National College of Ireland

BSc in Computing

Cybersecurity

2023/2024

Veselin Karamfilov

x20120885

x20120885@student.ncirl.ie

Take over Control

Technical Report

Contents

Executive Summary

This report aims to provide information about the intended use and technical details of the Take Over Control application. Although it will be explained in detail in the following sections, the purpose of the application is to deter the user from alcohol consumption by recording alcohol intake and money spent on alcohol and visualizing these records. It aims to comprehensively explain all the details in terms of technical functionality and design, as well as other features, including both frontend and backend aspects.

1.1. Background

I chose this project topic because, recently, I was asked how many units of alcohol I drink per week when requesting quotes from insurance companies for home and car insurance. I realized I didn't have an answer to this question. I don't really have a regular drinking habit, but I thought it wouldn't be bad to know how much alcohol I generally consume in a week, month, or year. I thought this would not only visualize my alcohol consumption from a health perspective, allowing me to take preventive steps if necessary, but also help me see the money I am spending on alcohol. Simply put, I want to know how much alcohol I consume and how much money I spend on it.

1.2. Aims

This project aims to visualize the statistics of users' alcohol consumption and money spent on alcohol, as well as to reduce the rate of alcohol consumption accordingly. Visualized facts have more impact and are more deterrent.

1.3. Technology

The Java programming language was utilized in Android Studio IDE and the primary library was AndroidX. Other libraries, like Firebase and Firestore SDK, were employed for user authentication and database integration purposes. Along with these main libraries, user authentication, database integration, RESTful API communication, error handling, and some security assessment algorithms and approaches were incorporated.

The application was developed in two stages. Basic features, manual data entry, and statistical results were implemented in the first stage. Later, depending on the timeline situation, advanced features like a barcode scanner and notifications were included in the application. For these features, libraries such as ML Kit Android libraries and Google Play Services were integrated into the application.

Regarding other technical details, since Android Studio provides a device emulator, the execution and testing of the code were performed by running the emulator. Adobe Photoshop was used to design the graphical interface of the application's pages. GitHub, an indispensable tool for software development and version control, was utilized.

Additionally, various other technologies and adaptations were made as required during the application development.

1.4. Structure

This document serves as a technical report for creating the "Take Over Control" application, emphasising the functional aspects and intended purpose. Below is an explanation of each section covered in the document:

1.4.1. Executive Summary

Provides a brief overview of the application's purpose and functionality.

1.4.2. Introduction

1.4.2.1. Background

Gives information about why this topic was chosen for application development.

1.4.2.2. Aims

Summarises the project's objectives.

1.4.2.3. Technology

Describes the specifics of the technologies (Java, Android Studio, Firebase, Firestore SDK, ML Kit Android libraries, and Google Play Services) that were utilised during development.

1.4.2.4. Structure

Explains all the structure that was used in the application. (section 1.4)

1.4.3. System

1.4.3.1. Requirements

Covers the application's functional, data, user, environmental, and usability requirements.

1.4.3.2. Design & Architecture

Illustrates the fundamental algorithms employed, as well as the system's design and parts.

1.4.3.3. Implementation

Defines the primary functions, classes, and algorithms utilised in the code, possibly with some code snippets.

1.4.3.4. Graphical User Interface (GUI)

Provides wireframe designs of the application's user interface, along with actual designs.

1.4.3.5. Testing

Provides proof and outcomes for unit, integration, and end-user testing procedures, plans, and tools used in testing.

1.4.3.6. Evaluation

Supplies quantitative results and discusses the evaluation process of the system, covering usage data, performance evaluations, scalability, and correctness.

1.4.4. Conclusions

Analyses the project's benefits, drawbacks, assets, and limitations.

1.4.5. Further Development or Research

Explores opportunities to provide more time and resources for development or research.

1.4.6. References

Lists all of the references used in the work.

1.4.7. Appendices

1.4.7.1. Project Proposal

Contains the initial proposal for the project.

1.4.7.2. Ethics Approval Application

Includes the forms if necessary for the project.

1.4.7.3. Reflective Journals

Contains reflections on the project's progress.

1.4.7.4. Other Materials Used

Consists of any supplementary reference materials utilised in the project.

2.0 System

2.1. Requirements

To use the application, you need a smart device with an Android operating system, an Android account to download the application from the Android market, and internet access.

2.1.1. Functional Requirements

Requirements	Description
1	User Registration
2	Login
3	Forgot Password
4	Value Addition
5	Quick Value Addition
6	Value Edition
7	Value Deletion
8	Stats
9	Logout

2.1.1.1. Requirement 1 – User Registration

2.1.1.1.1. Description & Priority

This use case allows users to create an account to log in and use the application. It is a high priority since an account is required to use the application's functionality.

2.1.1.1.2. Use Case

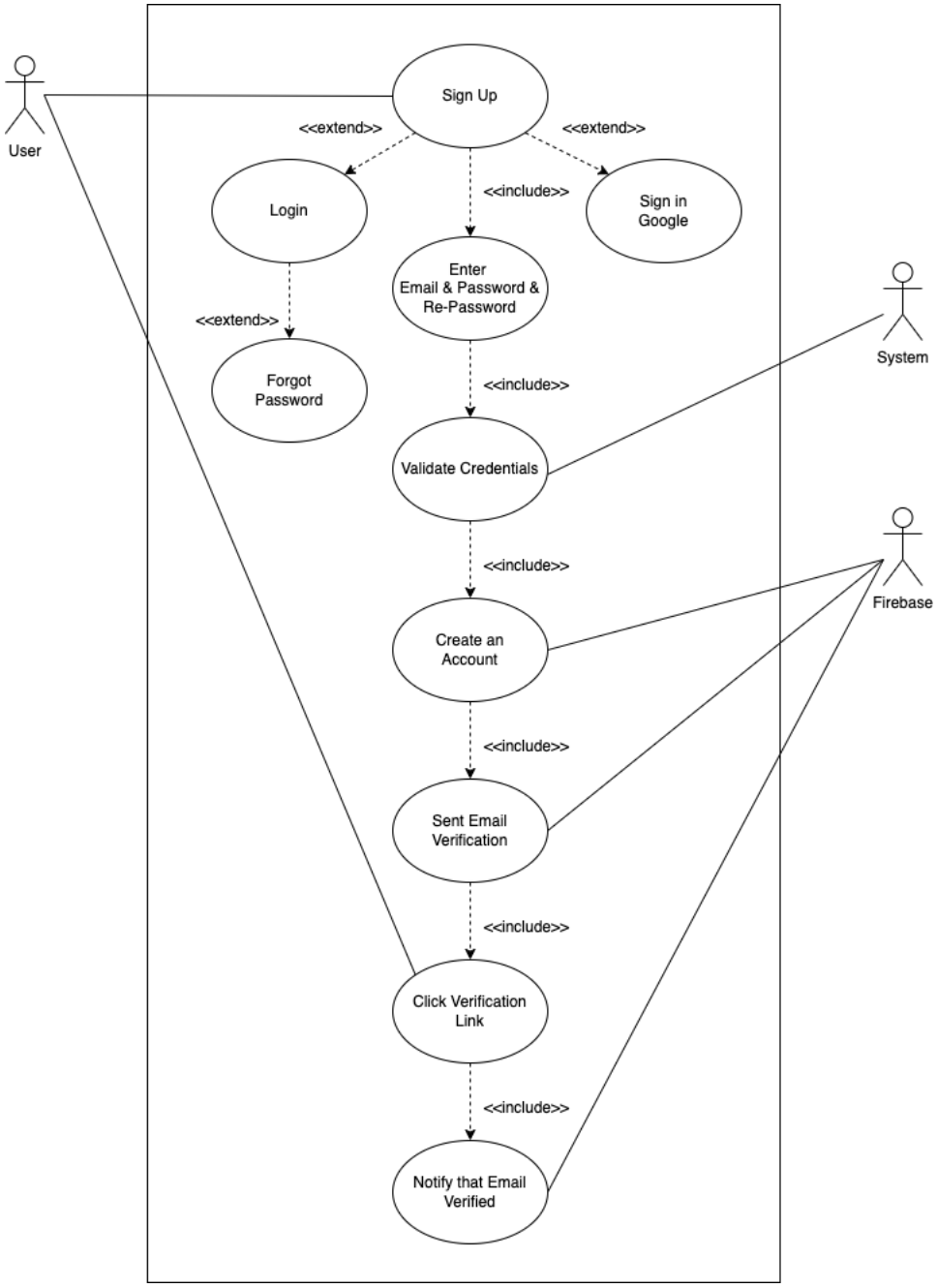
Scope

This use case scope is for creating an account for the user.

