TED UNIVERSITY

CMPE 492 Low-Level Design Report

Gizem ÖZYILDIZ

Eren Buldum

Ceren Bilge OYAR

Karya ERCAN

Table of Contents

1. Introduction

The idea behind the EcoFriends application is; How can we make the world a more livable and better place and at the same time create a fun and competitive environment. By using this app, users will be increasing their recycling rate and, accordingly, reduce their carbon footprint. By using this app, users will earn different points for each item they recycle, and they will be able to use these coins to create their own avatars, give them features, or compete with their friends with points. We will encourage our users who can earn points in different ways, for example: points earned based on the number of steps taken, points earned based on the items they recycle, for a more sustainable world.

1.1Object design trade-offs

In the development of the "EcoFriends" app, we encountered several design trade-offs to ensure an effective and user-friendly experience. These trade-offs address various aspects of the app's functionality and design philosophy:

1.1.1 Simplicity vs. Complexity

Our primary objective is to make "EcoFriends" user-friendly, encouraging individuals to participate in recycling activities with ease. To achieve this, we've focused on a straightforward user onboarding process, allowing users to register quickly and use the app after download. We aim for a clean and intuitive interface that users can navigate without a steep learning curve. We've ensured that the registration process is user-friendly, streamlining the initial setup while maintaining a clean and simple user experience. Our emphasis is on making the app accessible and easy to use after a brief registration process. Because, let alone being complex, on the contrary, the application should be able to appeal to the user as much as possible so that we can bring an issue such as recycling to the people as much as possible.

1.1.2 Efficiency vs. Accuracy

One critical feature of "EcoFriends" is the generation of a common leaderboard, tracking users' recycling achievements. As the user base grows, computing this leaderboard in real-time can become computationally intensive. However, to maintain app efficiency, we cannot just prioritize speed over absolute accuracy. Because a leaderboard means that there must be accuracy all the time. Due to the fact that if there is an issue in terms of accuracy, this can break the motivation of the user who is competitive. While our leaderboard may occasionally have minor delays, we ensure that the app remains responsive and doesn't suffer from performance issues due to extensive calculations.

1.1.3 Usability vs. Security

Balancing usability and security is paramount in "EcoFriends." While our primary goal is to provide a user-friendly experience, we also understand the critical importance of ensuring the security of user data and interactions. To achieve this, we have implemented a registration process that mandates users to create strong and unique passwords. While this may make the registration process slightly more time-consuming and involved, the advantage of enhanced security within the app makes it a worthwhile trade-off.

1.1.4 Reliability vs. Compatibility

"EcoFriends" strives to offer a dependable service where users can consistently participate in recycling and engage with the app without disruptions. We prioritize system reliability and robustness to prevent crashes and downtime. In the pursuit of reliability, we decided to focus on a single platform, primarily for mobile devices. This choice reduces the complexity of compatibility issues and allows us to concentrate on providing the best user experience possible, minimizing potential disruptions due to system errors.

1.2 Interface documentation guidelines

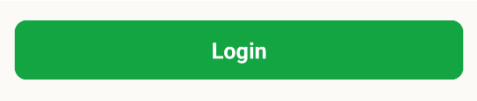
EcoFriends is an gamified app provides people to encourage recycling. With the help of the app, users earn coins for recycling different materials including paper, plastic bottles, and metal cans. Users can manually enter information to add their recycling activity to the app. Users can compete with peers using the app's competitive element, which is based on recycling activity and currency balance. An additional motivation for users to recycle is the ability to modify their avatar with the app's coin balance.

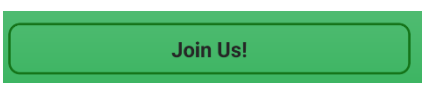
Color Scheme: We used specific color schemes for our application. For example we used a tone of green which is #12A541 for most of the time. We used other tones of green for various purposes like border color. In order not to tire the user's eyes, we changed the tone of white we used in the background of the application to #FBFAF7 instead of #FFFFFF. Likewise, we made the color tone of black lighter.

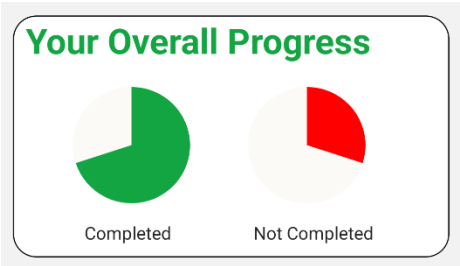
Typography: We used different font style for the “EcoFriends” text which will be “Oswald” apart from other texts. The reason behind this is we wanted to take attention to our logo. Also we were careful about the texts’ appearances to make them compatible with each other.

Iconography: As we all know, icons have an important place in an application. Because every icon used, whether it has text underneath it, must symbolize its function. That's why we take care to use icons that every user can understand when making the app.

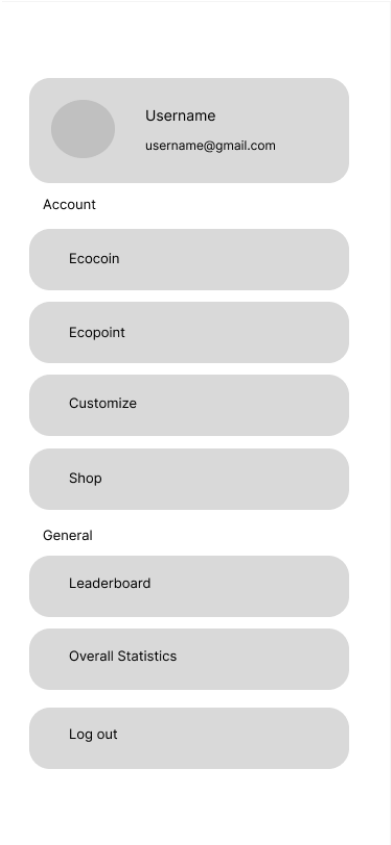
User Interface Components: In this app, we predominantly employed a minimalist and familiar UI design approach. The color palette leans towards designs that encourage recycling and environmental awareness, while the logos incorporate environmental elements. Additionally, we opted for the use of a navigation bar to facilitate easy access for users. Furthermore, when it comes to button designs, we generally favored soft-rounded edges over sharp corners. This decision aims to reduce eye strain and enhance user comfort.

