CmpE 451

Milestone Report 1

BeABee

20.11.2021

Adalet Veyis Turgut
Burakcan Kazım Yeşil
İhsan Gökcül
Kürşat Talha Berk Yamanoğlu
Ahmet Berat Can
Ömer Yılmaz
Refika Kalyoncu
Zahit Batuhan Tongarlak

Executive Summary **5**
Project Description 5
The Track System 5
Social Features 6
Project Status 6
Moving Forward 7
LIST AND STATUS OF DELIVERABLES **8**
EVALUATION of THE STATUS of DELIVERABLES and ITS IMPACT on OUR PROJECT PLAN **10**
BACKEND 10
Register Endpoint **10**
Login Endpoint **10**
Forgot and Reset Password Endpoints **10**
Goal Endpoints **10**
Entity Endpoints **10**
Entity Linking Endpoints **11**
API Documentation (Swagger Endpoint) **11**
Unittests **11**
FRONTEND 11
Register Page **11**
Login Page **11**
Reset Password Page **12**
Home Page **12**
Goals Page **12**
New Goals Page **12**
New Entity Page **12**
Edit Entity Page **12**
ANDROID 13
Register Page **13**
Login Page **13**
Reset Password Page **13**
Goals Page **13**
Goal Page **13**
Edit Goal Page **13**
Subgoal Page **13**
Edit Subgoal Page **13**
Entity Page **14**
Edit Entity Page **14**
DEPLOYMENT 14
Backend **14**
Frontend **14**

CI/CD in Github Actions **14**

EVALUATION of TOOLS and PROCESSES WE HAVE USED to MANAGE OUR TEAM PROJECT **15**
BACKEND 15
FRONTEND 16
ANDROID 17

THE REQUIREMENTS ADDRESSED IN THIS MILESTONE **18**

A SUMMARY of WORK PERFORMED by EACH TEAM MEMBER **19**
BACKEND 19
ANDROID 20
FRONTEND 23

REQUIREMENTS **25**
Glossary 25
1 Functional Requirements 26
2 Non-Functional Requirements 28

UML SOFTWARE DESIGNS **30**
Class Diagram 30
Use Case Diagram 31
Sequence Diagrams 32
1. Creating a New Goal 32
2. Joining a Group With Invitation Code 32
3. Getting token for a group goal 33
4. Link a Task to a Question 33
5. Searching Goals in Marketplace 34
6. Creating a Task and Setting a Deadline 34
7. Extend Deadline For a Task 35
8. Rate a Completed Task 35
9. Search for Users 36
10. Follow a User 36

RAM **36**

SCENARIOS and MOCKUPS **39**
Scenario 39
Mockups 41

PROJECT PLAN **56**

COMMUNICATION PLAN **59**

API DOCUMENTATION **60**
MODELS 60
ENDPOINTS 62
DETAILS OF SOME ENDPOINTS 64

Executive Summary

Project Description

BeABee is about self development. It is meant to help users to easily track their endeavors, goals about anything. It also aims to provide a limited social environment in which a user will be able to follow the development of another user. The project will consist of an Android application, a website and a backend server.

The Track System

A user can create a Goal which they can later divide into Subgoals and Subgoals of Subgoals. This behaviour allows users to split the Goal into many hierarchical Subgoals thus providing a flexible platform which can be used for any type of Goal. After the hierarchical structure is built, the user can link many types of entities to any level of the Goal Hierarchy.

The entities consist of Questions, Reflections, Tasks, and Routines. Once an Entity is registered to the system, the user can link it to the Goal itself, to any one or more of the Subgoals, or even to another Entity. This flexible and agile structure of the system allows users to create very unique Goal structures.

A Question is something that the user can use to note down their questions for them to find the answer later or to remind themselves about a confusing situation that they should keep in mind. They can assign a reminder to the question or they can link it to the other Entities or the other way around.

A Reflection is basically a note from the user to themselves which can be linked to the other Entities or to any point in the Goal Hierarchy. Because of its dynamic nature, it can serve various purposes thus being the best assist it can for various users with various working methods.

A Task is something that the user should do before a deadline. When creating a task, the user can set a deadline, link it to other entities to better understand the relation between other Entities, Subgoals and the Task. After the deadline of the Task, the user can extend the deadline or mark the Task as canceled if they didn't finish the task. They can give feedback to the Task about how well they were able to manage the task to later see the insights about their development process.

A Routine is very similar to a Task except it is not something that should be done once. Using a Routine, a user can track the things they should regularly do. When creating a Routine, the times that the Routine should be done must be clarified. The examples of the occurrence time can be every weekday between 09.00 and 10.00 or every wednesday evening between 19.00 and 20.00. Similar to the other Entities, any other entity can be linked to a Routine and vice versa.

The users can get insights and some analytics about their development process like how well they were able to follow the track in the last days.

Social Features

Besides the personal development track, the Project can be used to track a Group Goal. Users can create a Group Goal and share the Join Token with their future comrades for them to be able to join the Group Goal. The track system of the Group Goal is very similar to the personal Goal. The difference is that in the personal Goal all the Entities are automatically assigned to the owner, in the Group Goal there can be one or many assignees for any of the Entities which allows Group members to work together or separately according to the nature of the work in hand.

One additional feature is that sometimes it can be hard to prepare a Goal Track structure or the users might want to use a Goal Track that is prepared by somebody else. Whether it is a Goal the user just likes or a Goal that is prepared by a specialist, the user can pick a Goal Scheme from the Marketplace and create a copy of the Goal which they can later modify in their own interests. The marketplace is where the users can share their prepared Goal structure as a Scheme or browse through the Schemes created by other users and copy them.

Last but not least users can follow other users and their development process if they want to. A user can set their Goals or any Entities related to that Goal as public or private, which determines if another user will be able to see them or not. A user can search through other users and follow the public accounts. The following relationship is a one way relationship, the follower can see the information but they cannot make any comments to them. This is to prevent harassment and other negative effects on the self development of the user being followed.

Project Status

So far we have created the Backend Server, Android application and the Frontend page with only a part of the application. We have implemented the core of the application which is the Goal System as well as the registration system.

We started implementation with the registration and the login system. Even though it looks like the registration system is so common and not that important, we used it to start implementation of both client platforms and the backend server so that we could have a project that we can work on and see how the teams interact with each other and set up the work flow.

After completing the registration systems, we started implementation of the core goal structure. For starters we ignored the Group Goal functionality and focused on the hierarchical structure of goals, subgoals and other entities.

So now in our application a user can sign-up and login, and start using the main functionality of the app. By main functionality we mean the goal system. The user can create goals for themselves, can create subgoals and entities under a goal and link created entities to each other. Besides creating and organizing the hierarchy, the user can also edit those created entities and goals. The detailed data structure and functionality of the different entities have

not been yet implemented. So each entity has basic data like title, description and the linked entities.
We don’t yet support any other functionality that is not mentioned in the paragraph above. We are going according to the project plan that we have prepared and happy with the progress.

Moving Forward

After the customer meeting we have detected some changes that we want to do at the currently implemented part. We are going to implement the details of each entity and refine the relationship between them so that the user can clearly understand what is going on with their goal, and their created hierarchy. We may change the design of the client applications to be more user friendly according to the reviews of our customers.

In addition to the changes on the current implementation, we are looking ahead and we are planning to slowly add the social features of the app as well. The group goal and users being able to follow each other are the social features we are setting our aim to. We also want to add a functionality for our users to be able to search existing entities inside their goals as well as operate searches in the application database for existing goals including the other users goals as well.

LIST AND STATUS OF DELIVERABLES

	DELIVERABLE	DUE DATE	STATUS
1	PROJECT PLAN	20.11.2021	DELIVERED
2	PROJECT REQUIREMENTS	20.11.2021	DELIVERED
3	UML SOFTWARE DESIGNS	20.11.2021	DELIVERED
4	SCENARIOS AND MOCKUPS	20.11.2021	DELIVERED
	ANDROID	20.11.2021	DELIVERED
	FRONTEND	20.11.2021	DELIVERED
5	PROJECT PLAN	20.11.2021	DELIVERED
6	COMMUNICATION PLAN	20.11.2021	DELIVERED
7	RAM	20.11.2021	
8	FRONTEND	20.11.2021	
	REGISTER PAGE	20.11.2021	DELIVERED
	LOGIN PAGE	20.11.2021	DELIVERED
	RESET PASSWORD PAGE	20.11.2021	FAILED
	GOALS PAGE	20.11.2021	DELIVERED
	NEW GOAL PAGE	20.11.2021	DELIVERED
	NEW ENTITY PAGE	20.11.2021	DELIVERED
	SUBPAGES IN GOALS PAGE	20.11.2021	DELIVERED
	EDIT ENTITY PAGE	20.11.2021	PARTIALLY
9	ANDROID	20.11.2021	
	REGISTER PAGE	20.11.2021	DELIVERED
	LOGIN PAGE	20.11.2021	DELIVERED
	RESET PASSWORD PAGE	20.11.2021	FAILED
	GOALS PAGE	20.11.2021	DELIVERED
	NEW GOAL PAGE	20.11.2021	DELIVERED
	NEW ENTITY PAGE	20.11.2021	DELIVERED

	SUBPAGES IN GOALS PAGE	20.11.2021	DELIVERED
	EDIT ENTITY PAGE	20.11.2021	PARTIALLY
10	BACKEND	20.11.2021	
	REGISTER ENDPOINT	20.11.2021	DELIVERED
	LOGIN ENDPOINT	20.11.2021	DELIVERED
	RESET PASSWORD ENDPOINT	20.11.2021	DELIVERED
	GOAL CRUD ENDPOINTS	20.11.2021	DELIVERED
	ENTITY CRUD ENPOINTS	20.11.2021	DELIVERED
	LINKING ENTITIES ENDPOINTS	20.11.2021	DELIVERED
	RESOURCES, LABELING ENDPOINTS	20.11.2021	FAILED
	API DOCUMENTATION (SWAGGER)	20.11.2021	DELIVERED
	UNITTESTS	20.11.2021	PARTIALLY
11	CI/CD PIPELINE ON GITHUB ACTIONS	20.11.2021	PARTIALLY
12	DEPLOYMENT	20.11.2021	
	FRONTEND	20.11.2021	FAILED
	BACKEND	20.11.2021	DELIVERED

EVALUATION of THE STATUS of DELIVERABLES and ITS IMPACT on OUR PROJECT PLAN

BACKEND

Register Endpoint

Registration Endpoint ensures that users with no account can register to our backend and access functionalities of our application. Users register by entering their username, email and password. Two users can not have the same emails or passwords, and if they try to register with an username or email that is already in use they will get an error. In future we will also provide the mechanism to register using their Google accounts using Google OAuth services.
Relevant Requirements: 1.1.1.1

Login Endpoint

After registering, users can login to their BeABee account using our Login Endpoint. A user can login by entering their username, email and password.
Relevant Requirements: 1.1.2.1

Forgot and Reset Password Endpoints

These endpoints implement the functionality to ensure a user can change their password if they forget it. Forgot Password Endpoint sends a token to the user's email and users can reset their password using this token with Reset Password Endpoint.
Relevant Requirements: 1.1.2.3, 1.2.7.1

Goal Endpoints

Our backend provides a number of endpoints to achieve different operations regarding goals. Most of these endpoints provide standard CRUD operations. This provides functionality to create, update and delete a goal with the id of the goal. Also information about one specific goal can be reached by id of that goal, or all goals of a user can be listed by id of that user. In the future we are planning to implement the function to let users share their goals with the public and let them also use the goals that they have found from the public. Also group goals will be implemented which will create group goals and invite other users to work on their common goal.
Relevant Requirements: 1.1.3.1, 1.1.3.6, 1.1.3.11, 1.2.1.1, 1.2.5.1*

Entity Endpoints

Backend system also provides different operations with all kinds of entities. These endpoints are subgoal, task, routine, reflection and question. Our endpoints provide standard CRUD operations.
Entities can be created by providing all the relevant information about that entity. And to update an entity, the user should reach the entity with the entity's id and provide the fields

they want to update. All created entitier are linked to a main goal. To delete an entity they just need the entity's id.
Users also can get information about an entity with its Id or get entities of a user or goal providing user's or goal's id. In addition one can get every instance of one kind of entity such as all subgoals or tasks of a user with user's id.
In the future, creation of entities will not need a main goal since creation of standalone entities should be possible. Users also will be able to add resources to their entities. Also subgoals will no longer be an entity and will be a standalone type of objcect.
Relevant Requirements: 1.1.3.6, 1.1.3.14.2

Entity Linking Endpoints

In order to provide functionality to link entitis together, our API provides link related endpoints. These endpoints let users two entities with their ids, get all links related to an entity and delete links between entities.
Relevant Requirements: 1.1.3.7

API Documentation (Swagger Endpoint)

Our backend provides documentation for the whole API. It is generated by Swagger with example request and response objects. Since it is deployed with the Backend application, it can be reached with the same IP address.
Relevant Requirements: N/A

Unittests

There are tests covering all of gial space functions. When building our application we run these tests to ensure the application is running as expected.
Relevant Requirements: N/A

FRONTEND

Register Page

This page allows guest users to create an account on our App. Required fields for creating an account are username, email, and password. This page is partially completed. We will implement signing up by using Google accounts in the future.
Relevant requirements: 1.1.1.1

Login Page

On our App's Login Page, Users are prompted to give their credential. Required credentials are username or email, and password. This page is partially completed. We will implement login functionality by using Google accounts in the future. We created a Reset Password Page on frontend but didn't connect the related endpoint in our backend.
Relevant Requirements: 1.1.2.1

Reset Password Page

We created a Reset Password Page on frontend but didn't connect the related endpoint in our backend. This page is partially completed. We will implement a backend connection in the future.
Relevant Requirements: N/A

Home Page

This page contains Navigation Bar at top, SideBar at left, and Logo at the top left corner. Navigation Bar has links to "Home Page", "Login Page", "Register Page" when the user was not logged in. When a user logs in, the Navigation Bar has links to "Home Page", "Goals Page", and a sign-out button. This page is not completed. It is created as a mock page. We will implement related requirements in the future.
Relevant Requirements: N/A

Goals Page

This page allows logged in users to view their previously created goals. Users can view the status and deadline of goals. Users can edit or delete previously created goals.
This page is partially completed. We will implement adding subgoal features in the future.
Relevant Requirements: 1.1.3.1, 1.1.3.11

New Goals Page

This page allows users to create new goals. This page is completed. We met the requirements that were addressed in the milestone.
Relevant Requirements: 1.1.3.6.1, 1.1.3.6.2

New Entity Page

This page allows users to create a new entity in a goal, and edit or delete an entity. Users can link an entity to another entity, and view the linked entities of a specific entity. This page is partially completed. We will implement the related requirements in the future.
Relevant Requirements: 1.1.3.14.1, 1.1.3.14.2

Edit Entity Page

This page allows users to edit an entity. This page is partially completed. We will implement the related requirements in the future.
Relevant Requirements: 1.1.3.6

ANDROID

Register Page

Register page allows guest users to create an account on our App by providing username, email, and password. We will implement “signing up by using Google accounts” will be added in future.

Login Page

On our App’s Login Page, Users are prompted to give their credential. Required credentials are username or email, and password.” Login by using Google accounts” feature will be added in the future.

Reset Password Page

The reset password page isn’t implemented yet.

Goals Page

Goals page opens as an empty page with a plus sign in the middle of the screen where users can click anywhere to create a goal if there isn’t any goal. If there is an existing goal or goals the home page is able to show them all in a vertical list where each goal is also an item which you can enter to view. Also there exists a plus button to add new goals. This page contains a navigation bar on the bottom where users can go through different pages such as the market place in the future.

This page allows logged in users to view their previously created goals.

Goal Page

This page allows users to view an existing title, description. Users can add subgoals and entities with clicking the relative plus buttons in the goal page. Users can edit but cannot delete previously created goals.

Edit Goal Page

This page allows users to edit the title and the description of a goal.

Subgoal Page

This page allows users to view an existing goal and create new goals with title, description. Users can add subgoals and entities with clicking the relative plus buttons in the goal page. Users can edit but cannot delete previously created subgoals.

Edit Subgoal Page

This page allows users to edit the title and the description of a subgoal.

Entity Page

This page allows users to view existing entities and create a new entity in a goal.
Users can link an entity to another. Users are not able to create an entity under an entity directly. Users can view the linked entities of a specific entity. This page is partially completed. We will implement the related requirements in the future. Users can edit but cannot delete previously created entities.

Edit Entity Page

This page allows users to edit the title and the description of an entity. Users are not able to set the deadline or period of routines.

DEPLOYMENT

Backend

Backend application's latest version is deployed to an Amazon EC2 server. All of the endpoints and functionalities are present and usable.

Frontend

Deployment of the frontend was a bit problematic for us. The deployment was not successful because of the memory problems we had with the EC2 instance.

CI/CD in Github Actions

Github Actions has a workflow with our backend which builds and deploys the Docker images to the Amazon EC2 instance. However it only works for backend and we do not have one for the frontend application.

EVALUATION of TOOLS and PROCESSES WE HAVE USED to MANAGE OUR TEAM PROJECT

BACKEND

We used whatsapp as our main communication channel, Java as our main programming language, Spring Boot as our web framework, PostgreSql as our database management system, IntelliJ as our IDE, Swagger as our API documentation tool and git as our version control system.

Since our team consists of two people. We do talk about the little details on Whatsapp. But if the subject is important and needs careful consideration, we use Discord. We reach each other instantly via Whatsapp, we specify a special hour beforehand for discord meetings. Screen sharing and voice channel functionalities of Discord are amazing. We are satisfied until now, and do not see any reason to switch these channels.

We decided to use the Spring Boot framework since both of us knew it well. Main reason was minimizing the learning time and maximizing the implementation time. Spring Boot also has a great community and we found solutions to almost all of the problems we faced. So far we are happy that we chose Spring. Libraries like Lombok, mapstruct easened the implementation process and saved us from hundreds of lines of code. Also other libraries and frameworks like Junit4, sun-mail, spring-fox swagger helped us to achieve the ultimate goal. Also Spring boot easily integrates with the database with JPA and annotations (starting with @) ease the implementation process.

PostgreSql is used as the database management system in the project, since we are used to working with relational databases and worked with Postgres before. It easily integrates with the Spring Boot. Also we did not encounter any problem while dockerizing and deploying. We are glad we chose Postgres.

IntelliJ is probably by far the best IDE for java projects. It has a built-in terminal, vcs, database explorer, and http client. We used the built-in terminal for git pull-commit-push operations. built-in vcs for resolving conflicts on merges, database explorer for playing with database, and even tested endpoints with built-in http client. It's a great tool and contains most of our needs. We are glad we can use this tool for free with the student license.

Postman is an API testing application. We used it for testing our endpoints frequently. You can save a request to a collection and order them as you like. Configuring the body, header or parameter of the requests is very simple.

During the implementation, we used git as a version control system. We created dev branches for each team. Also we branched out from these dev branches for every feature we

implemented. After implementing a feature, we opened a pull request wanting to merge with the dev branch. After merging all code in dev branches, we wanted to merge to main branches, but we didn't have the time to do that. Git is great for collaborative work, our job is much easier with it.

FRONTEND

After the first meeting, we decided to use React for frontend development. Ahmet Berat is experienced in this framework, Batuhan and Ömer had no experience in frontend development. One of the main reasons for selecting this framework is that it is open-source, widely used, and easy to understand and develop applications.

After selecting a framework to develop the frontend, we decided to use Ant Design for the UI framework. It is said to be "The world's second most popular React UI framework" on its website. It is easy to use pre-implemented components such as login form, register form, buttons, and some other forms for our frontend project. There is no need to rediscover America!

We used the Route component from react-router-dom to connect pages and components. With routing, we can send some parameters such as goal_id, entity_id to other pages. Thanks to this feature we were able to split the work and everybody was able to work on their work.

While creating our pages we needed data to show users. Since the backend was not implemented yet, we used mock data. To use mock data, we implemented a basic Flask backend without any database connection. To send get, put, delete, post requests to the backend, we used the Axios library. It is one of the most popular libraries for sending HTTP requests and it is easy to use.

After the backend was implemented, they shared the IP and swagger page with us. We changed our requests to connect to the backend. To change requests properly, we used the swagger page that was shared with us.

While doing all these developments, we used git as the version control system. We had frontend/main, frontend/dev branches. The first is for a stable frontend version, the second one is for merging new features. We all created new branches from frontend/dev. After implementing features assigned to us, we created pull requests on git and merged them.

We used Visual Studio Code as code editor. It is the best and easy to use editor for developing React applications. It has lots of extensions to ease developers' jobs. We used an extension called Prettier Code Formatter which is used for formatting the code to increase readability. It sets the indentation automatically. We also used an extension called Typescript React code snippets. It is used for creating initial code snippets for React Typescript function based or class based components. It is very helpful for beginner React developers. We used another extension called Git Blame. It is used for knowing the author of every line in the code. It also

tells us when this line was created and what was the merged pull request. VS Code is a great tool for web development and it has many extensions that we will need in the future.

ANDROID

We have used Android Studio as the IDE and Java as the language of the implementation. The Android Studio is the recommended IDE by Google and it has support for many things that we needed during implementation like an ADB manager that we can use to deploy our app to a real android device and debug it or like an internal visual VCS connection tool and many more.

As for using Java as the implementation language, we didn't have any problems with it. Since we all knew Java we didn’t want to bury ourselves under another burden of learning Kotlin and using it. Everyone in the team is comfortable with Java and could complete the tasks given so far.

Besides these choices, we created a base structure for the Android application. Since we needed a home page that we can display the hierarchy to the user and other pages, for usability’s sake we created a bottom navigation bar and a navigation hierarchy like the instagram’s app. In this hierarchy the users can go deep in the hierarchy and come back using a stack like structure. For being able to create this structure we used an Activity as the main activity and container Fragments for each page connected to the bottom navigation bar. Later we created a fragment base which is called BaseInnerFragment that the other fragments to be displayed in a hierarchical view should extend. These BaseInnerFragment instances will be placed into the container fragments. So we have a structure that can be extended horizontally using the bottom navigation bar and adding new container fragments as well as it can be extended vertically using inner fragments and adding them to a container fragments stack. For easier implementation in future BaseInnerFragment contains necessary methods as ‘protected’ so child fragments extended from it can access and use them. A few examples to these functionality provided from BaseInnerFragment to it’s childs are changing the top bar’s visibility and it’s button’s functionality, opening and closing fragments from the stack, access to the network service api etc. This is a design pattern called Sandbox and it has been proven useful to us so far. By using these patterns and structures we were able to create a structure with a dynamic number of pages(as our application requires), and complete the implementation of many different views and merging them in such a short time(1 week). Also since our structure is dynamic we can modify it quickly according to our customers’ needs and reviews.

Using this structure also allowed us to share the pages to be implemented between team members, implement them separately and later connect them without having big problems or the need to change existing code so much.

THE REQUIREMENTS ADDRESSED IN THIS MILESTONE

- 1.1.1.1 Guests shall enter their email, name, surname, username and a password in order to sign up.
- 1.1.2.1 Users shall be able to login using their credentials.
- 1.1.2.3 Users shall be able to reset their passwords in case of forgetting. Confirmation mail should be used in this process. -- only in backend
- 1.1.3.1 Users shall be able to add their goals which contain a title, a description and subgoal(s). -- only with a title and a description. Users can add subgoals later on.
- 1.1.3.6 Users shall be able to modify their entities, goals, subgoals.
    - 1.1.3.6.1 Users shall be able to modify titles.
    - 1.1.3.6.2 Users shall be able to modify descriptions.
- 1.1.3.7 Users shall be able to connect entities and subgoals of a goal to other entities and subgoals of that same goal.
- 1.1.3.11 Users shall be able to delete goals.
- 1.1.3.14 Users shall be able to add entities.
    - 1.1.3.14.2 They shall be able to add entities under other items.

A SUMMARY of WORK PERFORMED by EACH TEAM MEMBER

BACKEND

Burakcan Kazım Yeşil	Review of Backend Application Pull Requests
	Deployment of Backend Application to the Amazon EC2
	Wrote Dockerfile for Frontend and Backend application
	Wrote the docker-compose file, which runs Backend Server and Database together to deploy to the Amazon instance smoothly.
	CI/CD Pipeline with Github Actions
	Reorganized the Backend application with Feature based packaging
	Fixed bugs on the backend application.
	Prepared Evaluation of Deliverables For Backend and Deployment for Milestone

Adalet Veyis Turgut	Arranging group-wide meetings when necessary.
	Taking meeting notes and uploading them to the wiki page.
	Helping frontend and Android teams to create scenarios for customer demo.
	Revising requirements.
	Revising project plan.
	Gardening wiki home page.
	Filling personal efforts weekly.
	Configuring database.
	Implementing the backend of the application.
	Documenting endpoints so that other teams can understand what endpoints do.
	Reviewing Burakcan's pull requests.

	Writing the “list of status of deliverables” part of this report.
	Writing the backend part of the “evaluation of tools and processes we used to manage the project” part of this report.
	Writing “the requirements addressed in this milestone” part of this report.
	Writing “API documentation” part of this report.
	Merging and finalizing this milestone.