Description

This use case describes the new user registration flow.

Use Case Diagram



Flow Description

Precondition

The application runs on the registration page.

Activation

This use case starts when a user clicks the Create Account button.

Main flow

1. The user enters an email and password.

2. The system validates if the email address is already registered; if not, it checks if the email address format and password requirements are met.
3. An account is created by Firebase and sent an email verification link via email.
4. The user clicks the link and verifies the email.
5. The user is notified that the email is verified.
6. The account is ready to log in.

Alternate flow

A1: The user already exists - 1
1. The firebase detects the email that already exists in DB.
2. The user enters another email.
3. The use case continues at position 2 of the main flow.

A2: The user already exists - 2
1. The firebase detects the email that already exists in DB.
2. The user clicks the Login page.
3. The use case continues at 2.1.1.2. Requirement 2 – Login to App position 1 of the main flow.

Exceptional flow

E1: Any other unexpected system errors.
1. The system displays an error message with the reason that the user creation is not successful
2. The use case continues at position 4 of the main flow.

Termination

The user gets confirmation that the registration process has been completed successfully.

Postcondition

After successful account registration, the user is able to log in using the registration credentials.

2.1.1.2. Requirement 2 – Login

2.1.1.2.1. Description & Priority

This use case allows users to log in and use the application. Since login is required to use the application, it's also a high priority after creating an account.

2.1.1.2.2. Use Case

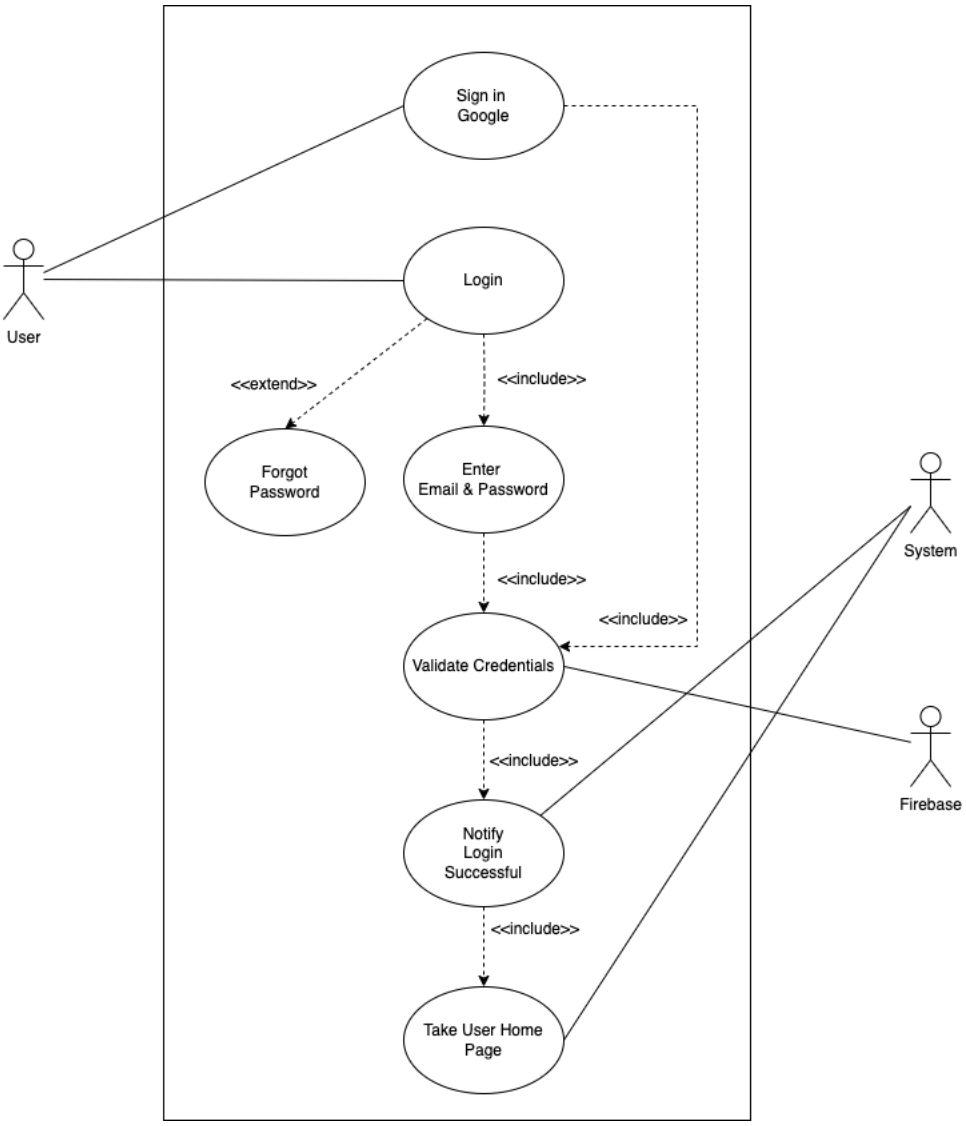
Scope

This use case scope is for logging in to the application.

Description

This use case describes the logging flow.

Use Case Diagram



Flow Description

Precondition

The application runs on the login page.

Activation

This use case starts when a user opens the application if it's not already logged in.

Main flow

1. The user enters the email and password.
2. The Firebase validates the credentials.
3. The user is notified that the Login is successful.
4. The user is taken to their Home Page.

Alternate flow

A1: Credentials are not valid
1. The Firebase validates the credentials as invalid.
2. The user is asked to enter the credentials again.
3. The use case continues at position 1 of the main flow.

A2: Forget the password
1. The Firebase validates the credentials as invalid.
2. The user clicks Forgot Password.
3. The use case continues at 2.1.1.3. Requirement 3 – Forgot Password position 1 of the main flow

A3: Sign in with a Google Account
1. The user clicks the Google Sign-in button
2. The user enters Google Account credentials or chooses the existing one
3. The user is taken to their Home Page

Exceptional flow

E1: Any other unexpected system errors
1. The system displays an error message with the reason that the user cannot log in successfully
2. The use case continues at position 1 of the main flow

Termination

The user gets confirmation that the log is successful.

Postcondition

After successfully logging in, the user is taken to their Home page.

2.1.1.3. Requirement 3 – Forgot the Password

2.1.1.3.1. Description & Priority

This use case allows users to reset their passwords to log in to the application. It is not a priority as long as the user enters their password correctly. However, if Firebase does not validate the password as valid, resetting the password by the user is a high priority with this use case.

2.1.1.3.2. Use Case

Scope

This use case scope is for resetting the account password.

Description

This use case describes the resetting password flow.

Use Case Diagram

Flow Description

Precondition

The user must have registered credentials.

Activation

This use case starts when a user clicks the Forgot Password button.

Main flow

1. The user enters the email.
2. The Firebase send a password reset link via email.
3. The user clicks the link and enters the new password.
4. The user is notified that the password changed.

5. The user logs in.

Alternate flow

A1: Email is not valid
1. The Firebase validates the email as invalid.
2. The user is asked to enter the email again.
3. The use case continues at position 1 of the main flow.

Exceptional flow

E1: Any other unexpected system errors.
1. The system displays an error message with the reason that the user cannot log in successfully.
2. The use case continues at position 1 of the main flow.

Termination

The user gets confirmation about password changes.

Postcondition

After successfully resetting the password, the user is able to log in.

2.1.1.4. Requirement 4 – Value Addition

2.1.1.4.1. Description & Priority

This use case allows users to enter values. This use case has no priority; however, the application would only be useful with value.

