1. Milestone Review

1.1. Executive Summary

Our project is a self organized development platform, named Renaissance.
In this platform, one can add a path to let other people enroll/collaborate on their goals. These paths can contain resources, links, milestones, images and have tags to describe what they're about.There are comment threads on every path that people can share their ideas in, ask questions and socialize. These comment threads can be either private or public, and people can be invited to private threads.This is a community-driven platform, so paths and tags can be favorited. As a user starts adding more paths/tags into favorites and enrolls in paths, the system recommends other paths/tags that they may like. The goal is to implement this platform on Web and Android.

1.2. List and status of deliverables

- ** Requirements: ** *Stable*
- ** UML Software Designs: ** *Stable*
- ** Scenarios and Mockups: ** *Stable*
- ** Project plan: ** *Updating*
- ** Communication Plan: ** *Stable*
- ** RAM ** *Stable*
- ** Milestone Review: ** *Stable*

1.3. Evaluation of the status of deliverables and its impact on your project plan

1.3.1. Evaluation of Requirements:

We have finalized our requirements according to the meetings with the customer. We could not predict the future stages of the project, so we needed to update the requirement more than once. Some requirements were changed due to the fact that they are not logical at the implementation phase.
We did not get good feedback for our requirements, so our requirement structure was badly formatted. After getting feedback from a new customer, we fixed all structures and terminology. Thus, We can easily implement it on designs, mockups, diagrams, and development.

1.3.2. Evaluation of UML Software Designs:

Due to structural major changes in requirements, all diagrams are updated.
Sequence diagrams are organized from scratch. Objects are updated according to the class diagram. We made minor changes to the use of the case diagram.
Our UML Diagrams helped us to design our backend and frontend/android user flow. We had done a very good UML Diagram.

1.3.3. Evaluation of Scenarios and Mockups:

Scenario 1 is removed due to changes in requirements. Mockups are organized from scratch. After requirement changes, we needed to change our scenarios and mockups. They are changed with the hard work of the team.

1.3.4. Evaluation of Project plan:

Milestone dates are prepared well. We left enough gap between our milestone and customer milestone, so we can handle unexpected events and we can easily be prepared for presentation and demo. It is well organized. So, we can meet the requirements of the first milestone plan. Our deployment and developments are very good. We need to work hard to do this, for the second milestone we may need to change some plans because it seems a little bit intense.

1.3.5. Evaluation of Communication Plan:

At the beginning, we needed a time to get used to check Slack regularly without using WhatsApp, but we handled it. Later, even though we need some changes on times of our meetings, we never skipped the general concept of assignment-review meetings once a week. It was successful and it is good to know that we don't have any problem regarding communication.

1.3.6. Evaluation of RAM:

Tasks are decided, groups selected their own tasks and review them with partners. We distributed all tasks to subgroups that consist of 1-2 people. So, we could handle most of the tasks easily. It can be thought of as pair programming.

1.3.7. Evaluation of Milestone Review:

Firstly, we discussed the titles together and splitted up them to work more efficiently. At the end, we reviewed all titles and contents altogether. By this way, every team member had idea about the current status of the group.
We saw that we are in a good point according to our project plan, now, and will be better.

1.4. Evaluation of tools and processes you have used to manage your team project.

1.4.1 General Tools and Processes Used By the Team

Github

Github is used as the project version control platform. We uploaded all work done by the team to Github and its wiki page so our work can be tracked by team members with no bother. We created proper issues for each work, assignment or anything related to project that is needs to be tracked or notified by one or more team members. We have tags for issues so they can be easily distinguished according to the their related topics. We are getting mails about the issues that is assigned to us. We are using milestone property of our repository for our project milestones and we are assigning issues to related milestones so we can track our milestones in a more reliable way.

Slack

Slack is used for text messaging and organizing jobs with the team. Its channel option was a reason for preference. We have a channel for different purposes. Each team has its own channel so they can use it for their related issues. We also have a channel to chat and talk about things that are not directly relevant to our project. Each team member has their own notifications settings about those channels so they are not being notified about issues that are not relevant to them.

Zoom

Zoom is used for online meetings. It is provided a wide range of tools, easy to access, annotation and better shared screen quality. We have a fixed zoom link for our meetings so we are not losing time to create and publish a new link for each meeting.

Diagrams.net

Diagrams.net is used for creating mockups. It is free, easy to use, and provided to work simultaneously. The property is that allows us to copy diagrams as image have made things easy for us.

Lucidchart

Lucidchart is used for creating diagrams. It is recommended, free for students, and provided to work simultaneously. It is easy to use and the diagram can be exported with a transparent background.

ProjectLibre

We used project libre for creating and updating the project plan. Although it is free and proper for creating Gantt Charts, it can be really troublesome while preparing the chart. First of all, copy-paste functions of the application do not work properly. You may lose your plan descriptions or plan dates while copy-pasting. Another issue is that dates are extremely difficult to adjust in this app. Sometimes, If you change a date or an assignee to a plan, the dates on the whole project was changing with it. Even though ProjectLibre is a troublesome application for us, we will continue to use it for our project since it is still the best free option for us.

1.4.2 Tools and Processes Used By the Design Team

Figma

Figma is a collaborative graphics editing and user interface design application. Since It is easy to use and provides a large range of the functionalities such as creating CSS properties for components, we choose Figma to design our web and mobile applications. It also has a real-time collaboration system that helped us to design the pages together with the team.

1.4.3 Tools and Processes Used By the Frontend Team

ReactJS

ReactJS is an open-source Javascript library that is used for building web applications specifically for single-page applications. Since our application is designed to have mostly interactions with users and update itself according to them, we needed a dynamic single-page application rather than a static website. Since it is simple, easy to learn, and has support for useful 3rd party libraries such as Material UI and Redux, we choose ReactJS over other frameworks.

Typescript

Typescript is a superset language of javascript. Since Typescript provides optional static types and it is easier to spot errors in typescript, we chose Typescript over javascript as a langauge.

Parcel, Eslint, Prettier

Since our project has restrictions about using the Create-react-app template, we need to create our filesystem and development tools from scratch. Create-react-app and other templates use mostly Webpack for a bundler. Although Webpack is more stable and reliable than Parcel, Parcel is faster and easy to configure from scratch. So we chose Parcel bundler over Webpack for our project. We have encountered some problems while developing the project with Parcel. One of them was that sometimes Parcel was giving errors while starting or building our app. ESLint is a tool for identifying and reporting on patterns found in ECMAScript code. We used it to have good-looking coding conventions and to detect possible bugs. Prettier is also a linter and it helped us to format our code in a better shape.

Redux and Redux-Saga

React apps are mostly component-based apps. Without a state container and management system, each of these components has its own states and has to manage them on its own. This property makes it hard to communicate between different components. Redux is a predictable state container for JS apps. Redux stores all the necessary states in a single store and makes the necessary changes according to the component actions. With the help of the redux, components can communicate each other much more easily. In the beginning, It was a little hard to implement the redux architecture to our project for the frontend team, But later on, We realized that this will help us a lot when our project becomes bigger and hard to manage.

Redux-Saga is an intuitive Redux side effect manager. We used Redux-saga for our asynchronous actions such as fetching or posting data to the backend server. With help of redux and redux-saga, We managed all of our asynchronous actions outside of our components and did not affect the component's current state until it is changed by the user or the result of the asynchronous actions.

Material UI

Material UI is a UI library that provides customizable, accessible, and robust components. We used MUI's responsive functionalities and components in some of our pages to have a consistent looking in different screen sizes. We imported some components from MUI and customized them to look just like our designs. This helps us to save time while implementing the UI part of our application.

Github Actions and AWS Amplify, AWS S3

AWS Amplify is a service that can be used for the deployment of React applications from GitHub repositories. Since it is easier to configure and does not require creating a bucket in

AWS, Firstly we tried to use AWS Amplify for deployment. Amplify gave us some errors about Parcel while building our application in its server. Then we decided to use Github Actions and AWS S3 for deployment. With the help of the AWS S3 and Github Actions, We have more consistent and trackable deployment. Our frontend application uses a CI/CD method now with Github actions and AWS S3 and this method help us a lot while developing and testing our application.

1.4.4 Tools and Processes Used By the Android Team

Flutter

Flutter is an open-source UI software development kit created by Google. It is used to develop cross platform applications
for Android, iOS, Linux, Mac, Windows, Google Fuchsia, and the web from a single codebase. However in the application, we only use flutter for Android Development.
It is decided becuase it has a modern approach for android development. The development has no drag&drop interface for design but code based design mechanisms like bootstrap.

1.4.5 Tools and Processes Used By the Backend Team

Django

After researching and discussing on various frameworks, we decided that Django could be good for our use cases. Django is a free and open source framework, which has big impact on our decision. Besides, most of us had experience on Django structure, which is another significant effect promoting us to choose it.

MongoDB

We decided to use a NoSQL database for this project. MongoDB was the most approriate choice satisfying our requirements. In addition to previous experiences, we chose it since it does not require an extra image in dockerization process. Our database is hosted separately from our server and therefore different implementations of the backend being developed could simultaneously use the same testing data.

Postman

We did most of our tests using the very convenient Postman software. Each of us wrote our own Postman collections for all endpoints and tested our local implementations as well as the deployed server.

Docker

As the de facto standard for containerization, we used Docker for creating containers for our backend service and deploying it. It allowed us to test the system in different machines with the same behavior. It also gave us confidence that the once we solve a problem locally,

venv

We used Python virtual environments to make sure every implementation use the same versions of the same python libraries in every machine, this is an additional level of virtualization below Docker and let us have confidence on the universal applicability of the code even without containerization. Because of the additional convenience of this system, It was a better alternative to Docker for testing minor updates on local machines.

Google Cloud

After having a minor issue with Amazon AWS, we chose to deploy our containerized application to Google Cloud. We found that it has a more convenient interface.

1.5. The requirements addressed in this milestone

For Milestone 1, We've mainly implemented User related pages which are Register, Login, Home and Profile Pages. This funconalities' related requirements and their explanation can be found below.
For clarification on the terminology, gloassary in the requirements should be checked. (Especially resource)

- **1.1.1.** User *shall* be able to register by providing a valid and verified e-mail address, username, and a strong password.
  Email verification and strong password not completed yet, but will be implemented on further.

- **1.1.2.** User *shall* be able to login by providing verified e-mail address.
  *Users are able to login by providing **usernames**. Also, verification part is not complete yet.*

- **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.
  - **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 .
    These requirements successfully implemented on both platforms.

- **1.2.4.** The System *shall* determine and inform its users about topics they could be potentially interested.
    - **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.
      Because topics are not part of this milestone, these requirements are implemented with mock data but not compeleted yet (Home Page).

- **1.2.5.** The System *shall* supply its users with recommendation of resources.
    - **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.
      Because resources are not part of this milestone, these requirements are implemented with mock data but not compeleted yet (Home Page).

- **1.3.1.** The users *should* have profile pages, and be able to modify their profiles.
  Users have profile pages, however editing profile page not part of this milestone, thus not completed yet

- **1.3.4.** The users *should* be able to search for other users via topics that is associated with them or via username.
  Users are able to search other users via username on Web Application, however topics are not part of this milestone.

- **1.3.5.** Users *shall* be able to mark any resource, topic or milestone as "in progress", "following" or "done".
    - **1.3.5.1.** The system interface *shall* provide a means of marking and unmarking.
    - **1.3.5.2.** The system interface *shall* display number of users marking each object.
    - **1.3.5.3.** System *shall* denote an association with the user and marked object as any of the above
      The objects other than user is not part of this milestone, so only 1.3.5's interface part implemented functionality is not implemented yet.

- **1.3.7.** System *should* include two ballots denoting "effort" and "rating" associated with every resource.
  Because resources are not part of this milestone, these requirements are implemented with mock data but not compeleted yet.

- **1.4.2.** The users *should* be able to customize the way the notifications are sent.
  Again interface implemented, however functionality not implemented.

- **1.4.4.** The users *should* be able to access other users' profiles.

    - **1.4.4.1.** The System Interface *should* display user's associations with topics, resources and milestones in the profile.
      These requirements partially implemented on Web Application. (with search user by username)

- **2.1.1.** An Android Application and a Web Application *shall* be implemented as instances of the Interface.
    - **2.1.1.1.** Android operating systems of version 10 and above *shall* be supported.
      These requirements successfully implemented on both platforms.

- **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.
    - **2.1.2.1.** Those commands *should* be either the standard input types provided by HTML 5 standard or those with equivalent functionality.
      These requirements successfully implemented on both platforms.

- **2.1.4.** The Web Application *should* only use standard HTML5 and Javacscript code that is compatible with Chromium and Firefox .
  This requirement successfully 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.
  This requirement successfully implemented.

1.6. A summary of work performed by each team member (in tabular format)

Person	Summary
Fikri Cem Yılmaz:	In first part of redesigns I've reviewed and redesigned Mockups with Meltem and Kürşat. After all redesigns completed, we've reviewed other designs. After project's details are clarified, second part of designs started and We've reviewed and redesigned Use Case diagram with Emin. Then, implementation part started I've mainly implemented profile page on Android. Then, helped some changes in home page and a minor change in register page and navigation between pages. Finally, reviewed my team mates codes and accepted their pull requests.

Person	Summary
Eren Altunoğlu:	I am a member of the backend team. Revised Project Plan&RAM. Reviewing updates for requirements, use cases and sequence diagrams. Revised Class Diagram twice. First one was before customer meeting. The second was done after customer meeting and updating sequence diagram. Uploaded meetings notes for one backend and one group meeting. Created tasks for backend features. Implemented login, signup, change password, user model, middlewares and get profile detail endpoints. Tested endpoints. Uploaded postman screen shots to endpoint issues. Yasin and I pair programmed some related tasks such as change password and forgot password issues. Solving bugs for CORS error and search user endpoint. Took part in dockerizing backend. Took part in documenting API endpoints. Deployed backend to ubuntu virtual machine on Google Cloud. Took notes from customer presentation.

Person	Summary
Bekir Tatar:	Beginning of the semester, we reviewed our requirements and decided on the main features of our project and application. According to these features and reviews from team members, Eren and I updated the implementation parts of our project plan using ProjectLibre. Deleted the unnecessary branches for our practice app in our repository. Reviewed the sequence diagrams and updated SignUp, Follow/UnFollow, Upvote /Downvote, and search sequence diagrams according to the feedback from the customer. Made research about frontend frameworks, tools, and technologies and made a decision on which of these technologies should and will be used for our web application with the frontend team. Designed signup and home pages of our application using Figma. After creating our project template, Introduced Redux and Redux-Saga to our application and configure our filesystem according to the Redux. Configured ESlint and prettier settings for our application. Initialized and configured routes for sign-in, sign-up, and home pages. Implemented the sign-in and home pages. Reviewed and approved signup, navigation bar, and profile page design and implementations. Created a bucket in AWS S3 for our deployment. Attended all of the team, design, customer and frontend meetings. Reviewed some of the android and backend implementations.
Mehmet Yasin Şeremet:	Updated the group wiki and sidebar to prepare them for new semester and course by separating and archiving the old materials. Updated the communication plan. Updated and uploaded the class diagram twice with backend team to make it appropriate according to feedbacks. Researched about potential frameworks we could use. Contributed the creation of the first API. Implemented ban user endpoint. Updated refresh JWT token endpoint. Implemented authentication middleware, change password and forgot password endpoints with Eren via pair programming. Tested all endpoints. Dockerized and deployed backend API with backend team. Wrote down API endpoint definitions. Besides, attended all team, backend and customer meetings.

Person	Summary
Meltem Arslan:	The materials of previous term were reviewed. Mockups were updated two times and uploaded to Github. Wiki page was updated with the names of newcomers and my personal wiki page. Group photo was added to our Wiki main page and meeting template. Uploaded meeting notes : 10.3, 10.frontend.2, 14.2 Customer questions were prepared. Web profile and web Login were designed using Figma. The last 4 sequence diagrams were updated and uploaded to the Github. The sign-up page was implemented. The Renaissance was deployed to AWS S3 container. Prepared to the presentation customer meeting (including scenario creation).
Selman Berk Özkurt:	I am a member of the Backend Team. Introduced the architectural details to the new members of the group. Added many further details to the requirements with Hazar. Introduced requirements revision system and updated most pages to have references to requirements revisions specified. Devised architectural changes in accordance with customer meetings and held another meeting to explain them to the group. Implemented several nonfunctional features in most endpoints ensuring proper access control and preventing breaches. Updated the class diagram with the Backend Team. Updated User model to accomodate many functionalities. Implemented user deletion-related functionalities and improved editing functionality. Improved user search functionality. Addressed the photograph handling in the backend and implemented the solution. I would like to note that Eren deserves special attribution for our success in the backend thanks to his expertise.

Person	Summary
Muhammed Emin Deniz:	I invited new group members to slack group. We as a whole group overviewed the whole requirements with the customer. After that, we needed to update diagrams. I and Fikri Cem updated use case diagrams according to new requirements. I reviewed the mockup changes. I am a member of the frontend team, I was responsible for the navigation bar, user search, and profile pages. I took a meeting note at a meeting. I had done code review for my friends and also suggested solutions for their issues like AWS s3 deployment for react-router. I attended all team, frontend, and customer meetings.
Kürşat Bakiş:	As he wasn't in the group during CMPE352, had an adaptation period. Read the wiki, checked the past issues and learnt about the project. Got to know the team members. Updated the mockups taking the new functionality into account. Fixed the small terminology mistakes at requirements with Hazar. Designed login, register, user profile, settings pages of mobile app. Made decisions about what language and frameworks to use. Pushed the Android team forward. Implemented login, register, settings, tab bar, http services, app colors, forgot password. Tested all features of the application. Made suggestions about fixing the improper UI designs. Changed path/tag following terminology to favorite paths/tags. Wrote the http header and bodies of backend requests. Attended all meetings. Reviewed and commented on Fikri and Hazar's pull requests.

Person	Summary
Hazar Çakır:	Because I am one of the new members of the group, I meet and get to know other group members and learn about the existing project in first weeks. While reviewing last years project, I updated the requirements with Selman and add a lot of details. Then I became a member of the mobile development subgroup. As part of the reviews, updated class diagrams with the backend team. Additionally, after customer review, requirements are updated once more by me and Kürşat. I joined to the design group and designed mobile main-page and contribute other mobile design choices. For the implementation, I code the mobile main page of the application with flutter and also fixed some bugs in the login-register pages. Feedback system is revised and improved. At the presentation, I and Fikri prepared and present the mobile part of the application. I attended nearly all meetings and reviewed most of the jobs from other members.

2. Deliverables

Requirements

Revision 1.4 - Status: Draft

This revision is made as the fulfillment of the 31.10.2021 meeting feedback notes.

Notice: Keywords “should”, “shall”, “should not”, “shall not” are used in accordance with [RFC 2119](#).

Questions:
** Argue admin privileges.
** 1.2.5. Hazar thinks this is meaningless

Glossary

- **Admin:** Some users who have the privilege of banning other users.
- **Android Application:** Software developed using Android Studio that can be distributed through Play Store. This is an instance of the interfaces in The System.
- **Ballot:** An object that displays the number of votes with each of positive and negative attributes and exposes a way for any user to add a new vote with positive or negative attributes.
- **Challenge-Response Authentication:** A service internal or external to The System that proves the entity causing requests of interacting with The System is indeed a human.
- **Collective Resource:** Similar to Resource with the additional set called roles, also with additional functionality of creating teams of one User associated with a role and providing Resource functionalities for each of the roles in the team.
- **Dashboard:** A single page in the Interface customized for each user that has a collection of links.
- **Effort:** Data representing a user's accomplishments specific to materials associated with some topics.
- **Infrastructure:** The functional component of The System that is responsible for handling, processing, recording synchronized data and making any programmed operations affecting overall functionality of the system. This is an application stack to be run on one or more computers connected to the global IP network.
- **Interface:** The functional component of The System that deals with representation of the logic and supplying a means of bidirectionally communicating with the infrastructure.
- **Material:** Some documents, applications or web-links associated with some topics users may benefit by studying.
- **Milestone:** An atomic objective in a path resource typically associated with a material.
- **Path Resource:** Similar to resource with the additional set called milestones, providing mark functionalities of the Resource for each of milestones separately.
- **Resource:** Container for material providing functionality to follow, mark as in progress, mark as finished and comment threads.
- **Role:** An atomic objective in a collective resource typically associated with a material.

- **Scheduled Resource:** Similar to path resource with each milestone associated with a period measured since the initial mark-in-progress event of the Scheduled resource.
- **The System:** The project in its all functional components including interface and back end.
- **Topic:** Abstract structure used for labeling most data flowing and being recorded in The System
- **User:** Entity accessing The System through its interfaces and the intended utilizer of overall functionality of The System. Users are associated with and are represented as *accounts* in The System
- **User Identifier:** A ISO/IEC 10646:2009 string that is associated with a user and is unique to that user.
- **Verified Contact:** Means of communication assumed to be accessible only by the User in real life which was proven by the user during signing up to be accessible by it.
- **Web Application:** Instance of the interface that can be run by a generic web browser. This shall be distributed on-demand through IP network using a DNS domain.

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.
- **1.1.2.** User *shall* be able to login by providing verified e-mail address.
- **1.1.3.** The user *shall* be able to delete their accounts.
- **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” .
    - **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.

- **1.1.5.** System *shall* include admin privileged user whose privileges are as follows:
    - **1.1.5.1** An admin privileged user *should* be ban or permanently delete users
    - **1.1.5.2** An admin privileged user *should* be remove comments, materials and resources.
    - **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.
- **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.
    - **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 .

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.
- **1.2.2.** The System *shall* reliably record any progress data supplied by the users grouped by materials and topics.
    - **1.2.2.1.** The System *should* display at least one central statistic of this information grouped by materials and topics.
    - **1.2.2.2.** The System *should* display progress data associated by a user to the users associated with that user.
- **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.
- **1.2.4.** The System *shall* determine and inform its users about topics they could be potentially interested.
    - **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.
- **1.2.5.** The System *shall* supply its users with recommendation of resources.

    - **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.

- **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.

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

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

- **1.2.9.** The users *shall* be able to work on their determined objectives as a group.

    - **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.

1.3. Social Interaction

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

- **1.3.2.** The users *should* be able to share their accomplishments and preferred topics via other applications.

- **1.3.3.** The users *should* be able to share their comments on specific topics or resources.

    - **1.3.3.1.** Any comment *shall* display user identifier of the user that posted that comment

        - **1.3.3.1.1.** If a user account is deleted, user identifier associated with a comment *shall* become “deleted user”.

    - **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.

    - **1.3.3.3.** Any user *shall* have the option to delete any comment posted by that user.

        - **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”.

    - **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.

- **1.3.4.** The users *should* be able to search for other users via topics that is associated with them or via username.

- **1.3.5.** Users *shall* be able to mark any resource, topic or milestone as “in progress”, “following” or “done”.

    - **1.3.5.1.** The system interface *shall* provide a means of marking and unmarking.
    - **1.3.5.2.** The system interface *shall* display number of users marking each object.
    - **1.3.5.3.** System *shall* denote an association with the user and marked object as any of the above

- **1.3.7.** System *should* include two ballots denoting “effort” and “rating” associated with every resource.

1.4. User Experience

- **1.4.1.** The System interface *should* give its users notification messages.

    - **1.4.1.1.** If implemented, those notifications *shall* be given for scheduled resource milestones reminding the deadlines.

- **1.4.2.** The users *should* be able to customize the way the notifications are sent.

- **1.4.3.** The users *should* be able to search data assigned to topics using strings.

- **1.4.4.** The users *should* be able to access other users’ profiles.

    - **1.4.4.1.** The System Interface *should* display user’s associations with topics, resources and milestones in the profile.

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.
    - **2.1.1.1.** Android operating systems of version 10 and above *shall* be supported.
- **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.
    - *2.2.2.1.* Those commands *should* be either the standard input types provided by HTML 5 standard or those with equivalent functionality.
- **2.1.3.** The System *shall* express topics using semantic representation as defined in Wikidata.
    - **2.1.3.1.** Topics *shall* be indexed using both the integer semantic ID as indexed in Wikidata in the Infrastructure.
    - **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.
        - **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.
- **2.1.4.** The Web Application *should* only use standard HTML5 and Javacscript code that is compatible with Chromium and Firefox .

2.2. Performance

- **2.2.1.** The Infrastructure *shall* be able to support at least 100 user actions per second.
- **2.2.2.** The Infrastructure *shall* be able to support at least 10000 accounts.
- **2.2.3.** The Infrastructure *shall* have a response time of less than 1.0 second excluding network delay.
- **2.2.4.** The Infrastructure *shall* have an uptime of at least 99%.

2.3. Privacy

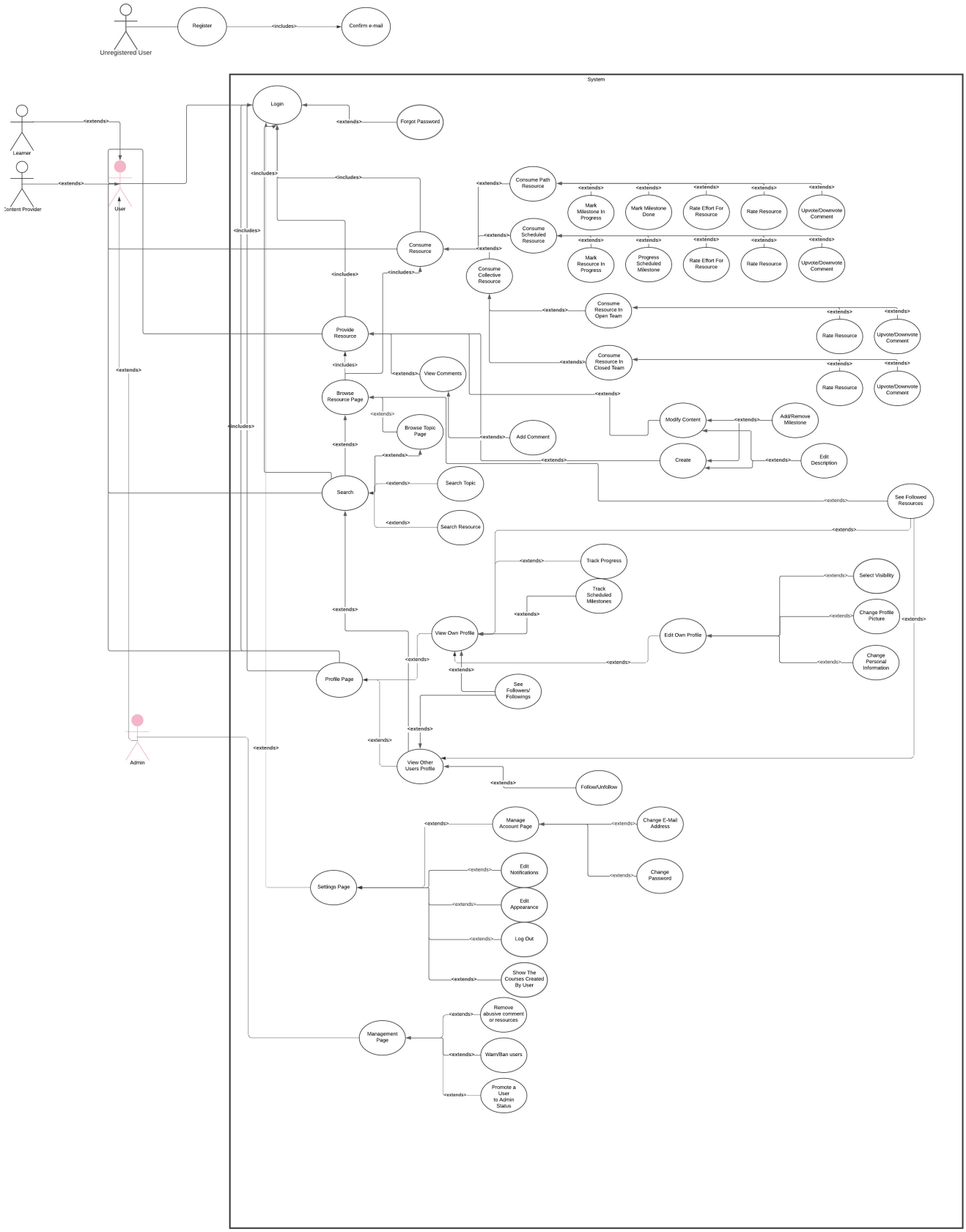
- **2.3.1.** The System *shall* be compliant with GDPR.
    - **2.3.1.1.** Users *shall* be given explicit and concise explanation of how their data may be used.
    - **2.3.1.2.** The interface *shall* give the users a means of approving that they understood how their data may be used at the time of account creation.
    - **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.
- **2.3.2.** Users *shall* have authority over visibility by third entities of data related to their accounts.
    - **2.3.2.1.** The interface *should* provide a simple page in which the users can select this.

