Medication Tracking Application

By

Alperen Ekin

Gülenay Kayadibi

Halil Balcı

Report Delivery Date: 14/06/2021

Report Version No: 1.1

A Thesis Report Submitted to the

Faculty of Engineering in Partial Fulfilment of the

Requirements for the Degree of

BACHELOR OF SCIENCE

Department: Computer Engineering

İzmir Institute of Technology
İzmir, Turkey

June, 2021

We approve the thesis of **Alperen Ekin, Gülenay Kayadibi, Halil Balcı**

Date of Signature

....

Burak Galip Aslan

Supervisor

Department of Computer Engineering

TABLE OF CONTENT

1. THE DESCRIPTION OF WORK

The project consists of developing a mobile application to keep medication inventory in order to enable better usage and tracking of medication for patients who struggle to keep up with their medications and medication intake. We are aware of the risks of the wrong medication use and how hard it becomes to memorize all those medications in the long term and because of that we believe, it would be better for users to have all the necessary information and warnings inside of one application.

Based on these problems, in our project, users add the medication they have to their inventory. We know medications have a lot of information and adding all of them can be problematic. Because of this, besides adding this information manually, users have an option to use barcodes and QR codes which exist on the box of the medication. Therefore the application needs camera input in order to use QR code and also needs to access medication databases to get related information.

Additionally while adding a medication to inventory, users need to set intake information about it. By using this information, our mobile application notifies users about intake time of the medication and warns users when there is a problem related to the amount of medication left. At the end users are able to see a summary of their usage and history. In addition, we aim to deliver information about the pandemic that started in 2020. The pandemic tracker includes numbers of cases, active cases, deaths and people who have recovered. Furthermore, our application plays a role like an information bank. In this perspective, in addition to medications and covid-19 data, the pharmacy information is available. According to their location, users are able to display pharmacies on duty, which is closest to their location.

The application provides a user-friendly interface to enhance understanding of the users and the mobile application is expected to be available for both Ios and Android platforms when it is approved. It is connected to our database and in order to provide extra features such as pharmacy on duty, the program is in communication with third party API's as well.

2. ANALYSIS & DESIGN
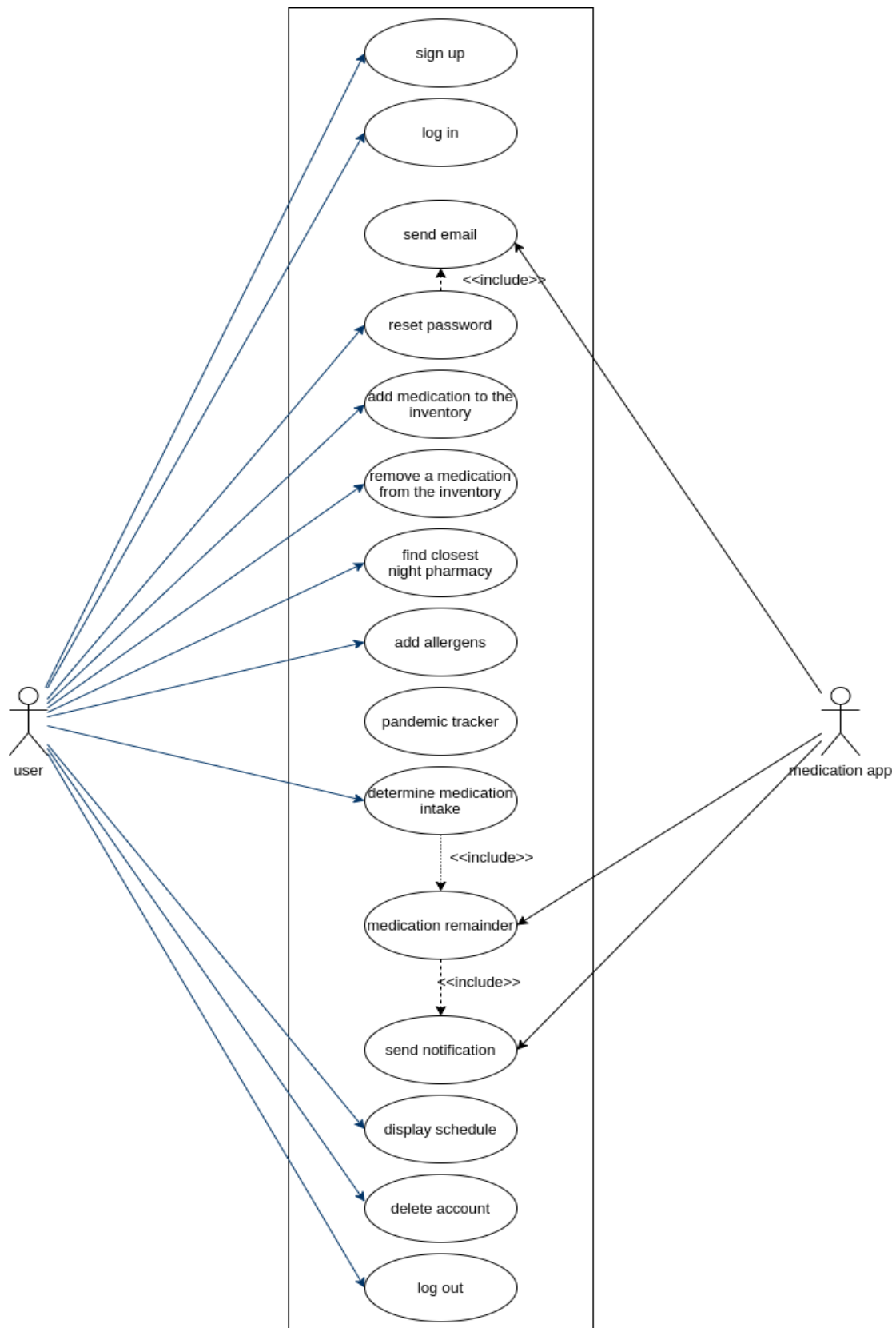
2.1 Use Case Scenarios

2.1.1. Sign Up
- Actor : User
- Description : User has to sign up to application to use application
- Preconditions :
  - -User must connect to internet.
  - -User should not already have an account for the application.
- Normal Course of Event :
  - -User enters password.
  - -User enters name.
  - -User enters e-mail.
  - -User enters birth date.
  - -User clicks on create account button.
  - -The User is created and added to the database.
- Exceptions:
  - -If the user clicks on create account button, All forms have to be filled. Otherwise, the application pop-up a warning and does not create an account.
  - -If an error occurs on database connection, the addition is canceled. Operation failed message shows up on the screen.

2.1.2. Log In
- Actor : User
- Description : User has to log in to the application.
- Preconditions : User has to sign up to the system already.
- Normal Course of Events :
  - -The user logs into the application using their credentials.
  - -User clicks on login button.
  - -The application checks the credentials of the user, if they are valid credentials, he user logged into the application
- Alternative Courses :
  - Alt 1-User clicks on forgot password button.
  - Alt 1.1-User directs to the forgot password screen.
  - Alt 2 -User click to login with google button.
  - Alt 2 -User logs into the application using their google credentials.
- Exceptions:
  - -The credentials of the user are not valid, the user doesn't log into the application and the application pop-up a warning.
  - -If an error occurs on database connection, then the operation failed message shows up on the screen.

2.1.3. Add Medication to Inventory
- Actor : User
- Description : User adds her/his medications to the medication inventory one by one.
- Preconditions : User has to log in to the application already.
- Normal Course of Events :
  - -Read QR code of the medication via camera by clicking on read QR code button.
  - -Verify the medication name and add it to the inventory by clicking on add medication button.

-The medication added to the medication inventory.
-Successfully added message shows up on the screen.
- Alternative Courses :
Alt 1-User adds medication to inventory manually.
Alt 1.1-User must enter medication name and also user can enter company name and active ingredient by hand.
Alt 1.2-The medication added to the medication inventory.
Alt 1.3-Successfully added message shows up on the screen.
- Exceptions:
-If the read QR code or the entered medicine name does not belong to a medicine, then the application pop-up a warning.
-If an error occurs on database connection, the addition is canceled. Operation failed message shows up on the screen.

2.1.4. Remove Medication from Inventory
- Actor : User
- Description : User deletes her/his medications from the medication inventory one by one.
- Preconditions :
-User has to log in to the application.
-There must already be medications in the inventory.

- Normal Course of Events :
-Select medication to be deleted.
-Click on remove medication button.
-The medication is deleted from the inventory.
-The medication is deleted and a message shows up on the screen.
- Exceptions:
-If an error occurs on database connection, the removing operation is canceled. Operation failed message shows up on the screen.

2.1.5. Determine Medication Intake

- Actors : User
- Description : User adds her/his daily medication intake to the application. If medication requires daily, weekly, or monthly use, user selects period as daily ,weekly or monthly.
- Preconditions :
-User has to log in to the application.
-User has to create a medication inventory by adding her/his medications.
- Normal Course of Events :
-User click to add reminder button.
-User selects period of use as day/week/month.
-User enters the number of times to use the medication during the period.
-User selects first intake time.
-User enters number of reminders.
-User clicks on the save button.

-Data are recorded to the database

-Successfully saved message shows up on the screen.
- Exceptions:

-If an error occurs on database connection, the operation is canceled. Operation failed message shows up on the screen.