2.1.1.4.2. Use Case

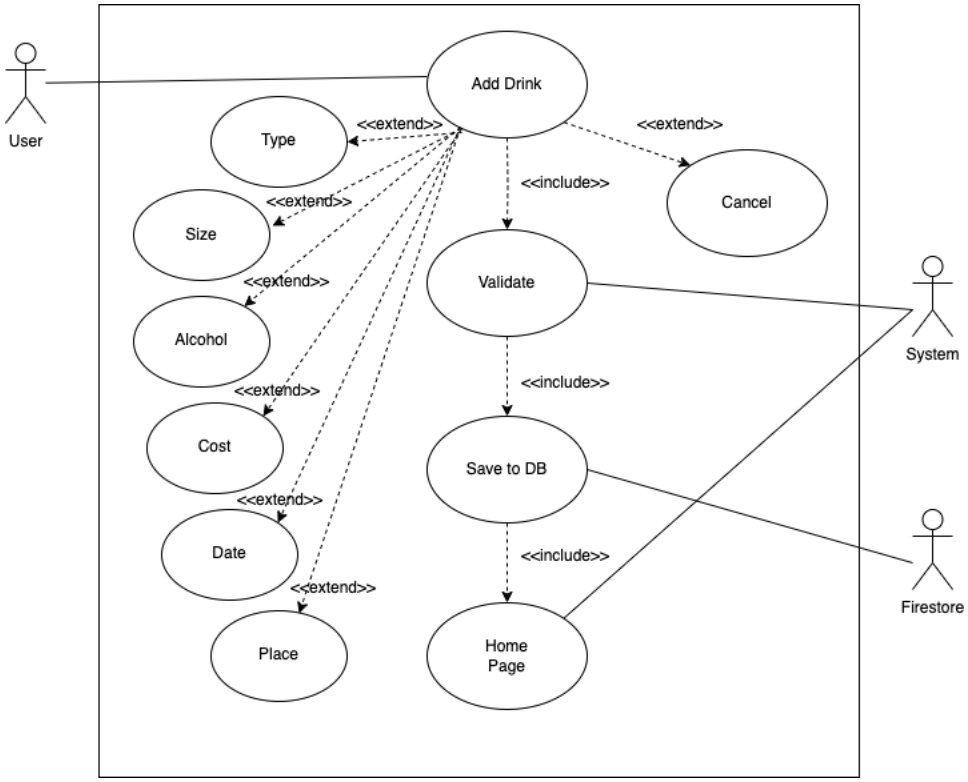
Scope

This use case scope is for adding values to the user account.

Description

This use case describes the adding a value flow.

Use Case Diagram



Flow Description

Precondition

The application runs on the Home page.

Activation

This use case starts when a user clicks the Add button.

Main flow

1. The user enters the type of drinks, size, alcohol percentage, cost, date and place.
2. The system validates the entries.
3. The user clicks the Save button.
4. The values are saved to DB.
5. The home page is shown again to the user after saving the values.

Alternate flow

A1: Cancel the addition
1. The user clicks the Cancel button.
2. The use case continues at position 5 of the main flow.

Exceptional flow

E1: Any other unexpected system errors.
1. The system displays an error message with the reason that the user cannot save the details successfully.

2. The use case continues at position 1 of the main flow.

Termination

The user adds the values successfully.

Postcondition

Provided that there is existing data, this data can be edited or deleted afterwards.

2.1.1.5. Requirement 5 – Quick Value Addition

2.1.1.5.1. Description & Priority

This use case allows users to enter values quickly, either by choosing a brand or scanning a barcode. This use case has no priority.

2.1.1.5.2. Use Case

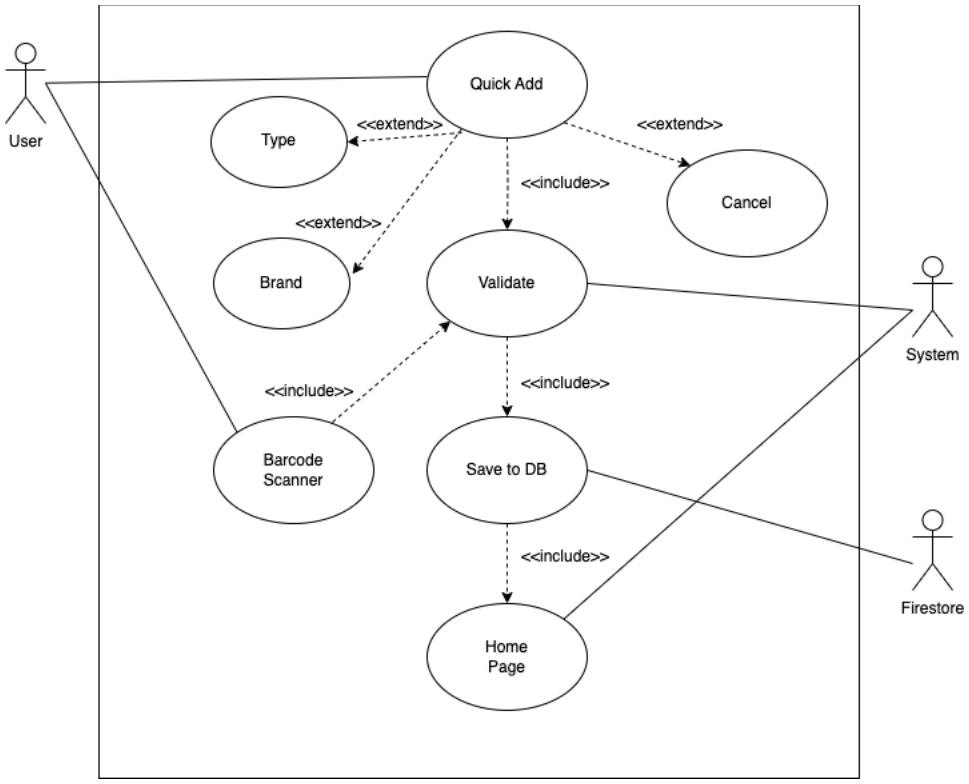
Scope

This use case scope is for adding values quickly to the user account.

Description

This use case describes the adding a value flow.

Use Case Diagram



Flow Description

Precondition

The application runs on the Home page.

Activation

This use case starts when a user clicks the Quick Add button.

Main flow

1. The user chooses the type of drinks and brand. Alternatively, the user clicks the “Open Barcode Scanner” button and scans a barcode.
2. The user clicks the Save button.
3. The values are saved to DB.
4. The home page is shown again to the user after saving the values.

Alternate flow

A1: Cancel the addition
1. The user clicks the Cancel button.
2. The use case continues at position 4 of the main flow.

Exceptional flow

E1: Any other unexpected system errors.
1. The system displays an error message with the reason that the user cannot save the details successfully.

2.1.1.6. Requirement 6 – Value Edition

2.1.1.6.1. Description & Priority

This use case allows users to edit the values. This use case has no priority, and its functionality depends on an existing value.

2.1.1.6.2. Use Case

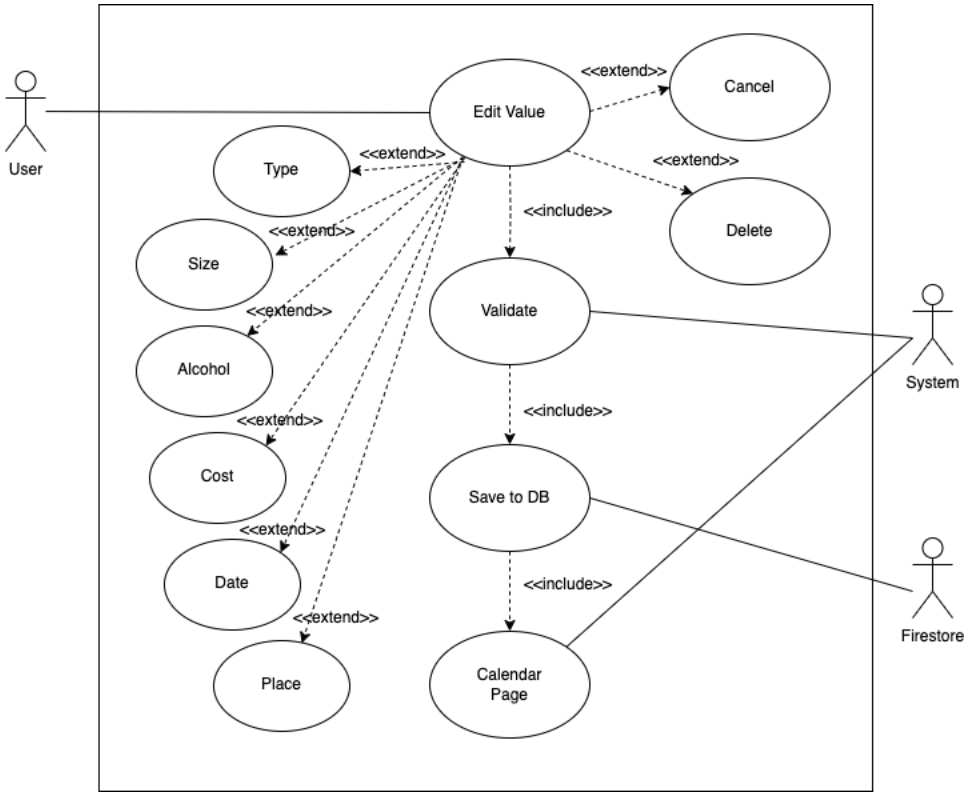
Scope

This use case scope is for editing an existing value.