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.
- **2.4.2.** Application layer protocol used by The System *shall* be restricted to HTTPS.
- **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.

UML Software Designs

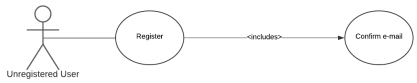
Use Case Diagram

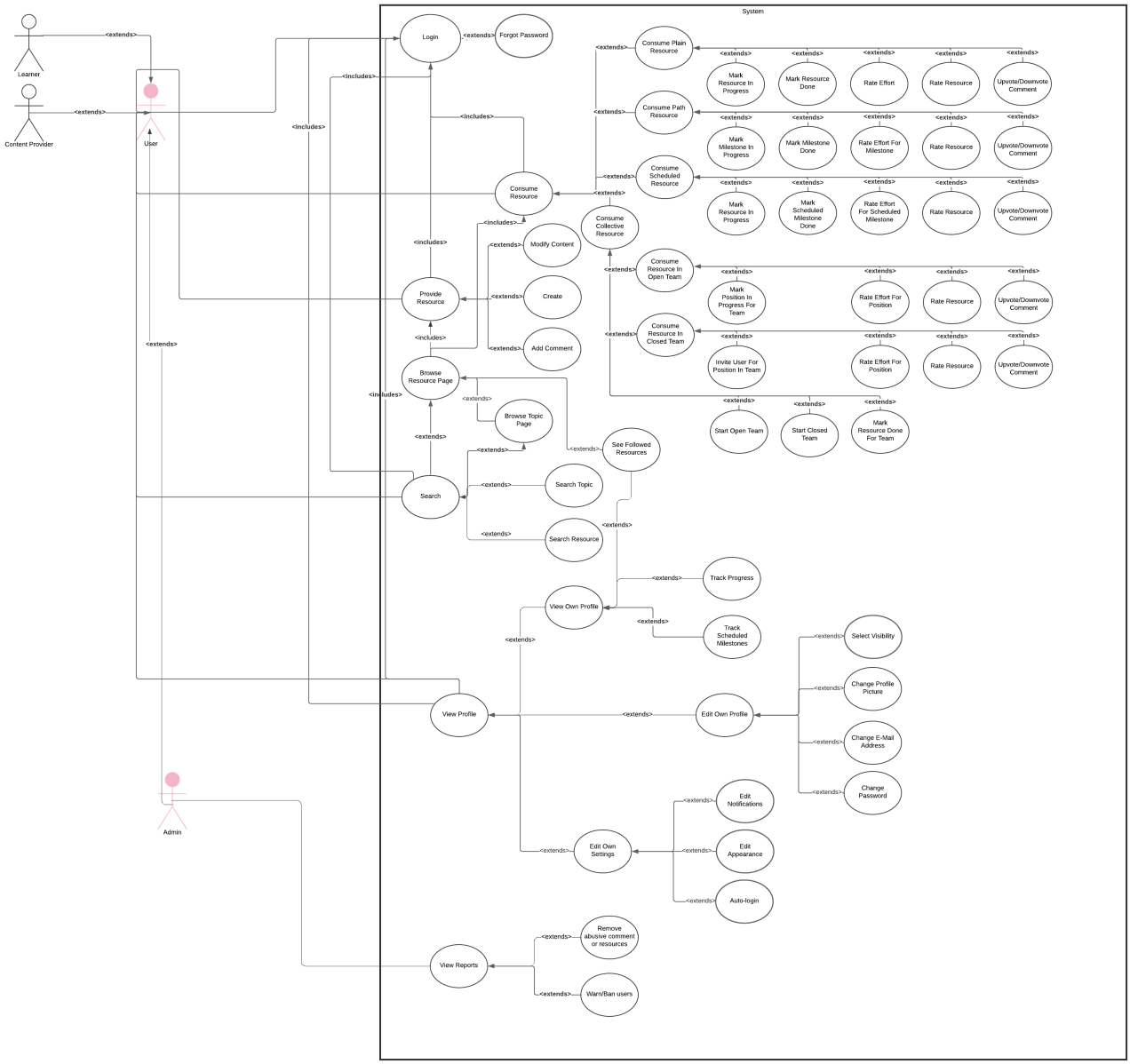


Old Versions

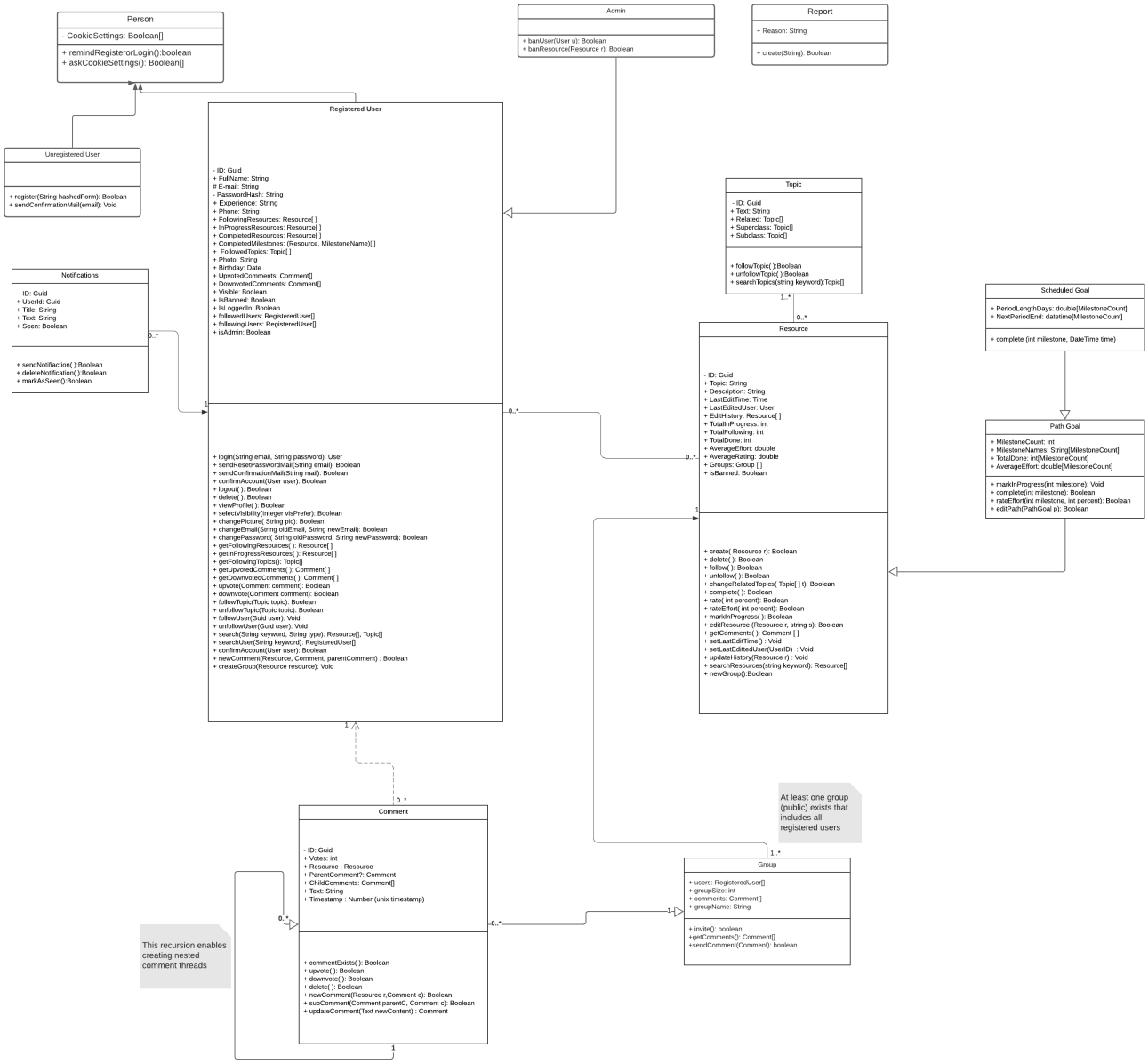
Changelog: 31.10.2021

- Admin can add new admin
- Admin has management page
- Direct access settings page after login
- They can see your courses from the settings page
- Move some features from edit own profile to new manage account page
- View comment under resource



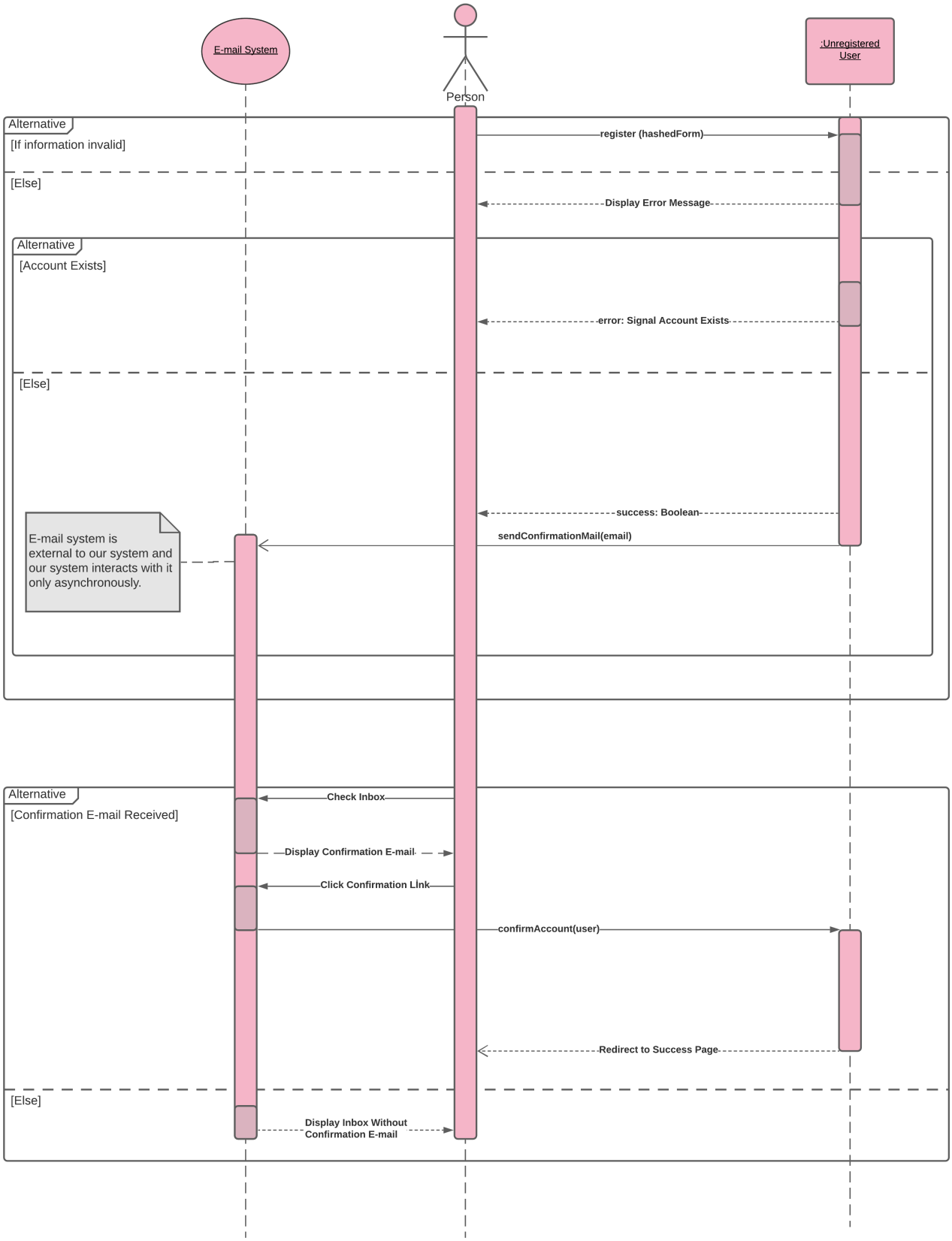


Class Diagram

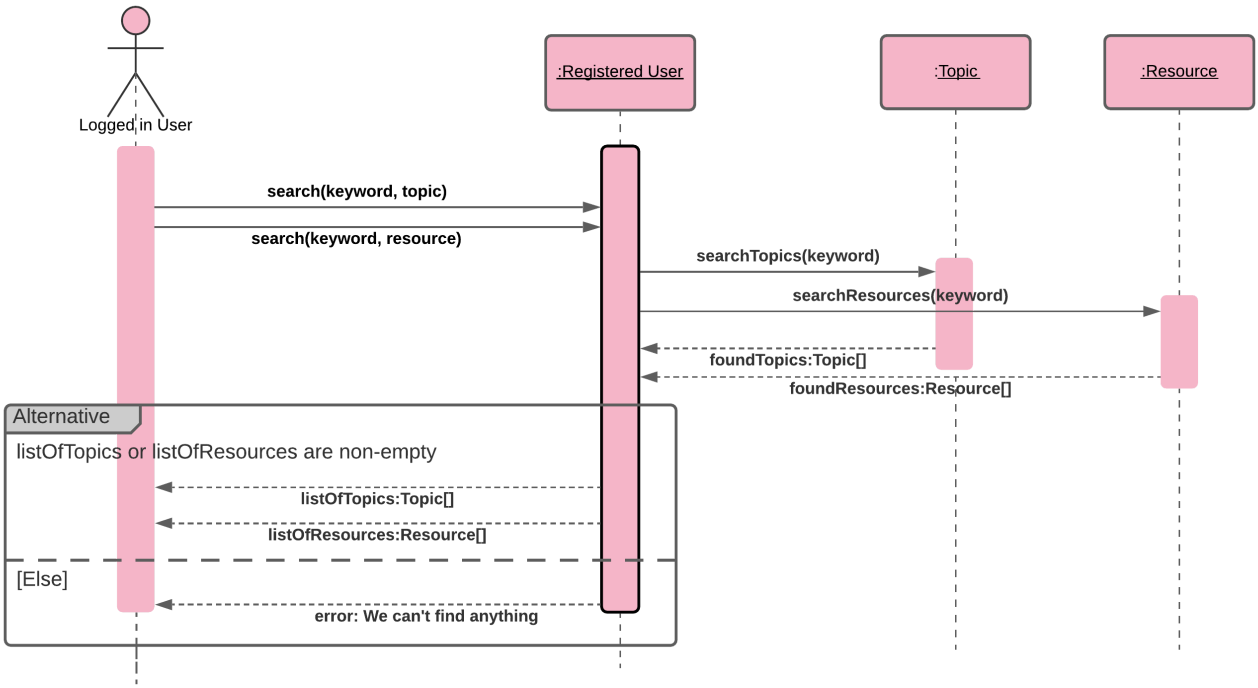


Sequence Diagram

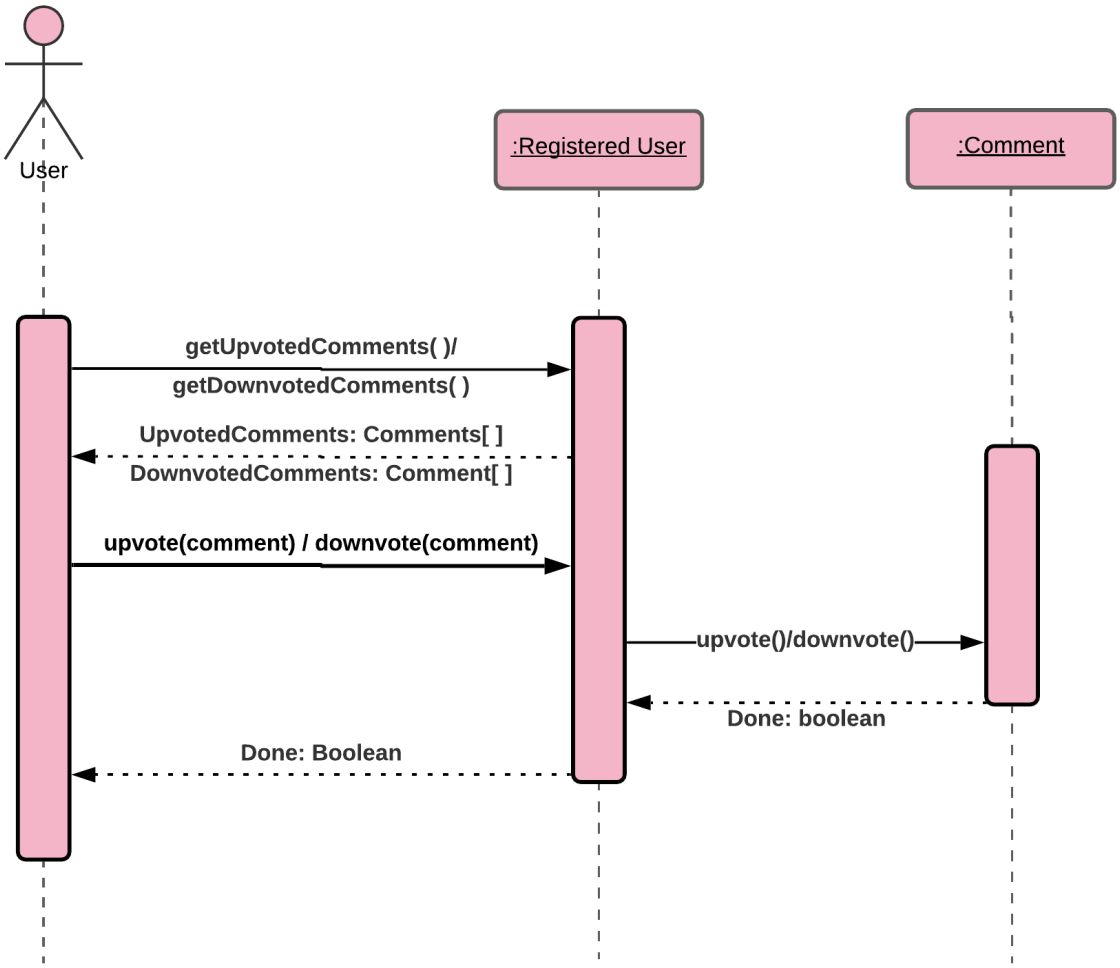
Sign Up


E-mail System
Person
:Unregistered User
Alternative
[If information invalid]
register (hashedForm)
[Else]
Display Error Message
Alternative
[Account Exists]
error: Signal Account Exists
[Else]
success: Boolean
sendConfirmationMail(email)
E-mail system is external to our system and our system interacts with it only asynchronously.
Alternative
[Confirmation E-mail Received]
Check Inbox
Display Confirmation E-mail
Click Confirmation Link
confirmAccount(user)
Redirect to Success Page
[Else]
Display Inbox Without Confirmation E-mail