2.1.6. Display Schedule

- Actors : User
- Description : User displays her/his weekly medication usage.
- Preconditions :

-User has to log in to the application.

-User has to create a medication inventory by adding her/his medications.

-User has to add her/his daily medication intake one by one into the application.
- Normal Course of Events :

-User enters the application's home page.

-The medicine reminders brought from local storage.

-The schedule shows up on the screen.

2.1.7. Find Closest Night Pharmacy
- Actor : User
- Description :User finds nearest night pharmacy to her/his location.
- Preconditions :

-User has to log in to the application.

-User has to allow the application to use the user's location from phone settings.
- Normal Course of Events :

-User enters province and district then press the find pharmacy button.

-Location of the closest night pharmacy will be seen on the screen.
- Exceptions:

-If the application does not reach the map, the application pop-up a warning and interrupt operation.

2.1.8. Add Allergens
- Actors : User
- Description : User adds her/his allergies and allergens to the application.
- Preconditions : User has to log in to the system.
- Normal Course of Events :

-User enters allergens one by one.

-User clicks on save button.

-The allergen is added to the local storage.

-Successfully added message shows up on the screen.
- Exceptions:

-If an error occurs on the local storage connection, the addition is canceled.

Operation failed message shows up on the screen.

2.1.9. Delete Allergens
- Actors : User
- Description : User deletes allergens.
- Preconditions :
    -User has to log in to the application.
    -User has to add allergen into the application.
- Normal Course of Events :
    -User selects allergen to be deleted.
    -User clicks on delete button.
    -The allergen is removed from the local storage.
    -Medication is deleted and a message shows up on the screen.
- Exceptions:
    -If an error occurs on the local storage connection, the removing operation is canceled. Operation failed message shows up on the screen.

2.1.10. Pandemic Tracker
- Actors:User
- Description: The pandemic situation in Turkey is shown.
- Preconditions :
    -Covid API must be active.(https://api.covid19api.com)
- Normal Course of Events :
    -User click to covid tracker button.
    -User directed to the covid tracker page.
    -User sees covid information from past to present.
- Exceptions:
    -If an error occurs on the API connection, operation failed message shows up on the screen.

2.1.11. Log out
- Actors : User
- Description : User logs out from the application.
- Preconditions :
    -User has to log in to the application.
- Normal Course of Events :
    -User clicks on the log out button.
    -User is quitted from the application.
    -Logged out message shows up on the screen.
    -User is directed to the login screen.

2.1.12. Delete account
- Actors : User
- Description : User deletes her/his account.
- Preconditions :
    -User has to log in to the application.
- Normal Course of Events :
    -User goes to settings page.
    -User clicks on the delete my account button.
    -User clicks on the confirm button.
    -The application asks "Are you sure?".
    -The user clicks to the yes button.
    -The account is deleted from the database.

-Account is deleted message shows up on the screen.
-User is directed to the login screen.
- Alternative Course of Events:
-The user clicks to the no button.
-The account is not deleted from the database.
- Exceptions:
-If an error occurs on database connection, the removing operation is canceled. Operation failed message shows up on the screen.

2.1.13 Send Email
- Actors: Medication Application
- Description: The application sends emails to the user.
- Preconditions:
-The application has to be online.
- Normal Course of Events:
-Forgot the password triggers a sending email.
-The application sends an email that includes a new password.
-The application updates the database.
-Successful message is displayed on the screen.
- Exceptions:
-If the mail is not delivered to the user, Failed to send the email message is displayed on the screen

2.1.14 Medication Reminder
- Actors: Medication Application
- Description: The application sends a notification at the medication intake time.
- Preconditions:
- The user has to determine a medication intake.
-The user has to allow for taking notifications from the medication app.
- Normal Course of Events:
-At the medication intake time, The application sends a notification.
-The user clicks the notification and the home page is opened.
-The user signs the check box whether he/she intake his/her medicine.
- Alternative Course of Event:
-If the user did not click the notification, Each of 15 minutes, A new notification is sent to the user. In the daily schedule, the check box is signed with x.
- Exceptions:
-If the user does not allow to take notification, then send a permission request.

2.1.15 Send notification
- Actors: Medication Application
- Description: The application sends a notification
- Preconditions:
-The user has to allow for taking notifications from the medication app.
- Normal Course of Event:
-The notification is sent to the user's phone.
-The user sees the notification on the phone screen and he/she can click to it.

-After the user clicks to the notification, the related page of the application is opened.
- ● Exceptions:
-If the user does not allow to take notification, then send a permission request.

2.2 ER Diagrams

According to our needs we moved our application to Firebase Realtime Database which is a nosql database. Therefore we do not maintain relationships between data anymore. The user and medication data stored according to the uid field. The database is key-value based, each uid contains a value according to table type.

User information - Firebase
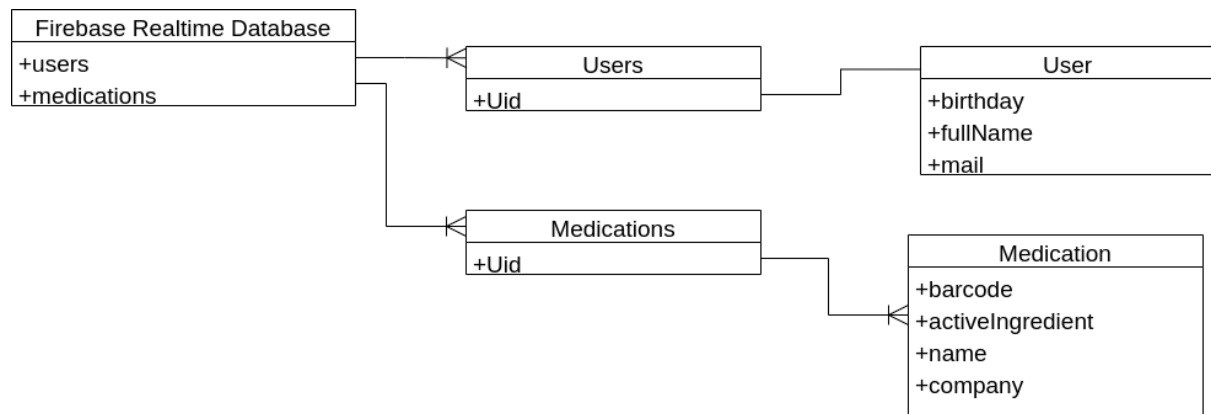
medicationapp-92ffd-default-rtdb
- **medications**
  - **EewYv5PTGcS7idFYnFlhUAmV0Si1**
  - **bAe34cWZmjPqOfx6P8nT9mOW5Uw1**
  - **wKwx5DNwCQdoCyEuzvGTGDqK7Bp1**
    - **-MaABDFX5D5g9nXLmOYg**
    - **-MaABEfR19ADjfhZ_zf-**
    - **-MaABGbRYGjjVLY2NPhG**
    - **-MaAGbY8TGn84dTL3BKg**
    - **-MaJzfwsa3RNCFBE4aGc**
      - **activeIngredient:** "silver sulfadiazine"
      - **company:** " "
      - **name:** "SILVERDIN % 1 KREM, 40 G"

Database location: Belgium (europe-west1)

2.3 Interface Descriptions

Our medication API is used to retrieve detailed medication information to our mobile application. It only accepts GET requests and if the request is successful it returns HTTP status 200. A successful request returns Medication barcode, company name, medication name and its active ingredient.

For Firebase Realtime Database, the GET, POST, PUT and DELETE requests are used. In order to create a new user POST method is used. PUT method is used to update user information. GET method fetches necessary user information and DELETE method is used to remove stored data.

3. SOLUTION/PRODUCT & RESULTS

3.1 Feasibility Results

As it is explained in our first report, as a new feature we make the use of barcodes that are given on the box of medication. However this feature comes up with some disadvantages. We use the Health Ministry records as a data source and they should be up to date, in order to provide correct information and include all the barcode numbers we need. There is a public

excel sheet about medication licence information that is provided by the Health Ministry and it shows the available medication list with their specific information(Such as name, barcode). By using spring boot, we make the data publicly available on a server. However, we have been using a free server(in order to avoid financial burden) which can be down under some circumstances. Another problem is the medications that are not available in the excel file. According to our trials, most of the medications can be found in our system, in some cases this problem occurs. We simply solve this problem by letting users add their medication without using our system and write the information by themselves.

3.2 Methodology Results

We applied Agile methodology all project long. The tasks are divided into subtasks in order to handle them in each sprint. We tried to release one feature at the end of each sprint. Once a release was made, we reviewed it in a meeting and made necessary changes according to provided feedback. The project started with the implementation of previously decided interfaces and we separated interface tasks such as Login screen, Registration screen, Homepage, Add medication screen, Night Pharmacy screen, Medication inventory and the more detailed screens.

Simultaneously, the back-end tasks were divided into subtasks and implemented. They were mainly divided into two sections. One was implementation of medication service and the other was focused on integrating Firebase services to the application in order to provide a database for data to be stored.

By applying agile methodology, we had a chance to keep better track of the tasks that are done. Having features done in a sprint allowed us to get more feedback on the task and improve it at the later stages. Additionally, it was possible for us to realize what problems we might face and how to resolve them in earlier stages.

3.3 Solution Results

In the project perspective, we aimed to utilize medication tracking for the users by providing visualization of medication inventory, showing a more efficient and user friendly way to add new medications to the system, keeping track of medication intakes, finding the closest night pharmacy and providing multi language in case it is necessary. As we tested our application, we realized how much it can be helpful for managing medications from one point, especially for the people who use more medications. Only requirements for the user are having a mobile phone and internet connection with it.