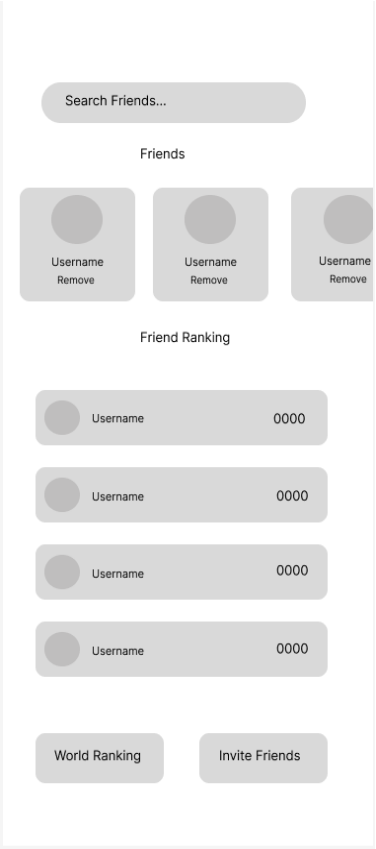






Layout and Grid System: We utilized Figma for the layout and grid system. Figma provided a specific grid system that is composed of larger squares rather than tiny ones. This grid system was designed to create a more balanced screen. Initially, we constructed wireframes using these grids. Subsequently, we applied our app's theme by adding colors to these wireframes. Here are some example wireframes:



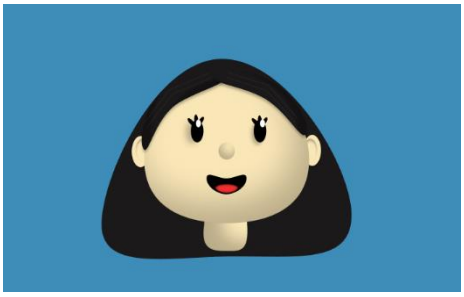


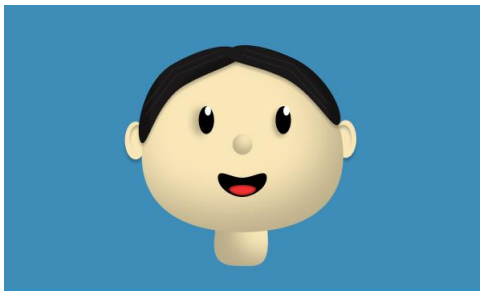
Gamification Elements: We provided each user with an avatar as part of the gamification elements. Users can customize these avatars using their Ecocoins, allowing them to create unique and personalized avatars. Additionally, users can explore different experiences by purchasing various accessories from the in-app shop. This approach enhances user engagement and enjoyment within the app while promoting sustainability and responsible recycling.

User Experience (UX) Guidelines: For our UX guidelines, we drew inspiration from apps that have been well-received by users and are known for their high usability. We found that, in general, simpler and easy-to-use apps tend to be more beloved. As a result, we followed the KISS (Keep It Simple) principle when designing our own app. This approach prioritizes simplicity, efficiency, and user-friendliness, ultimately contributing to a positive user experience.

Design assets: We created our logo using Figma. Our logo features hands holding the Earth and incorporates the recycling symbol. This symbolizes the need for collective efforts to create a sustainable world. As for our avatars, they are still a work in progress. We intend to use Blender for creating both avatar accessories and the avatars themselves. This approach will allow us to have full creative control over the avatars and their customization.

Examples:

