Login.kt

The `login.kt` file implements a login screen in an Android app using Firebase for authentication. Key points:

LoginViewModel:

- Uses a ViewModel to manage authentication and obtain user information.

LoginScreenPane:

- Displays a login interface with a logo, an image pager, and descriptive text.
- Includes a button to log in via Google, handled with a callback function to navigate to the team list after login.

In summary, this file handles the application's login interface and uses Compose for UI composition along with Firebase for user authentication.

TeamList.kt

The `TeamList.kt` file manages the display of a list of teams:

TeamListViewModel:

- ViewModel to manage the team data logic.
- Methods to get lists of teams and members, and to filter teams.

TeamListPane:

- Main screen that displays the teams.
- Uses Scaffold with a CenterAlignedTopAppBar for the title and a filter icon.
- BottomAppBar with navigation between app sections.
- FloatingActionButton to add a new team.
- Displays a ModalBottomSheet for advanced filter settings.
- Lists the teams using Column and Row, displaying cards (TeamCard) for each team with user images, categories, and associated actions.

TeamCardBuilder:

- Builds a single card for the team.
- Displays the team and user images using AsyncImage.
- Handles various actions such as navigations and team operations.

AssignedUserIcons and ClickhereToAssignUserIcon:

- Components to display the assigned user images and the icon to add a new user.

This file implements a dynamic screen using Compose for UI management and offers real-time data interactions using ViewModel and Firebase.

TeamDetails.kt

The `TeamDetails.kt` file displays the details of a team in an Android app using Jetpack Compose:

TeamDetailsViewModel:

- ViewModel to manage the team details data logic.
- Methods to get specific team information and associated members.

TeamDetailsPane:

- Main screen that shows the details of a specific team.
- Uses Scaffold with a CenterAlignedTopAppBar for the team title and a back navigation icon.
- Displays the team image using AsyncImage.
- Uses ElevatedCard to show detailed information such as category, creation date, and team description.
- Lists team members with their names and images, using Button and AsyncImage to display profile images.

This file implements a detailed screen using Jetpack Compose for dynamic UI management and interaction with team and member data.

TeamAchievement.kt

The `TeamAchievement.kt` file manages the display of team achievements in an Android app using Jetpack Compose:

TeamAchievementViewModel:

- ViewModel to manage the team achievements data logic.
- Methods to get specific team information and associated members.

TeamAchievementPane:

- Main screen that shows the team's achieved goals.
- Uses Scaffold with a CenterAlignedTopAppBar for the team title and a back navigation icon.
- Displays a circular progress indicator with text indicating the project status.
- Lists team members represented through Card with their profile images.

CircularProgressIndicatorWithText:

- Composable that shows a customized circular progress indicator.
- Displays the project status text (e.g., "Project Status:") and progress percentage.

This file implements a screen to graphically display the team's achieved goals using Jetpack Compose for dynamic UI management and interaction with team and member data.

NewTeam.kt

The `NewTeam.kt` file manages the creation of a new team:

NewTeamViewModel:

- ViewModel that manages the business logic for creating a new team.
- Methods to get existing team information, add and remove members, and create a new team with an associated chat.
- Uses coroutines to handle asynchronous operations such as chat creation.

NewTeamPane:

- Main screen for creating a new team.
- Uses Scaffold with a customized TopAppBar.
- Uses a LaunchedEffect to load existing team members once the teamId is available.
- Contains fields to enter the team name, category, and description, managed through TextField.
- Allows users to add members to the team through a Dialog that shows available users with a checkbox to select them.
- Displays selected users with a visual list and allows removal with a delete icon.
- Handles team update and submission of data to the ViewModel for the actual team creation.

This file uses advanced Jetpack Compose components to handle complex user interactions such as selecting and adding members, making the team creation experience interactive and intuitive for users.

EditTeam.kt

This file proposes similar code to the content of `NewTeam.kt`, with the difference that it allows editing the characteristics of an existing team rather than creating a new one.