In order to provide additional features in our app, we have benefited from 2 public API: The night pharmacy API and the Covid-19 stats API. From the night pharmacy API, it was possible to get a list of Pharmacies according to district and city. In addition to the pharmacy name,address and phone, it provided us the latitude and longitude values for a pharmacy.

Therefore, we were able to find the user's location and calculate the distance between the user and the pharmacies. The Covid-19 API provides us the daily critical numbers by country related to Covid-19 pandemic.

To make the medication adding process easier for the user, we used the information provided in an excel file by the [Health Ministry](). We created our own public API by using Spring Boot which only allows users to send a GET request by providing a barcode number with 17 digits. In order to make the API publicly available, we used Heroku as a service provider.

In addition to adding medications to medication inventory, we let users define reminders for medications they choose. They are able to choose, frequency of medication, starting time and for how long they want to be reminded. Once the reminder is defined, the users are able to see defined reminders in a calendar. When the intake time comes, they are notified by our system so they are able to keep better track of their medication intake regularly.

The users must register to the application in order to use it. Therefore we keep their information in our database and ensure their data security. No data can be accessed without reaching the user's username,password and the token. Same process is valid for login to the application with a Google account. Furthermore, once the user add a medication to their inventory, these medications and the detailed information about medications are kept in Firebase Realtime Database as well. Once the user is logged in, they will be directed to a calendar view where they can see added medications and mark them as taken etc. By changing the screens, they can see Covid 19 page, medication inventory, night pharmacy, the profile page and new medication adding page.

In our next releases, we are planning to fix detected bugs and make improvements in user interface and in notification system when it is necessary.

3.4 Used Technologies

1. **Flutter :** Flutter is Google's UI toolkit for building beautiful, natively compiled applications for mobile, web, and desktop from a single codebase. We used Flutter for developing our mobile application and integrating it with the back-end. Since Flutter works cross-platform, we are able to easily integrate our application for both Ios and Android.

2. **Spring Boot:** Spring Boot is an open source, microservice-based Java web framework By using Java Spring Boot, we created an application programming interface for the list of medications we have in our system. By sending HTTP requests, anyone can access this publicly available API.

3. **Github :** GitHub, Inc. is a subsidiary of Microsoft which provides hosting for software development and version control using Git. We created 2 different repositories in github one for the back-end of our application and the other one is for our mobile application. For each feature in our application, we created a new branch

to separate tasks from each other.

4. **Firebase :** Firebase is a Backend-as-a-Service (BaaS) app development platform that provides hosted backend services such as a realtime database, cloud storage, authentication, crash reporting, machine learning, remote configuration, and hosting for your static files. In the first report, we mentioned that all of our back-end systems will be dependent on Java Spring Boot. However, later we decided to use it only for medication list since deploying the API in a server seemed to be costly and free server options were not stable enough. Therefore, we decided to move our User and Medication inventory databases to Firebase Realtime Database.

As a result of this decision, we have converted our relational database model to NoSQL database, since Firebase uses NoSQL model. We mapped relationships between models by using the User IDs. Thanks to Firebase, we are able to keep the user data safer because Firebase also provides an authentication system which runs fast and has easy integration with Flutter, by the help of Flutter Firebase packages. Also Firebase saved us from cost burden and also it was possible to integrate our application with other platforms like Google Login.

3.5. Use Case Descriptions & Requirements & Screenshots

3.5.1 Login: Since the application is integrated with Firebase services, the login process is handled by Firebase for both standard login and login by Gmail. For standard login, the user enters the email and password that is used in the registration. If any of these fields do not match with the data stored in Firebase, the user will be warned about it. In order to login with Google, the user should connect the google account and then they can start using the application. The user must have internet connection for both of the operations.
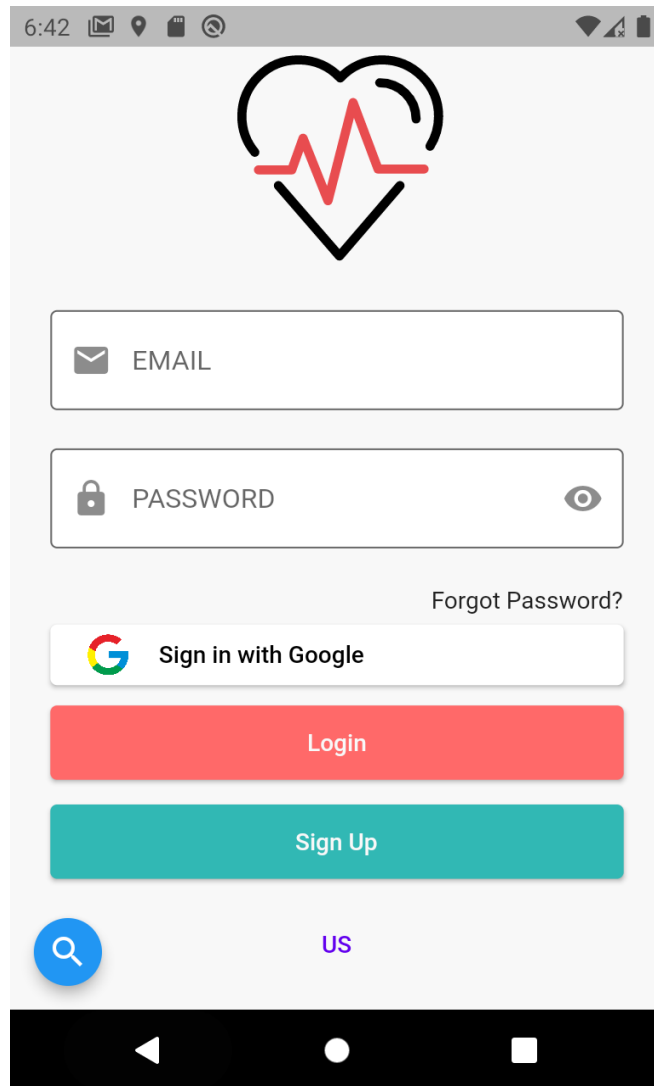
Figure 1 : Login Page

3.5.2 Signup: User has 2 opportunities to register for the application. He/She registers via Google or creates an account manually. In order to register via Google user should connect to a Google account like login process and then can start using the application. For creating an account manually, the user should enter name, id, e-mail address, password and birthday manually, then he/she will be able to use the application. If there is any missing or meaningless information, then the user will be warned. The user must have internet connection for both of the operations.
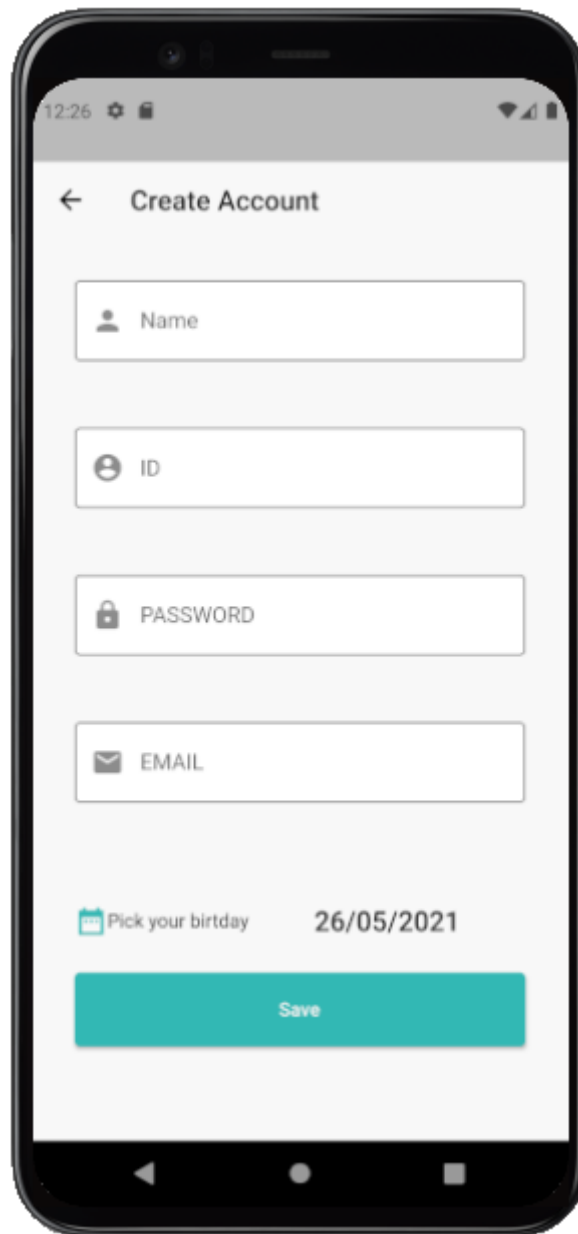
Figure 2 : Signup Page