ANDROID

Kürşat Talha Berk Yamanoğlu	Prepare the repo structure and branch structure to be used, document it in the wiki.
	Created Initial Android Application
	Review and revise the requirements
	Meeting with the team and creating Login/Signup Page and Network
	Structure while showing and teaching them. Added necessary SDK’s and libraries to the project.
	Created network connection structure(with the team, like a tutorial)
	Created Login/Signup Pages(with the team, like a tutorial)
	Connected Login/Signup to our Backend
	Created Home Page Structure
	Created BaseInnerFragment and BaseContainerFragment as the base classes to be used during future development
	Created Goals Page.
	Meeting with the team and telling them about the structure and helping them with their questions.
	Review created pages by Ihsan and Refika

	Merged Entity Pages(implemented by Ihsan), Goal/Subgoal Pages(implemented by Refika), and Goals Page(implemented by me).
	Connected above mentioned pages to each other and created navigation flow
	Connected above mentioned pages to the backend server.
	Reported a few bugs and structural problems with the backend server api to the backend team
	Helped preparation of the Scenario for the Customer Meeting.
	Presented the scenario with Omer
	Prepare Executive Summary for the Group Milestone Report
	Prepare the evaluation of tools and processes for the Android team

Refika Kalyoncu	Meeting with the team and creating Login/Signup Page and Network Structure. Added necessary SDK's and libraries to the project.
	Created network connection structure with the team
	Created Login/Signup Pages with the team
	Created Goal Page
	Created Subgoal page
	Created Subgoal card view
	Created The tag entity
	Created responses for tag and subgoal
	Connected the goal and subgoal pages with the relative entities
	Connected the goal and subgoal pages with the relative entities
	Attended android team meetings
	Attended lab meeting and group meetings
	Revised project plan
	Took notes during android team and group meetings

	Review created pages by Ihsan and Berk
	Helped preparation of the Scenario for the Customer Meeting.
	Prepared Evaluation of the status of deliverables for the Android team

İhsan Gökcül	Attended all group and customer meetings
	Studied Android and Android Studio.
	Created Bug Report issue template.
	Created login page with the team to learn Android.
	Attended meetings with Android team
	Tried to do the things that Berk taught us in Android meetings.
	Created task fragment and its related layouts and classes for Android app.
	Created routine fragment and its related layouts and classes for Android app.
	Created question fragment and its related layouts and classes for Android app.
	Created reflection fragment and its related layouts and classes for Android app.
	Created pull request for my work.
	Created necessary issues for my work.
	Reviewed android team pull requests.
	Created the scenario for customer presentation.
	Added Requirements, UML Software Designs, Scenarios and Mockups, Project plan and Communication Plan to the Milestone Report.
	Documented the scenario and mockup of the customer presentation.
	Helped to organize the milestone report.

FRONTEND

Ömer Yılmaz	Created Add Goal Page
	Created Edit Goal Page
	Created View Goal Page
	Created View All Goals Page
	Connected the Goal Pages to the backend
	Created several issues issues for backend
	Helped frontend team about the deployment
	Helped the implementation of Entity Pages
	Tested the scenario on frontend
	Gave feedback for the backend API and asked new endpoints

Zahit Batuhan Tongarlak	Updated and re-organizing the project plan
	Created Issue template for feature request
	Designed a new logo
	Created View Entity Page
	Created Add Entity Page
	Created Link Entity Page
	Created Edit Entity Page
	Tested the front-end on my local with help of Ömer
	Created a bug issue about Editing an Entity
	I re-organized the RAM

Ahmet Berat Can	Created initial file structure for frontend using React Typescript
	Added Ant design , React-Router,React-Router Dom, and Axios libraries to the frontend app
	Created Login Page
	Created Home Page

	Connected Login Page to the backend
	Created Register Page
	Connected Register Page to the backend
	Created Forgot Password Page
	Implemented routing mechanism of frontend using react-router
	Helped other members of the frontend team when they needed help

REQUIREMENTS

Glossary

Credentials: Users' identifiable information and details to log in. It consists of email, username and password.

Encryption: The method by which information is converted into secret code that hides the information's true meaning.

Entity: Comprehensive term for Tasks, Routines, Questions and Reflections.

Goal: Task or tasks that users can set and track their progress on the application.

Subgoal: A task that is a smaller part of a goal.
Description: A brief explanation of a particular goal.
Guest: A user that hasn’t registered yet.

Item: Comprehensive term for Goals, Subgoals and Entities.

Resources: Useful materials beneficial to achieve a specific goal.

Reflections: General evaluation about experiences during a goal.

Questions: A user's illustrative self-questions regarding to a specific goal. They help users to clarify the path of their goals.

Search: A tool that helps a user to find similar goals according to their input keywords.

Semantic Search: A search method that uses context to use the meaning of the search instead of the literal words.
Semantic tag: Tag that defines the theme of a goal clearly, used in semantic search.

User: A person who uses the application to reach their goals.

Group: Multiple users who are working on a common goal.
User Interface: The point of human-computer interaction and communication on a device or app.

Visibility: A property which specifies a goal is whether private or public.

1 Functional Requirements

1.1 User Requirements

1.1.1 Guests
1.1.1.1 Guests shall enter their e-mail, name, surname, username and a password in order to sign up.
1.1.1.2 Guests should be able to sign up using their Google accounts.
1.1.1.3 Guests shall be able to search for users by their username.
1.1.1.4 Guests shall be able to search for public goals using the search functionality mentioned (1.2.2).

1.1.2 Login
1.1.2.1 Users shall be able to login using their credentials.
1.1.2.2 Users should be able to login using their Google accounts.
1.1.2.3 Users shall be able to reset their passwords in case of forgetting. Confirmation mail should be used in this process.

1.1.3 Users
1.1.3.1 Users shall be able to add their goals which contains a title, a description and subgoal(s).
1.1.3.2 Users shall be able to rate their efforts towards their tasks and routines after deadline.
1.1.3.3 Users shall be able to describe the practices that support these goals under resources section.
1.1.3.4 Users shall be able to view and assess their progress using analytics section.
    1.1.3.4.1 Users should be able to filter these analytics with respect to time.
1.1.3.5 Users shall be able to add the related resources to their goals.
1.1.3.6 Users shall be able to modify their entities, goals, subgoals.
    1.1.3.6.1 Users shall be able to modify titles.
    1.1.3.6.2 Users shall be able to modify descriptions.
1.1.3.7 Users shall be able to connect entities and subgoals of a goal to other entities and subgoals of that same goal.
1.1.3.8 Users shall be able to tag goals and items with semantic tags.
1.1.3.9 Users shall be able to search goals in marketplace.
1.1.3.10 Users shall be able to add goals that came out of the search to themselves.
1.1.3.11 Users shall be able to delete goals.
1.1.3.12 Users should be able to mute and enable notifications.
1.1.3.13 Users shall be able to see the items of the other users they are following in their feed.
1.1.3.14 Users shall be able to add entities.
    1.1.3.14.1 They shall be able to add standalone entities.
    1.1.3.14.2 They shall be able to add entities under other items.
1.1.3.15 Users shall be able to search entities.
1.1.3.16 Users should be able to mute or enable notifications.
1.1.3.17 Users shall be able to give feedback about their goals / subgoals.

1.1.4 Groups
1.1.4.1 Users shall be able to invite others using the invitation code (token).
1.1.4.2 Users shall be able to join others' goals using invitation code (token).
1.1.4.3 Members of the groups shall be able to rate their efforts towards entities after completing (or maybe failing to complete) them.
1.1.4.4 Members of the groups shall be able to describe the practices that support this group goal under resources section.
1.1.4.5 Members of the groups shall be able to view and assess their progress using analytics section.
1.1.4.6 Members of the groups shall be able to add the related resources to group goal.
1.1.4.7 Members of the groups shall be able to modify the entities or subgoals.
1.1.4.8 Members of the groups shall be able to tag items with semantic tags.

1.1.5 Interaction Between Users
1.1.5.1 Users shall be able to visit others' personal page if their visibiliy is set to public, and see their public goals and analytics.
1.1.5.2 Users shall be able to follow other users.

1.1.6 User Interface
1.1.6.1 Users shall be able to track their progress over time. They should see their weekly and monthly completion rate.
1.1.6.2 Users shall be able to see their analytics related to their goals and effort expended on these goals.

1.2 System Requirements

1.2.1 Goals / Subgoals
1.2.1.1 The system shall provide resources section under goals / subgoals. This section should takes text, photo or question-answer as input.
1.2.1.2 The system shall not allow guests to be group members.

1.2.2 Searching
1.2.2.1 The system shall support semantic search, which provides users to find semantically similar goals.
1.2.2.2 The system shall show the popular goals to users and guests.
1.2.2.3 The system shall supply usage of tags while searching.

1.2.3 Notifications
1.2.3.1 The system shall send notifications.
1.2.3.2 The system shall notify users regarding development with respect to a plan or others' in case of collaborative contexts.
1.2.3.3 The system shall notify users 1 hour before the start time of their tasks and routines.
1.2.3.4 The system should notify users about users they follow.

1.2.3.4.1 These notifications should be about their new goals, feedbacks, entities.

1.2.4 Analytics
1.2.4.1 The system shall provide analytics about activities those have common lifetimes.
1.2.4.2 The system shall provide analytics about related activities.
1.2.4.3 The system shall provide feedback button after deadline of items.

1.2.5 Profile Management
1.2.5.1 The system shall let users to view others' and their goals.

1.2.6 Questions / Reflections
1.2.6.1 The system shall let users to set reminders for questions.
1.2.6.2 The system shall let users to mark questions as resolved.

1.2.7 Login / Account Management
1.2.7.1 The system should use confirmation mail upon password resetting.

2 Non-Functional Requirements

2.1 Standarts
2.1.1 The system shall use the W3C Activity Stream Standart 2.0 for subscriptions and notifications.
2.1.2 The system shall support semantic tags and these semantic tags should be supported with wikidata.org

2.2 Security
2.2.1 User passwords shall be encrypted before storing in database using md5 or better hashing algorithms.
2.2.2 The System shall use HTTPS protocol.

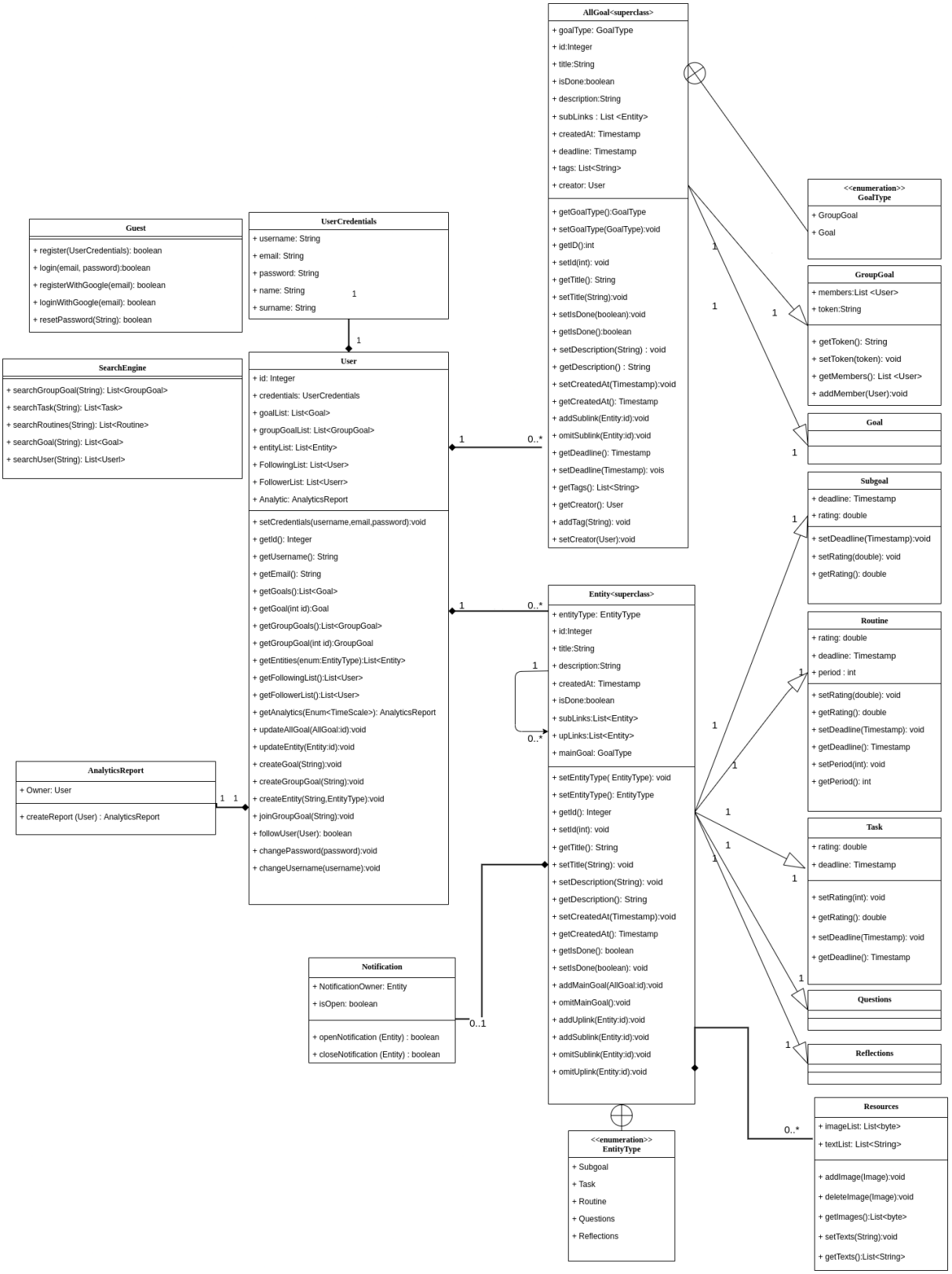
2.3 Portability
2.3.1 The system should support modern web browsers.
2.3.1.1 Chrome version >= 89
2.3.1.2 Opera version >= 75
2.3.1.3 Safari version >= 14
2.3.1.4 Firefox version >= 85
2.3.4.5 Edge version >= 89
2.3.2 Android application should support mid-tier mobile phones.
2.3.2.1 Android version >= 21
2.3.2.2 Free storage >= 1 GB
2.3.2.3 RAM >= 4 GB

2.4 Privacy

2.4.1 Users should accept Privacy Policy and Terms Of Use when they sign up.
2.4.2 The System shall protect the user data under KVKK and GDPR rules.

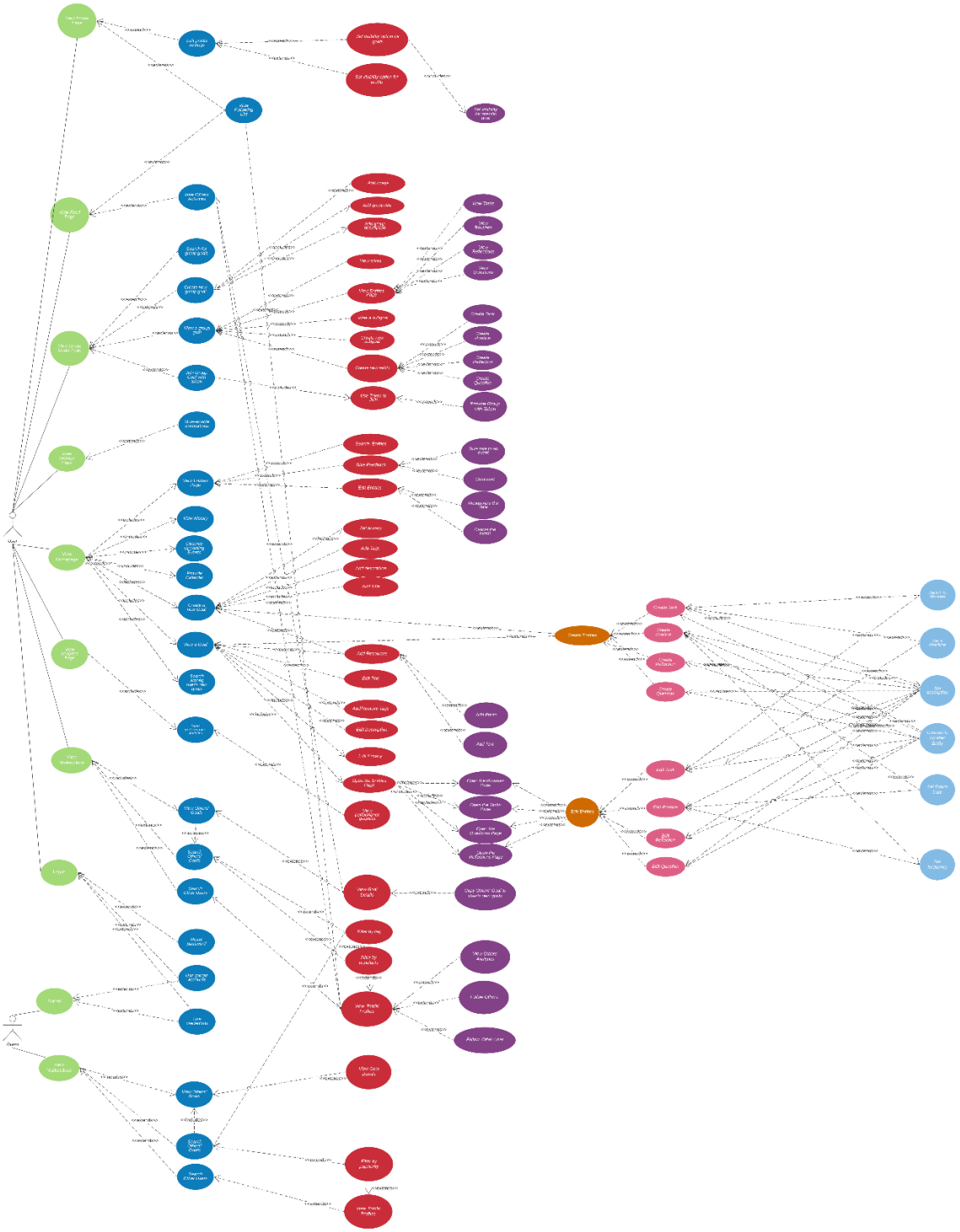
UML SOFTWARE DESIGNS

Class Diagram



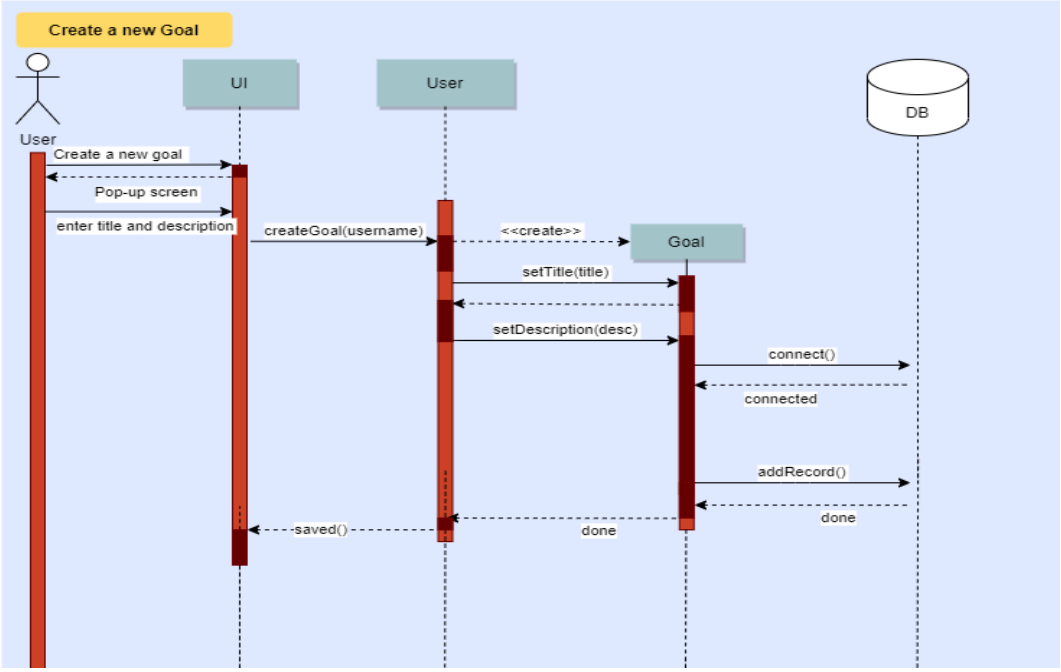
Use Case Diagram

Group12 Use Case Diagram

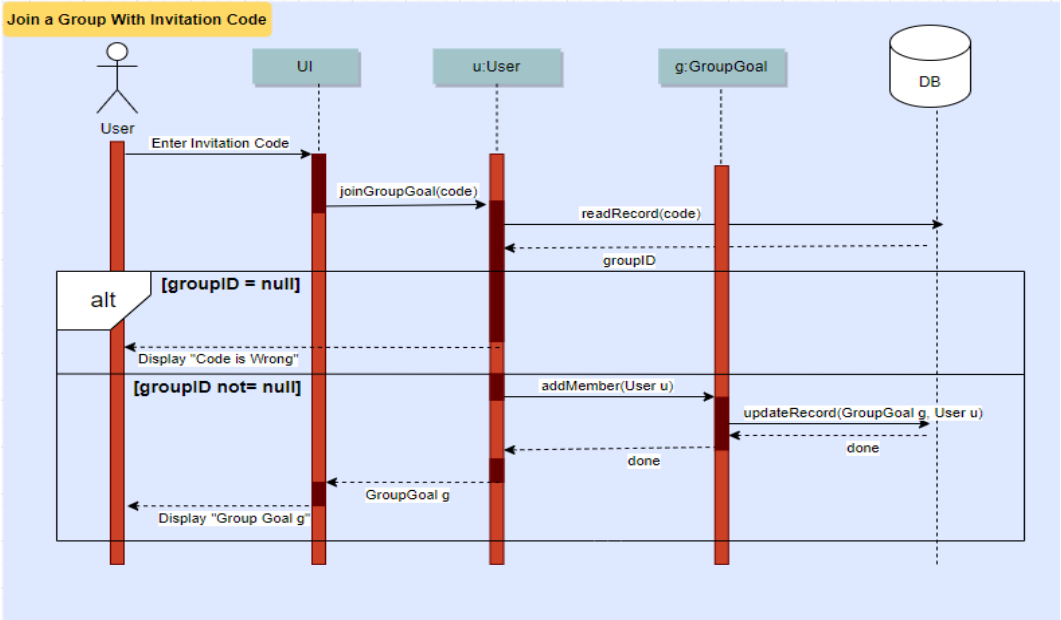


Sequence Diagrams

1. Creating a New Goal

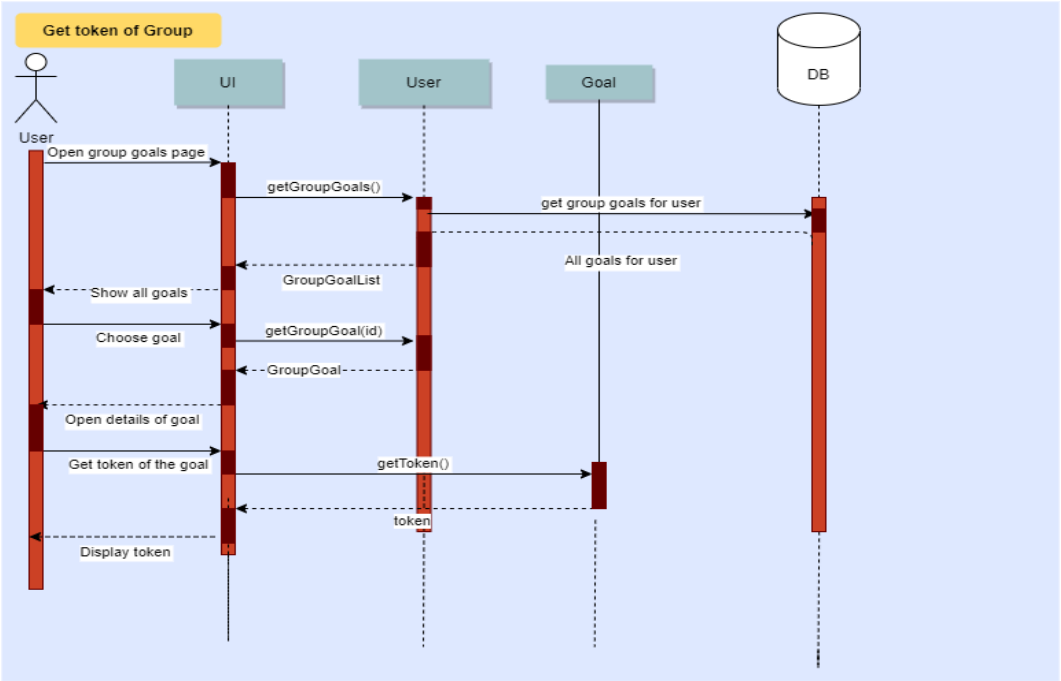


2. Joining a Group With Invitation Code

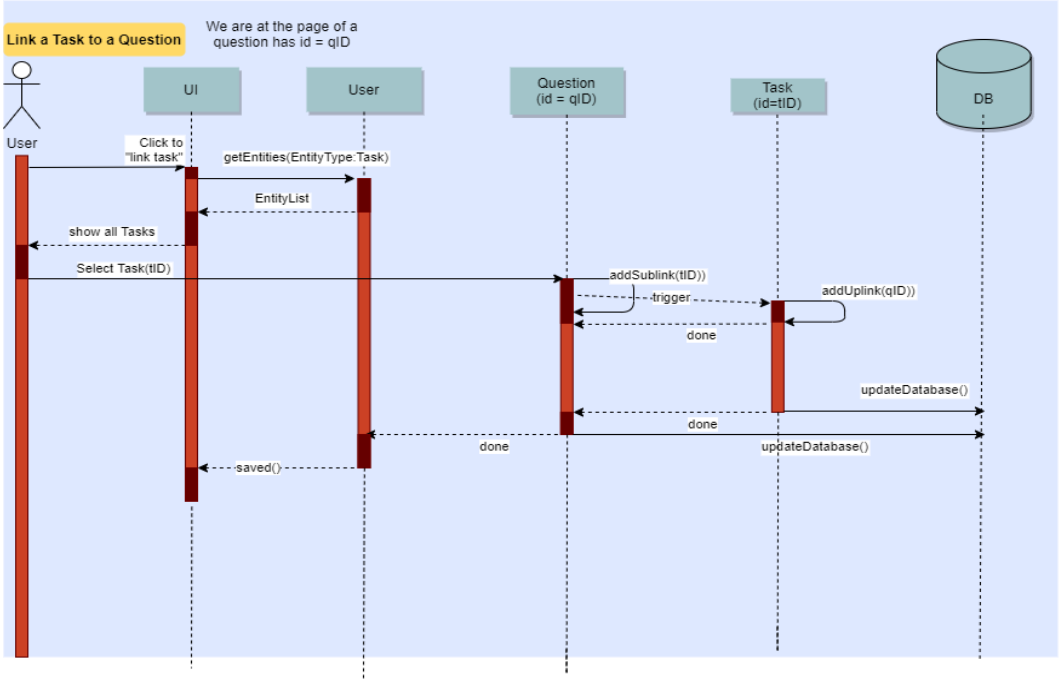


"Invitation Code" and "Token" words are interchangeable. They both represent the same thing.

