CS319

Final Report

BILSYNC

Team 8

Section 1

Tuna Saygın - 22102566
Hüseyin Burhan Tabak - 22102516
Ahmet Tarık Uçur - 22102946
Işıl Özgü - 22102276
Kanan Zeynalov - 22101007

Work Allocations **3**
Tuna Saygın - 22102566 3
Işıl Özgü - 22102276 3
Kanan Zeynalov - 22101007 3
Hüseyin Burhan Tabak - 22102516 3
Ahmet Tarık Uçur - 22102946 3
User Manual **4**
Build Instructions **16**
Database 16
Sources 16
Frontend 16
Usage 16

Work Allocations

Tuna Saygın - 22102566

- Allocated work for each team member and led the team.
- Implemented authentication class
- Implemented token and mail services
- Integrated of frontend to backend
- Created class and sequence diagram
- Created the frontend design in Figma
- Managed other subsystems' implementation and integration
- Created subsystem decomposition diagram

Işıl Özgü - 22102276

- Implementation and integration of transaction subsystem
- Implementation of admin control endpoints and their business logic
- Implementation of Report management and its business logic
- Creating the chat subsystem's primitive version
- Creation of Activity Diagram
- Reviewing the State Diagram
- Management of system functionality and Use Case Diagram
- Management of the layout and content of reports, creation of final demo video

Kanan Zeynalov - 22101007

- Creating frontend components in the class diagram
- Creating class and sequence diagram
- Integration of frontend to backend
- Implemented all the Figma design to React.js
- Frontend bugfix jobs
- Transaction system implementation for frontend
- Worked on quality improvements of frontend
- Post, User, Chat search algorithms and implementation of them in frontend

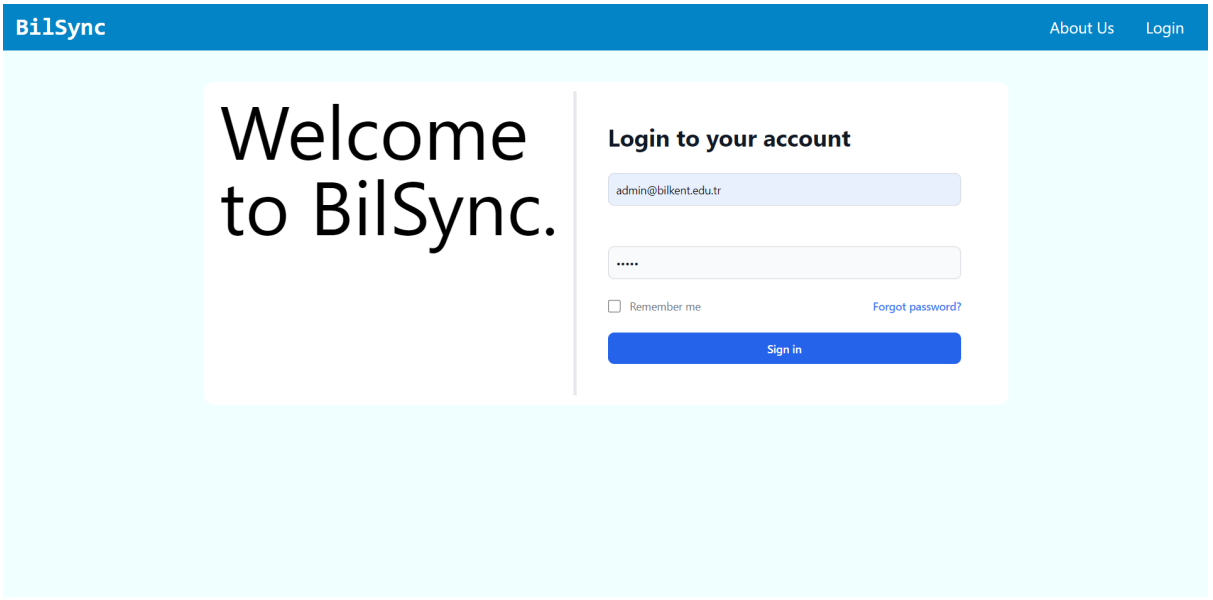
Hüseyin Burhan Tabak - 22102516

- Implemented the chat subsystem's final version
- Postgres database deployment & docker file creation
- Implemented use case diagram and its relations with other diagrams
- Created an image system for chat and chat messages
- Code refactoring and improvement for backend team
- Vote system and bidirectional relationship implementation for posts
- Release creation and deployment manual
- Debugged requests and endpoints for image retrieval systems

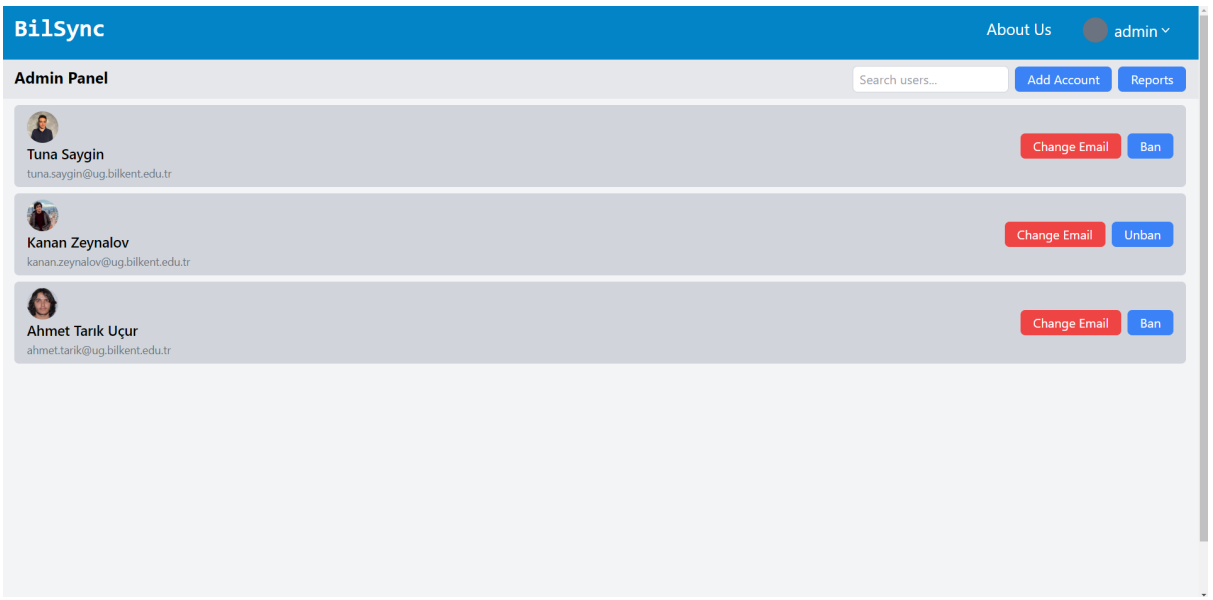
Ahmet Tarık Uçur - 22102946

- Implemented post and its derivatives' subsystem
- Implemented comment and vote subsystem for backend
- Report feature implementation
- File system and file retrieval feature from frontend to backend
- Creation and management of State Diagrams
- Added comments and special exceptions for backend functions
- Bug testing and edge case detection for frontend and backend team
- Inheritance in database tables and facade pattern application for post service