Description

This use case describes the editing of a value flow.

Use Case Diagram



Flow Description

Precondition

The application runs on the Calendar page.

Activation

This use case starts when a user chooses one of the values from the listed date.

Main flow

1. The user selects one of the values saved for the selected day from the calendar.
2. The user edits the type of drinks, size, alcohol percentage, cost and place.
3. The system validates the entries.
4. The user clicks the Save button.
5. The values are saved to DB.
6. The calendar page is shown again to the user after saving the values.

Alternate flow

A1: Cancel the edition
1. The user clicks the Cancel button.
2. The use case continues at position 1 of the main flow.

A2: Delete the value

1. The user clicks the Delete button.
2. The use case continues at position 1 of the main flow.

Exceptional flow

E1: Any other unexpected system errors.

1. The system displays an error message with the reason that the transaction couldn't be made successfully.
2. The use case continues at position 1 of the main flow.

Termination

The user edits and saves the value successfully.

Postcondition

The data can be edited again or deleted.

2.1.1.7. Requirement 7 – Value Deletion

2.1.1.7.1. Description & Priority

This use case allows users to delete the values. This use case has no priority, and its functionality depends on an existing value.

2.1.1.7.2. Use Case

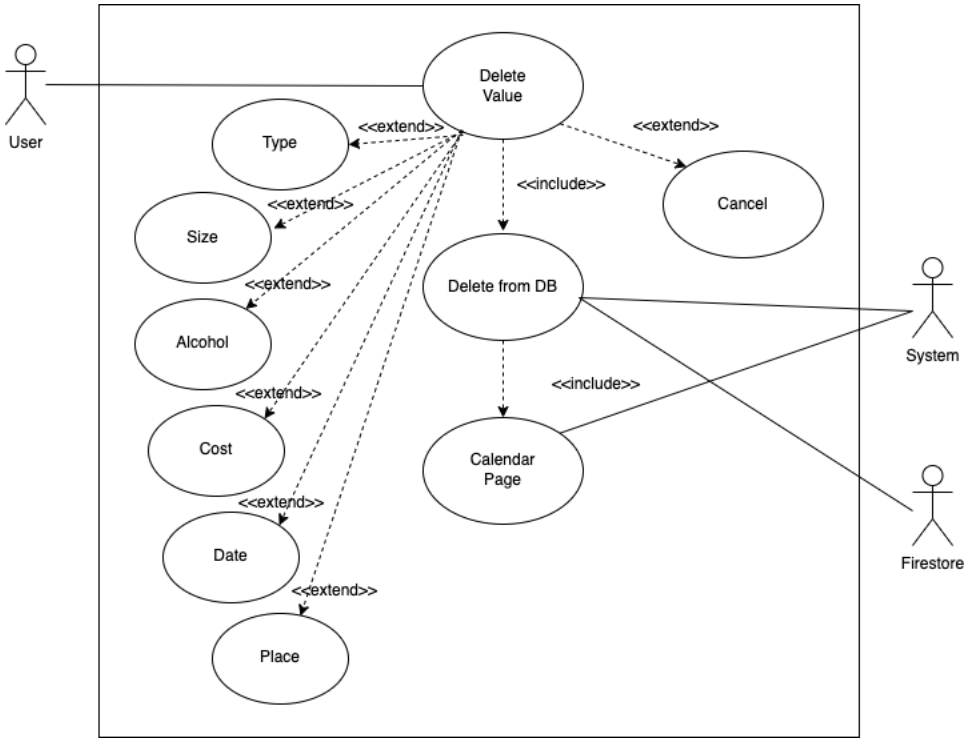
Scope

This use case scope is for deleting an existing value.

Description

This use case describes the deleting of a value flow.

Use Case Diagram



Flow Description

Precondition

The application runs on the Calendar page.

Activation

This use case starts when a user chooses one of the values from the calendar.

Main flow

1. The user selects one of the values saved for the selected day from the calendar.
2. The user clicks the Delete button.
3. The value is deleted from DB.
4. The calendar page is shown again to the user after deleting a value.

Alternate flow

A1: Cancel the deletion
1. The user clicks the Cancel button.
2. The use case continues at position 1 of the main flow.

Exceptional flow

E1: Any other unexpected system errors.
1. The system displays an error message with the reason that the transaction couldn't be made successfully.
2. The use case continues at position 1 of the main flow.

Termination

The user deletes a value successfully.

Postcondition

The deleted data can be added again by following 2.1.1.4. Requirement 3 – Value Addition or 2.1.1.5. Requirement 4 – Quick Value Addition

2.1.1.8. Requirement 8 – Stats

2.1.1.8.1. Description & Priority

This use case allows users to see the stats. Since statistics constitute the application's primary purpose, this use case has the highest priority after user registration, login and data addition.

2.1.1.8.2. Use Case

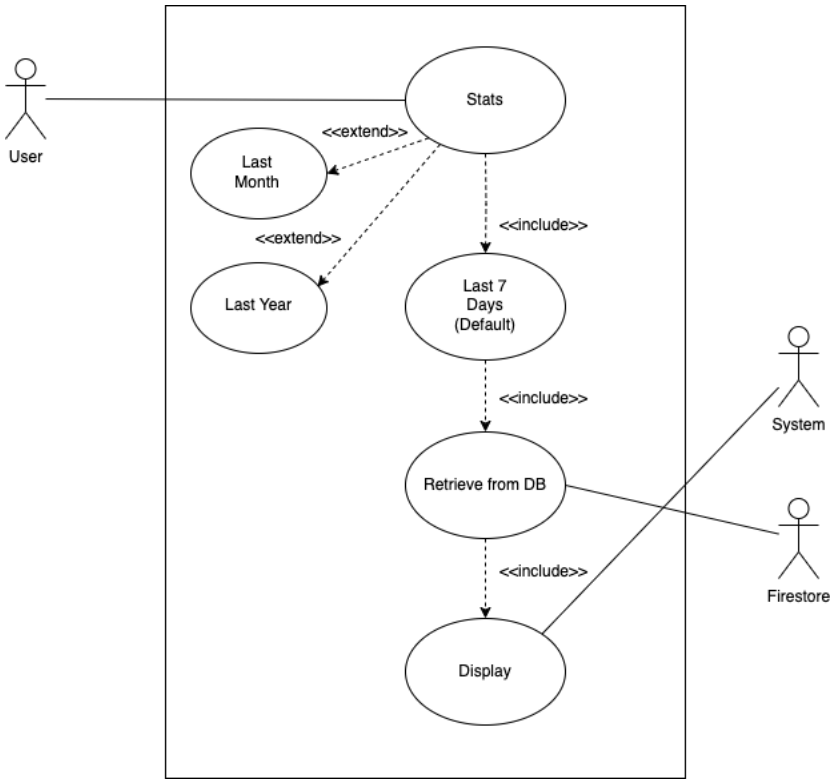
Scope

This use case scope is for showing the stats.

Description

This use case describes the flow of the stats.

Use Case Diagram



Flow Description

Precondition

The application runs on the Stats page.

Activation

This use case starts when a user clicks the Stats button.

Main flow

1. The user chooses an option from the drop-down menu.
2. The filtered values are retrieved from DB.
3. The values are displayed.
4. The user can click on the statistic bars to see specific information related to the bar graph.

Alternate flow

A1: The other app menu
1. The user clicks one of the other buttons from the bottom menu.
2. The user is directed to the relevant page clicked.

Exceptional flow

E1: Any other unexpected system errors.
1. The system displays an error message with the reason that the values cannot be retrieved or displayed successfully.
2. The use case continues at position 1 of the main flow.

Termination

The user chooses an option from the drop-down menu, and the values are displayed accordingly and successfully.

Postcondition

Options can be selected each time on the same page.

2.1.1.9. Requirement 9 – Logout

2.1.1.9.1. Description & Priority

This use case allows users to log out.