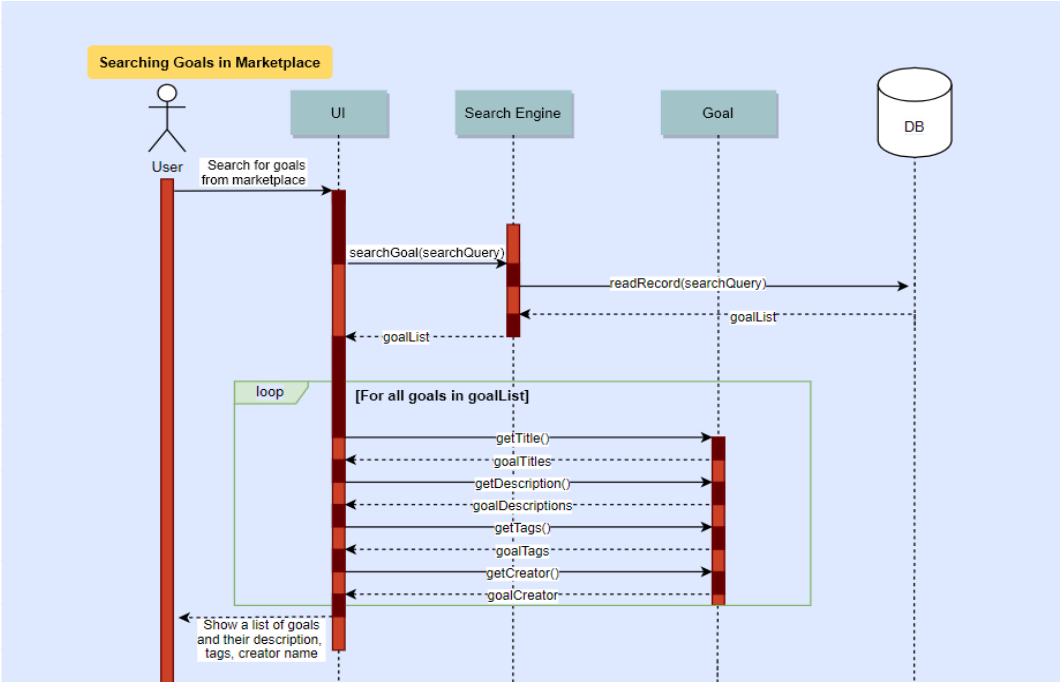
3. Getting token for a group goal



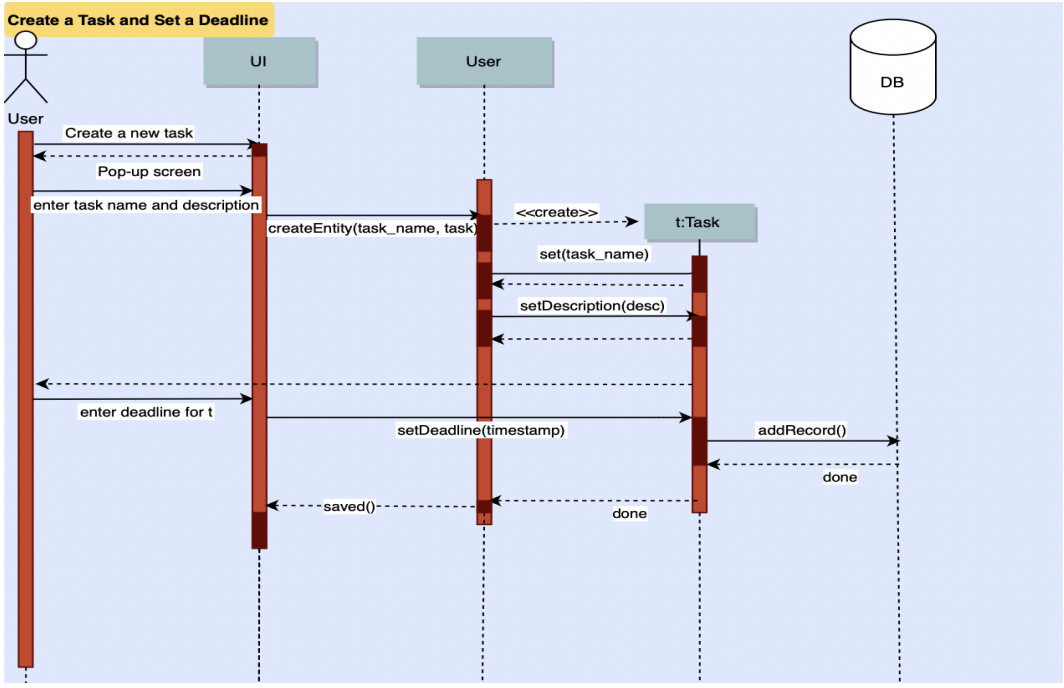
4. Link a Task to a Question



5. Searching Goals in Marketplace

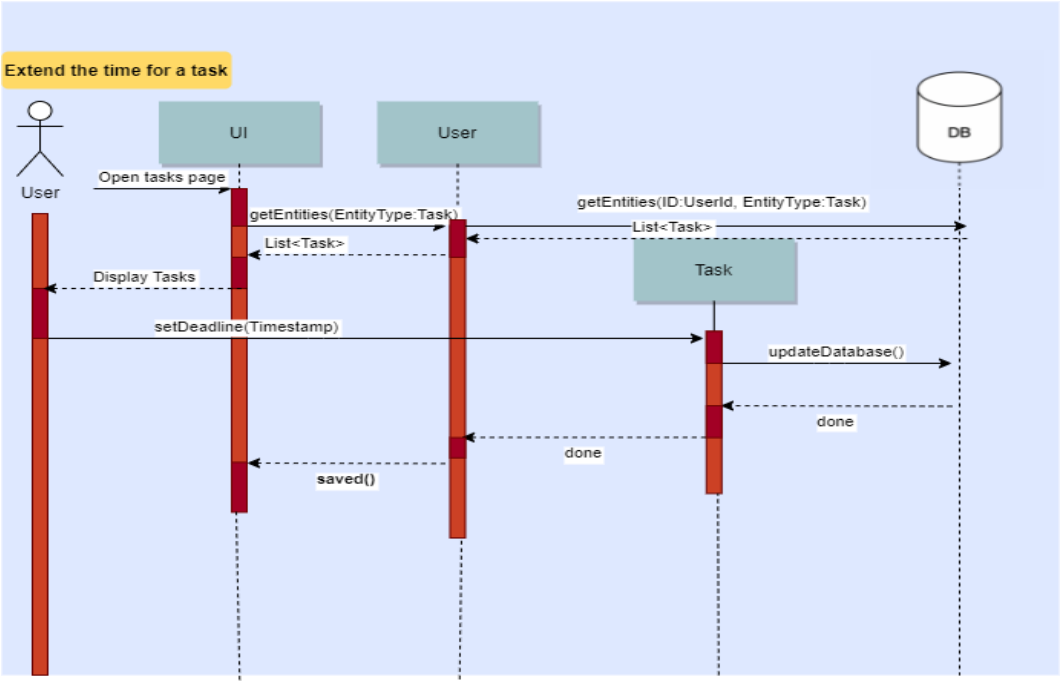


6. Creating a Task and Setting a Deadline

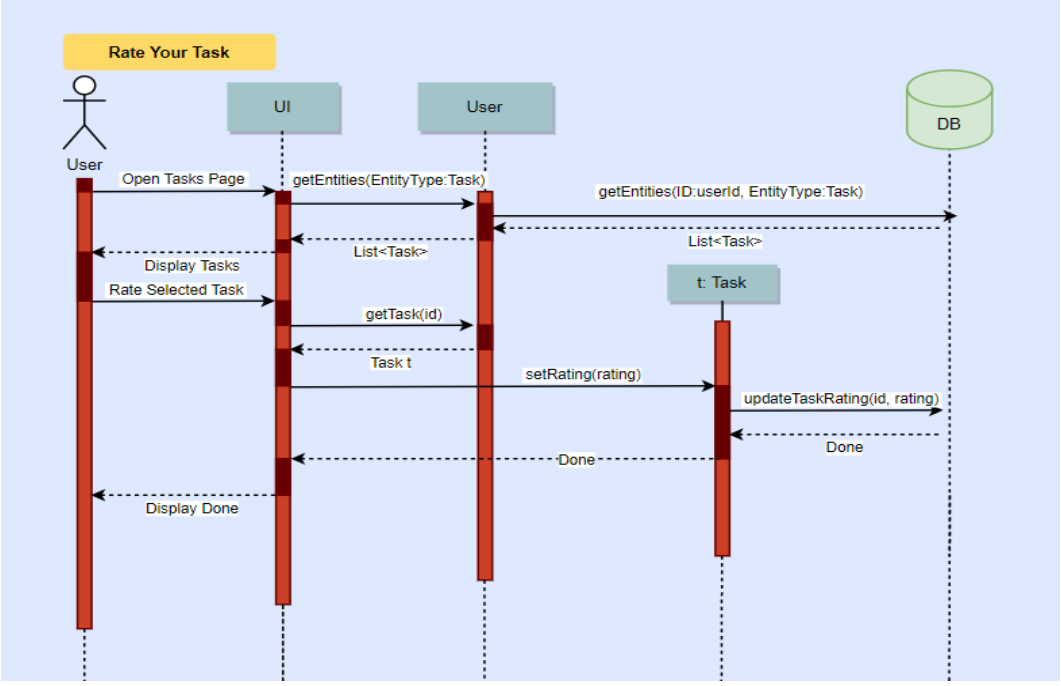


*Creation of a deadline is implemented while creating a task in web design. This diagram is consistent with the mobile design.

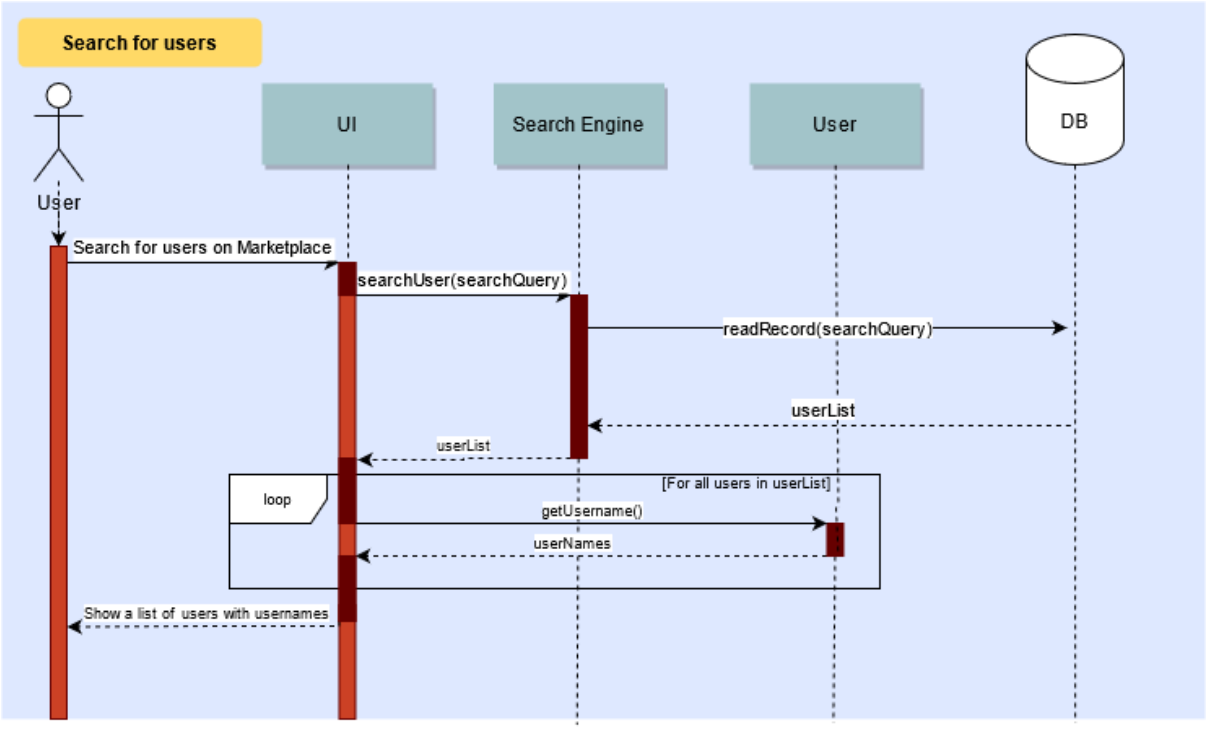
7. Extend Deadline For a Task



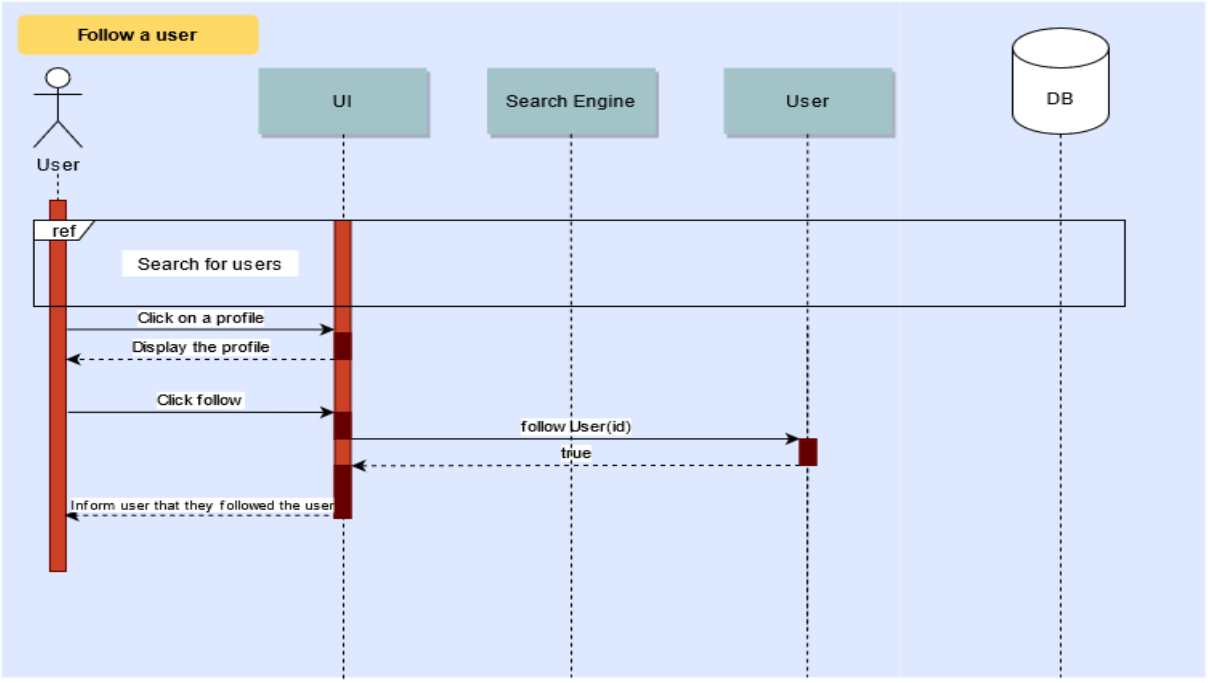
8. Rate a Completed Task



9. Search for Users



10. Follow a User



*draw.io webpage is used for drawings.

RAM

L: Lead S: Secondary C: Contributor A: Approval R: Reviewer N: None	Adalet Veyis Turgut	Burakcan Kazım Yeşil	Ömer Yılmaz	Ahmet Berat Can	İhsan Gökcül	Zahit Batuhan Tongarlak	Kürşat Talha Berk Yamanoğlu	Refika Kalyoncu
Milestone 3	C	C	C	C	C	C	C	C
Implementations								
Backend								
Register Endpoint	L	C	A	A	N	A	A	A
Login Endpoint	L	C	A	A	N	A	A	A
Reset Password Endpoint	L	C	A	A	N	N	N	N
Goal Crud Endpoints	L	A	A	N	N	N	A	A
Entity Crud Enpoints	L	A	N	N	N	A	A	A
Linking Entities Endpoints	L	A	N	N	N	A	A	N
Resources, Labeling Endpoints	N	N	N	N	N	N	N	N
Api Documentation (Swagger)	L	A	N	N	N	A	A	A
Unittests	L	A	N	N	N	N	N	N
Frontend								
Register Page	N	N	R	L	N	R	N	N
Login Page	N	N	R	L	N	R	N	N
Reset Password Page	N	N	R	L	N	R	N	N
Goals Page	N	N	L	R	N	R	N	N
New Goal Page	N	N	L	R	N	R	N	N
New Entity Page	N	N	C	R	N	L	N	N
Subpages in Goals Page	N	N	L	R	N	R	N	N
Edit Entity Page	N	N	R	R	N	L	N	N
Entities in Goal Page	N	N	R	R	N	L	N	N

Android								
Register Page	N	N	N	A	A	N	L	A
Login Page	N	N	N	A	A	N	L	A
Reset Password Page	N	N	N	N	N	N	L	N
Goals Page	N	N	N	N	A	N	L	A
New Goal Page	N	N	N	N	A	N	C	L
New Entity Page	N	N	N	N	L	N	C	L
Subpages in Goals Page	N	N	N	N	A	N	C	L
Edit Entity Page	N	N	N	N	L	N	C	C
Scenario & Mockup								
Frontend	C	N	R	N	L	R	C	N
Android	C	N	R	N	L	R	L	N
Documentations								
Project Plan	L	N	R	N	A	C	C	N
Project Requirements	L	N	N	N	A	R	A	N
UML Software Designs	C	N	N	N	A	N	A	N
Communication Plan	C	N	L	N	C	A	A	N
Deployment								
Frontend	N	N	L	N	N	N	N	N
Backend	R	L	R	N	N	N	N	N
CI/CD Pipeline on Github Actions	R	L	N	N	N	N	N	N

SCENARIOS and MOCKUPS

Scenario

Our scenario is a composite scenario which combines both android and frontend scenarios in a single scenario.

User and Persona

James Bond
Bootblack at Government Building
28 years old
High-school Graduate

Story

James wants to be more intellectual. James is not an organized person. He failed his attempts on his self development. James' friend Albert recommends James a new website called BeABee to track his development. He searches on the internet and finds the website. James don't have an account on BeABee

Goals

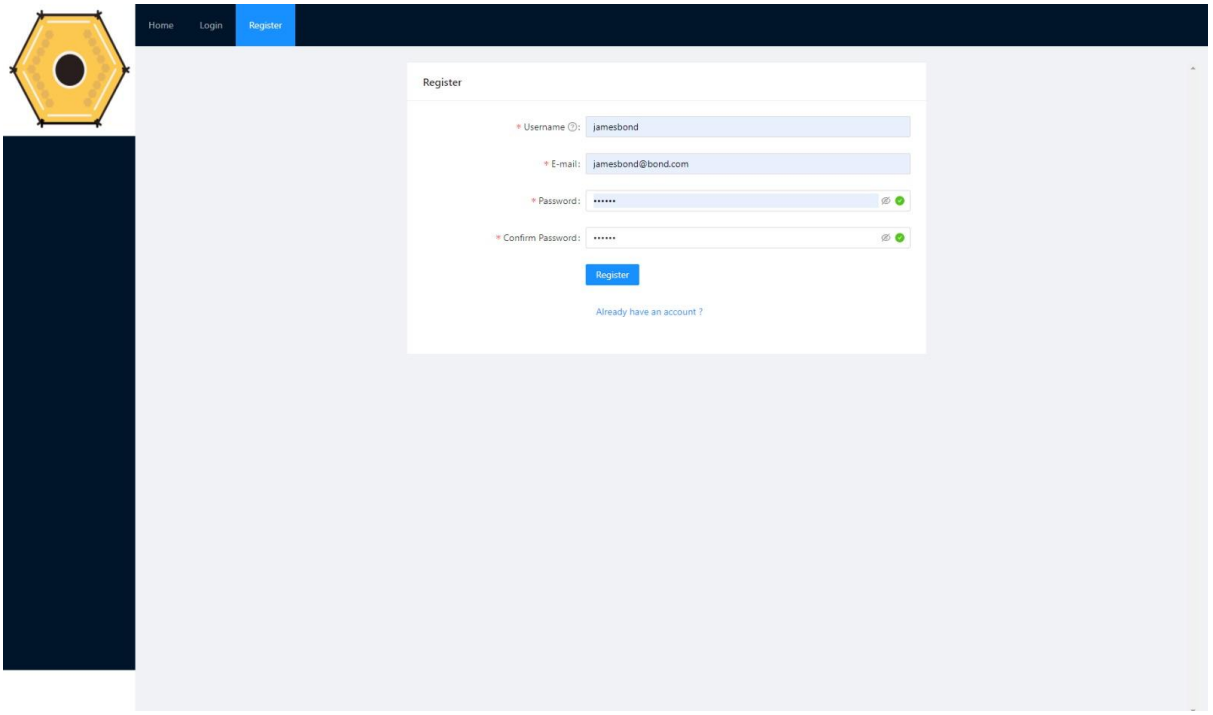
- James wants to become a more organized person
- James wants to become more intellectual

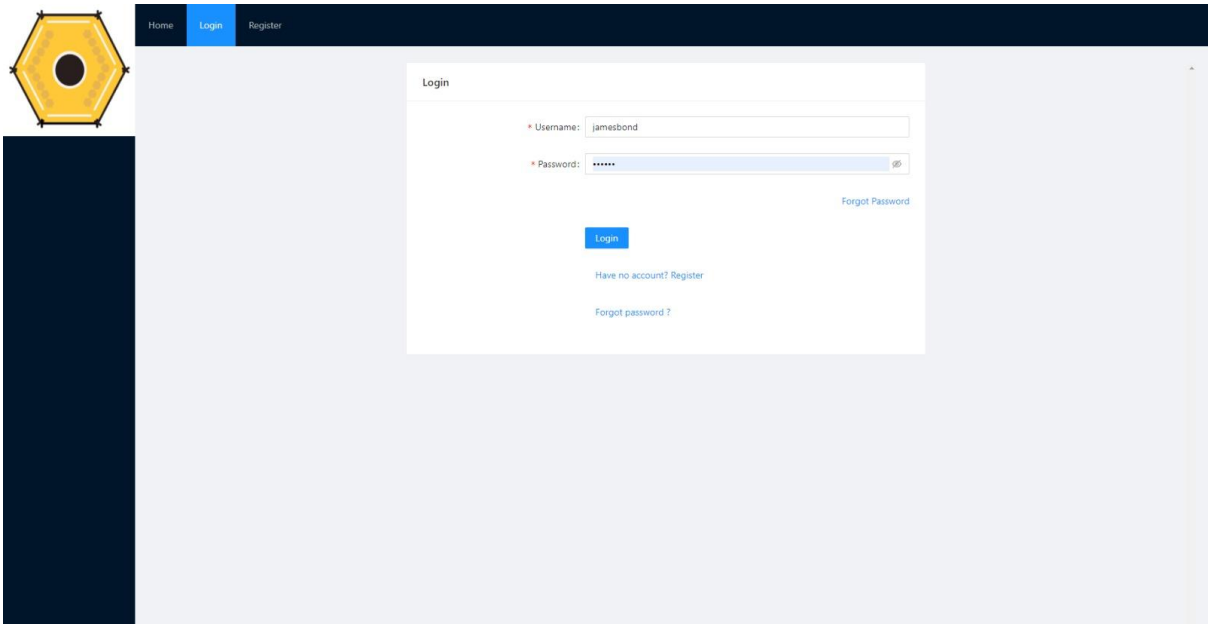
Scenario

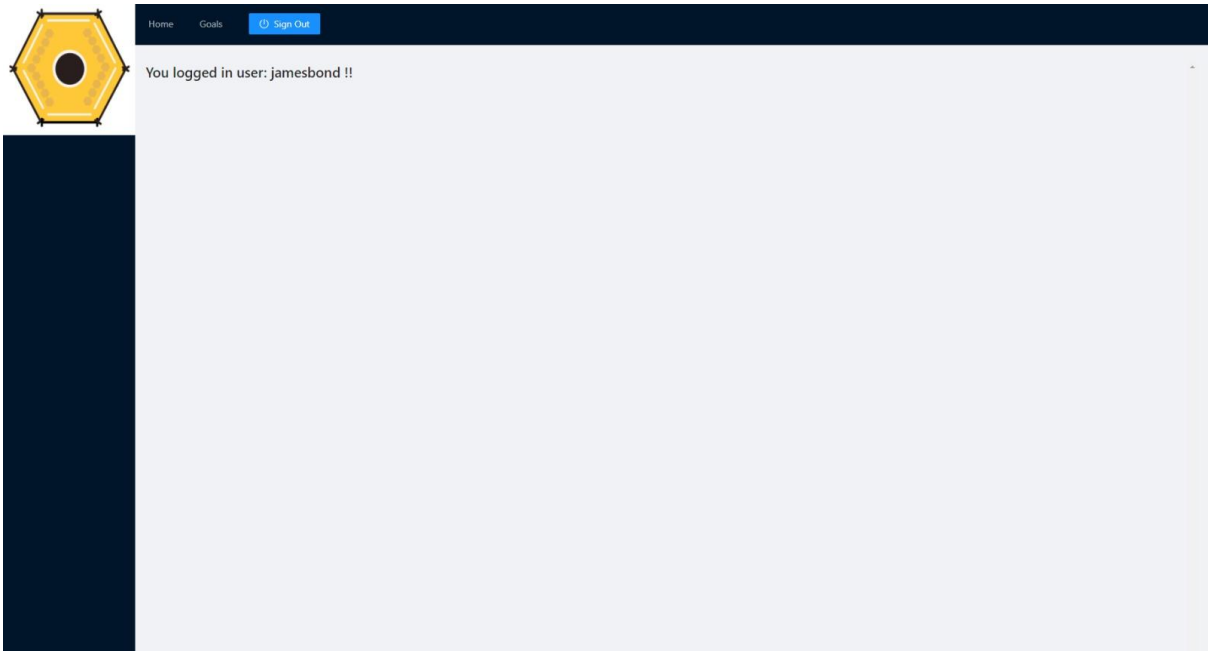
- James fills Username, mail, password and password confirmation fields
- Then he clicks the signup button and sees successful signup
- He enters his Username and password on login page
- He clicks the login button and sees successful login
- He wants to add a new goal. On main page, he clicks goal and add a new goal using add goal button
- He writes "Being more intellectual" to goal title and " I will be more intellectual. I can do it!" to goal description
- He clicks submit button
- He wants to add a new subgoal to his goal. Then he clicks create entity button
- He writes " Reading Books" to entity title and "Reading will improve myself." to entity description
- He selects subgoal for entity type and clicks submit button.
- Then he wants to add a new task. He clicks create entity button
- He writes " Reading Fahrenheit 451" to entity title and " I should read this book. My friend Albert recommended this." to entity description
- He selects task for entity type and clicks submit button.