3.5.3. Add Medication to Inventory: User adds her/his medications to the medication inventory one by one by reading the QR code with a mobile phone camera. Medication name, company and active ingredient of the medication comes after reading the QR code of the medication. For each request, Medication API returns a response, therefore the user must have internet connection. Additionally, while adding a new medication, if the user has an allergy to active ingredient of the medication, the user will be warned about it, however the user can still add the medication if they want.
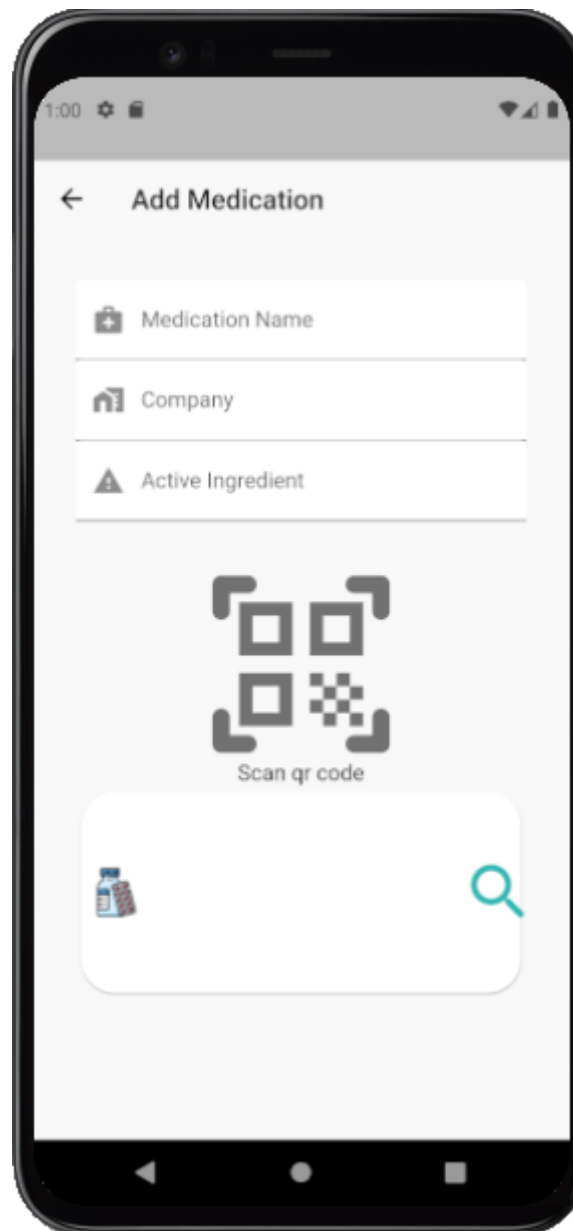
Figure 3: Adding Medication by QR Code
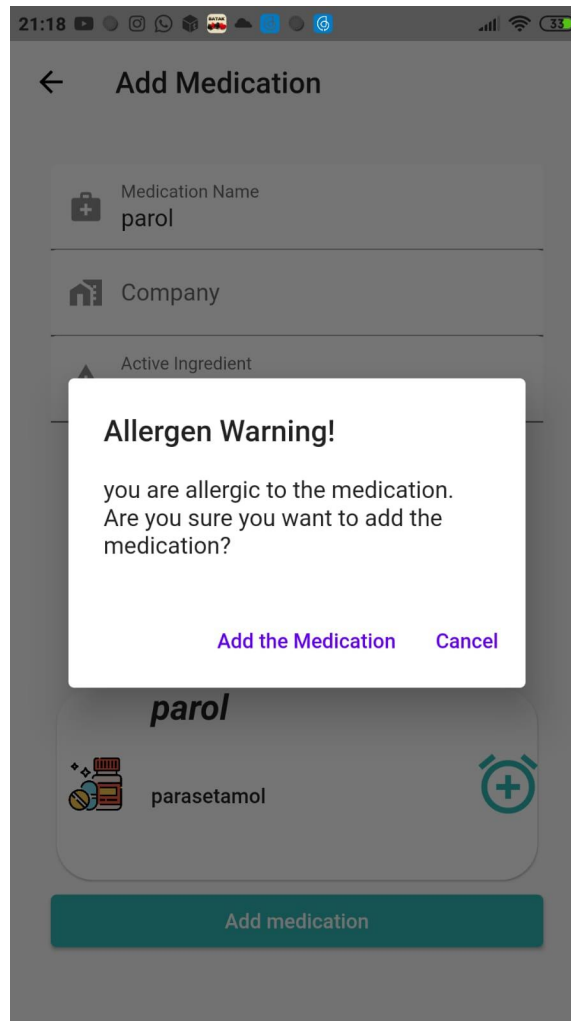
Figure 3.1: Allergy Check When Adding Medication
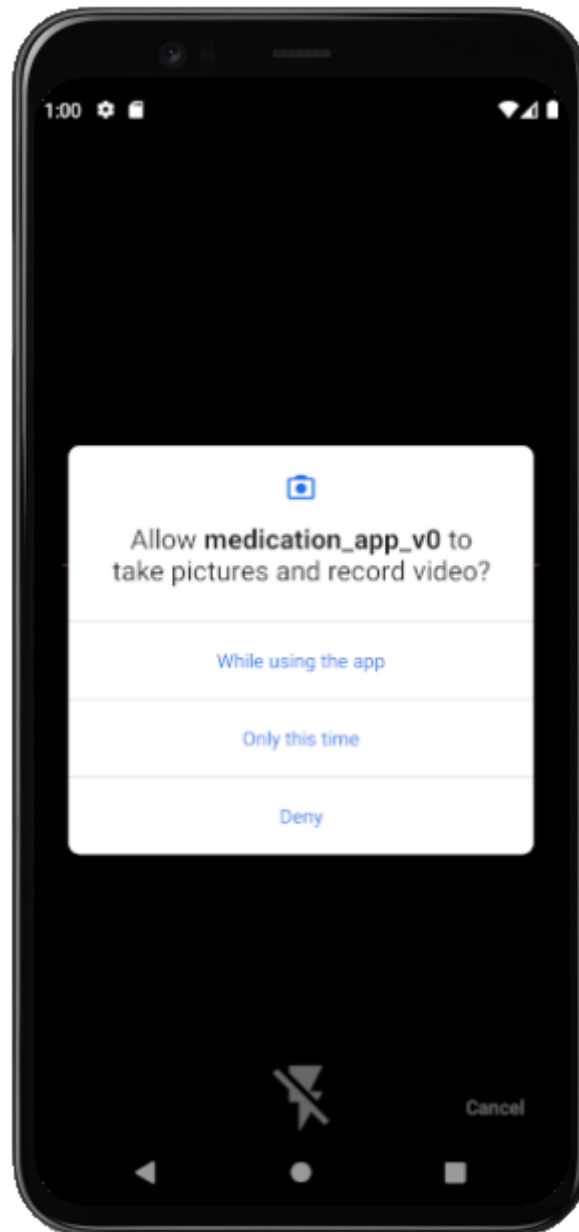
Figure 4: Allow Camera Access

Figure 5: QR Code Reading Screen
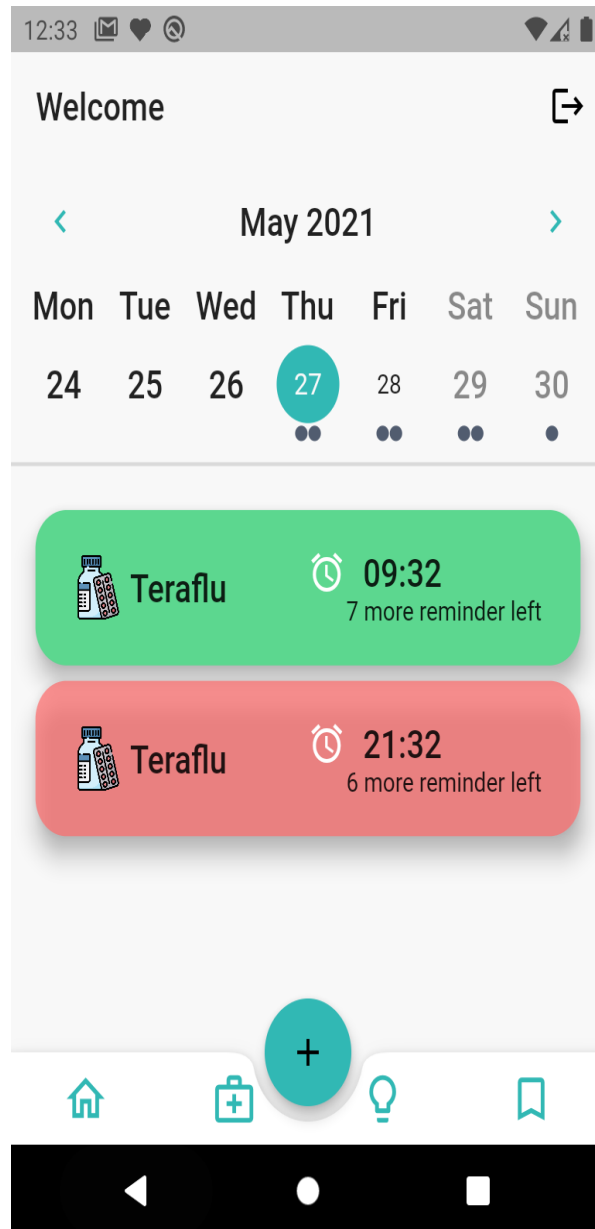
Figure 6: Medication Calendar