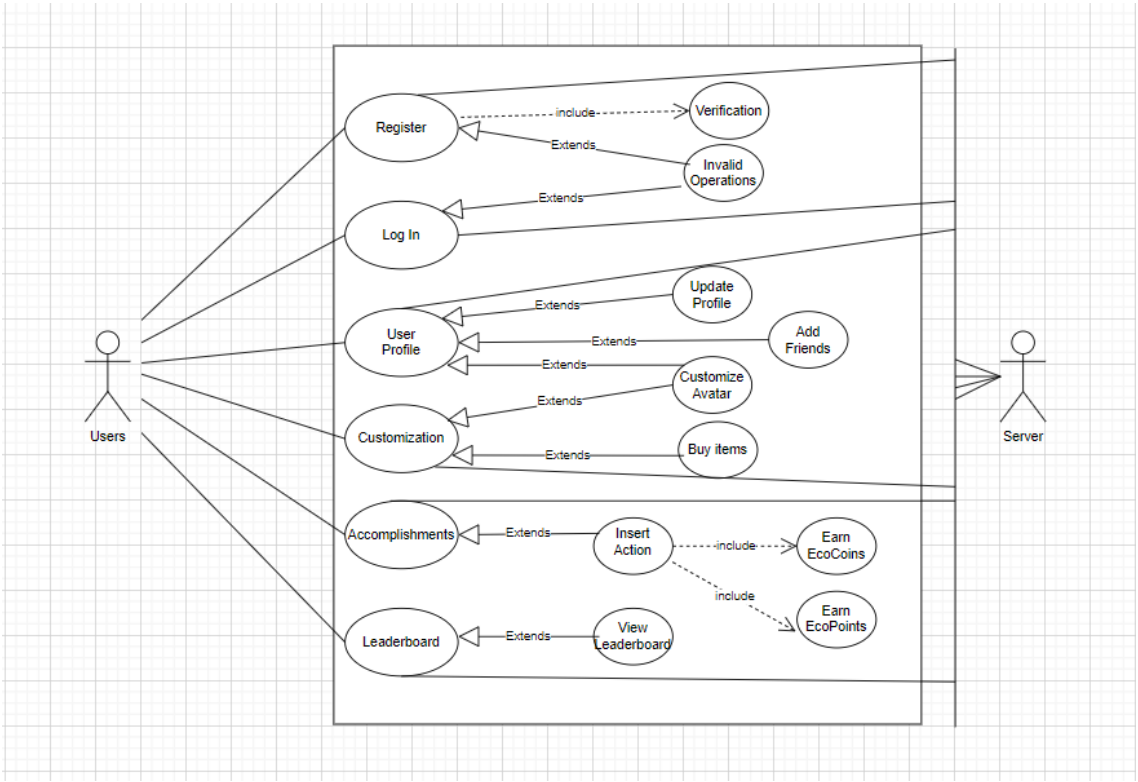


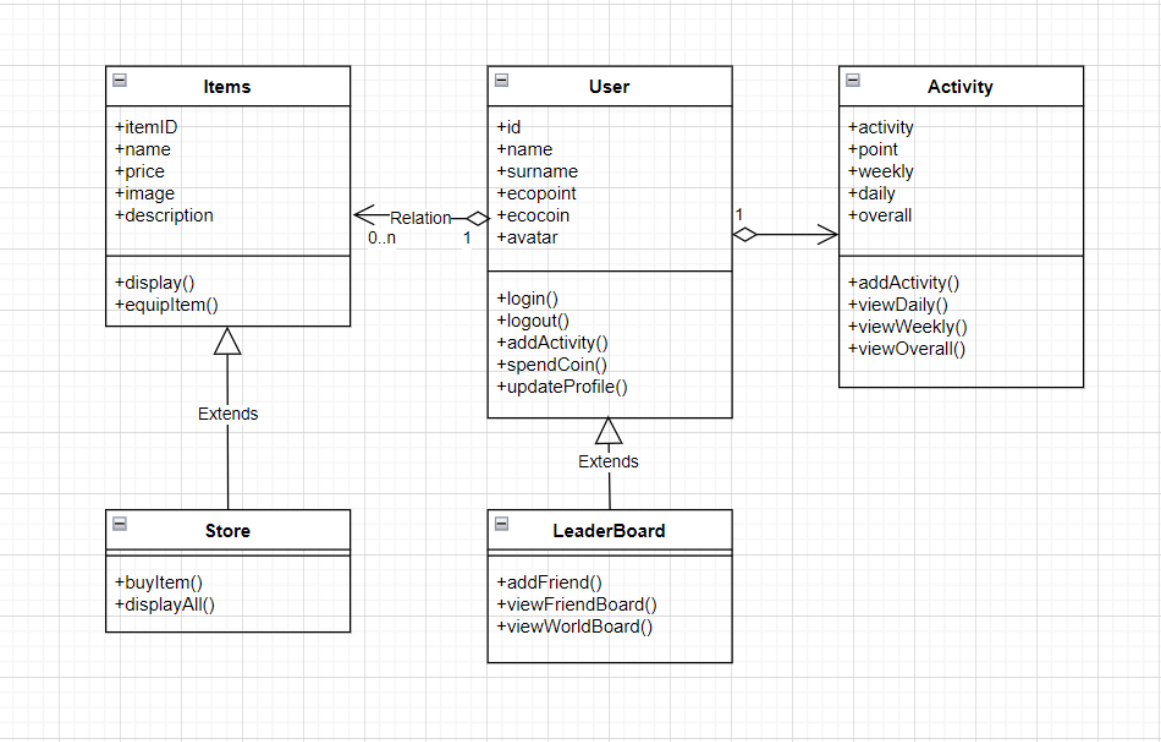


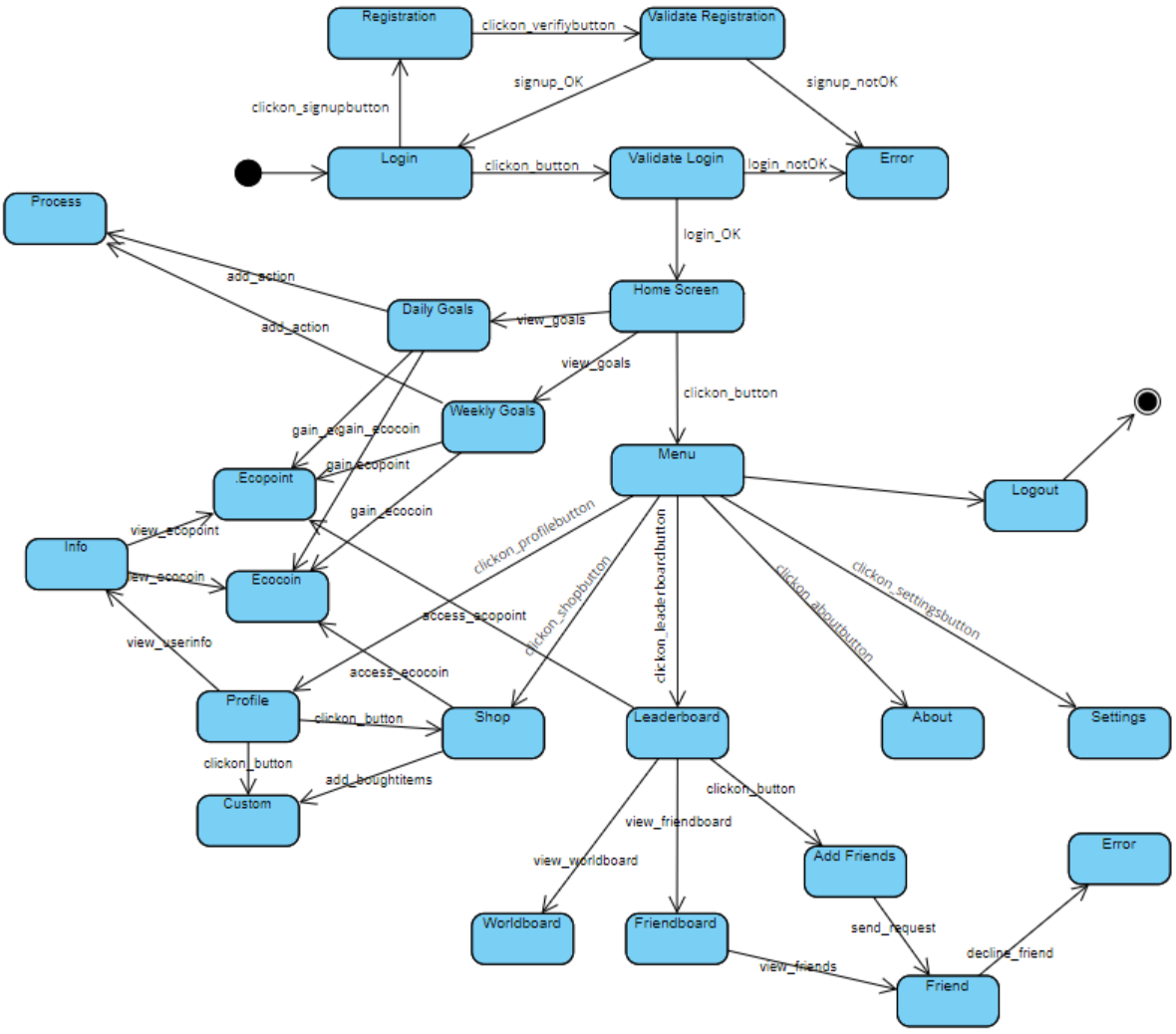
Design Principles: While designing our application via React Native, we followed some design principles.

- Component-Based Development: We divided our code into different components. So each component can make their task and helps us to maintain code more easily. Also, since Component-Based Development is more modular, it makes our code easy to update.
- Navigation and Roam: We used a React-Native Navigation library to roam different pages via buttons.
-

1.3 Engineering standards (e.g., UML and IEEE)




Items
+itemID
+name
+price
+image
+description
+display()
+equipItem()
Relation
0..n
1
User
+id
+name
+surname
+ecopoint
+ecocoin
+avatar
+login()
+logout()
+addActivity()
+spendCoin()
+updateProfile()
1
Activity
+activity
+point
+weekly
+daily
+overall
+addActivity()
+viewDaily()
+viewWeekly()
+viewOverall()
Extends
Extends
Store
+buyItem()
+displayAll()
LeaderBoard
+addFriend()
+viewFriendBoard()
+viewWorldBoard()