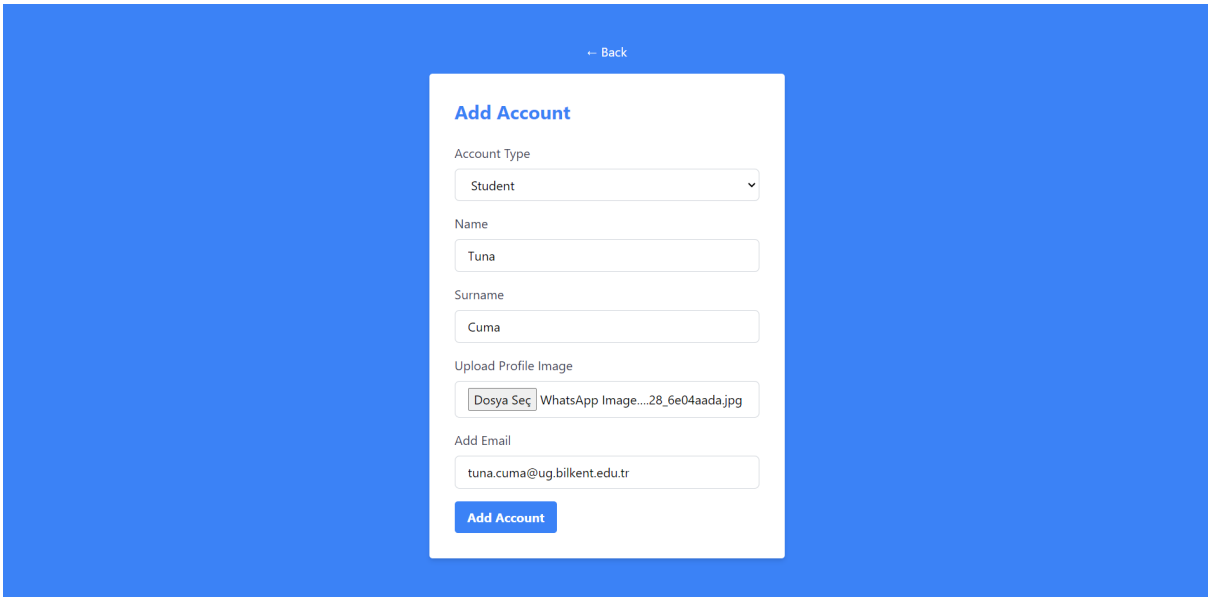
User Manual



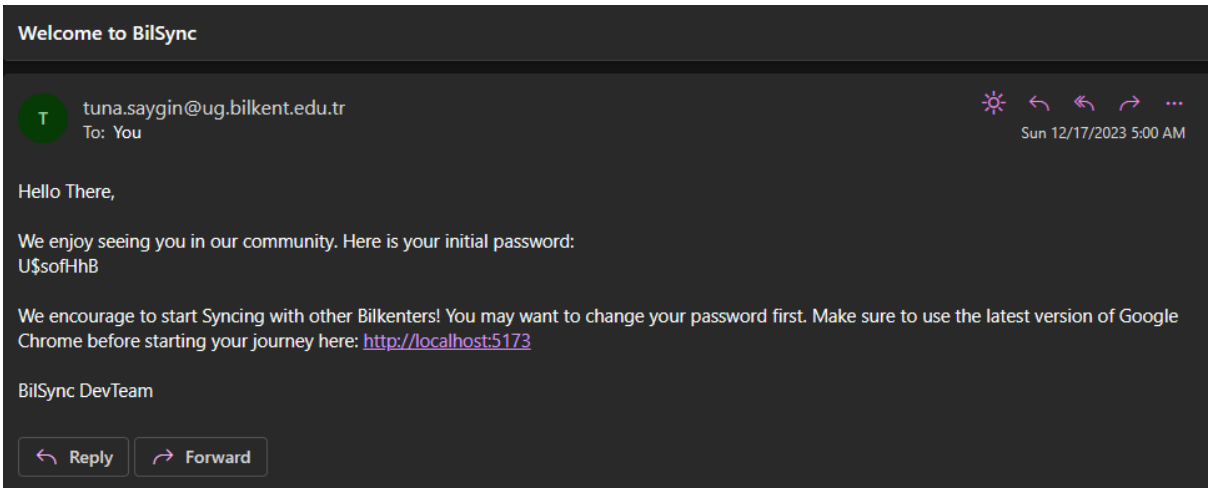
Since BilSync provides a “gated” community for Bilkenters, it is not possible for the users to create an account. Whenever a new student, instructor, or staff joins Bilent University, the admin needs to create an account for them. But first, the admin should log in from the main page with his account in which the admin email is accepted as **admin@bilkent.edu.tr**, and their respective password is **admin**. (Make sure to sign out from the admin account using the top right panel before you start testing your regular user account. Also, do not forget to check the registered email account’s inbox to retrieve your randomly generated password).



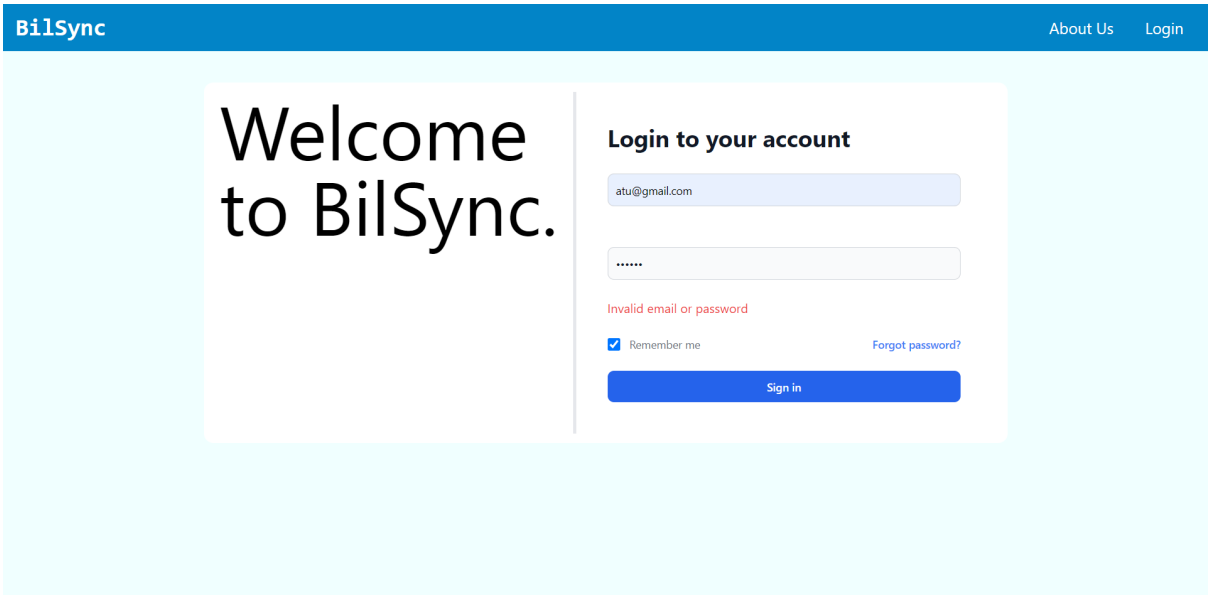
After logging in with the admin information, the admin will be sent to the admin panel where he is allowed to view reports, ban or edit existing user accounts, and add accounts.



