CmpE 451

Milestone Report 2

Group 12 - BeABee - tag: G12 Customer Milestone 2 v0.2

18.12.2021

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

Prepared by:

Adalet Veyis Turgut
İhsan Gökcül
Refika Kalyoncu
Kürşat Talha Berk Yamanoğlu
Burakcan Kazım Yeşil
Zahit Batuhan Tongarlak

Executive Summary **4**
Project Description 4
The Track System 4
Social Features 5
Project Status 5
Moving Forward 6
Progress Based On Teamwork **6**
Summary of Work 6
Backend 6
Android 7
Frontend 9
Progress According to Requirements 12
1 Functional Requirements 12
2 Non-Functional Requirements 14
API Documentation 15
ENDPOINTS 15
1.1. User Credentials 15
1.2. Login 15
1.3. Goals 16
1.4. Subgoals 16
1.5. Entities 17
1.6. Groupgoals 19
1.7. Resources 20
1.8. Users 21
MODELS 21
Implementation of Core Functionality 26
1. Describe the status of this functionality 26
2. Describe in detail the work that has been completed regarding this functionality 26
a- Goals 26
b- Group Goals 26
c- Subgoals 27
d- Entities 27
1- Tasks 27
2- Routines 27
3- Questions 27
4- Reflections 27
e- Resources 27

f- Analytics 28
3. Describe the work that remains for this functionality. 28
User Interface/User Experience 28
Web 28
Mobile 39
Standards 64
Wikidata 64
W3C Activity Stream 2.0 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 if that user has permission for their profile or goals to be seen by the public. 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 subgoals are implicitly assigned to the owner, in the Group Goal there can be one or many assignees for subgoals 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

After the first milestone and the customer meeting we have taken some notes about the feedback we received from the customer. So according to this feedback we have altered the Goal/Subgoal/Entity relationship and related UI’s to give better understanding to the user. We believe it is much easier to use now.

About the latest version of the app, we have implemented most of the functionalities of the application. A user can now register, login, create/edit personal and group goals and

their subgoals and entities. They can also join a group goal created by other users if they have the join code. After they make some progress on their goals/subgoals/tasks/routines they can mark them as completed and provide their feedback on their performance during the process. This performance info is collected by the system to provide analytics to the user. In the analytics page users can see their performance data like their average score, their best/worst performed goals, how much did they take to complete goals in average.

Moving Forward

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 is already implemented but we will also implement the follow and activity stream features to allow users to follow each other and see their followed users recent activities like creating or completing tasks/subgoals/goals. Another social feature will be the marketplace where users can share their created goals and copy existing goals shared by other users to their dashboard to be later used as a template and so that they can edit and refine them according to their needs. We also want to add functionality for our users to be able to search existing entities inside their goals. And they will be able to tag entities and goals with special tags and later they will be able to use these tags for related search etc.

Progress Based On Teamwork

Summary of Work

Backend

Burakcan Kazım Yeşil	Review of Backend Application Pull Requests of Veyis
	Deployment of Frontend to the Amazon EC2
	Wrote Dockerfile for Frontend and updated the docker-compose file
	Finished CI/CD pipeline for the backend application.
	Implemented Group Goal function
	Fixed bugs in the backend application.
	Prepared Implementation of Core Functions for Milestone 2.

Adalet Veyis Turgut	Taking meeting note #6 and uploading it to the wiki page.
	Creating base for the scenario of customer demo
	Filling personal efforts weekly.
	Implementing the backend of the application except group goal. – -- includes resource, analytics, extension/completion behaviors of goals/subgoals/entities, and endpoints written upon requests.
	Documenting endpoints so that other teams can understand what endpoints do.
	Reviewing Burakcan's pull requests.
	Writing "API documentation" part of this report.
	Helping teammates to run the application locally until CI/CD is integrated.
	Changing previous codebase according to the feedback from milestone 1.
	Writing unittests.

Android

Kürşat Talha Berk Yamanoğlu	Define Tasks for the next milestone, create issues, distribute them among team members.
	Review previous code according to customer feedback
	Change the subgoal design for better understandibility
	Change the Base Fragments to fit with new design
	Add loading dialog
	Test new backend endpoint related to the revisions of the design and provide feedback to backend team
	Discuss with Veyis about Backend endpoints for new featuress and create issues
	Review Ihsan's work and help him
	Review Refika's work and help her
	Review PR's
	Implement rating and complete features.
	Review the scenario for the presentation created by Ihsan
	Present the scenario.

Refika Kalyoncu	Created Group Goals Home Page
	Connected Group Goals page with bottom Bar
	Created Group Goal Detail Page and Edit Page
	Connected Group Goals with API for creating, editing, viewing
	Connected the group goal with relative entites and subgoal
	Reorganized the bottom bar
	Created Join with Token Page
	Created Members Page
	Implemented Leave Group Goal Feature
	Implemented Delete Group Goal Feature

	Made each API connection that is relative with Group Goal
	Created the Analytics Page
	Connected user's Analytics with Analytics Page
	Revised APIs and give feedback about Internal Server errors to backend team
	Revised Berk's pull Request
	Attended android team meetings
	Attended lab meeting and group meetings
	Helped preparation of the Scenario for the Customer Meeting.

İhsan Gökcül	Attended all group and customer meetings
	Implementing deadline calendar for task and routine entities
	Deadline calendar for createTask, createRoutine pages
	Implementing Routine period feature
	Adding deadline calendar for goal, subgoal and groupgoal
	Adding deadline extension feature for task and routines
	Adding Calendar view for date selection
	Tried to do the things that Berk taught us in Android meetings.
	Attended meetings with Android team
	Merged and organized the milestone report 2.
	Created pull request for my work.
	Created necessary issues for my work.
	Reviewed android team pull requests.
	Helped to create the scenario for customer presentation.
	Added User interface and links for Android to the Milestone Report.
	Documented the scenario and mockup of the customer presentation.

	Controlled the Android phone in customer presentation

Frontend

Ömer Yılmaz	Updated View All Goals Page
	Updated Goal Pages
	Updated SubGoal Pages
	Created Group Goal Pages
	Created rating and completing for all goals
	Created assigning for group and its sub goals.
	Joining group with token
	Tested the scenario on frontend
	Gave feedback for the backend API and asked new endpoints

Zahit Batuhan Tongarlak	Updated Link Entity Page
	Updated Edit Entity Page
	Created Add Resource Page
	Created Date Picker Input to Task and Routines
	Created Linking System from start
	Updated View Entity Page
	Updated Goal Page
	Created Add Question Page
	Created Add Task Page
	Created Add Routine Page
	Created Add Reflection Page
	Created a bug issues for backend team
	Helped Berat to update the visualization of the frontend
	Merged last version of frontend manually

	Added screenshots of new functionalities of frontend to M2
	Attended group meetings except week8

Ahmet Berat Can	Created Calendar Page
	Added ‘redux’ and ‘redux-thunk’ libraries to the App
	Added new logo to the frontend
	Created Used Technologies page on wiki
	Updated Frontend color palette

Progress According to Requirements

Note: C stands for Completed, I stands for In progress, N stands for Not started. First term is the status of backen team, the middle term is the status of the Android team, and the last term is the status of the frontend team about the requirement.
For example C/N/I means that the backend team completed, the Android team did not start, and the Frontend team is still working on it.

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.C/C/C
1.1.1.2 Guests should be able to sign up using their Google accounts.N/N/N
1.1.1.3 Guests shall be able to search for users by their username.N/N/N
1.1.1.4 Guests shall be able to search for public goals using the search functionality mentioned (1.2.2).N/N/N

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

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).C/C/C
1.1.3.2 Users shall be able to rate their efforts towards their tasks and routines after deadline. C/C/I

1.1.3.3 Users shall be able to describe the practices that support these goals under resources section.C/C/C
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.N/N/N
1.1.3.5 Users shall be able to add the related resources to their goals.C/I/C
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.C/C/C
    1.1.3.6.2 Users shall be able to modify descriptions.C/C/C
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.C/C/C
1.1.3.8 Users shall be able to tag goals and items with semantic tags.N/N/N
1.1.3.9 Users shall be able to search goals in marketplace.N/N/N
1.1.3.10 Users shall be able to add goals that came out of the search to themselves.N/N/N
1.1.3.11 Users shall be able to delete goals.C/C/C
1.1.3.12 Users should be able to mute and enable notifications.N/N/N
1.1.3.13 Users shall be able to see the items of the other users they are following in their feed.N/N/N
1.1.3.14 Users shall be able to add entities.
    1.1.3.14.1 They shall be able to add standalone entities.C/C/C
    1.1.3.14.2 They shall be able to add entities under other items.C/C/C
1.1.3.15 Users shall be able to search entities.N/N/N
1.1.3.16 Users should be able to mute or enable notifications.N/N/N
1.1.3.17 Users shall be able to give feedback about their goals / subgoals.C/C/C

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

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.N/N/N
1.1.5.2 Users shall be able to follow other users.N/N/N

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.I/I/N
1.1.6.2 Users shall be able to see their analytics related to their goals and effort expended on these goals.C/C/N

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.C/N/C
1.2.1.2 The system shall not allow guests to be group members.C/C/C

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

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

1.2.4 Analytics
1.2.4.1 The system shall provide analytics about activities those have common lifetimes.C/N/
1.2.4.2 The system shall provide analytics about related activities.N/N/N
1.2.4.3 The system shall provide feedback button after deadline of items. -/ I/N
1.2.5 Profile Management
1.2.5.1 The system shall let users to view others' and their goals. C/I/I

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

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

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. N/N/N
2.1.2 The system shall support semantic tags and these semantic tags should be supported with wikidata.orgN/N/N

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

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.C
    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.N/N/N
2.4.2 The System shall protect the user data under KVKK and GDPR rules.N/N/N

API Documentation

Swagger URL: Beabee API
Postman collections can be found here:
https://github.com/bounswe/2021SpringGroup12/tree/master/deliverables/postman_collections

1. ENDPOINTS

1.1. User Credentials

Functionality	URL	Method	Parameters	Responses
Retrieve user details by username	/v2/users/{username}	GET	username(path)*	UserCredentialsGetDTO

Send a password reset token to user's email address	/v2/users/forgot/{email_or_username}	GET	email(path)* or username(path)*	MessageResponse
Reset a user's password	/v2/users/reset/{token}	POST	token(path)*, UserCredentialsPostDTO*	MessageResponse

Table 1: User credentials endpoints

1.2. Login

Functionality	URL	Method	Parameters	Responses
Logs in the user	/v2/login	POST	username*, email*, password*	AuthenticationResponse
Signs the user up	/v2/signup	POST	UserCredentialsPostDTO	MessageResponse

Table 2: Login endpoints

1.3. Goals

Functionality	URL	Method	Parameters	Responses
Get the info of a goal	/v2/goals/{goal_id}	GET	goal_id(path)*	GoalGetDTO
Update the info of a goal	/v2/goals	PUT	GoalGetDTO*	MessageResponse
Create a goal for the given user	/v2/goals/{user_id}	POST	GoalPostDTO*	MessageResponse
Delete the goal with the given id	/v2/goals/{goal_id}	DELETE	goal_id(path)*	MessageResponse
Complete a goal	/v2/goals/complete/{goal_id}	PUT	goal_id(path)*	MessageResponse
Extend a goal	/v2/goals/extend/{goal_id}	PUT	goal_id(path)*, DateDTO*	MessageResponse

Get the analytics of a goal	/v2/goals/analytics/{goal_id}	GET	goal_id(path)*	GoalAnalyticsDTO
Get all goals the given user	/v2/goals/of_user/{user_id}	GET	user_id(path)*	List<GoalDTOShort>
Create a subgoal under the given goal	/v2/goals/subgoal	POST	goal_id(path)*, SubgoalPostDTO*	MessageResponse

Table 3: Goal endpoints

1.4. Subgoals

Functionality	URL	Method	Parameters	Responses
Get the info of a subgoal	/v2/subgoals/{id}	GET	id(path)*	SubgoalGetDTO
Update the info of a subgoal	/v2/subgoals	PUT	SubgoalPostDTO*	MessageResponse
Create a subgoal under the given subgoal	/v2/subgoals	POST	SubgoalPostDTO*	MessageResponse
Delete the subgoal with the given id	/v2/subgoals/{id}	DELETE	id(path)*	MessageResponse
Get all subgoals of the given user	/v2/subgoals/of_user/{user_id}	GET	user_id(path)*	List<SubgoalGetDTO>
Complete the given subgoal	/v2/subgoals/complete/{subgoal_id}/{rating}	PUT	subgoal_id(path)*, rating*	MessageResponse
Extend the given subgoal	/v2/subgoals/extend/{subgoal_id}	PUT	subgoal_id(path)*, DateDTO*	MessageResponse
Get analytics of the given subgoal	/v2/subgoals/analytics/{subgoal_id}	GET	subgoal_id(path)*	SubgoalAnalyticsDTO
Revoke the assignment of subgoal to the users	/v2/subgoals/{subgoal_id}/assignees	DELETE	subgoal_id(path)*, List<Long>*	MessageResponse
Assign the subgoal to the users	/v2/subgoals/{subgoal_id}/assignees	POST	subgoal_id(path)*, List<Long>*	MessageResponse

Table 4: Subgoal endpoints

1.5. Entities

Functionality	URL	Method	Parameters	Responses
Create a task	/v2/entities/task	POST	TaskPostDTO	MessageResponse
Update the given task	/v2/entities/task/	PUT	TaskGetDTO	MessageResponse
Get the task with the given id	/v2/entities/task/{id}	GET	id(path)*	TaskGetDTO
Delete the task with the given id	/v2/entities/task/{id}	DELETE	id(path)*	MessageResponse
Create a routine	/v2/entities/routine	POST	RoutinePostDTO	MessageResponse
Update the given routine	/v2/entities/routine/	PUT	RoutineGetDTO	MessageResponse
Get the routine with the given id	/v2/entities/routine/{id}	GET	id(path)*	RoutineGetDTO
Delete the routine with the given id	/v2/entities/routine/{id}	DELETE	id(path)*	MessageResponse
Create a question	/v2/entities/question	POST	QuestionPostDTO	MessageResponse
Update the given question	/v2/entities/question/	PUT	QuestionGetDTO	MessageResponse
Get the question with the given id	/v2/entities/question/{id}	GET	id(path)*	QuestionGetDTO
Delete the question with the given id	/v2/entities/question/{id}	DELETE	id(path)*	MessageResponse
Create a reflection	/v2/entities/reflection	POST	ReflectionPostDTO	MessageResponse
Update the given reflection	/v2/entities/reflection/	PUT	ReflectionGetDTO	MessageResponse
Get the reflection with the given id	/v2/entities/reflection/{id}	GET	id(path)*	ReflectionGetDTO
Delete the reflection with the given id	/v2/entities/reflection/{id}	DELETE	id(path)*	MessageResponse