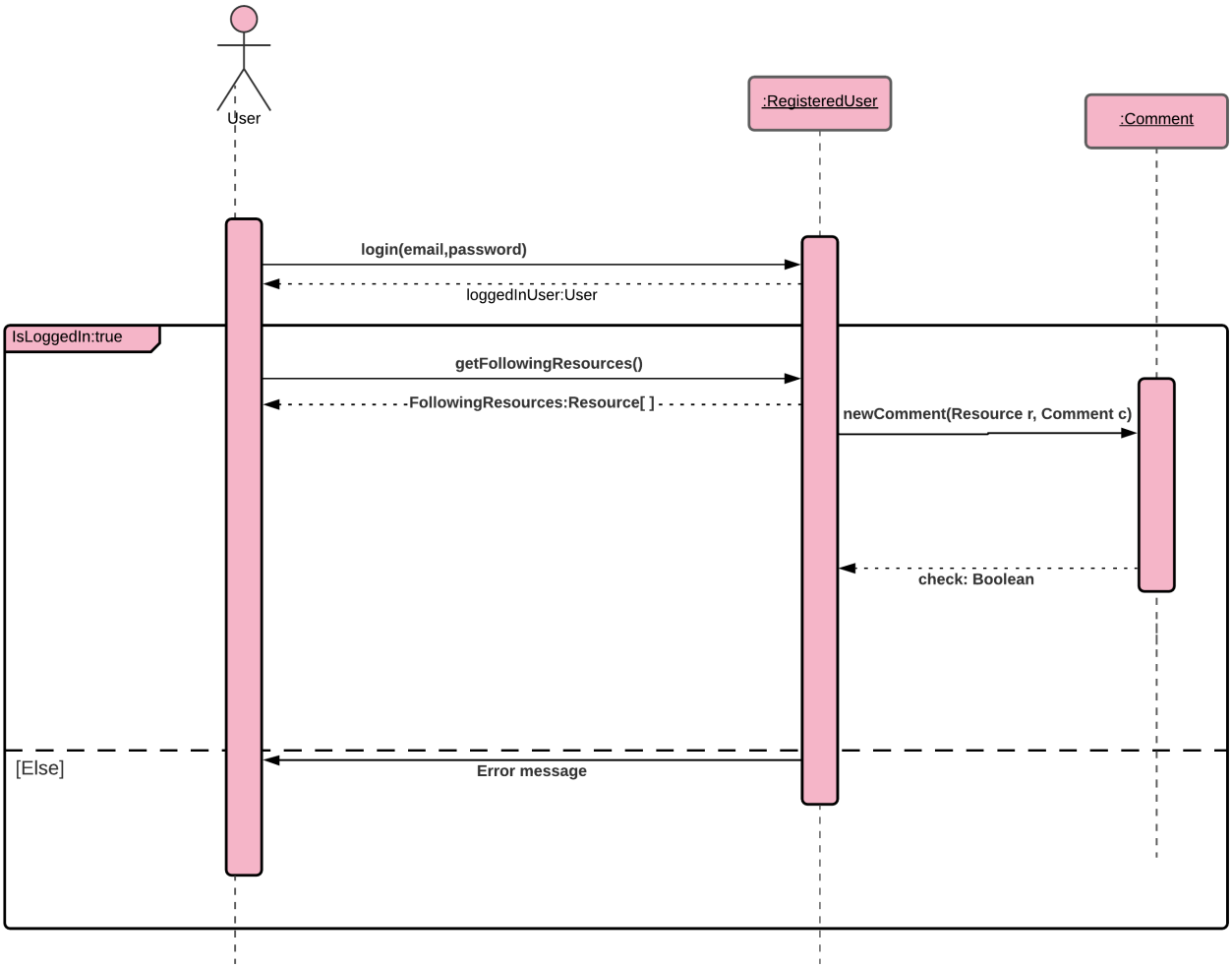
Search



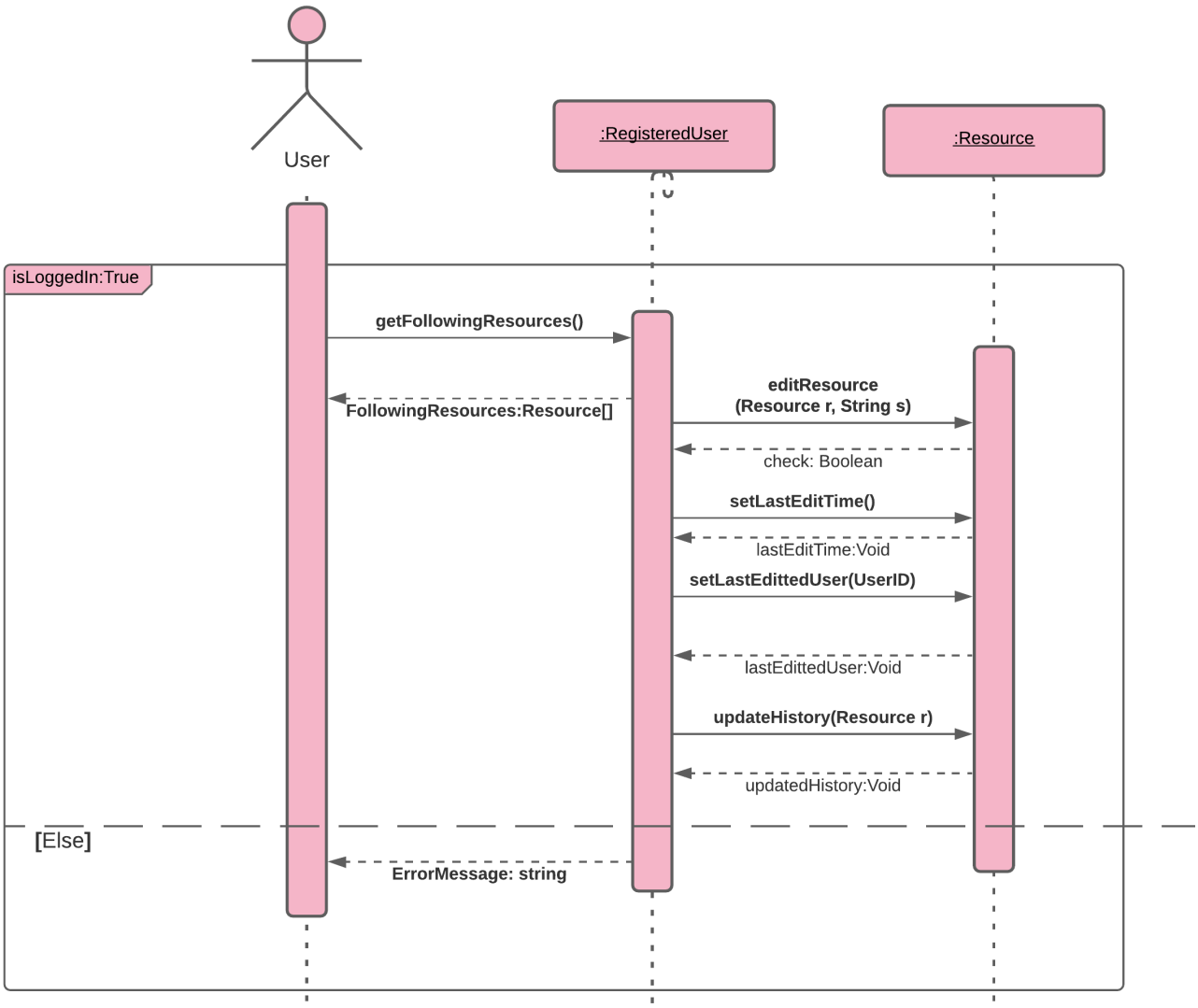
Upvote/Downvote



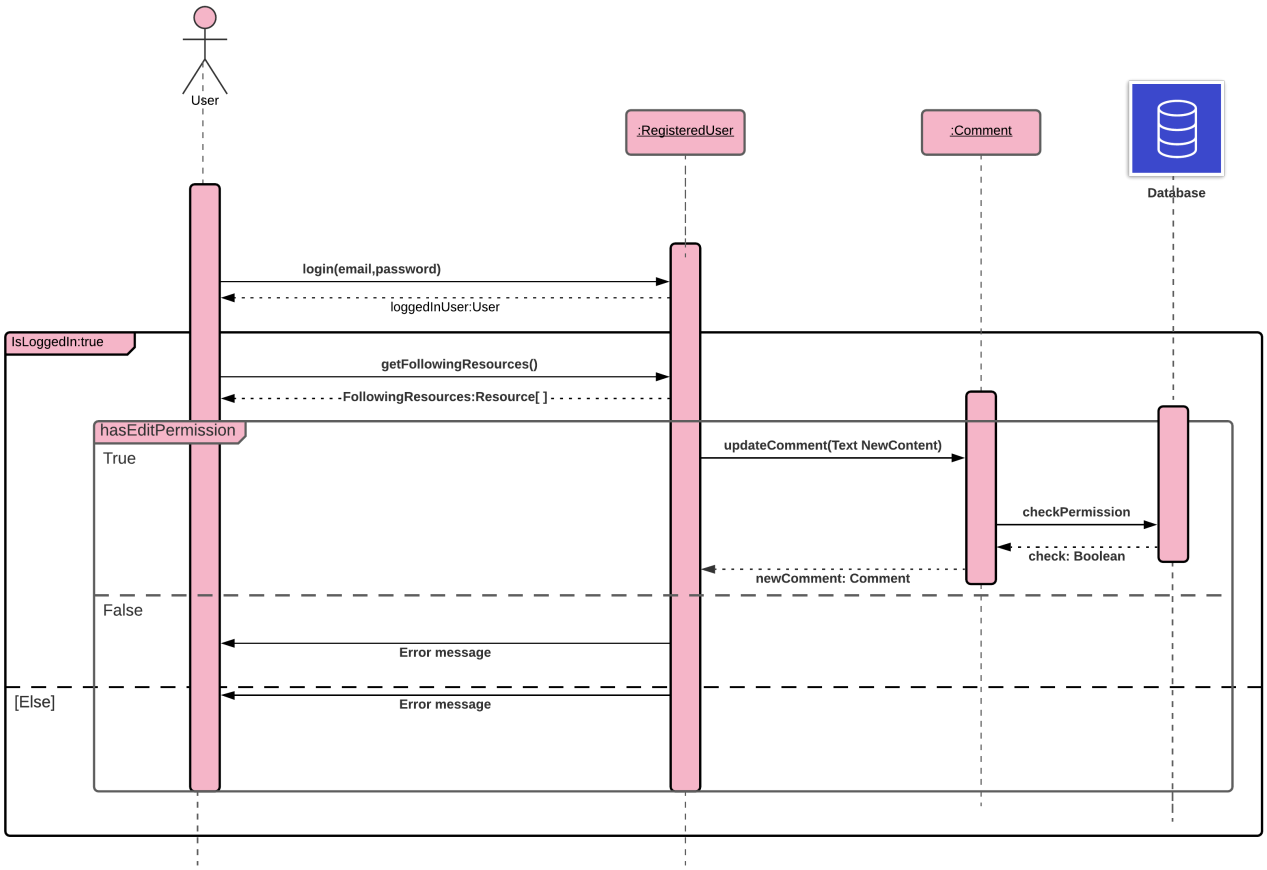
Post a Comment



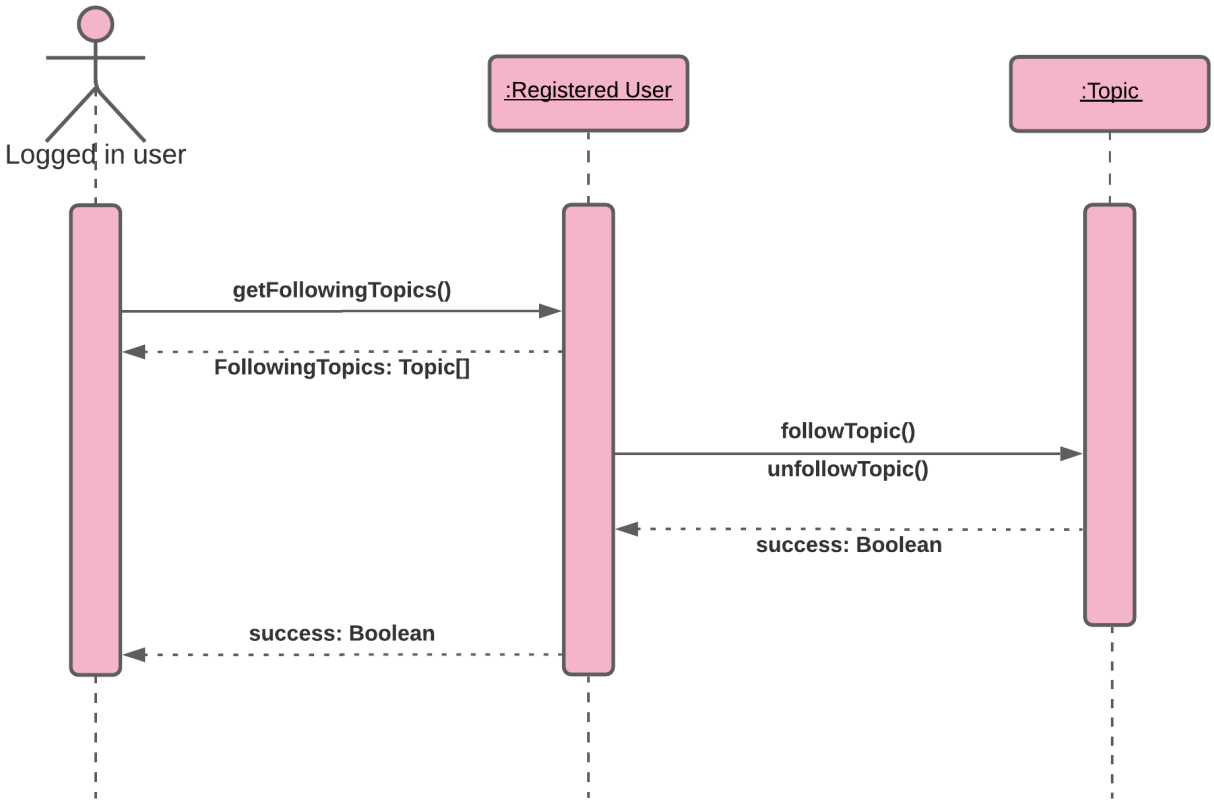
Edit Resource



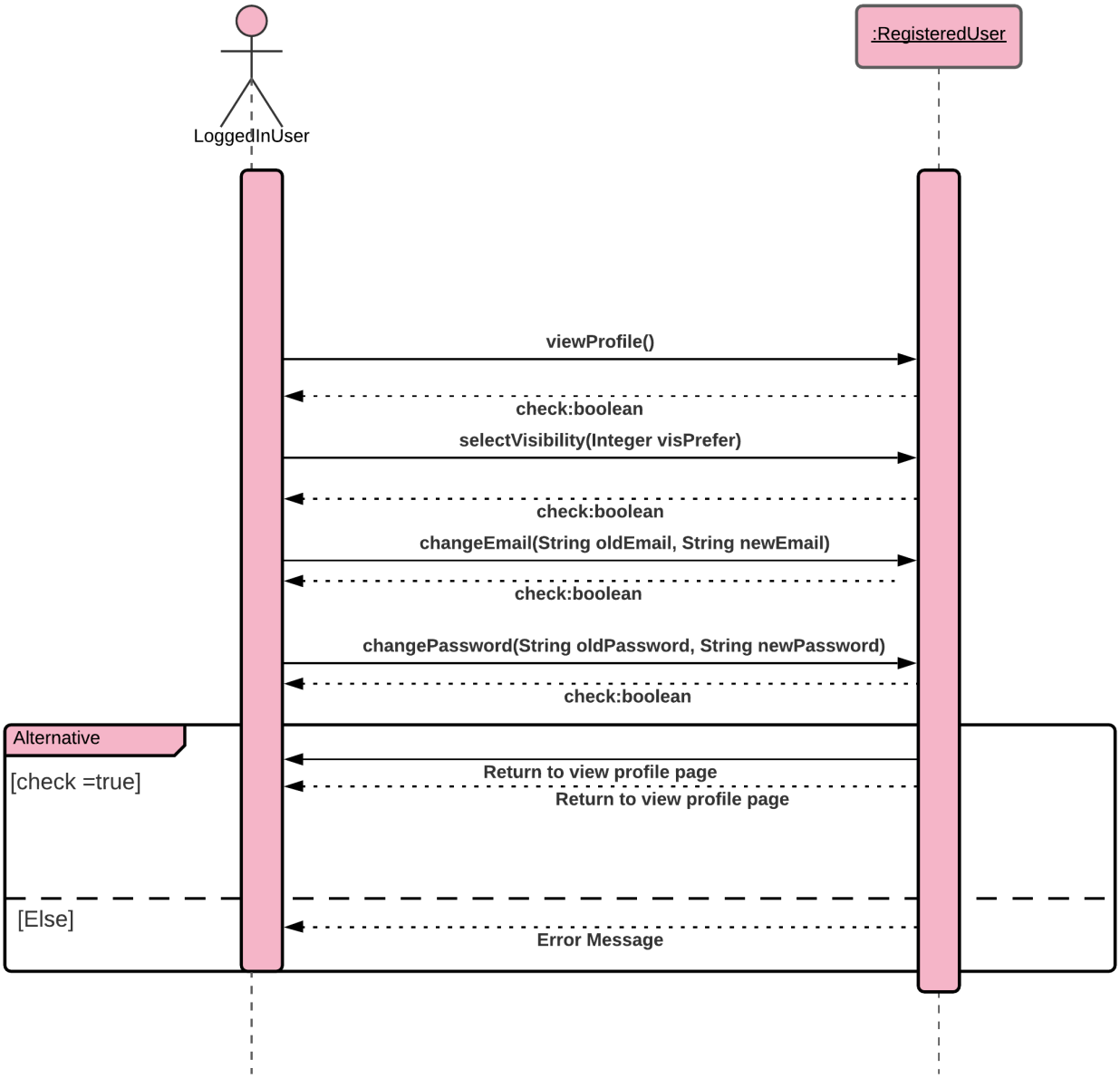
Edit Comment



Follow/Unfollow Topic



Edit Profile



Scenarios and Mockups

Scenario 1

User Persona

- **Photo:**

- **Name:** Mohammad Lee
- **Age:** 42
- **Job:** Businessperson
- **Interest:**

1. Mohammed tries to improve himself in various programming topics in order to strengthen his communication with the technical team of his company.
2. At the weekends, he likes to cook for his wife and his little daughter.
3. He workouts for 30 minutes every morning as starting the day.

Preconditions

- Mohammad is already signed up and logged in the android app.
- The app is started in Home page.

Goals

- Mohammad has more leisure time nowadays, so he wants to explore some new topics to be acknowledged about them according to his past interest areas.
- He decides to enter the his favorite app and see what it suggests.

Acceptance Criteria

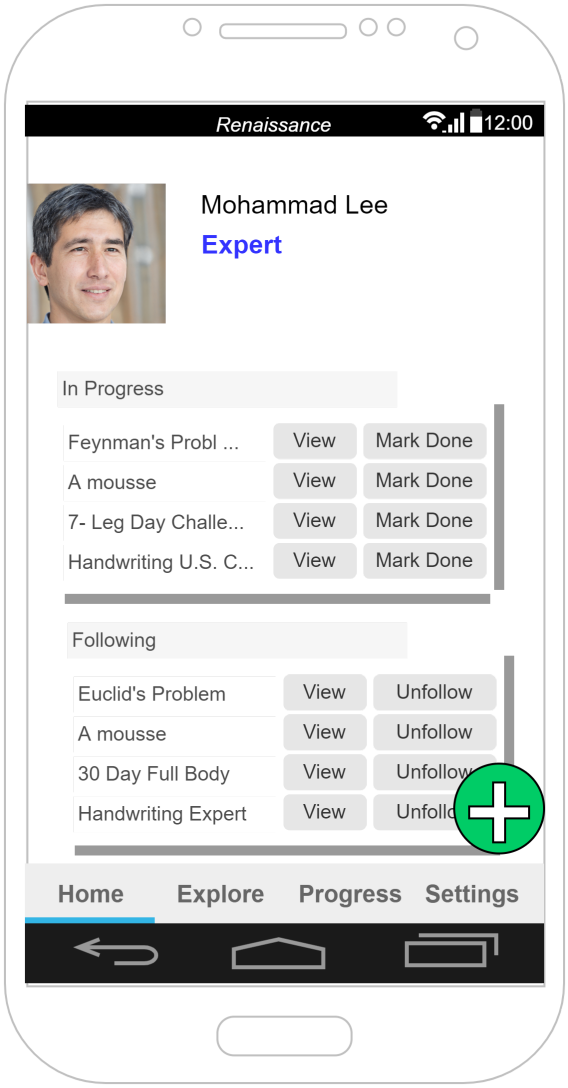
In reference to the Revision 1.1 of the requirements

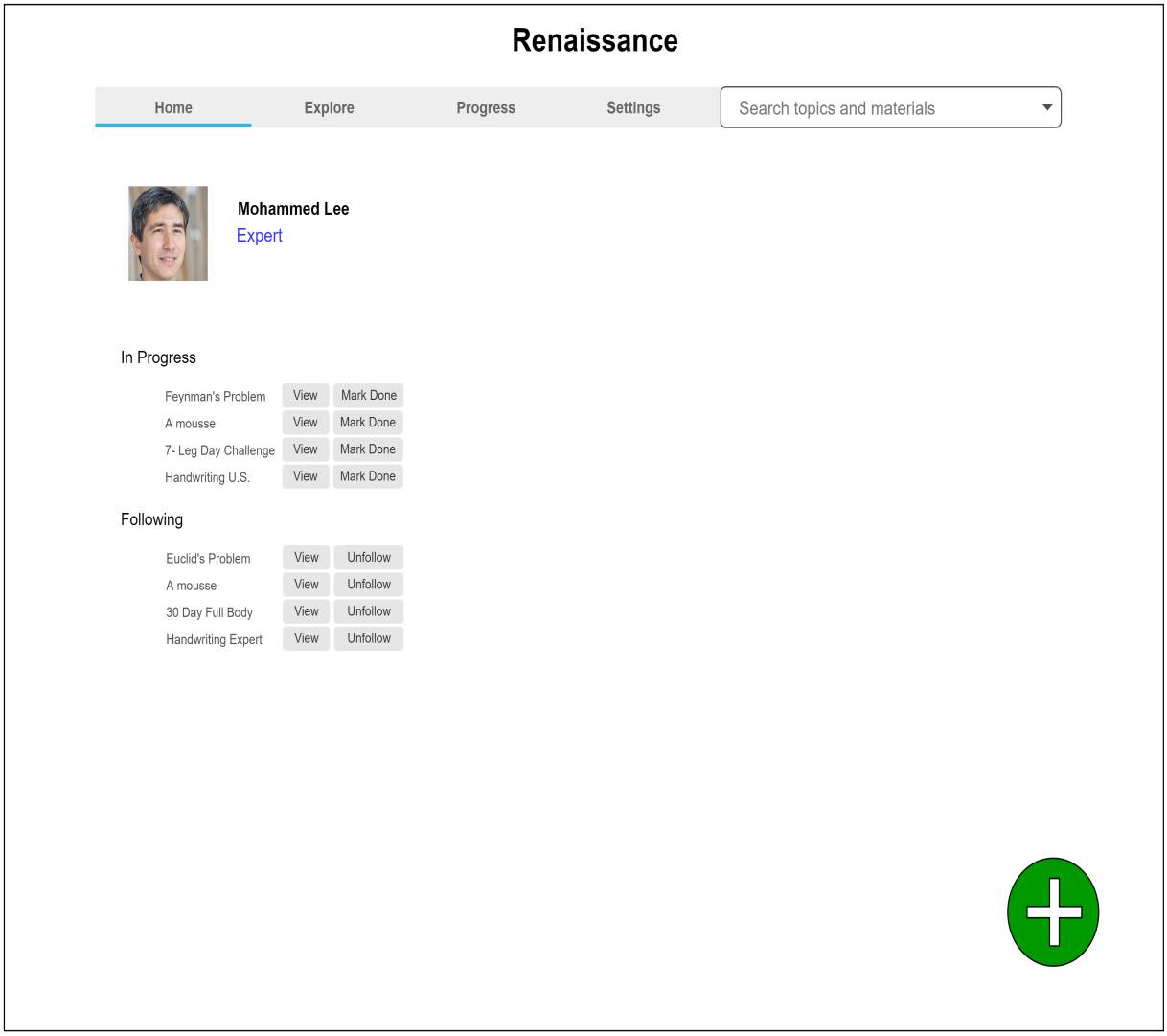
- **(Requirement 1.1.2.)** The System shall provide its users access to it by means of specialized accounts.
- **(Requirement 1.2.3.)** The System shall determine and inform its users about topics they could be potentially interested.

- **(Requirement 1.2.4.)** The System shall supply its users with recommendation of materials.
- **(Requirement 1.2.7.)** The users should be able to reliably record topics they want to improve on.
- **(Requirement 1.3.1.)** The users should be able to create custom pages representing their accounts and those pages that ease their access to other functionalities, like a dashboard.
- **(Requirement 1.3.4.)** The users should be able to share their comments on specific topics.
- **(Requirement 1.4.2.)** The users should be able to customize the way the notifications are sent.
- **(Requirement 2.1.1.)** An Android Application and a Web Application shall be implemented as instances of the Interface.

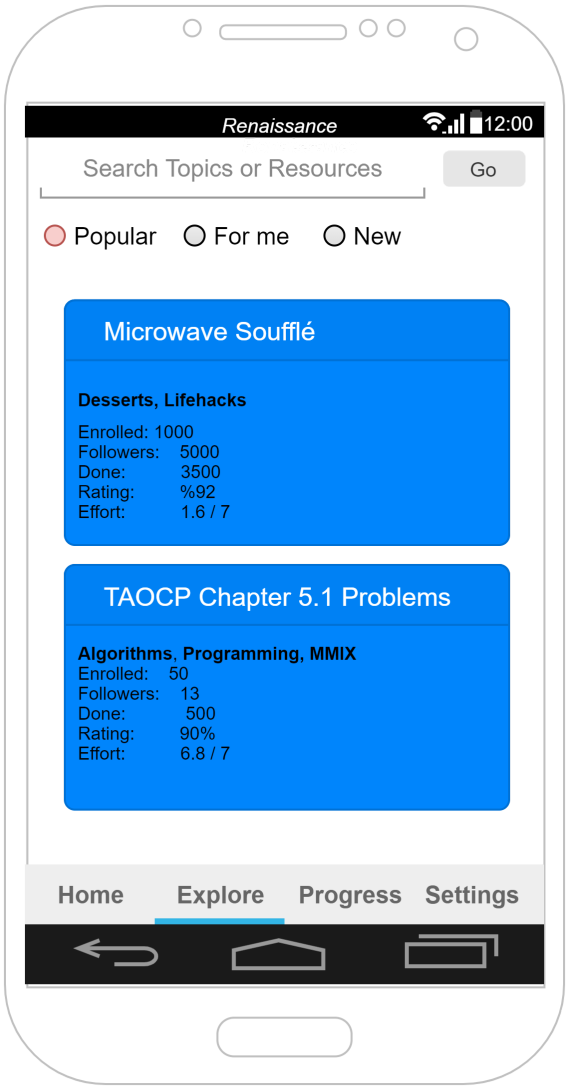
Actions & Mockups

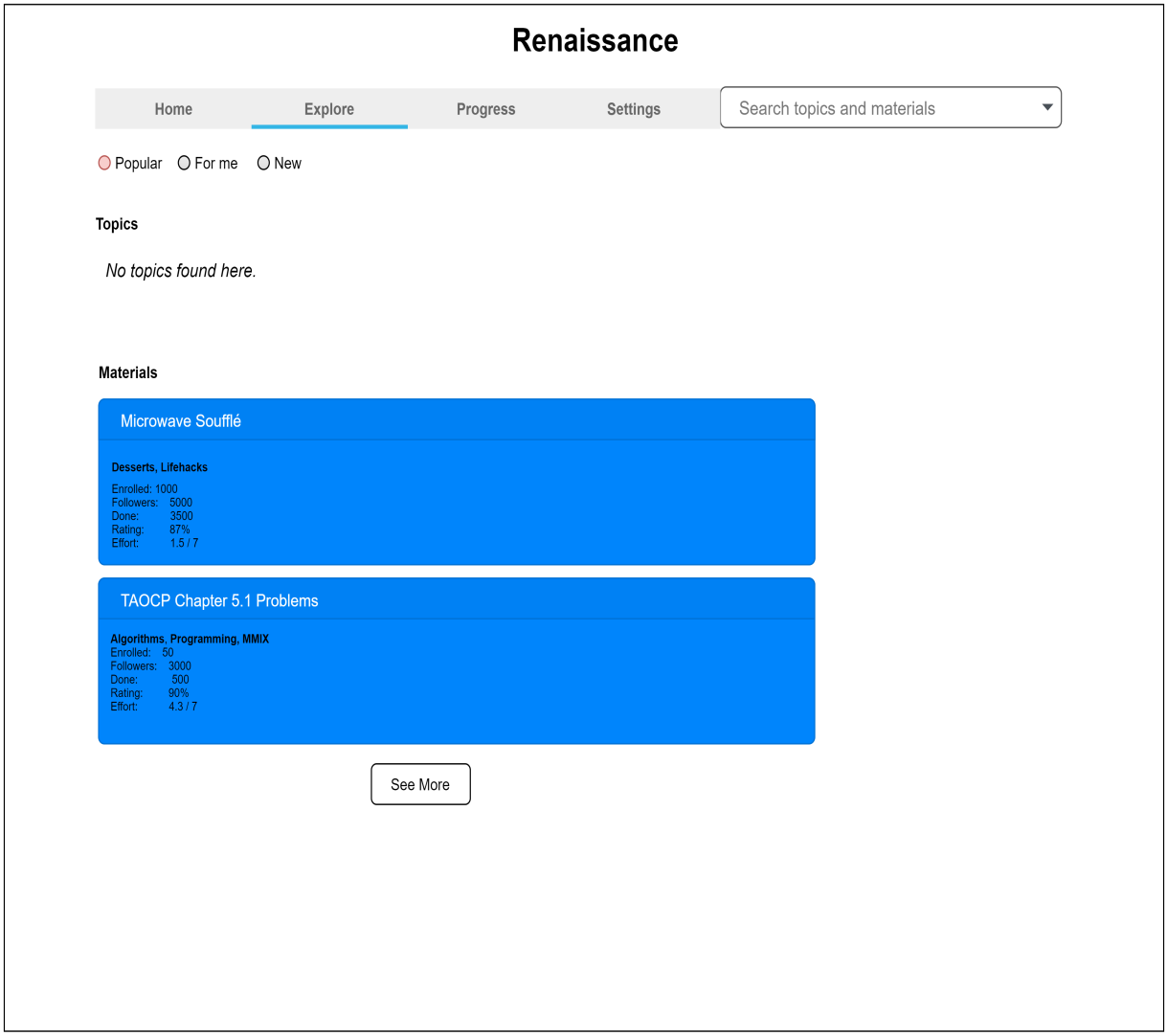
1. Mohammad is in pursuit of meeting a new topic increasing his technical knowledge. He is already an experienced user of our app. Since he is auto logged in to the app, when he clicks, the home page is opened.


Renaissance
12:00
Mohammad Lee
Expert
In Progress
Feynman's Probl ...
View
Mark Done
A mousse
View
Mark Done
7- Leg Day Challe...
View
Mark Done
Handwriting U.S. C...
View
Mark Done
Following
Euclid's Problem
View
Unfollow
A mousse
View
Unfollow
30 Day Full Body
View
Handwriting Expert
View
Home
Explore
Progress
Settings

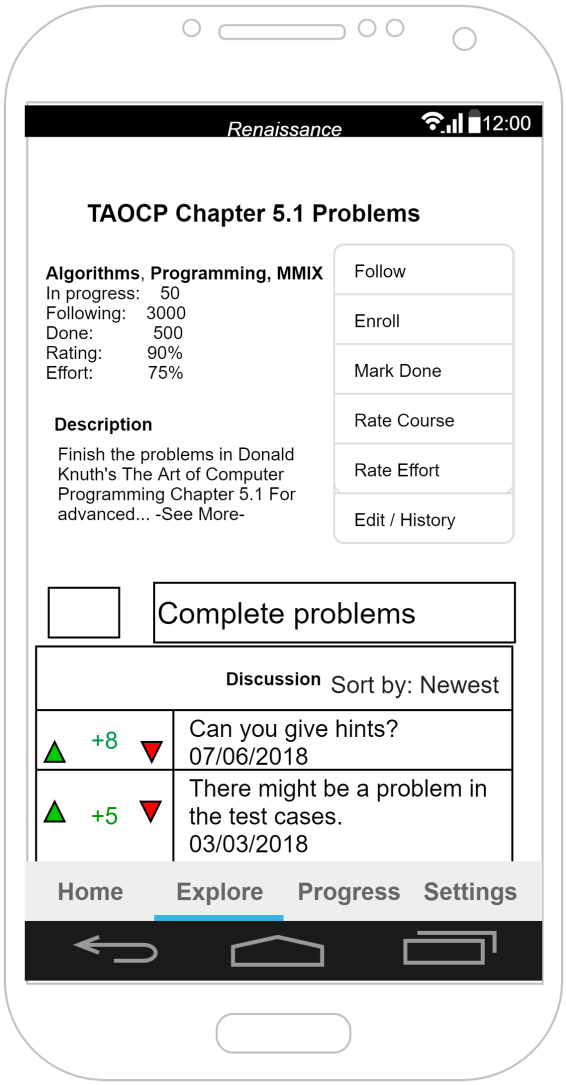


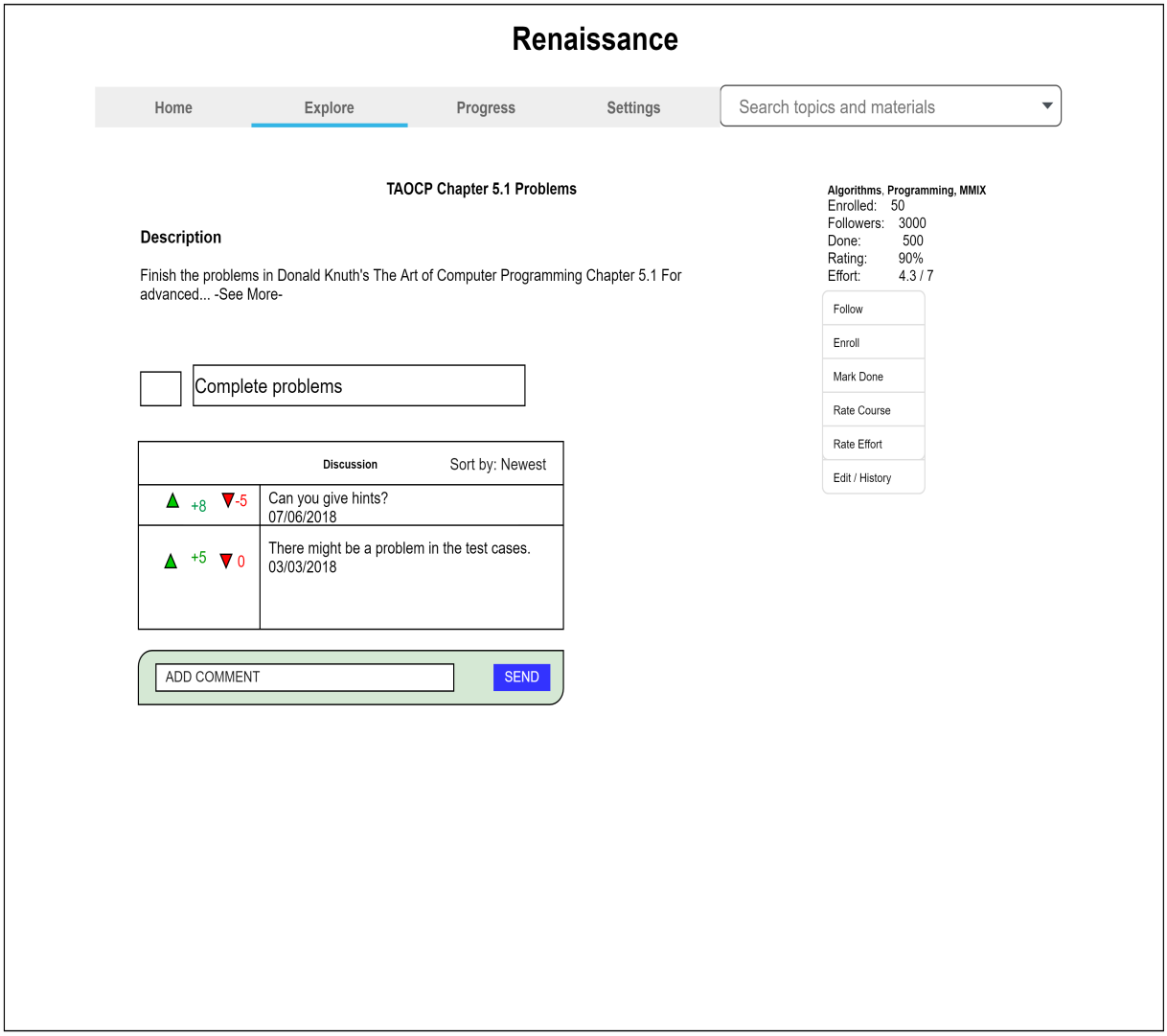
2. He goes to the Explore tab to see which topics the app suggests him. There are some options according to his past interests.