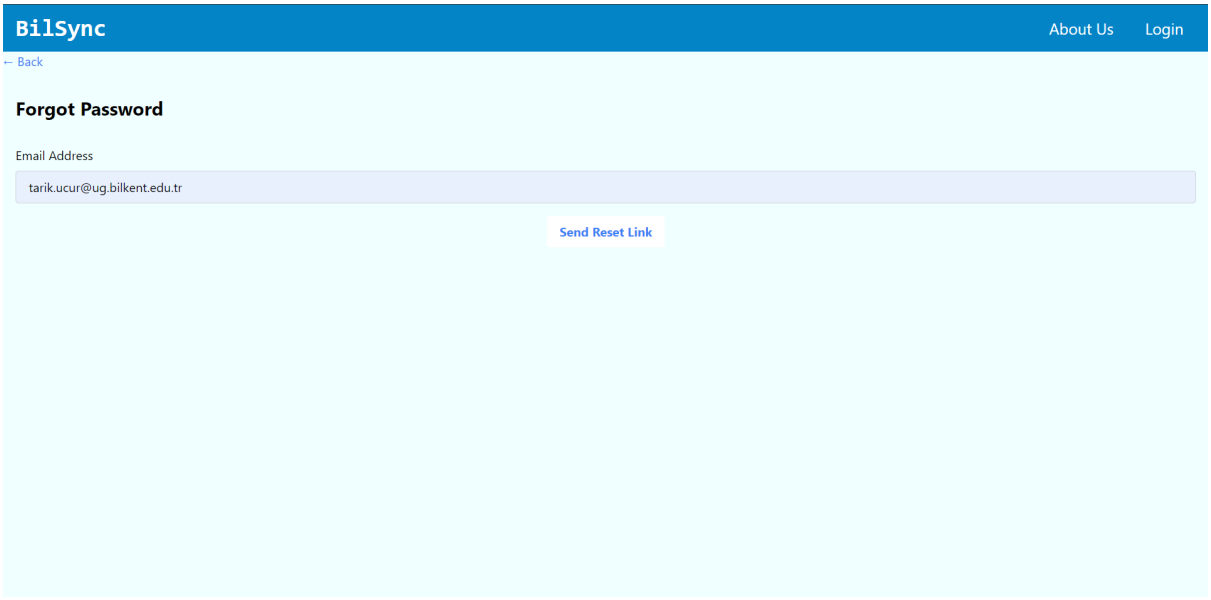
For the add account feature, the admin can specify the account type (user, instructor, staff, alumni), the email of the user, and their name. The password will be randomly generated and sent to the user's email to be registered. Also, changing the users' emails is only allowed for the admin since the application will be specific to Bilkenters only.



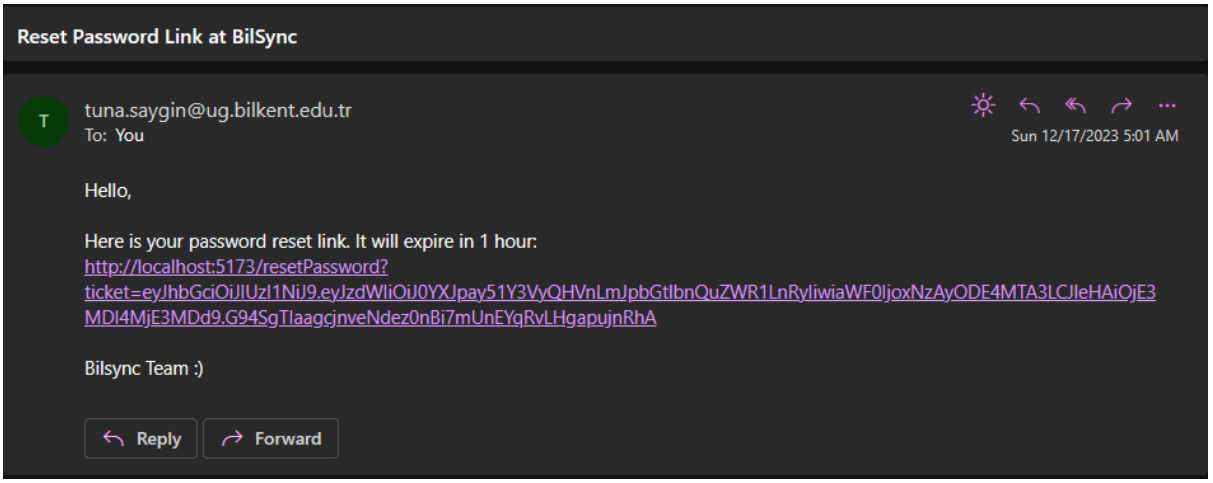
Such new users will get an email from the system containing the user’s randomly generated password, a link to the BilSync website, and a warm welcome message for the new members.



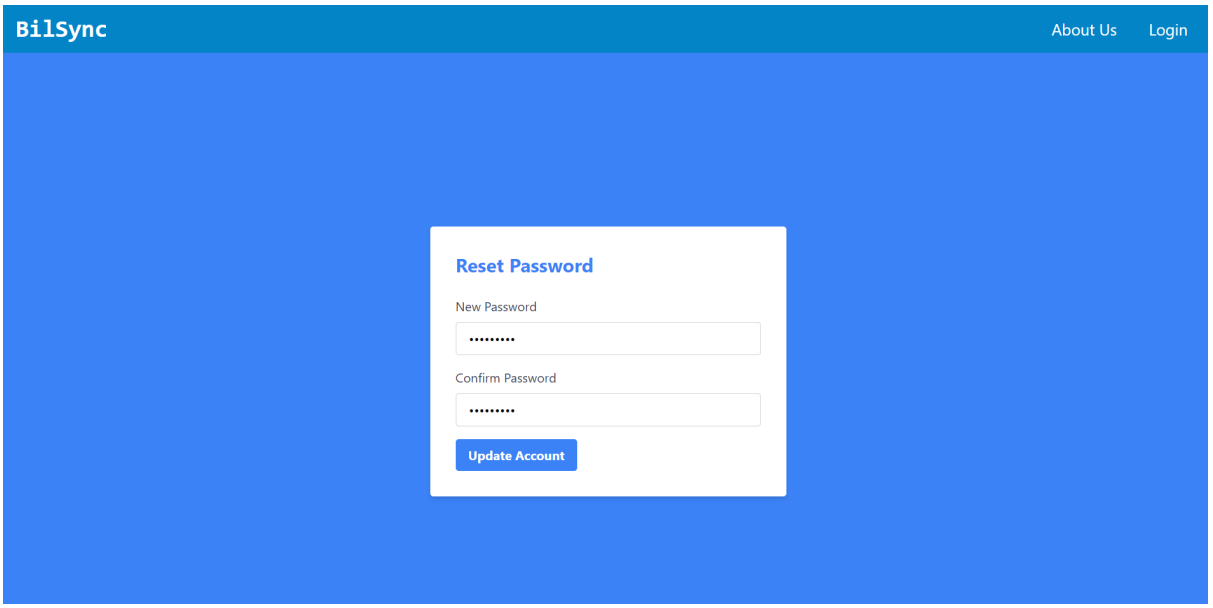
Using the link, the user will be redirected to the login page. Here, the user will be able to log in to their account, either by using their predefined password or by using the “forgot password” button below.



