • "Progress according to requirements" part of the report was written (functional part)
Bekir	• Path Page is implemented. • Enroll page functionality is added. • Backend connections related to the path are made. • "Progress according to requirements" part of the report was written (non-functional part)
Selman	• Help revising plan for the milestone • Some API definitions • Finish path endpoint • Rate path endpoint • Rate path's effort endpoint • Finish path endpoint • Favorite/Unfavortie path endpoints • Postman testing of many endpoints • Pull request reviews and conflict resolutions for some backend features. • Wordcloud generation endpoints for users and paths

	• Help with semantic web integration
Eren	• Edit path endpoint • My paths endpoint • My topics endpoint • Get path endpoint • Search topics endpoint • Get paths of related topics endpoint • Enroll path endpoint • Get followed/enrolled paths endpoints response update • Follow user endpoint • Review a part of backend implementations • Create issues for some missing endpoints • Deployment of the latest stable code
Yasin	• Planning on milestone 4 endpoints • Path modeling • Create Path Endpoint(as observer) • Wikidata API Research • Follow/Unfollow User - Get Followed Users Endpoints(as observer) • Enroll/Unenroll Path - Get Enrolled Paths Endpoints • Follow/Unfollow Path - Get Followed Paths Endpoints • Rate Path - Effort Path - Finish Path Endpoints(as observer) • Search Path Endpoint • Tests on Postman • Contributed API documentation

3. Progress according to requirements

1. Functional Requirements

1.1. Accounts

- **1.1.1.** User *shall* be able to register by providing a valid and verified e-mail address, username, and a strong password. -> Completed.
- **1.1.2.** User *shall* be able to login by providing verified e-mail address. -> Completed.
- **1.1.3.** The user *shall* be able to delete their accounts. -> Not started

- **1.1.4.** The user *shall* be able to reset its password via providing its e-mail address and via passing a challenge-response test such as "captcha" . -> In progress
    - **1.1.4.1.** In this case, System *should* send a mail for resetting password. In order for resetting to happen, user should initiate resetting process from the e-mail that is specifically sent from the system. -> In progress
- **1.1.5.** System *shall* include admin privileged user whose privileges are as follows: -> Not implemented.
    - **1.1.5.1** An admin privileged user *should* be ban or permanently delete users -> Not implemented.
    - **1.1.5.2** An admin privileged user *should* be remove comments, materials and resources. -> Not implemented.
    - **1.1.5.3** An admin privileged user *should* be able to give admin privileges to a normal user while declaring it as another admin. -> Not implemented.
- **1.1.6.** Every user *shall* have a unique and non-alterable username called user identifier which will *shall* be determined at the creation of the account. -> In progress.
    - **1.1.6.1.** System *shall* not allow new accounts to use an user identifier associated with a deleted account, or that is not associated with any account anymore for any reason. -> In progress.

1.2. Self-Development

- **1.2.1.** User *shall* create a resource which includes a description, image and materials. The details of resource is explained in the glossary. -> Completed.
- **1.2.2.** The System *shall* reliably record any progress data supplied by the users grouped by materials and topics. -> Not implemented.
    - **1.2.2.1.** The System *should* display at least one central statistic of this information grouped by materials and topics. -> Not implemented.
    - **1.2.2.2.** The System *should* display progress data associated by a user to the users associated with that user. -> Not implemented.
- **1.2.3.** The System *should* provide the user with a graphical overview of their past effort. There *should* be a page to show this information to the user. -> In progress.
- **1.2.4.** The System *shall* determine and inform its users about topics they could be potentially interested. -> Completed.
    - **1.2.4.1.** The System Interface *shall* provide a page that the users can intentionally navigate to and see this information that may also contain other functionality. -> Completed.
- **1.2.5.** The System *shall* supply its users with recommendation of resources. -> In Progress

    - **1.2.5.1.** The System Interface *shall* provide a page that the users can intentionally navigate to and see this information that may also contain other functionality. -> In Progress

- **1.2.6.** The System *should* offer a means of reliably proving realized effort and any accomplishment of a user to any third party authorized by that user. -> Not implemented

- **1.2.7.** The users *shall* be able to reliably record and edit some objectives they determine as a set of material-deadline pairs. -> In progress

- **1.2.8.** The users *should* be able to record topics they want to improve on, or break/develop a habit. -> In progress.

- **1.2.9.** The users *shall* be able to work on their determined objectives as a group. -> Not implemented.

    - **1.2.9.2.** Users *should* be able to create groups under collective resources that can be only visible to users that are invited to this group. -> Not implemented

1.3. Social Interaction

- **1.3.1.** The users *should* have profile pages, and be able to modify their profiles. -> Completed.

- **1.3.2.** The users *should* be able to share their accomplishments and preferred topics via other applications. -> Not implemented

- **1.3.3.** The users *should* be able to share their comments on specific topics or resources. -> Not implemented

    - **1.3.3.1.** Any comment *shall* display user identifier of the user that posted that comment -> Not implemented

        - **1.3.3.1.1.** If a user account is deleted, user identifier associated with a comment *shall* become "deleted user". -> Not implemented

    - **1.3.3.2.** A comment *should* be able to have a comment list associated with it, which *should* have nested aligning with respect to the owning comment. -> Not implemented

    - **1.3.3.3.** Any user *shall* have the option to delete any comment posted by that user. -> Not implemented

        - **1.3.3.3.1.** If a comment is deleted by the posting user and there exists a nonempty comment list associated with it, then the Interface *shall* display the deleted parent comment as the message "deleted comment". -> Not implemented

    - **1.3.3.4.** A comment *should* have a ballot associated with it which shows rate of the comment. Detailed ballot description is done at the glossary. -> Not implemented

- **1.3.4.** The users *should* be able to search for other users via topics that is associated with them or via username. -> In progress
- **1.3.5.** Users *shall* be able to mark any resource, topic or milestone as "in progress", "following" or "done". -> In progress
  - **1.3.5.1.** The system interface *shall* provide a means of marking and unmarking. -> Completed.
  - **1.3.5.2.** The system interface *shall* display number of users marking each object. -> Not implemented.
  - **1.3.5.3.** System *shall* denote an association with the user and marked object as any of the above -> Completed.
- **1.3.7.** System *should* include two ballots denoting "effort" and "rating" associated with every resource. -> Completed.

1.4. User Experience

- **1.4.1.** The System interface *should* give its users notification messages. -> Not implemented.
  - **1.4.1.1.** If implemented, those notifications *shall* be given for scheduled resource milestones reminding the deadlines. -> Not implemented.
- **1.4.2.** The users *should* be able to customize the way the notifications are sent. -> Not implemented.
- **1.4.3.** The users *should* be able to search data assigned to topics using strings. -> Completed.
- **1.4.4.** The users *should* be able to access other users' profiles. -> Completed.
  - **1.4.4.1.** The System Interface *should* display user's associations with topics, resources and milestones in the profile. -> Completed.

2. Non-Functional Requirements

2.1. Availability

- **2.1.1.** An Android Application and a Web Application *shall* be implemented as instances of the Interface. -> Completed.
  - **2.1.1.1.** Android operating systems of version 10 and above *shall* be supported. -> Completed.
- **2.1.2.** The Interface *shall not* require any user to give text commands for any of its functionalities. It *shall* be possible to give all commands through the interface using a graphical interface. -> Completed.

    - *2.2.2.1.* Those commands *should* be either the standard input types provided by HTML 5 standard or those with equivalent functionality. -> Completed.

- **2.1.3.** The System *shall* express topics using semantic representation as defined in Wikidata. -> Completed.
    - **2.1.3.1.** Topics *shall* be indexed using both the integer semantic ID as indexed in Wikidata in the Infrastructure. -> Completed.
    - **2.1.3.2.** The Interface *should* represent the topics as strings in the preferred language of the operating system of the user and a representation in whatever language is available if it is not available. -> Not implemented.
        - **2.1.3.2.1.** If the representation of the user's language is not found and there are multiple languages for which a representation is available, English *shall* be prioritized. -> Not implemented.

- **2.1.4.** The Web Application *should* only use standard HTML5 and Javacscript code that is compatible with Chromium and Firefox. -> Completed.

2.2. Performance

- **2.2.1.** The Infrastructure *shall* be able to support at least 100 user actions per second. -> In progress.