Renaissance
12:00
Search Topics or Resources
Go
Popular
For me
New
Microwave Soufflé
Desserts, Lifehacks
Enrolled: 1000
Followers: 5000
Done: 3500
Rating: %92
Effort: 1.6 / 7
TAOCP Chapter 5.1 Problems
Algorithms, Programming, MMIX
Enrolled: 50
Followers: 13
Done: 500
Rating: 90%
Effort: 6.8 / 7
Home
Explore
Progress
Settings

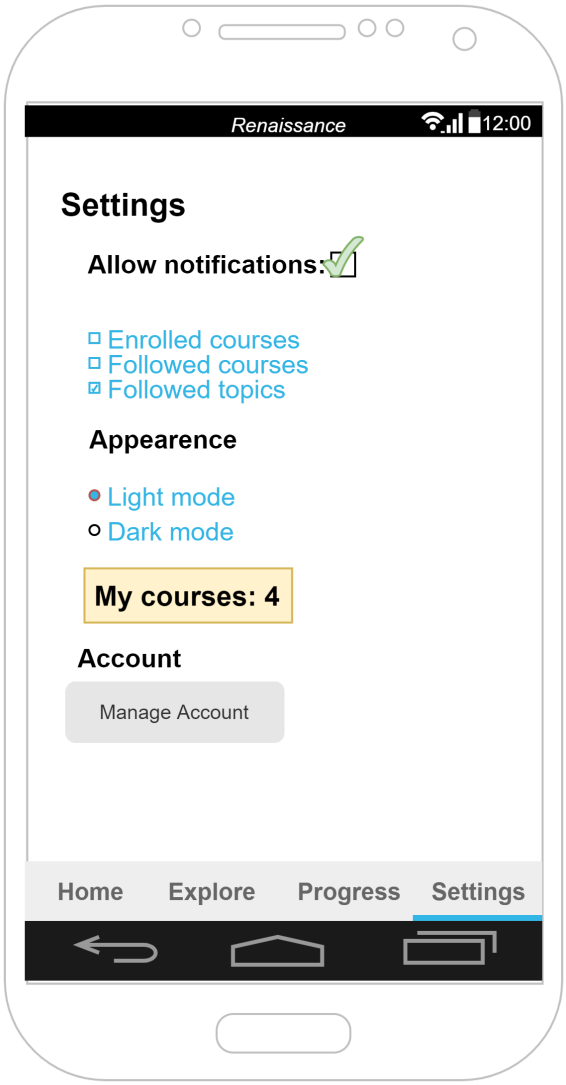


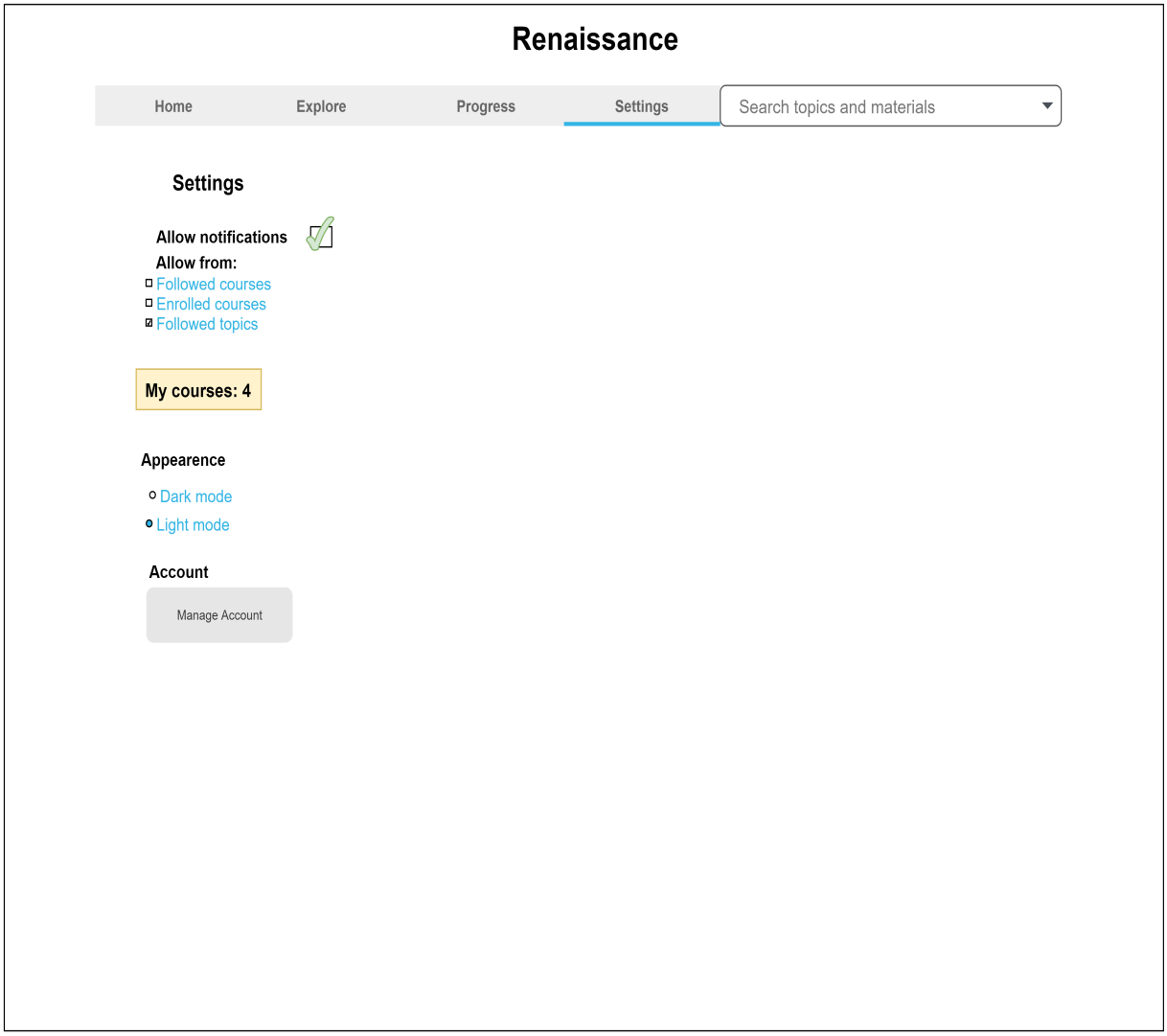
3. He clicks the second topic to view the related page including discussion part and various materials of this topic and follows it.


Renaissance
12:00
TAOCP Chapter 5.1 Problems
Algorithms, Programming, MMIX
In progress: 50
Following: 3000
Done: 500
Rating: 90%
Effort: 75%
Description
Finish the problems in Donald Knuth's The Art of Computer Programming Chapter 5.1 For advanced... -See More-
Follow
Enroll
Mark Done
Rate Course
Rate Effort
Edit / History
Complete problems
Discussion
Sort by: Newest
+8
Can you give hints?
07/06/2018
+5
There might be a problem in the test cases.
03/03/2018
Home
Explore
Progress
Settings



4. After he finishes his job in the Explore tab, he goes to the Settings tab. He turns on his notifications of the topics he follows with the curiosity of starting following a new topic before quitting the app.


Renaissance
12:00
Settings
Allow notifications:
Enrolled courses
Followed courses
Followed topics
Appearence
Light mode
Dark mode
My courses: 4
Account
Manage Account
Home
Explore
Progress
Settings



Scenario 2

User Persona

- **Photo:**

- **Name:** Ramesh Kardashayan
- **Age:** 32
- **Job:** Computer Engineer

- **Interest:** Programming, Handwriting, Rick Astley

Preconditions

- Ramesh is already signed up and logged in the android app.

Goals

- He wants to change the classification of an HTML exercise.

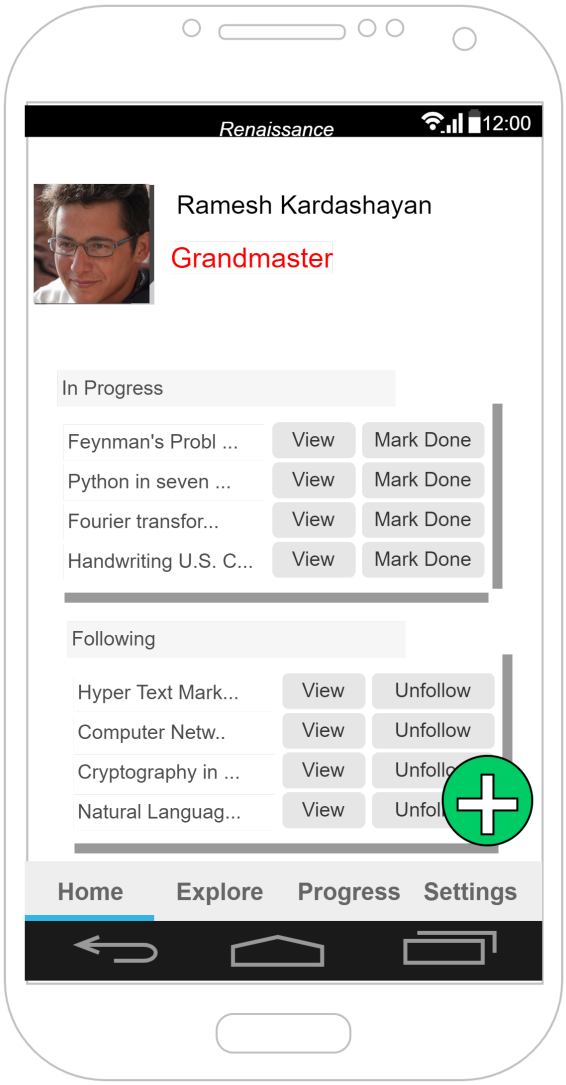
Acceptance Criteria

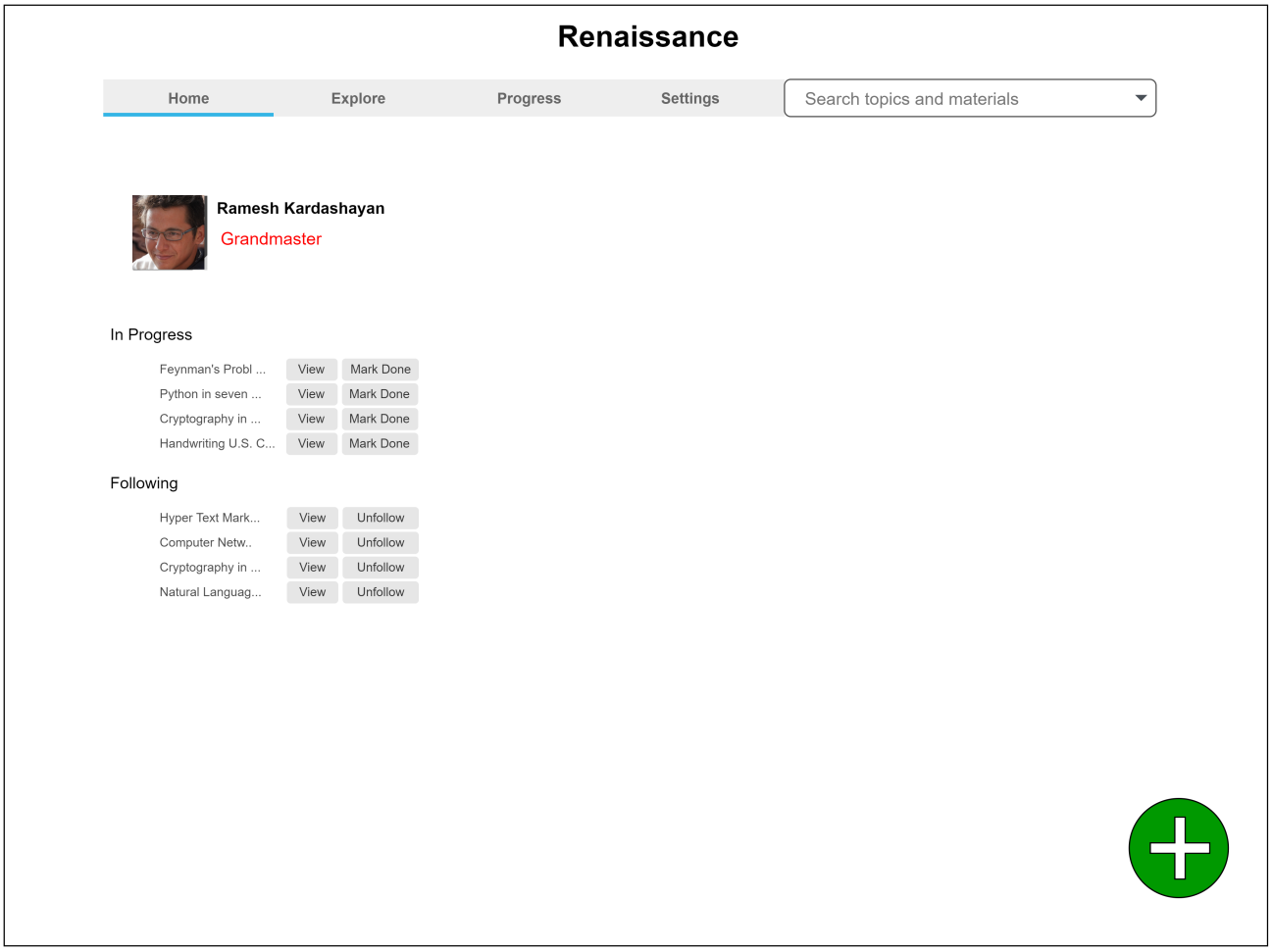
In reference to the Revision 1.1 of the requirements

- **(Requirement 1.1.2.)** The System shall provide its users access to it by means of specialized accounts.
- **(Requirement 1.2.4.)** The System shall supply its users with recommendation of materials.
- **(Requirement 1.2.7.)** The users should be able to reliably record topics they want to improve on.
- **(Requirement 1.3.4.)** The users should be able to share their comments on specific topics.
- **(Requirement 2.1.1.)** An Android Application and a Web Application shall be implemented as instances of the Interface.

Actions & Mockups

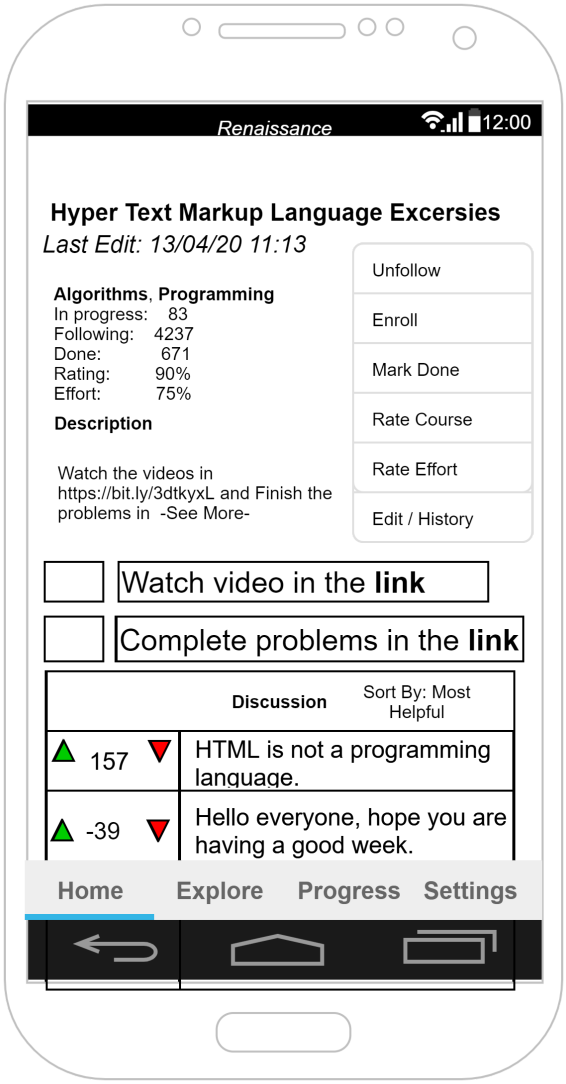
1. Ramesh opens his app and home page appears (Figure 1). He wonders what's new in his following content Hyper Text Markup Language Exercises and clicks the View Button.

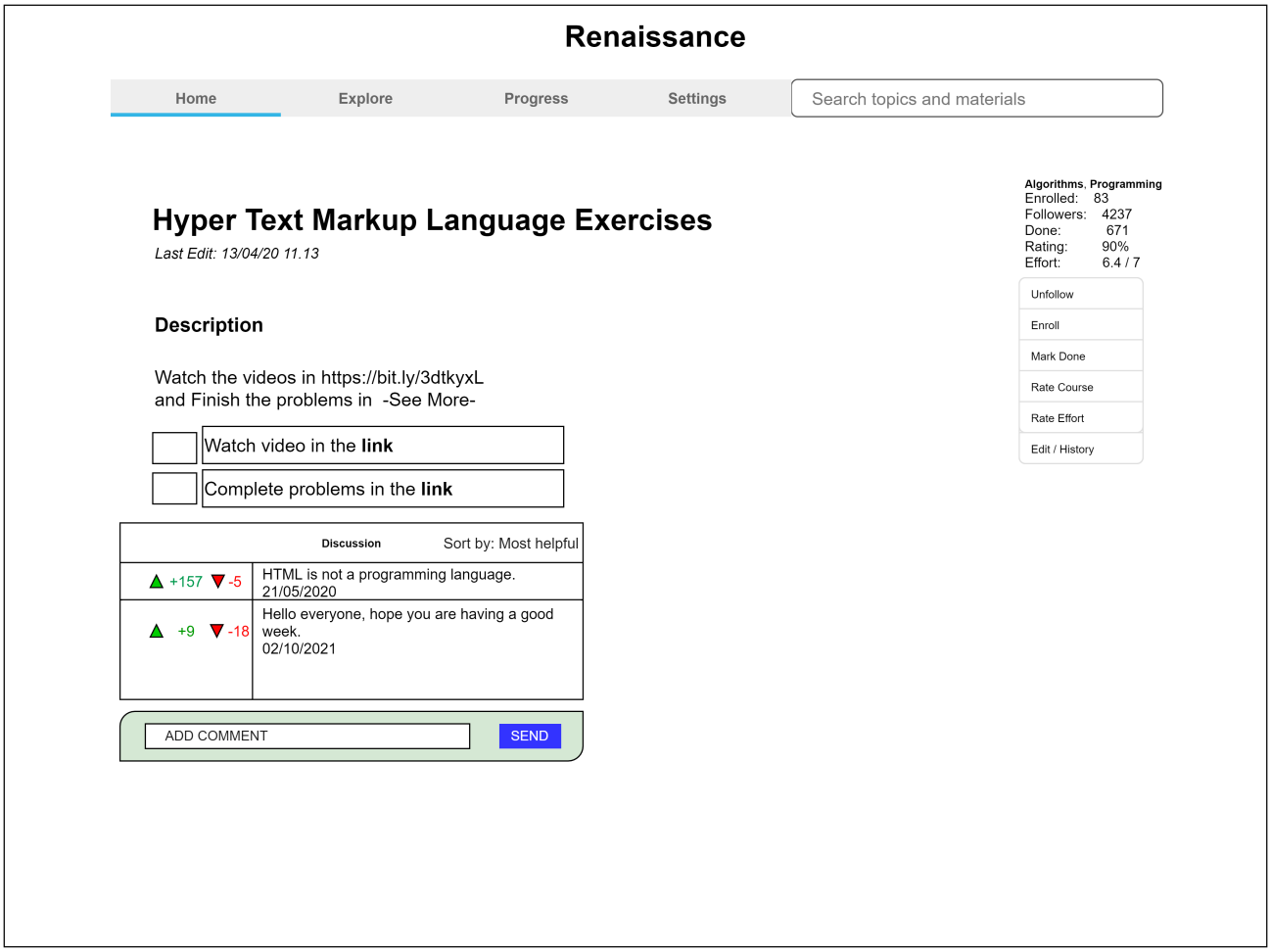

Renaissance
12:00
Ramesh Kardashayan
Grandmaster
In Progress
Feynman's Probl ...
View
Mark Done
Python in seven ...
View
Mark Done
Fourier transfor...
View
Mark Done
Handwriting U.S. C...
View
Mark Done
Following
Hyper Text Mark...
View
Unfollow
Computer Netw..
View
Unfollow
Cryptography in ...
View
Natural Languag...
View
Home
Explore
Progress
Settings



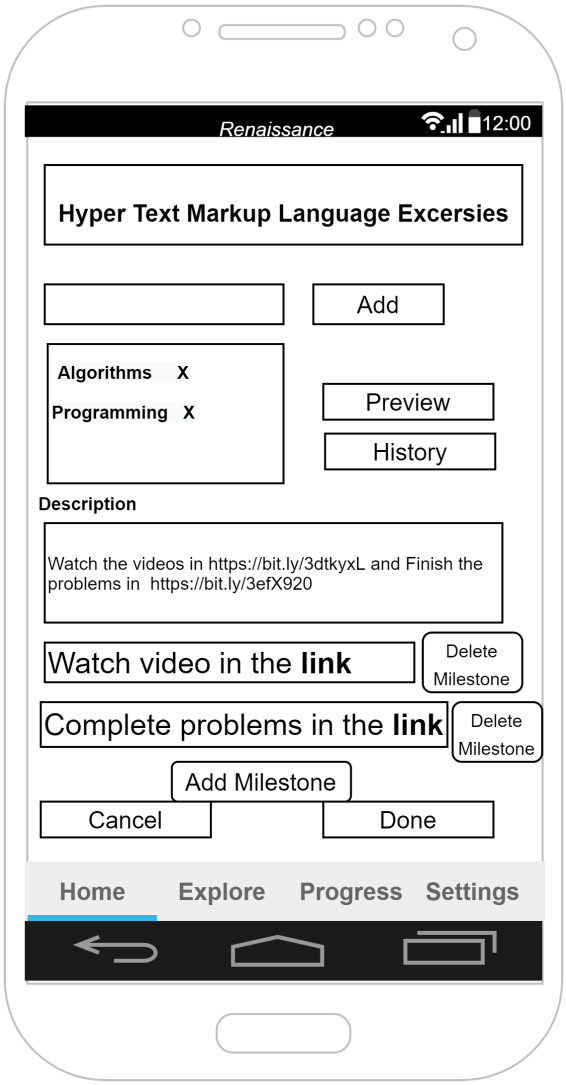
2. Hyper Text Markup Language Exercises page appears and he noticed that related topics of this page are wrong. So, he decides to change the related topics and press the

Edit/History button.





3. Edit page appears. He can add/delete topics, change the description or header. He deletes the related topics and add the new ones. Then press the Done Button.