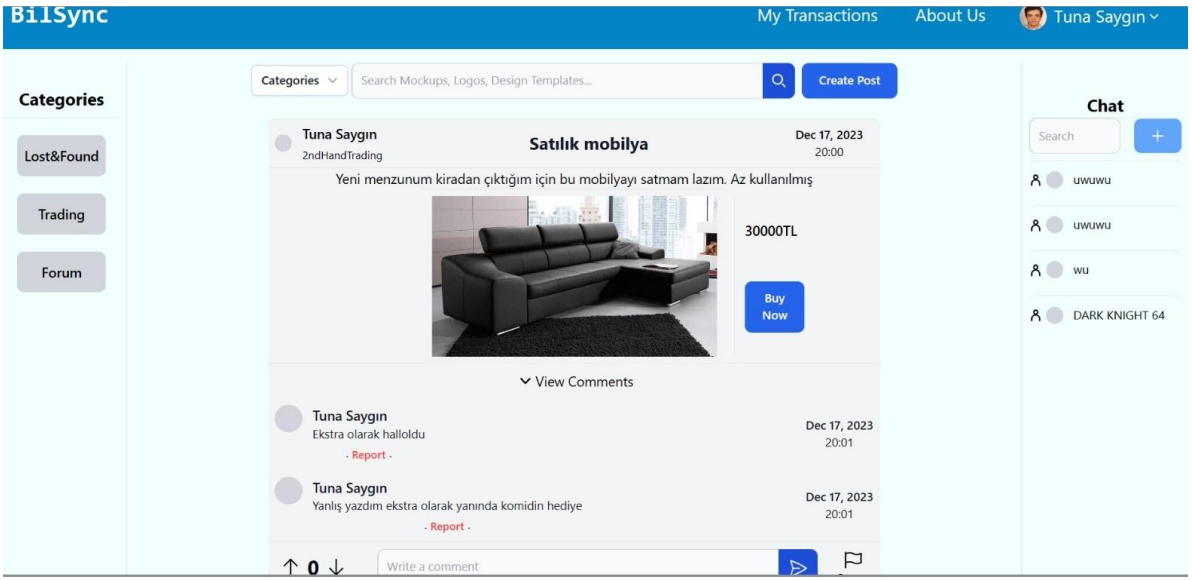
After clicking the “forgot password” button and entering an email address of their own, a confirmation email will be sent to the user.



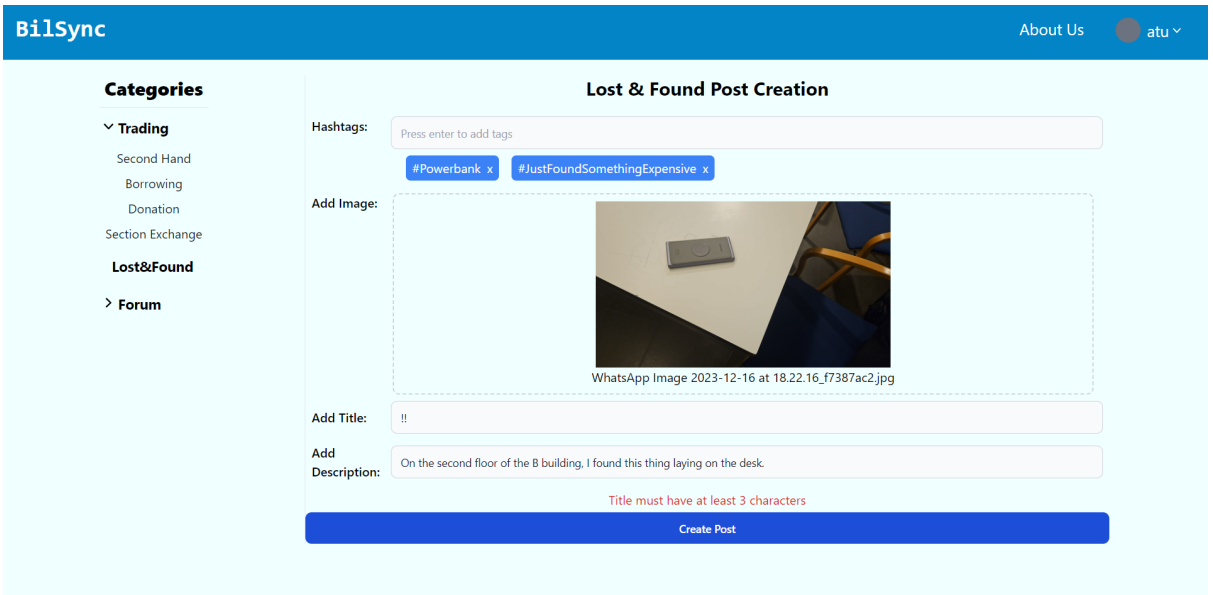
The email includes a user-specific password reset link, which will expire after a 1-hour period. Upon clicking the reset password link, the user will be redirected to the “reset password” page.

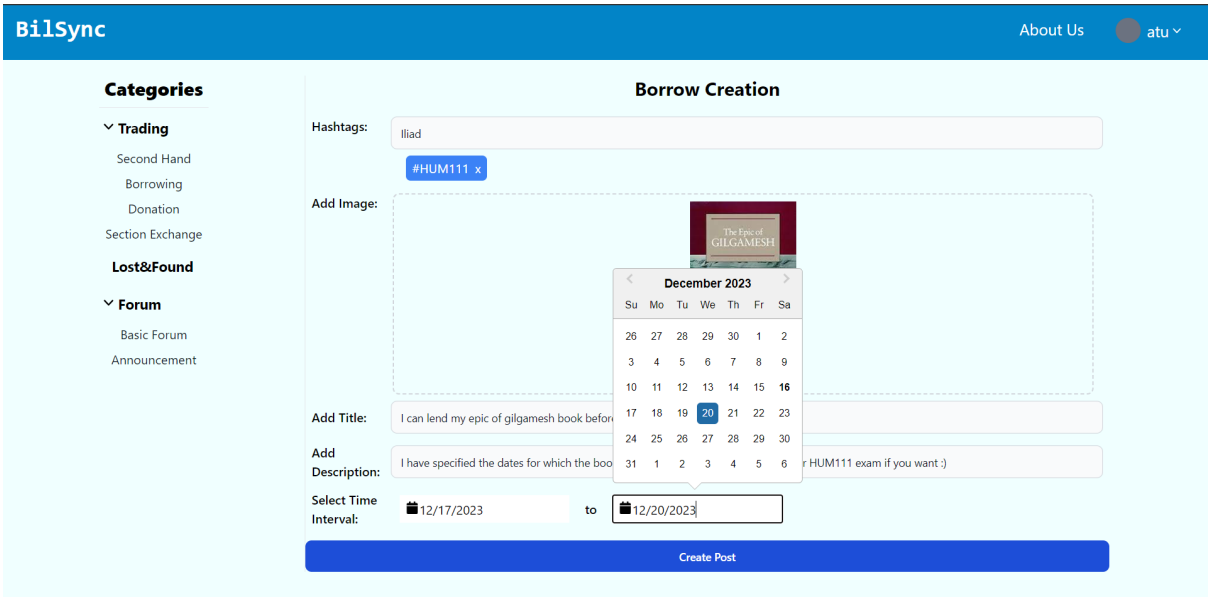


In the reset password page, when the new password is entered twice in both fields and when they match, the user's password will be changed, and they will be redirected to the login page.