2.1.1.9.2. Use Case

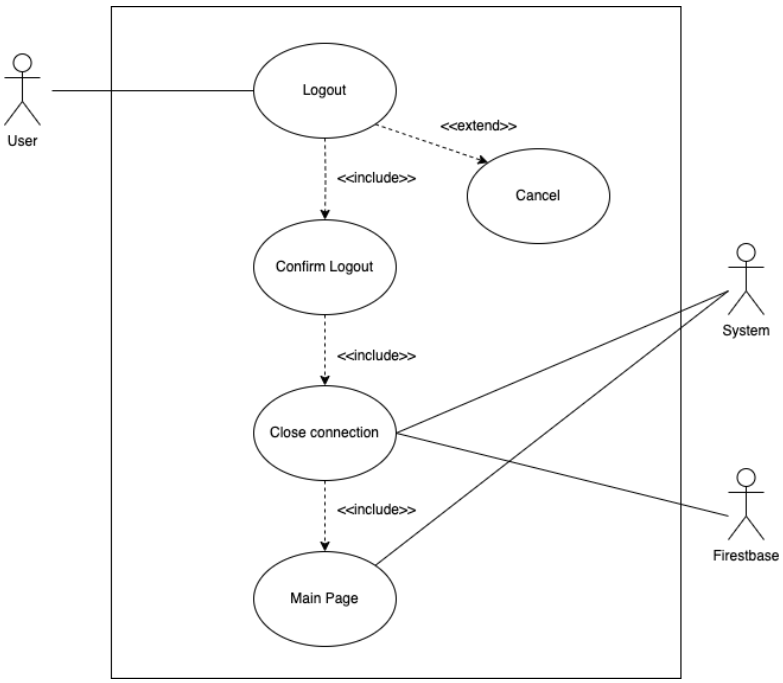
Scope

This use case scope is for logging out from the app.

Description

This use case describes the flow of the logout. This use case has the least priority.

Use Case Diagram



Flow Description

Precondition

The application runs on any page.

Activation

This use case starts when a user clicks the Logout button.

Main flow

1. The user clicks the Logout Button.
2. The Firebase connection is closed.
3. The user is directed to the Main (Login) page.

Alternate flow

A1: Cancel login out
1. The user clicks on anywhere to cancel the Logout process.
2. The user continues to use the app.

Exceptional flow

E1: Any other unexpected system errors.
1. The system displays an error message with the reason that the user cannot log out successfully.
2. The use case continues at position 1 of the main flow.

Termination

The user logs out successfully.

Postcondition

The user logs in again either with the same account or with another account. Also, the user can create a new account.

2.1.2. Data Requirements

The purpose of the application is to create statistics for the user in various ways using the data entered by the user. Each user can only view and edit their own data. Therefore, for users to display statistical data output from the application, they must enter sufficient data for the application to generate statistical output. Since the main focus of the application is to determine the amount of alcohol consumed by the user, the essential data for the application is the type and amount of alcohol consumed. Accordingly, to display the money spent on alcohol consumption, the money spent must also be entered correctly during data entry. For alcohol consumed without spending money, the cost is shown as 0 euros by default. Thus, the aim is to increase the accuracy of the statistical data entered. Another critical piece of information is the location data. If the user enters the location information consistently and in an orderly manner, they will obtain not only the amount of alcohol they consumed and the money they spent but also information on where they consumed the most or least alcohol and spent money on alcohol. This will play an influential role in helping the user review their habits by providing them with self-awareness.

2.1.3. User Requirements

As mentioned in the Data Requirements section, each user can only see, enter, and edit their own data, so a user must first create an account to use the application. The user account consists only of an email and password; no other profile information is required. The application also allows logging in with a Google account. Thus, the application can be accessed and used with the help of a Google account without needing any other registration. Moreover, account creation enhances application security and privacy, as a password is required to enter the application. Additionally, it allows the user to log in from a different device in case of a device change or use the application with multiple devices. Therefore, without exporting/importing app data, users can access their data instantly from any device.

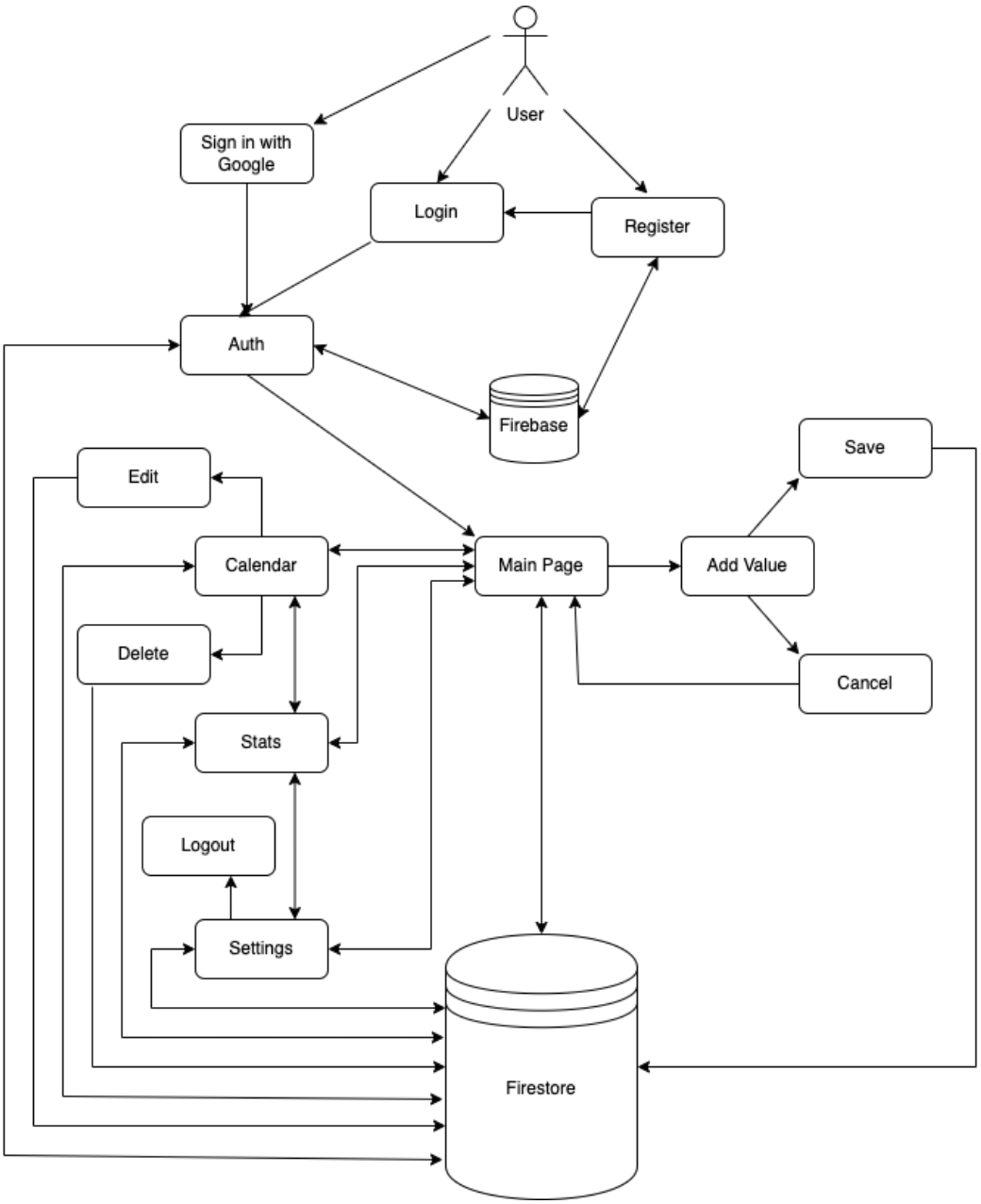
2.1.4. Environmental Requirements

To use the application, the user must have at least a smart device with an Android 14: Upside Down Cake operating system and an internet connection. To create a user account, an active email address, a password that meets the password requirements, and an internet connection are required for the confirmation link to be sent to the email address. Alternatively, a Google account can be used to log in. Additionally, camera access is required for the barcode scanner.

2.1.5. Usability Requirements

The application has a simple design that is easy to use and does not have unnecessary or duplicate menus. In terms of usability requirements, a stable internet connection is required first. In addition, the device on which the application will be run must meet the hardware and software requirements. From a development perspective, Error Handling is used to prevent application crashes or deadlocks. Top of that, since cloud services are used for user authentication and database, the application performance is aimed at the highest level. As mentioned, since the application interface is simple enough, there is no need for any educational or guiding documentation for use.