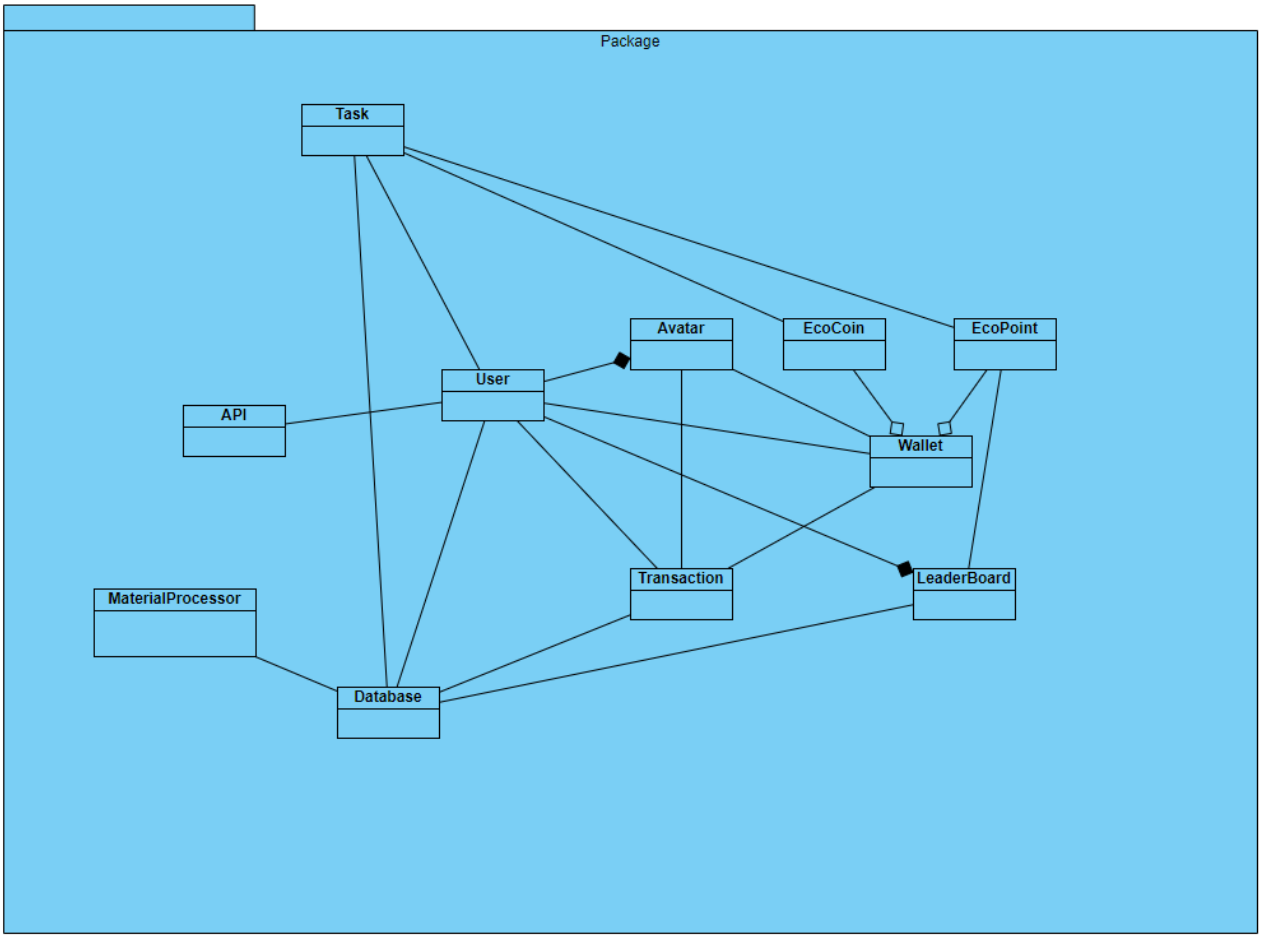

Registration
clickon_verifiybutton
Validate Registration
signup_OK
signup_notOK
clickon_signupbutton
Login
clickon_button
Validate Login
login_notOK
Error
Process
login_OK
add_action
Daily Goals
view_goals
Home Screen
add_action
view_goals
clickon_button
Weekly Goals
gain_again_ecocoin
gain_ecopoint
Menu
Logout
Ecopoint
gain_ecocoin
view_ecopoint
clickon_profilebutton
Info
view_ecocoin
Ecocoin
clickon_shopbutton
clickon_leaderboardbutton
clickon_aboutbutton
clickon_settingsbutton
access_ecopoint
view_userinfo
access_ecocoin
Profile
clickon_button
Shop
Leaderboard
About
Settings
clickon_button
add_boughtitems
Custom
clickon_button
view_friendboard
view_worldboard
Add Friends
Error
Worldboard
Friendboard
send_request
decline_friend
view_friends
Friend

1.4 Definitions, acronyms, and abbreviations

As a design pattern we used KISS (Keep It Simple) which is about simplicity. Also, the users should discriminate EcoPoints from EcoCoins. EcoPoints are for ranking and competing while EcoCoins are for customization.

2. Packages

2.1 Server



User: The User class represents the core user data of the application. It creates user accounts, handles logins, and manages task completion. User information, including nickname, email address, password, Ecocoins, and Ecopoints balances, is stored within this class. Users can perform actions such as logging in, completing tasks, customizing avatars, and purchasing accessories.

Wallet: The Wallet class manages users' financial information. It tracks users' balances of Ecocoins and Ecopoints and facilitates transactions. Users can deposit Ecocoins and Ecopoints using this wallet. Additionally, this wallet is used when purchasing or customizing avatar accessories.

Avatar: The Avatar class represents customizable avatars for users. Users can personalize their avatars by adding or removing accessories. Users can purchase accessories for their avatars using Ecocoins. These accessories are used for personalizing the avatar's appearance.

Transaction: The Transaction class represents users' financial transactions, including actions such as adding Ecocoins and Ecopoints. This class records and updates user balances by documenting financial transactions.

Database: The Database class serves as the source where application data is stored. It includes user accounts, tasks, leaderboards, accessories, and financial transactions. User data, tasks, leaderboards, and other application data are stored and accessed within this class.

API: The API class is used for communication with the server. It retrieves and updates user data, tasks, leaderboards, and accessories from the server. The API ensures the synchronization of application data with the server.

Task: The Task class represents tasks that users need to complete. Users earn Ecocoins and Ecopoints by completing these tasks. Tasks help users track their progress and achievements.

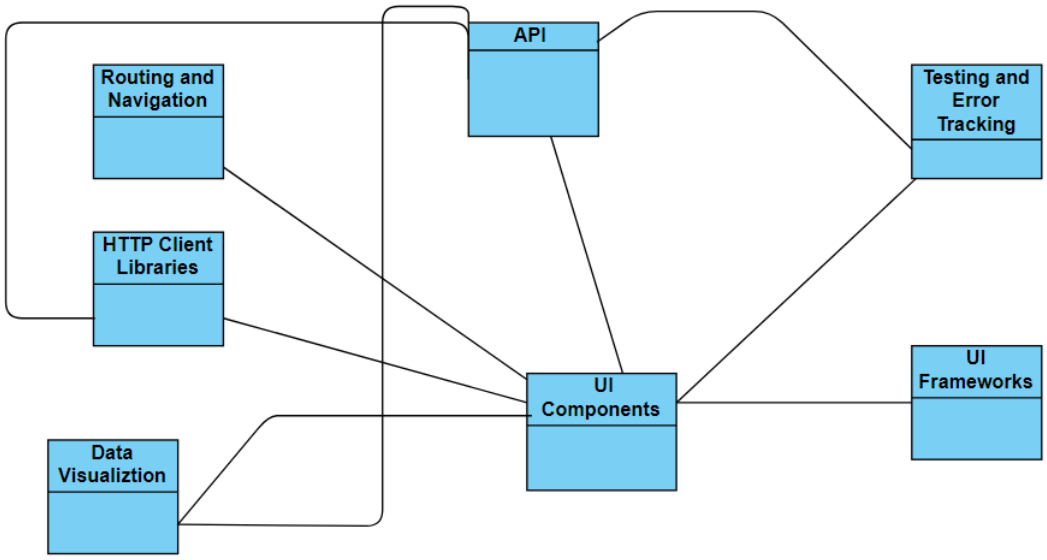
LeaderBoard: The Leaderboard class allows users to track their rankings among friends and in the overall leaderboard. Users can view rankings and compete with others.