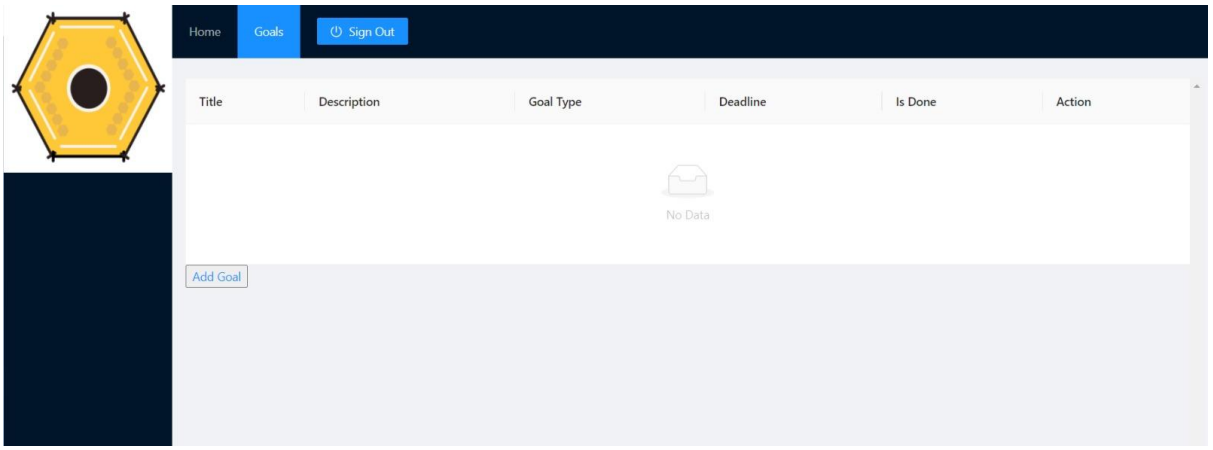
- Then he wants to add another task. He clicks create entity button
- He writes “ Write a summary of this book” to entity title and “ Writing a summary improves understanding” to entity description
- He selects task for entity type and clicks submit button.
- Then he wants to add a new routine. He clicks create entity button
- He writes “ Read daily morning briefing of The New York Times.” to entity title and “ There is a "What to read?" section in it. It will help me to staying up to date with new books.” to entity description
- He selects routine for entity type and clicks submit button.
- Then he wants to add a new question. He clicks create entity button
- He writes “ What are other books that I should read?” to entity title and “ I can ask my best friend Albert for more book recommendations. He is a bookworm. ” to entity description
- He selects question for entity type and clicks submit button.
- Then he wants to add a new reflection. He clicks create entity button
- He writes “I have read 13 books until now.” to entity title and “ I have read only 13 books last year. I think this app will help me to read much more than last year.” to entity description
- He selects reflection for entity type and clicks submit button.
- Then he got an urgent call and leaves the house.
- He downloads BeABee app to continue his work.
- He enters his username and password
- He clicks the login button and sees successful login
- He sees his the goal that he created earlier on the main page.
- He clicks on the goal.
- He clicks his Subgoal
- He links his First Task to the Subgoal
- Then he clicks on the Task and links the Second Task to the First Task
- He makes edits on the Second Task and make it done.
- Later he checks the website and sees its marked as done.

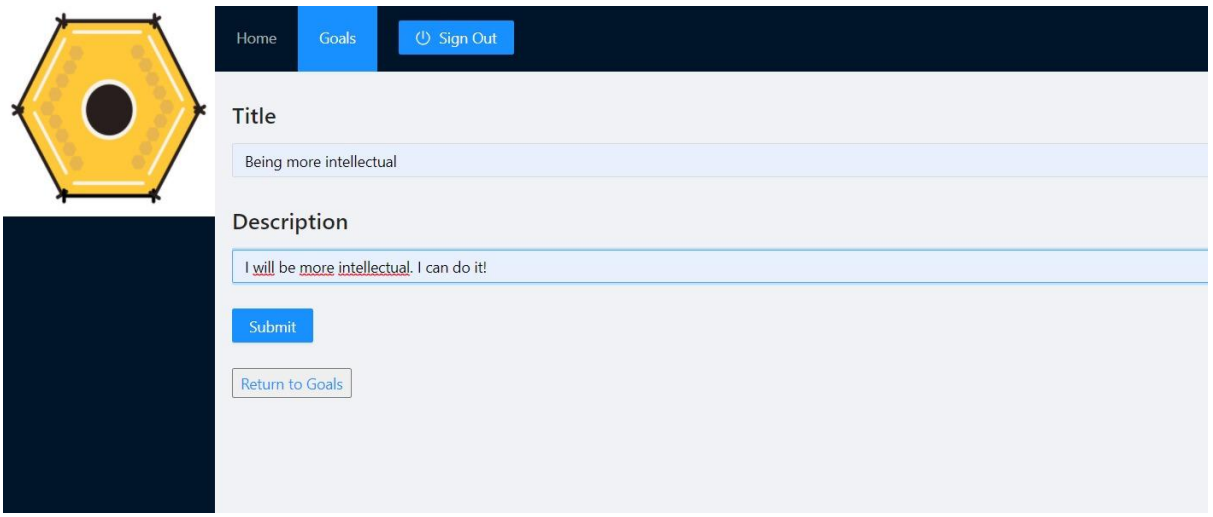
Mockups

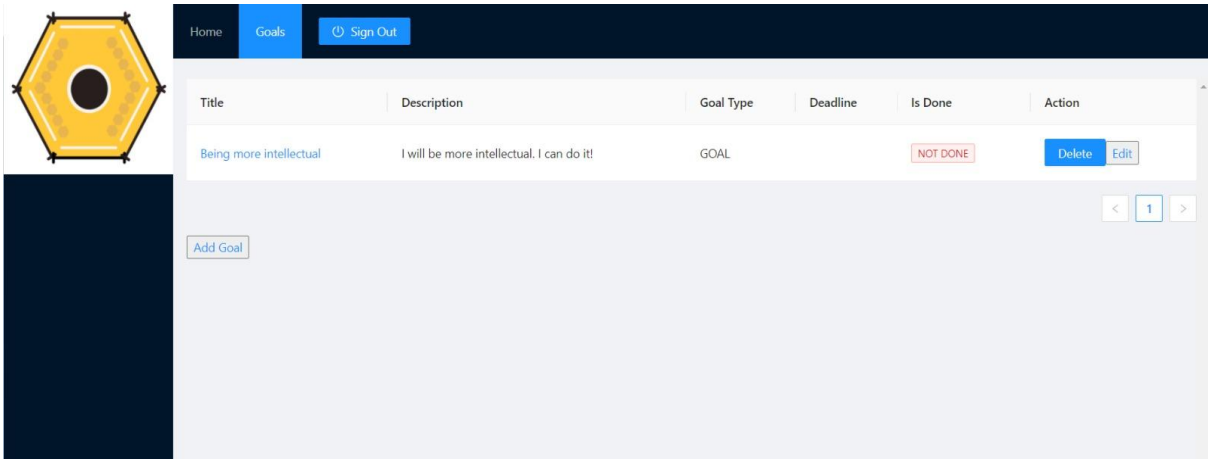

Home
Login
Register
Register
* Username ⓘ:
jamesbond
* E-mail:
jamesbond@bond.com
* Password:
* Confirm Password:
Register
Already have an account ?

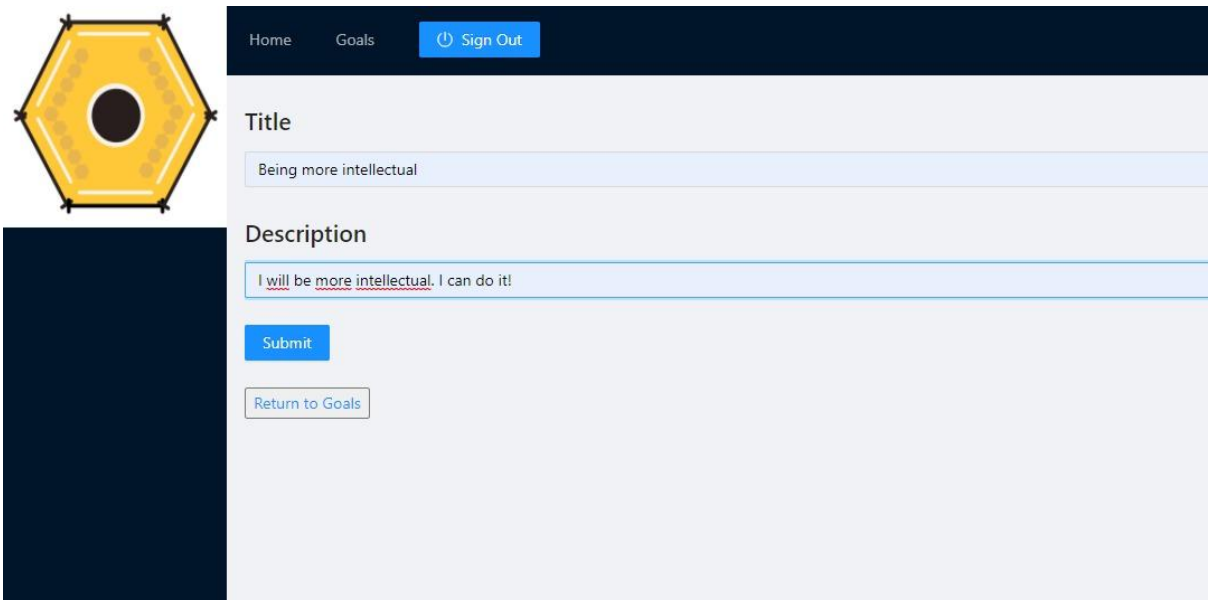

Home
Login
Register
Login
* Username:
jamesbond
* Password:
Forgot Password
Login
Have no account? Register
Forgot password ?

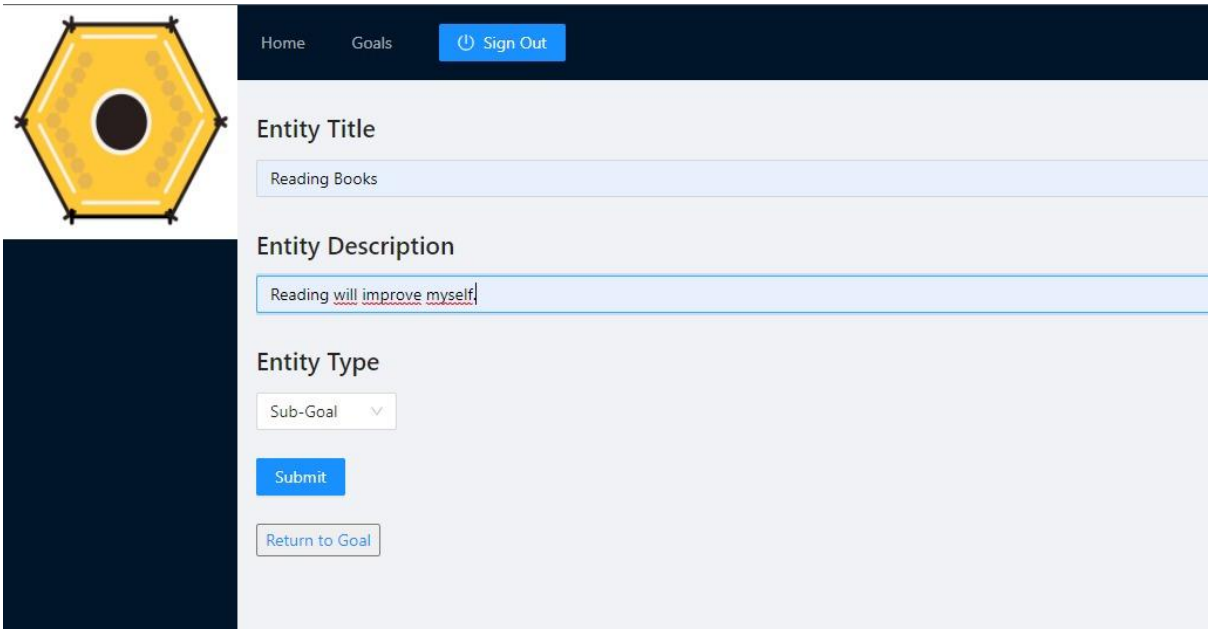

Home
Goals
Sign Out
You logged in user: jamesbond !!

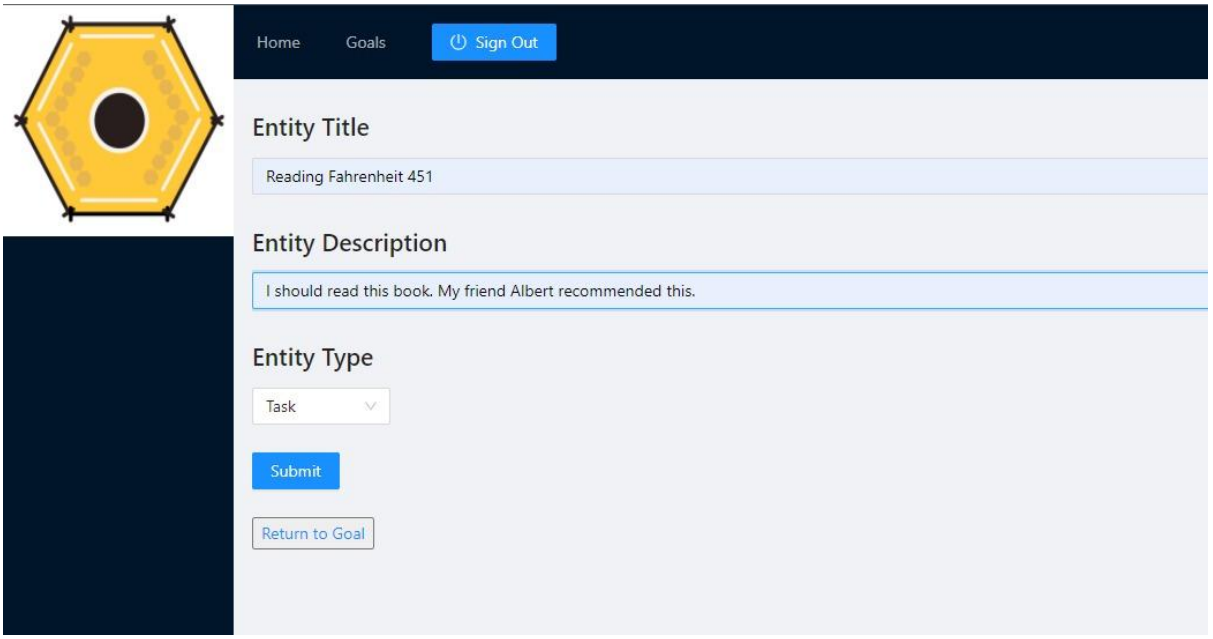

Home
Goals
Sign Out
Title
Description
Goal Type
Deadline
Is Done
Action
No Data
Add Goal

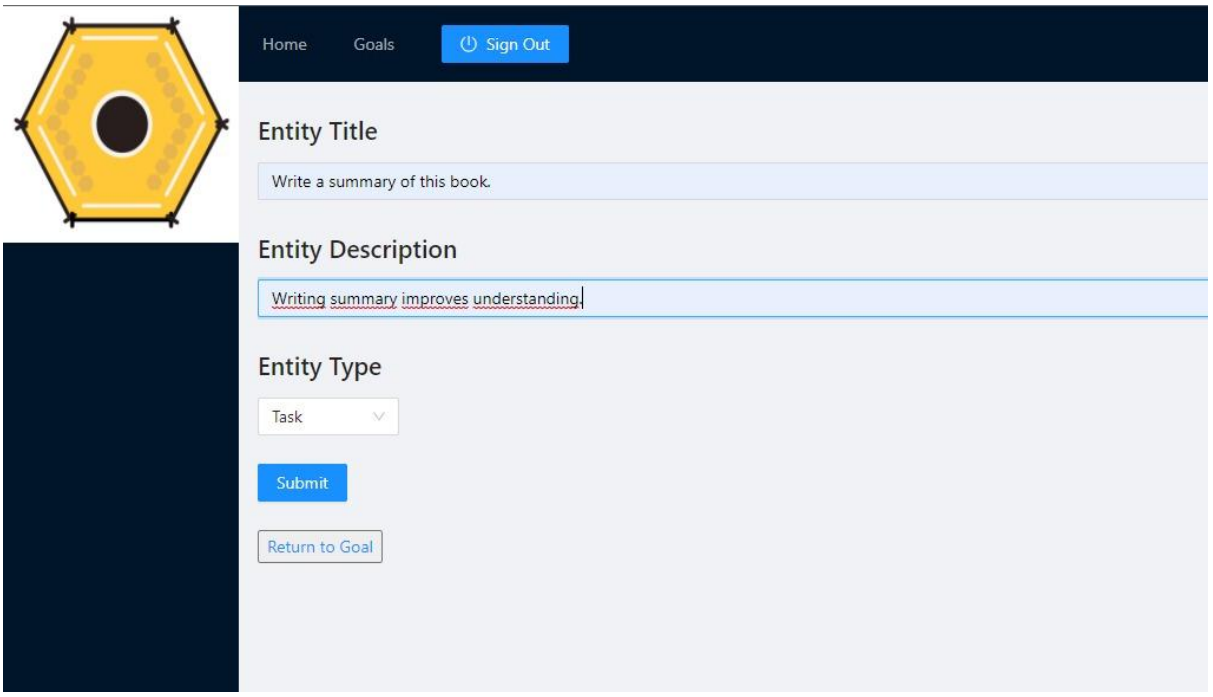

Home
Goals
Sign Out
Title
Being more intellectual
Description
I will be more intellectual. I can do it!
Submit
Return to Goals

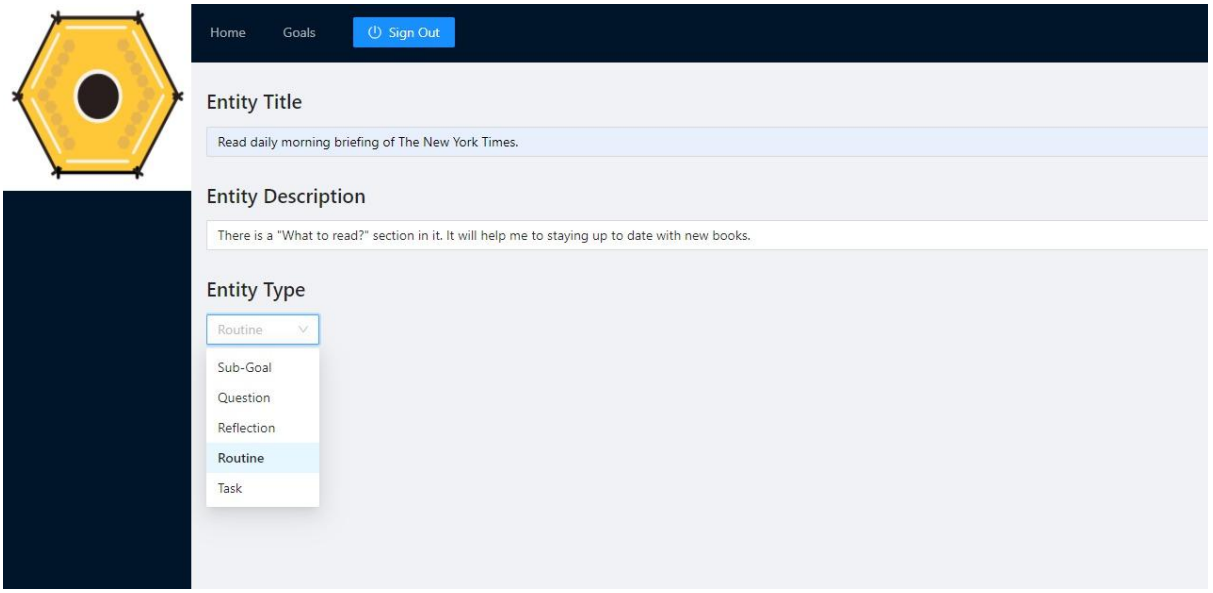

Home
Goals
Sign Out
Title
Description
Goal Type
Deadline
Is Done
Action
Being more intellectual
I will be more intellectual. I can do it!
GOAL
NOT DONE
Delete
Edit
1
Add Goal

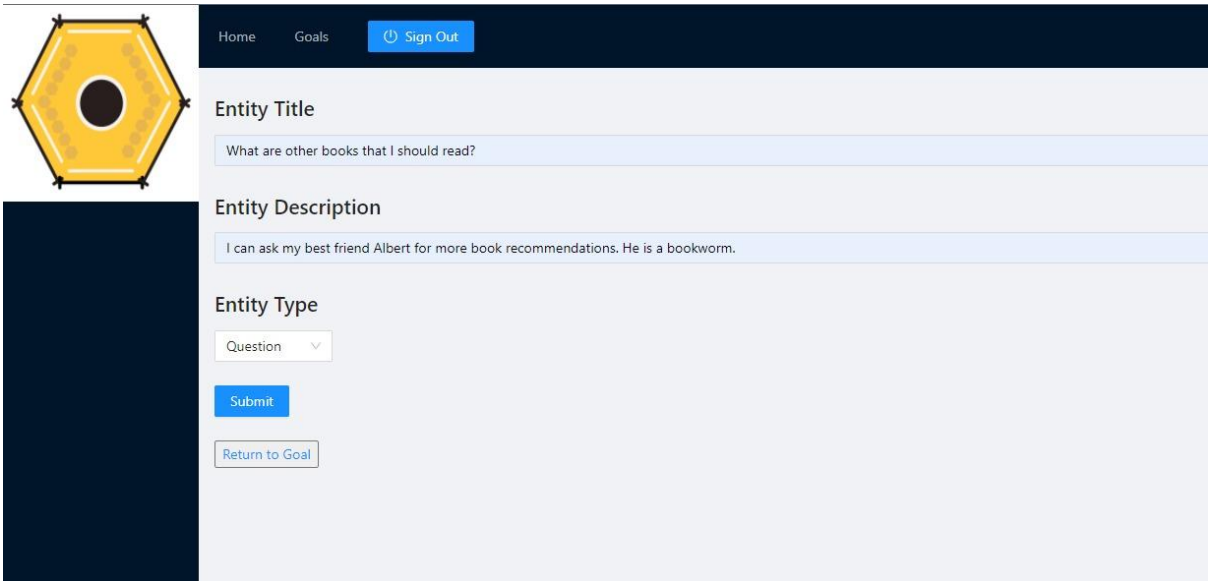

Home
Goals
Sign Out
Title
Being more intellectual
Description
I will be more intellectual. I can do it!
Submit
Return to Goals