Get all entities of the goal with the given id including entities of subgoals' and entities of subgoals' of subgoals of the given goal	/v2/entities/goal/{goal_id}	GET	goal_id(path)*	List<EntitiDTOShort>
Get all entities of the groupgoal with the given id including entities of subgoals' and entities of subgoals' of subgoals of the given groupgoal	/v2/entities/groupgoal/{groupgoal_id}	GET	groupgoal_id(path)*	List<EntitiDTOShort>
Get all entities of the given user	/v2/entities/user/{user_id}	GET	user_id(path)*	List<EntitiDTOShort>
Complete the question with the given id	/v2/entities/completequestion/{question_id}	PUT	question_id(path)*	MessageResponse
Complete the reflection with the given id	/v2/entities/completereflection/{reflection_id}	PUT	reflection_id(path)*	MessageResponse
Complete the routine with the given id	/v2/entities/completeroutine/{routine_id}/{rating}	PUT	routine_id(path)*, rating(path)*	MessageResponse
Complete the task with the given id	/v2/entities/completetask/{task_id}/{rating}	PUT	task_id(path)*, rating*	MessageResponse
Rate the interval of the routine with the given id	/v2/entities/rate/{routine_id}/{rating}	PUT	routine_id(path)*, rating(path)*	MessageResponse
Extend the entity with the given id	/v2/entities/extend/{entiti_id}	PUT	entiti_id(path)*, DateDTO*	MessageResponse
Link two entities	/v2/entities/{id}/link/{child_id}	POST	id(path)*, child_id(path)*	MessageResponse
Delete the link between given entities	/v2/entities/{id}/delete_link/{child_id}	DELETE	id(path)*, child_id(path)*	MessageResponse

Table 5: Entity endpoints

1.6. Groupgoals

Functionality	URL	Method	Parameters	Responses
Get the info of a groupgoal	/v2/groupgoals/{groupgoal_id}	GET	goal_id(path)*	GroupGoalGetDto
Update the info of a groupgoal	/v2/groupgoals	PUT	GroupGoalGetDto*	MessageResponse
Create a groupgoal for the given user	/v2/groupgoals/{user_id}	POST	groupGoalPostDTO*	MessageResponse
Delete the groupgoal with the given id	/v2/groupgoals/{groupgoal_id}	DELETE	goal_id(path)*	MessageResponse
Create a subgoal under the group goal with the given id	/v2/groupgoals/subgoal	POST	SubgoalPostDTO*	MessageResponse
Get all group goals created by a user	/v2/groupgoals/created_by/{user_id}	GET	user_id(path)*	List<GroupGoalDTOShort>
Get all group goals a user is a member of	/v2/groupgoals/member_of/{user_id}	GET	user_id(path)*	List<GroupGoalDTOShort>
Generate a new token for the group goal	/v2/groupgoals/{groupgoal_id}/token	POST	groupgoal_id(path)*	MessageResponse
Add a new member to the Group Goal	/v2/groupgoals/{groupgoal_id}/members	PUT	groupgoal_id(path)*, username(path)*	MessageResponse
Join a group goal with token	/v2/groupgoals/{user_id}/join	POST	user_id(path)*, token(path)*	GroupGoalDTOShort
Leave a group goal	/v2/groupgoals/{user_id}/{groupgoal_id}	DELETE	user_id(path)*, groupgoal_id(path)*	MessageResponse

Table 6: Group goal endpoints

1.7. Resources

Functionality	URL	Method	Parameters	Responses
Get the resource with the given id	/v2/resources/{id}	GET	id(path)*	ResponseEntity<byte []>
Post a resource under the given entity.	/v2/resources/{entiti_id}	POST	entiti_id(path)*, resource*	ResponseEntity<String>
Delete the resource with the given id	/v2/resources/{id}	DELETE	id(path)*	ResponseEntity<String>
Get all resources of the logged in user	/v2/resources/	GET	None	List<Resource>

Table 7: Resource endpoints

1.8. Users

Functionality	URL	Method	Parameters	Responses
Get the analytics of the user with the given id	/v2/users/analytics/{user_id}	GET	user_id(path)*	UserAnalyticsDTO

Table 8: User analytics endpoints

2. MODELS

Name	Fields
AuthenticationResponse	jwt:string, message:string, messageType: [SUCCESS, INFO, ERROR], userCredentialsGetDTO: UserCredentialsGetDTO
DateDTO	newDeadline: date-time
EntitiDTOShort	id: long, title: string, description: string, deadline: date-time,

	entitiType: EntitiType
EntitiType	[TASK, ROUTINE, QUESTION, REFLECTION]
GoalAnalyticsDTO	activeSubgoalCount: long, averageCompletionTimeOfSubgoals: long, bestSubgoal: SubgoalDTOShort, completedSubgoalCount: long, completionTimeInMiliseconds: long, extensionCount: long, finishTime: date-time, goal_id:long, goalsWithCommonLifetime: [GoalDTOShort], longestSubgoal: SubgoalDTOShort, rating: double, shortestSubgoal: SubgoalDTOShort, startTime: date-time, status: [ACTIVE, COMPLETED], worstSubgoal: SubgoalDTOShort
GoalDTOShort	id: long, title: string, description: string, deadline: date-time
GoalGetDTO	id: long, title: string, description: string, deadline: date-time, user_id: long, createdAt: date-time, completedAt: date-time, entities: [EntitiDTOShort], subgoals: [SubgoalDTOShort], isDone: boolean goalType: [GROUPGOAL, GOAL]
GoalPostDTO	title: string, description: string, deadline: date-time
GroupGoalDTOShort	id: long, title: string, description: string, deadline: date-time
GroupGoalGetDTO	id: long, title: string, description: string, deadline: date-time,

	user_id: long, createdAt: date-time, entities: [EntitiDTOShort], members: [UserCredentialsGetDTO], subgoals: [SubgoalDTOShort], isDone: boolean goalType: [GROUPGOAL, GOAL], token: string
GroupGoalPostDTO	title: string, description: string, deadline: date-time
MessageResponse	message: string, messageType: [SUCCESS, INFO, ERROR]
QuestionGetDTO	id: long, title: string, description: string, goal_id: long, groupgoal_id: long, createdAt: date-time, sublinks: [EntitiDTOShort], subgoal_id: [SubgoalDTOShort], isDone: boolean entitiType: EntitiType, resources: [ResourceDTOShort]
QuestionPostDTO	parent_id: long, title: string, description: string, deadline: date-time, parentType: [NONE, SUBGOAL, GOAL, GROUPGOAL, ENTITY]
ReflectionGetDTO	id: long, title: string, description: string, goal_id: long, groupgoal_id: long, createdAt: date-time, sublinks: [EntitiDTOShort], subgoal_id: [SubgoalDTOShort], isDone: boolean entitiType: EntitiType, resources: [ResourceDTOShort]
ReflectionPostDTO	parent_id: long, title: string, description: string, deadline: date-time,

	parentType: [NONE, SUBGOAL, GOAL, GROUPGOAL, ENTITY]
Resource	id: long, name: string, createdAt: date-time, contentType: string, data: byte, size: long, url: string
ResourceDTOShort	id: long, name: string, createdAt: date-time, contentType: string
RoutineGetDTO	id: long, title: string, description: string, deadline: [date-time], goal_id: long, groupgoal_id: long, createdAt: date-time, sublinks: [EntitiDTOShort], subgoal_id: [SubgoalDTOShort], isDone: boolean entitiType: EntitiType, rating: double, period: long, resources: [ResourceDTOShort]
RoutinePostDTO	parent_id: long, title: string, description: string, deadline: date-time, parentType: [NONE, SUBGOAL, GOAL, GROUPGOAL, ENTITY], period: long
SubgoalAnalyticsDTO	activeSubgoalCount: long, averageCompletionTimeOfSubgoals: long, bestSubgoal: SubgoalDTOShort, completedSubgoalCount: long, completionTimeInMiliseconds: long, extensionCount: long, finishTime: date-time, longestSubgoal: SubgoalDTOShort, rating: double, shortestSubgoal: SubgoalDTOShort, startTime: date-time, status: [ACTIVE, COMPLETED], subgoal_id: long,

	worstSubgoal: SubgoalDTOShort
SubgoalDTOShort	id: long, title: string, description: string, deadline: date-time
SubgoalGetDTO	assignees: [Long], completedAt: date-time, createdAt: date-time, deadline: date-time, description: string, entities: [EntitiDTOShort], id: long, isDone: boolean, main_goal_id: long, main_groupgoal_id: long, parent_subgoal_id: long, rating: double, sublinks: [SubgoalDTOShort], title: string
SubgoalPostDTO	deadline: date-time, description: string, main_goal_id: long, main_groupgoal_id: long, parent_subgoal_id: long, title: string
TaskGetDTO	id: long, title: string, description: string, deadline: date-time, goal_id: long, groupgoal_id: long, createdAt: date-time, sublinks: [EntitiDTOShort], subgoal_id: [SubgoalDTOShort], isDone: boolean entitiType: EntitiType, rating: double, resources: [ResourceDTOShort]
TaskPostDTO	parent_id: long, title: string, description: string, deadline: date-time, parentType: [NONE, SUBGOAL, GOAL, GROUPGOAL, ENTITY],
UserAnalyticsDTO	activeGoalCount: long,

	averageCompletiontimeOfGoalsInMiliseconds: long, averageExtensionCount: long, averageRating: double, bestGoal: GoalDTOShort, completedGoalCount: long longestGoal: GoalDTOShort, shortestGoal: GoalDTOShort, worstGoal: GoalDTOShort
UserCredentialsGetDTO	user_id: long, username: string, password: string, email: string
UserCredentialsPostDTO	username: string, password: string, email: string

Table 9: Models

Implementation of Core Functionality

1. Describe the status of this functionality

The path functionality is named goals in our project. We have implemented the basic functionality of the goals, subgoals that are linked to these goals and entities that can be a part of these goals or subgoals before the previous milestone. Since then we have implemented group goals and analytics. Users can create goals that they want to follow and finish, they can add subgoals that will help them create the path in this journey and add entities that will further modularize the goals and help them improve or think about certain parts of the goal. In addition, users can also create group goals that they can create to complete goals with their friends. Users can also see their statistics so they can fix problems or reflect on the path they chose to improve themselves.

2. Describe in detail the work that has been completed regarding this functionality

a- Goals

Goals are the paths that users can create and complete as a single person, they are at the top of the entity hierarchy in our system. This means they have subgoals and entities that are linked to them.

First we implemented the main goal objects and methods to give the user to create the paths they want to work on. Goals have a parent object that is named “AllGoals” that will be also

used in the Group Goal objects. Goals have all the necessary fields to inform the user about the creator of the goal, its creation date, its completion date, its extension, its rating etc. We also implemented the Data Transfer Objects to transfer the data we get from the database via models to other layers and users of the API. And to make the conversion between models and DTOs easier we also implemented mappers that make these conversions.

b- Group Goals

Group Goals are nearly identical to goals. However, the biggest difference between them is that more than one user can complete and follow the path of these goals. So there is an implementation differentiation between these two. They both stem from the same interface "AllGoals".

To accommodate more than one user, group goals have fields and methods to add members, kick members etc.

c- Subgoals

Subgoals are the one lower level of our system hierarchy. Users can create them under some groups or group goals or other subgoals. They are smaller parts of improvement paths that users define. They also can also contain entities under them just like goals. We also implemented models to get data from the database, DTOs to transfer the info to other layers and mappers to convert between two.

d- Entities

Entities are the smallest part of our goal system. They are at the bottom of our hierarchy, and they represent the simplest part of the users' paths. They can be things that the user wants to achieve, or can be questions they want to answer etc.

They can get created under any goal, group goal or subgoal and they can get linked with other entities that are under the same goal.

All entities share some fields that are used to identify them, indicate their start date, end date, creator, the goal or subgoal that they are part of and other entities they are linked to. In addition, every entity type has some fields that are special to them.

1- Tasks

Tasks are the simplest types of entities. They represent one and done tasks that users want to complete. They have deadlines that can be extended by the user.

2- Routines

Routines are things that users want to do periodically. Users can choose the period that they want to do this task and do this task repeatedly.

3- Questions

Questions represent the questions that user have about one concept or task. They are simple entities with no real logic.

4- Reflections

Reflections are entities that represent the comments or reflections they have about themselves about any tasks, goals, or other concepts.

e- Resources

Resources are outside sources like links, texts, or images that users can add to a related entity. They can help users complete a task, answer a question, or strengthen the ideas behind a reflection.

f- Analytics

Our API also provides some statistics about goals, subgoals or users. So, applications can display some analytic info like average completion time, rating or extension times of a goal, or lifetime of user. So, the user can evaluate their efforts and can improve the way they approach paths that they have created.

3. Describe the work that remains for this functionality.

In the future our users will be available to search for goals that are created by other users, so they can use them as an example or follow a person's path as in a community. This can be used by communities or give birth to communities that use the same path to improve themselves.

Also, to show the paths that they created users will have profiles that showcase the goals or paths they created and share them with other users.

And if a user is really interested in some other user's paths, they can follow them to see their new goals, or their progression about a goal. To achieve this, we will implement a feed that users can see the activity of users that they follow.