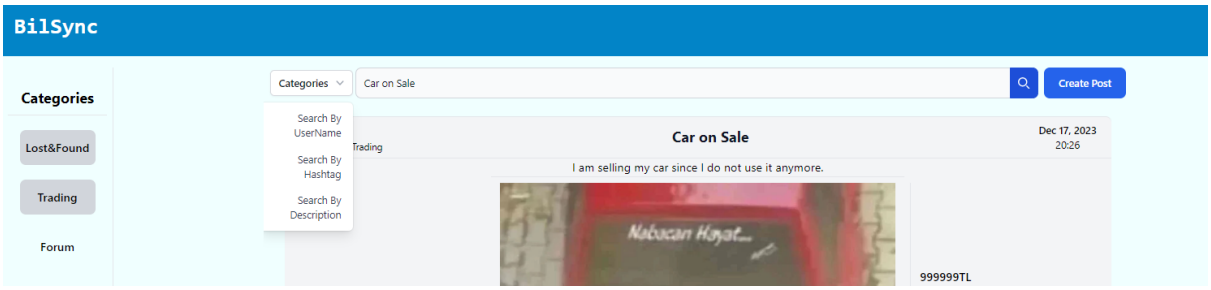


After logging in with the correct information, the user will be redirected to the main page. Here on the left panel, the user can sort each post by choosing one or more categories. On the right panel, the user can create a chat with someone else or with multiple people. Also, they can open a chat panel by clicking on the channel. On the top panel, the user can create a post or search multiple posts by their hashtags, post author's username, or description. Also, the user can create a post near the search bar. And finally, in the middle panel, the user can interact with other posts, such as up/downvoting, commenting, reporting, buying, borrowing…

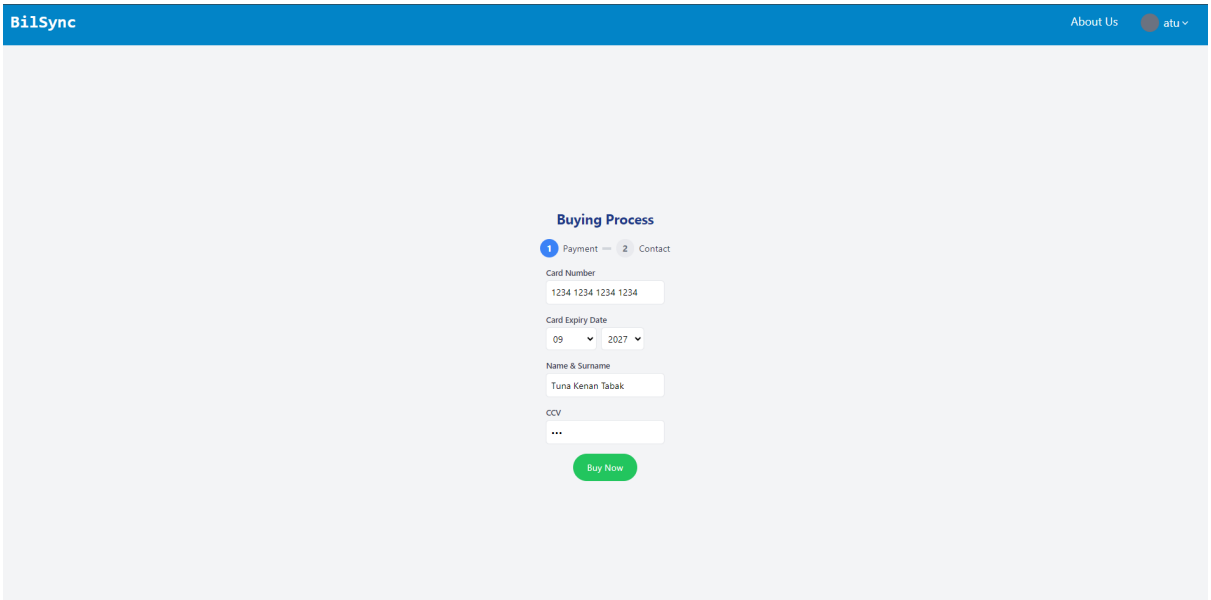




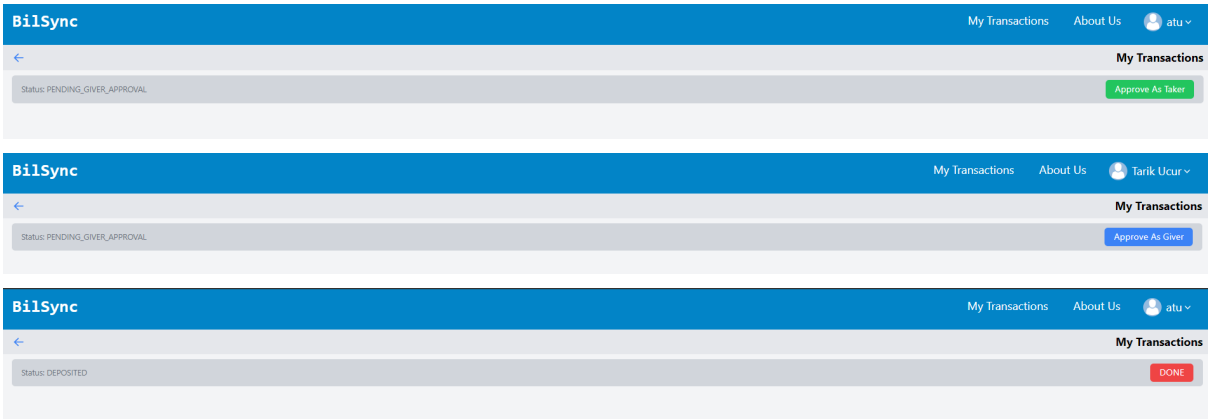
In the post creation page, the user can choose any post type from the left panel. Different post types may have different fields to create the post. The posts will be created when the user fills out the required fields. Later on, the user will be sent back to the main page. The main page will be updated according to the user's and other users' changes.



In the search bar, the user can search from all the posts based on his specific requirements and different criteria. Also, the search bar can be combined with the “categories” panel’s choosing.