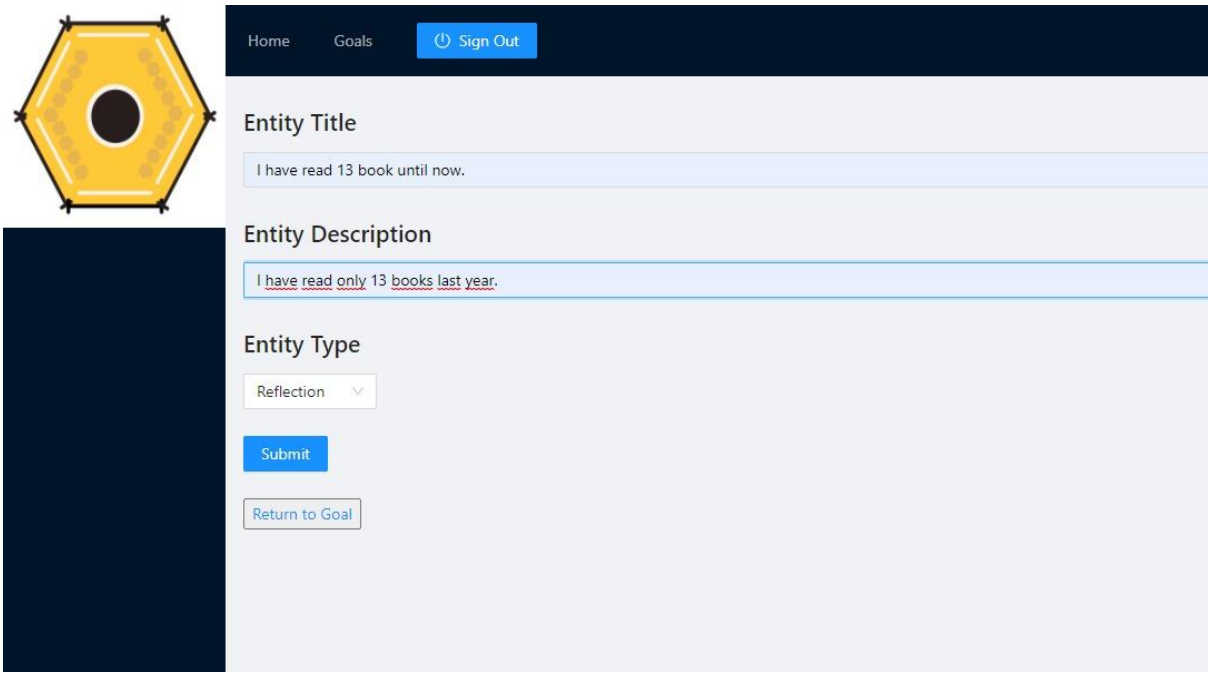

Home
Goals
Sign Out
Entity Title
Reading Books
Entity Description
Reading will improve myself.
Entity Type
Sub-Goal
Submit
Return to Goal

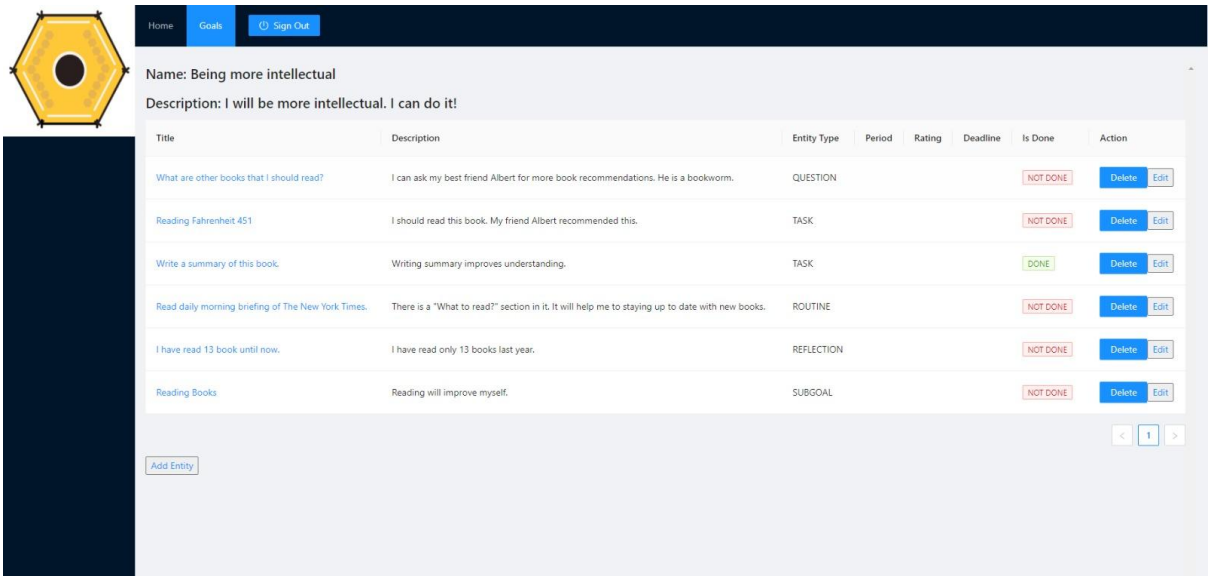

Home
Goals
Sign Out
Entity Title
Reading Fahrenheit 451
Entity Description
I should read this book. My friend Albert recommended this.
Entity Type
Task
Submit
Return to Goal

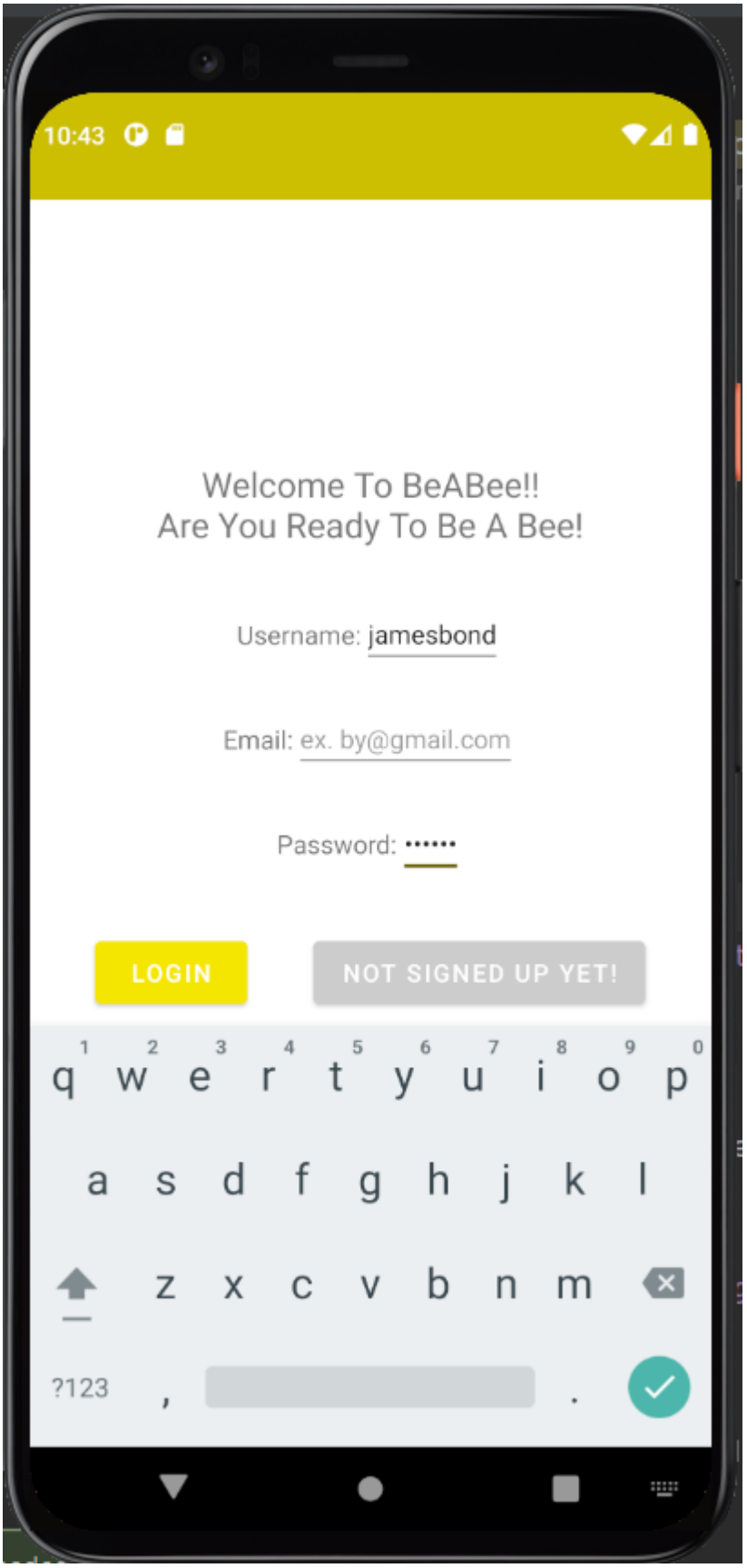

Home
Goals
Sign Out
Entity Title
Write a summary of this book.
Entity Description
Writing summary improves understanding.
Entity Type
Task
Submit
Return to Goal

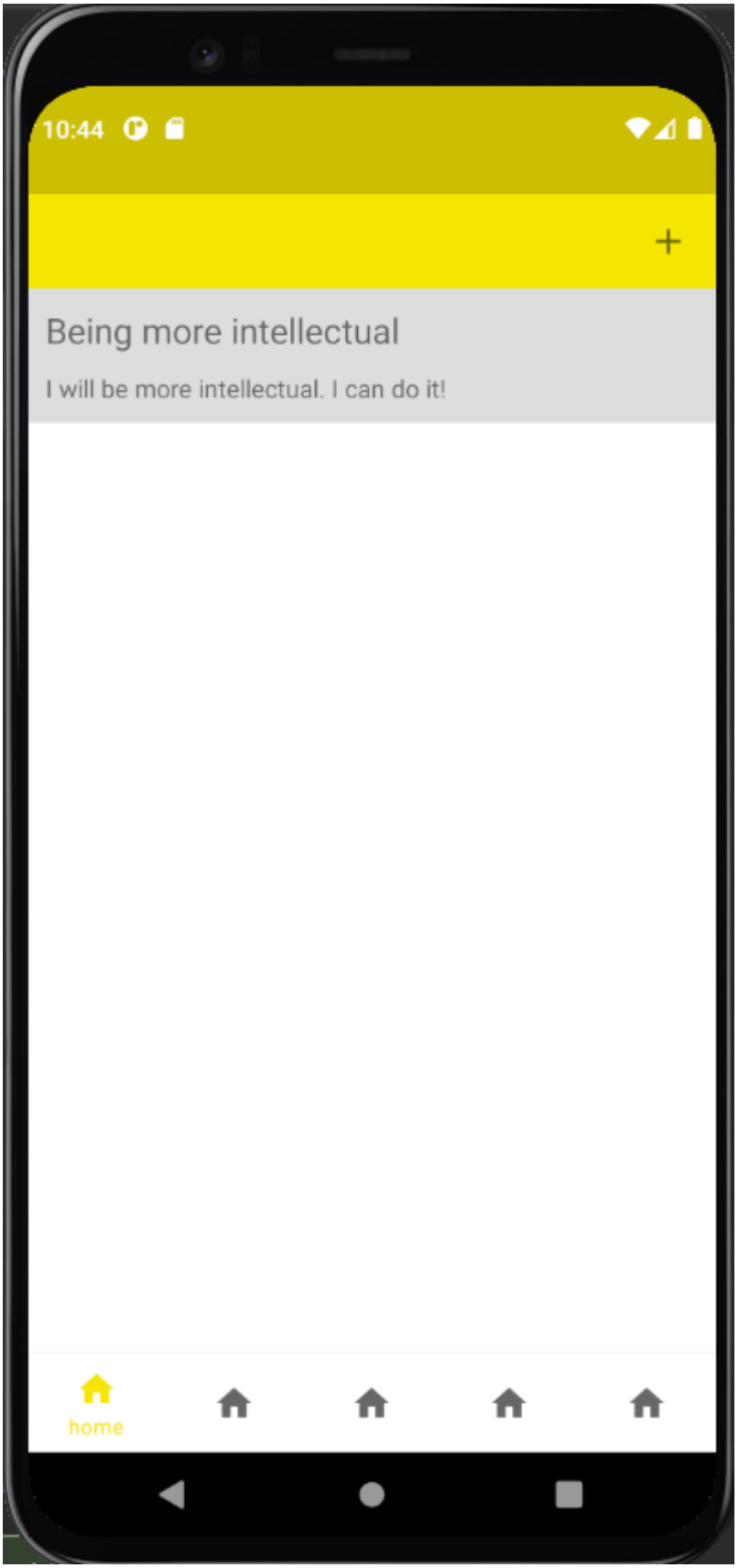

Home
Goals
Sign Out
Entity Title
Read daily morning briefing of The New York Times.
Entity Description
There is a "What to read?" section in it. It will help me to staying up to date with new books.
Entity Type
Routine
Sub-Goal
Question
Reflection
Routine
Task

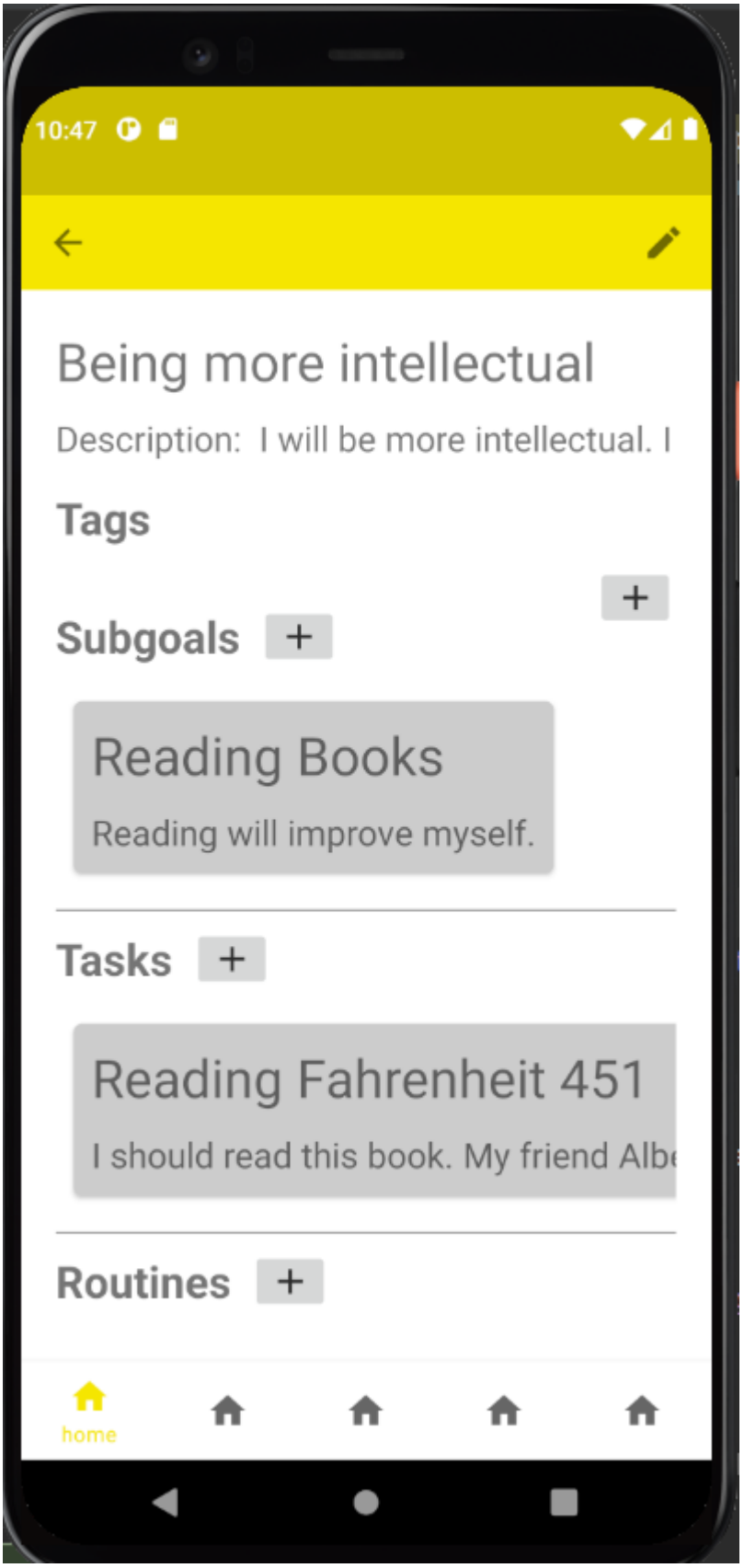

Home
Goals
Sign Out
Entity Title
What are other books that I should read?
Entity Description
I can ask my best friend Albert for more book recommendations. He is a bookworm.
Entity Type
Question
Submit
Return to Goal

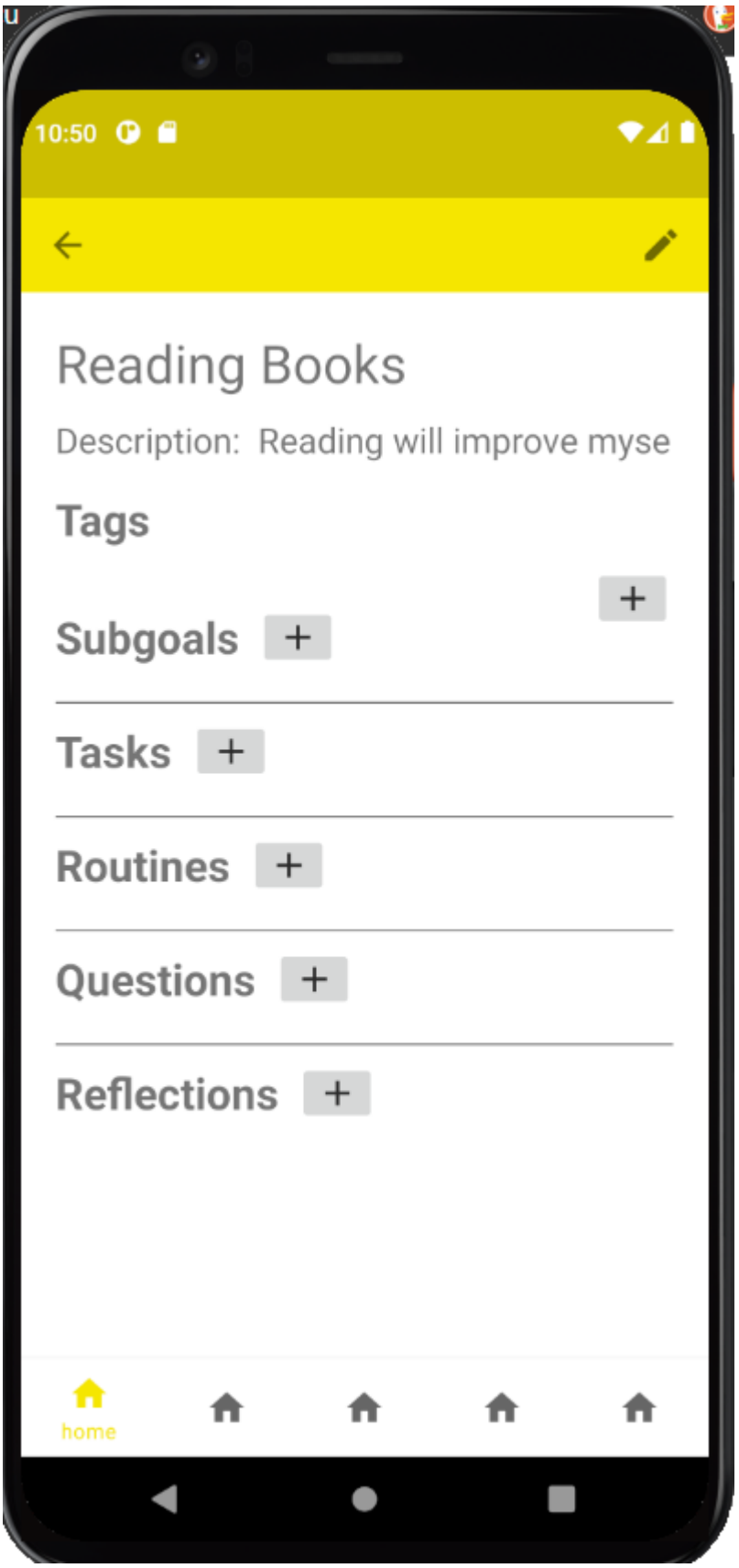

Home
Goals
Sign Out
Entity Title
I have read 13 book until now.
Entity Description
I have read only 13 books last year.
Entity Type
Reflection
Submit
Return to Goal

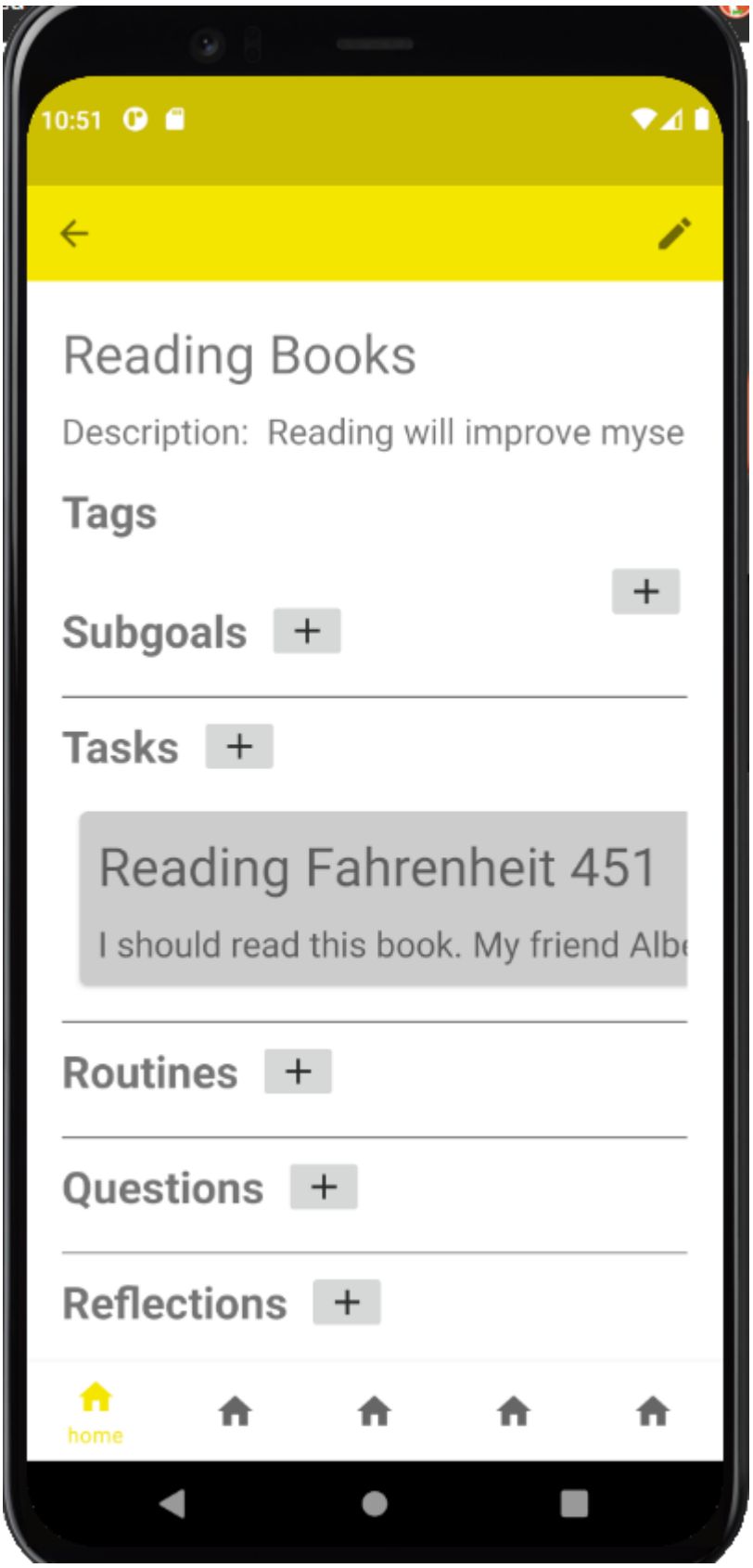

Home
Goals
Sign Out
Name: Being more intellectual
Description: I will be more intellectual. I can do it!
Title
Description
Entity Type
Period
Rating
Deadline
Is Done
Action
What are other books that I should read?
I can ask my best friend Albert for more book recommendations. He is a bookworm.
QUESTION
NOT DONE
Delete
Edit
Reading Fahrenheit 451
I should read this book. My friend Albert recommended this.
TASK
NOT DONE
Delete
Edit
Write a summary of this book.
Writing summary improves understanding.
TASK
DONE
Delete
Edit
Read daily morning briefing of The New York Times.
There is a "What to read?" section in it. It will help me to staying up to date with new books.
ROUTINE
NOT DONE
Delete
Edit
I have read 13 book until now.
I have read only 13 books last year.
REFLECTION
NOT DONE
Delete
Edit
Reading Books
Reading will improve myself.
SUBGOAL
NOT DONE
Delete
Edit
1
Add Entity