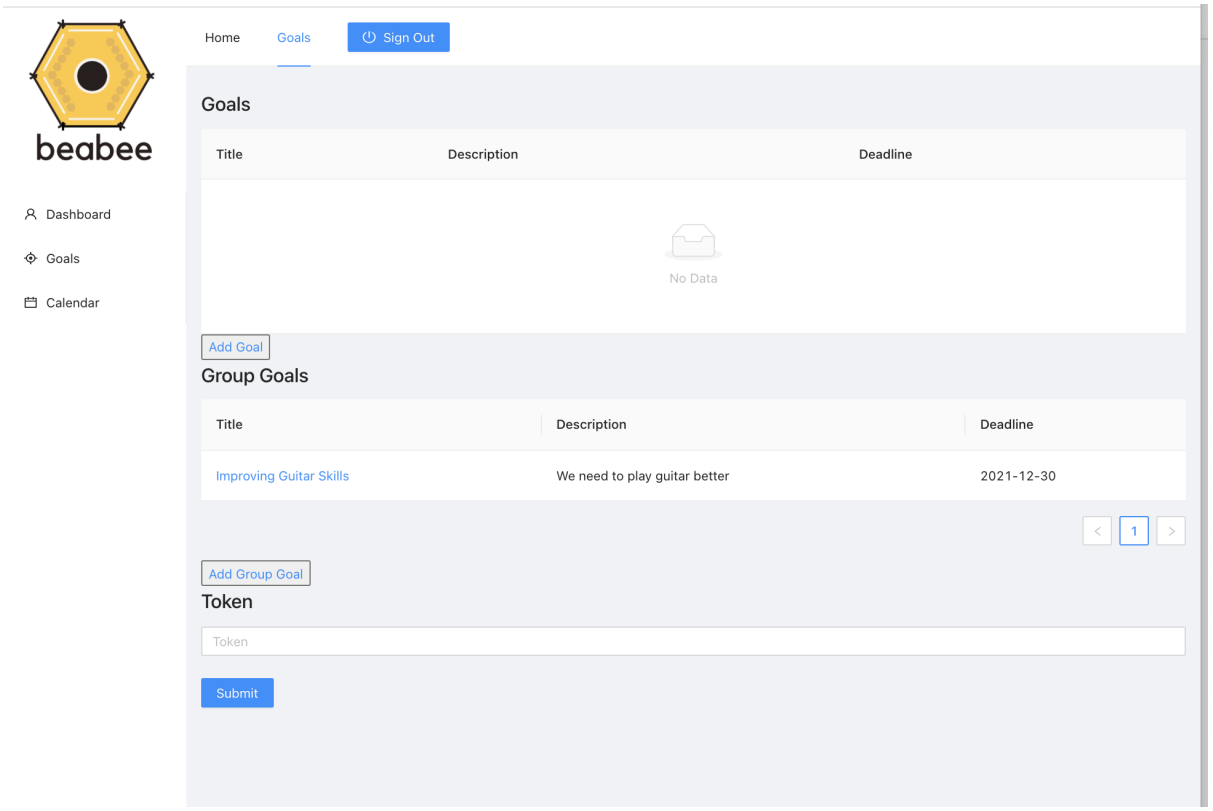
And in the future users will get notifications from the app. These will be about deadlines, user activity, followers etc. Users will be able to choose which notifications they want to receive if any at all.

User Interface/User Experience

Web

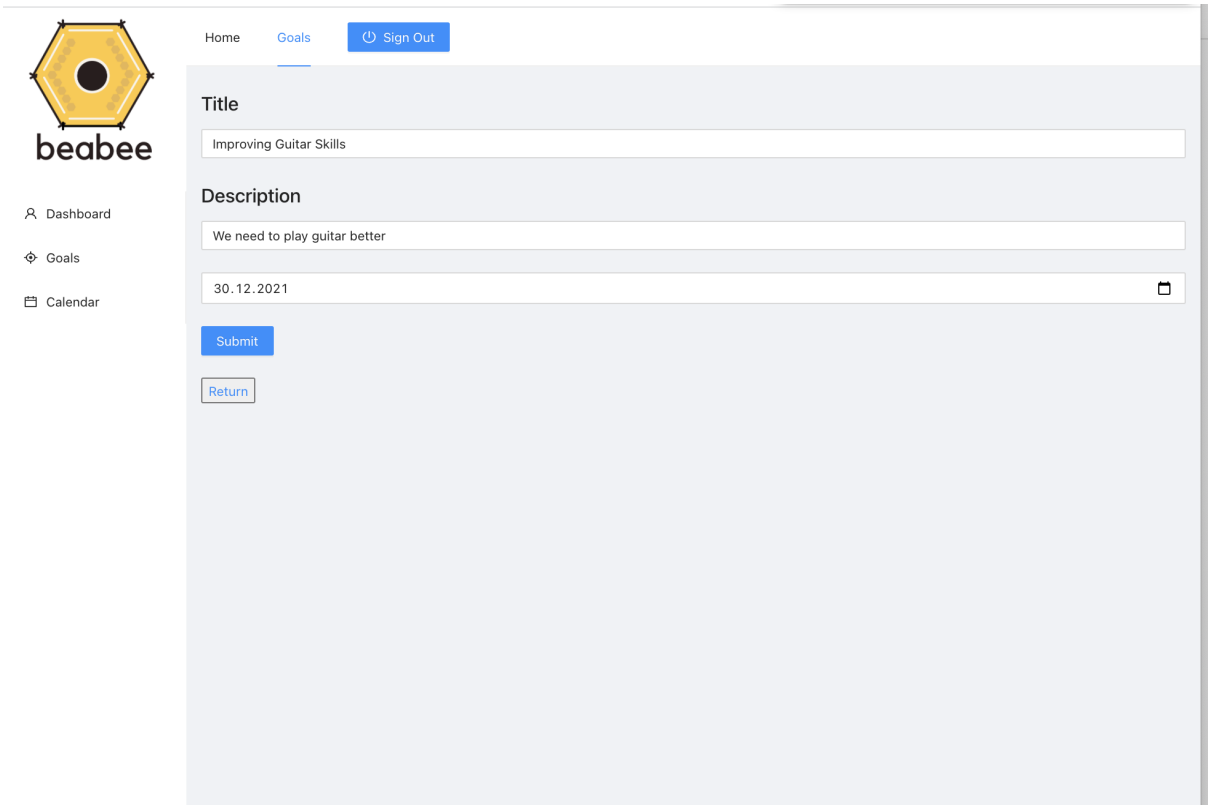
Goals Page

[2021SpringGroup12/GoalsPage.tsx at frontend/dev · bounswe/2021SpringGroup12 (github.com)](https://github.com/bounswe/2021SpringGroup12)

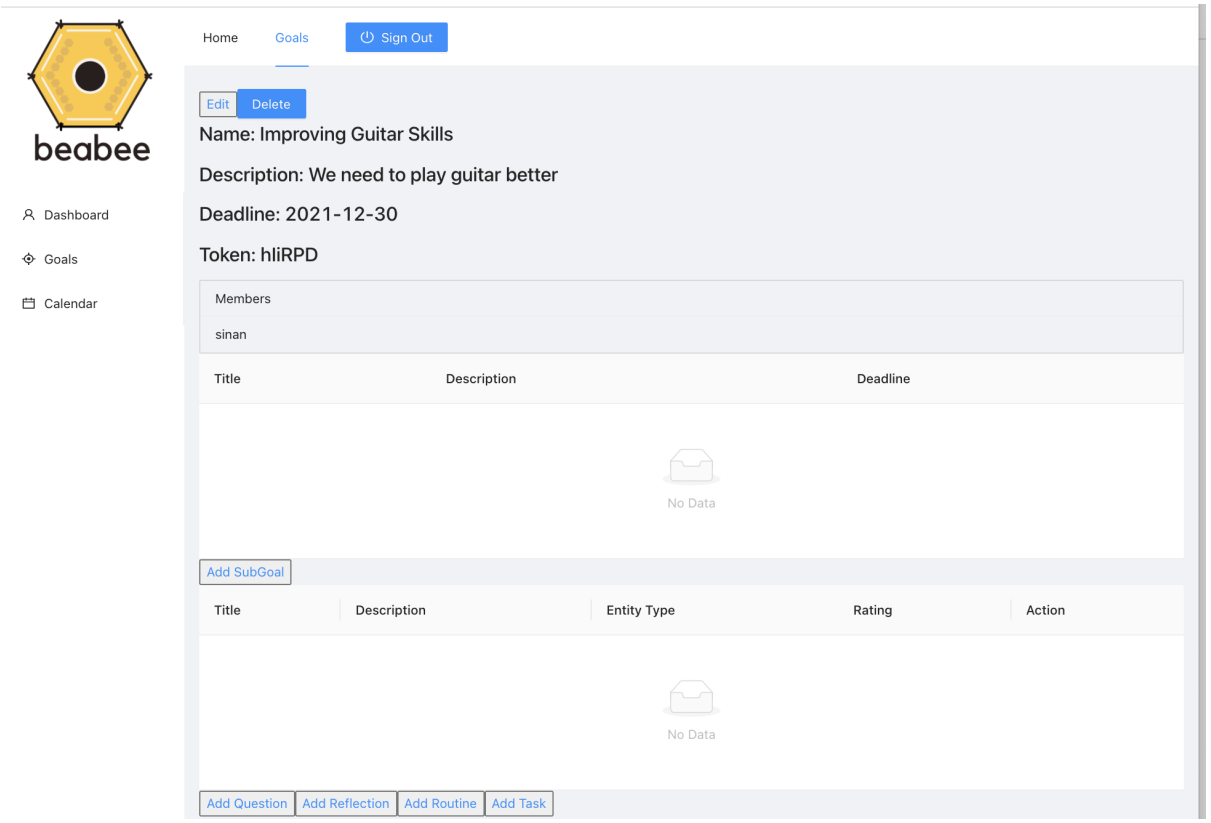


Adding Group Goal Page

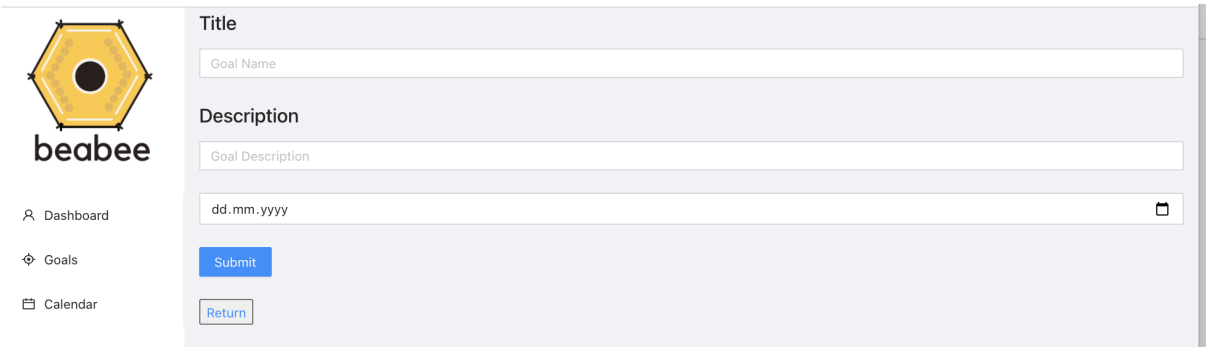
[2021SpringGroup12/AddGoal.tsx at frontend/dev · bounswe/2021SpringGroup12 (github.com)](https://github.com/bounswe/2021SpringGroup12)



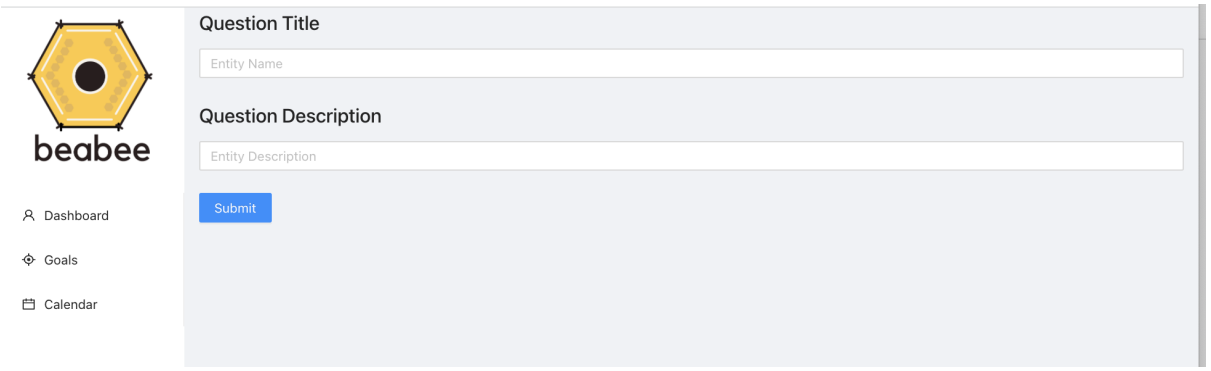
Group Goal Page
[2021SpringGroup12/GoalPage.tsx at frontend/dev · bounswe/2021SpringGroup12 (github.com)](https://github.com/bounswe/2021SpringGroup12)



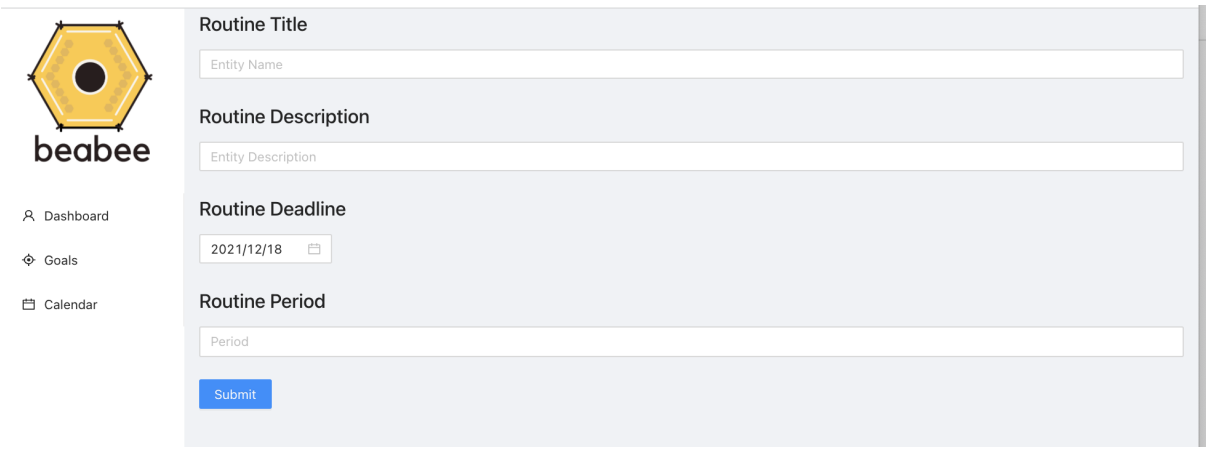
Add Sub-goal
[2021SpringGroup12/AddGoal.tsx at frontend/dev · bounswe/2021SpringGroup12 (github.com)]