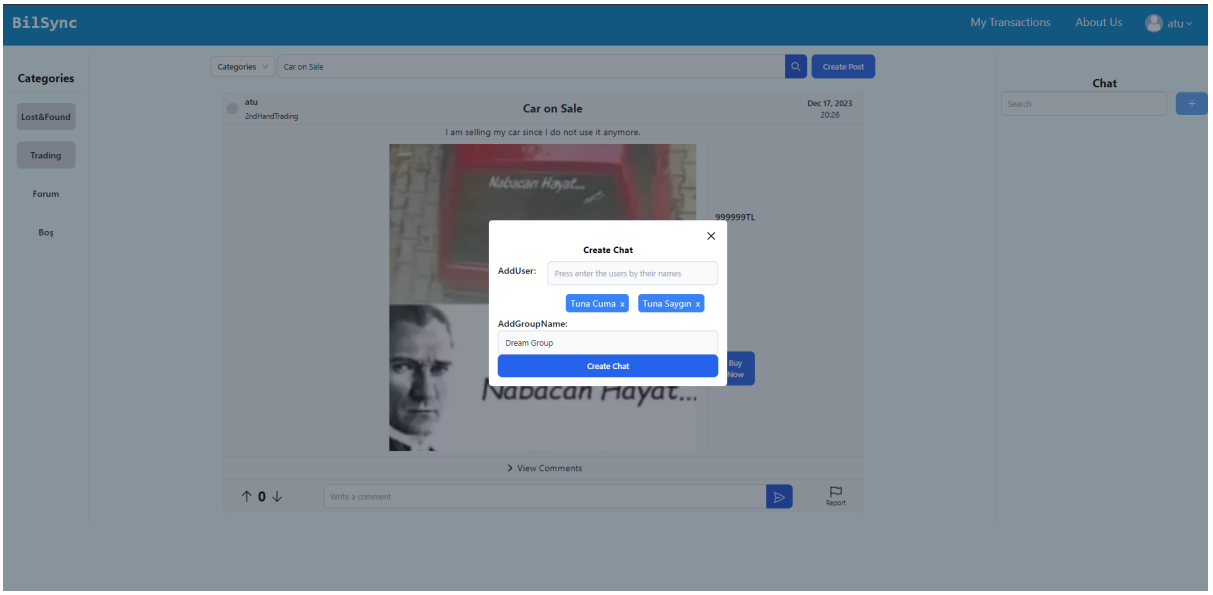


Some post types have a buying process. After you decide to buy something, you will be shown a “buying process” page. When you enter your payment information, the money will be taken from the user’s card. After that, the user will be sent to the contact chat with the seller.

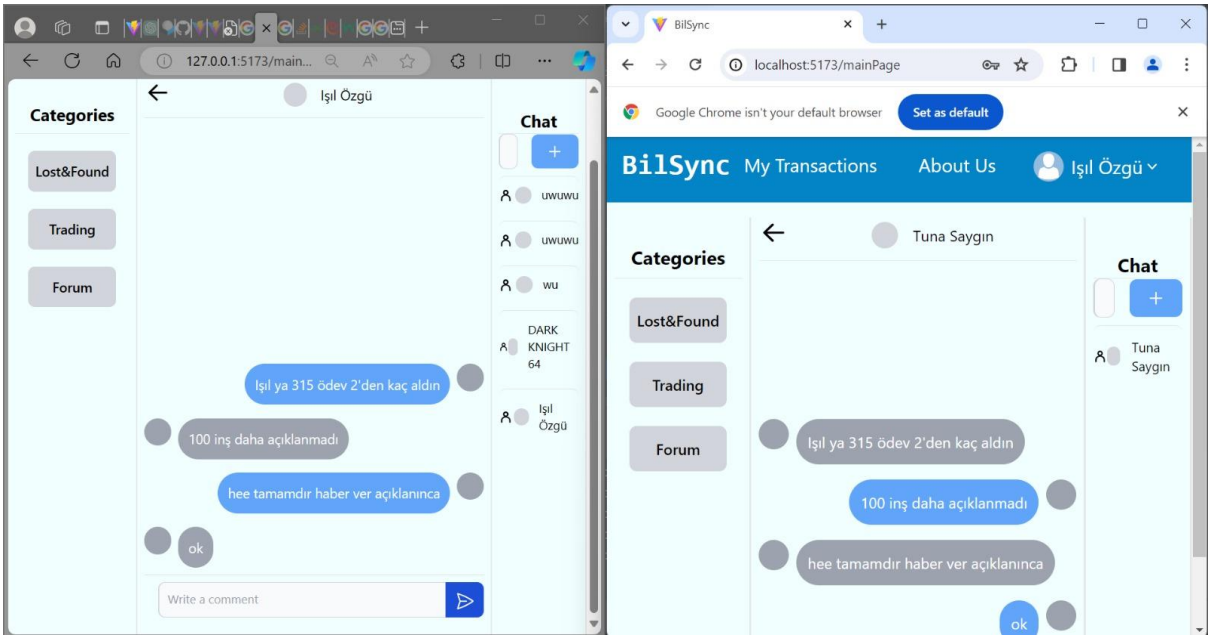


After the seller delivers the item to our user, the giver will approve the selling process. After that, the user can also approve that they have received the listed item. If a side does not approve the transaction, the money will be refunded to the user. If both of them agree in 3 days, the money will be sent to the seller via IBAN. This way, we ensure that there won’t be any scam attempts hence the user can receive his money back after a failed attempt to purchase.

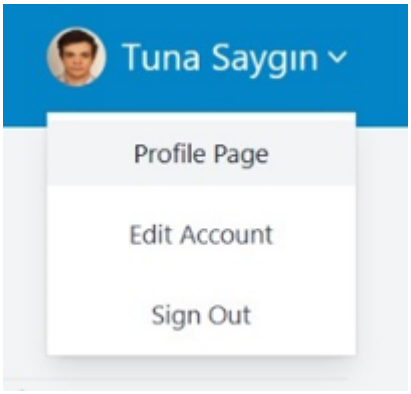
After the purchase is completed, the post will be shown as sold. This way, no other people will attempt to purchase it, and know that the item they were going to buy is already sold.



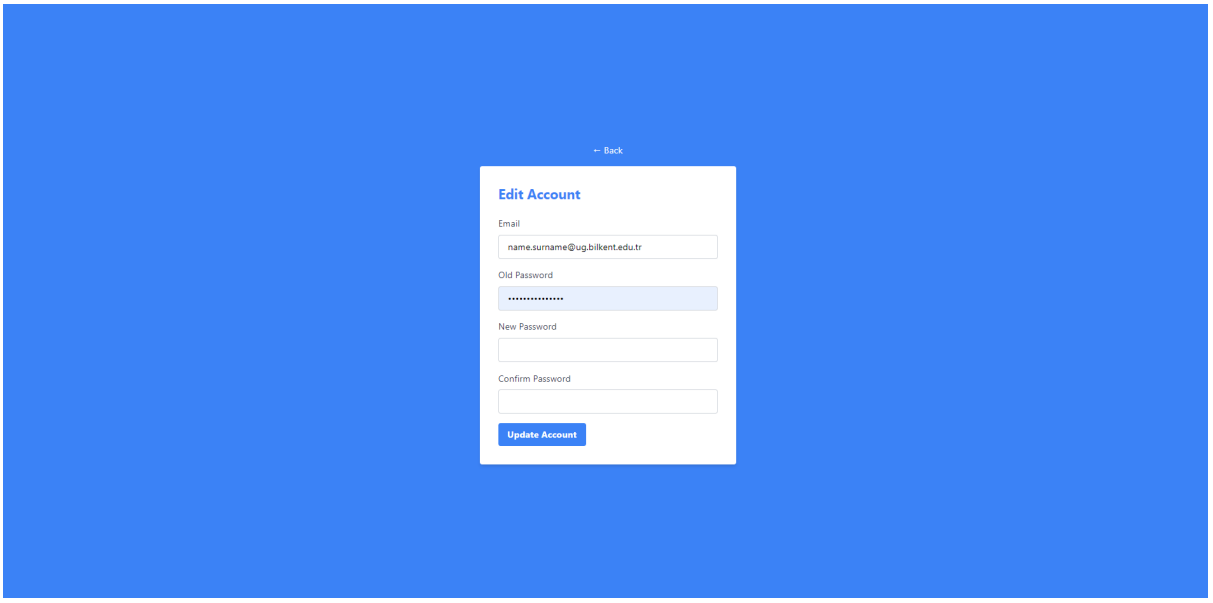
Other than buying and borrowing processes, the user can also create a chat, either as a private or group chat. If only one member is specified, the chat will be a private chat. If more than one member is provided, it will become a group chat. To enter other users to the chat, you need to type their name and press “enter”. If it is determined that the chat will be a group chat, you need to specify a group chat name.