3.5.4. Remove Medication Reminder:
Users can delete the added reminder if he/she does not need it anymore. Pressing on a reminder will show a dialog on which the user can click on the delete button and delete that reminder.
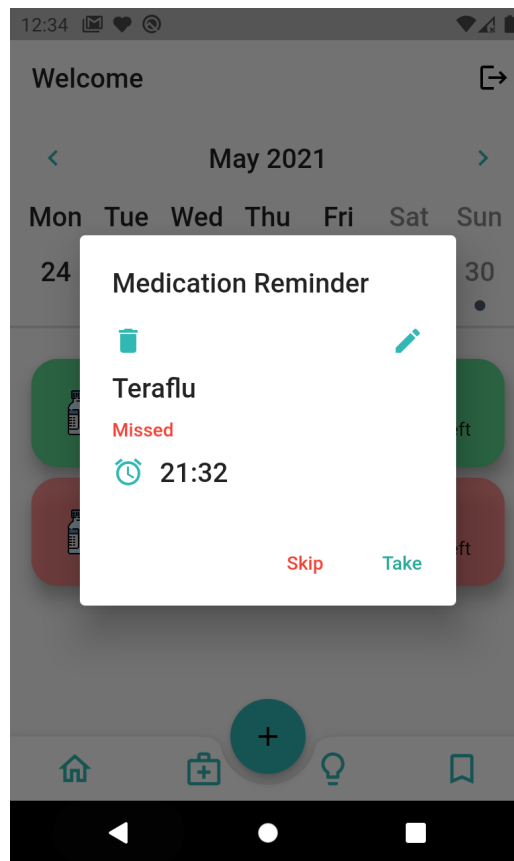
Figure 7: Medication Reminder

3.5.5. Display Medication Information: User displays detailed official information about selected medication. The details of a medication can be found in the medication view.
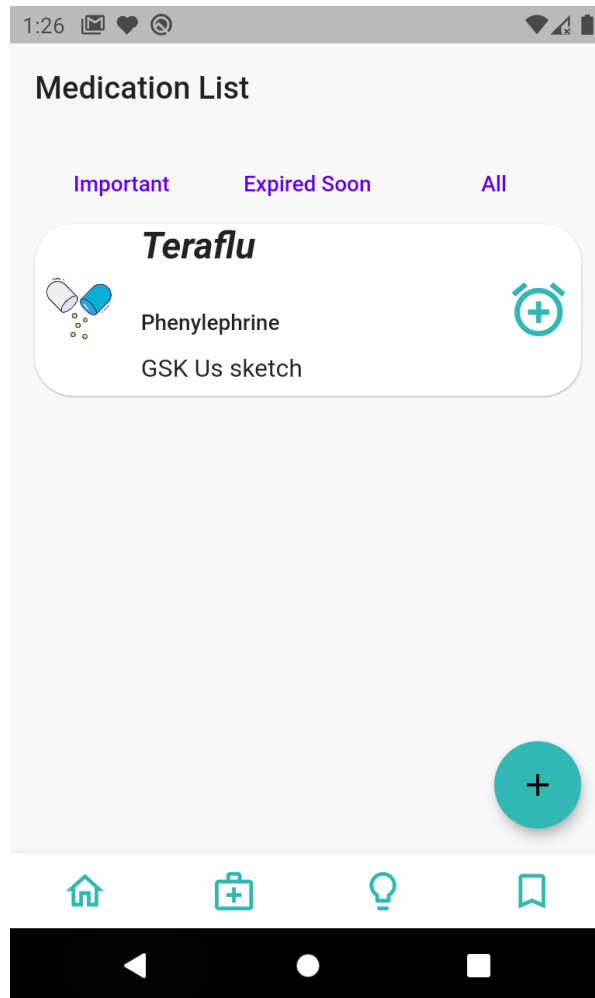
Figure 8: Medication Details

3.5.6. Find Closest Night Pharmacy: User finds night pharmacies in her/his city and district. The application detects the user's location and shows distances between the user and pharmacies in a given district. User can see the name, address and phone of the closest pharmacy. Night pharmacies are updated automatically, we used night pharmacy API to get information from the internet.
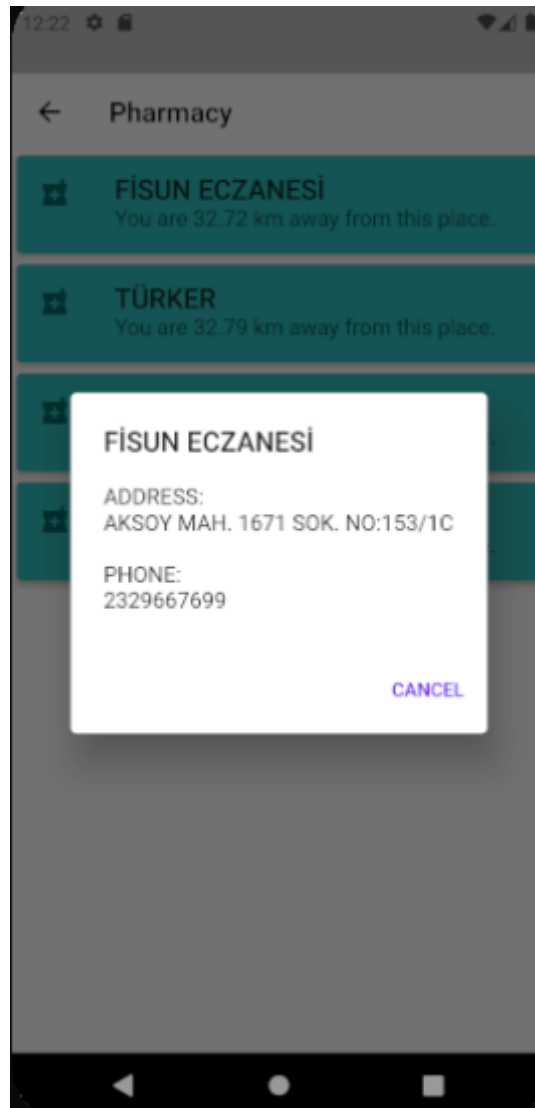can see it on the map.
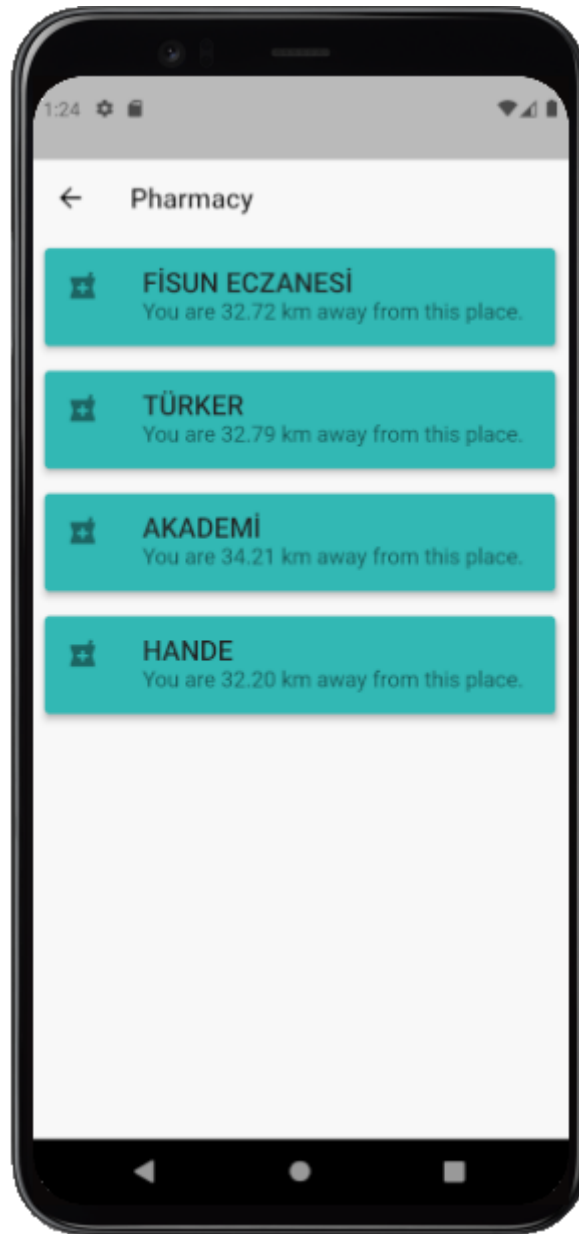
Figure 9: Closest Night Pharmacies