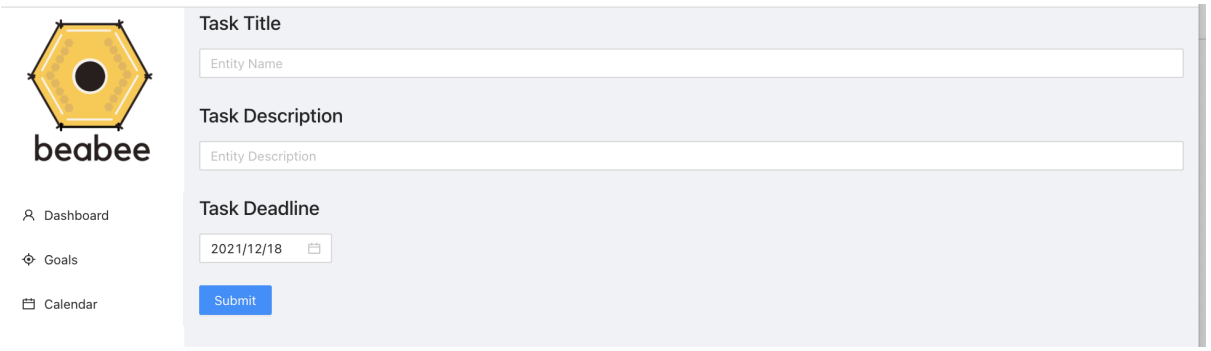
Add Question Page (Add reflection page is same with different names)
[2021SpringGroup12/AddEntity.tsx at frontend/dev · bounswe/2021SpringGroup12 (github.com)]



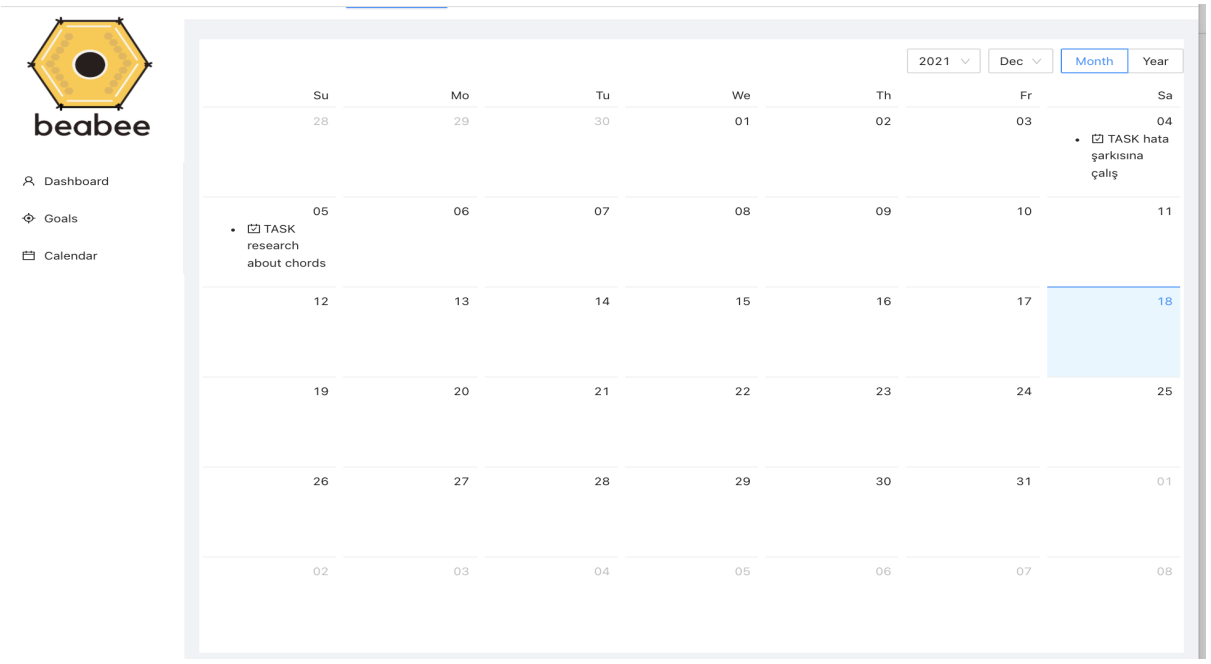
Add Routine Page
[2021SpringGroup12/AddEntity.tsx at frontend/dev · bounswe/2021SpringGroup12 (github.com)]



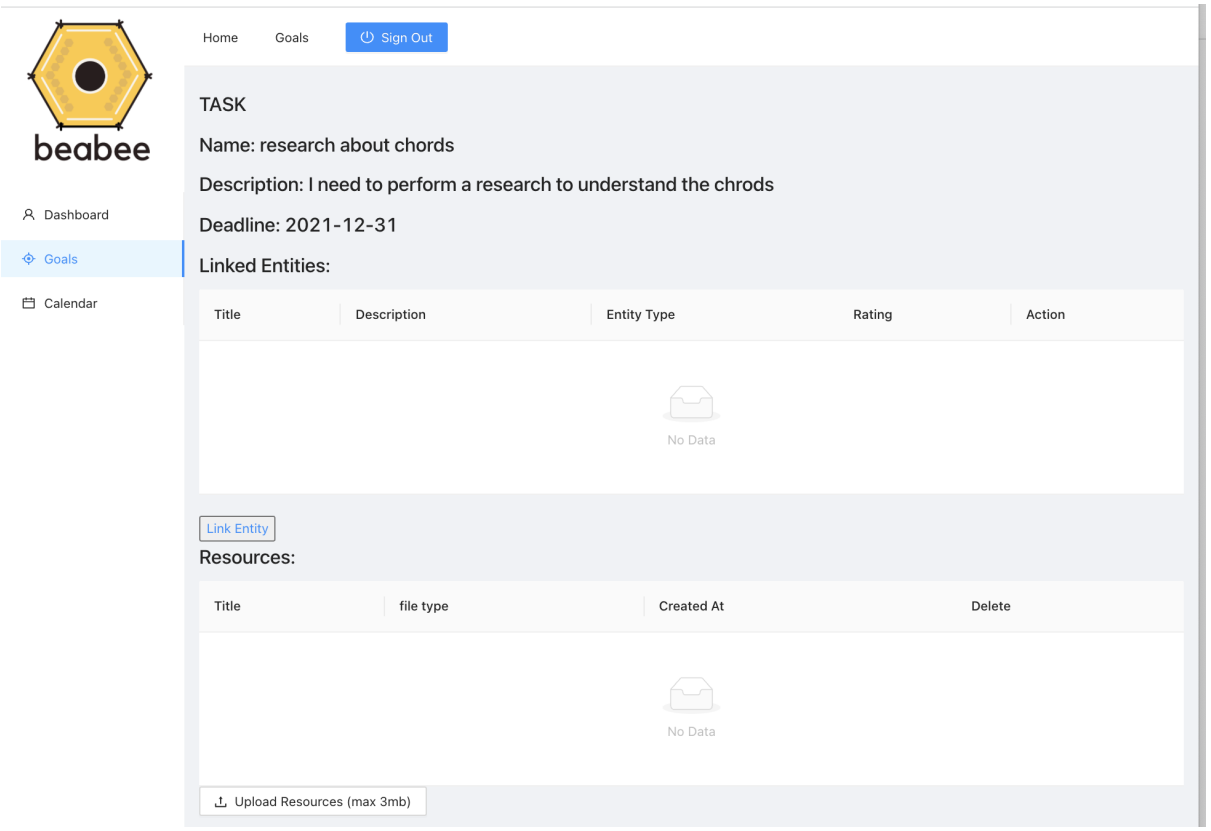
Add Task Page
[2021SpringGroup12/AddEntity.tsx at frontend/dev · bounswe/2021SpringGroup12 (github.com)]



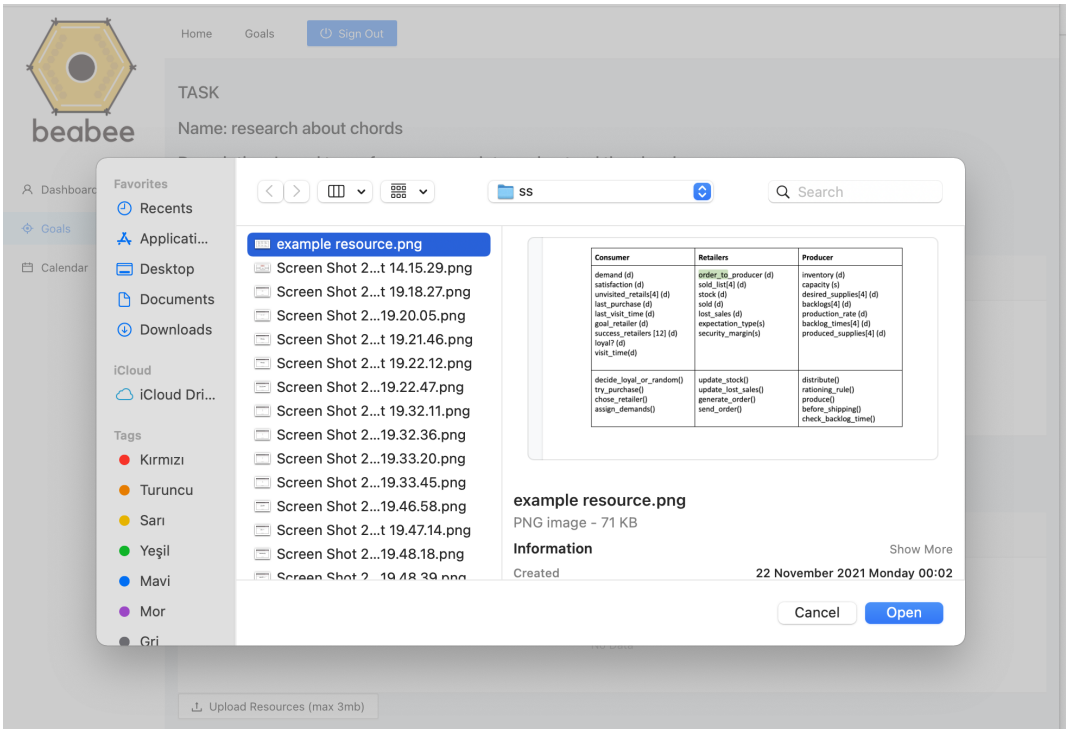
Calendar
[2021SpringGroup12/CalendarPage.tsx at frontend/dev · bounswe/2021SpringGroup12 (github.com)]



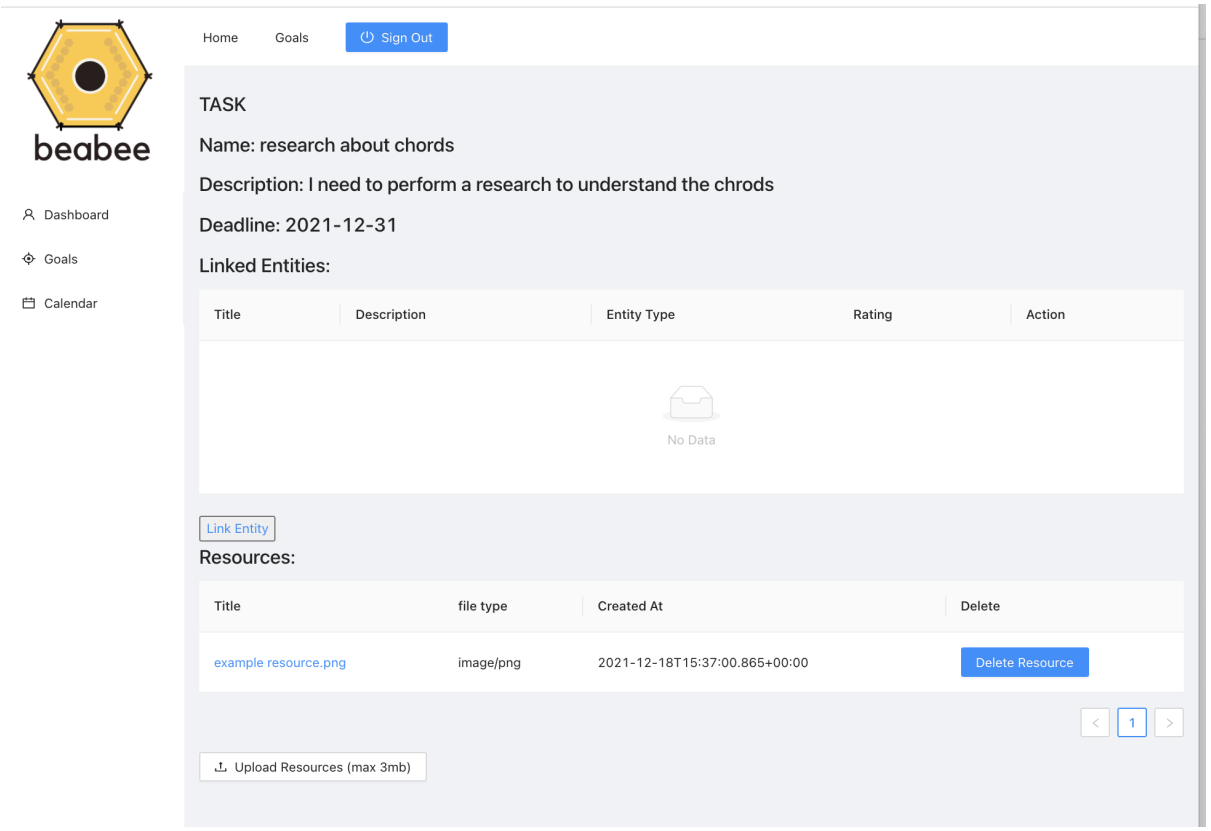
Entity Pages(Task will be given as an instance)
[2021SpringGroup12/EntityPage.tsx at frontend/dev · bounswe/2021SpringGroup12 (github.com)]