- **2.2.2.** The Infrastructure *shall* be able to support at least 10000 accounts. -> In progress.

- **2.2.3.** The Infrastructure *shall* have a response time of less than 1.0 second excluding network delay. -> In progress.

- **2.2.4.** The Infrastructure *shall* have an uptime of at least 99%. -> In progress.

2.3. Privacy

- **2.3.1.** The System *shall* be compliant with GDPR. -> Not implemented.
    - **2.3.1.1.** Users *shall* be given explicit and concise explanation of how their data may be used. -> Not implemented.
    - **2.3.1.2.** The interface *shall* give the users a means of approving that they understood how their data may be used at the time of account creation. -> Not implemented.
    - **2.3.1.3.** If any change of how data may be used, all users *shall* be given a notice and prohibited from using the System until they approve it. -> Not implemented.

- **2.3.2.** Users *shall* have authority over visibility by third entities of data related to their accounts. -> Not implemented.
    - **2.3.2.1.** The interface *should* provide a simple page in which the users can select this. -> Not implemented.

2.4. Security

- **2.4.1.** Data that shall be reliably recorded in the Infrastructure *should* be recorded using a means of redundancy, such as using RAID. -> Not implemented.
- **2.4.2.** Application layer protocol used by The System *shall* be restricted to HTTPS. -> Not implemented.
- **2.4.3** Passwords *shall* be hashed before being stored and verified using a NIST-approved cryptographically secure hash function using a salt that depends on the exact user and a system-wide nonce acquired from a reliable entropy source. -> Not implemented.

4. API endpoints

API Endpoint Definitions

URIs to access the APIs is provided in the link above.

Example 1 Example 2 Example 3

5. Implementation of core functionality

Describe the status of this functionality. A learning path is where the users can share their knowledge/experience on topics and make the others' learning/development easier. At the moment, a path has a name, description, effort and rating points, milestones that denote the tasks to complete the learning path and optional tags. Tags are fetched from WikiData and useful to label the paths according to their purpose.

Describe in detail the work that has been completed regarding this functionality. Firstly, the entire team reached a consensus on the term 'path' and we made sure everyone in the team has the same understanding and concept. Then, backend, android and frontend teams came together and decided on API endpoints. While backend was implementing API endpoints for creating a path, enrolling a path, favoriting a path, rating a path, rating effort for a path, finishing a milestone or path, getting path detail and displaying users' previously created paths, frontend and mobile teams were preparing the user interface and other functionalities of the related pages with mock data. After backend finished with endpoints and deployed to test server, these teams integrated with backend, wrote http requests, decoded data and displayed them in real-time. To sum up, our application can now display detailed information about path, also supports the following operations: create a path, enroll a path, favorite a path, rate a path, give effort for a path, finish a milestone or path, getting path detail and displaying users' previously created paths.

Describe the work that remains for this functionality Users will be able to add comments in different groups. This is going to enhance the interaction among users. Comments will be able to voted by users. Users will be able to edit the path. With this feature, paths will remain up to date. Paths will optionally have repeatable tasks. Those tasks will have an in progress state. Milestone containers will support markdown syntax. So that author/editor of the path will freely add different resources and customizations to the milestone.

6. User Interface/User Experience

Home Page

Mobile Code:

https://github.com/bounswe/2021SpringGroup11/blob/master/source/android/Renaissance/lib/home_page.dart
https://github.com/bounswe/2021SpringGroup11/blob/master/source/android/Renaissance/lib/widget/tag_container.dart
https://github.com/bounswe/2021SpringGroup11/blob/master/source/android/Renaissance/lib/widget/course_container.dart
https://github.com/bounswe/2021SpringGroup11/blob/master/source/android/Renaissance/lib/file_converter.dart

Mobile UI:

Web Code:

https://github.com/bounswe/2021SpringGroup11/blob/master/source/frontend/src/components/Home/components/HomeTabPanel.tsx

https://github.com/bounswe/2021SpringGroup11/blob/master/source/frontend/src/components/Home/components/Main.tsx
https://github.com/bounswe/2021SpringGroup11/blob/master/source/frontend/src/components/Home/components/ProfileBar.tsx
https://github.com/bounswe/2021SpringGroup11/blob/master/source/frontend/src/components/Home/actions.ts
https://github.com/bounswe/2021SpringGroup11/blob/master/source/frontend/src/components/Home/constants.ts
https://github.com/bounswe/2021SpringGroup11/blob/master/source/frontend/src/components/Home/index.tsx
https://github.com/bounswe/2021SpringGroup11/blob/master/source/frontend/src/components/Home/reducer.ts
https://github.com/bounswe/2021SpringGroup11/blob/master/source/frontend/src/components/Home/saga.ts
https://github.com/bounswe/2021SpringGroup11/blob/master/source/frontend/src/components/Home/selectors.ts

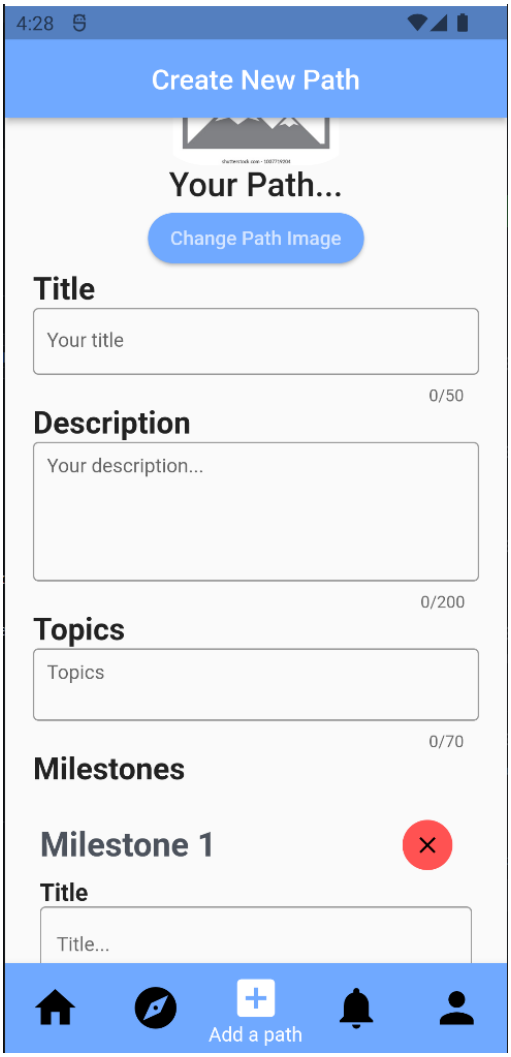
Web UI:

Create Path

Mobile Code:

https://github.com/bounswe/2021SpringGroup11/blob/master/source/android/Renaissance/lib/create_path.dart
https://github.com/bounswe/2021SpringGroup11/blob/master/source/android/Renaissance/lib/file_converter.dart

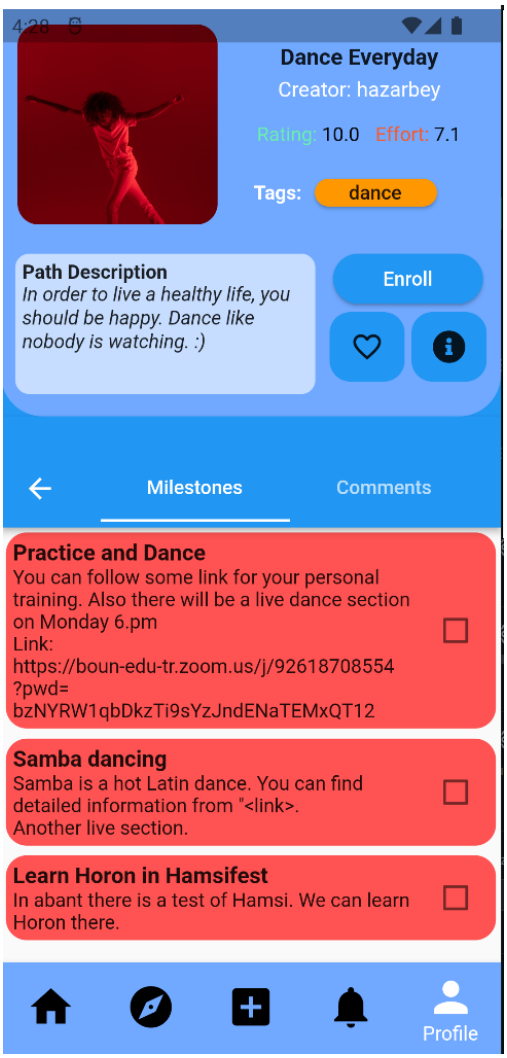
Mobile UI:



Path Page

Mobile Code:
https://github.com/bounswe/2021SpringGroup11/blob/master/source/android/Renaissance/lib/path_page.dart
https://github.com/bounswe/2021SpringGroup11/blob/master/source/android/Renaissance/lib/widget/milestone_widget.dart
https://github.com/bounswe/2021SpringGroup11/blob/master/source/android/Renaissance/lib/widget/cutsom_checkbox_widget.dart
https://github.com/bounswe/2021SpringGroup11/blob/master/source/android/Renaissance/lib/file_converter.dart

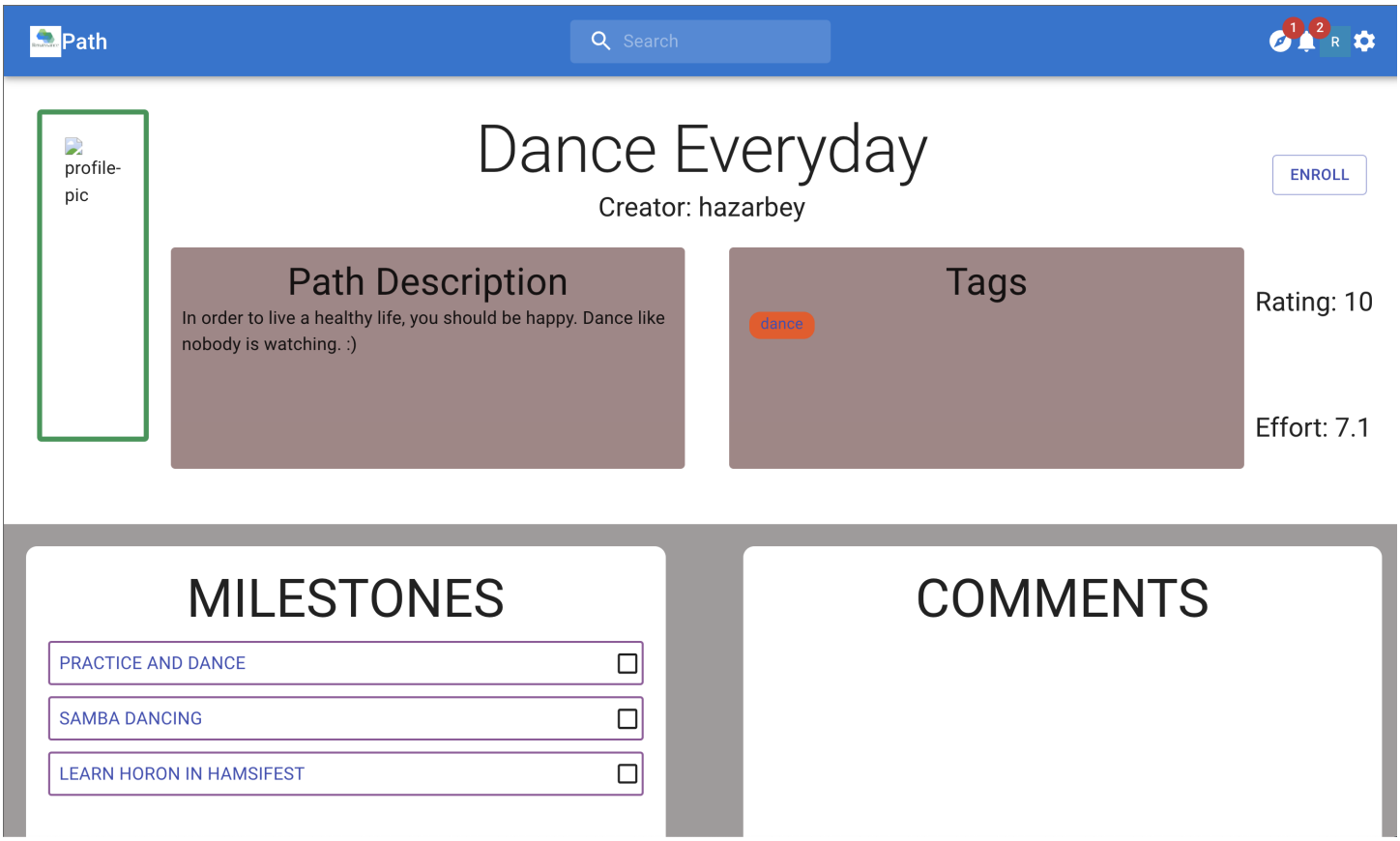
Mobile UI:



Web Code:

https://github.com/bounswe/2021SpringGroup11/blob/master/source/frontend/src/components/Path/components/Milestone.tsx
https://github.com/bounswe/2021SpringGroup11/blob/master/source/frontend/src/components/Path/components/Tag.tsx
https://github.com/bounswe/2021SpringGroup11/blob/master/source/frontend/src/components/Path/actions.ts
https://github.com/bounswe/2021SpringGroup11/blob/master/source/frontend/src/components/Path/constants.ts
https://github.com/bounswe/2021SpringGroup11/blob/master/source/frontend/src/components/Path/index.tsx
https://github.com/bounswe/2021SpringGroup11/blob/master/source/frontend/src/components/Path/reducer.ts
https://github.com/bounswe/2021SpringGroup11/blob/master/source/frontend/src/components/Path/saga.ts
https://github.com/bounswe/2021SpringGroup11/blob/master/source/frontend/src/components/Path/selectors.ts

Web UI:

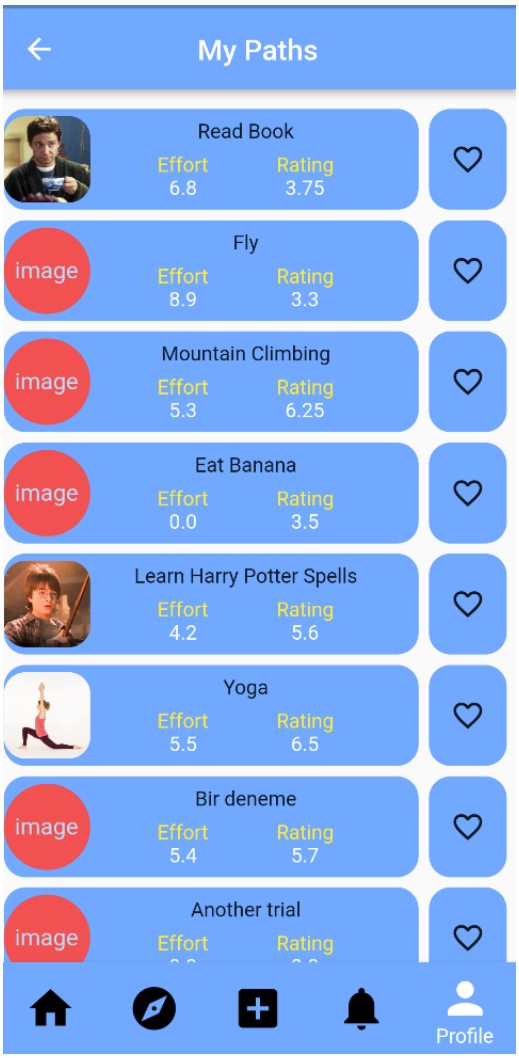


My Paths

Mobile Code:

https://github.com/bounswe/2021SpringGroup11/blob/master/source/android/Renaissance/lib/my_paths_page.dart
https://github.com/bounswe/2021SpringGroup11/blob/master/source/android/Renaissance/lib/widget/course_container.dart
https://github.com/bounswe/2021SpringGroup11/blob/master/source/android/Renaissance/lib/file_converter.dart