Figure 10: Night Pharmacy Information

3.5.7. Add Allergens: The user adds allergens to the application. The user clicks on the (+) button then a pop-up screen occurs. The user enters the allergen and clicks on the save button. Given allergen will be displayed on the screen. The application sends a warning to the user if any of her/his medication has an allergenic ingredient. In this way, possible allergic reactions can be prevented by the application warnings.
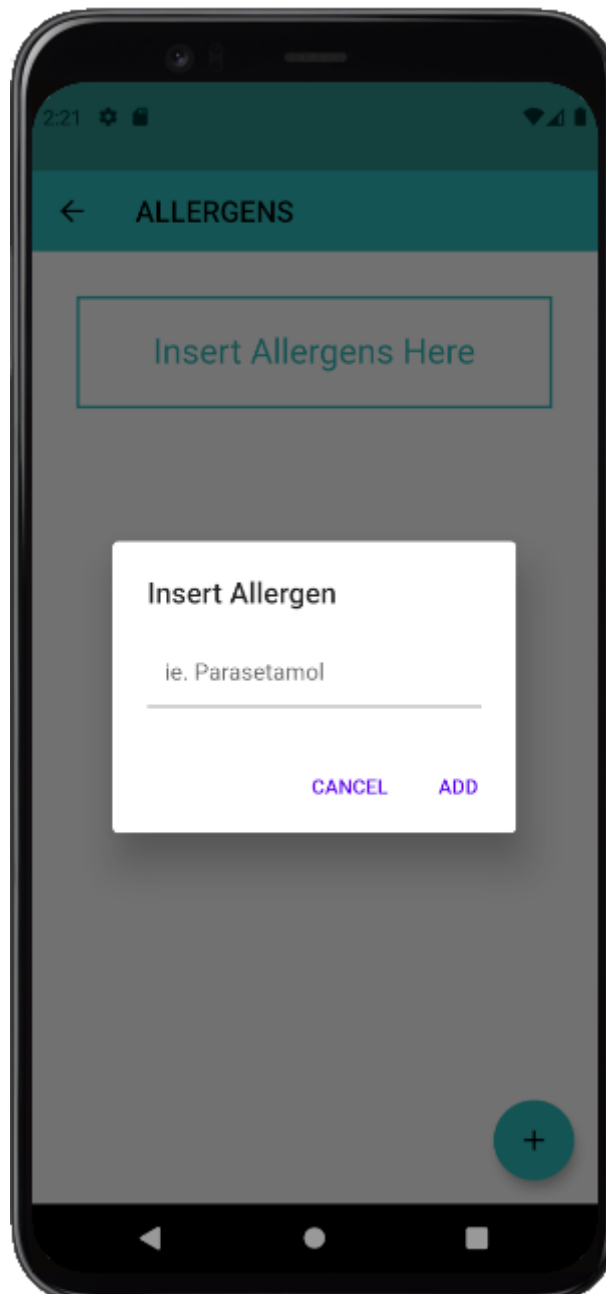
Figure 11: Add Allergen

3.5.8. Delete Allergen: The user deletes selected allergens. User presses long on the allergen that he/she wants to delete and then a pop-up screen occurs. The user clicks on the delete button and the allergen will be deleted.
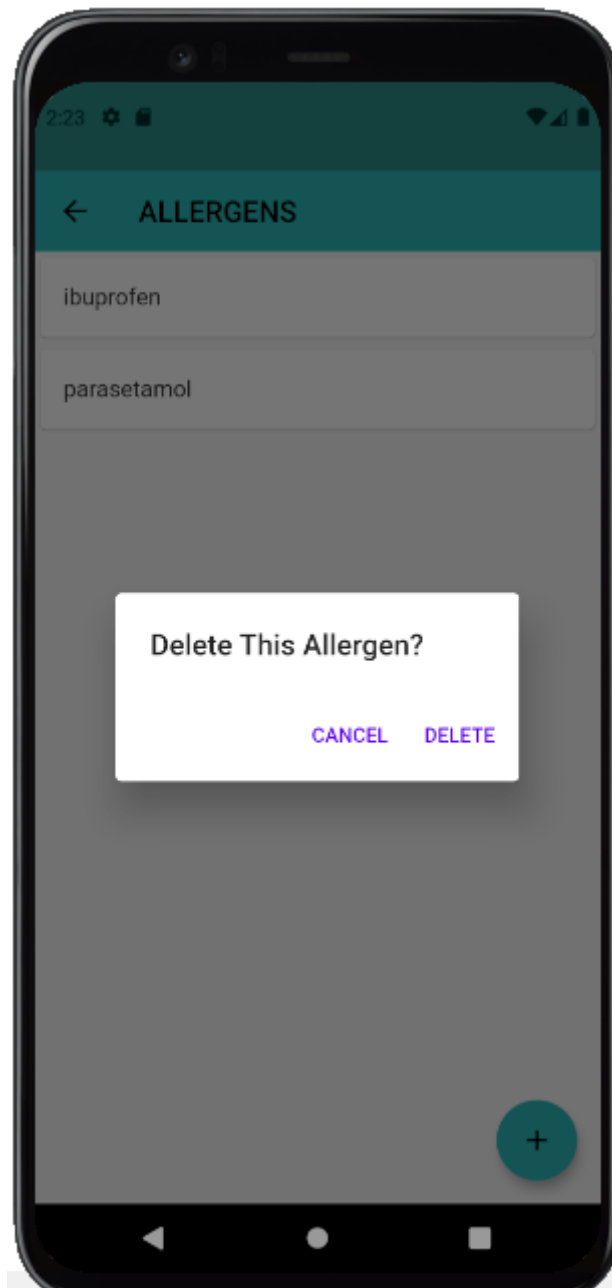
Figure 12: Delete Allergen

3.5.9.Determine Medication Intake: User adds her/his daily medication intake to the application. If medication requires weekly use or monthly use, the user selects the period as week or month. Users can track the intake period by using the application. The application sends notifications to remind the user of intake times of the medications.

3.5.10.Edit Medication Intake Time: User edits her/his daily medication intake time, if he/she would like to change the time for the medication.
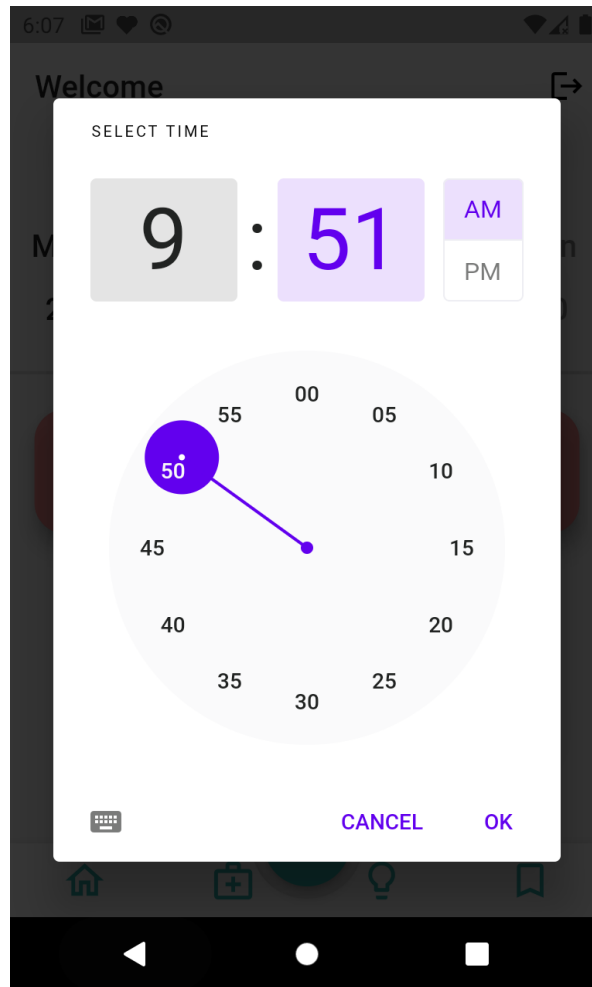
Figure 13: Editing Medication Intake Time

3.5.11. Display Schedule: User displays the medication for a certain time of period. In order to use this feature, the user should be logged in to the system.

3.5.12. Log out: User logs out from the application by clicking on the logout button. If the user is not logged in he/she can not see the logout button.
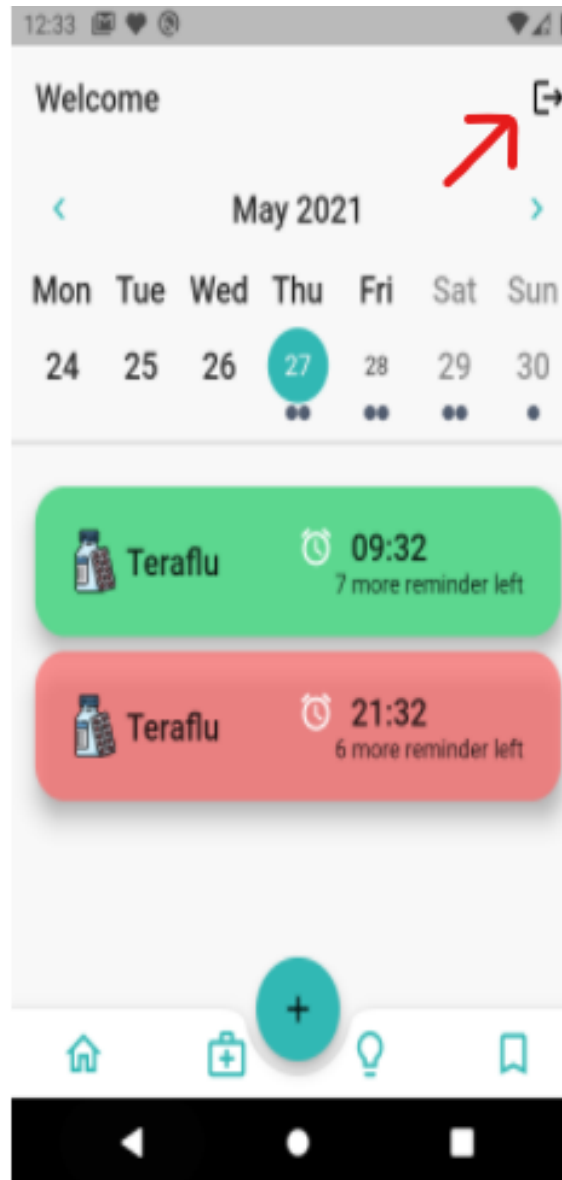
Figure 14: Logout