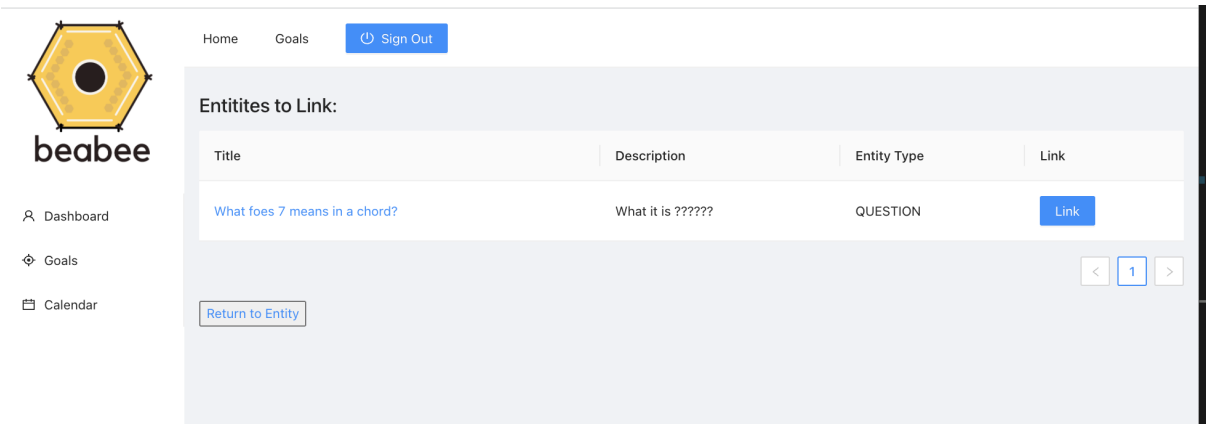
Adding a resource
[2021SpringGroup12/EntityPage.tsx at frontend/dev · bounswe/2021SpringGroup12 (github.com)]



After adding a resource
[2021SpringGroup12/EntityPage.tsx at frontend/dev · bounswe/2021SpringGroup12 (github.com)]

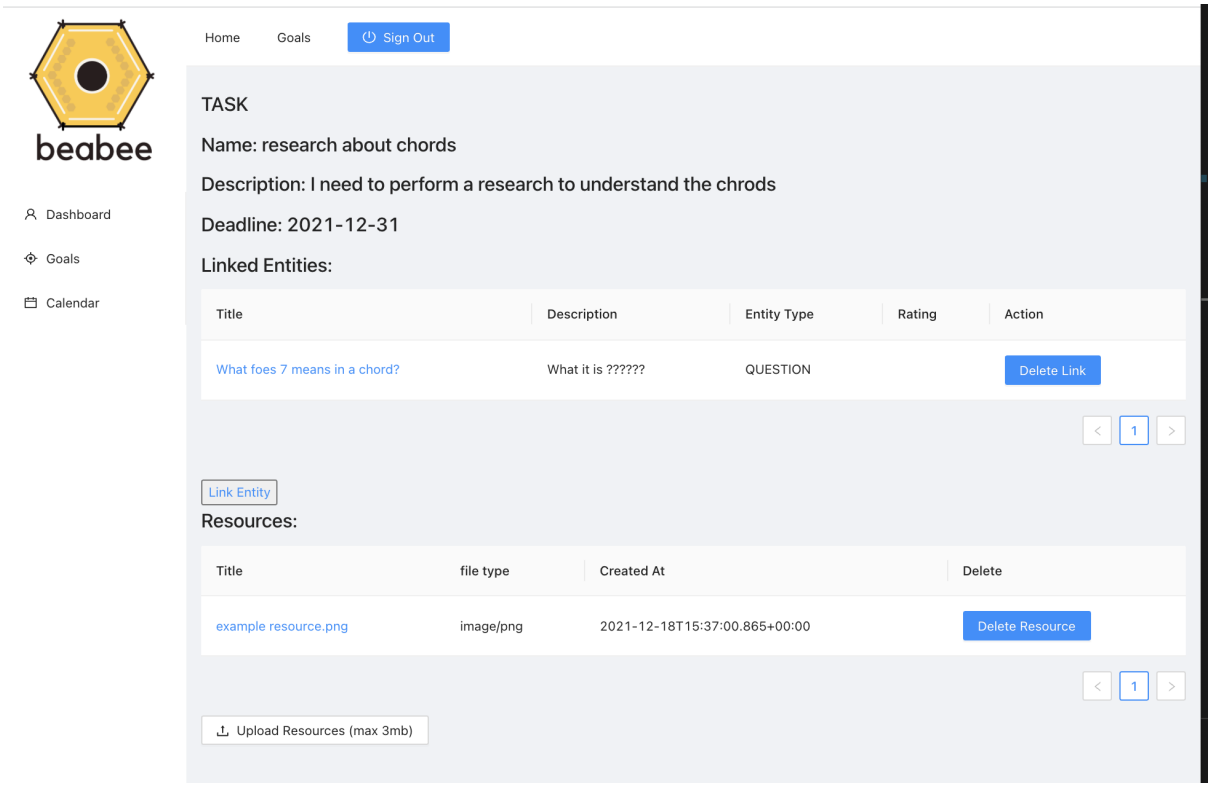


Linking Entities
[2021SpringGroup12/LinkEntity.tsx at frontend/dev · bounswe/2021SpringGroup12 (github.com)]

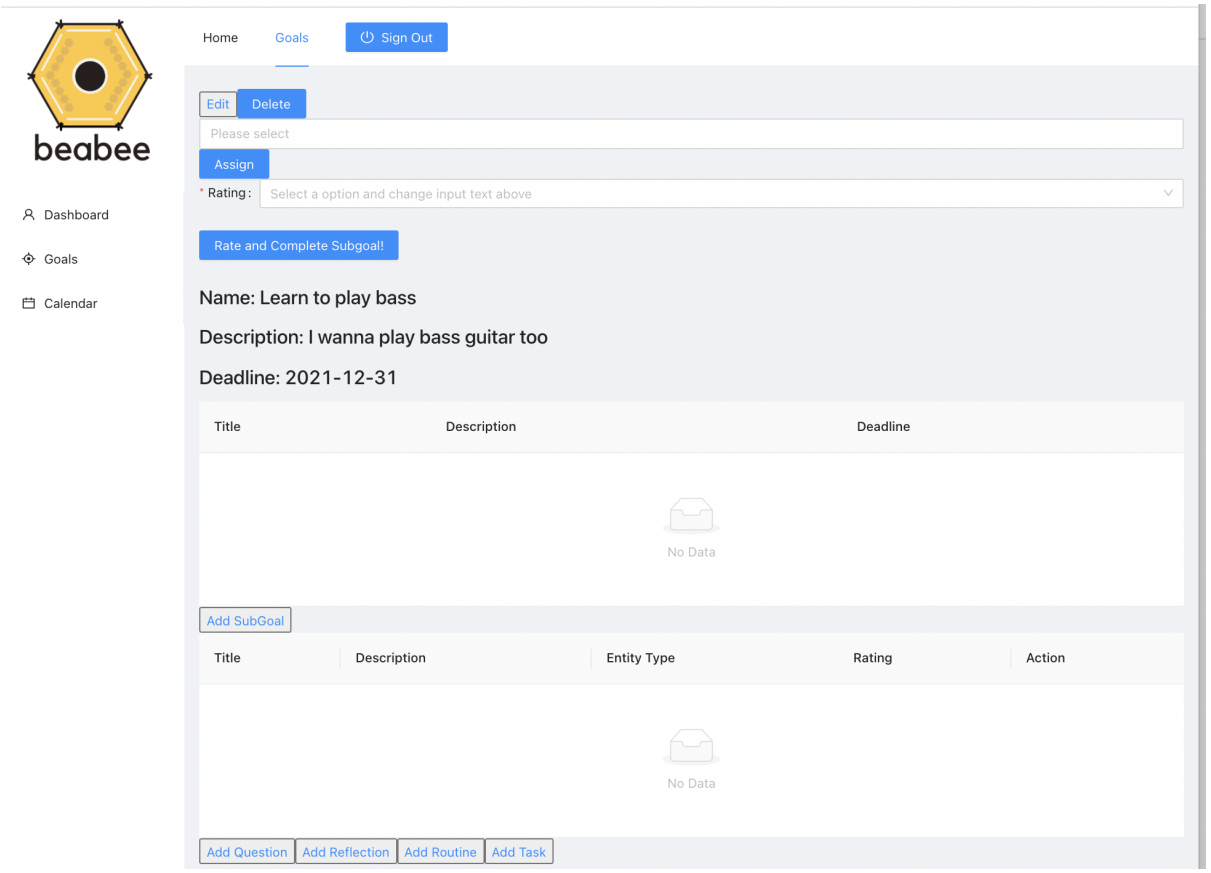


After clicking Link Button
[2021SpringGroup12/LinkEntity.tsx at frontend/dev · bounswe/2021SpringGroup12 (github.com)]

,

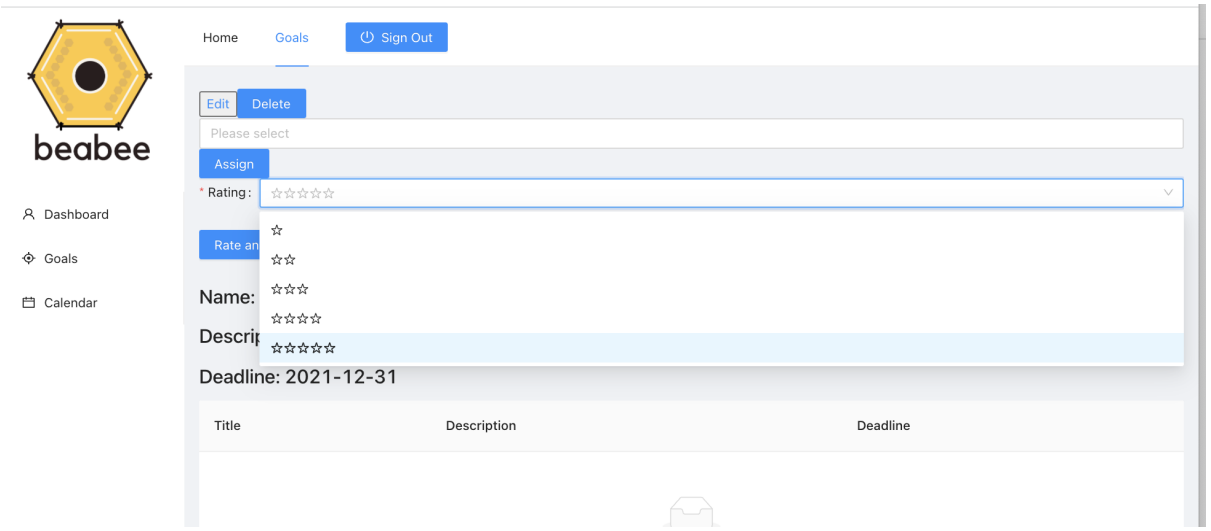


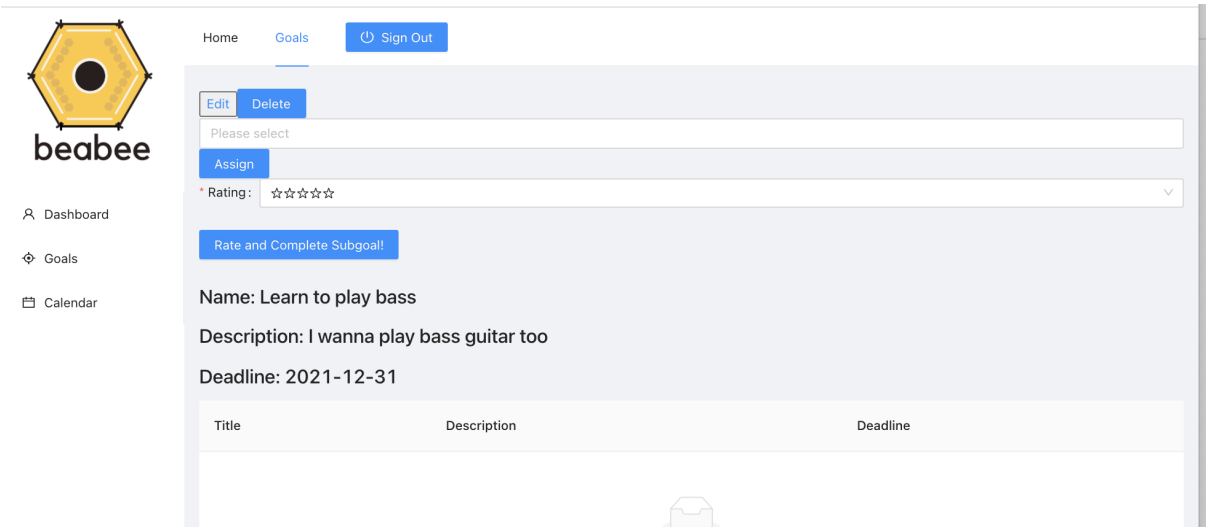
Subgoal Page
[2021SpringGroup12/GoalPage.tsx at frontend/dev · bounswe/2021SpringGroup12 (github.com)]



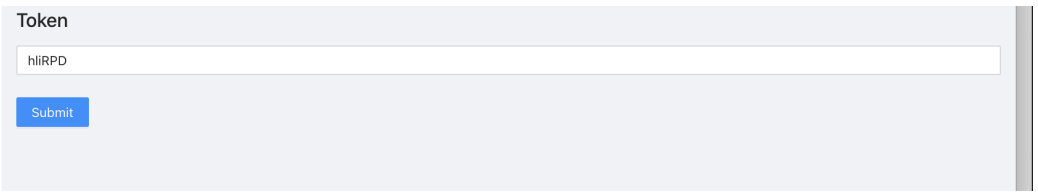
Rate the subgoal and close it

2021SpringGroup12/GoalPage.tsx at frontend/dev · bounswe/2021SpringGroup12 (github.com)





Joining a group with a token (batuhan will join sinan's group goal)
[2021SpringGroup12/GoalsPage.tsx at frontend/dev · bounswe/2021SpringGroup12 (github.com)]



beabee

Dashboard

Goals

Calendar

Name: Improving Guitar Skills

Description: We need to play guitar better

Deadline: 2021-12-30

Token: hliRPD

Members

sinan

batuhan

Title	Description	Deadline
Learn to play bass	I wanna play bass guitar too	2021-12-31