Renaissance
12:00
Hyper Text Markup Language Excersies
Add
Algorithms X
Programming X
Preview
History
Description
Watch the videos in https://bit.ly/3dtkyxL and Finish the problems in https://bit.ly/3efX920
Watch video in the link
Delete Milestone
Complete problems in the link
Delete Milestone
Add Milestone
Cancel
Done
Home
Explore
Progress
Settings

Renaissance

Home | Explore | Progress | Settings | Search topics and materials

Title

Hyper Text Markup Language Exercises

Tags

Add

Algorithms X

Programming X

Description

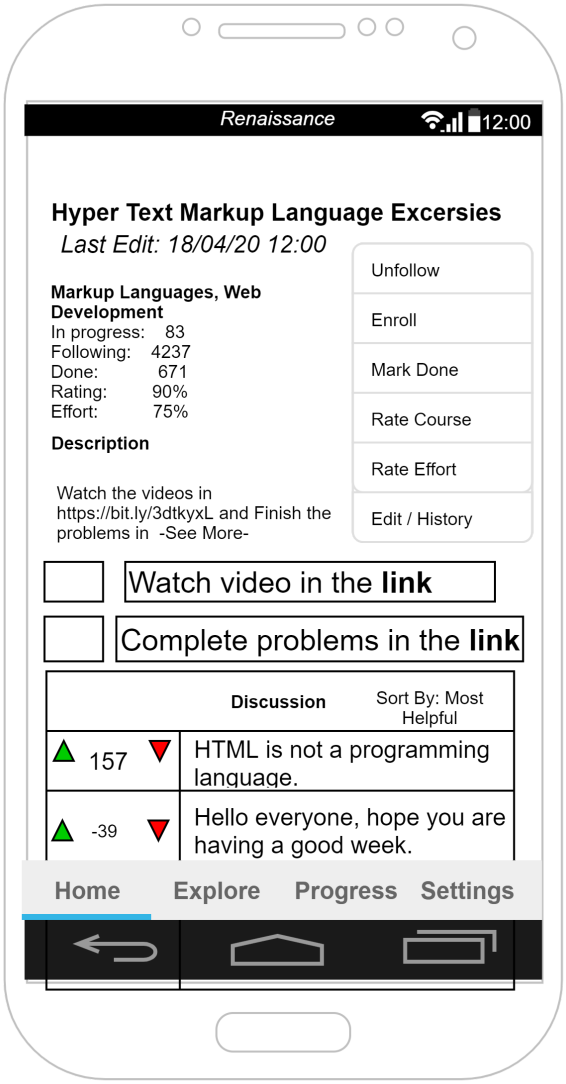
Watch the videos in https://bit.ly/3dtkyxL
and Finish the problems in -See More-

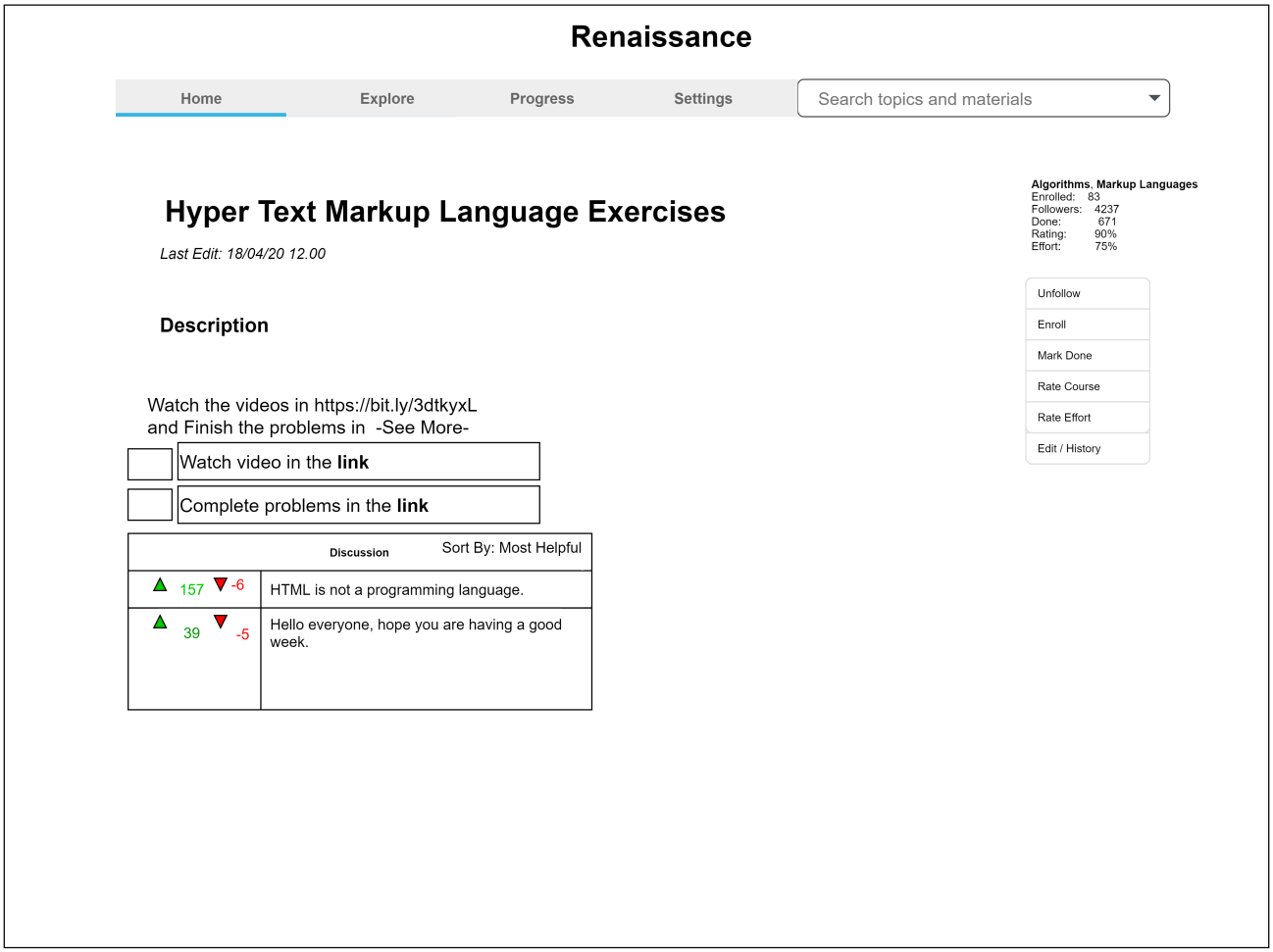
- [ ] Watch video in the **link** — Delete Milestone
- [ ] Complete problems in the **link** — Delete Milestone

Add Milestone

Done | Cancel

4. Related topics and the last edit date changed.





Scenario 3

User Persona

- **Name:** Tom Marvolo Riddle
- **Age:** 16
- **Job:** Student
- **Interest:** Dark Magic, Magic, Duelling

Preconditions

- Tom is in the web app.
- Tom has not any account and is not signed up.

Goals

- To become immortal. (to find a material about a topic he is interested in)

- To find out about how to create a crucifix to ensure that.

Acceptance Criteria

In reference to the Revision 1.1 of the requirements

- **(Requirement 1.1.2.)** The System shall provide its users access to it by means of specialized accounts.
- **(Requirement 1.1.3.)** The users shall be able to create and delete their accounts and be able to gain access if they prove identity in case they forget their password.
- **(Requirement 1.2.7.)** The users should be able to reliably record topics they want to improve on.
- **(Requirement 1.4.3.)** The users should be able to search data assigned to topics using keywords.
- **(Requirement 2.1.1.)** An Android Application and a Web Application shall be implemented as instances of the Interface.
- **(Requirement 2.1.3.)** The System shall express topics using semantic representation as defined in Wikidata.

Actions & Mockups

1. He opens the site.

Renaissance

Sign in

Email address

email@example.com

Password

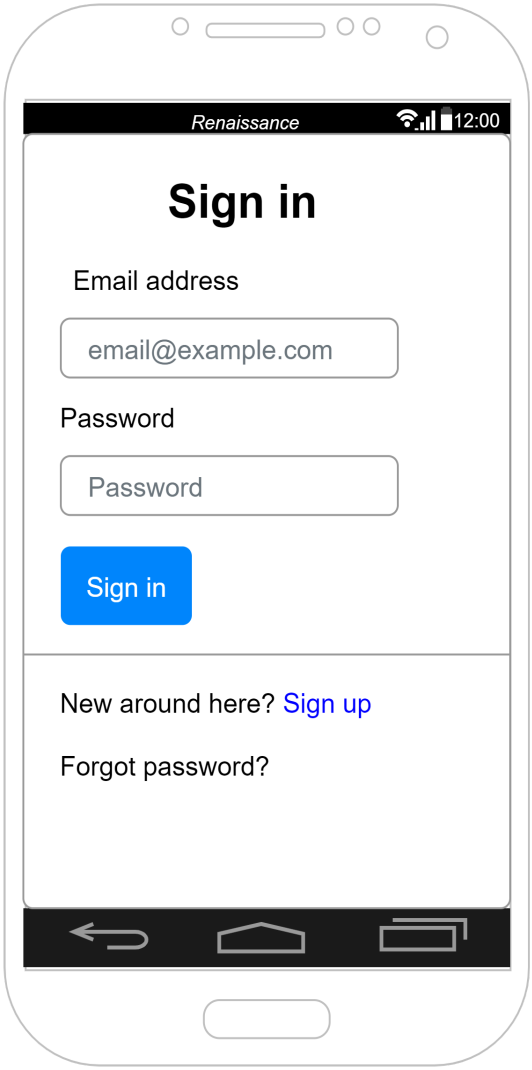
Password

Remember me

Sign in

New around here? Sign up

Forgot password?



2. He realizes that he does not have an account. He clicks the sign up button then redirected to sign up page.

Renaissance

Sign up

Name

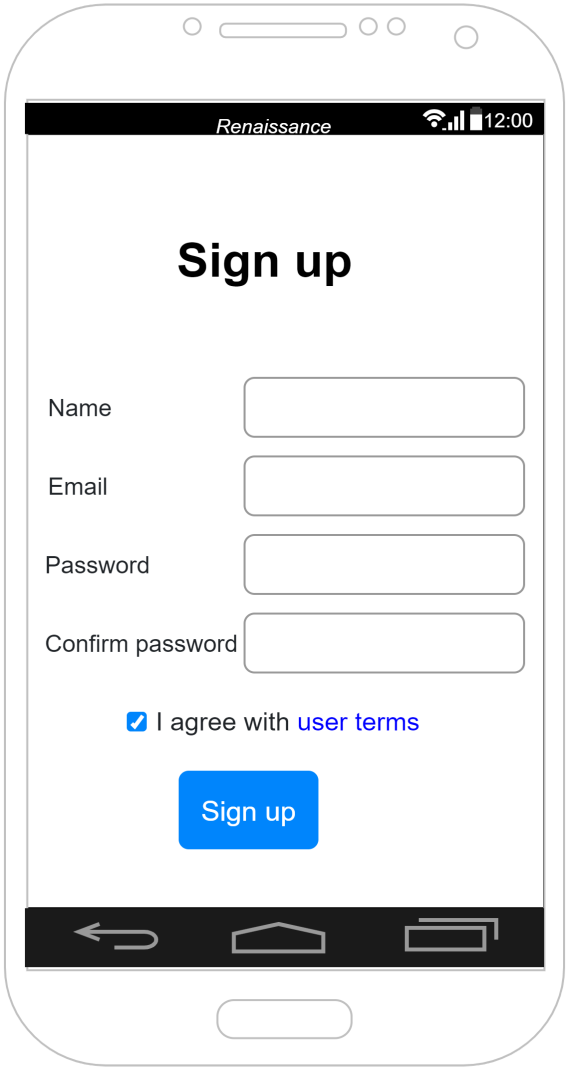
Email

Password

Confirm Password

☑ I agree with user terms

Sign up

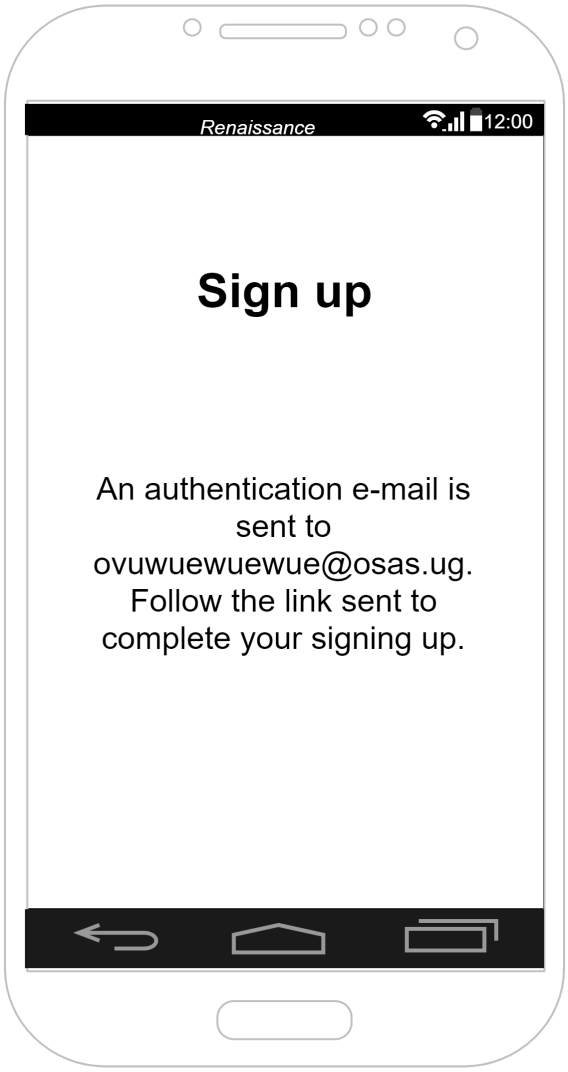


3. He provides the required information and by clicking the sign-up button he is redirected to the e-mail confirmation page.

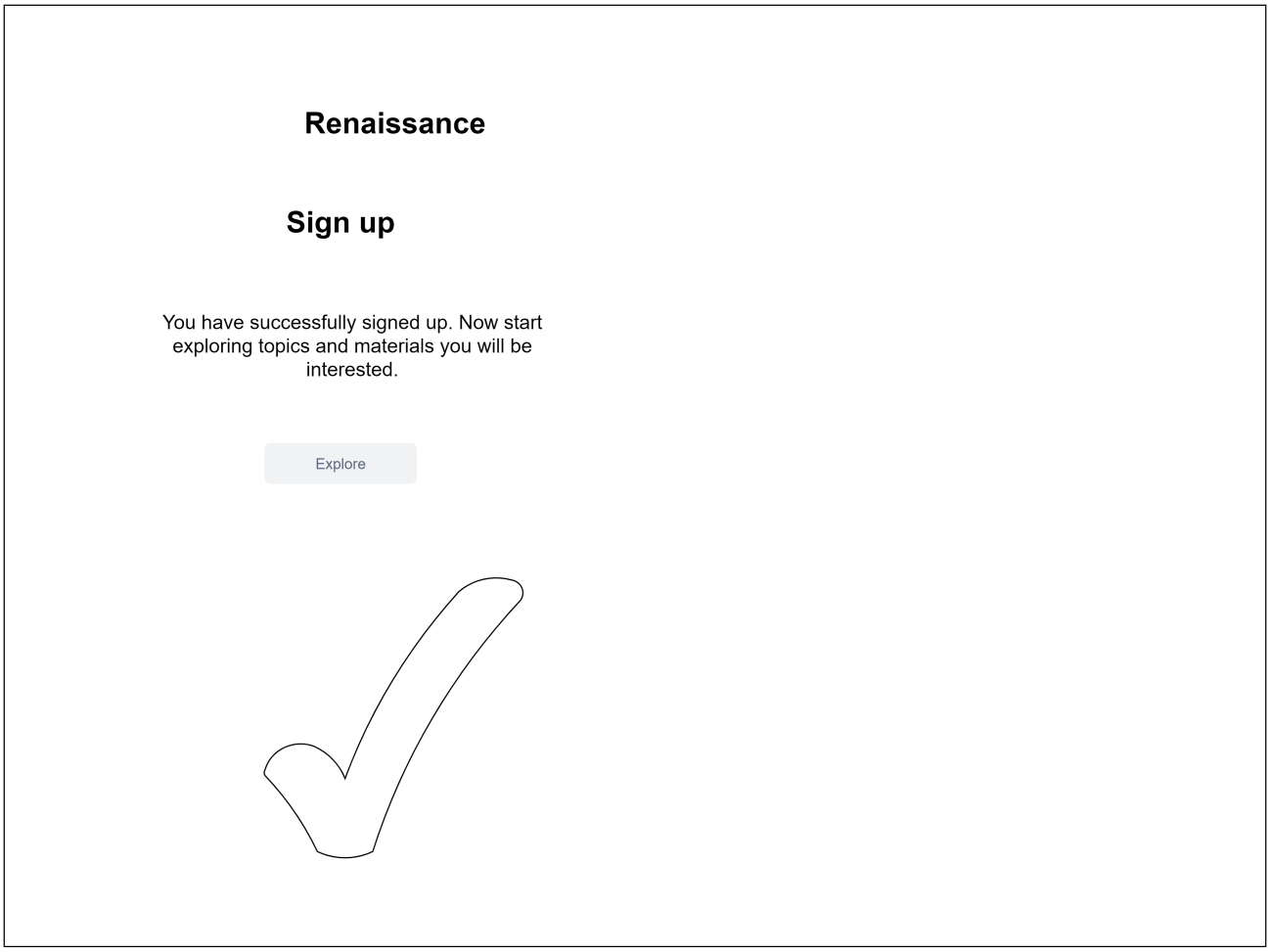
Renaissance

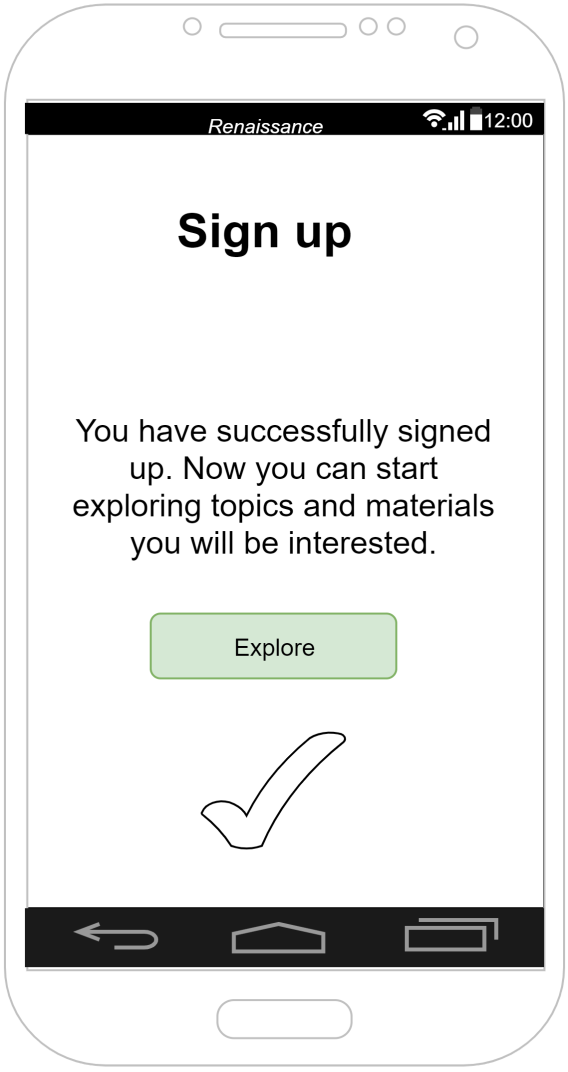
Sign up

An authentication e-mail is sent to
ovuwuewuewue@osas.ug.
Follow the link sent to complete your signing up.



4. After confirming his e-mail he is redirected to the following page.


Renaissance
Sign up
You have successfully signed up. Now start exploring topics and materials you will be interested.
Explore


Renaissance
12:00
Sign up
You have successfully signed up. Now you can start exploring topics and materials you will be interested.
Explore

5. He does not have any topics or material followed in his account. (Figure 5)

Renaissance

Home | Explore | Progress | Settings | Search topics and materials

Popular For me New

Topics

No topics found here.

Materials

Microwave Soufflé

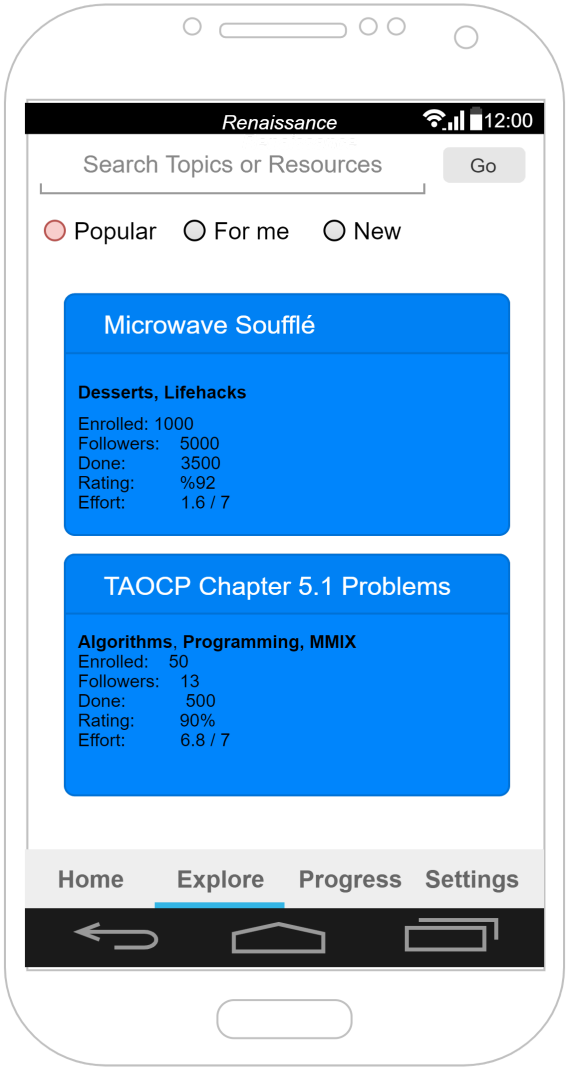
Desserts, Lifehacks

Enrolled: 1000
Followers: 5000
Done: 3500
Rating: 87%
Effort: 1.5 / 7

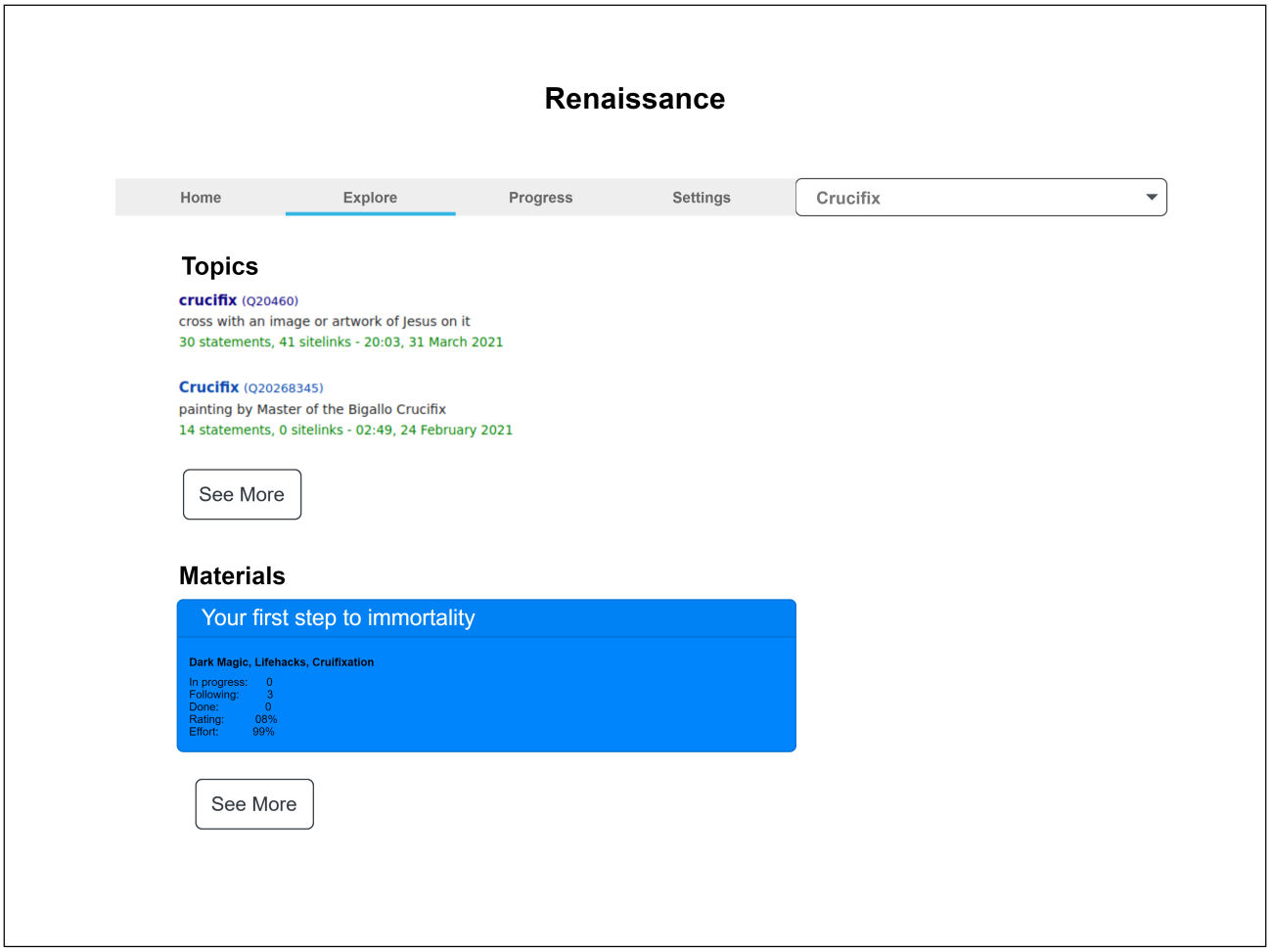
TAOCP Chapter 5.1 Problems

Algorithms, Programming, MMIX
Enrolled: 50
Followers: 3000
Done: 500
Rating: 90%
Effort: 4.3 / 7

See More



6. He searches for crucifix on the search bar on top, and he is redirected to the page in Figure 6. (Figure 6)


Renaissance
Home
Explore
Progress
Settings
Crucifix
Topics
crucifix (Q20460)
cross with an image or artwork of Jesus on it
30 statements, 41 sitelinks - 20:03, 31 March 2021
Crucifix (Q20268345)
painting by Master of the Bigallo Crucifix
14 statements, 0 sitelinks - 02:49, 24 February 2021
See More
Materials
Your first step to immortality
Dark Magic, Lifehacks, Cruifixation
In progress: 0
Following: 3
Done: 0
Rating: 08%
Effort: 99%
See More

Renaissance
12:00
Crucifix
crucifix (Q20460)
cross with an image or artwork of Jesus on it
30 statements, 41 sitelinks - 20:03, 31 March 2021
Crucifix (Q20268345)
painting by Master of the Bigallo Crucifix
14 statements, 0 sitelinks - 02:49, 24 February 2021
See More
Your first step to immortality
Dark Magic, Lifehacks, Cruifixation
Enrolled: 0
Followers: 3
Done: 0
Rating: 8%
Effort: 5.5 / 7
See More
Home
Explore
Progress
Settings

7. He clicks the first topic result and he is prompted to the page in Figure 7, which he is not satisfied with. (Figure 7)

Renaissance

Home | Explore | Progress | Settings | Crucifix

Crucifix (Q20460) See at Wikidata

Subclass of:

- Visual Artwork
- Christian Cross
- Cross

Depicts:

- Crucifixation of Jesus

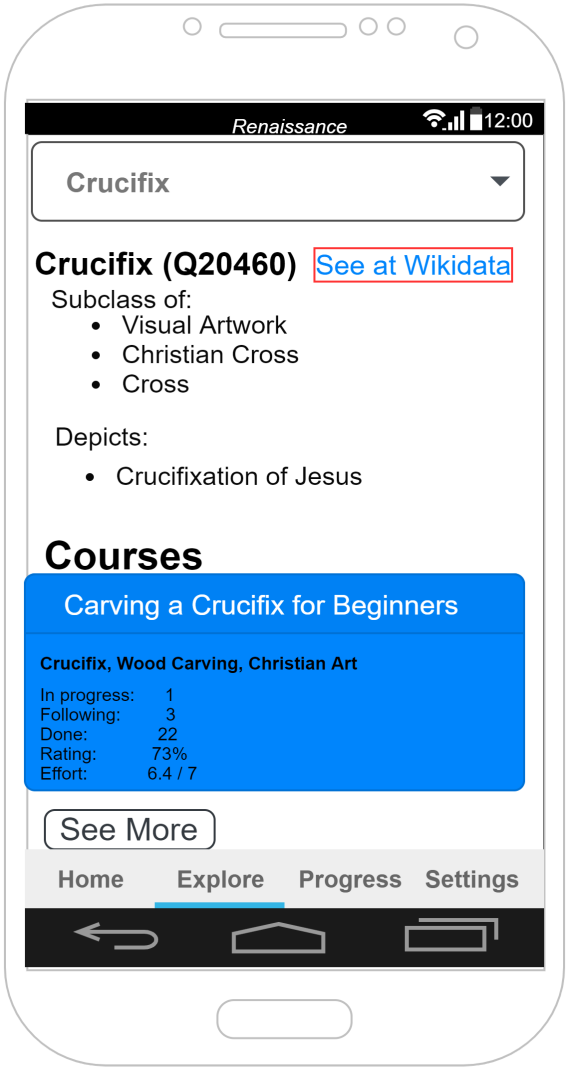
Materials

Carving a Crucifix for Begineers

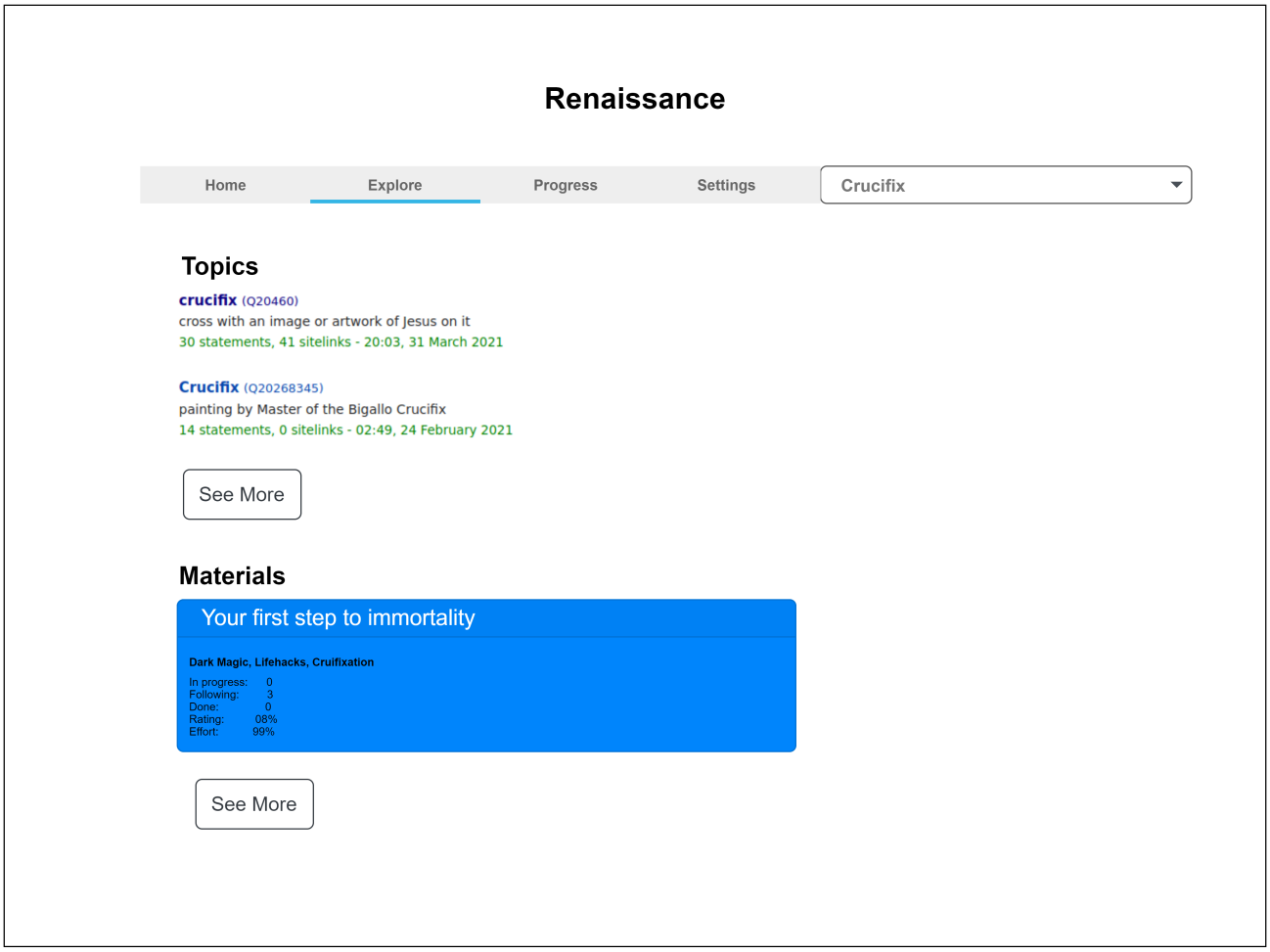
Crucifix, Wood Carving, Christian Art

In progress: 1
Following: 3
Done: 22
Rating: 73%
Effort: 62%

See More



8. So he goes back to the search results, where he sees the material in the bottom, which he finds interesting.


Renaissance
Home
Explore
Progress
Settings
Crucifix
Topics
crucifix (Q20460)
cross with an image or artwork of Jesus on it
30 statements, 41 sitelinks - 20:03, 31 March 2021
Crucifix (Q20268345)
painting by Master of the Bigallo Crucifix
14 statements, 0 sitelinks - 02:49, 24 February 2021
See More
Materials
Your first step to immortality
Dark Magic, Lifehacks, Cruifixation
In progress: 0
Following: 3
Done: 0
Rating: 08%
Effort: 99%
See More

Renaissance
12:00
Crucifix
Topics
crucifix (Q20460)
cross with an image or artwork of Jesus on it
30 statements, 41 sitelinks - 20:03, 31 March 2021
Crucifix (Q20268345)
painting by Master of the Bigallo Crucifix
14 statements, 0 sitelinks - 02:49, 24 February 2021
See More
Courses
Your first step to immortality
Dark Magic, Lifehacks, Cruifixation
Enrolled: 0
Followers: 3
Done: 0
Rating: 8%
Effort: 5.1 / 7
See More
Home
Explore
Progress
Settings

9.

Renaissance

Home | Explore | Progress | Settings | Crucifix

Your first step to immortality

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et sapnu puas dolore magna aliqua.

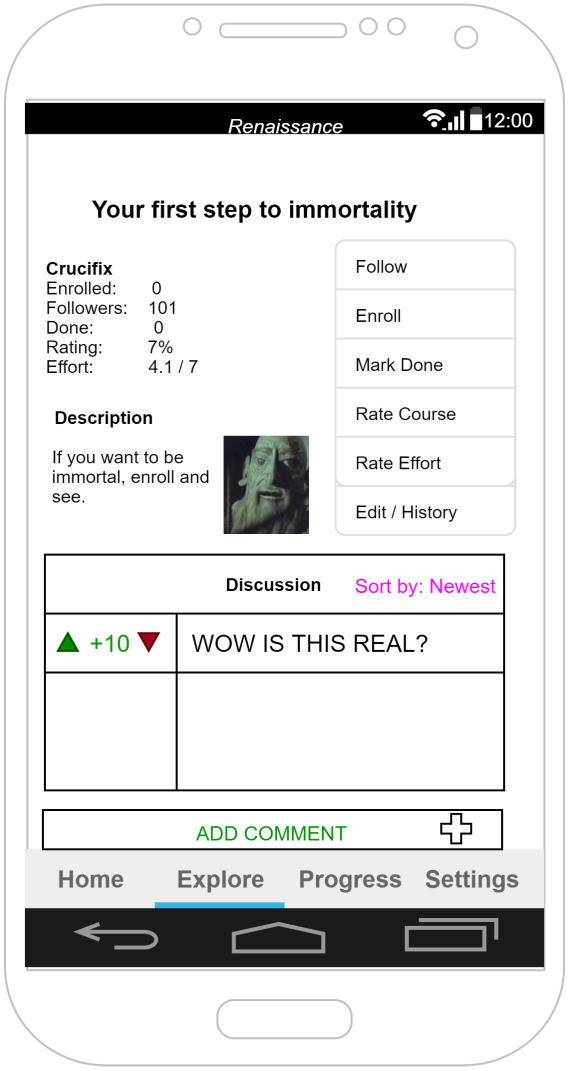
Undead

- Enrolled: 0
- Followers: 3
- Done: 0
- Rating: 7%
- Effort: 4.5 / 7

Unfollow
Enroll
Mark Done
Rate Course
Rate Effort
Edit / History

Discussion

Discussion	Sort By: Most Helpful
▲ 157 ▼	Crucifix is not a programming language.
▲ -39 ▼	Hello everyone, hope you will live forever.



Scenario 4

User Persona

- **Photo:**

- **Name:** Yolanda Graver
- **Age:** 35
- **Job:** Baker
- **Interest:** Baking, Boxing, Cleaning

Preconditions

- Yolanda is already signed up and logged in the android app.

Goals

- She wants to create a new resource including a path.

Acceptance Criteria

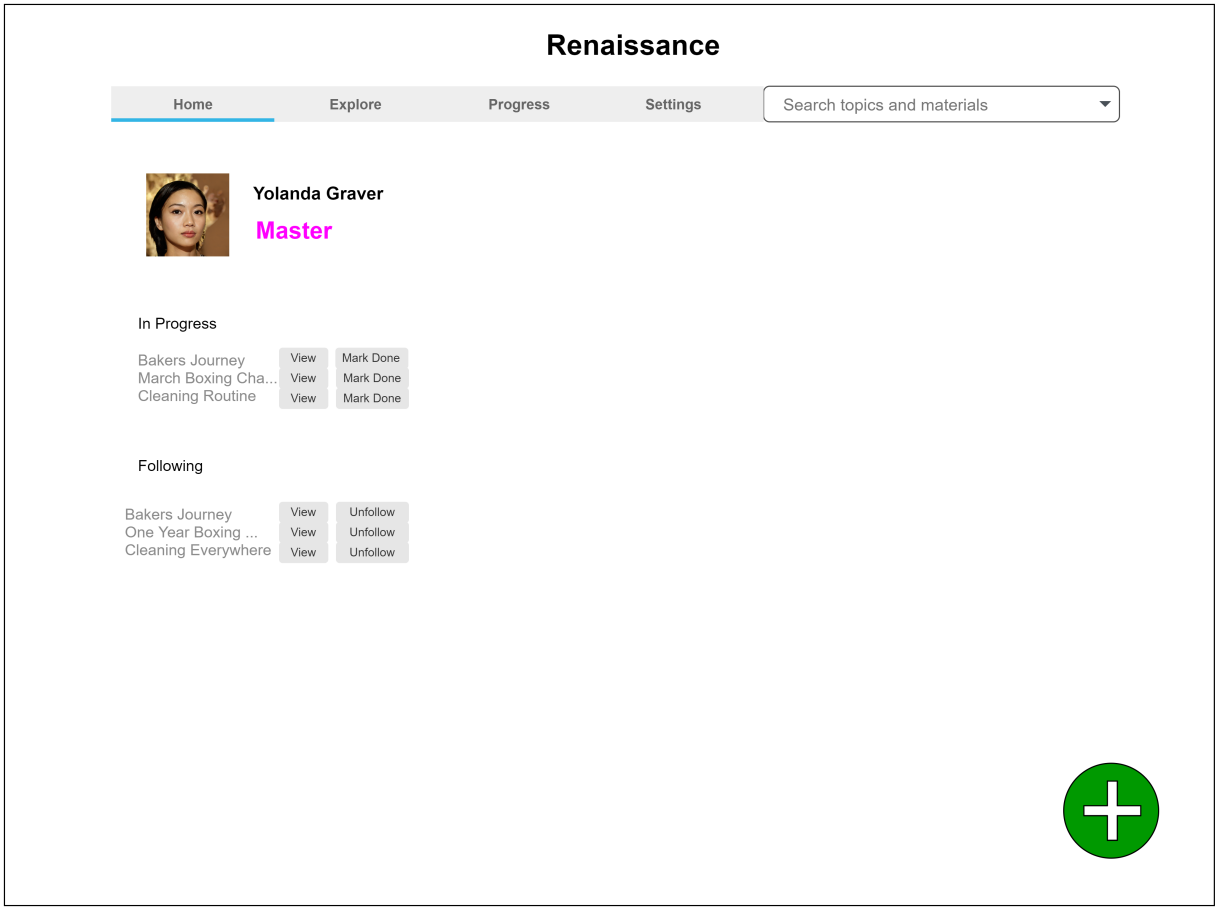
In reference to the Revision 1.1 of the requirements

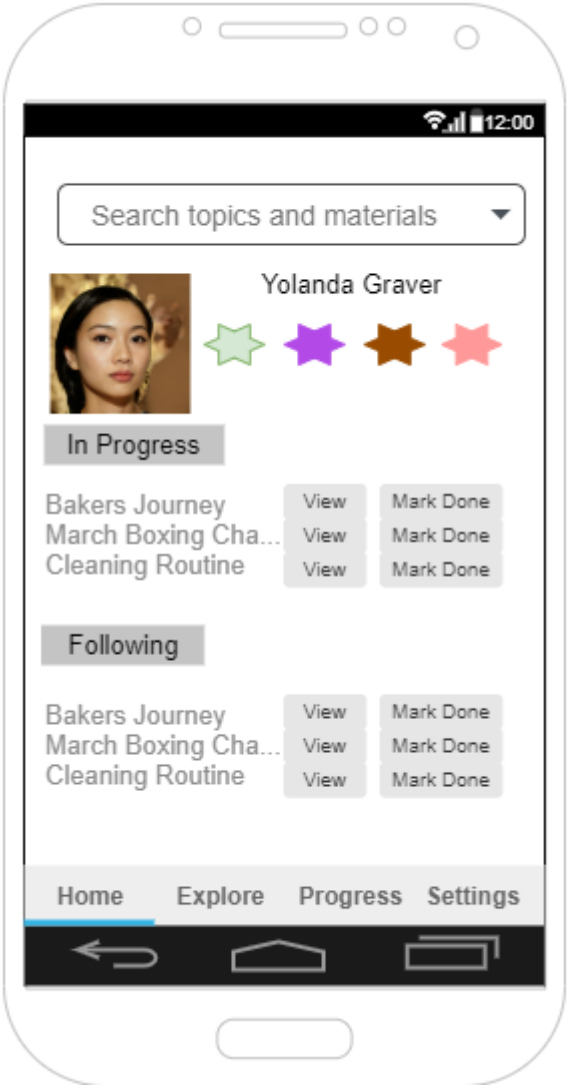
- **(Requirement 1.1.2.)** The System shall provide its users access to it by means of specialized accounts.
- **(Requirement 1.2.6.)** The users shall be able to reliably record and edit some goals they determine as a set of material-deadline pairs.

- **(Requirement 1.2.7.)** The users should be able to reliably record some topics they want to improve on.
- **(Requirement 2.1.1.)** An Android Application and a Web Application shall be implemented as instances of the Interface.

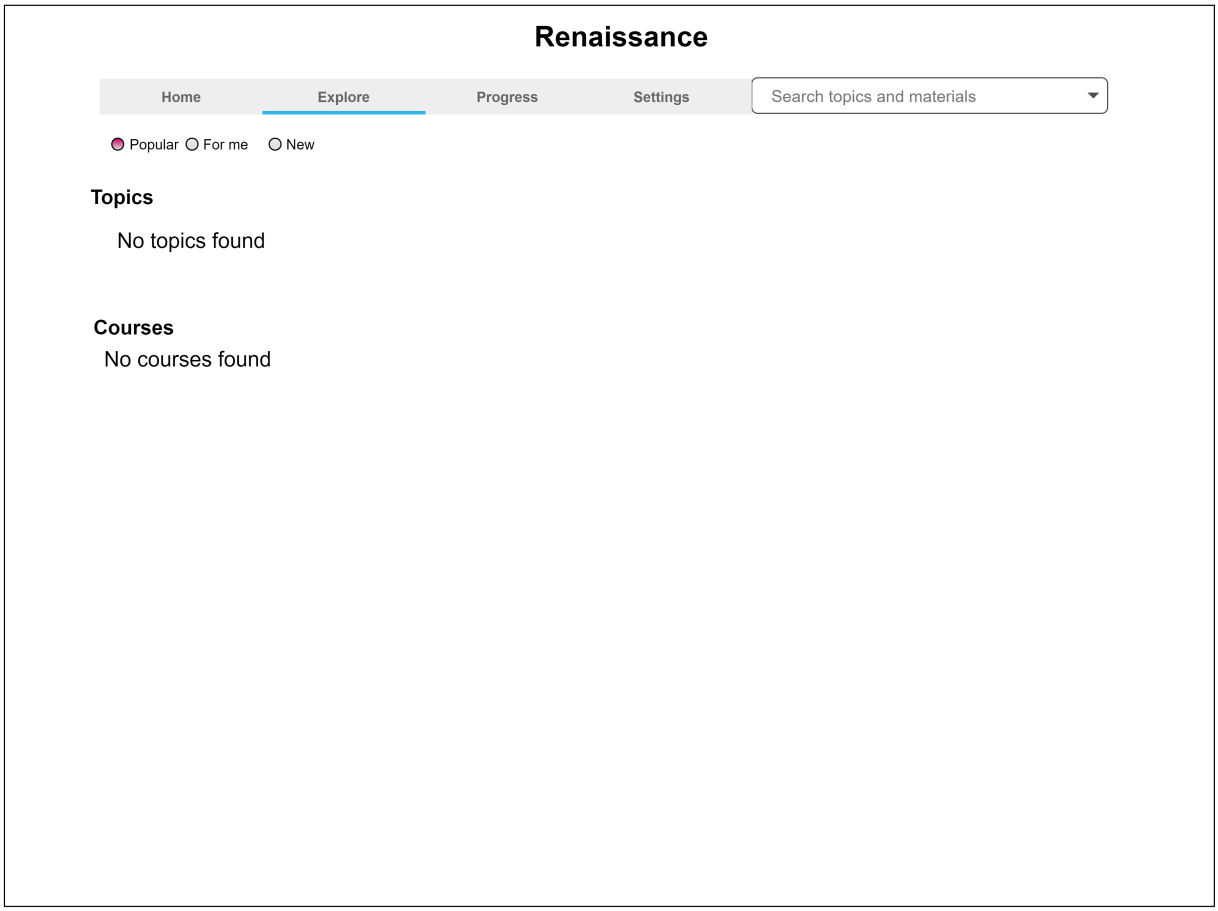
Actions & Mockups

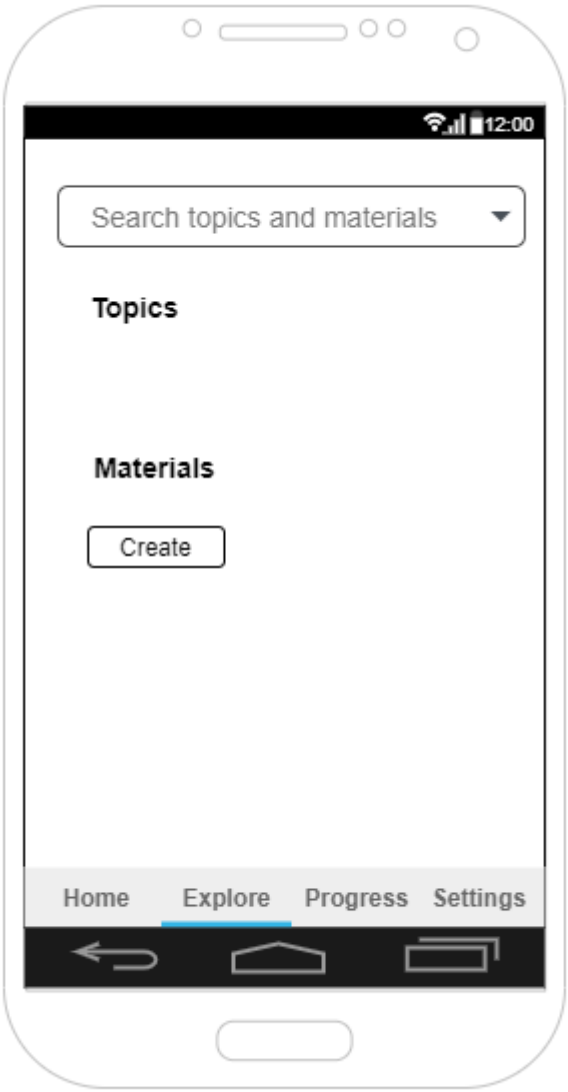
1. Yolanda opens his app and home page appears (Figure 1). She wants to create a new resource to share her mouse shaped cake recipe.


Renaissance
Home
Explore
Progress
Settings
Search topics and materials
Yolanda Graver
Master
In Progress
Bakers Journey
March Boxing Cha...
Cleaning Routine
View
Mark Done
Following
Bakers Journey
One Year Boxing ...
Cleaning Everywhere
View
Unfollow

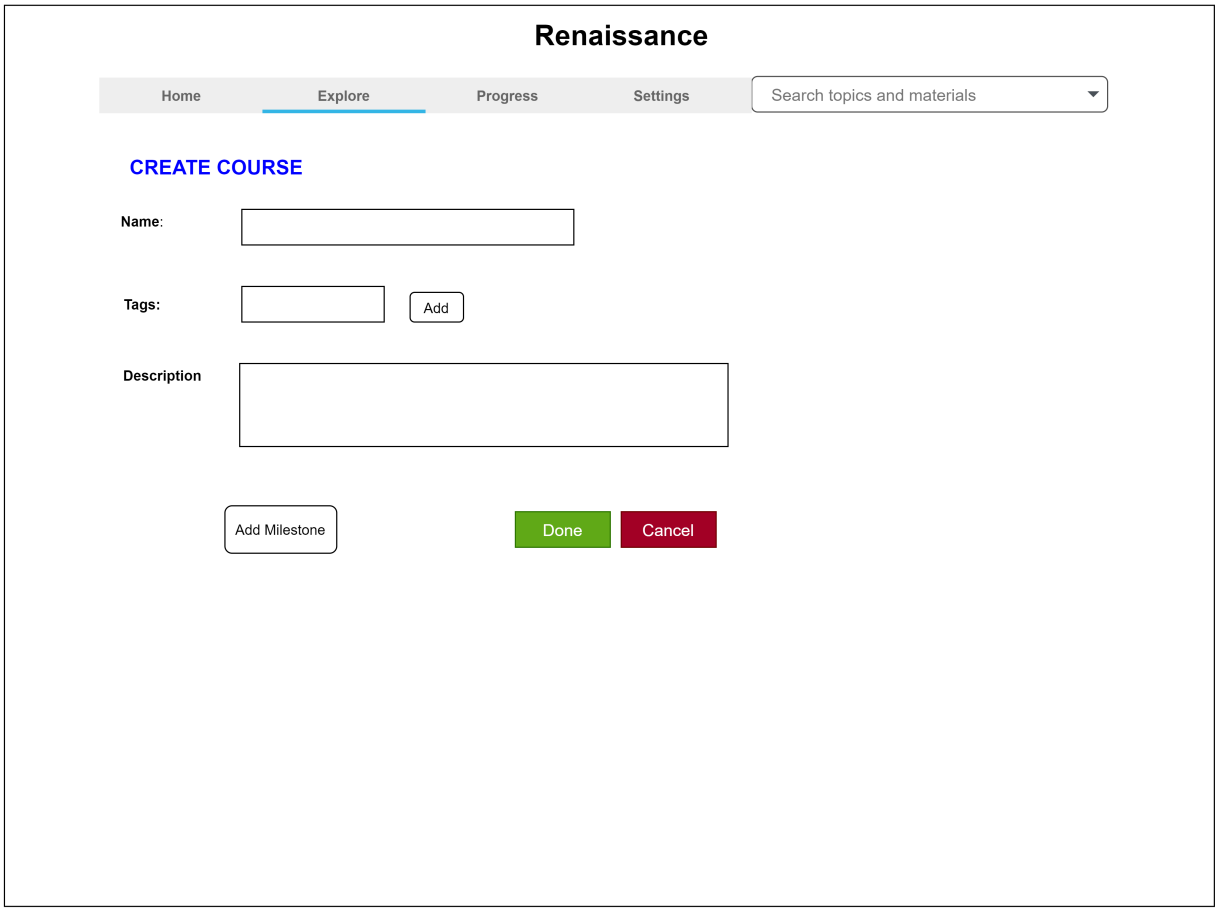

12:00
Search topics and materials
Yolanda Graver
In Progress
Bakers Journey
March Boxing Cha...
Cleaning Routine
View
Mark Done
Following
Bakers Journey
March Boxing Cha...
Cleaning Routine
View
Mark Done
Home
Explore
Progress
Settings

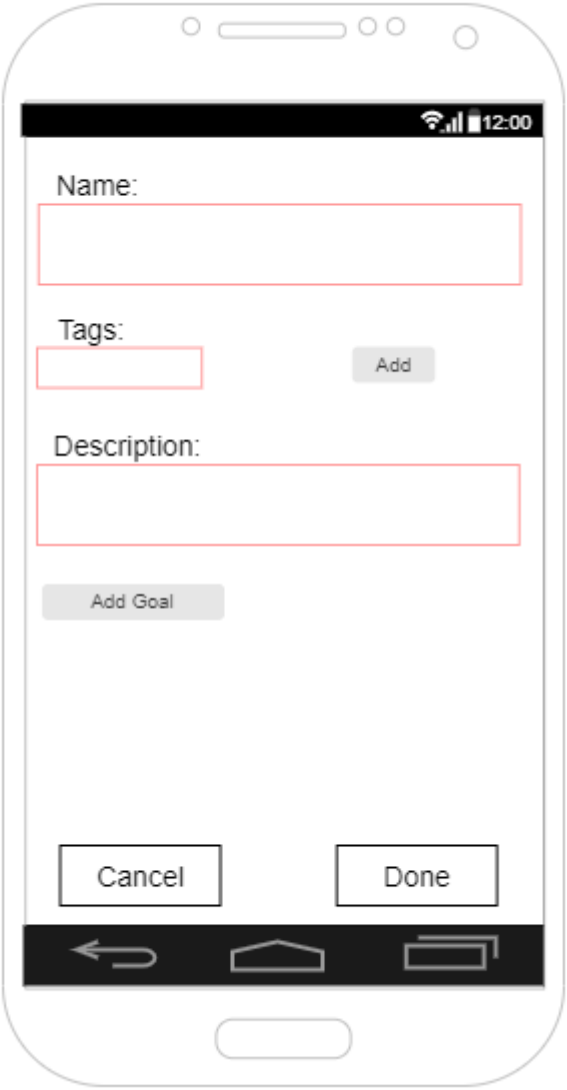
2. She goes to explore page. (Figure 2) Then she clicks on the “Create” button.