3.5.13. Medication Reminder: The application sends a notification at the medication intake time to remind the user. The user clicks the notification and the daily schedule is opened. The user signs the check box whether he/she takes his/her medicine. In order to have the medication reminder feature, the user must enter the medication intake time in advance. User should enter starting time, start date, period, how many times medication should be taken in a given period to set a reminder.
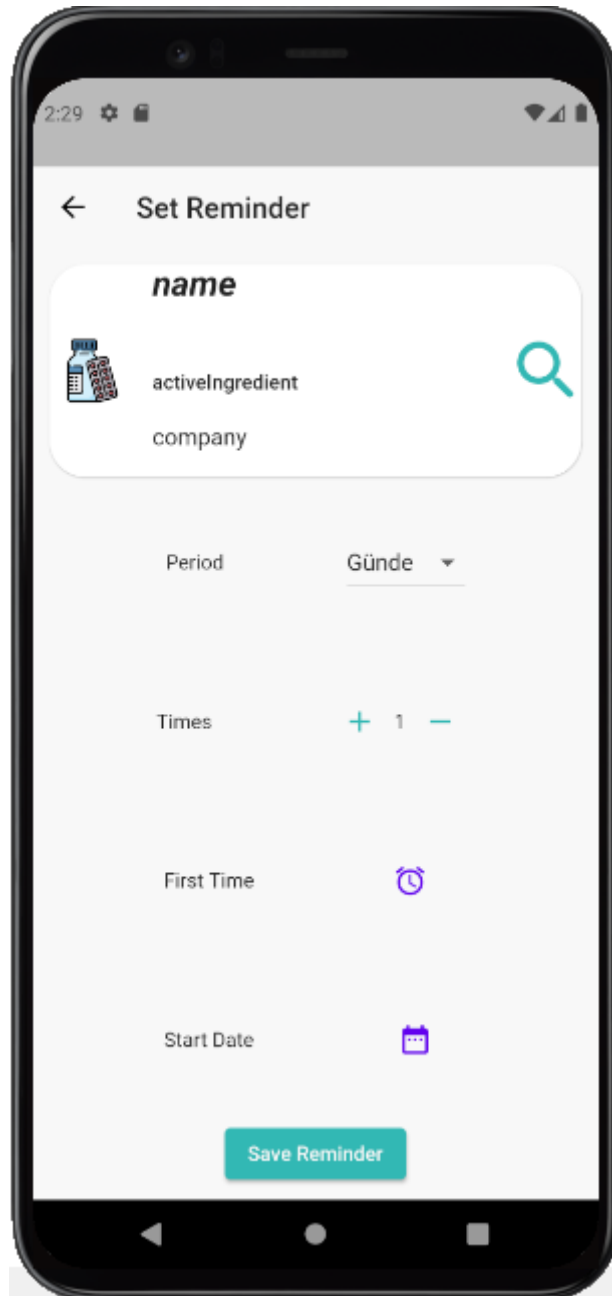
Figure 15: Set Medication Reminder

3.5.14 Send Notification: The application sends a notification. Firstly the user has to allow for taking notifications from the medication app, if the user does not allow to take notification, then send a permission request. After giving permission, the notification is sent to the user's phone, the user sees the notification on the phone screen and he/she can click on it. After the user clicks on the notification, the related page of the application will be opened.

3.5.15. Profile: User information will be filled automatically after the registration process. User can change his/her personal information from the profile page. The user can enter his/her age and gender into the application which is not obligatory. After changes in profile, the user clicks on the save button.
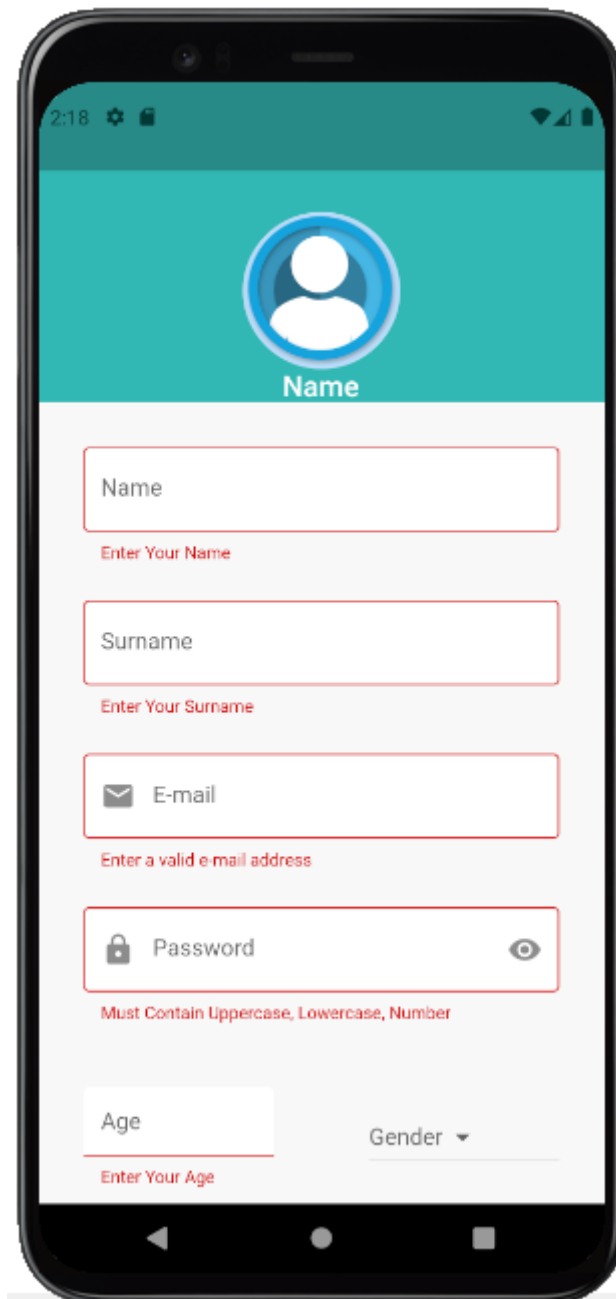
Figure 16 : Profile

3.5.16. Forgot Password: The user enters his/her email address which is used in the registration process and clicks on the send button to reset the password. After clicking on the send button a link comes to the user's email. When the user clicks on the link then he/she will be directed to the reset password page.
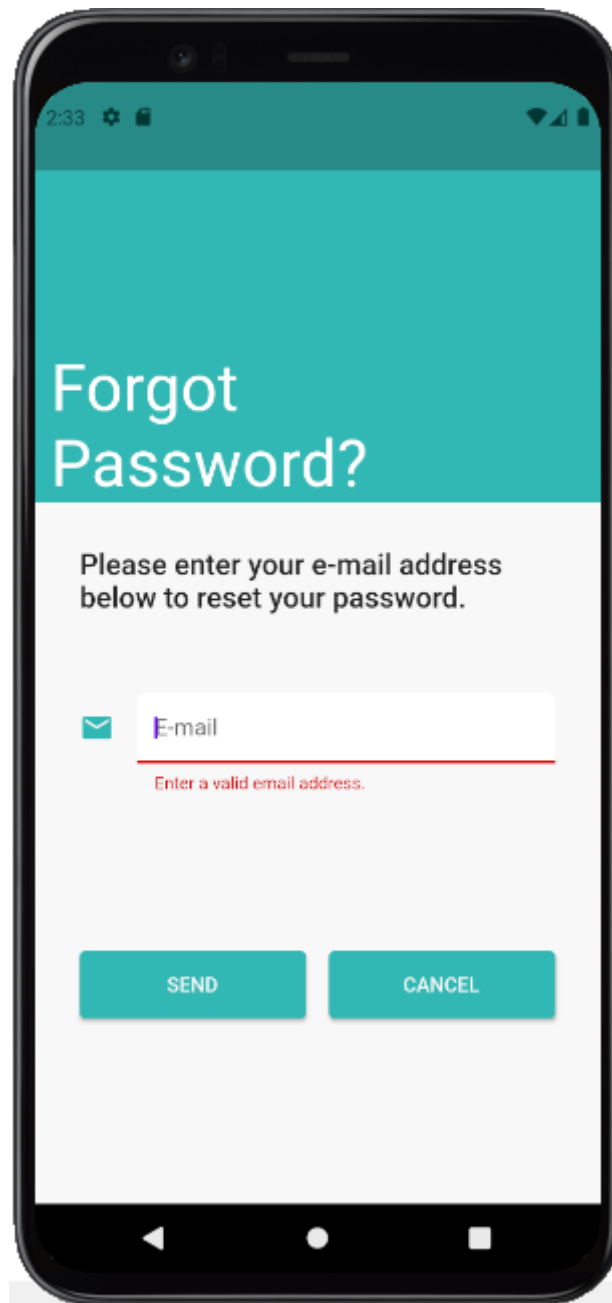
Figure 17: Forgot Password