UserProfile.kt

This file allows viewing and editing user information if it is the profile of the logged-in user. It is divided into two main Panes:

UserPane:

- Allows viewing user details, specifically "Personal Information", task and team information linked to the user, role, rating, skills, and achievements.

EditPane:

- Allows, if the displayed user is the same as the logged-in user, to edit their personal information.

TaskList.kt

TaskListViewModel: Manages task logic, including filters, sorting, and data management. The `TaskListManager` function, annotated with @Composable, represents the main screen that shows and manages the task list with filtering and sorting options. The `TaskListPane` and `TaskCardComposable` functions are used to display individual tasks with details and user interactions.

TaskDetails.kt

Implements a detailed screen for viewing and managing a single task.

TaskDetailViewModel:

- This ViewModel class manages the business logic for retrieving task details, assigned members, and updating task status.

CountdownTimer:

- Composable that displays a linear progress bar and a countdown for the task based on its due date.

BoxIcon:

- Component that displays an icon or number inside a colored circle, used to represent task progress status.

CircularBorderButton:

- Round button with a border and an icon in the center, used for actions such as adding comments or other interactions.

TaskDetail:

- Main composable that shows all task details, including title, current status, due date, task history, description, assigned users, and a button to update task status. Includes a confirmation banner for confirming the status update.

ConfirmationDialog:

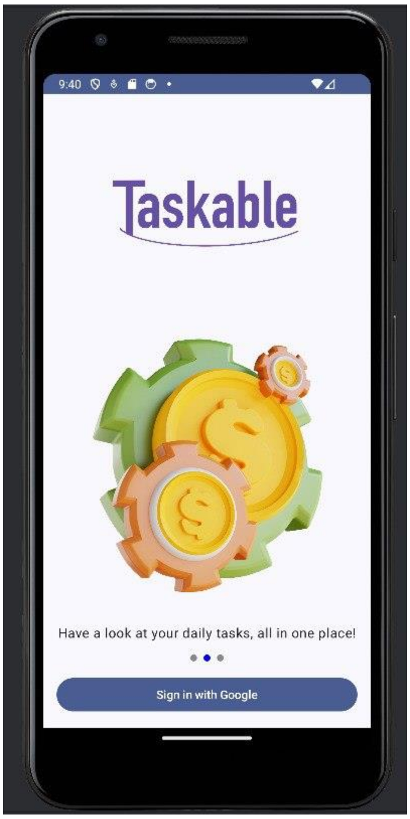
- Confirmation dialog displayed when the user attempts to change the task status, to ensure the user confirms their action.

NewEditTask.kt

This file allows users to modify or create a task from scratch. Unlike `UserProfile.kt`, both task modification and creation are managed within a single pane.

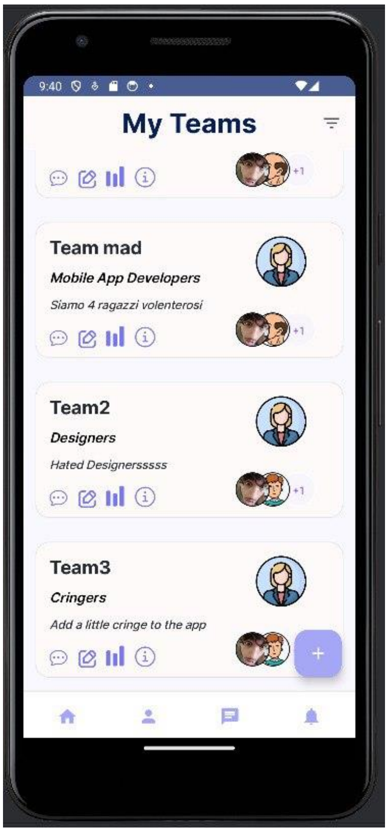
CommentSection.kt

Contains a Pane that allows users to comment on tasks related to the chat the task is linked to. It displays a chat screen between users, allowing viewing previous comments and adding new ones. Each comment indicates the user who wrote it by displaying their profile image.



Have a look at your daily tasks, all in one place!