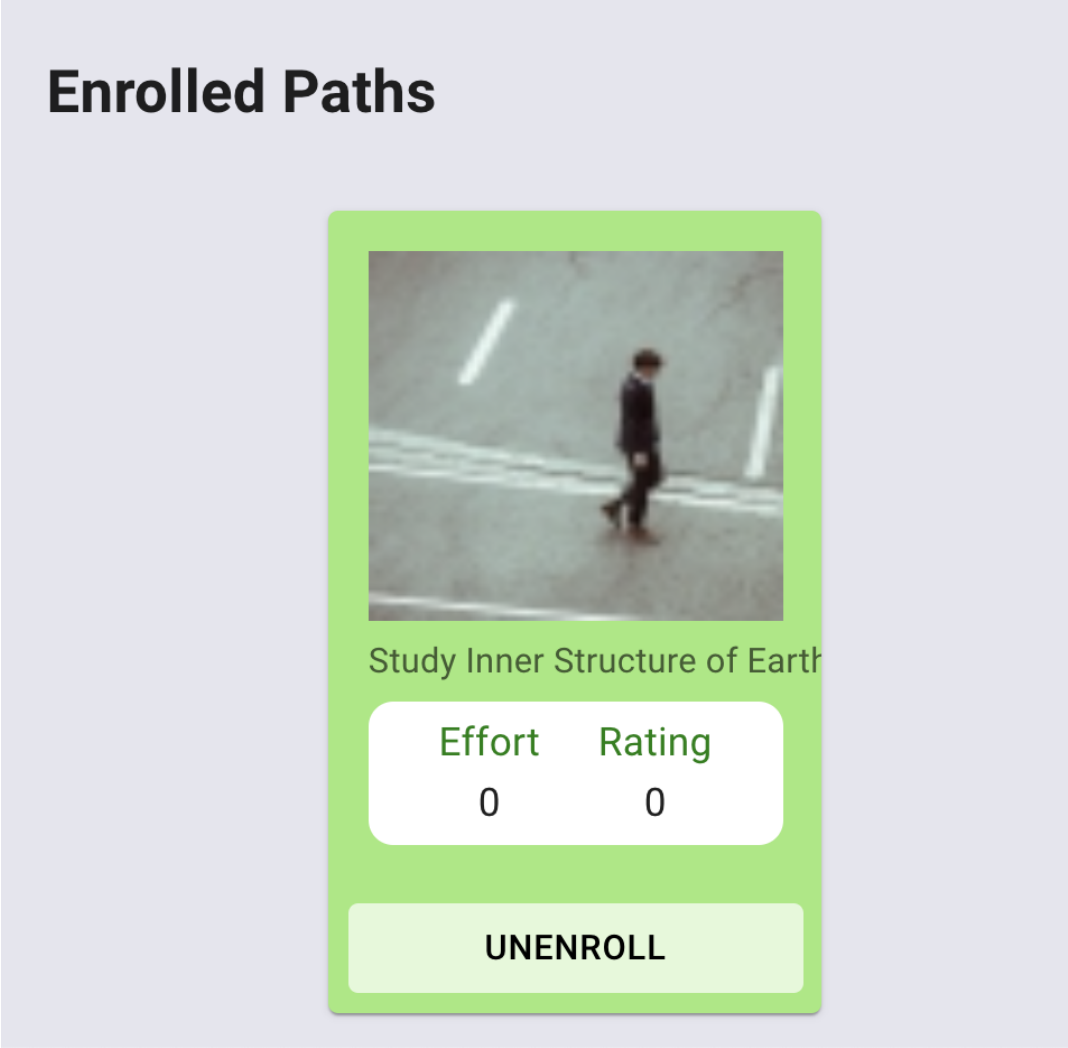
Mobile UI:



Web Code:

https://github.com/bounswe/2021SpringGroup11/blob/master/source/frontend/src/components/Profile/index.tsx
https://github.com/bounswe/2021SpringGroup11/blob/master/source/frontend/src/components/Profile/helper.ts

Web UI:

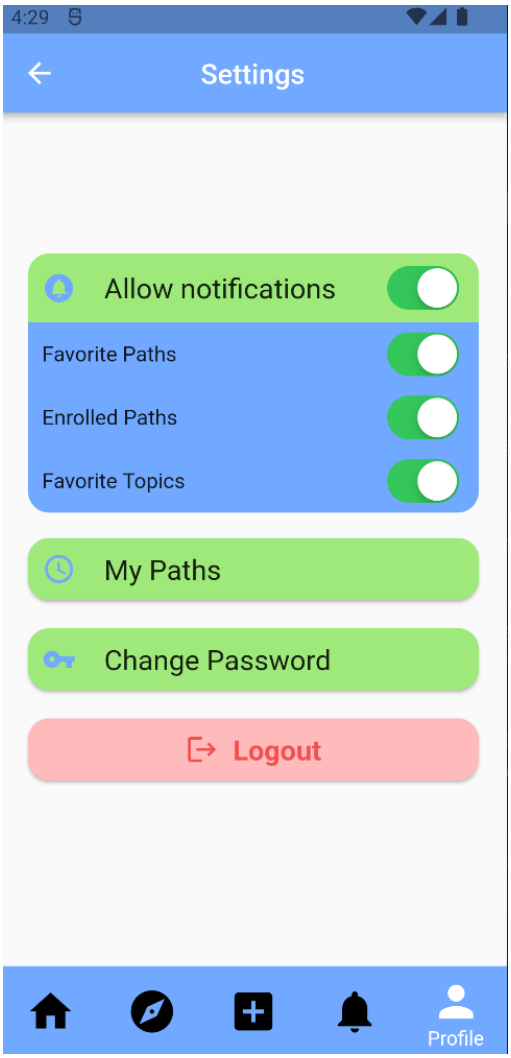


Settings Page

Mobile Code:

https://github.com/bounswe/2021SpringGroup11/blob/master/source/android/Renaissance/lib/settings_page.dart

Mobile UI:

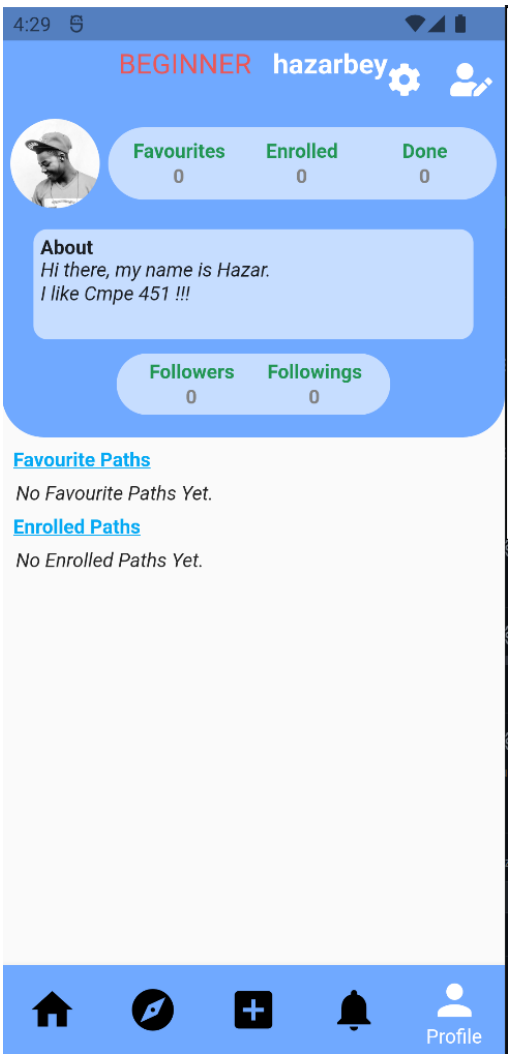


Profile Page

Mobile Code:

https://github.com/bounswe/2021SpringGroup11/blob/master/source/android/Renaissance/lib/profile_page.dart
https://github.com/bounswe/2021SpringGroup11/blob/master/source/android/Renaissance/lib/widget/profile_appbar_widget.dart
https://github.com/bounswe/2021SpringGroup11/blob/master/source/android/Renaissance/lib/widget/profile_follow_widget.dart
https://github.com/bounswe/2021SpringGroup11/blob/master/source/android/Renaissance/lib/widget/profile_stats_widget.dart
https://github.com/bounswe/2021SpringGroup11/blob/master/source/android/Renaissance/lib/file_converter.dart