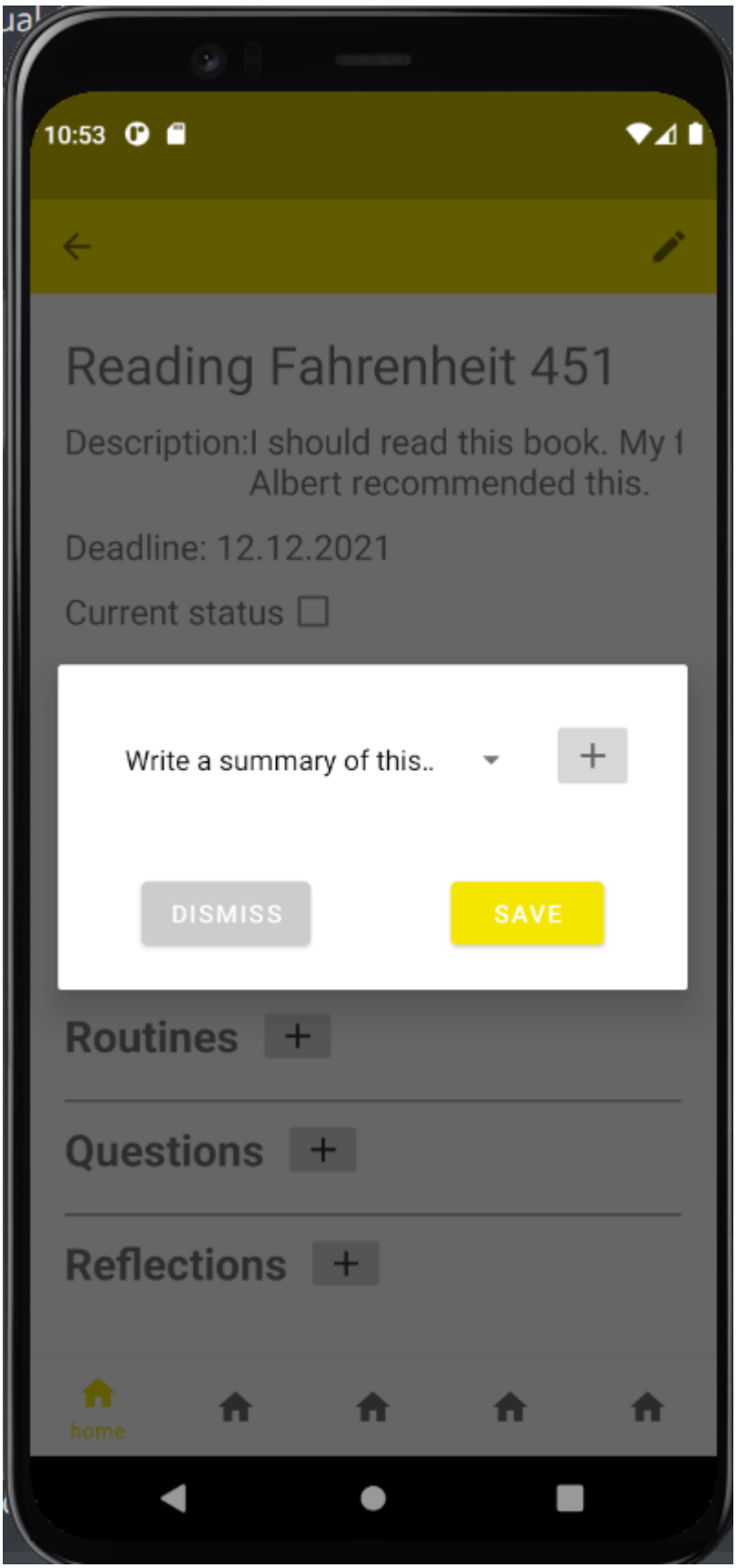

10:43
Welcome To BeABee!!
Are You Ready To Be A Bee!
Username: jamesbond
Email: ex. by@gmail.com
Password: ••••••
LOGIN
NOT SIGNED UP YET!
q w e r t y u i o p
a s d f g h j k l
z x c v b n m
?123

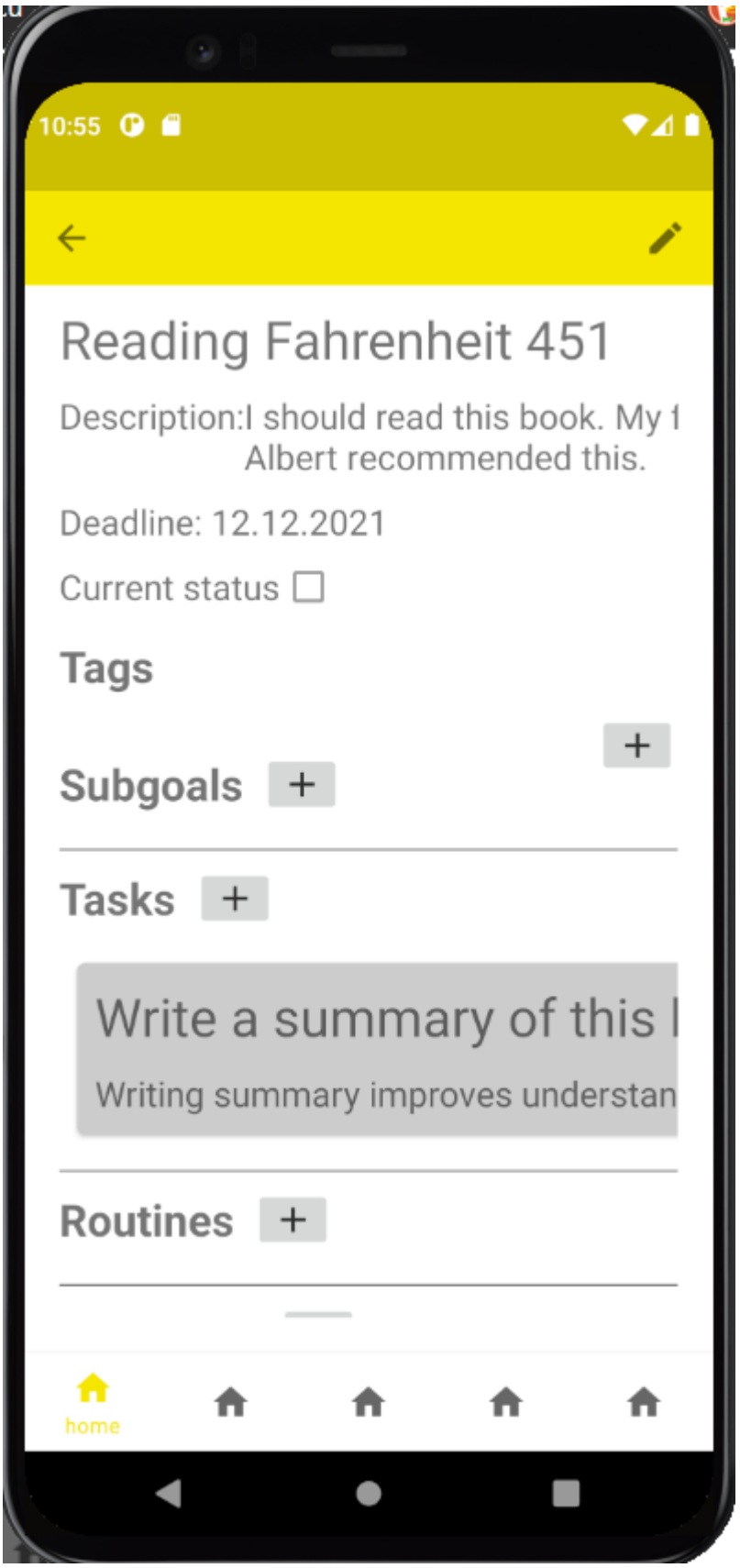

10:44
Being more intellectual
I will be more intellectual. I can do it!
home

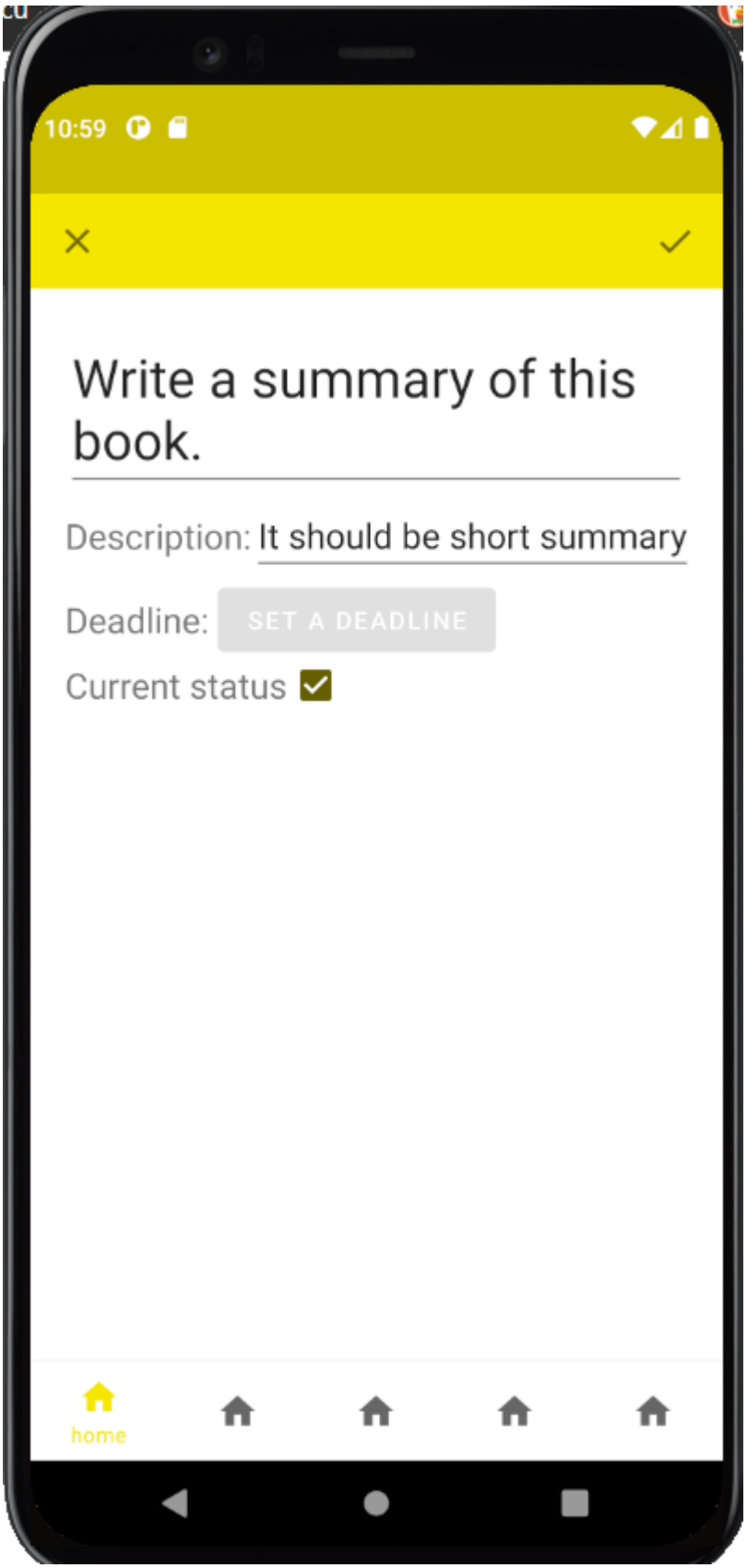

10:47
Being more intellectual
Description: I will be more intellectual. I
Tags
Subgoals
Reading Books
Reading will improve myself.
Tasks
Reading Fahrenheit 451
I should read this book. My friend Albe
Routines
home

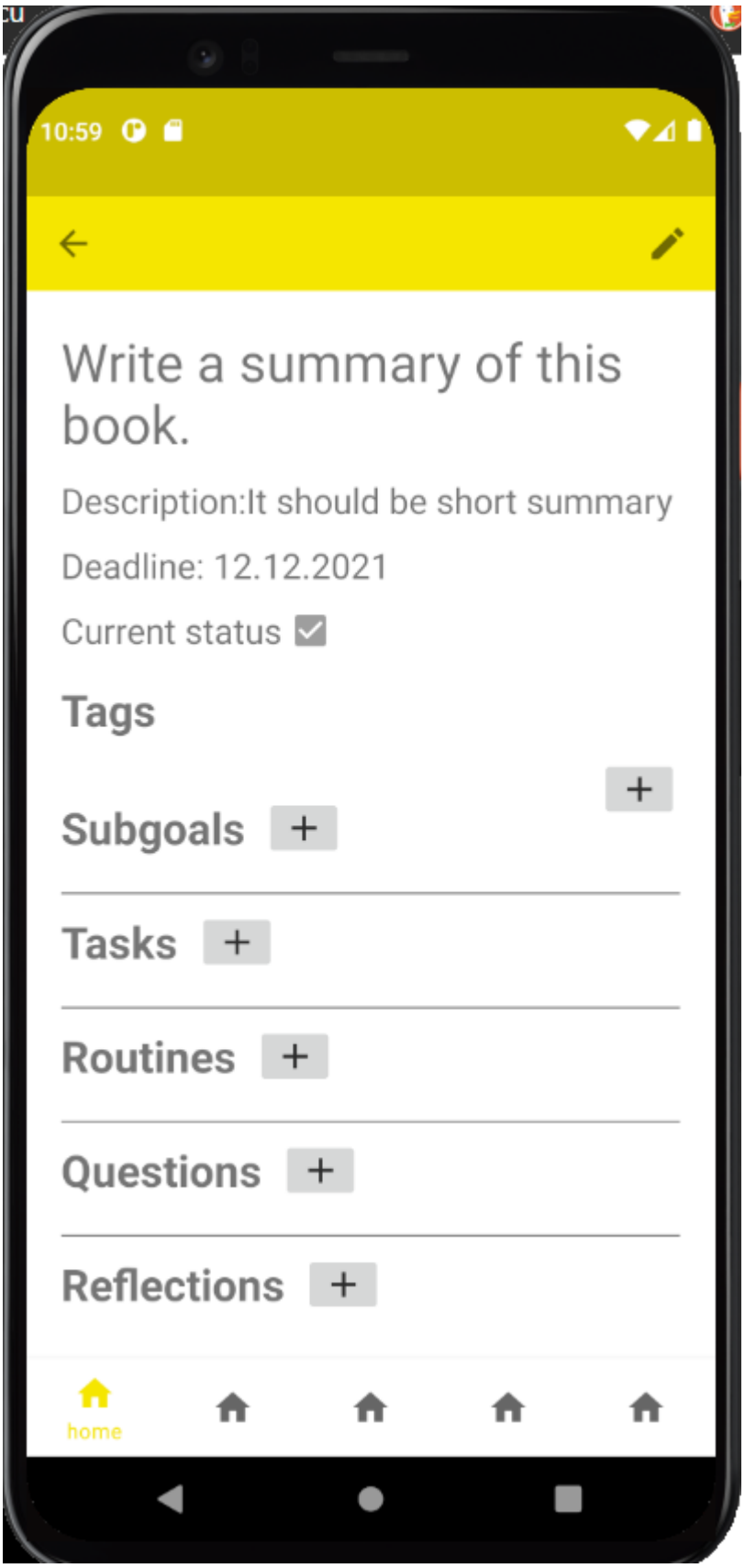

10:50
Reading Books
Description: Reading will improve myse
Tags
Subgoals
Tasks
Routines
Questions
Reflections
home


10:51
Reading Books
Description: Reading will improve myse
Tags
Subgoals
Tasks
Reading Fahrenheit 451
I should read this book. My friend Albe
Routines
Questions
Reflections
home


10:53
Reading Fahrenheit 451
Description:I should read this book. My 1
Albert recommended this.
Deadline: 12.12.2021
Current status
Write a summary of this..
DISMISS
SAVE
Routines
Questions
Reflections
home


10:55
Reading Fahrenheit 451
Description:I should read this book. My f
Albert recommended this.
Deadline: 12.12.2021
Current status
Tags
Subgoals
Tasks
Write a summary of this
Writing summary improves understan
Routines
home


10:59
Write a summary of this book.
Description: It should be short summary
Deadline:
SET A DEADLINE
Current status
home


10:59
Write a summary of this book.
Description:It should be short summary
Deadline: 12.12.2021
Current status
Tags
Subgoals
Tasks
Routines
Questions
Reflections
home

PROJECT PLAN

		Name	Duration	Start	Finish	Predeces...	Resource Names
1		**Orientation**	**3.5 days?**	**3/26/21, 8:00 AM**	**3/29/21, 1:00 PM**		
2		First group meeting	0.056 days?	3/26/21, 8:00 AM	3/26/21, 8:26 AM		Adalet Veyis Turgut;Batuhan Tongarlak;Burakcan Kazm Yeil;Doukan Türksoy;Kürat Talha Berk Yamanolu;Mehmet ...
3		Researching about version control system	0.222 days?	3/26/21, 8:26 AM	3/26/21, 10:13 AM	2	Adalet Veyis Turgut;Batuhan Tongarlak;Burakcan Kazm Yeil;Doukan Türksoy;Kürat Talha Berk Yamanolu;Mehmet ...
4		Researching favourite GitHub repositories	0.222 days?	3/26/21, 8:26 AM	3/26/21, 10:13 AM	2	Adalet Veyis Turgut;Batuhan Tongarlak;Burakcan Kazm Yeil;Doukan Türksoy;Kürat Talha Berk Yamanolu;Mehmet ...
5		Researching about markdown	0.222 days?	3/26/21, 8:26 AM	3/26/21, 10:13 AM	2	Adalet Veyis Turgut;Batuhan Tongarlak;Burakcan Kazm Yeil;Doukan Türksoy;Kürat Talha Berk Yamanolu;Mehmet ...
6		Creating Slack channel	0.5 days?	3/27/21, 8:00 AM	3/29/21, 1:00 PM	2	Adalet Veyis Turgut
7		Creating a discord channel	0.5 days?	3/27/21, 8:00 AM	3/29/21, 1:00 PM	2	Refika Kalyoncu
8		**Documentation**	**1.5 days**	**3/29/21, 1:00 PM**	**3/30/21, 5:00 PM**	**1**	
9		Creating wiki home page	0 days	3/29/21, 1:00 PM	3/29/21, 1:00 PM		Adalet Veyis Turgut[50%];Burakcan Kazm Yeil;Refika Kalyoncu[50%]
10		Creating wiki sidebar	0 days	3/29/21, 1:00 PM	3/29/21, 1:00 PM		Adalet Veyis Turgut[50%];Batuhan Tongarlak[50%];Kürat Talha Berk Yamanolu;Refika Kalyoncu[50%]
11		Creating readme.md file	0 days	3/29/21, 1:00 PM	3/29/21, 1:00 PM		Mehmet Gökay Yldz
12		Creating personal wiki pages	1.081 days	3/29/21, 1:00 PM	3/30/21, 1:39 PM		Adalet Veyis Turgut;Batuhan Tongarlak;Burakcan Kazm Yeil;Doukan Türksoy;Kürat Talha Berk Yamanolu;Mehmet ...
13		Creating personal effort pages	1.081 days	3/29/21, 1:00 PM	3/30/21, 1:39 PM		Adalet Veyis Turgut;Batuhan Tongarlak;Burakcan Kazm Yeil;Doukan Türksoy;Kürat Talha Berk Yamanolu;Mehmet ...
14		Preparing Communication Plan	1.5 days	3/29/21, 1:00 PM	3/30/21, 5:00 PM		Ömer Ylmaz
15		Customizing issue labels	0 days	3/29/21, 1:00 PM	3/29/21, 1:00 PM		Adalet Veyis Turgut
16		Creating favourite Github Repositories page	0.2 days	3/29/21, 1:00 PM	3/29/21, 2:36 PM		Adalet Veyis Turgut;Doukan Türksoy
17		Creating version control system research page	1.077 days	3/29/21, 1:00 PM	3/30/21, 1:36 PM		Adalet Veyis Turgut;Batuhan Tongarlak;Burakcan Kazm Yeil;Mehmet Gökay Yldz
18		**Requirements**	**28 days?**	**3/31/21, 8:00 AM**	**4/27/21, 5:00 PM**	**8**	
19		Listing preferences of projects	0.083 days?	3/31/21, 8:00 AM	3/31/21, 8:39 AM		Adalet Veyis Turgut;Batuhan Tongarlak;Burakcan Kazm Yeil;Doukan Türksoy;Kürat Talha Berk Yamanolu;Mehmet ...
20		Examining related software systems	0.083 days?	3/31/21, 8:39 AM	3/31/21, 9:19 AM	19	Adalet Veyis Turgut;Batuhan Tongarlak;Burakcan Kazm Yeil;Doukan Türksoy;Kürat Talha Berk Yamanolu;Mehmet ...
21		Creating common and missing features table	2.878 days	4/2/21, 5:00 PM	4/5/21, 4:01 PM	20	Adalet Veyis Turgut[75%];Ömer Ylmaz[75%];hsan Gökcül
22		Requirements analysis	0.732 days	4/5/21, 4:01 PM	4/6/21, 1:52 PM	21	Adalet Veyis Turgut;Batuhan Tongarlak;Burakcan Kazm Yeil;Doukan Türksoy;Kürat Talha Berk Yamanolu;Mehmet ...
23		Creating glossary	1.125 days?	4/6/21, 1:52 PM	4/7/21, 2:52 PM	22	Batuhan Tongarlak;Burakcan Kazm Yeil
24		Creating user requirements	0.417 days?	4/6/21, 1:52 PM	4/7/21, 8:12 AM	22	Doukan Türksoy;Refika Kalyoncu;hsan Gökcül
25		Creating system requirements	0.417 days?	4/6/21, 1:52 PM	4/7/21, 8:12 AM	22	Batuhan Tongarlak;Mehmet Gökay Yldz;Ömer Ylmaz
26		Creating non-functional requirements	3.25 days?	4/6/21, 1:52 PM	4/9/21, 3:52 PM	22	Adalet Veyis Turgut
27		Creating/altering customer questions	0.429 days?	4/6/21, 1:52 PM	4/7/21, 8:18 AM	22	Burakcan Kazm Yeil;Doukan Türksoy;Kürat Talha Berk Yamanolu;Mehmet Gökay Yldz;Refika Kalyoncu;Ömer Ylmaz...
28		Customer meeting #1	0.292 days?	4/15/21, 10:00 AM	4/15/21, 1:20 PM	27	Adalet Veyis Turgut;Kürat Talha Berk Yamanolu;Mehmet Gökay Yldz;Refika Kalyoncu;Ömer Ylmaz;hsan Gökcül
29		Customer meeting #2	2.254 days	4/24/21, 11:00 AM	4/26/21, 2:01 PM	27;28	Adalet Veyis Turgut;Batuhan Tongarlak;Burakcan Kazm Yeil;Doukan Türksoy;Kürat Talha Berk Yamanolu;Mehmet ...
30		Documenting customer meeting notes	1 day?	4/26/21, 5:00 PM	4/27/21, 5:00 PM	28;29	Refika Kalyoncu
31		**Revisions Regarding Customer Meetings**	**25.458 d...**	**4/15/21, 1:20 PM**	**5/10/21, 5:00 PM**	**28**	
32		Revising Glossary	25.458 da...	4/15/21, 1:20 PM	5/10/21, 5:00 PM	28	Adalet Veyis Turgut;Batuhan Tongarlak
33		Revising User Requirements	25.458 da...	4/15/21, 1:20 PM	5/10/21, 5:00 PM	28	Adalet Veyis Turgut;Refika Kalyoncu
34		Revising System Requirements	25.458 da...	4/15/21, 1:20 PM	5/10/21, 5:00 PM	28	Adalet Veyis Turgut;Refika Kalyoncu