12:00
Search topics and materials
Topics
Materials
Create
Home
Explore
Progress
Settings

3. Edit page appears. (Figure 3)





4. She names the resource , adds tags and description. Then she clicks on to the “Add Milestone” button. (Figure -4)

Renaissance

Home | Explore | Progress | Settings | Search topics and materials

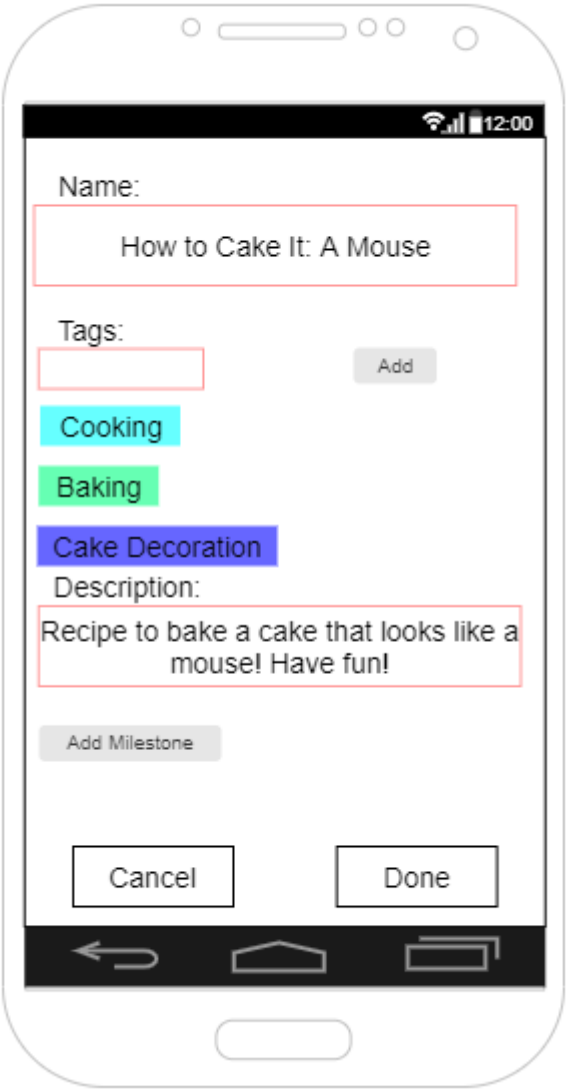
Name: How to Cake It : A Mouse

Tags: Add

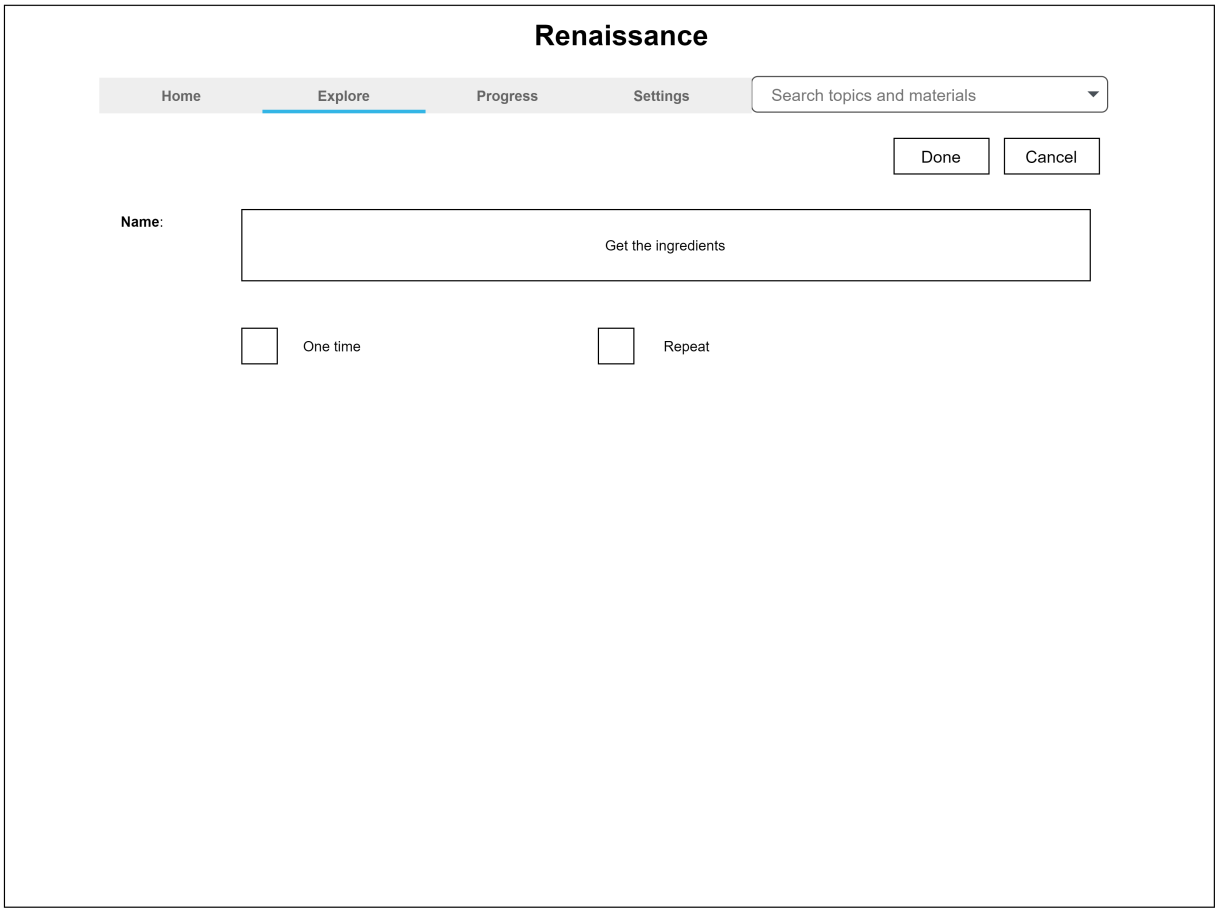
Description Recipe to bake a cake that looks like a mouse! Have fun!

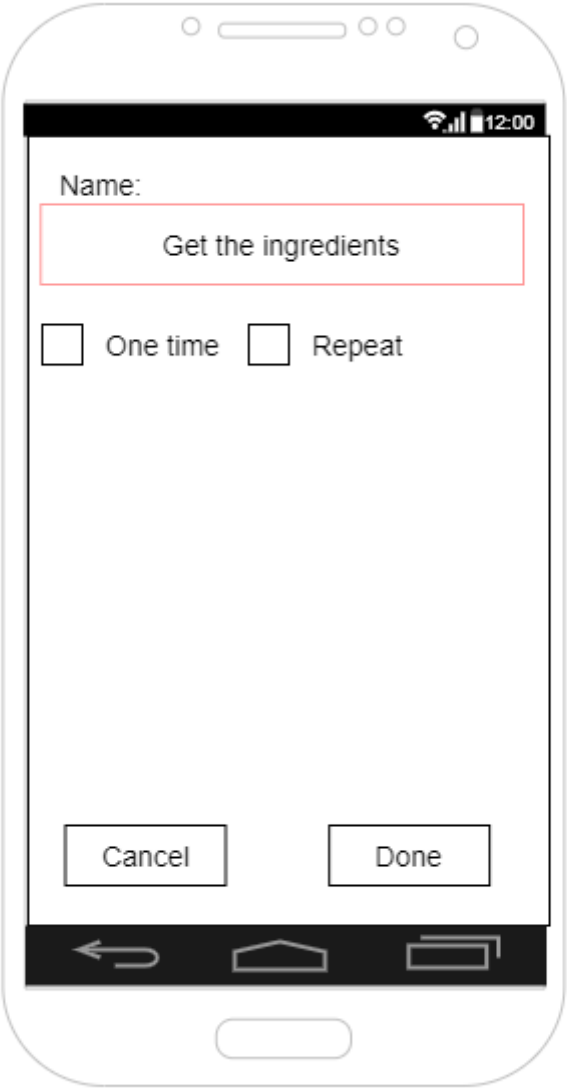
Add Milestone

Done Cancel

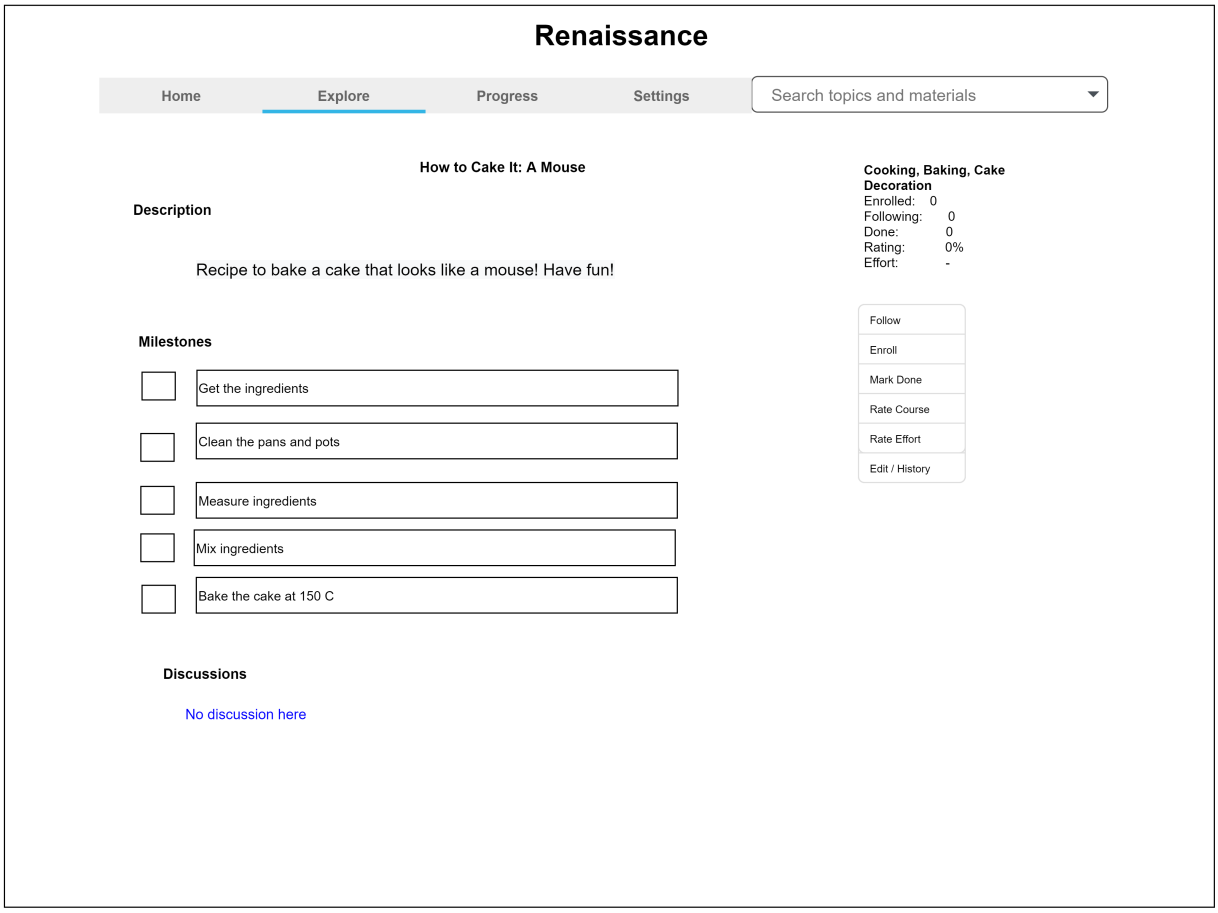

12:00
Name:
How to Cake It: A Mouse
Tags:
Add
Cooking
Baking
Cake Decoration
Description:
Recipe to bake a cake that looks like a mouse! Have fun!
Add Milestone
Cancel
Done

5. She adds the new milestone, sets it recurring type and clicks done. (Figure-5)




12:00
Name:
Get the ingredients
One time
Repeat
Cancel
Done

6. Final resource page is as shown below.



Scenario 5

User Persona

- **Photo:**

- **Name:** John Jones
- **Age:** 50

- **Job:** Teacher
- **Interest:** Dance, Group Activities

Preconditions

- John is already signed up and logged in to the android and web app.
- The app is started at in-progress goals.

Goals

- John tries to learn dancing. He wants his friend to learn dancing, so they can dance after the pandemic together.

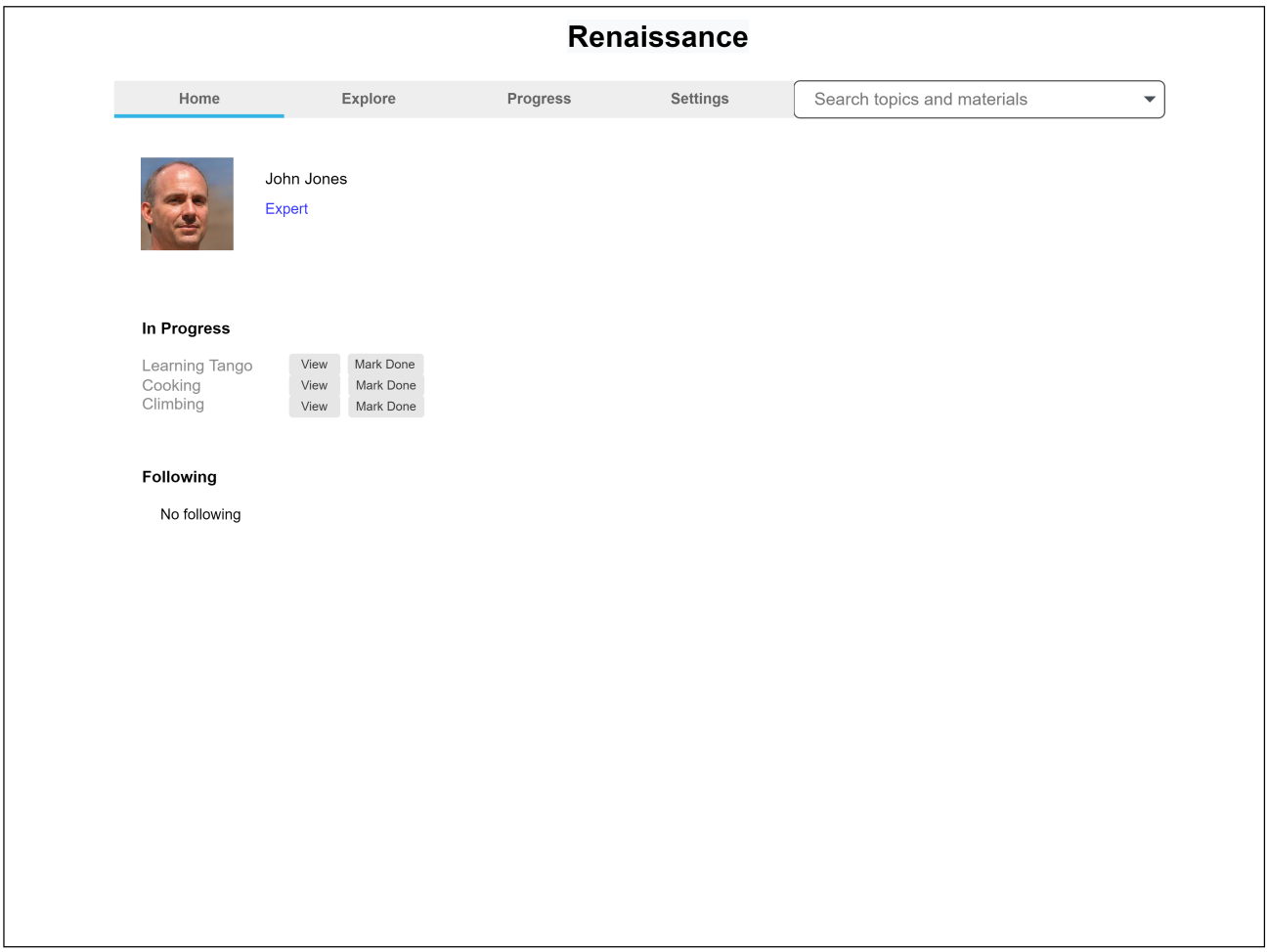
Acceptance Criteria

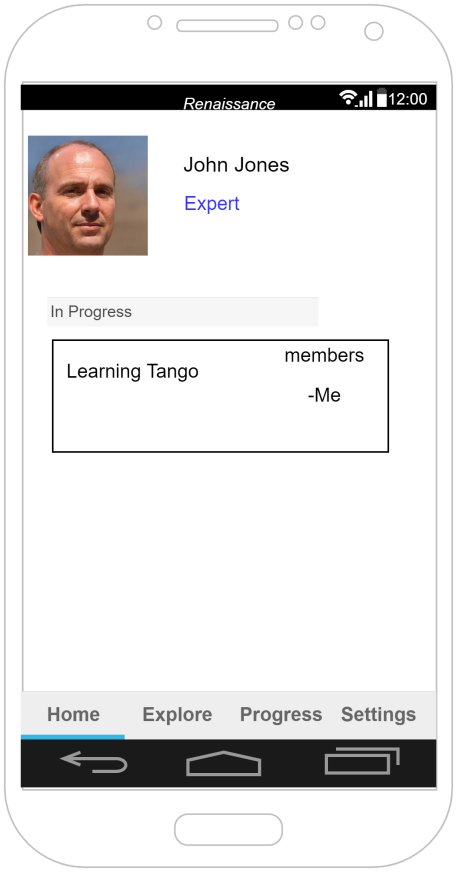
In reference to Revision 1.1 of the requirements

- **(Requirement 1.2.8.)** The users *shall* be able to work on their determined goals as a group.
- **(Requirement 1.3.4.)** The users *shall* be able to search for other users.
- **(Requirement 2.1.1.)** An Android Application and a Web Application shall be implemented as instances of the Interface.

Actions & Mockups

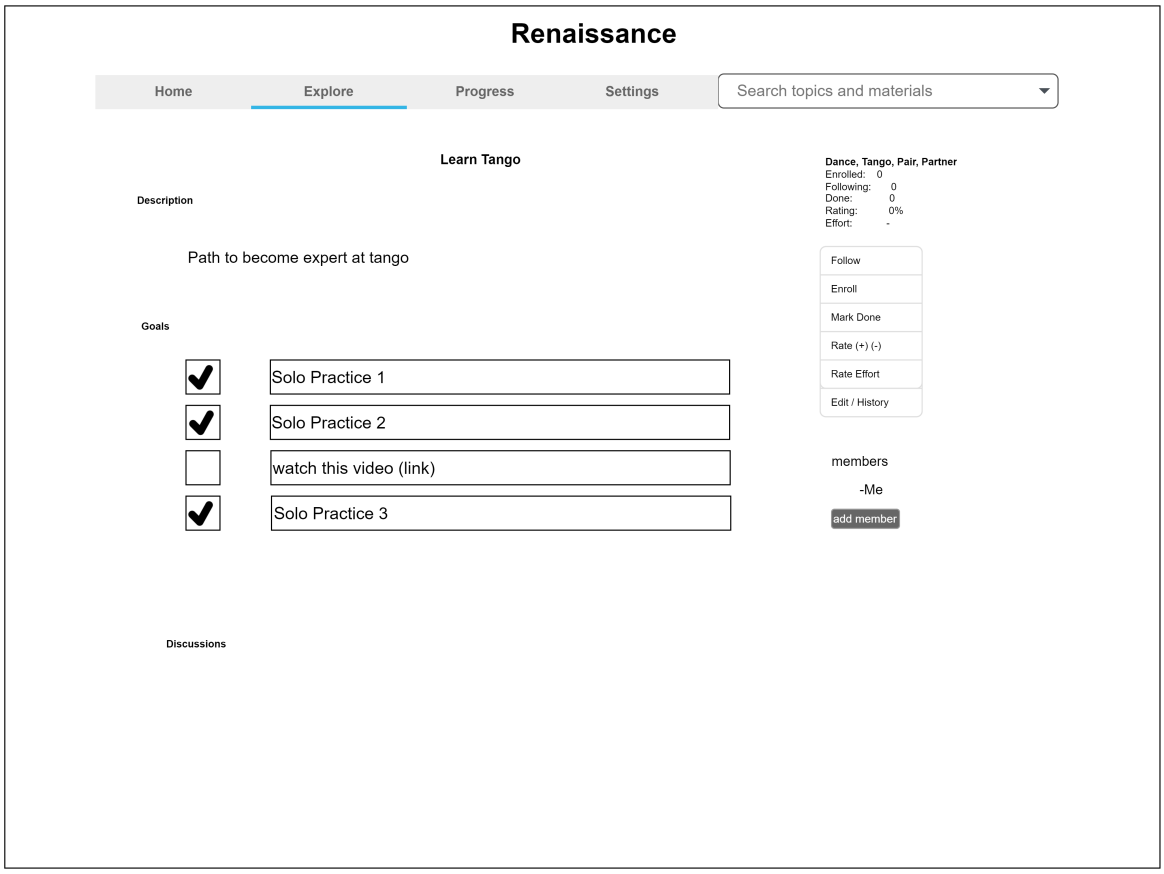
1. John opens the app and sees in-progress goals, and opens one relating to tango.

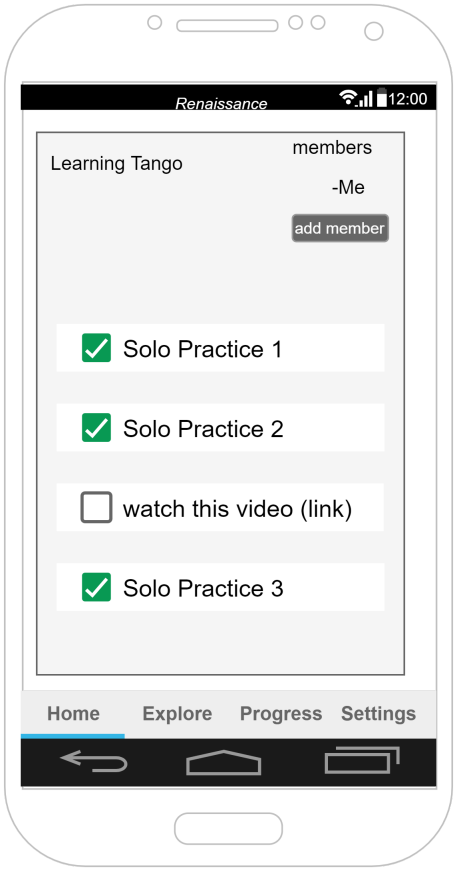




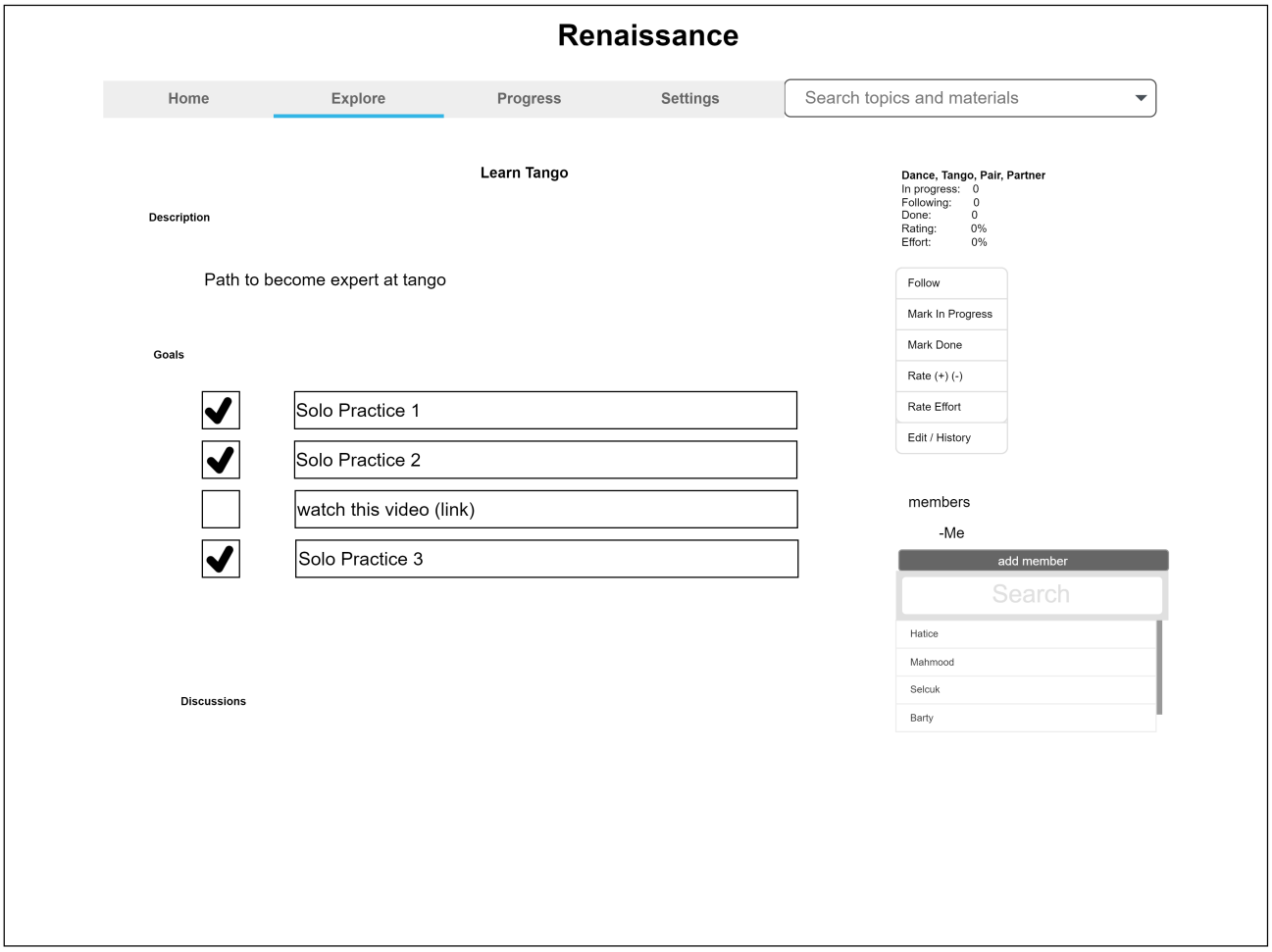
2. He presses add member button to add his friend Hatice