MaterialProcessor: The MaterialProcessor class assists users in identifying materials and calculating conversion rates. It enables the addition and transformation of materials through image processing.

EcoCoin: The Ecocoin class represents the digital currency Ecocoins used by users. Users use this currency to earn rewards and purchase accessories. Specifically, the Ecocoins balance is part of a wallet.

EcoPoint: The Ecopoint class represents a scoring system used to measure users' eco-friendly behaviors. Ecopoints are used to encourage sustainable behavior.

2.2 Controller



UI Components: This is a package that contains the basic components of the user. This package includes buttons, forms, menus, graphics and other visual content.

HTTP Client Libraries: This package provides communication with the server. Sending requests and receiving responses is possible via HTTP. It is especially important when calling the API and exchanging transactions with the server.

Routing and Navigation: It allows users to navigate between different pages. We also used this system, which is used in multi-page applications, when switching between different pages.

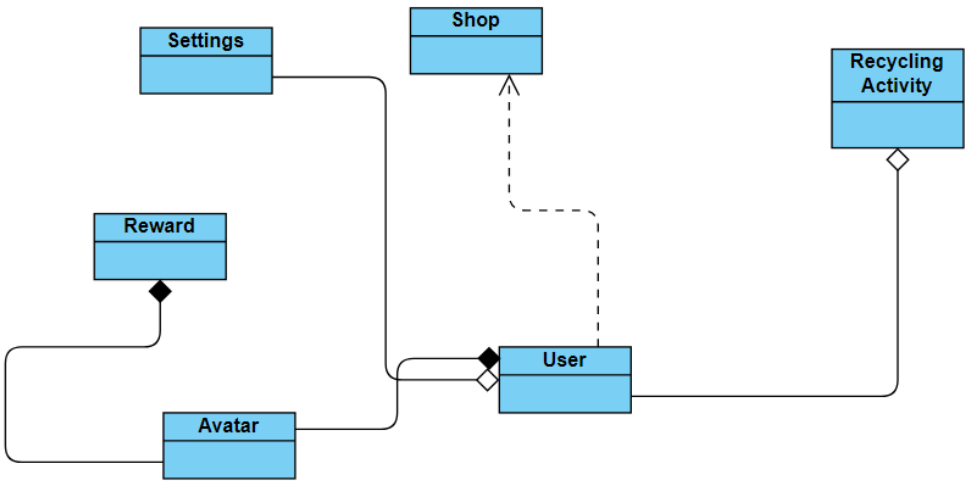
API: The client library we use to retrieve data from the server. Thanks to this library, communication with the API running on the server side becomes easier.

UI Frameworks: enable the user interface to be organized in a larger structure.

Testing and Error Tracking: A package used to test the smooth operation of the user interface and track errors.

Data Visualization: Package used for graphing and visualizing data.

2.3 Client



User: The package where users are registered, logged in, logged out and managed.

Recycling Activity: Package in which recycling activities are recorded and organized.

Reward: The package where the money collected by users is kept and the rewards are calculated and managed.

Avatar: A package that allows users to customize their avatars and process those customizations.

Settings: Package where users manage basic application preferences.

Shop: A package where users customize their avatars, list the items they can use, and guide sales transactions.

3. Class Interfaces

In our application, we provide our users with several classes with interfaces that can be seen by our users as pages. In order to make our application more user-friendly, easy to use, and more effective, we make sure to follow some principles. While designing our application we followed and applied KISS design principles which is about keeping the design as simple as possible. We're still trying to finish our application and still following the KISS design principle. While some teammates are dealing with image processing and the backend, also some of the UIs are coded. Since there are not much backend code we cannot add pseudocode however, we can share our UI codes if it's required.

We divided our application into 4 main components which have unique functions inside them.

3.1 Component 1: Welcome Page, Signup and Login

3.1.1 Purpose and Functions

We have opted to use a single screen for both the login and sign-up processes which is Welcome Page. On this screen, we've chosen to create a user-friendly, simple interface that prominently features our application's logo. We decided to select a background that complements the color tones of our application and pair it with a minimalistic font for a sleek look. Along with our slogan, we have highlighted the functionality of our application. Users can log in using the "Join us!" button and sign up by clicking the "Login" button.