Cmpe 352 - Group12 - page1

Cmpe 352 - Group12 - page2

		Name	Duration	Start	Finish	Predeces...	Resource Names
35		**Scenarios**	**0.75 days?**	**4/16/21, 8:00 AM**	**4/16/21, 3:00 PM**	**28**	
36		Creating Scenario #1	0.375 days?	4/16/21, 8:00 AM	4/16/21, 11:00 AM	28	Batuhan Tongarlak;Refika Kalyoncu
37		Creating Scenario #2	0.25 days?	4/16/21, 8:00 AM	4/16/21, 10:00 AM	28	Adalet Veyis Turgut;Doukan Türksoy;hsan Gökcül
38		Creating Scenario #3.1	0.75 days?	4/16/21, 8:00 AM	4/16/21, 3:00 PM	28	Ömer Ylmaz
39		Creating Scenario #3.2	0.75 days?	4/16/21, 8:00 AM	4/16/21, 3:00 PM	28	Kürat Talha Berk Yamanolu
40		Creating Scenario #4	0.75 days?	4/16/21, 8:00 AM	4/16/21, 3:00 PM	28	Mehmet Gökay Yldz
41		**Web Mockups**	**3.4 days?**	**4/16/21, 3:00 PM**	**4/20/21, 9:12 AM**	**35**	
42		General outlines of Mockup	2.4 days	4/16/21, 3:00 PM	4/19/21, 9:12 AM	35	Adalet Veyis Turgut;Batuhan Tongarlak;Burakcan Kazm Yeil;Doukan Türksoy;Kürat Talha Berk Yamanolu;Mehmet ...
43		Creating Web Mockup for Scenario #1	1 day	4/19/21, 9:12 AM	4/20/21, 9:12 AM	42	Batuhan Tongarlak;Refika Kalyoncu
44		Creating Web Mockup for Scenario #2	1 day	4/19/21, 9:12 AM	4/20/21, 9:12 AM	42	Adalet Veyis Turgut;hsan Gökcül
45		Creating Web Mockup for Scenario #3.1	1 day?	4/19/21, 9:12 AM	4/20/21, 9:12 AM	42	Ömer Ylmaz
46		Creating Web Mockup for Scenario #3.2	1 day?	4/19/21, 9:12 AM	4/20/21, 9:12 AM	42	Kürat Talha Berk Yamanolu
47		Creating Web Mockup for Scenario #4	1 day?	4/19/21, 9:12 AM	4/20/21, 9:12 AM	42	Mehmet Gökay Yldz
48		**Android Mockups**	**1 day?**	**4/19/21, 9:12 AM**	**4/20/21, 9:12 AM**	**42;35**	
49		Creating Andriod Mockup for Scenario #1	1 day	4/19/21, 9:12 AM	4/20/21, 9:12 AM	42	Batuhan Tongarlak;Refika Kalyoncu
50		Creating Andriod Mockup for Scenario #2	1 day?	4/19/21, 9:12 AM	4/20/21, 9:12 AM	42	Doukan Türksoy
51		Creating Andriod Mockup for Scenario #3.1	1 day?	4/19/21, 9:12 AM	4/20/21, 9:12 AM	42	Ömer Ylmaz
52		Creating Andriod Mockup for Scenario #3.2	1 day?	4/19/21, 9:12 AM	4/20/21, 9:12 AM	42	Kürat Talha Berk Yamanolu
53		Creating Andriod Mockup for Scenario #4	1 day?	4/19/21, 9:12 AM	4/20/21, 9:12 AM	42	Mehmet Gökay Yldz
54		**Diagrams**	**9.637 da...**	**4/20/21, 5:00 PM**	**4/30/21, 2:05 PM**	**41;48**	
55		Preparing Use Case Diagram	0.571 days?	4/20/21, 5:00 PM	4/21/21, 1:34 PM		Adalet Veyis Turgut;Batuhan Tongarlak;Kürat Talha Berk Yamanolu;Mehmet Gökay Yldz;Refika Kalyoncu;Ömer Ylma...
56		Preparing Class Diagram	4.321 days?	4/23/21, 5:00 PM	4/28/21, 10:34 AM		Adalet Veyis Turgut;Burakcan Kazm Yeil;Doukan Türksoy;Mehmet Gökay Yldz;Refika Kalyoncu
57		Preparing Sequence Diagrams	2.316 days?	4/28/21, 10:34 AM	4/30/21, 2:05 PM	55;56	Adalet Veyis Turgut;Batuhan Tongarlak;Burakcan Kazm Yeil;Doukan Türksoy;Kürat Talha Berk Yamanolu;Mehmet ...
58		**Logo and Name**	**7.75 days?**	**5/5/21, 10:00 AM**	**5/12/21, 5:00 PM**		
59		Finding a suitable name for application	0.75 days?	5/5/21, 10:00 AM	5/5/21, 5:00 PM	18	Mehmet Gökay Yldz
60		Creating logo	7 days	5/6/21, 8:00 AM	5/12/21, 5:00 PM	59	Refika Kalyoncu;hsan Gökcül
61		**Project Plan**	**9 days?**	**5/1/21, 10:00 AM**	**5/10/21, 10:00 AM**	**54**	
62		Planning Until	2.75 days?	5/1/21, 10:00 AM	5/3/21, 5:00 PM	54	Refika Kalyoncu;Adalet Veyis Turgut
63		Planning After	3.125 days?	5/1/21, 10:00 AM	5/4/21, 11:00 AM	54	Mehmet Gökay Yldz;Ömer Ylmaz
64		Writing Executive Summary	2 days	5/3/21, 10:00 AM	5/5/21, 10:00 AM		Kürat Talha Berk Yamanolu
65		Listing Deliverables	2.5 days	5/3/21, 10:00 AM	5/5/21, 3:00 PM		Refika Kalyoncu;Doukan Türksoy
66		Evaluating Deliverables	2 days	5/3/21, 10:00 AM	5/5/21, 10:00 AM		Burakcan Kazm Yeil
67		Evaluating Tools	2.5 days	5/3/21, 10:00 AM	5/5/21, 3:00 PM		Refika Kalyoncu;Adalet Veyis Turgut
68		Creating Table of Work Done by Each Member	2 days	5/3/21, 10:00 AM	5/5/21, 10:00 AM		Batuhan Tongarlak

Cmpe 352 - Group12 - page8

Cmpe 352 - Group12 - page9

		Name	Duration	Start	Finish	Predeces...	Resource Names
69		Creating RAM	2.25 days	5/1/21, 10:00 AM	5/3/21, 1:00 PM		Refika Kalyoncu;Adalet Veyis Turgut;Mehmet Gökay Yldz;Ömer Ylmaz
70		Creating Project Plan Table	1 day	5/3/21, 10:00 AM	5/4/21, 10:00 AM		Refika Kalyoncu;Adalet Veyis Turgut;Kürat Talha Berk Yamanolu;Burakcan Kazm Yeil;hsan Gökcül
71		Merging all	3 days	5/7/21, 10:00 AM	5/10/21, 10:00 AM	65;66;69;...	Batuhan Tongarlak;hsan Gökcül
72		Milestone 1	0 days	5/10/21, 10:00 AM	5/10/21, 10:00 AM	61	Adalet Veyis Turgut;Batuhan Tongarlak;Burakcan Kazm Yeil;Doukan Türksoy;Kürat Talha Berk Yamanolu;Mehmet ...
73		**Review & Revisions**	**7 days**	**5/18/21, 8:00 AM**	**5/24/21, 5:00 PM**		
74		Review Mockup	7 days	5/18/21, 8:00 AM	5/24/21, 5:00 PM	72	Refika Kalyoncu;hsan Gökcül;Ömer Ylmaz
75		Review Requirements	7 days	5/18/21, 8:00 AM	5/24/21, 5:00 PM	72	Kürat Talha Berk Yamanolu;Burakcan Kazm Yeil;Batuhan Tongarlak
76		Review Class Diagram	7 days	5/18/21, 8:00 AM	5/24/21, 5:00 PM	72	Adalet Veyis Turgut;Doukan Türksoy;Mehmet Gökay Yldz
77		Review Sequence Diagram	7 days	5/18/21, 8:00 AM	5/24/21, 5:00 PM	72	Refika Kalyoncu;Adalet Veyis Turgut;Kürat Talha Berk Yamanolu;Burakcan Kazm Yeil;Batuhan Tongarlak;Doukan T...
78		**Essentials of the Project**	**7 days**	**5/25/21, 5:00 PM**	**6/1/21, 5:00 PM**	**73**	
79		Research on APIs	7 days	5/25/21, 5:00 PM	6/1/21, 5:00 PM		Refika Kalyoncu;Adalet Veyis Turgut;Kürat Talha Berk Yamanolu;Burakcan Kazm Yeil;Batuhan Tongarlak;Doukan T...
80		Research on Databases	7 days	5/25/21, 5:00 PM	6/1/21, 5:00 PM		Refika Kalyoncu;Adalet Veyis Turgut;Kürat Talha Berk Yamanolu;Burakcan Kazm Yeil;Batuhan Tongarlak;Doukan T...
81		Research on Servers	7 days	5/25/21, 5:00 PM	6/1/21, 5:00 PM		Refika Kalyoncu;Adalet Veyis Turgut;Kürat Talha Berk Yamanolu;Burakcan Kazm Yeil;Batuhan Tongarlak;Doukan T...
82		**Determine a Requirement Tracking System**	**8 days**	**6/2/21, 8:00 AM**	**6/9/21, 5:00 PM**	**78**	
83		Provisional Design of the Project Structure	2 days	6/2/21, 8:00 AM	6/3/21, 5:00 PM		Refika Kalyoncu;Adalet Veyis Turgut;Kürat Talha Berk Yamanolu;Burakcan Kazm Yeil;Batuhan Tongarlak;Doukan T...
84		Provisional Design of Backend	3 days	6/5/21, 8:00 AM	6/7/21, 5:00 PM	86	Refika Kalyoncu;Adalet Veyis Turgut;Kürat Talha Berk Yamanolu;Burakcan Kazm Yeil;Batuhan Tongarlak;Doukan T...
85		Provisional Design of Frontend	2 days	6/8/21, 8:00 AM	6/9/21, 5:00 PM	83;84	Refika Kalyoncu;Adalet Veyis Turgut;Kürat Talha Berk Yamanolu;Burakcan Kazm Yeil;Batuhan Tongarlak;Doukan T...
86		Provisional Design of Database Structure	1 day	6/4/21, 8:00 AM	6/4/21, 5:00 PM	83	Refika Kalyoncu;Adalet Veyis Turgut;Kürat Talha Berk Yamanolu;Burakcan Kazm Yeil;Batuhan Tongarlak;Doukan T...
87		**API & Prototyping**	**4.667 days**	**6/10/21, 8:00 AM**	**6/14/21, 2:19 PM**	**82**	
88		API Usage	2 days	6/10/21, 8:00 AM	6/11/21, 5:00 PM		Refika Kalyoncu;Adalet Veyis Turgut;Kürat Talha Berk Yamanolu;Burakcan Kazm Yeil;Batuhan Tongarlak;Doukan T...
89		Design of API	2.667 days	6/12/21, 8:00 AM	6/14/21, 2:19 PM	88	Refika Kalyoncu;Adalet Veyis Turgut;Kürat Talha Berk Yamanolu;Burakcan Kazm Yeil;Batuhan Tongarlak;Doukan T...
90		Milestone 2	0 days	6/15/21, 8:00 AM	6/15/21, 8:00 AM	87	Refika Kalyoncu;Adalet Veyis Turgut;Kürat Talha Berk Yamanolu;Burakcan Kazm Yeil;Batuhan Tongarlak;Doukan T...
91		Arrange Teams	0 days	10/12/21, 3:00 PM	10/12/21, 3:00 PM		Adalet Veyis Turgut;Ahmet Berat Can;Batuhan Tongarlak;Burakcan Kazm Yeil;Kürat Talha Berk Yamanolu;Refika Kal...
92		**Group Meetings**	**1 day**	**10/19/21, 8:00 AM**	**10/19/21, 5:00 PM**	**91**	
93		All	0.25 days	10/19/21, 8:00 AM	10/19/21, 9:59 AM		Adalet Veyis Turgut;Ahmet Berat Can;Batuhan Tongarlak;Burakcan Kazm Yeil;Kürat Talha Berk Yamanolu;Refika Kal...
94		Backend	0.75 days	10/19/21, 9:59 AM	10/19/21, 5:00 PM	93	Adalet Veyis Turgut;Burakcan Kazm Yeil
95		Frontend	0.214 days	10/19/21, 9:59 AM	10/19/21, 11:42 AM	93	Ahmet Berat Can;Batuhan Tongarlak;Ömer Ylmaz
96		Android	0.667 days	10/19/21, 9:59 AM	10/19/21, 4:19 PM	93	Kürat Talha Berk Yamanolu;Refika Kalyoncu;hsan Gökcül
97		**Pre Implementation**	**6.752 days**	**10/19/21, 9:59 AM**	**10/25/21, 5:00 PM**	**91;90**	
98		Revising Requirements, Designs, Plans	2 days	10/19/21, 9:59 AM	10/21/21, 9:59 AM	90	Adalet Veyis Turgut;Ahmet Berat Can;Batuhan Tongarlak;Burakcan Kazm Yeil;Kürat Talha Berk Yamanolu;Refika Kal...
99		Gardening Wiki Page	2 days	10/19/21, 9:59 AM	10/21/21, 9:59 AM		
100		**Backend**	**3 days**	**10/23/21, 8:00 AM**	**10/25/21, 5:00 PM**	**84**	
101		Initialize Server	3 days	10/23/21, 8:00 AM	10/25/21, 5:00 PM		Adalet Veyis Turgut;Burakcan Kazm Yeil
102		**Frontend**	**3 days**	**10/23/21, 8:00 AM**	**10/25/21, 5:00 PM**	**85**	

Cmpe 352 - Group12 - page15

Cmpe 352 - Group12 - page16

		Name	Duration	Start	Finish	Predeces...	Resource Names
103		Create Initial Design	3 days	10/23/21, 8:00 AM	10/25/21, 5:00 PM		Ahmet Berat Can;Batuhan Tongarlak;Ömer Yılmaz
104		**Android**	**3 days**	**10/23/21, 8:00 AM**	**10/25/21, 5:00 PM**	**85**	
105		Create Initial Design	3 days	10/23/21, 8:00 AM	10/25/21, 5:00 PM		Kürat Talha Berk Yamanolu;Refika Kalyoncu;hsan Gökcül
106		Progress Review Meeting	0.1 days?	10/26/21, 8:00 AM	10/26/21, 8:48 AM		
107		**Login Register**	**6 days**	**10/26/21, 8:48 AM**	**11/1/21, 8:48 AM**		
108		**Backend**	**6 days**	**10/26/21, 8:48 AM**	**11/1/21, 8:48 AM**		
109		Implement User, User Credentials and Guest Class...	6 days	10/26/21, 8:48 AM	11/1/21, 8:48 AM		Adalet Veyis Turgut;Burakcan Kazım Yeil
110		Implement Register System	4 days	10/26/21, 8:48 AM	11/1/21, 8:48 AM		Adalet Veyis Turgut;Burakcan Kazım Yeil
111		Implement Login System	6 days	10/26/21, 8:48 AM	11/1/21, 8:48 AM		Adalet Veyis Turgut;Burakcan Kazım Yeil
112		Implement Reset Password System	6 days	10/26/21, 8:48 AM	11/1/21, 8:48 AM		Adalet Veyis Turgut;Burakcan Kazım Yeil
113		**Frontend**	**6 days**	**10/26/21, 8:48 AM**	**11/1/21, 8:48 AM**		
114		Register Page	6 days	10/26/21, 8:48 AM	11/1/21, 8:48 AM		Batuhan Tongarlak;Ömer Yılmaz
115		Login Page	6 days	10/26/21, 8:48 AM	11/1/21, 8:48 AM		Ahmet Berat Can;Ömer Yılmaz
116		Reset Password Page	6 days	10/26/21, 8:48 AM	11/1/21, 8:48 AM		Ahmet Berat Can;Batuhan Tongarlak
117		**Android**	**6 days**	**10/26/21, 8:48 AM**	**11/1/21, 8:48 AM**		
118		Register Page	6 days	10/26/21, 8:48 AM	11/1/21, 8:48 AM		Refika Kalyoncu;hsan Gökcül
119		Login Page	6 days	10/26/21, 8:48 AM	11/1/21, 8:48 AM		Kürat Talha Berk Yamanolu;hsan Gökcül
120		Reset Password Page	6 days	10/26/21, 8:48 AM	11/1/21, 8:48 AM		Kürat Talha Berk Yamanolu;Refika Kalyoncu
121		**CI/CD Implementation**	**0.5 days?**	**11/1/21, 8:48 AM**	**11/1/21, 1:48 PM**	**107**	
122		Github Actions	0.5 days?	11/1/21, 8:48 AM	11/1/21, 1:48 PM		Adalet Veyis Turgut;Burakcan Kazım Yeil
123		Configuring AWS	0.5 days?	11/1/21, 8:48 AM	11/1/21, 1:48 PM		Adalet Veyis Turgut;Burakcan Kazım Yeil
124		Progress Review Meeting	0.012 days?	11/2/21, 8:48 AM	11/2/21, 8:53 AM		Adalet Veyis Turgut;Ahmet Berat Can;Batuhan Tongarlak;Burakcan Kazım Yeil;Kürat Talha Berk Yamanolu;Refika Kal...
125		**GoalSpace**	**14.4 days**	**11/2/21, 8:48 AM**	**11/16/21, 1:00 PM**	**107**	
126		**Backend**	**10.9 days**	**11/2/21, 8:48 AM**	**11/12/21, 5:00 PM**	**107**	
127		AllGoal, Goal Classes	3.9 days	11/2/21, 8:48 AM	11/5/21, 5:00 PM		Adalet Veyis Turgut;Burakcan Kazım Yeil
128		Entity, Subgoal, Routine, Task, Questions, Reflection	4 days	11/6/21, 8:00 AM	11/9/21, 5:00 PM	127	Adalet Veyis Turgut;Burakcan Kazım Yeil
129		Resources, Labeling and Classification	3 days	11/10/21, 8:00 AM	11/12/21, 5:00 PM	128	Adalet Veyis Turgut;Burakcan Kazım Yeil
130		**Frontend**	**7.4 days**	**11/2/21, 8:48 AM**	**11/9/21, 1:00 PM**		
131		Goals page	3.9 days	11/2/21, 8:48 AM	11/5/21, 5:00 PM		Ahmet Berat Can;Batuhan Tongarlak;Ömer Yılmaz
132		New Goal Page	3 days	11/6/21, 8:00 AM	11/8/21, 5:00 PM	131	Batuhan Tongarlak;Ömer Yılmaz
133		New Entity Page	3 days	11/6/21, 8:00 AM	11/8/21, 5:00 PM	131	Ahmet Berat Can;Batuhan Tongarlak
134		Subpages in goals page	0.5 days	11/8/21, 5:00 PM	11/9/21, 1:00 PM	131	Ahmet Berat Can;Ömer Yılmaz
135		**Android**	**8 days**	**11/4/21, 8:00 AM**	**11/11/21, 5:00 PM**		
136		Goals page	5 days	11/4/21, 8:00 AM	11/8/21, 5:00 PM		Refika Kalyoncu;hsan Gökcül

		Name	Duration	Start	Finish	Predeces...	Resource Names
137		New Goal Page	3 days	11/9/21, 8:00 AM	11/11/21, 5:00 PM	136	Kürat Talha Berk Yamanolu;hsan Gökcül
138		New Entity Page	3 days	11/9/21, 8:00 AM	11/11/21, 5:00 PM	136	Kürat Talha Berk Yamanolu;Refika Kalyoncu
139		Subpages in goals page	3 days	11/9/21, 8:00 AM	11/11/21, 5:00 PM	136	Kürat Talha Berk Yamanolu;Refika Kalyoncu
140		**Testing**	**3.5 days**	**11/13/21, 8:00 AM**	**11/16/21, 1:00 PM**	**126;130;...**	
141		Test Creation of Goal and other Entity pages	3.5 days	11/13/21, 8:00 AM	11/16/21, 1:00 PM		Adalet Veyis Turgut;Ahmet Berat Can;Batuhan Tongarlak;Burakcan Kazım Yeil;Kürat Talha Berk Yamanolu;Refika Kal...
142		Test Editing of Goal&Entity Pages	3.5 days	11/13/21, 8:00 AM	11/16/21, 1:00 PM		Adalet Veyis Turgut;Ahmet Berat Can;Batuhan Tongarlak;Burakcan Kazım Yeil;Kürat Talha Berk Yamanolu;Refika Kal...
143		Test Linking of Entities and Goals	3.5 days	11/13/21, 8:00 AM	11/16/21, 1:00 PM		Adalet Veyis Turgut;Ahmet Berat Can;Batuhan Tongarlak;Burakcan Kazım Yeil;Kürat Talha Berk Yamanolu;Refika Kal...
144		Test Labeling	3.5 days	11/13/21, 8:00 AM	11/16/21, 1:00 PM		Adalet Veyis Turgut;Ahmet Berat Can;Batuhan Tongarlak;Burakcan Kazım Yeil;Kürat Talha Berk Yamanolu;Refika Kal...
145		Progress Review Meeting	0.012 days?	11/9/21, 8:00 AM	11/9/21, 8:05 AM		Adalet Veyis Turgut;Ahmet Berat Can;Batuhan Tongarlak;Burakcan Kazım Yeil;Kürat Talha Berk Yamanolu;Refika Kal...
146		Milestone Meeting	0.012 days?	11/16/21, 8:00 AM	11/16/21, 8:05 AM		Adalet Veyis Turgut;Ahmet Berat Can;Batuhan Tongarlak;Burakcan Kazım Yeil;Kürat Talha Berk Yamanolu;Refika Kal...
147		Customer Meeting 1	0 days	11/16/21, 4:15 PM	11/16/21, 4:15 PM	125	Adalet Veyis Turgut;Ahmet Berat Can;Batuhan Tongarlak;Burakcan Kazım Yeil;Kürat Talha Berk Yamanolu;Refika Kal...
148		**Group Goal**	**6 days**	**11/16/21, 4:15 PM**	**11/22/21, 4:15 PM**	**147**	
149		**Backend**	**5.5 days**	**11/16/21, 4:15 PM**	**11/22/21, 11:15 AM**		
150		Group Goal	2 days	11/16/21, 4:15 PM	11/18/21, 4:15 PM		Adalet Veyis Turgut;Burakcan Kazım Yeil
151		Integration of goal and members	3.5 days	11/18/21, 4:15 PM	11/22/21, 11:15 AM	150	Adalet Veyis Turgut;Burakcan Kazım Yeil
152		**Frontend**	**6 days**	**11/16/21, 4:15 PM**	**11/22/21, 4:15 PM**		
153		Group Goal page	2 days	11/16/21, 4:15 PM	11/18/21, 4:15 PM		Batuhan Tongarlak;Ömer Yılmaz
154		New Group Goal Page	4 days	11/18/21, 4:15 PM	11/22/21, 4:15 PM	153	Ahmet Berat Can;Ömer Yılmaz
155		New Entity Page	4 days	11/18/21, 4:15 PM	11/22/21, 4:15 PM	153	Ahmet Berat Can;Ömer Yılmaz
156		Subpages in goals page	4 days	11/18/21, 4:15 PM	11/22/21, 4:15 PM	153	Ahmet Berat Can;Batuhan Tongarlak
157		Members Page	2 days	11/16/21, 4:15 PM	11/18/21, 4:15 PM		Ahmet Berat Can;Batuhan Tongarlak
158		**Android**	**6 days**	**11/16/21, 4:15 PM**	**11/22/21, 4:15 PM**		
159		Group Goal page	2 days	11/16/21, 4:15 PM	11/18/21, 4:15 PM		Kürat Talha Berk Yamanolu;Refika Kalyoncu
160		New Group Goal Page	4 days	11/18/21, 4:15 PM	11/22/21, 4:15 PM	159	Kürat Talha Berk Yamanolu;Refika Kalyoncu
161		New Entity Page	4 days	11/18/21, 4:15 PM	11/22/21, 4:15 PM	159	Kürat Talha Berk Yamanolu;hsan Gökcül
162		Subpages in goals page	4 days	11/18/21, 4:15 PM	11/22/21, 4:15 PM	159	Kürat Talha Berk Yamanolu;hsan Gökcül
163		Members Page	2 days	11/16/21, 4:15 PM	11/18/21, 4:15 PM		Refika Kalyoncu;hsan Gökcül
164		Progress Review Meeting	0.012 days?	11/23/21, 8:00 AM	11/23/21, 8:05 AM		Adalet Veyis Turgut;Ahmet Berat Can;Batuhan Tongarlak;Burakcan Kazım Yeil;Kürat Talha Berk Yamanolu;Refika Kal...
165		**Calendar**	**3.375 days**	**11/23/21, 8:00 AM**	**11/26/21, 11:00 AM**	**148**	
166		**Backend**	**3 days**	**11/23/21, 8:00 AM**	**11/25/21, 5:00 PM**	**148**	
167		Calendar System	3 days	11/23/21, 8:00 AM	11/25/21, 5:00 PM		Adalet Veyis Turgut;Burakcan Kazım Yeil
168		**Frontend**	**3 days**	**11/23/21, 11:00 AM**	**11/26/21, 11:00 AM**		
169		Calendar Page	3 days	11/23/21, 11:00 AM	11/26/21, 11:00 AM		Ahmet Berat Can;Batuhan Tongarlak;Ömer Yılmaz
170		**Android**	**3 days**	**11/23/21, 11:00 AM**	**11/26/21, 11:00 AM**		