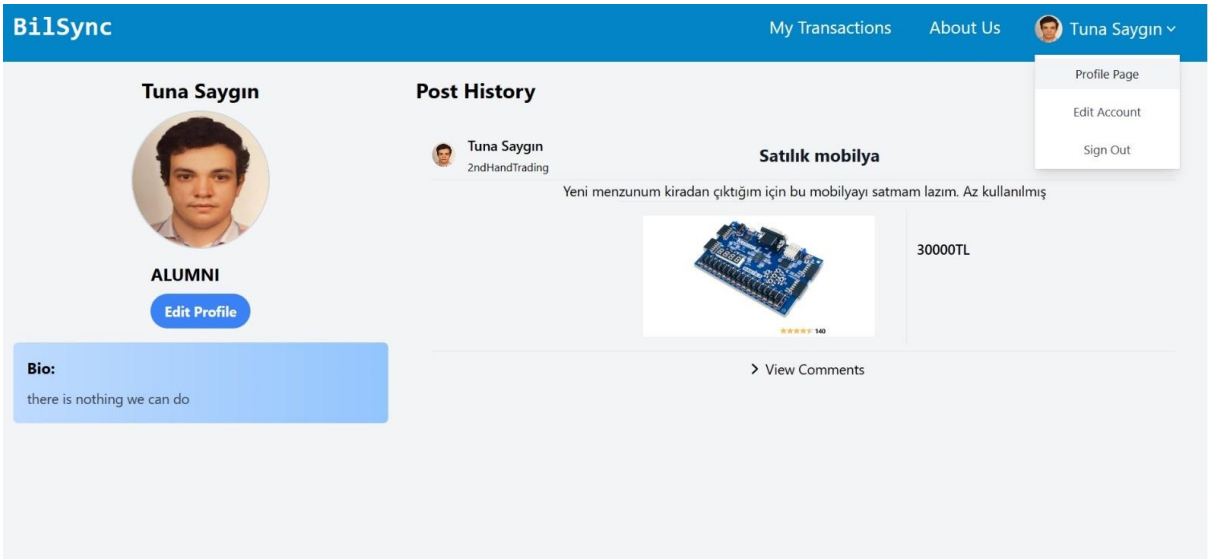
For chat, both users can send messages and receive message updates immediately. This way, we ensure a handy chatting feature for our application. For private chats, both users will see the other user’s name on the right panel in the chat field, while for group chats, they both will see the group chat name which is a specifically given name.



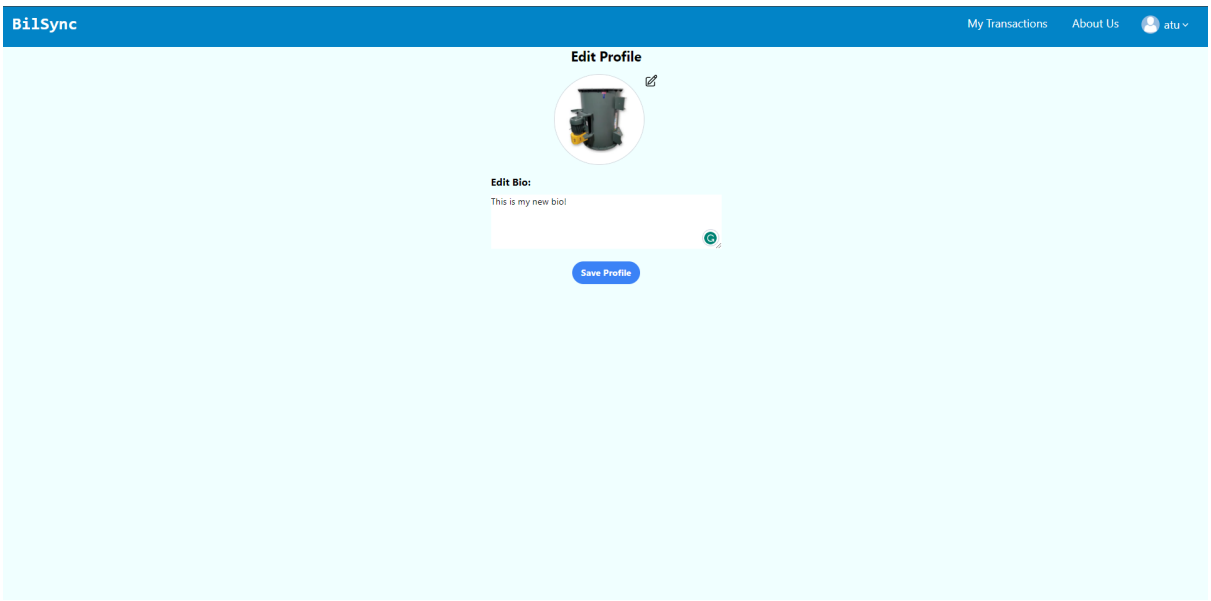
In the profile options, you can see the “profile page,” “edit account,” and “sign out” options. The user can use these to check their profile, change their password, and sign out from the current logged-in account.