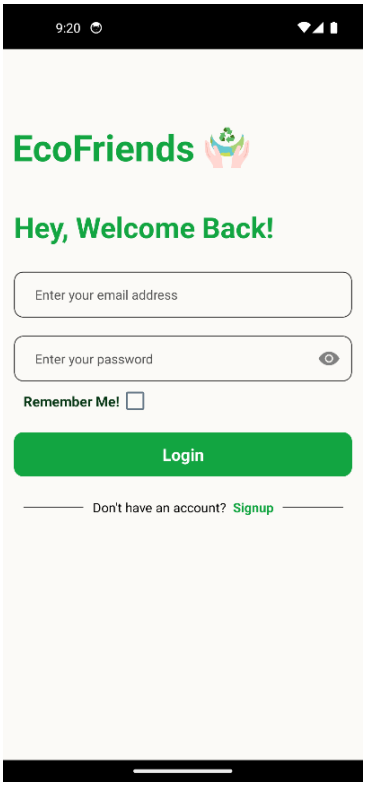
The purpose of the Signup Screen is to enable both new and existing users to register and gain access to the app. We aimed to provide users with a familiar and user-friendly UI design, much like the login screen. On this registration screen, users are required to enter their email, password, and confirm the password. After filling in this information, they can agree to the terms and conditions and proceed to create their account.

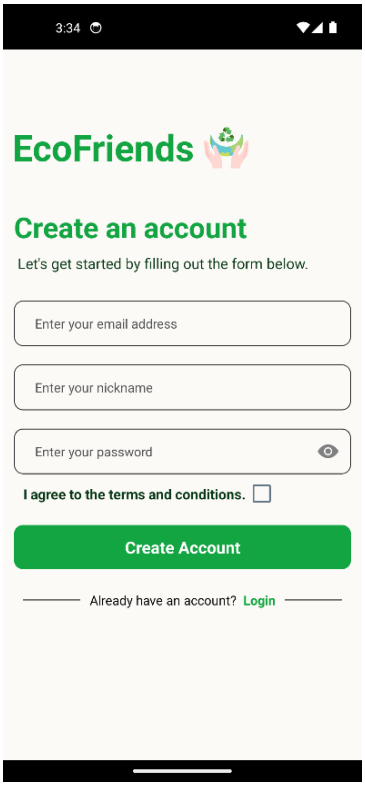
On Login Screen, our goal was to provide the user with a minimalist and straightforward login interface. We aimed to differentiate our app with its name and logo, and we added a touch of sustainability to the design. This way, users would encounter a familiar layout while also sensing our application's unique identity. In this interface, the user is required to enter their email address first and then their password to access the application. We've also included an option to show the password for ensuring the correctness of what is being typed. At the same time they have to accept our policy with remember me! button. Furthermore, when the user presses the "Login" button, they will be able to access the application. Simultaneously, if the user doesn't have an account, they can sign up.

3.1.2 Interface

Welcome page does not contain any information from users. It is made for to lead users to login or signup page. In our signup and login page we wanted some critical information from users such as their e-mail addresses, nicknames, and passwords. Nicknames do not have to be unique since the app will give every user a unique tag. After the backend part is finished by team members we can return the values and the users can operate these functions successfully. After that, the app will lead them to the Home Screen.

You can see our coded UIs below.





3.1.3 Dependencies and Relationships

In Welcome Page, we have 2 relationships. When we first open the app we get the welcome screen then if we push the Join Us! Button we directly go to account creation screen. The next relationship will be login. If the user has already an account they can directly go to the account by entering email and password.

In Signup Page, as a relationship if the user enters their password and email and press the login button, they will see home screen of our app. Also if they press the signup button they will go to the sign up screen.

The purpose of the Login Page is to enable both new and existing users to register and gain access to the app. We aimed to provide users with a familiar and user-friendly UI design, much like the login screen. On this registration screen, users are required to enter their email, password, and confirm the password. After filling in this information, they can agree to the terms and conditions and proceed to create their account.

3.2 Component 2: Home Screen

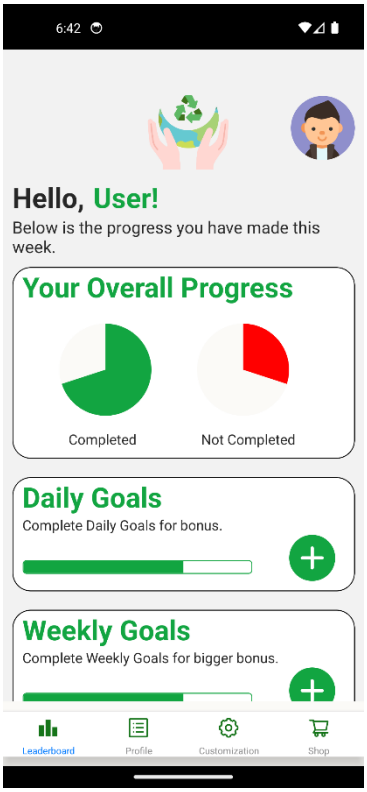
3.2.1 Purpose and Functionality

On the home screen, the user can view their tasks and track their daily progress through charts. Additionally, if they wish to add new items, they can enter these items and increase their Ecocoins and Ecopoints. Furthermore, users can navigate between pages using the navigation bar.

3.2.2 Interface

Our Home Screen will show the users their daily activity and daily and weekly goals. The home screen will also have a bottom nav bar which makes it easier to reach some functions. Also, by clicking the avatar, the users will be directed to the user profile. We used some progress bars to make user's progress more visible to them which will be a inspiration for them to continue.

Our Home Screen code has not finished yet but you can see the progress below.



3.2.3 Dependencies and Relationships

In Daily goals, user can Access their Daily goals. Also with + icon they can add more materials. It's the sam efor weekly goals. In navigation bar they can Access leaderboard screen, profile, customization and shop screens.