3.5.16. Reset Password: User resets his/her password. If the user forgets the password then he/she will be directed to the reset password page. The user enters a new valid password two times. Password validation exists in two stages. In the first stage it checks whether the user enters a password which contains at least one upper case, one lower case, one numeric number and 8 characters long, if conditions are not held then it warns the user. In the second stage it checks whether two entered passwords are the same, if they are not the same then the application warns the user. User clicks on the save button if he/she entered a new valid password.
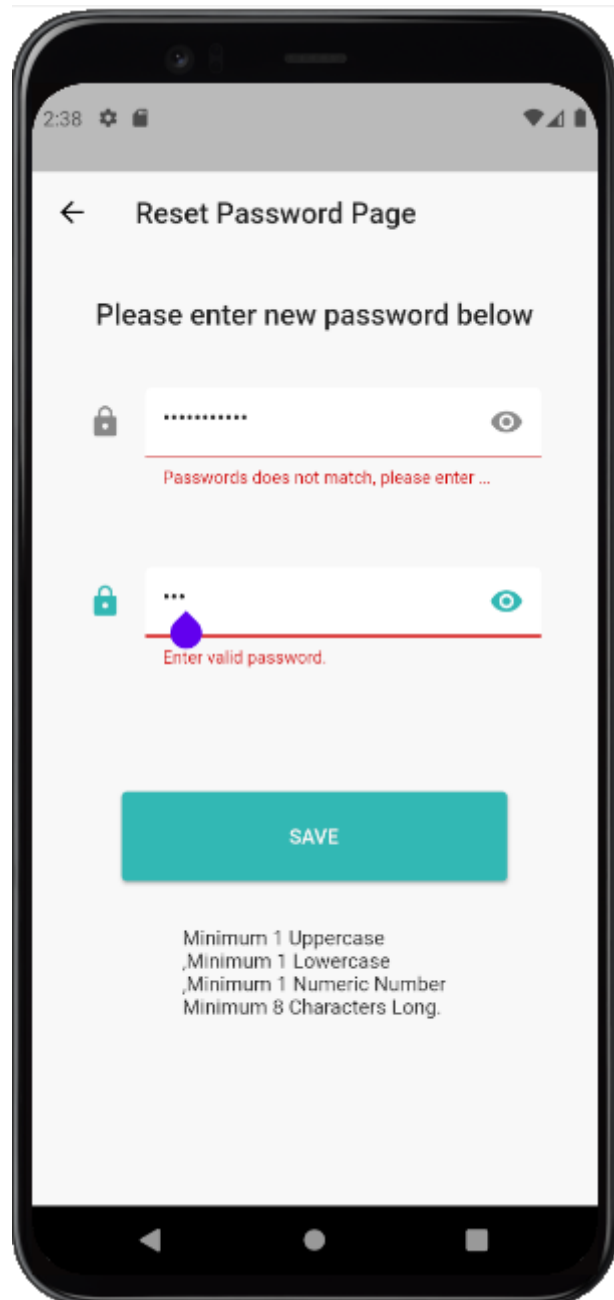
Figure 18: Reset Password

3.5.17. Show Turkey Covid Table: User can see covid information about Turkey. Total and daily number of patients, covid cases and healing number is given by the application. We used an API to get related information from the internet, the table is updated automatically.
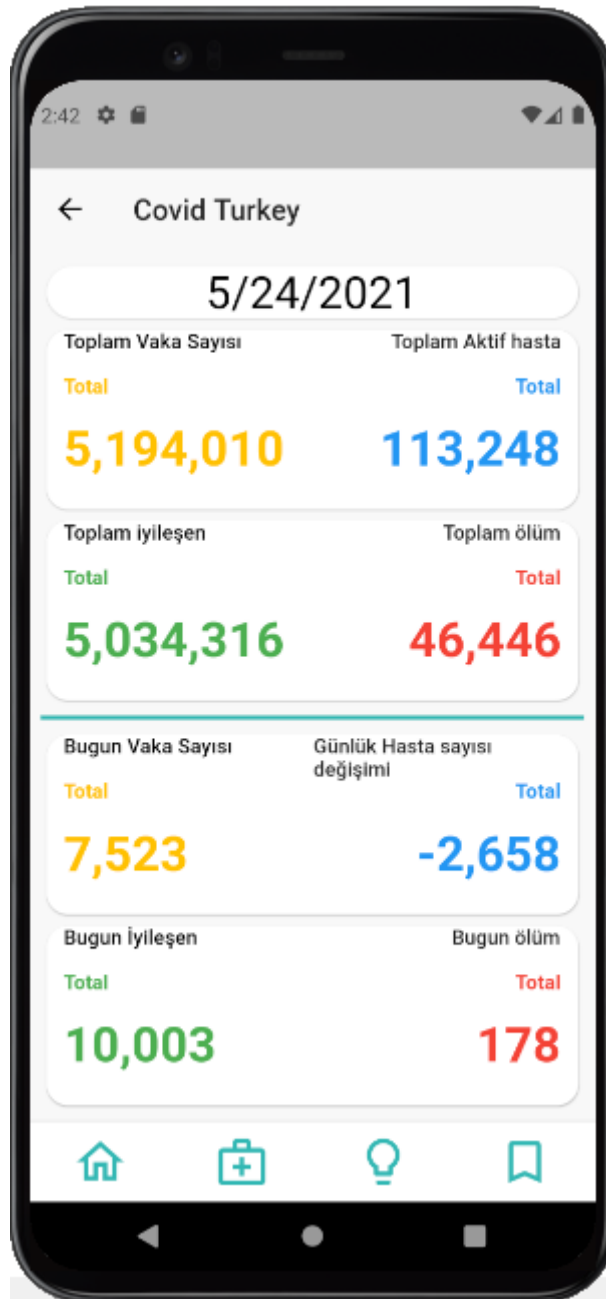
Figure 19: Turkey Covid Table

3.6 Non-Functional Requirements

- **Performance** : The application responds to operations such as adding reminders instantly. The operations which require server connection such as Covid-19 table, scanning a medication by using barcode number, checking profile and login operation responds under 300ms.
- **Usability** : Application designed to be user friendly with different types of animations.
- **Reliability** : The application has an error catching system. When an error occurs related to server side or in the application itself, the users are notified about it with snackbars.
- **Maintainability** : In case any unexpected error is detected during further tests and by the users, we will release a fixed version of it in the next stages.
- **Availability** : A mobile phone with an internet connection is enough to use the application. Online services will not be available when there is no connection. Additionally the application will be possible to be used from both Ios and Android devices.
- **Security** : The user's personal data and medication data is stored in a secure way in Firebase.When a user logs in to the application, they receive a token, which is used to access their personal data. Therefore without having user credentials, the user data can not be accessed.

4. RELATED WORK/SIMILAR SOLUTIONS

Top Competitors to our medication application:
- Medisafe
- Dosecast
- MyTherapy

Reasons to choose our medication application / Good sides of our application:
- Clearer design for the users.
- No pro version, every user will be able to reach all features.
- Better medication adding mechanism, by using QR code and barcode, users will fill less information and save time.
- More specific medication detail information.

Application Name	Medication Reminder	Medication History	Covid-19 daily information	Display pharmacies on duty	Cross-Platform	Warning for allergens	Display official information about medication	QR code	Multi language support	Medication Friend
Medisafe	+	+	-	-	+	-	-	-	-	+

Dose cast	+	Needs Pro version	-	-	+	-	-	-	-	-
MyT hera py	+	+	-	-	+	-	-	-	-	+
Our App	+	+	+	+	+	+	+	+	+	-

5. CONCLUSION & IMPACT

The application helps people to track daily/weekly/monthly medication intake and they can reach detailed information about medications in a second. Users can use a QR code system to add medications to her/his inventory, this provides easier usage of the application. All information about the medication like name, company, amount, ingredients etc. comes after reading the QR code. The user does not need to waste time to add medication information one by one. Application also provides a notification system to remind user medication intakes, the user enters the intake times then he/she does not need to remember intake times of several medications.With this feature forgetting rate of medication intake is going to be decreased. In addition the application warns users about allergens therefore bad situations because of allergens can be prevented. With these properties as reminders, advice, practicality etc. users can be more systematic about medication usage and can be in good health.

The application displays night pharmacies of the current day (updated automatically) and distances between the user and night pharmacies, therefore the user can see the closest night pharmacy. This will help users to get the address and phone of the pharmacies which are working that day. With this property, the user does not waste time finding a pharmacy in emergency situations.

After the Covid-19 pandemic, everyone checks the number of cases every day, our application provides a table that shows the total and the daily number of Covid-19 cases, patients and healing number. By looking at this table the user can be aware of up-to-date information about cases and can take precautions with the knowledge of how serious the Covid-19 virus is.

- Economics: Mobile application is free, therefore there is no economical constraint.
- Environmental problem : There is not a problem because application will be developed to be helpful to users.
- Sustainability : Application will be alive until creators pull the application from the market.

- Producibility : The application can be published in the market once it is approved.
- Ethics : There is no ethical problem about the application, it does not harm anything in out opinion. It will be developed to be beneficial to users. Data about whether the application harms any user or not can be determined by looking at scores and comments of the application on Play Store and App Store.
- Health : The aim of the application is to be helpful to users about their medication usage therefore health impact will be in a very good way.
- Social and Political Issues : There is no social and political issue because this application's only aim is to be beneficial to users about their medication usage.

6. REFERENCES

[1]. Flutter Development https://flutter.dev

[2]. Java Spring Boot - https://www.jrebel.com/blog/what-is-spring-boot

[3]. Firebase - https://flutter.dev/docs/development/data-and-backend/firebase

[4]. Github - https://en.wikipedia.org/wiki/GitHub

[5]. Non-Functional requirements
https://www.altexsoft.com/blog/non-functional-requirements/