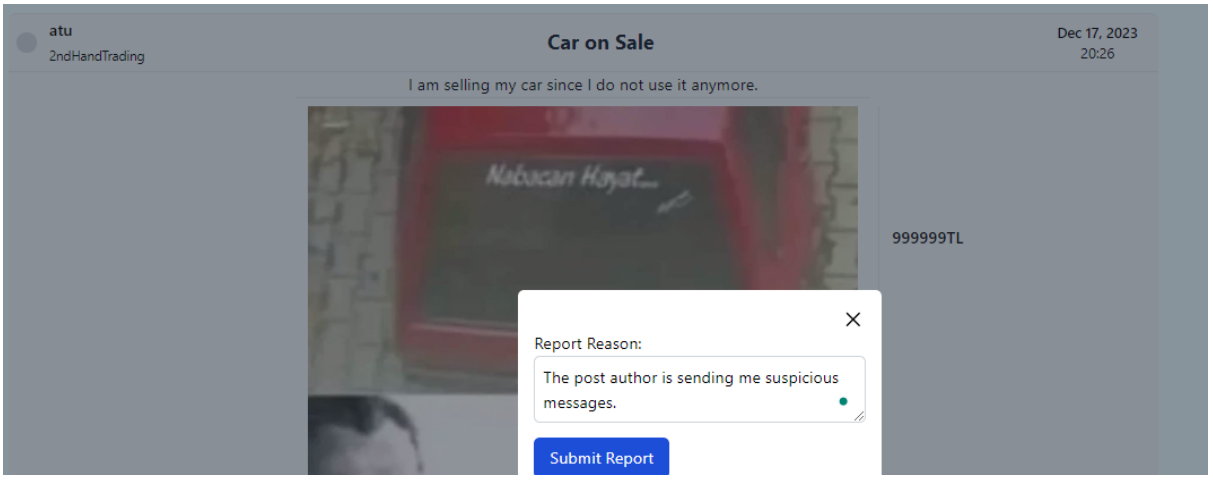
You can set a new password on the “edit account” page. This is different than the “forgot password” option that is in the login page. Since you are able to log in to your account with your current password, confirming that you are the owner of the account, you can easily change your password. When both fields (new password and confirm password) match, the new password will be set for the user.



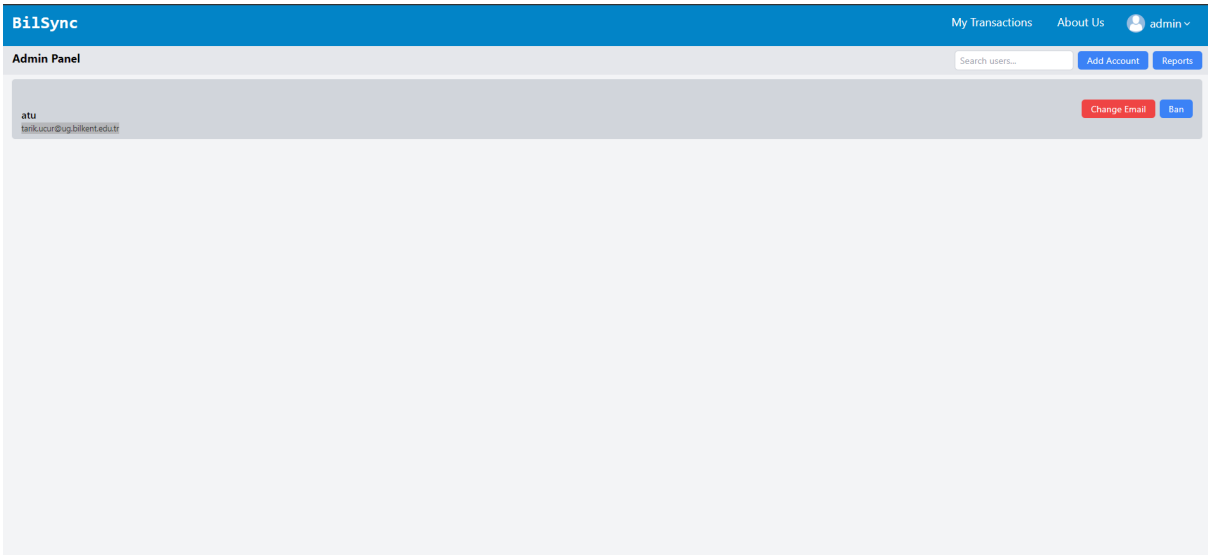
The user can see his information and image on the left panel of the profile page. The user's posts will be listed on the right panel.



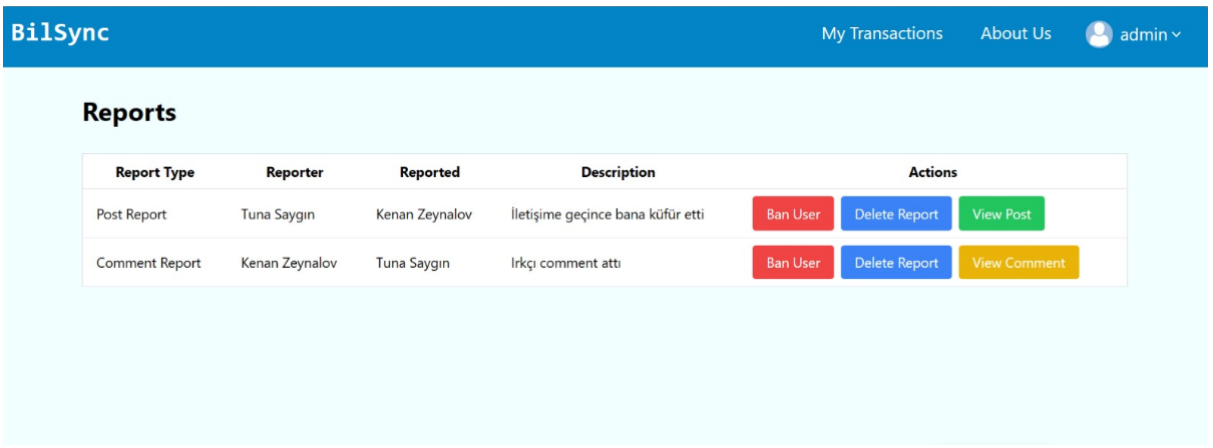
In the “edit profile” button that is inside the user page, the user can change their profile picture and bio, different than changing the password. This way, the user can change the default image they have if they do not like it, or set a bio so that the other users will be able to see it too.



When the user sees a post that is not fit for the friendly Bilkenter environment, they can report the post by using the report button given under the post, or the comment. Each report takes a description so that the admin will see the reason more clearly. After the report is created, the report will be sent to the admin's report page.



Admin will see all the reports shown in the admin's report page in order. Then, the admin can choose one of the reports to view its contents.



After viewing the report, the admin can decide whether he wants to ban the user based on the description given by the reporter. If the admin bans the user, the other user cannot log in to BilSync until the admin unbans the user. However, if the admin decides not to ban the reported user, the reported user will not be banned after the admin deletes the report.

Build Instructions

Database

Server requires Postgresql as database.
To install Postgresql on your local, follow the steps below:

- Install docker (If you are on windows, you should enable Windows Subsystem for Linux).
- On a terminal, run `docker volume create bilsync-sqldata`
- Change directory to `server/docker`
- Run `docker compose up -d postgresql`

These commands should run postgresql on its default port.

Sources

We have used maven. To compile the server codes, follow the steps below:

- On the terminal, navigate to `server`
- Run `./mvnw clean install` if you are in sh or run `mvnw.cmd clean install` if you are using windows. Or just run `mvn clean install` if you have maven installed.
- Then, run jar with `java -jar target/bilsync-0.0.1-SNAPSHOT.jar`

We have used JDK 21 for development and test environments.

Frontend

For the frontend ensure you are not using a headless environment, and you have npm installed,

- On the terminal, navigate to `Frontend/SynchFront`
- Run `npm install`
- Run `npm run dev`
- Go to the link provided in the console.
- From the link, you can start interacting with the frontend of the application.

Usage

- To create a user, log in as admin, predefined in the code for now.
- Log in with that user's email and password.
- Enjoy the perfect world of Bilkent.