2.2. Design & Architecture

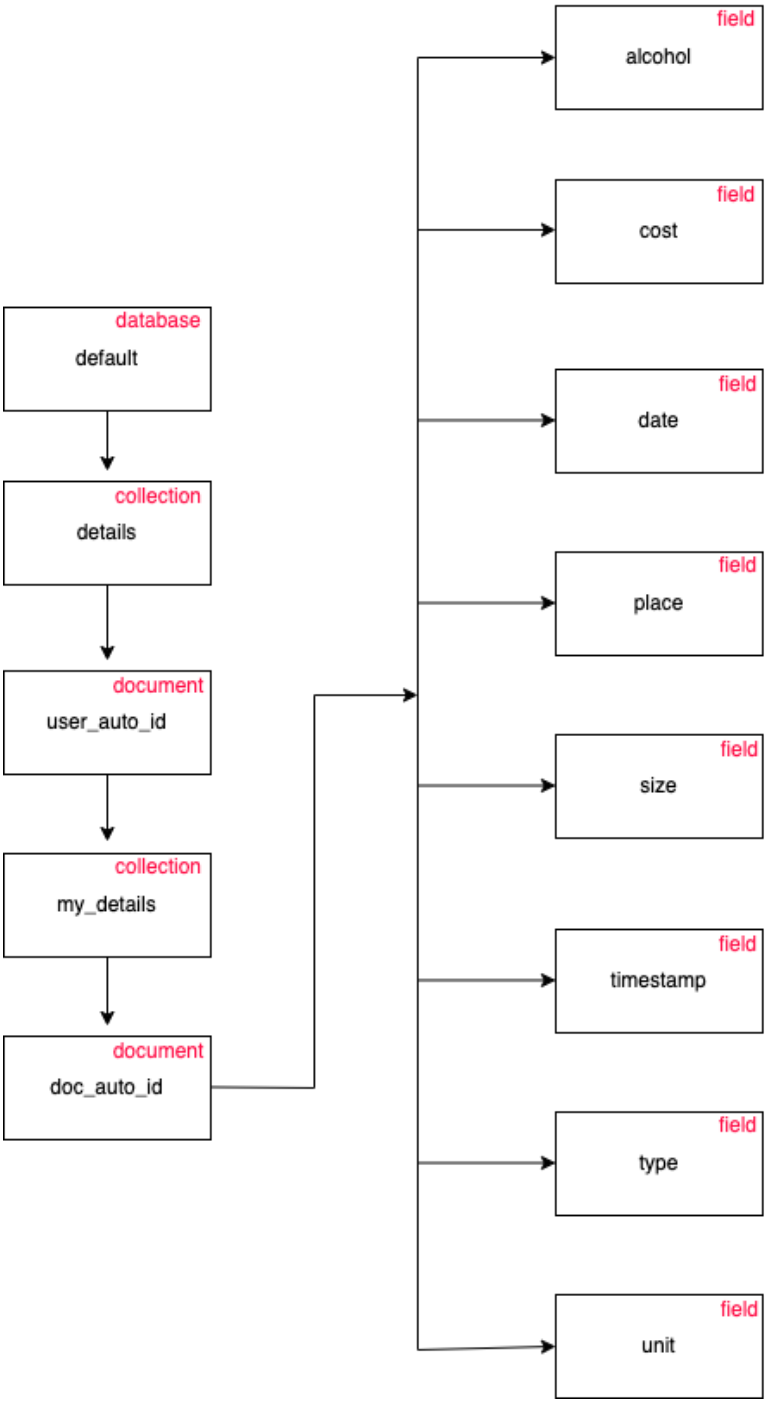


If the user is registered, they can log in or if they are not registered, they can create an account (alternatively, they can log in using a Google account). When the user logs in with registered credentials, authentication is handled by Firebase. If the credentials are verified, the user is directed to the home page. If the user creates a new account, the email is checked by Firebase to see whether it exists in the database. If the entered email address is not registered in the database, it is saved along with the entered password, and a verification link is sent to the email in question by Firebase. Once the user clicks on this link, their account is activated and ready to log in.

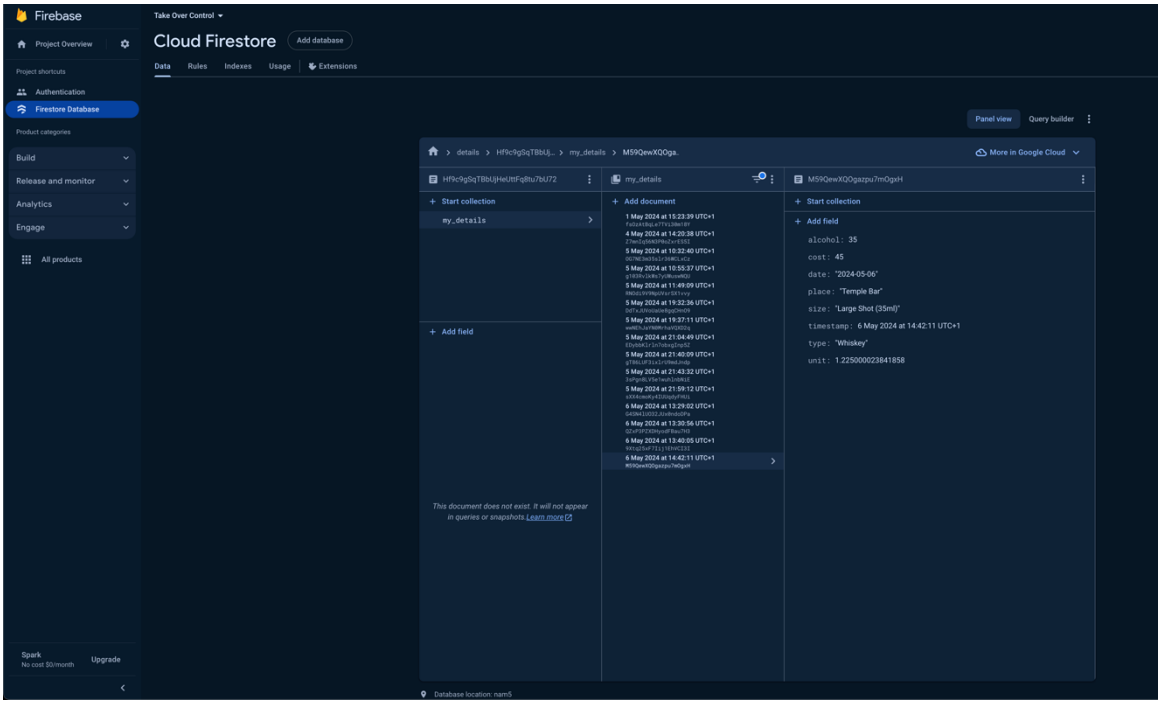
On the home page, the user sees the statistics of the day and the statistics of the last seven days from the database. Here, they can add data using the "Add" or "Quick Add" buttons or access other pages using the Calendar, Stats, and Settings buttons at the bottom. When the "Add" button is clicked, the user selects the desired data from the dropdown menus, enters the data into the text fields, clicks the "Save" button, and the data is saved in the database. Users can also add data using the "Quick Add" page. Here, the user can either choose a brand or scan a barcode, allowing them to add data quickly. Alternatively, they can return to the main page by pressing the "Cancel" button or by clicking the "Home" button. They can also switch to other pages using the buttons at the bottom.