< 1 >

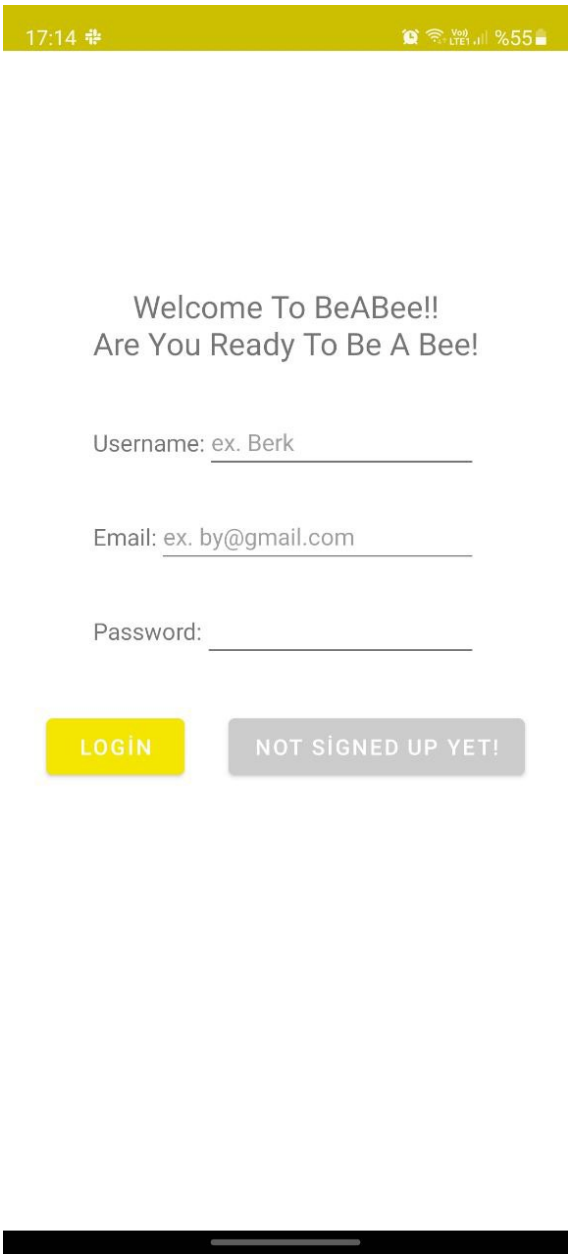
Add SubGoal

Title	Description	Entity Type	Rating	Action
research about chords	I need to perform a research to understand the chrods	TASK		Delete Edit
What foes 7 means in a chord?	What it is ??????	QUESTION		Delete Edit

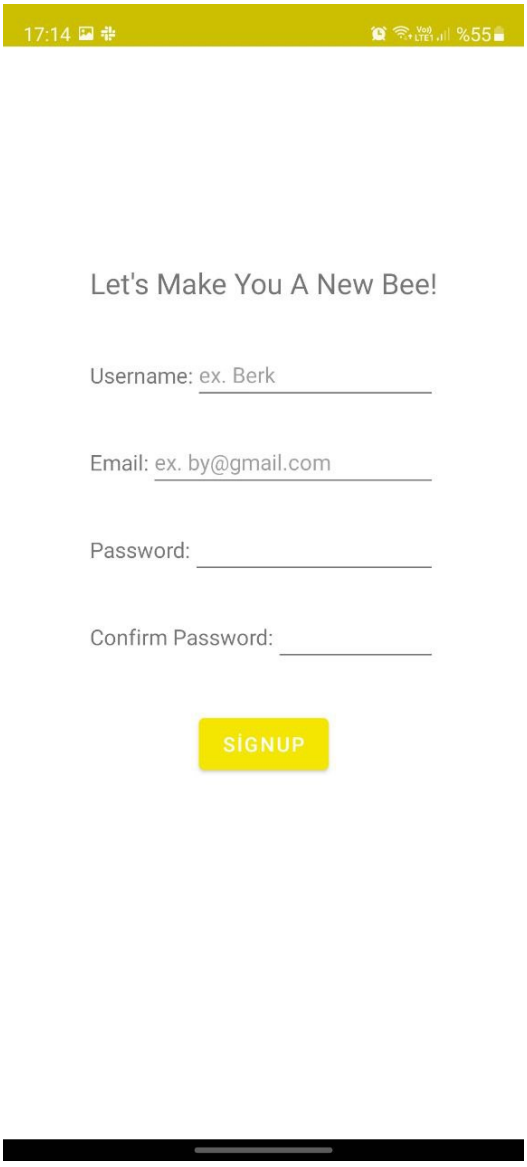
< 1 >

Add Question | Add Reflection | Add Routine | Add Task

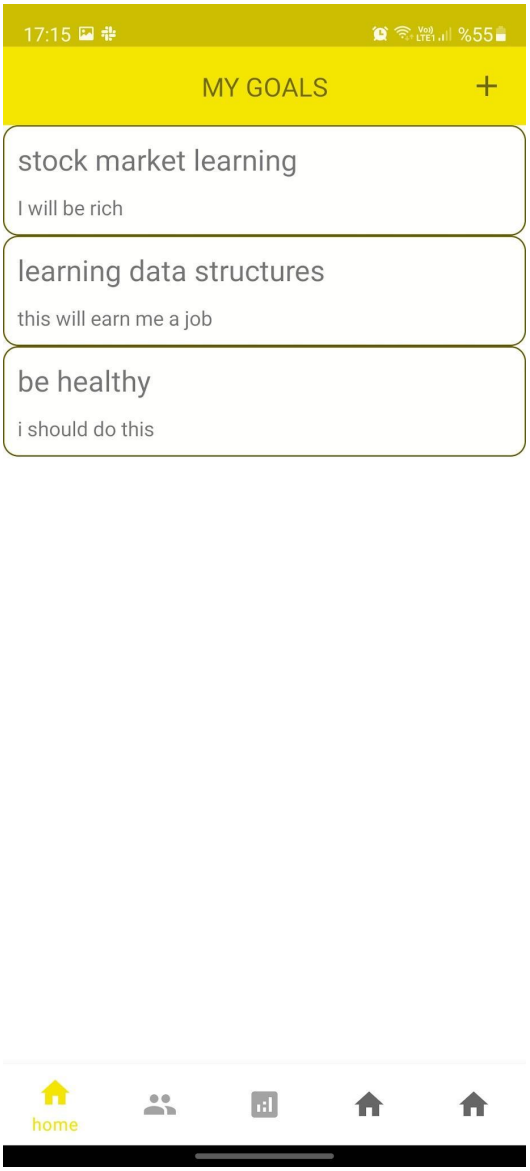
Mobile



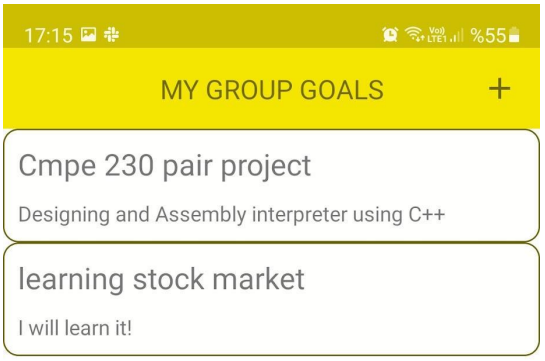
Login Page:https://github.com/bounswe/2021SpringGroup12/blob/android/dev/application/android/app/src/main/java/com/group12/beabee/views/LoginActivity.java

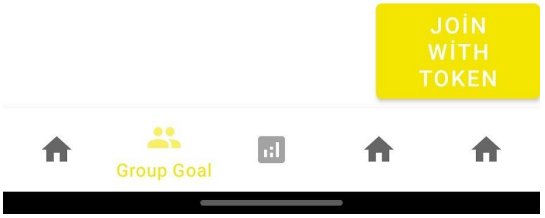


Signup Page:
https://github.com/bounswe/2021SpringGroup12/blob/android/dev/application/android/app/src/main/java/com/group12/beabee/views/SignUpActivity.java



Goals Page:
https://github.com/bounswe/2021SpringGroup12/blob/android/dev/application/android/app/src/main/java/com/group12/beabee/views/goals/HomeFragment.java

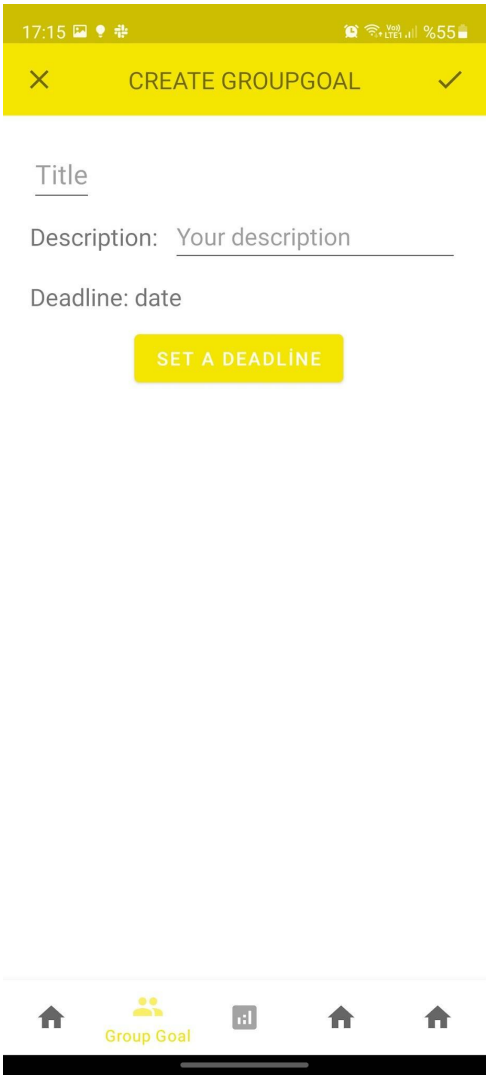




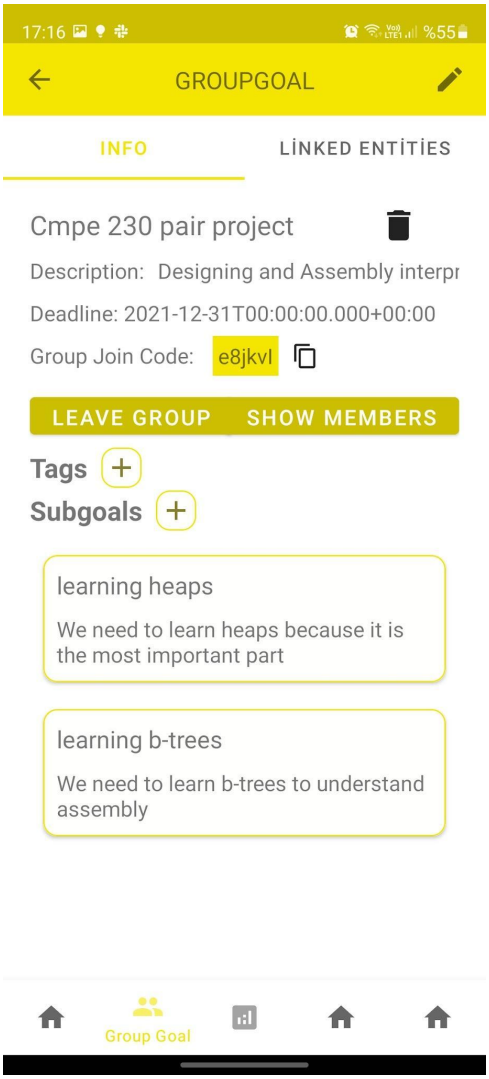
The Group Goals Page:

- https://github.com/bounswe/2021SpringGroup12/blob/android/dev/application/android/app/src/main/java/com/group12/beabee/views/goals/Home2Fragment.java

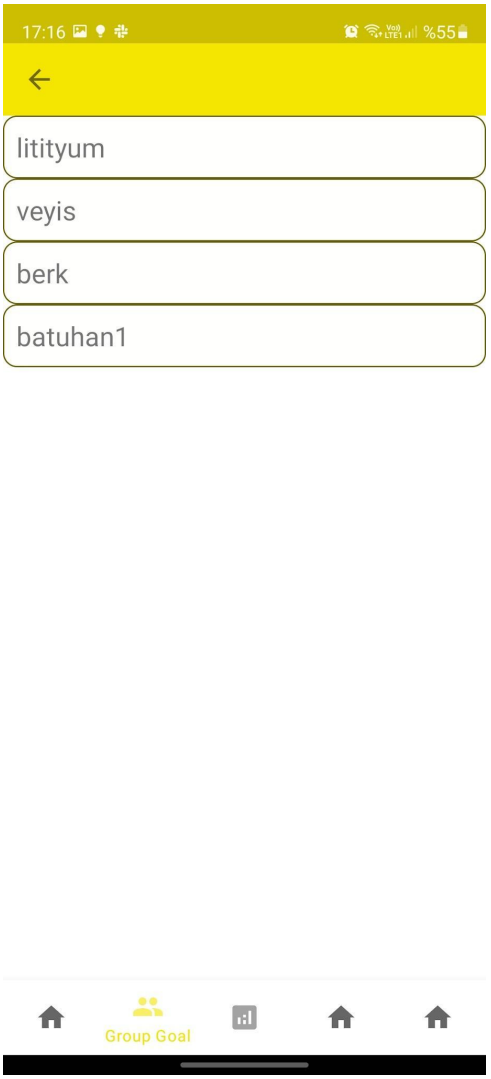
Join to aGroup Goal Page:
https://github.com/bounswe/2021SpringGroup12/blob/android/dev/application/android/app/src/main/java/com/group12/beabee/views/goals/JoinTokenFragment.java



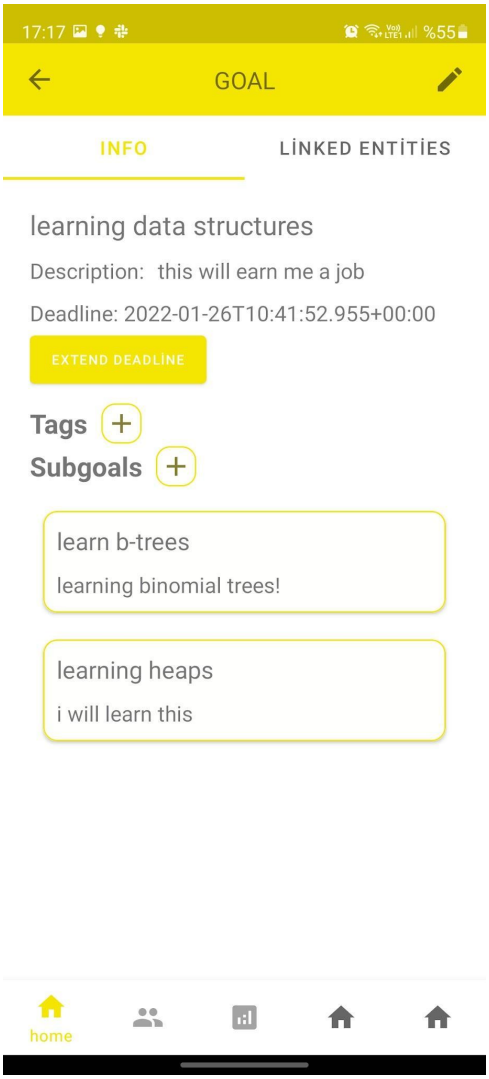
Create a Group Goal Page:
https://github.com/bounswe/2021SpringGroup12/blob/android/dev/application/android/app/src/main/java/com/group12/beabee/views/goals/GroupGoalCreateFragment.java