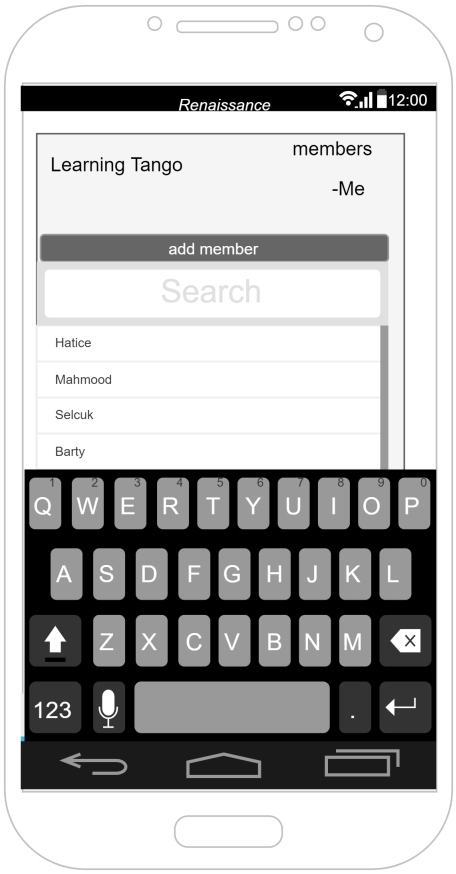
2



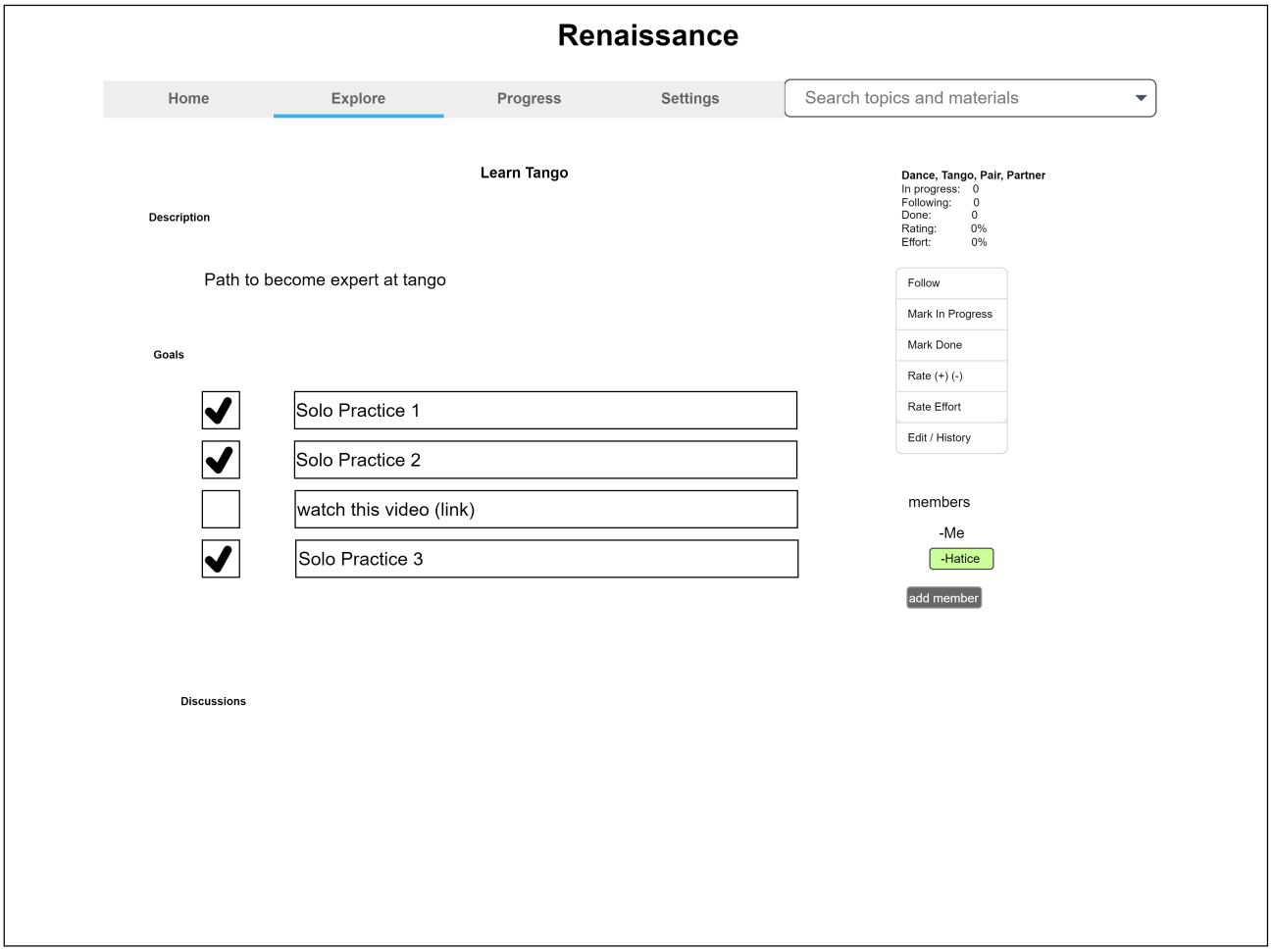


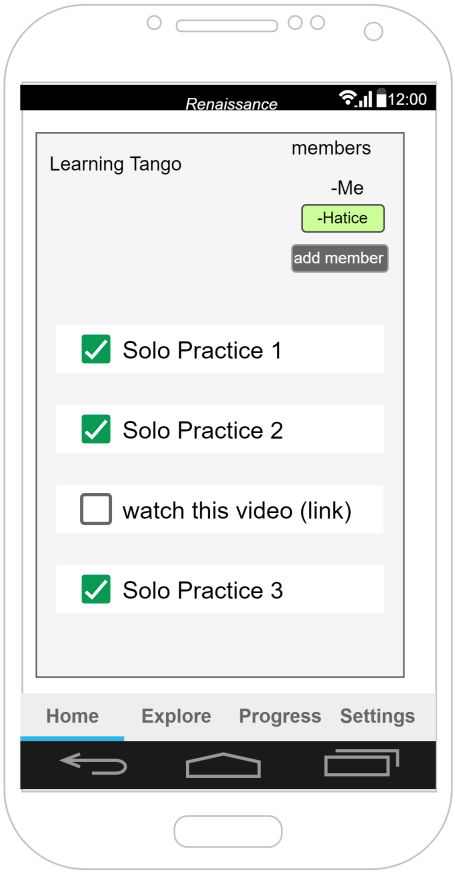
3. He searches, finds the friend and selects it.



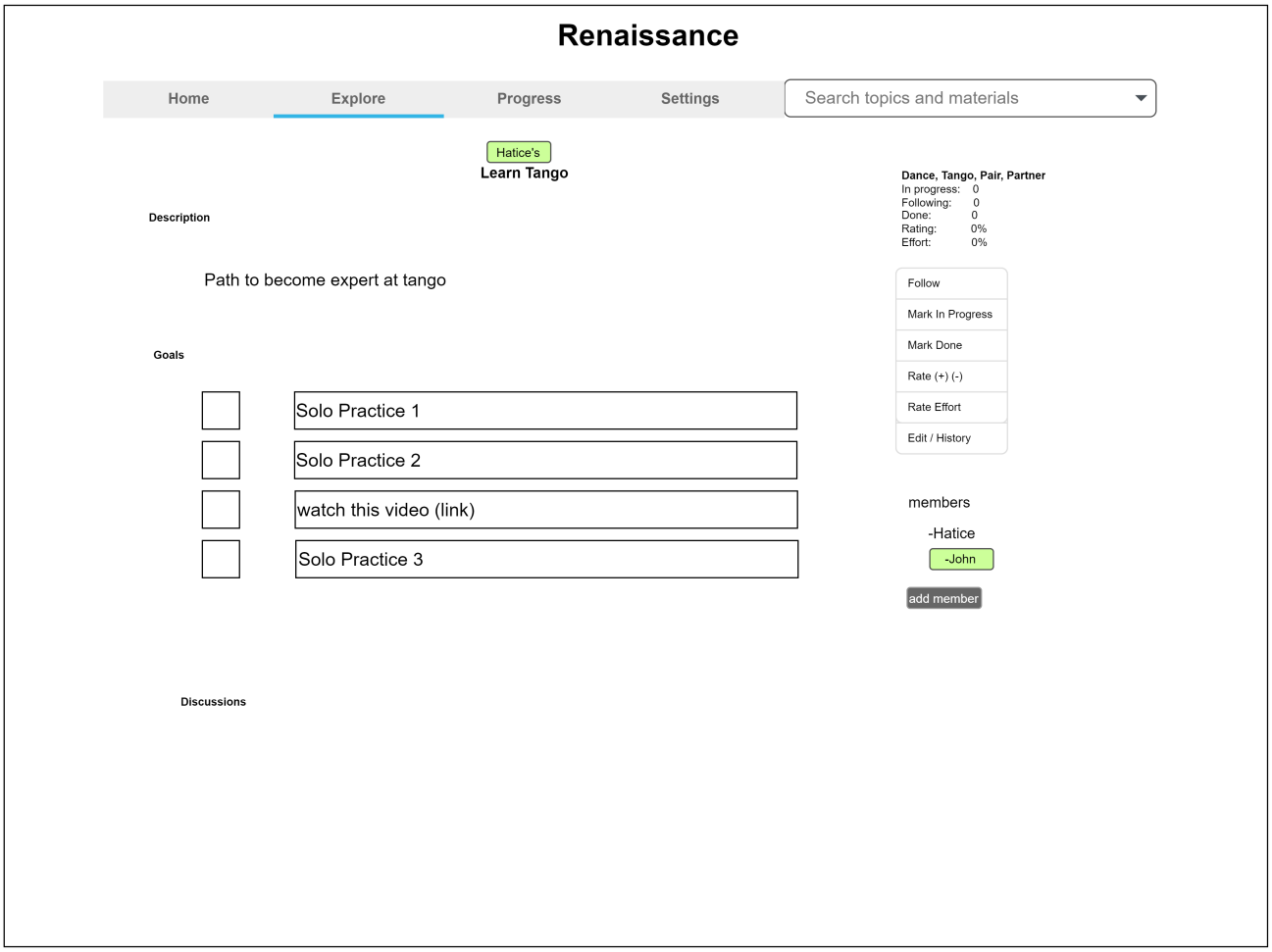


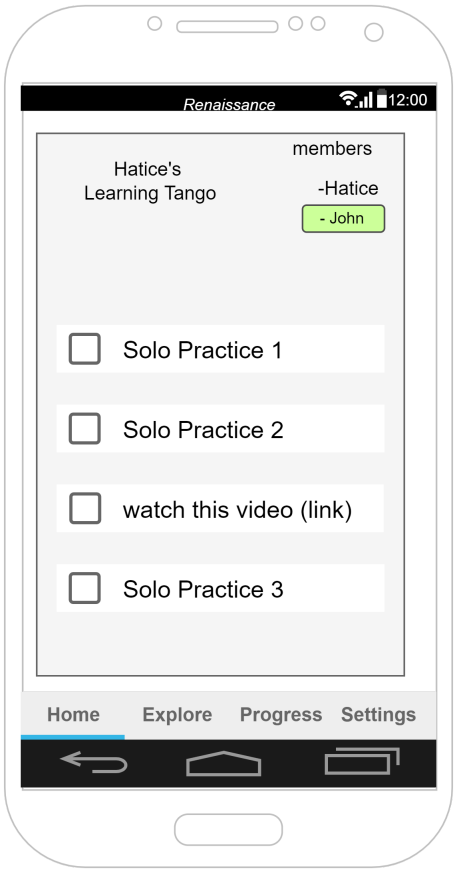
4. The new member added to this path, he can see members now.





5. He goes to his friend's path to see her progress





Project plan

	Name	Duration	Start	Finish	Predecessors	Resource Names
	Review And Update Project Material	**7 days?**	**10/12/21 8:00 AM**	**10/18/21 5:00 PM**		
	Reqiurements	7 days?	10/12/21 8:00 AM	10/18/21 5:00 PM		Fikri Cem Yılmaz;Selman Berk Özkurt
	Mockups	7 days?	10/12/21 8:00 AM	10/18/21 5:00 PM		Fikri Cem Yılmaz;Meltem Arslan;Kürşat Bakış
	Design	7 days?	10/12/21 8:00 AM	10/18/21 5:00 PM		Kürşat Bakış;Meltem Arslan;Fikri Cem Yılmaz;Muhammed ...
	Project Plan	7 days?	10/12/21 8:00 AM	10/18/21 5:00 PM		Eren Altunoğlu;Bekir Tatar
	Determine the developer team roles	3 days?	10/13/21 8:00 AM	10/15/21 5:00 PM		Each Team Member
	Backend 1	**15 days?**	**10/19/21 8:00 AM**	**11/2/21 5:00 PM**	**67**	**Backend Team**
	Decide on API and Database technologies	4 days?	10/19/21 8:00 AM	10/22/21 5:00 PM		Backend Team
	Configure database for users	11 days?	10/19/21 8:00 AM	11/2/21 5:00 PM		Eren Altunoğlu
	API for Signup - Login	5 days?	10/19/21 8:00 AM	10/25/21 5:00 PM		Selman Berk Özkurt
	API for Delete User	6 days?	10/25/21 5:00 PM	11/2/21 5:00 PM		Mehmet Yasin Şeremet
	Admin user functionalities	11 days?	10/19/21 8:00 AM	11/2/21 5:00 PM		Eren Altunoğlu
	API for User Search	6 days?	10/25/21 5:00 PM	11/2/21 5:00 PM		Selman Berk Özkurt
	Frontend 1	**15 days?**	**10/19/21 8:00 AM**	**11/2/21 5:00 PM**	**67**	**Frontend Team**
	Decide on Frontend Technologies	4 days?	10/19/21 8:00 AM	10/22/21 5:00 PM		Frontend Team
	Initial version of home page	11 days?	10/19/21 8:00 AM	11/2/21 5:00 PM		Meltem Arslan
	Signup - Login	5 days?	10/19/21 8:00 AM	10/25/21 5:00 PM		Bekir Tatar
	Initial version of profile page	5 days?	10/19/21 8:00 AM	10/25/21 5:00 PM		Muhammed Emin Deniz
	User Search	7 days?	10/25/21 8:00 AM	11/2/21 5:00 PM		Bekir Tatar
	Navigation Bar	7 days?	10/25/21 8:00 AM	11/2/21 5:00 PM		Muhammed Emin Deniz
	Android 1	**15 days?**	**10/19/21 8:00 AM**	**11/2/21 5:00 PM**	**67**	**Android Team**
	Decide on Mobile Technologies	4 days?	10/19/21 8:00 AM	10/22/21 5:00 PM		Android Team
	Initial version of home page	11 days?	10/19/21 8:00 AM	11/2/21 5:00 PM		Fikri Cem Yılmaz
	Sign up - Login	5 days?	10/19/21 8:00 AM	10/25/21 5:00 PM		Hazar Çakır
	Initial version of Profile page	5 days?	10/19/21 8:00 AM	10/25/21 5:00 PM		Kürşat Bakış
	User Search	6 days?	10/25/21 5:00 PM	11/2/21 5:00 PM		Hazar Çakır
	Navigation Bar	6 days?	10/25/21 5:00 PM	11/2/21 5:00 PM		Kürşat Bakış
	Milestone 3 report	**13 days?**	**11/3/21 8:00 AM**	**11/15/21 5:00 PM**	**76;83;90**	**Each Team Member**
	Review on initial stage of application	8 days?	11/3/21 8:00 AM	11/10/21 5:00 PM		Each Team Member
	Connecting backend with frontend and android	8 days?	11/3/21 8:00 AM	11/10/21 5:00 PM		Android Team;Backend Team;Frontend Team
	Initial deployment of application	9 days?	11/3/21 8:00 AM	11/11/21 5:00 PM		Frontend Team;Backend Team;Android Team
	Testing	9 days?	11/3/21 8:00 AM	11/15/21 5:00 PM		Each Team Member
	Delivarables	13 days?	11/3/21 8:00 AM	11/15/21 5:00 PM		Each Team Member
	Report	13 days?	11/3/21 8:00 AM	11/15/21 5:00 PM		Each Team Member
	Backend 2	**18 days?**	**11/16/21 8:00 AM**	**12/3/21 5:00 PM**	**97**	**Backend Team**
	Configure database for resources and topics	6 days?	11/16/21 8:00 AM	11/23/21 5:00 PM		Eren Altunoğlu
	Wikidata integration	6 days?	11/16/21 8:00 AM	11/23/21 5:00 PM		Selman Berk Özkurt
	Follow Topics and Resources	6 days?	11/16/21 8:00 AM	11/23/21 5:00 PM		Eren Altunoğlu
	Provide resource	6 days?	11/16/21 8:00 AM	11/23/21 5:00 PM		Selman Berk Özkurt
	Consume resource	6 days?	11/16/21 8:00 AM	11/23/21 5:00 PM		Selman Berk Özkurt
	Search topics and resources	6 days?	11/16/21 8:00 AM	11/23/21 5:00 PM		Eren Altunoğlu
	Notification services	8 days?	11/23/21 5:00 PM	12/3/21 5:00 PM		Eren Altunoğlu
	Rate material	8 days?	11/23/21 5:00 PM	12/3/21 5:00 PM		Mehmet Yasin Şeremet
	User profile editing	8 days?	11/23/21 5:00 PM	12/3/21 5:00 PM		Mehmet Yasin Şeremet
	Comments	8 days?	11/23/21 5:00 PM	12/3/21 5:00 PM		Mehmet Yasin Şeremet
	Frontend 2	**18 days?**	**11/16/21 8:00 AM**	**12/3/21 5:00 PM**	**97**	**Frontend Team**
	Resource and topic pages	14 days?	11/16/21 8:00 AM	12/3/21 5:00 PM		Bekir Tatar
	Editing profile	14 days?	11/16/21 8:00 AM	12/3/21 5:00 PM		Muhammed Emin Deniz
	Search topics and resources	14 days?	11/16/21 8:00 AM	12/3/21 5:00 PM		Muhammed Emin Deniz
	Follow topics and resources	14 days?	11/16/21 8:00 AM	12/3/21 5:00 PM		Muhammed Emin Deniz
	Provide resource	14 days?	11/16/21 8:00 AM	12/3/21 5:00 PM		Bekir Tatar
	Consume resource	14 days?	11/16/21 8:00 AM	12/3/21 5:00 PM		Bekir Tatar
	Notification integration	14 days?	11/16/21 8:00 AM	12/3/21 5:00 PM		Meltem Arslan
	Comments	14 days?	11/16/21 8:00 AM	12/3/21 5:00 PM		Meltem Arslan
	Effort and ratings	14 days?	11/16/21 8:00 AM	12/3/21 5:00 PM		Meltem Arslan
	Android 2	**18 days?**	**11/16/21 8:00 AM**	**12/3/21 5:00 PM**	**97**	**Android Team**
	Resource and topic pages	14 days?	11/16/21 8:00 AM	12/3/21 5:00 PM		Fikri Cem Yılmaz
	Editing profile	14 days?	11/16/21 8:00 AM	12/3/21 5:00 PM		Hazar Çakır
	Search topics and resources	14 days?	11/16/21 8:00 AM	12/3/21 5:00 PM		Hazar Çakır
	Follow Topics and Resources	14 days?	11/16/21 8:00 AM	12/3/21 5:00 PM		Hazar Çakır
	Provide resource	14 days?	11/16/21 8:00 AM	12/3/21 5:00 PM		Fikri Cem Yılmaz
	Consume resource	14 days?	11/16/21 8:00 AM	12/3/21 5:00 PM		Fikri Cem Yılmaz
	Notification integration	14 days?	11/16/21 8:00 AM	12/3/21 5:00 PM		Kürşat Bakış
	Comments	14 days?	11/16/21 8:00 AM	12/3/21 5:00 PM		Kürşat Bakış
	Effort and ratings	14 days?	11/16/21 8:00 AM	12/3/21 5:00 PM		Kürşat Bakış
	Milestone 4 report	**7 days?**	**12/4/21 8:00 AM**	**12/10/21 5:00 PM**	**104;115;125**	**Each Team Member**
	Review on current version of application	4 days?	12/4/21 8:00 AM	12/9/21 5:00 PM		Each Team Member
	Second deployment	5.08 days?	12/4/21 8:00 AM	12/9/21 8:38 AM		Frontend Team;Backend Team;Android Team
	Testing	7 days?	12/4/21 8:00 AM	12/10/21 5:00 PM		Android Team;Backend Team;Frontend Team

Delivarables	5 days?	12/4/21 8:00 AM	12/10/21 5:00 PM		Each Team Member
Report	5 days?	12/4/21 8:00 AM	12/10/21 5:00 PM		Each Team Member
Backend 3	**20 days?**	**12/11/21 8:00 AM**	**12/30/21 5:00 PM**	**135**	**Backend Team**
Recommendation of materials	14 days?	12/11/21 8:00 AM	12/30/21 5:00 PM		Eren Altunoğlu
Potentially interested topics algorithm	14 days?	12/11/21 8:00 AM	12/30/21 5:00 PM		Eren Altunoğlu
Tracking Progress for all goal types	14 days?	12/11/21 8:00 AM	12/30/21 5:00 PM		Selman Berk Özkurt
Colective goal integration	14 days?	12/11/21 8:00 AM	12/30/21 5:00 PM		Selman Berk Özkurt
Scheduled goal integration	14 days?	12/11/21 8:00 AM	12/30/21 5:00 PM		Mehmet Yasin Şeremet
Path goal integration	14 days?	12/11/21 8:00 AM	12/30/21 5:00 PM		Mehmet Yasin Şeremet
Frontend 3	**20 days?**	**12/11/21 8:00 AM**	**12/30/21 5:00 PM**	**135**	**Frontend Team**
Explore page	14 days?	12/11/21 8:00 AM	12/30/21 5:00 PM		Muhammed Emin Deniz
Personal page - past efforts	14 days?	12/11/21 8:00 AM	12/30/21 5:00 PM		Muhammed Emin Deniz
Tracking progress	14 days?	12/11/21 8:00 AM	12/30/21 5:00 PM		Meltem Arslan
Colective goal integration	14 days?	12/11/21 8:00 AM	12/30/21 5:00 PM		Meltem Arslan
Scheduled goal integration	14 days?	12/11/21 8:00 AM	12/30/21 5:00 PM		Bekir Tatar
Path goal integration	14 days?	12/11/21 8:00 AM	12/30/21 5:00 PM		Bekir Tatar
Share accomplishments and topics	14 days?	12/11/21 8:00 AM	12/30/21 5:00 PM		Bekir Tatar
Mobil 3	**20 days?**	**12/11/21 8:00 AM**	**12/30/21 5:00 PM**	**135**	**Android Team**
Explore page	14 days?	12/11/21 8:00 AM	12/30/21 5:00 PM		Kürşat Bakış
Personal page - past efforts	14 days?	12/11/21 8:00 AM	12/30/21 5:00 PM		Kürşat Bakış
Tracking progress	14 days?	12/11/21 8:00 AM	12/30/21 5:00 PM		Hazar Çakır
Colective goal integration	14 days?	12/11/21 8:00 AM	12/30/21 5:00 PM		Hazar Çakır
Scheduled goal integration	14 days?	12/11/21 8:00 AM	12/30/21 5:00 PM		Fikri Cem Yılmaz
Path goal integration	14 days?	12/11/21 8:00 AM	12/30/21 5:00 PM		Fikri Cem Yılmaz
Share accomplishments and topics	14 days?	12/11/21 8:00 AM	12/30/21 5:00 PM		Fikri Cem Yılmaz
Milestone 5	**6 days?**	**12/31/21 8:00 AM**	**1/5/22 5:00 PM**	**141;148;156**	**Each Team Member**
Review on current version of application	3 days?	12/31/21 8:00 AM	1/4/22 5:00 PM		Each Team Member
Final deployment	5 days?	12/31/21 8:00 AM	1/4/22 5:00 PM		Frontend Team;Backend Team;Android Team
Testing	6 days?	12/31/21 8:00 AM	1/5/22 5:00 PM		Android Team;Backend Team;Frontend Team
Delivarables	4 days?	12/31/21 8:00 AM	1/5/22 5:00 PM		Each Team Member
Report	4 days?	12/31/21 8:00 AM	1/5/22 5:00 PM		Each Team Member

Communication Plan

Name	Purpose	Platform	Time
Assignment Meetings	After the lecture, distributing the assignments to the members accordingly.	Zoom	Every Tuesday at 15:00 GMT+3 (at LAB hours)
Review Meetings	Before the lecture, checking out what each of us have done, since the last meeting.	Zoom	Every Sunday at 22:00 GMT+3
Instant Messaging	Helping each other with simple stuff like how to write a for loop 💯. Also, micro managing without meetings is important.	Slack	Anytime

Name	Purpose	Platform	Time
Emergency Messaging	Emergency communication alternative.	Whatsapp	Anytime
Out of Context Messaging	To disscuss any topic that doesn't concern the project.	Whatsapp	Anytime

RAM

Detailed version can be found on [wiki]

L - Lead C - Contributor R - Reviewer A - Approval N - None	Bekir Tatar	Eren Altunoğlu	Fikri Cem Yılmaz	Hazar Çakır	Kürşat Bakış	Mehmet Yasin Şeremet	Meltem Arslan	Muhammed Emin Deniz	Selman Berk Özkurt
Documentation									
Updating Meeting Notes Template	N	N	N	C	N	N	C	R	N
Updating Communication Plan Page	N	N	N	R	N	N	R	N	N
Updating Project Plan	L	C	R	R	R	R	R	R	R
Organizing Slack Channels	N	N	N	A	L	N	C	R	N
Creating own wiki page (for the new team members)	R	N	N	L	L	N	L	N	N
Updating own wiki page (for the existing team members)	C	N	C	N	N	N	N	C	C
Updating Group Wiki	N	N	N	C	C	N	C	C	N
Updating labels	C	C	N	A	N	N	C	N	C
Requirements									
Updating Requirements	R	R	R	L	C	R	R	R	L
Questioning with customer	R	R	L	R	R	R	L	R	C
Mockups									
Updating Scenario 1 Design	R	R	C	R	L	R	C	R	R
Updating Scenario 2 Design	R	R	C	R	L	R	C	R	R
Updating Scenario 3 Design	R	R	L	R	C	R	C	R	R
Updating Scenario 4 Design	R	R	C	R	C	R	L	R	R
Updating Scenario 5 Design	R	R	C	R	C	R	L	R	R
Uploding Scenarios to Wiki Page	R	R	C	R	C	R	L	R	R
Diagrams									
Updating Use Case Diagram	R	R	C	R	R	R	R	C	R
Updating Class Diagram	R	C	R	C	R	C	R	R	C
Updating Sequence Diagrams	C	R	R	R	R	R	C	R	R
Design									
Deciding Design Platform	C	N	N	C	C	N	C	C	N
Designing Web Homapage	L	N	N	R	R	N	R	R	N
Designing Web Profile Page	R	N	N	R	R	N	L	R	N
Designing Web Login	R	N	N	R	R	N	L	R	N
Designing Web Sign Up	L	N	N	R	R	N	R	R	N
Designing Mobile Homapage	R	R	R	L	C	R	R	R	R
Designing Mobile Profile Page	R	R	R	C	C	R	R	R	R
Designing Mobile Login	R	R	R	C	L	R	R	R	R
Designing Mobile Sing Up	R	R	R	C	L	R	R	R	C
Designing Mobile Settings	R	R	R	R	L	R	R	R	R

Configuration and Setup									
Name selection	C	C	C	C	C	C	L	C	C
Android, Backend and Frontend Team selection	C	C	C	C	C	C	C	C	C
Deciding & Setting up Development Technologies (mobile)	N	N	C	C	C	N	N	N	N
Deciding & Setting up Development Technologies (front-end)	C	N	N	N	N	N	C	C	N
Deciding & Setting up Development Technologies (back-end)	N	C	N	N	N	C	N	N	C
Backend 1									
Configuring database for users	R	L	R	R	R	R	R	R	C
Configuring database for resources and topics	R	C	R	R	R	R	R	R	R
API for Login	R	L	R	R	R	C	R	R	R
API for Sign Up	R	L	R	R	R	C	R	R	R
API for Delete User	R	A	R	R	R	L	R	R	L
Implementing Admin User functionalities	R	L	R	R	R	C	R	R	C
API for User Search	R	L	R	R	R	A	R	R	C
Frontend 1									
Home page	L	R	R	R	R	R	R	R	R
Login	L	R	R	R	R	R	C	R	R
Sign Up	C	R	R	R	R	R	L	R	R
Profile page	C	R	R	R	R	R	R	L	R
User Search	A	R	R	R	R	R	R	L	R
Navigation	A	R	R	R	R	R	R	L	R
Deployment	C	R	R	R	R	R	L	R	R
Connecting with Backend	L	C	N	N	N	C	C	C	C
Android 1									
Home page	R	R	C	L	C	R	R	R	R
Login	R	R	C	C	L	R	R	R	R
Sign Up	R	R	C	C	L	R	R	R	R
Profile page	R	R	L	C	C	R	R	R	R
User Search	R	R	C	C	C	R	R	R	R
Navigation	R	R	C	C	L	R	R	R	R
Connecting with Backend	N	C	C	C	C	C	N	N	C
Milestone 1 Report									
RAM	C	C	C	L	C	C	L	C	C
Milestone 1 Report	C	C	C	C	C	C	C	C	C

Milestone Review

Customer Demo 1 was a successful presentation for us. Preparing this document makes us realize that what we did right and what we did wrong. It was a good first reflection. We will be better at Milestone 2.