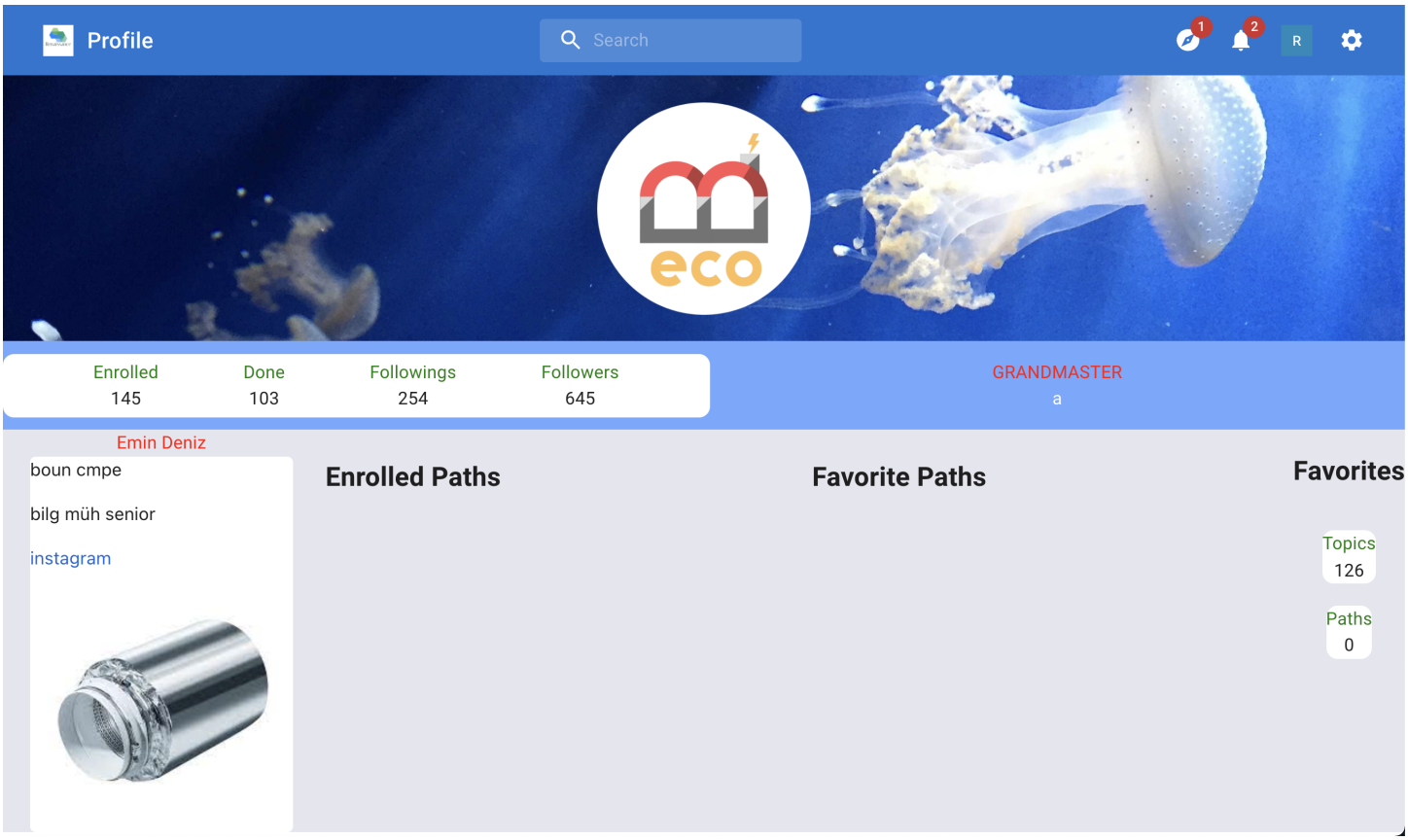
Mobile UI:



Web Code:

https://github.com/bounswe/2021SpringGroup11/blob/master/source/frontend/src/components/Profile/index.tsx
https://github.com/bounswe/2021SpringGroup11/blob/master/source/frontend/src/components/Profile/helper.ts

Web UI:

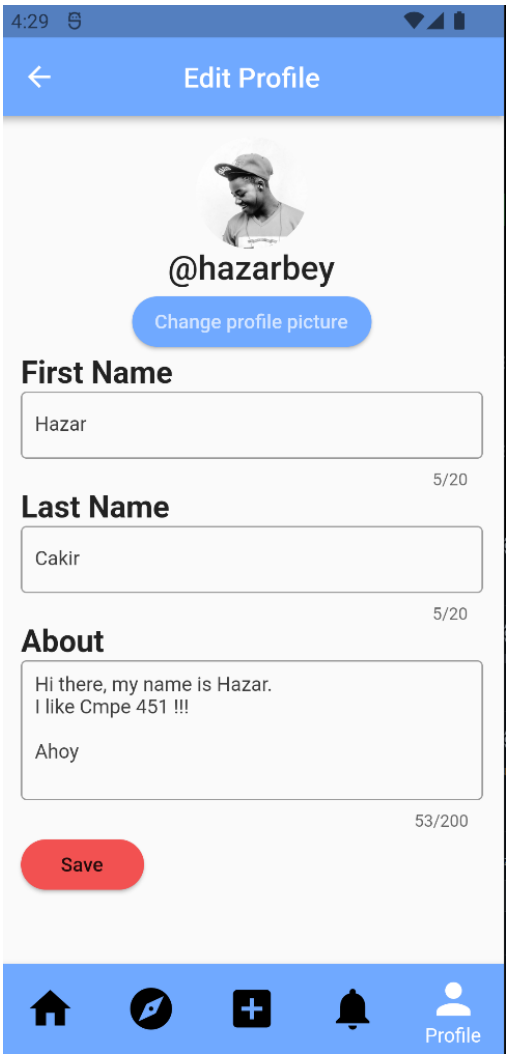


Edit Profile Page

Mobile Code:

https://github.com/bounswe/2021SpringGroup11/blob/master/source/android/Renaissance/lib/edit_profile.dart
https://github.com/bounswe/2021SpringGroup11/blob/master/source/android/Renaissance/lib/file_converter.dart

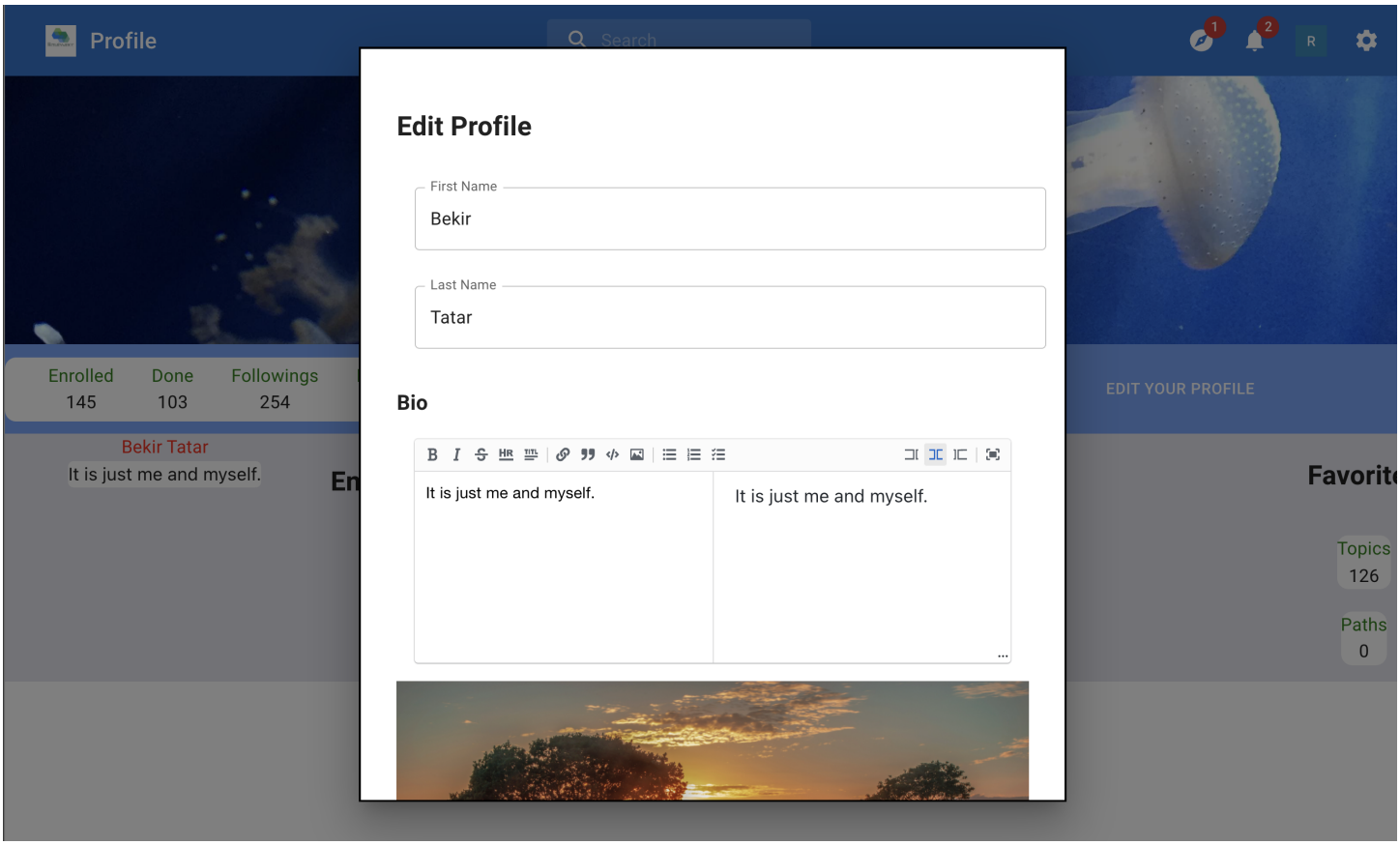
Mobile UI:



Web Code:

https://github.com/bounswe/2021SpringGroup11/blob/master/source/frontend/src/components/Profile/index.tsx
https://github.com/bounswe/2021SpringGroup11/blob/master/source/frontend/src/components/Profile/helper.ts

Web UI:

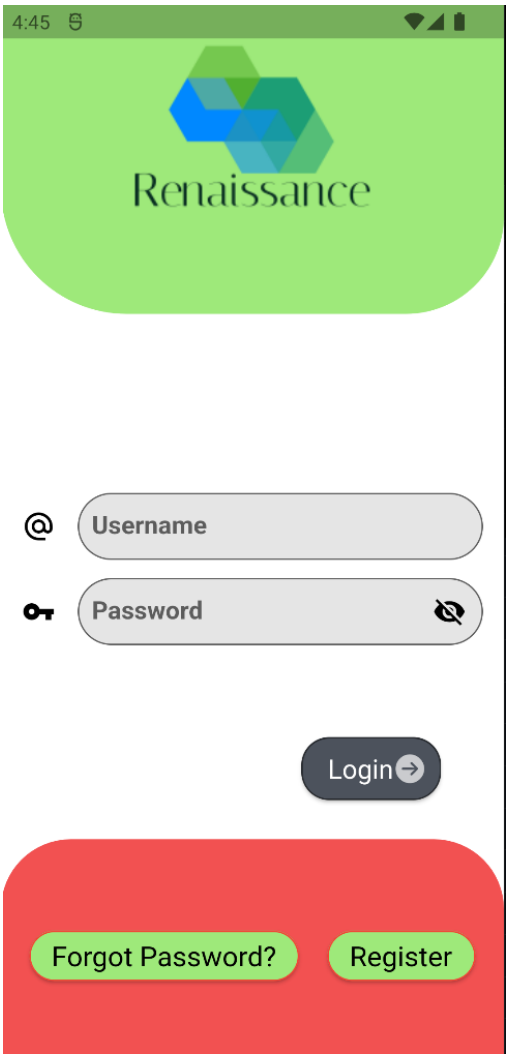


Log In Page

Mobile Code:

https://github.com/bounswe/2021SpringGroup11/blob/master/source/android/Renaissance/lib/login_page.dart

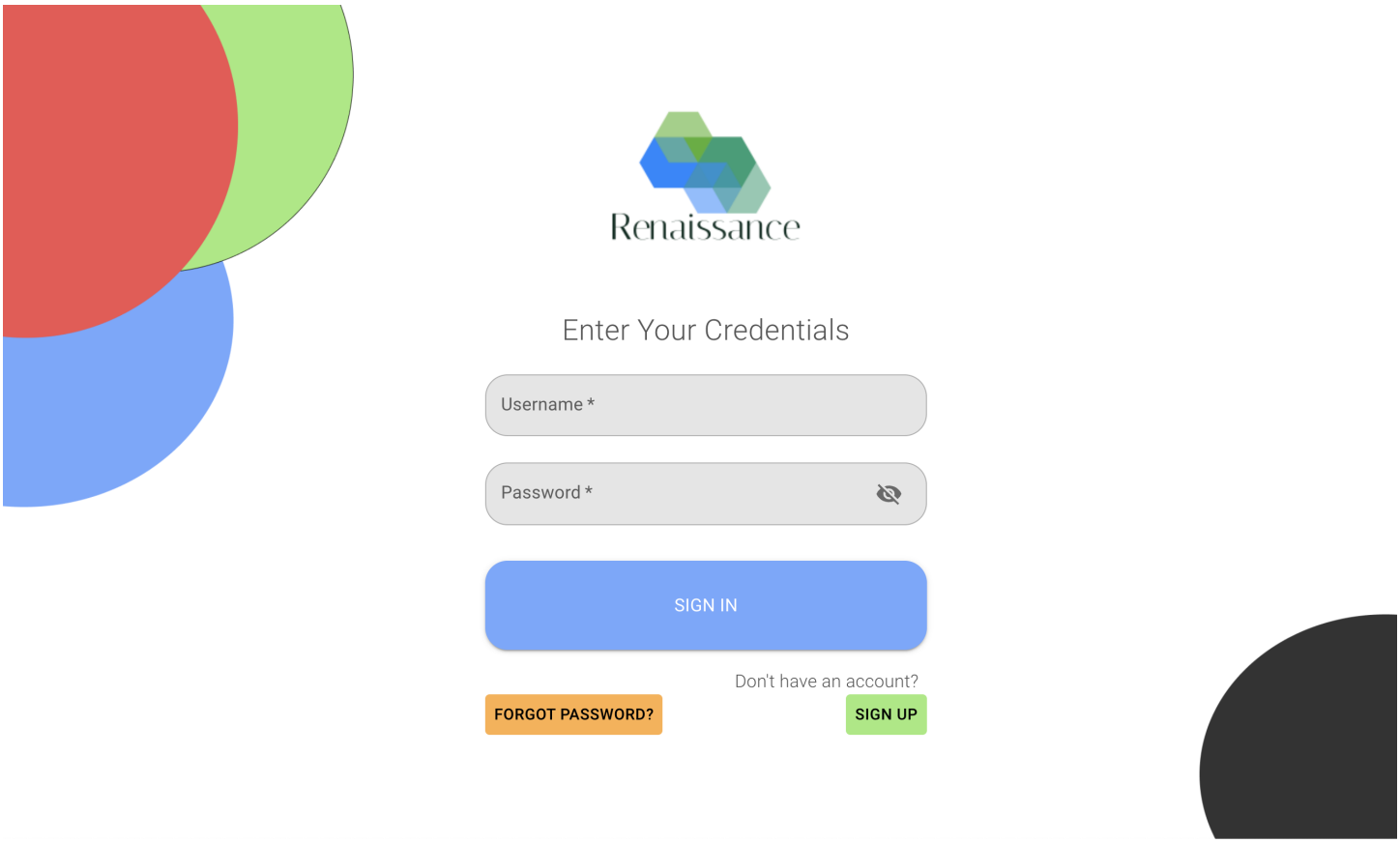
Mobile UI:



Web Code:

https://github.com/bounswe/2021SpringGroup11/blob/master/source/frontend/src/components/Login/actions.ts
https://github.com/bounswe/2021SpringGroup11/blob/master/source/frontend/src/components/Login/constants.ts
https://github.com/bounswe/2021SpringGroup11/blob/master/source/frontend/src/components/Login/index.tsx
https://github.com/bounswe/2021SpringGroup11/blob/master/source/frontend/src/components/Login/reducer.ts
https://github.com/bounswe/2021SpringGroup11/blob/master/source/frontend/src/components/Login/saga.ts
https://github.com/bounswe/2021SpringGroup11/blob/master/source/frontend/src/components/Login/selectors.ts
https://github.com/bounswe/2021SpringGroup11/blob/master/source/frontend/src/components/Login/components/ForgotPassModal.tsx