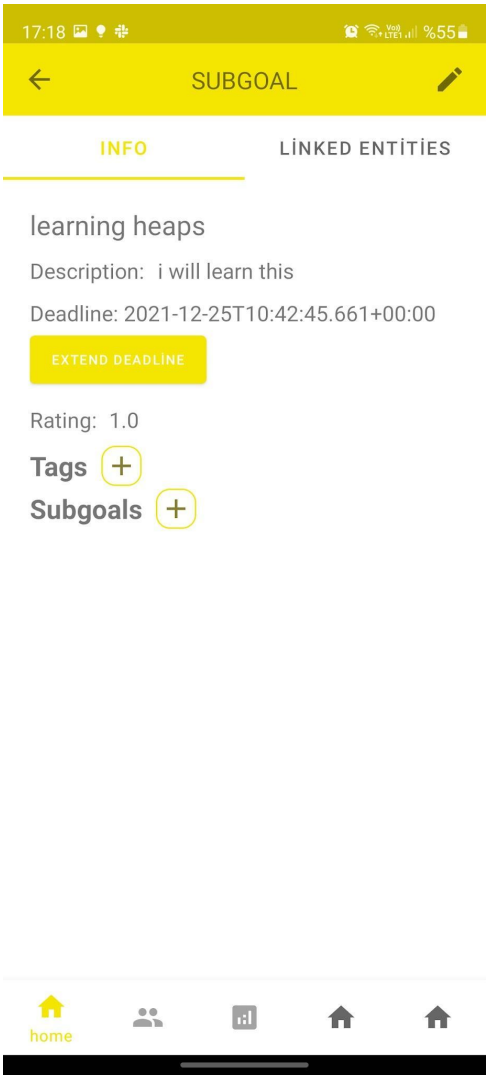
Group Goal Page:
https://github.com/bounswe/2021SpringGroup12/blob/android/dev/application/android/app/src/main/java/com/group12/beabee/views/goals/GroupGoalFragment.java



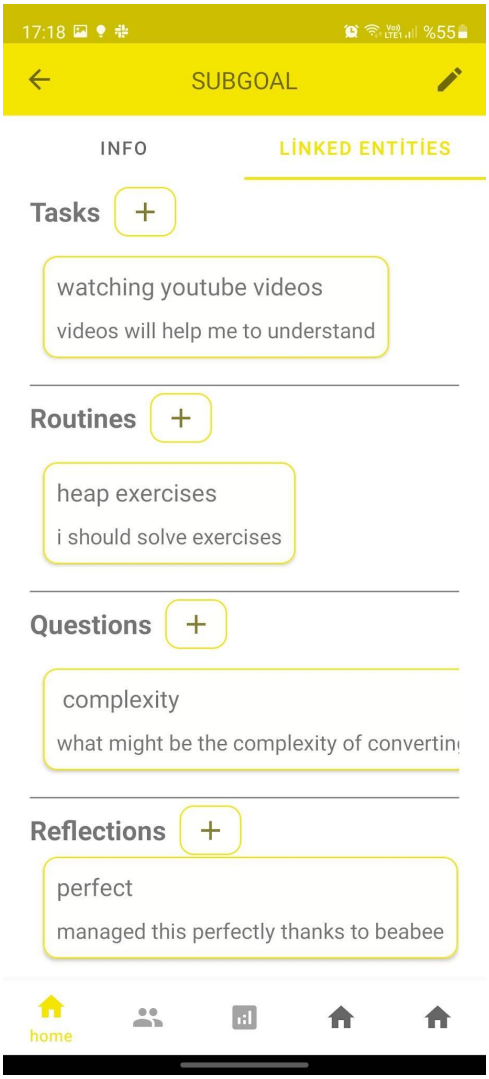
Group Goal Members Page:
https://github.com/bounswe/2021SpringGroup12/blob/android/dev/application/android/app/src/main/java/com/group12/beabee/views/goals/MemberListFragment.java



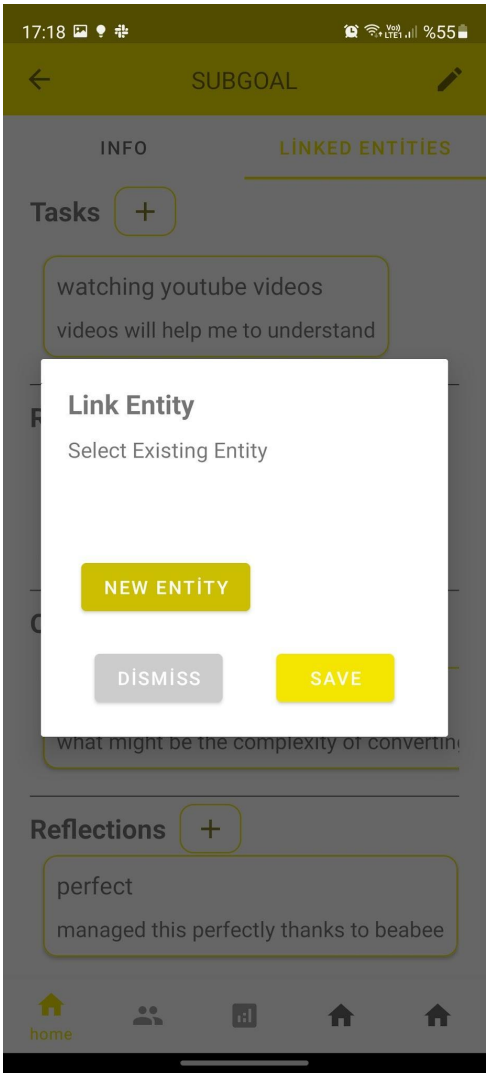
Goal Page:
https://github.com/bounswe/2021SpringGroup12/blob/android/dev/application/android/app/src/main/java/com/group12/beabee/views/goals/GoalFragment.java



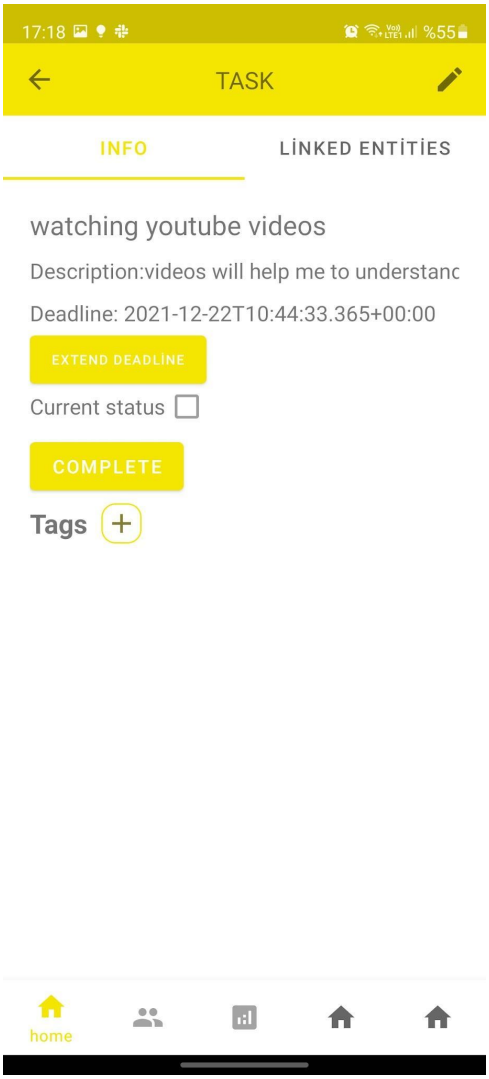
Subgoal Page:
https://github.com/bounswe/2021SpringGroup12/blob/android/dev/application/android/app/src/main/java/com/group12/beabee/views/goals/SubgoalFragment.java



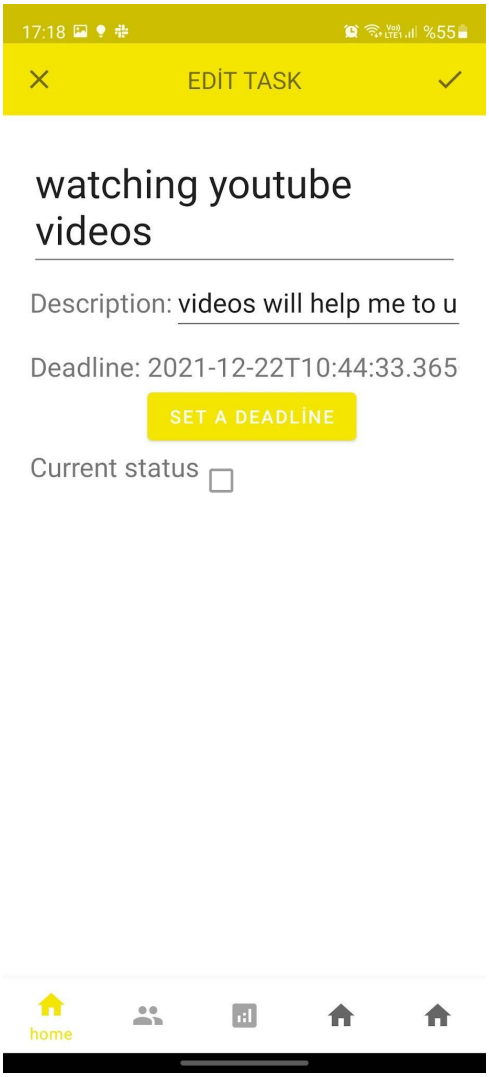
Subgoal Page:
https://github.com/bounswe/2021SpringGroup12/blob/android/dev/application/android/app/src/main/java/com/group12/beabee/views/goals/SubgoalFragment.java



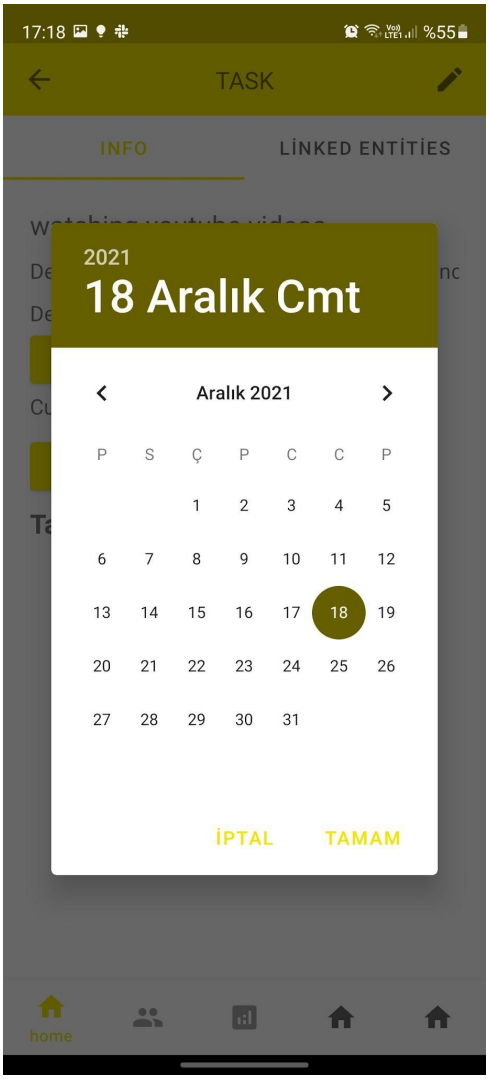
Entity selector dialog:https://github.com/bounswe/2021SpringGroup12/blob/android/dev/application/android/app/src/main/java/com/group12/beabee/views/dialogs/EntitySelectorDialog.java



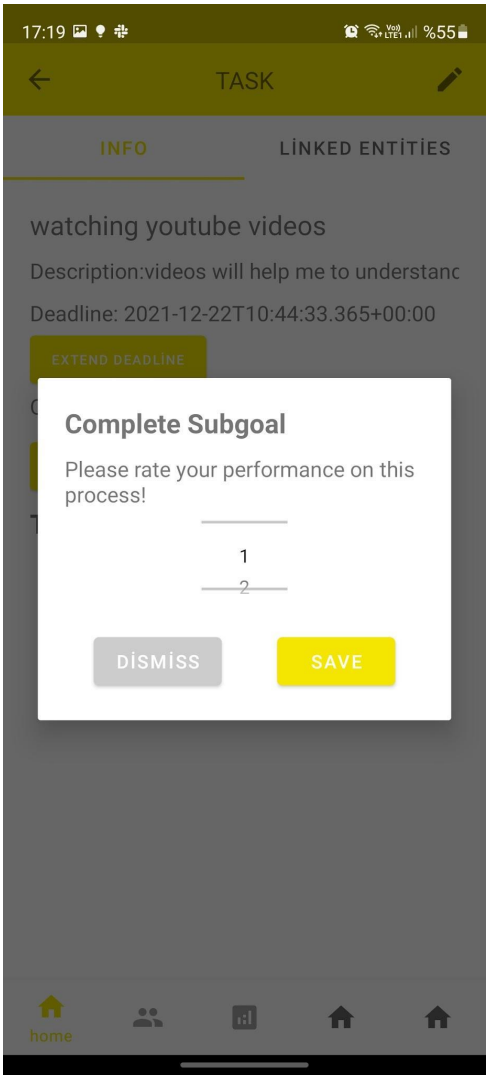
Task Page:
https://github.com/bounswe/2021SpringGroup12/blob/android/dev/application/android/app/src/main/java/com/group12/beabee/views/entities/TaskFragment.java