		Name	Duration	Start	Finish	Predeces...	Resource Names
171		Calendar Page	3 days	11/23/21, 11:00 AM	11/26/21, 11:00 AM		Kürat Talha Berk Yamanolu;Refika Kalyoncu;hsan Gökcül
172		**Analytics**	**5 days**	**11/26/21, 11:00 AM**	**12/1/21, 11:00 AM**		
173		**Backend**	**5 days**	**11/26/21, 11:00 AM**	**12/1/21, 11:00 AM**		
174		Analytics System	5 days	11/26/21, 11:00 AM	12/1/21, 11:00 AM		Adalet Veyis Turgut;Burakcan Kazm Yeil
175		**Frontend**	**5 days**	**11/26/21, 11:00 AM**	**12/1/21, 11:00 AM**		
176		Analytics Page	5 days	11/26/21, 11:00 AM	12/1/21, 11:00 AM		Ahmet Berat Can;Batuhan Tongarlak;Ömer Ylmaz
177		**Android**	**5 days**	**11/26/21, 11:00 AM**	**12/1/21, 11:00 AM**		
178		Analytics Page	5 days	11/26/21, 11:00 AM	12/1/21, 11:00 AM		Kürat Talha Berk Yamanolu;Refika Kalyoncu;hsan Gökcül
179		Progress Review Meeting	0.012 days?	11/30/21, 8:00 AM	11/30/21, 8:05 AM		Adalet Veyis Turgut;Ahmet Berat Can;Batuhan Tongarlak;Burakcan Kazm Yeil;Kürat Talha Berk Yamanolu;Refika Kal...
180		**User Profile**	**14.094 d...**	**11/22/21, 4:15 PM**	**12/6/21, 5:00 PM**	**148**	
181		Studying W3C Activity Stream	0.125 days?	11/22/21, 4:15 PM	11/23/21, 8:14 AM		Adalet Veyis Turgut;Ahmet Berat Can;Batuhan Tongarlak;Burakcan Kazm Yeil;Kürat Talha Berk Yamanolu;Refika Kal...
182		**Backend**	**5 days**	**12/2/21, 8:00 AM**	**12/6/21, 5:00 PM**	**125;148;...**	
183		Follow System	5 days	12/2/21, 8:00 AM	12/6/21, 5:00 PM		Adalet Veyis Turgut;Burakcan Kazm Yeil
184		**Frontend**	**5 days**	**12/2/21, 8:00 AM**	**12/6/21, 5:00 PM**	**181**	
185		Profile page	5 days	12/2/21, 8:00 AM	12/6/21, 5:00 PM		Ahmet Berat Can;Batuhan Tongarlak;Ömer Ylmaz
186		**Android**	**5 days**	**12/2/21, 8:00 AM**	**12/6/21, 5:00 PM**	**181**	
187		Profile Page	5 days	12/2/21, 8:00 AM	12/6/21, 5:00 PM		Kürat Talha Berk Yamanolu;Refika Kalyoncu;hsan Gökcül
188		Progress Review Meeting	0.012 days?	12/7/21, 8:00 AM	12/7/21, 8:05 AM		Adalet Veyis Turgut;Ahmet Berat Can;Batuhan Tongarlak;Burakcan Kazm Yeil;Kürat Talha Berk Yamanolu;Refika Kal...
189		**Search Engine**	**7 days**	**12/7/21, 8:00 AM**	**12/13/21, 5:00 PM**	**180**	
190		**Backend**	**7 days**	**12/7/21, 8:00 AM**	**12/13/21, 5:00 PM**	**180**	
191		User, Goal semantic search	3 days	12/7/21, 8:00 AM	12/9/21, 5:00 PM	180	Adalet Veyis Turgut;Burakcan Kazm Yeil
192		Sorting system	4 days	12/10/21, 8:00 AM	12/13/21, 5:00 PM	191	Adalet Veyis Turgut;Burakcan Kazm Yeil
193		Filter System	4 days	12/10/21, 8:00 AM	12/13/21, 5:00 PM	191	Adalet Veyis Turgut;Burakcan Kazm Yeil
194		**Frontend**	**7 days**	**12/7/21, 8:00 AM**	**12/13/21, 5:00 PM**		
195		Marketplace page	3 days	12/7/21, 8:00 AM	12/9/21, 5:00 PM		Ahmet Berat Can;Batuhan Tongarlak;Ömer Ylmaz
196		Search Results Page	4 days	12/10/21, 8:00 AM	12/13/21, 5:00 PM	195	Ahmet Berat Can;Batuhan Tongarlak;Ömer Ylmaz
197		**Android**	**7 days**	**12/7/21, 8:00 AM**	**12/13/21, 5:00 PM**		
198		Marketplace page	4 days	12/7/21, 8:00 AM	12/10/21, 5:00 PM		Kürat Talha Berk Yamanolu;Refika Kalyoncu;hsan Gökcül
199		Search Results Page	3 days	12/11/21, 8:00 AM	12/13/21, 5:00 PM	198	Kürat Talha Berk Yamanolu;Refika Kalyoncu;hsan Gökcül
200		Milestone Meeting	0 days	12/13/21, 8:00 AM	12/13/21, 8:00 AM		Adalet Veyis Turgut;Ahmet Berat Can;Batuhan Tongarlak;Burakcan Kazm Yeil;Kürat Talha Berk Yamanolu;Refika Kal...
201		Customer Meeting 2	0 days	12/14/21, 3:40 PM	12/14/21, 3:40 PM	180	Adalet Veyis Turgut;Ahmet Berat Can;Batuhan Tongarlak;Burakcan Kazm Yeil;Kürat Talha Berk Yamanolu;Refika Kal...
202		**Feed Page**	**14.833 d...**	**12/7/21, 8:00 AM**	**12/21/21, 3:40 PM**	**180**	
203		Studying W3C Activity Stream	0.125 days?	12/7/21, 8:00 AM	12/7/21, 8:59 AM		Adalet Veyis Turgut;Ahmet Berat Can;Batuhan Tongarlak;Burakcan Kazm Yeil;Kürat Talha Berk Yamanolu;Refika Kal...
204		**Backend**	**7 days**	**12/14/21, 3:40 PM**	**12/21/21, 3:40 PM**	**180;203**	

		Name	Duration	Start	Finish	Predeces...	Resource Names
205		Activity Stream System	7 days	12/14/21, 3:40 PM	12/21/21, 3:40 PM		Adalet Veyis Turgut;Burakcan Kazm Yeil
206		**Frontend**	**7 days**	**12/14/21, 3:40 PM**	**12/21/21, 3:40 PM**	**180;203**	
207		Feed Page	7 days	12/14/21, 3:40 PM	12/21/21, 3:40 PM		Ahmet Berat Can;Batuhan Tongarlak;Ömer Ylmaz
208		**Android**	**7 days**	**12/14/21, 3:40 PM**	**12/21/21, 3:40 PM**	**180;203**	
209		Feed Page	7 days	12/14/21, 3:40 PM	12/21/21, 3:40 PM		Kürat Talha Berk Yamanolu;Refika Kalyoncu;hsan Gökcül
210		Progress Review Meeting	0.012 days?	12/21/21, 3:40 PM	12/21/21, 3:45 PM		Adalet Veyis Turgut;Ahmet Berat Can;Batuhan Tongarlak;Burakcan Kazm Yeil;Kürat Talha Berk Yamanolu;Refika Kal...
211		**Notification**	**7 days**	**12/21/21, 3:40 PM**	**12/28/21, 3:40 PM**	**202**	
212		**Backend**	**7 days**	**12/21/21, 3:40 PM**	**12/28/21, 3:40 PM**	**202**	
213		Notification System	7 days	12/21/21, 3:40 PM	12/28/21, 3:40 PM		Adalet Veyis Turgut;Burakcan Kazm Yeil
214		**Frontend**	**7 days**	**12/21/21, 3:40 PM**	**12/28/21, 3:40 PM**		
215		Notification Page	7 days	12/21/21, 3:40 PM	12/28/21, 3:40 PM		Ahmet Berat Can;Batuhan Tongarlak;Ömer Ylmaz
216		**Android**	**7 days**	**12/21/21, 3:40 PM**	**12/28/21, 3:40 PM**		
217		Notification Page	7 days	12/21/21, 3:40 PM	12/28/21, 3:40 PM		Kürat Talha Berk Yamanolu;Refika Kalyoncu;hsan Gökcül
218		Progress Review Meeting	0.167 days?	12/28/21, 3:40 PM	12/28/21, 5:00 PM		Adalet Veyis Turgut;Ahmet Berat Can;Batuhan Tongarlak;Burakcan Kazm Yeil;Kürat Talha Berk Yamanolu;Refika Kal...
219		**HomePage**	**6 days?**	**12/28/21, 5:00 PM**	**1/3/22, 5:00 PM**	**202**	
220		Backend	4 days?	12/28/21, 5:00 PM	1/1/22, 5:00 PM	202	
221		**Frontend**	**6 days**	**12/28/21, 5:00 PM**	**1/3/22, 5:00 PM**		
222		Homepage	6 days	12/28/21, 5:00 PM	1/3/22, 5:00 PM		Ahmet Berat Can;Batuhan Tongarlak;Ömer Ylmaz
223		**Android**	**6 days**	**12/28/21, 5:00 PM**	**1/3/22, 5:00 PM**		
224		Homepage	6 days	12/28/21, 5:00 PM	1/3/22, 5:00 PM		Kürat Talha Berk Yamanolu;Refika Kalyoncu;hsan Gökcül
225		Milestone Meeting	0.167 days?	1/3/22, 2:03 PM	1/3/22, 3:23 PM		Adalet Veyis Turgut;Ahmet Berat Can;Batuhan Tongarlak;Burakcan Kazm Yeil;Kürat Talha Berk Yamanolu;Refika Kal...
226		Customer Meeting 3 / Final Presentation	0 days	1/4/22, 2:03 PM	1/4/22, 2:03 PM	219	Adalet Veyis Turgut;Ahmet Berat Can;Batuhan Tongarlak;Burakcan Kazm Yeil;Kürat Talha Berk Yamanolu;Refika Kal...

COMMUNICATION PLAN

Who will participate	Where	Purpose	When
All Team Members are expected	Cmpe451 Ps Zoom Session	Discussion about tasks and distributing them.	Tuesday 15:00
Necessary Team Members	Slack	For urgent issues.	When necessary
All Team Members are expected	Slack	Weekly progress review.	Sundays
Backend Team	whatsapp,Slack	Discussing backend development.	When necessary
Android Team	Discord	Discussing android development.	Every Thursday 15.00
Frontend Team	Discord	Discussing frontend development.	Every Thursday 15.00

API DOCUMENTATION

MODELS

```
AuthenticationResponse ⌄ {
    jwt                     string
    message                 string
    messageType             string
                            Enum:
                             ⌄ [ SUCCESS, INFO, ERROR ]
    userCredentialsDTO      UserCredentialsDTO ⌄ {
                                email               string
                                password            string
                                user_id             integer($int64)
                                username            string
                            }
}
```

```
EntitiDTO ⌄ {
    createdAt               string($date-time)
    deadline                string($date-time)
    description             string
    entitiType              string
                            Enum:
                             ⌄ [ SUBGOAL, TASK, ROUTINE, QUESTION, REFLECTION ]
    id                      integer($int64)
    isDone                  boolean
    mainGoal_id             integer($int64)
    period                  integer($int64)
    rating                  number($double)
    title                   string
}
```

```
GoalDTO ⌄ {
    createdAt               string($date-time)
    deadline                string($date-time)
    description             string
    goalType                string
                            Enum:
                             ⌄ [ GROUPGOAL, GOAL ]
    id                      integer($int64)
    isDone                  boolean
    title                   string
}
```

```
MessageResponse ⌄ {
    message                 string
    messageType             string
                            Enum:
                             ⌄ [ SUCCESS, INFO, ERROR ]
}
```

```
AuthenticationResponse ⌄ {
    jwt                     string
    message                 string
    messageType             string
                            Enum:
                             ⌄ [ SUCCESS, INFO, ERROR ]
    userCredentialsDTO      UserCredentialsDTO ⌄ {
                                email                   string
                                password                string
                                user_id                 integer($int64)
                                username                string
                            }
}
```

```
EntitiDTO ⌄ {
    createdAt               string($date-time)
    deadline                string($date-time)
    description             string
    entitiType              string
                            Enum:
                             ⌄ [ SUBGOAL, TASK, ROUTINE, QUESTION, REFLECTION ]
    id                      integer($int64)
    isDone                  boolean
    mainGoal_id             integer($int64)
    period                  integer($int64)
    rating                  number($double)
    title                   string
}
```

```
GoalDTO ⌄ {
    createdAt               string($date-time)
    deadline                string($date-time)
    description             string
    goalType                string
                            Enum:
                             ⌄ [ GROUPGOAL, GOAL ]
    id                      integer($int64)
    isDone                  boolean
    title                   string
}
```

```
MessageResponse ⌄ {
    message                 string
    messageType             string
                            Enum:
                             ⌄ [ SUCCESS, INFO, ERROR ]
}
```

```
TaskDTO ⌄ {
    createdAt               string($date-time)
    description             string
    entitiType              string
                            Enum:
                             ⌄ [ SUBGOAL, TASK, ROUTINE, QUESTION, REFLECTION ]
    id                      integer($int64)
    isDone                  boolean
    mainGoal_id             integer($int64)
    title                   string
}
```

```
UserCredentialsDTO ⌄ {
    email                   string
    password                string
    user_id                 integer($int64)
    username                string
}
```

ENDPOINTS

entiti-controller Entiti Controller

GET	/entities/{entity_id}/sublinks	getSublinksOfAnEntity
DELETE	/entities/{id}/delete_link/{child_id}	deleteSublinkOfAnEntity
POST	/entities/{id}/link/{child_id}	linkEntities
GET	/entities/entiti/{id}	getEntiti
GET	/entities/goal/{goal_id}	Get All the Entities Of a Goal
POST	/entities/question	createQuestion
GET	/entities/question/{id}	getQuestion
PUT	/entities/question/{id}	updateQuestion
DELETE	/entities/question/{id}	deleteQuestion
POST	/entities/reflection	createReflection
GET	/entities/reflection/{id}	getReflection
PUT	/entities/reflection/{id}	updateReflection
DELETE	/entities/reflection/{id}	deleteReflection
POST	/entities/routine	createRoutine
GET	/entities/routine/{id}	getRoutine
PUT	/entities/routine/{id}	updateRoutine
DELETE	/entities/routine/{id}	deleteRoutine
POST	/entities/subgoal	createSubgoal

GET	/entities/subgoal/{id}	getSubgoal
PUT	/entities/subgoal/{id}	updateSubgoal
DELETE	/entities/subgoal/{id}	deleteSubgoal
POST	/entities/task	createTask
GET	/entities/task/{id}	getTask
PUT	/entities/task/{id}	updateTask
DELETE	/entities/task/{id}	deleteTask
GET	/entities/user/{user_id}	getEntitiesOfAUser
GET	/entities/user/question/{user_id}	getQuestionsOfAUser
GET	/entities/user/reflection/{user_id}	getReflectionsOfAUser
GET	/entities/user/routine/{user_id}	getRoutinesOfAUser
GET	/entities/user/subgoal/{user_id}	getSubgoalsOfAUser
GET	/entities/user/task/{user_id}	getTasksOfAUser

goal-controller Goal Controller

PUT	/goals	Update a goal.
GET	/goals/{goal_id}	Get a Goal
DELETE	/goals/{goal_id}	Delete a goal.
POST	/goals/{user_id}	Create a goal.
GET	/goals/of_user/{user_id}	Get all goals of a user.

login-controller Login Controller

POST	/login	Logs in a user
POST	/signup	Signs up a user

user-credentials-controller User Credentials Controller

GET	/users/{username}	Get the info of a user.
GET	/users/forgot/{email_or_username}	Sends a password reset token to user's email address.
POST	/users/reset/{token}	Resets a user's password

DETAILS OF SOME ENDPOINTS

GET /entities/{entity_id}/sublinks getSublinksOfAnEntity

Parameters Try it out

Name	Description
entity_id integer($int64) (path)	Id of the entity.

Responses Response content type */*

Code	Description
200	OK
401	Unauthorized
403	Forbidden
404	Not Found

Example Value | Model

```
[EntitiDTO {
  createdAt          string($date-time)
  deadline           string($date-time)
  description        string
  entitiType         string
                     Enum:
                      [ SUBGOAL, TASK, ROUTINE, QUESTION, REFLECTION ]
  id                 integer($int64)
  isDone             boolean
  mainGoal_id        integer($int64)
  period             integer($int64)
  rating             number($double)
  title              string
}]
```

DELETE `/entities/{id}/delete_link/{child_id}` deleteSublinkOfAnEntity

Parameters Try it out

Name	Description
child_id integer($int64) *(path)*	Id of the child entity.
id integer($int64) *(path)*	Id of the parent entity.

Responses Response content type `*/*`

Code	Description
200	*OK* Example Value | **Model** MessageResponse { message string, messageType string Enum: [SUCCESS, INFO, ERROR] }
204	*No Content*
401	*Unauthorized*
403	*Forbidden*

POST `/entities/{id}/link/{child_id}` linkEntities

Parameters Try it out

Name	Description
child_id integer($int64) *(path)*	Id of the child entity.
id integer($int64) *(path)*	Id of the parent entity.

Responses Response content type `*/*`

Code	Description
200	*OK* Example Value | **Model** MessageResponse { message string, messageType string Enum: [SUCCESS, INFO, ERROR] }
201	*Created*
401	*Unauthorized*
403	*Forbidden*
404	*Not Found*

GET `/entities/entiti/{id}` getEntiti

Parameters Try it out

Name	Description
id integer($int64) *(path)*	Id of the entity.

Responses Response content type `*/*`

Code	Description
200	*OK*

Example Value | **Model**

```
EntitiDTO {
    createdAt          string($date-time)
    deadline           string($date-time)
    description        string
    entitiType         string
                       Enum:
                        [ SUBGOAL, TASK, ROUTINE, QUESTION, REFLECTION ]
    id                 integer($int64)
    isDone             boolean
    mainGoal_id        integer($int64)
    period             integer($int64)
    rating             number($double)
    title              string
}
```

Code	Description
401	*Unauthorized*
403	*Forbidden*
404	*Not Found*

GET `/entities/goal/{goal_id}` Get All the Entities Of a Goal

Parameters Try it out

Name	Description
goal_id integer($int64) *(path)*	Id of the goal.

Responses Response content type `*/*`

Code	Description
200	*OK*

Example Value | **Model**

```
[EntitiDTO {
    createdAt          string($date-time)
    deadline           string($date-time)
    description        string
    entitiType         string
                       Enum:
                        [ SUBGOAL, TASK, ROUTINE, QUESTION, REFLECTION ]
    id                 integer($int64)
    isDone             boolean
    mainGoal_id        integer($int64)
    period             integer($int64)
    rating             number($double)
    title              string
}]
```

Code	Description
401	*Unauthorized*
403	*Forbidden*
404	*Not Found*

POST `/entities/question` createQuestion

Parameters Try it out

Name	Description
question_dto *(body)*	A JSON value representing a transaction. An example of the expected schema can be found down here. Example Value | **Model**

```
QuestionDTO ⌄ {
    createdAt          string($date-time)
    description        string
    entitiType         string
                       Enum:
                       ⌄ [ SUBGOAL, TASK, ROUTINE, QUESTION, REFLECTION ]
    id                 integer($int64)
    isDone             boolean
    mainGoal_id        integer($int64)
    title              string
}
```

Responses Response content type `*/*`

Code	Description
200	*OK* **Example Value** | Model `{ "message": "string", "messageType": "SUCCESS" }`
201	*Created*
401	*Unauthorized*
403	*Forbidden*
404	*Not Found*

GET `/entities/question/{id}` getQuestion

Parameters Try it out

Name	Description
id integer($int64) *(path)*	Id of the question.

Responses Response content type `*/*`

Code	Description
200	*OK* Example Value | **Model**

```
QuestionDTO ⌄ {
    createdAt          string($date-time)
    description        string
    entitiType         string
                       Enum:
                       ⌄ [ SUBGOAL, TASK, ROUTINE, QUESTION, REFLECTION ]
    id                 integer($int64)
    isDone             boolean
    mainGoal_id        integer($int64)
    title              string
}
```

401	*Unauthorized*
403	*Forbidden*
404	*Not Found*

PUT `/entities/question/{id}` updateQuestion

Parameters Try it out

Name	Description
question_dto *(body)*	A JSON value representing a transaction. An example of the expected schema can be found down here. Example Value | **Model**

```
QuestionDTO {
    createdAt          string($date-time)
    description        string
    entitiType         string
                       Enum:
                        [ SUBGOAL, TASK, ROUTINE, QUESTION, REFLECTION ]
    id                 integer($int64)
    isDone             boolean
    mainGoal_id        integer($int64)
    title              string
}
```

Responses Response content type `*/*`

Code	Description
200	*OK* Example Value | **Model**

```
MessageResponse {
    message            string
    messageType        string
                       Enum:
                        [ SUCCESS, INFO, ERROR ]
}
```

201	*Created*
401	*Unauthorized*
403	*Forbidden*
404	*Not Found*

DELETE `/entities/question/{id}` deleteQuestion

Parameters Try it out

Name	Description
id integer($int64) *(path)*	Id of the question.

Responses Response content type `*/*`

Code	Description
200	*OK* Example Value | **Model**

```
MessageResponse {
    message            string
    messageType        string
                       Enum:
                        [ SUCCESS, INFO, ERROR ]
}
```

204	*No Content*
401	*Unauthorized*
403	*Forbidden*

GET `/entities/user/{user_id}` getEntitiesOfAUser

Parameters Try it out

Name	Description
user_id integer($int64) *(path)*	Id of the user.

Responses Response content type */*

Code	Description
200	*OK*

Example Value | **Model**

```
[EntitiDTO {
  createdAt          string($date-time)
  deadline           string($date-time)
  description        string
  entitiType         string
                     Enum:
                      [ SUBGOAL, TASK, ROUTINE, QUESTION, REFLECTION ]
  id                 integer($int64)
  isDone             boolean
  mainGoal_id        integer($int64)
  period             integer($int64)
  rating             number($double)
  title              string
}]
```

Code	Description
401	*Unauthorized*
403	*Forbidden*
404	*Not Found*

GET `/entities/user/question/{user_id}` getQuestionsOfAUser

Parameters Try it out

Name	Description
user_id integer($int64) *(path)*	Id of the user.

Responses Response content type */*

Code	Description
200	*OK*

Example Value | **Model**

```
[QuestionDTO {
  createdAt          string($date-time)
  description        string
  entitiType         string
                     Enum:
                      [ SUBGOAL, TASK, ROUTINE, QUESTION, REFLECTION ]
  id                 integer($int64)
  isDone             boolean
  mainGoal_id        integer($int64)
  title              string
}]
```

Code	Description
401	*Unauthorized*
403	*Forbidden*
404	*Not Found*

PUT `/goals` Update a goal.

Parameters Try it out

Name	Description
goalDTO *(body)*	A JSON value representing a transaction. An example of the expected schema can be found down here.

Example Value | **Model**

```
GoalDTO ⌄ {
    createdAt          string($date-time)
    deadline           string($date-time)
    description        string
    goalType           string
                       Enum:
                        ⌄ [ GROUPGOAL, GOAL ]
    id                 integer($int64)
    isDone             boolean
    title              string
}
```

Responses Response content type `*/*`

Code	Description
200	*OK*
201	*Created*
401	*Unauthorized*
403	*Forbidden*
404	*Not Found*

Example Value | **Model**

```
MessageResponse ⌄ {
    message            string
    messageType        string
                       Enum:
                        ⌄ [ SUCCESS, INFO, ERROR ]
}
```

GET `/goals/{goal_id}` Get a Goal

Parameters Try it out

Name	Description
goal_id `integer($int64)` *(path)*	Id of the goal.

Responses Response content type `*/*`

Code	Description
200	*OK*
401	*Unauthorized*
403	*Forbidden*
404	*Not Found*

Example Value | **Model**

```
GoalDTO ⌄ {
    createdAt          string($date-time)
    deadline           string($date-time)
    description        string
    goalType           string
                       Enum:
                        ⌄ [ GROUPGOAL, GOAL ]
    id                 integer($int64)
    isDone             boolean
    title              string
}
```

DELETE `/goals/{goal_id}` Delete a goal.

Parameters Try it out

Name	Description
goal_id integer($int64) (path)	Id of the goal.

Responses Response content type `*/*`

Code	Description
200	OK Example Value | Model MessageResponse { message string; messageType string Enum: [SUCCESS, INFO, ERROR] }
204	No Content
401	Unauthorized
403	Forbidden

POST `/goals/{user_id}` Create a goal.

Parameters Try it out

Name	Description
goalDTO (body)	A JSON value representing a transaction. An example of the expected schema can be found down here. Example Value | Model GoalDTO { createdAt string($date-time); deadline string($date-time); description string; goalType string Enum: [GROUPGOAL, GOAL]; id integer($int64); isDone boolean; title string }
user_id integer($int64) (path)	Id of the user.

Responses Response content type `*/*`

Code	Description
200	OK Example Value | Model MessageResponse { message string; messageType string Enum: [SUCCESS, INFO, ERROR] }
201	Created
401	Unauthorized
403	Forbidden
404	Not Found

GET `/goals/of_user/{user_id}` Get all goals of a user.

Parameters Try it out

Name	Description
user_id integer($int64) *(path)*	Id of the user.

Responses Response content type `*/*`

Code	Description
200	*OK* Example Value | **Model** [GoalDTO { { createdAt string($date-time) deadline string($date-time) description string goalType string Enum: [GROUPGOAL, GOAL] id integer($int64) isDone boolean title string }]
401	*Unauthorized*
403	*Forbidden*
404	*Not Found*

POST `/login` Logs in a user

One of the email or username can be used, not needed both.

Parameters Try it out

Name	Description

userCredentialsDTO
(body)
	A JSON value representing a transaction. An example of the expected schema can be found down here.
	Example Value | **Model**
	UserCredentialsDTO { {
	email string
	password string
	user_id integer($int64)
	username string
	}

Responses Response content type `*/*`

Code	Description

200	*OK*
	Example Value | **Model**
	AuthenticationResponse { {
	jwt string
	message string
	messageType string
	Enum:
	[SUCCESS, INFO, ERROR]
	userCredentialsDTO UserCredentialsDTO { {
	email string
	password string
	user_id integer($int64)
	username string
	}
	}

201	*Created*
401	*Unauthorized*
403	*Forbidden*
404	*Not Found*

POST `/signup` Signs up a user

Parameters Try it out

Name	Description
userCredentialsDTO *(body)*	A JSON value representing a transaction. An example of the expected schema can be found down here. Example Value | **Model** UserCredentialsDTO { email string, password string, user_id integer($int64), username string }

Responses Response content type */*

Code	Description
200	*OK* Example Value | **Model** MessageResponse { message string, messageType string Enum: [SUCCESS, INFO, ERROR] }
201	*Created*
401	*Unauthorized*
403	*Forbidden*
404	*Not Found*

GET `/users/forgot/{email_or_username}` Sends a password reset token to user's email address.

One of the email or username can be used, not needed both.

Parameters Try it out

Name	Description
email_or_username `string` *(path)*	Email or username of the user.

Responses Response content type */*

Code	Description
200	*OK* Example Value | **Model** MessageResponse { message string, messageType string Enum: [SUCCESS, INFO, ERROR] }
401	*Unauthorized*
403	*Forbidden*
404	*Not Found*

POST `/users/reset/{token}` Resets a user's password

Parameters Try it out

Name	Description
token `string` *(path)*	Password reset token.

user
(body)
	A JSON value representing a transaction. An example of the expected schema can be found down here.
	Example Value | **Model**
	UserCredentialsDTO { {
	email string
	password string
	user_id integer($int64)
	username string
	}

Responses Response content type `*/*`

Code	Description

200	*OK*
	Example Value | Model
	{
	"message": "string",
	"messageType": "SUCCESS"
	}

201	*Created*
401	*Unauthorized*
403	*Forbidden*
404	*Not Found*