3.3 Component 3: Profile and Customization

3.3.1 Purpose and Functionality

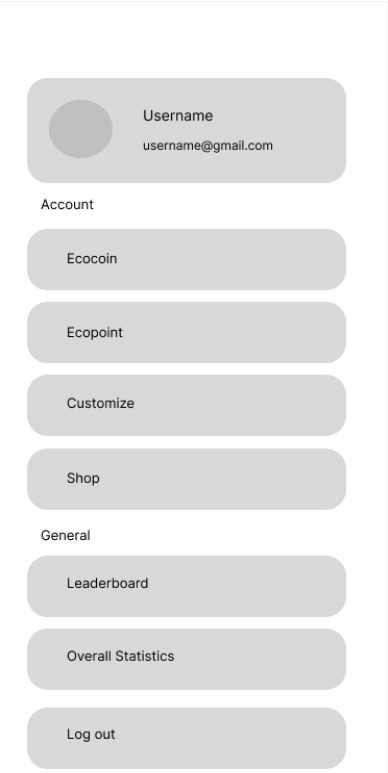
In Profile Page, our aim was to create a general profile menu. Here, users can view their nickname, Ecocoins, and Ecopoints, allowing them to keep track of their possessions. Users can access this screen to see what they have and how much of it. In addition to these features, users can also customize their avatars using Ecocoins and view their rankings among friends. If they want to see their overall recycling statistics, such as the amount of each item they have recycled, they can do so by clicking the "Overall Statistics" button.

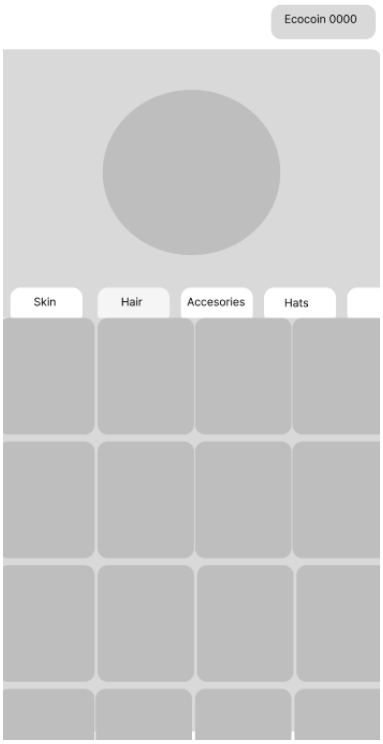
When the user clicks on the "Shop" section, they will encounter their avatars. They can dress up their avatars as they wish using their Ecocoins, creating different combinations. This customization feature allows them to intensify the competition among friends and create unique characters.

3.3.2 Interface

Our profile page will be the root of the applications since it contains the main functions inside it. For example, leaderboard, customization, and shop. The options will be visualized by icons to make them easy to follow. From the profile, the user can see their nickname, tag, and profile. From below, the users can also view their points and coins. Coding the UI will be the easiest part however, the navigations and functions are the most important part.

Since the Home Page has been coded recently, we cannot add our coded UI Design for the profile page but we can show our profile page from wireframe. (It's just a design without code, after the coding the UI will be completed, and we will be inserting our coded pages in the next reports.)





3.3.3 Dependencies and Relationships

When you click the "Customize" button, it displays the items you own and can dress up your avatar with. By clicking the "Shop" section, it shows you the items available for purchase using your Ecocoins. The "Leaderboard" section allows us to add friends, view our rankings among friends, and see our global position within the application. You can access this screen by clicking the "Leaderboard" button. Also when you push overall statistics you can see your statistics. Log out button provides quiting your account.

3.4 Component 4: Leaderboard

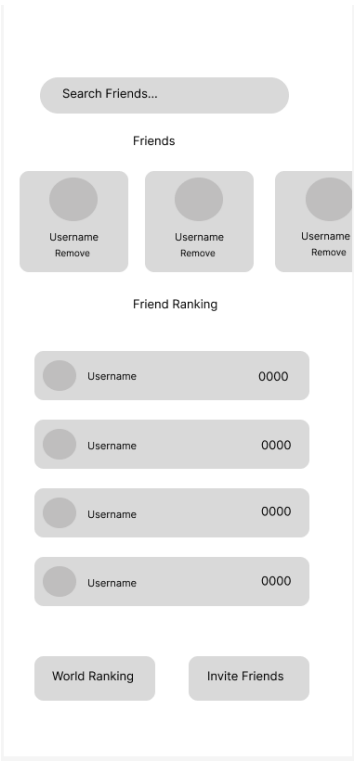
3.4.1 Purpose and Functions

On this screen, you can view friends who have the application, along with their Ecopoints and avatars. You can also see your ranking among your friends and your worldwide ranking. Additionally, you can check your global position as a user.

3.4.2 Interface

Our simple leaderboard page contains the users with their avatars and their points. You can add friends from the leaderboard page. You can view both friends' ranking and worldwide ranking from here.

Since the Home Page has been coded recently, we cannot add our coded UI Design for the leaderboard page but we can show our page from wireframe. (It's just a design without code, after the coding the UI will be completed, and we will be inserting our coded pages in the next reports.)



3.4.3 Dependencies and Relationships

When user enters the nickname of a friend they can send a request or they can see the friends' profile. World ranking also shows the worldwide ranking in the app. Invite friends button provides a link that you can send your friend to invite them.

4. Glossary

There are some terms that the customers should know while reading this report.

React-Native: It is a mobile application development tool which is based on JavaScript and React.js. It allows application development on different mobile platforms such as iOS and Android.

Navigation: Navigation is the action of roaming between pages. To make is possible we downloaded and imported a packet named "react-native navigation".

EcoCoin: Allows users to buy items.

EcoPoint: Allows users to compete with friends and others.

5. References

- Our previous reports from CMPE 491 Course
- https://www.visual-paradigm.com
- https://reactnative.dev