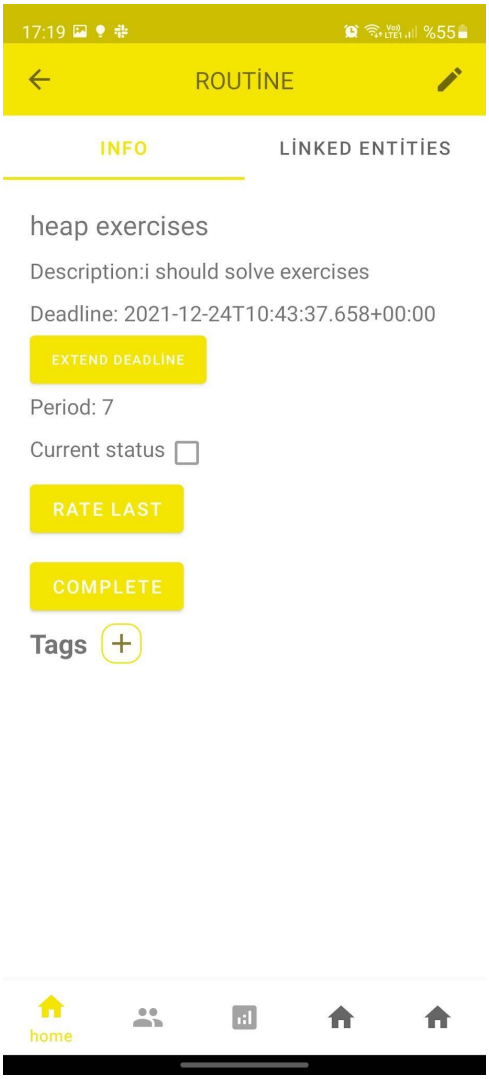
Edit Task Page:
https://github.com/bounswe/2021SpringGroup12/blob/android/dev/application/android/app/src/main/java/com/group12/beabee/views/entities/TaskFragmentEdit.java



Deadline extension:https://github.com/bounswe/2021SpringGroup12/blob/android/dev/application/android/app/src/main/java/com/group12/beabee/views/entities/DeadlineCalendarFragment.java

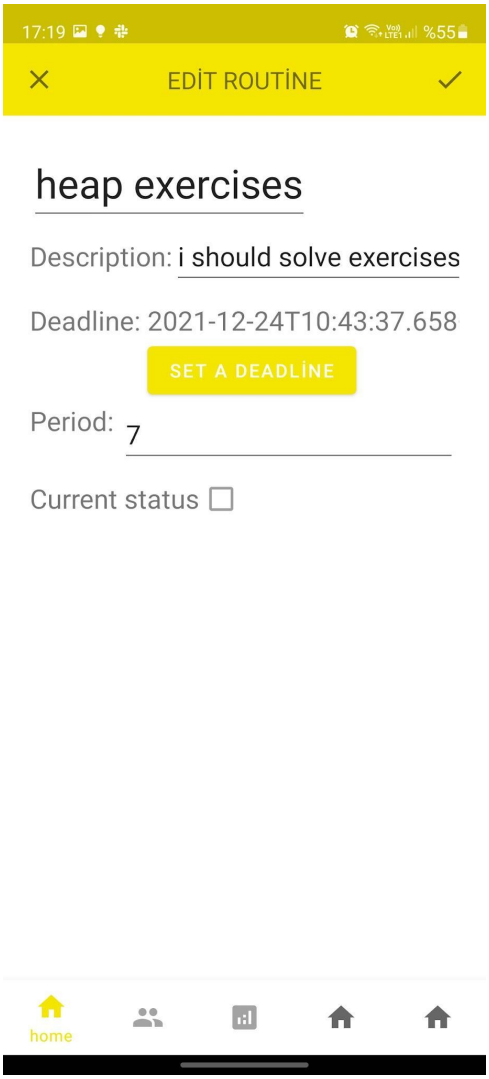


Complete and rate:
https://github.com/bounswe/2021SpringGroup12/blob/android/dev/application/android/app/src/main/java/com/group12/beabee/Utils.java

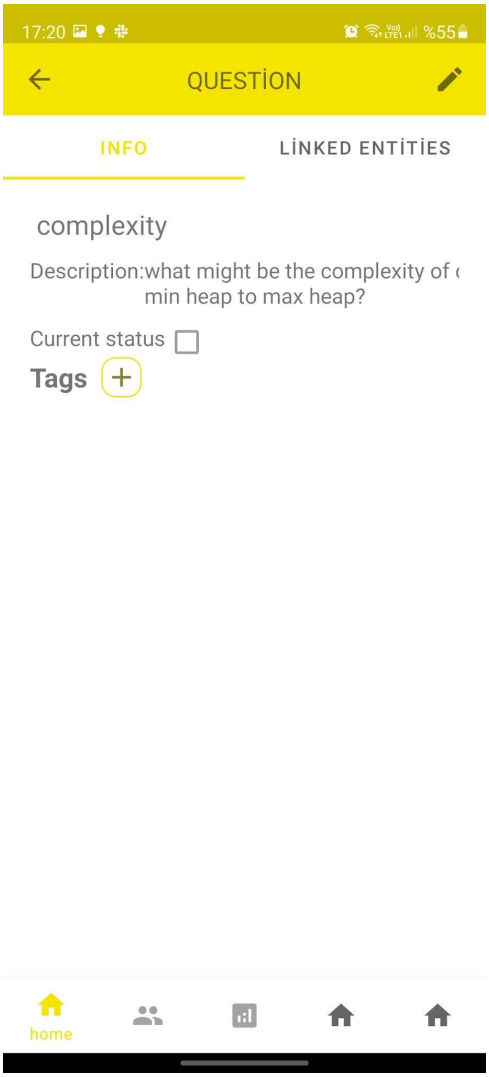


Routine Page:
https://github.com/bounswe/2021SpringGroup12/blob/android/dev/application/android/app/src/main/java/com/group12/beabee/views/entities/RoutineFragment.java

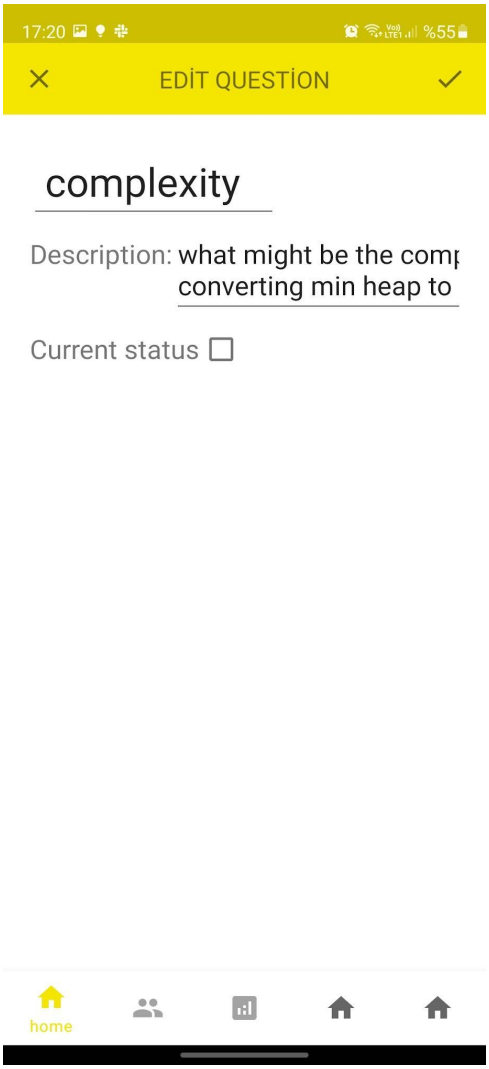
Deadline selection:https://github.com/bounswe/2021SpringGroup12/blob/android/dev/application/android/app/src/main/java/com/group12/beabee/views/entities/DeadlineCalendarFragment.java



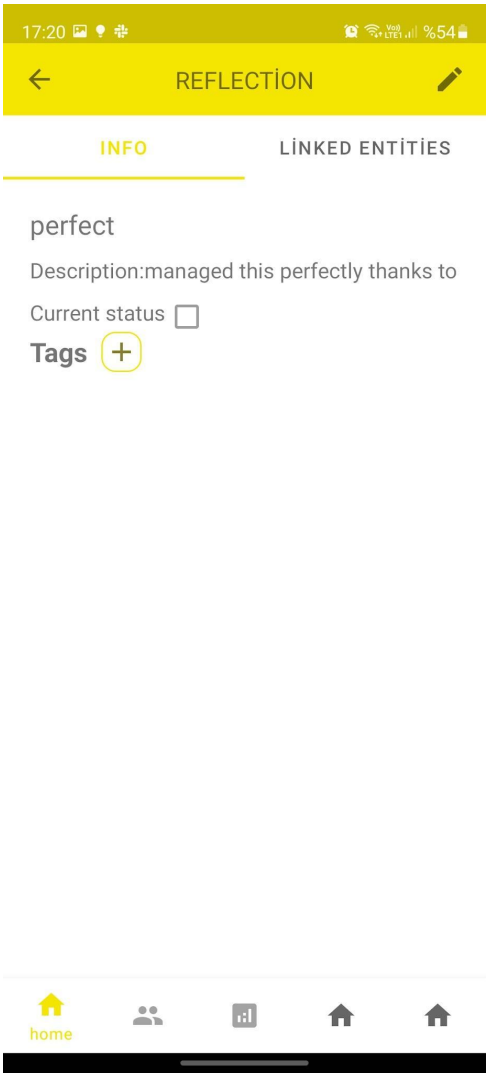
Edit Routine Page:
https://github.com/bounswe/2021SpringGroup12/blob/android/dev/application/android/app/src/main/java/com/group12/beabee/views/entities/RoutineFragment.java



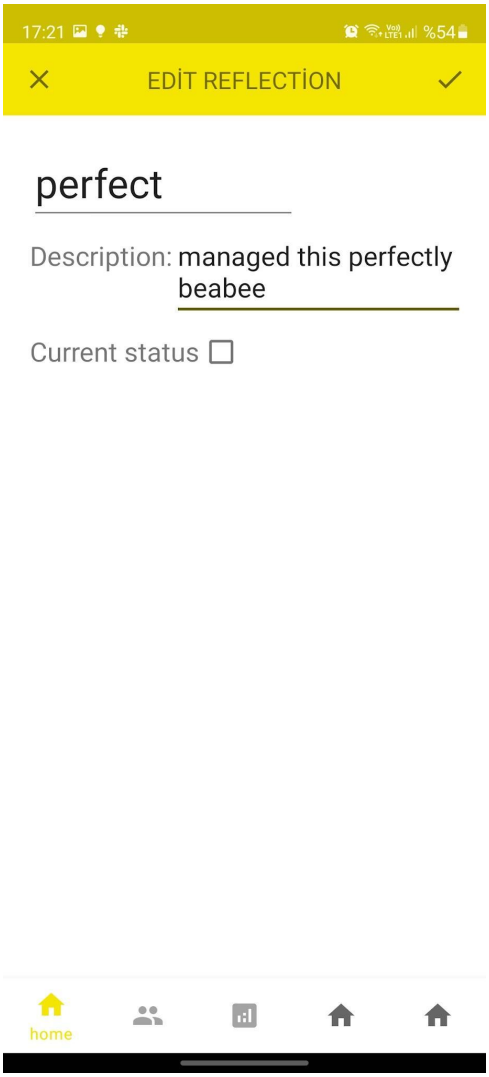
Question Page:
https://github.com/bounswe/2021SpringGroup12/blob/android/dev/application/android/app/src/main/java/com/group12/beabee/views/entities/RoutineFragment.java



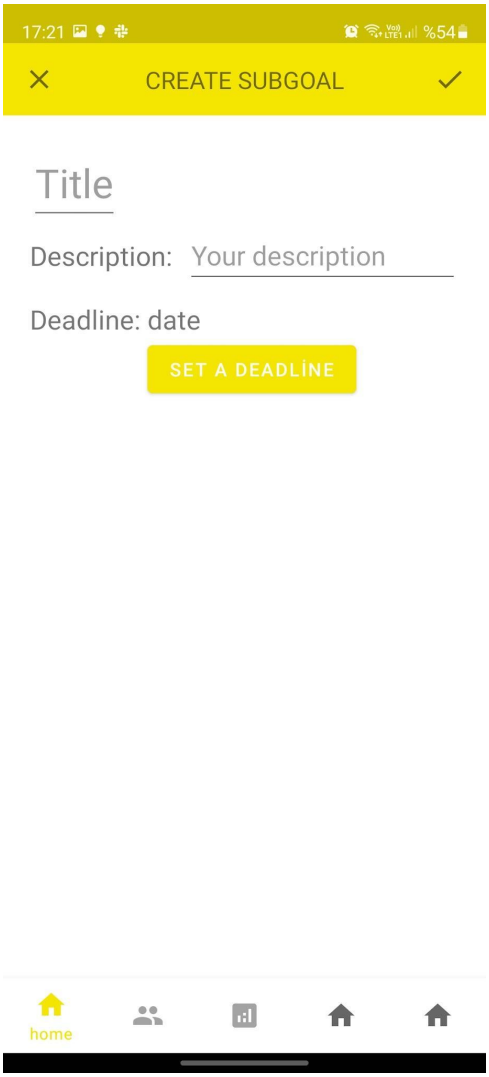
Edit Question Page:
https://github.com/bounswe/2021SpringGroup12/blob/android/dev/application/android/app/src/main/java/com/group12/beabee/views/entities/QuestionFragmentEdit.java



Reflection Page:
https://github.com/bounswe/2021SpringGroup12/blob/android/dev/application/android/app/src/main/java/com/group12/beabee/views/entities/ReflectionFragment.java



Edit Reflection:
https://github.com/bounswe/2021SpringGroup12/blob/android/dev/application/android/app/src/main/java/com/group12/beabee/views/entities/ReflectionFragmentEdit.java



Create Subgoal Page:
https://github.com/bounswe/2021SpringGroup12/blob/android/dev/application/android/app/src/main/java/com/group12/beabee/views/goals/SubgoalCreateFragment.java

Analytics Page:
https://github.com/bounswe/2021SpringGroup12/blob/android/dev/application/android/app/src/main/java/com/group12/beabee/views/goals/Home3Fragment.java

Standards

Wikidata

We have researched it and decided to use the following Java library: https://github.com/Wikidata/Wikidata-Toolkit. Will begin implementation immediately after this report.

W3C Activity Stream 2.0

We have not started it yet. Planning to deliver it in the final milestone.