Image 1 - Login.kt



Team mad

Mobile App Developers

Siamo 4 ragazzi volenterosi

Team2

Designers

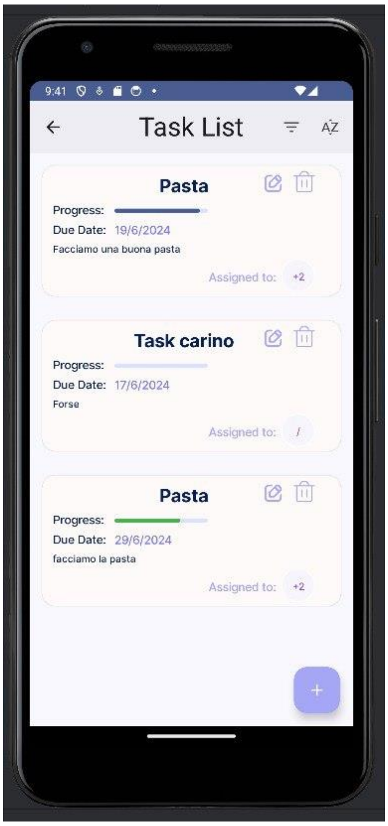
Hated Designersssss

Team3

Cringers

Add a little cringe to the app

Image 2 - TeamList.kt



Pasta

Progress:

Due Date: 19/6/2024

Facciamo una buona pasta

Assigned to: +2

Task carino

Progress:

Due Date: 17/6/2024

Forse

Assigned to: /

Pasta

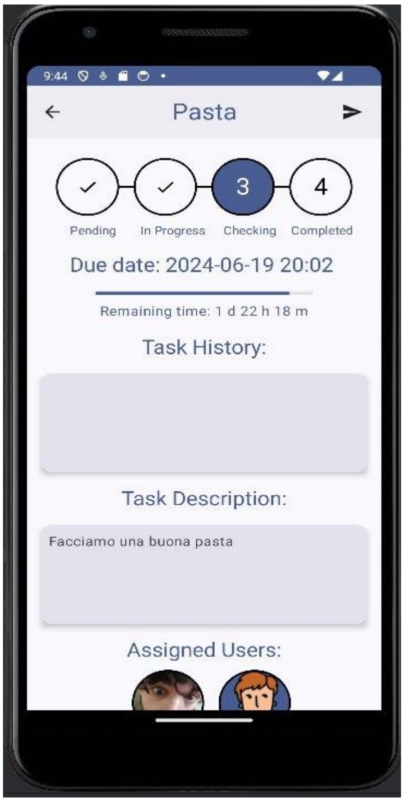
Progress:

Due Date: 29/6/2024

facciamo la pasta

Assigned to: +2

Image 3 - TaskList.kt



Due date: 2024-06-19 20:02

Remaining time: 1 d 22 h 18 m

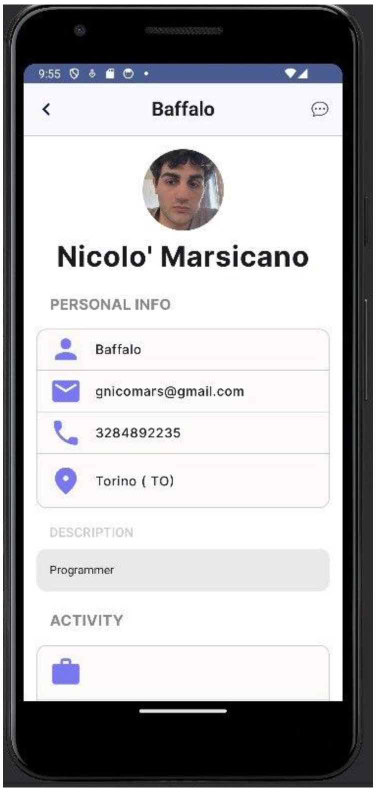
Task History:

Task Description:

Facciamo una buona pasta

Assigned Users:

Image 4 - TaskDetail.kt



Nicolo' Marsicano

PERSONAL INFO

Baffalo

gnicomars@gmail.com

3284892235

Torino (TO)

DESCRIPTION

Programmer

ACTIVITY

Image 5 - UserProfile.kt