Web UI:

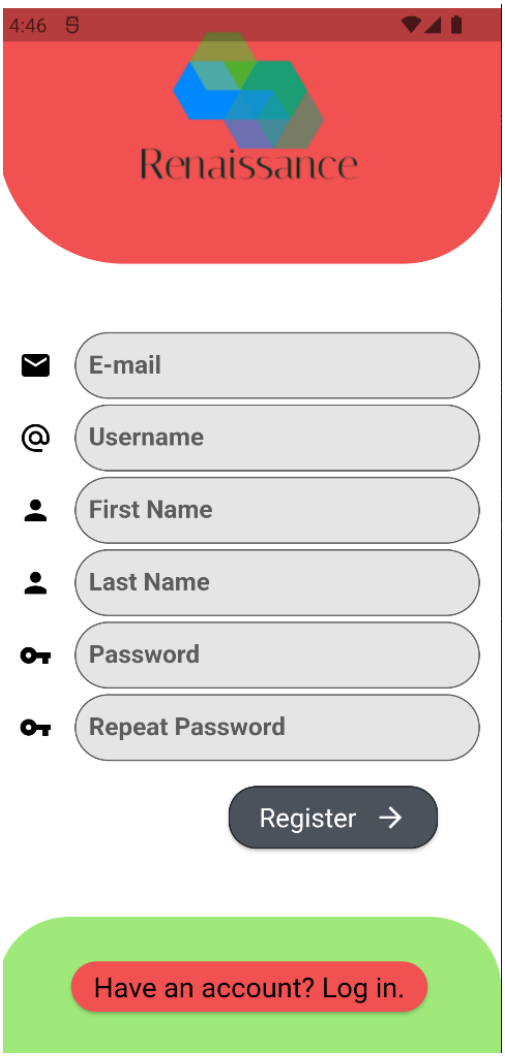


Sign Up Page

Mobile Code:

https://github.com/bounswe/2021SpringGroup11/blob/master/source/android/Renaissance/lib/register_page.dart

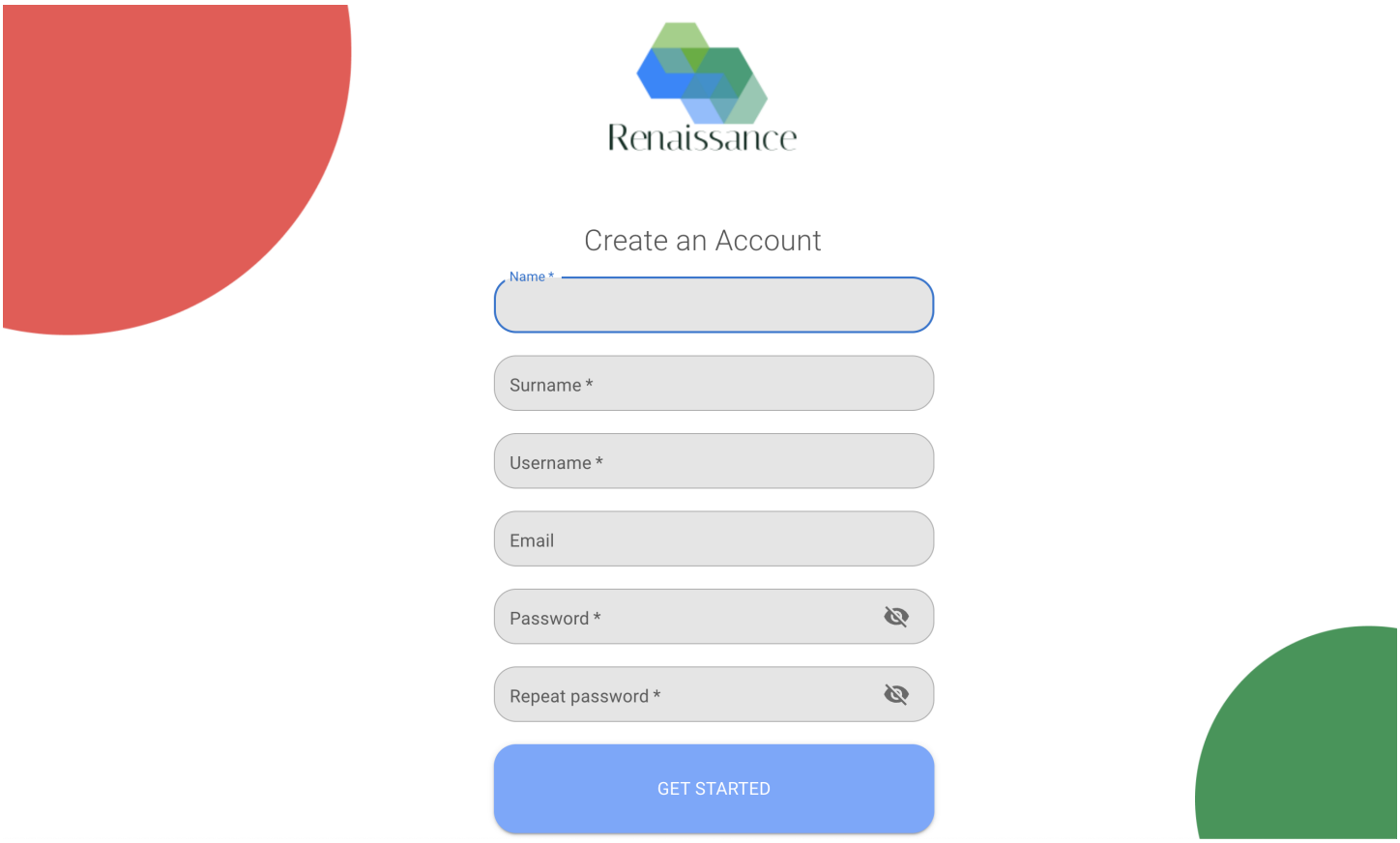
Mobile UI:



Web Code:

https://github.com/bounswe/2021SpringGroup11/blob/master/source/frontend/src/components/SignUp/actions.ts
https://github.com/bounswe/2021SpringGroup11/blob/master/source/frontend/src/components/SignUp/constants.ts
https://github.com/bounswe/2021SpringGroup11/blob/master/source/frontend/src/components/SignUp/index.tsx
https://github.com/bounswe/2021SpringGroup11/blob/master/source/frontend/src/components/SignUp/reducer.ts
https://github.com/bounswe/2021SpringGroup11/blob/master/source/frontend/src/components/SignUp/saga.ts
https://github.com/bounswe/2021SpringGroup11/blob/master/source/frontend/src/components/SignUp/selectors.ts

Web UI:

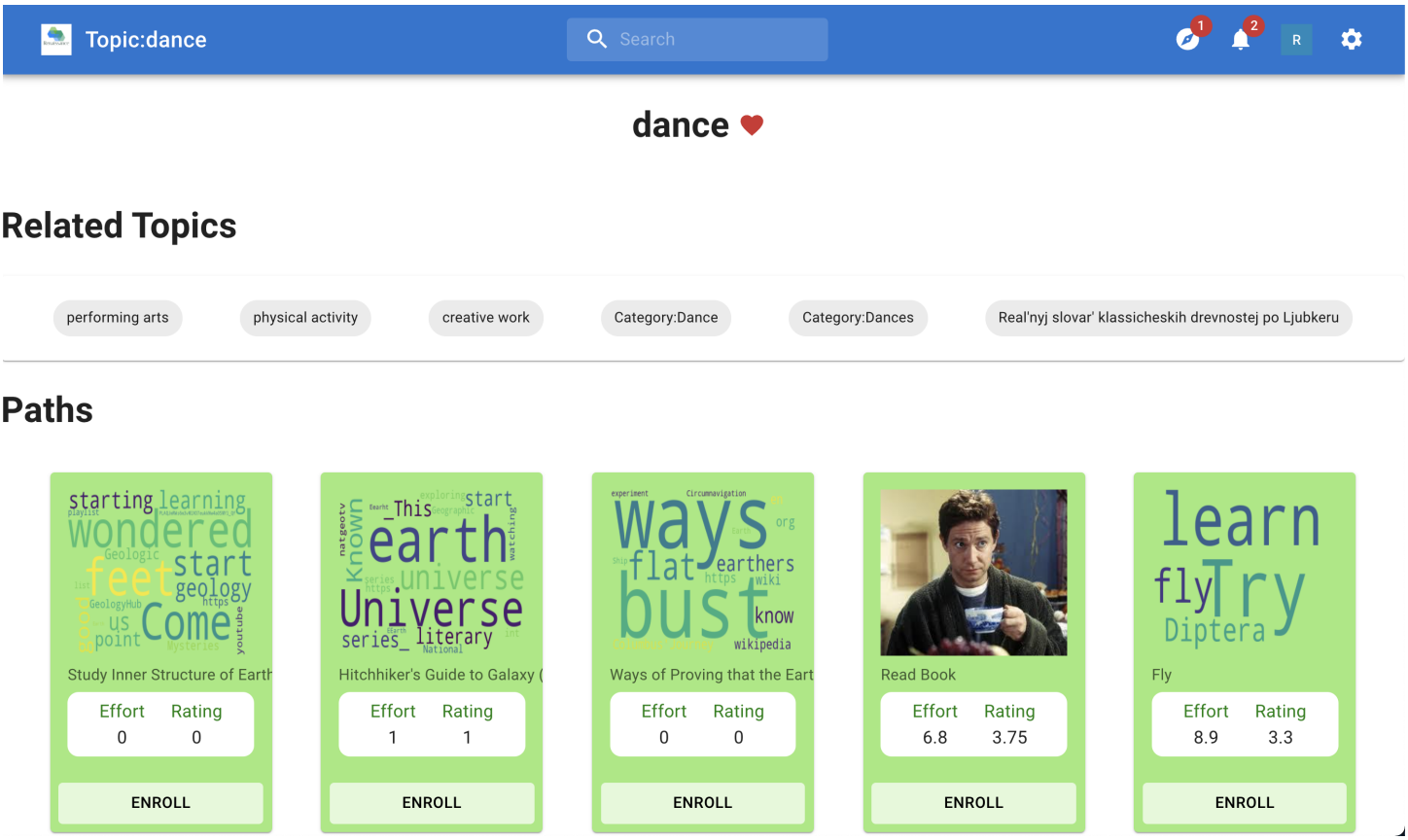


Topic Page

Web Code:

https://github.com/bounswe/2021SpringGroup11/blob/master/source/frontend/src/components/Topic/index.tsx
https://github.com/bounswe/2021SpringGroup11/blob/master/source/frontend/src/components/Topic/helper.ts

Web UI:

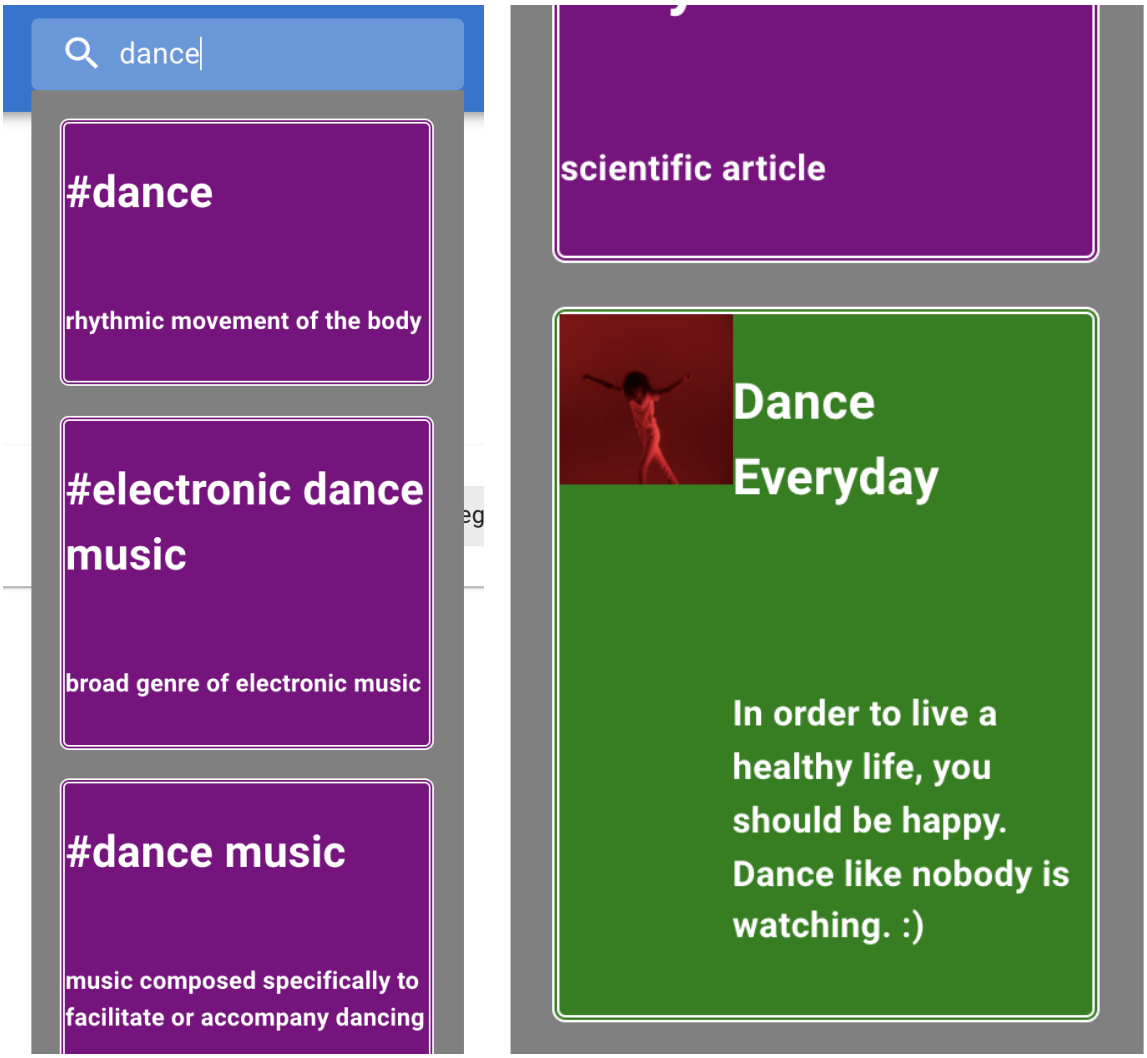


Search

Web Code:

https://github.com/bounswe/2021SpringGroup11/blob/master/source/frontend/src/components/NavBar/SearchBox.tsx
https://github.com/bounswe/2021SpringGroup11/blob/master/source/frontend/src/components/NavBar/search.util.ts

Web UI:



7. Standards(W3C Activity Stream, Wikidata)

Semantic Web

In our application, all the tags, called topics in our requirement glossary, correspond to semantic items in Wikidata. An example is the Q2, meaning PLanet Earth. This allows us to find the abstract relationships between different paths (generalizable to our formal first-class entity "resource" as defined in our requirements glossary) without the need for users to explicitly state or further data mining in use patterns.

This enables users to give a less number of tags to the entities and still get beyond satisfactory similarity reports and precise recommendations. They also come with descriptions and further information about those tags.

Our backend services make extensive use of the APIs provided by Wikidata to recommend related topics, paths and improve search experience. This even includes getting the names of those tags from their servers. Every topic is kept only as an integer representing the Wikidata item ID of that tag. This ability may prove invaluable in internationalization if applied in the future. Sometimes such requests slow down the overall operation of the system. Yet, we are going to keep leaning on Wikidata services to provide the best semantic web experience.

W3C Activity Stream 2.0

We were not aware of the existence of this standard until recently. In the planning session in which we refined our project plan for this milestone, we scheduled this functionality, along with social interaction aspects of the project, for the final milestone.

Some additional functionalities we added over this semester like being able to follow/enroll/favorite all entities of the architecture were intended to pave the way for implementing more complicated activity streams that may enhance both personal use and user interactions. We don't need any further changes in the architecture to make effective use of the activity streams.