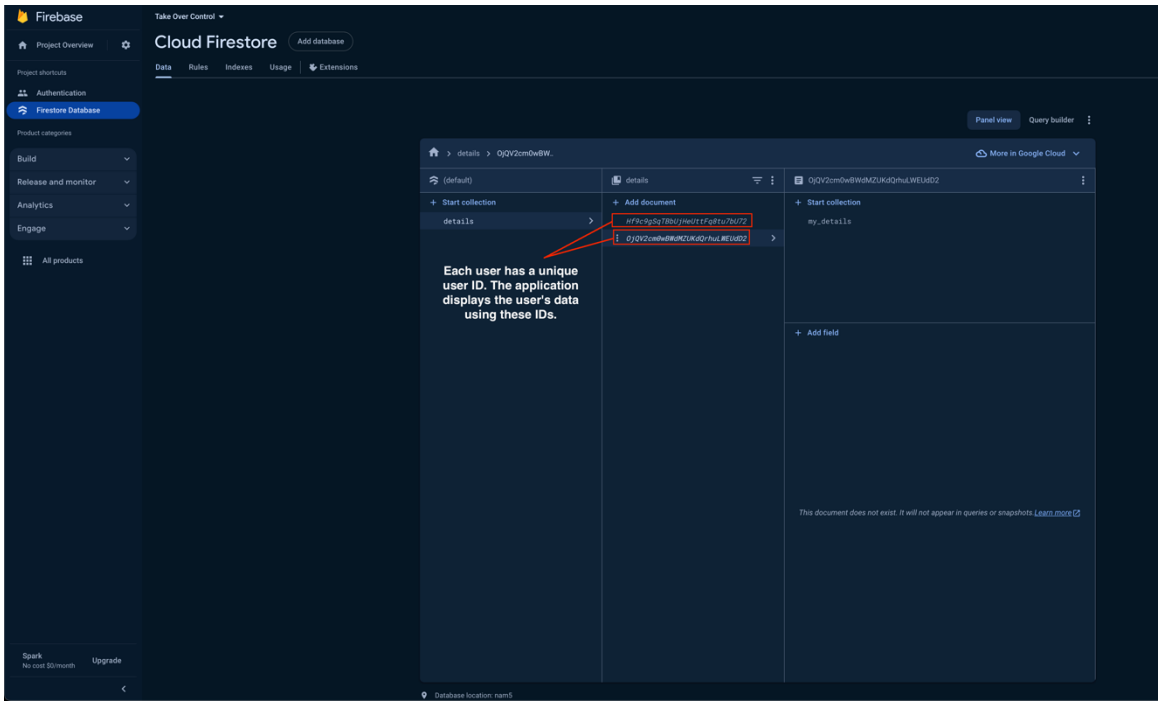
On the calendar page, the user encounters a calendar view from which they can select any date. A query is created from the database based on the selected date, and the relevant data is listed. Any data selected from the list can be edited or deleted. The user can switch between different pages using the buttons at the bottom. On the "Stats" page, the user can create a query filtered according to their desired selection. Lastly, the user logs out of their account by clicking "Logout" under the "Settings" button.

The database structure is as shown in the table below, with a general collection called "details" under the default database. Within this collection are documents created specifically for each user. For each user, there is another collection called "my_details" under the document with user_id, and the data created by the users is saved under this collection with a unique document ID.

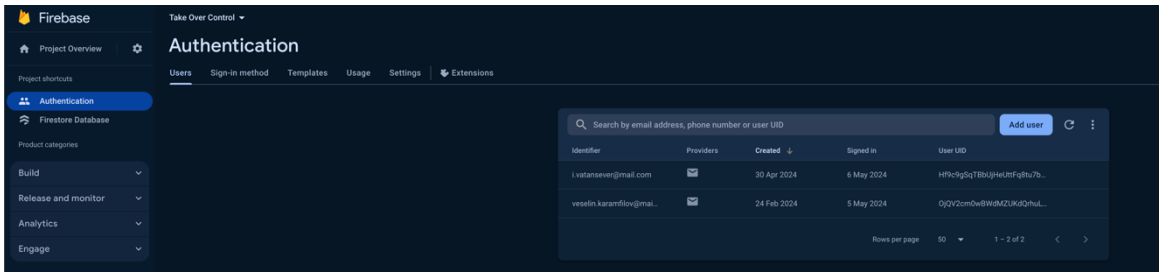

database
default
collection
details
document
user_auto_id
collection
my_details
document
doc_auto_id
field
alcohol
field
cost
field
date
field
place
field
size
field
timestamp
field
type
field
unit

A sample document recorded in the database can be seen from the screenshots below.





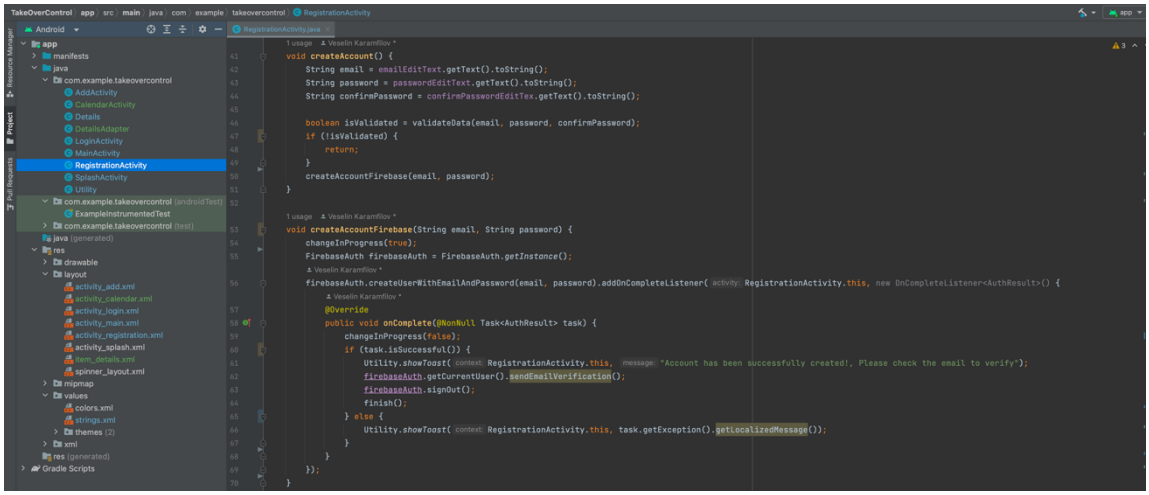
Plus, below is the screenshot showing how registered users stored are in Firebase.



2.3. Implementation

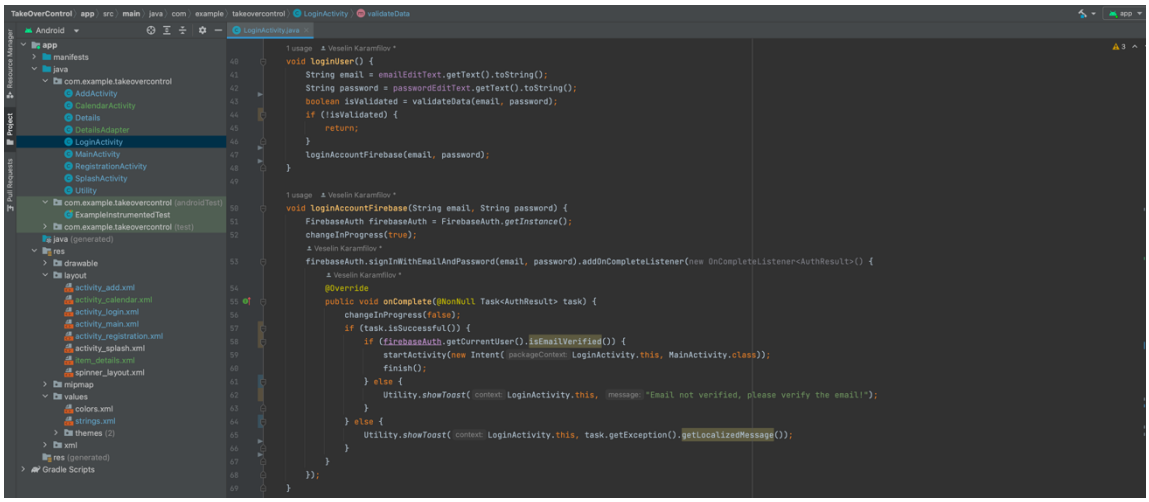
Screenshots of some code snippets as examples of implementation can be found below.

A snippet from the Registration activity:



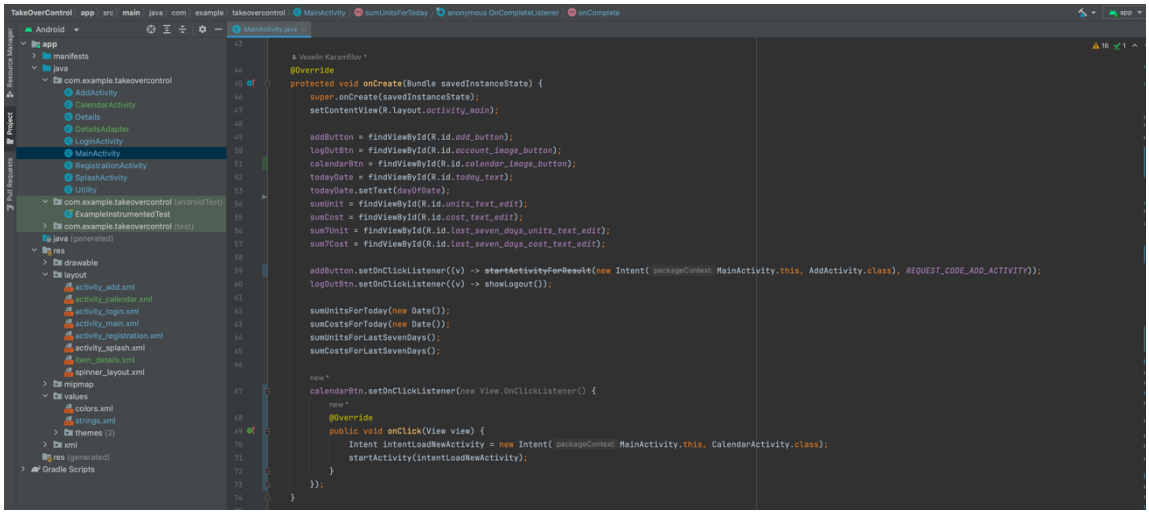
```
void createAccount() {
    String email = emailEditText.getText().toString();
    String password = passwordEditText.getText().toString();
    String confirmPassword = confirmPasswordEditTex.getText().toString();

    boolean isValidated = validateData(email, password, confirmPassword);
    if (!isValidated) {
        return;
    }
    createAccountFirebase(email, password);
}

void createAccountFirebase(String email, String password) {
    changeInProgress(true);
    FirebaseAuth firebaseAuth = FirebaseAuth.getInstance();
    firebaseAuth.createUserWithEmailAndPassword(email, password).addOnCompleteListener( activity: RegistrationActivity.this, new OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {
            changeInProgress(false);
            if (task.isSuccessful()) {
                Utility.showToast( context: RegistrationActivity.this,  message: "Account has been successfully created!, Please check the email to verify");
                firebaseAuth.getCurrentUser().sendEmailVerification();
                firebaseAuth.signOut();
                finish();
            } else {
                Utility.showToast( context: RegistrationActivity.this, task.getException().getLocalizedMessage());
            }
        }
    });
}
```

A snippet from the Login activity:

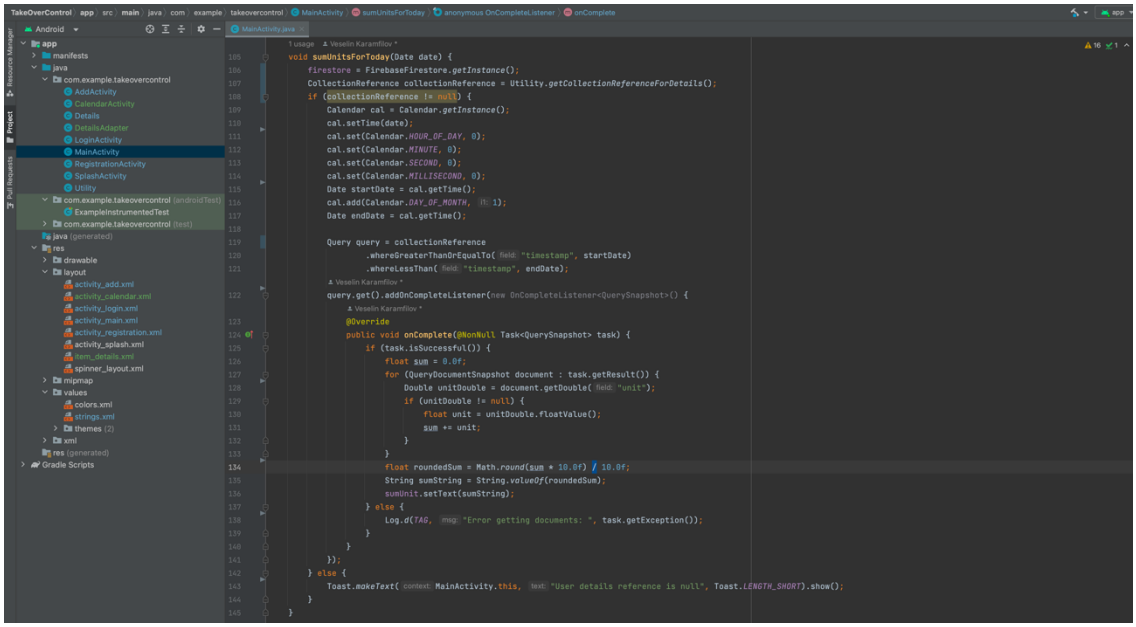


```
void loginUser() {
    String email = emailEditText.getText().toString();
    String password = passwordEditText.getText().toString();
    boolean isValidated = validateData(email, password);
    if (!isValidated) {
        return;
    }
    loginAccountFirebase(email, password);
}

void loginAccountFirebase(String email, String password) {
    FirebaseAuth firebaseAuth = FirebaseAuth.getInstance();
    changeInProgress(true);
    firebaseAuth.signInWithEmailAndPassword(email, password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {
            changeInProgress(false);
            if (task.isSuccessful()) {
                if (firebaseAuth.getCurrentUser().isEmailVerified()) {
                    startActivity(new Intent( packageContext: LoginActivity.this, MainActivity.class));
                    finish();
                } else {
                    Utility.showToast( context: LoginActivity.this,  message: "Email not verified, please verify the email!");
                }
            } else {
                Utility.showToast( context: LoginActivity.this, task.getException().getLocalizedMessage());
            }
        }
    });
}
```

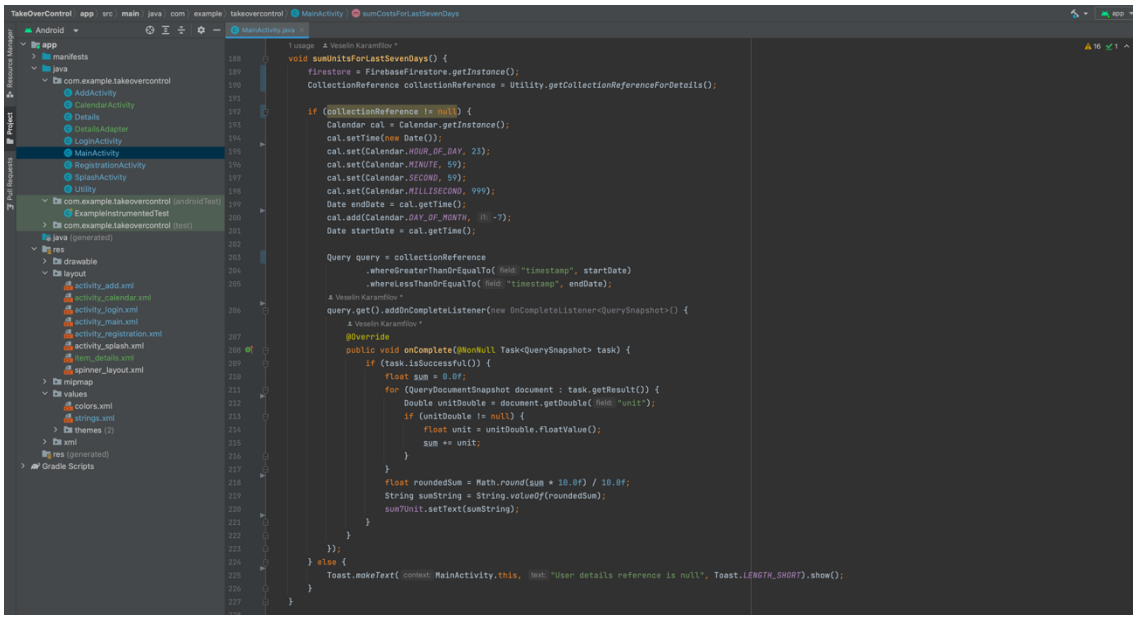
Snippets from the Main activity:



```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    addButton = findViewById(R.id.add_button);
    logOutBtn = findViewById(R.id.account_image_button);
    calendarBtn = findViewById(R.id.calendar_image_button);
    todayDate = findViewById(R.id.today_text);
    todayDate.setText(dayOfDate);
    sumUnit = findViewById(R.id.units_text_edit);
    sumCost = findViewById(R.id.cost_text_edit);
    sum7Unit = findViewById(R.id.last_seven_days_units_text_edit);
    sum7Cost = findViewById(R.id.last_seven_days_cost_text_edit);

    addButton.setOnClickListener((v) -> startActivityForResult(new Intent( packageContext: MainActivity.this, AddActivity.class), REQUEST_CODE_ADD_ACTIVITY));
    logOutBtn.setOnClickListener((v) -> showLogout());

    sumUnitsForToday(new Date());
    sumCostsForToday(new Date());
    sumUnitsForLastSevenDays();
    sumCostsForLastSevenDays();

    new *
    calendarBtn.setOnClickListener(new View.OnClickListener() {
        new *
        @Override
        public void onClick(View view) {
            Intent intentLoadNewActivity = new Intent( packageContext: MainActivity.this, CalendarActivity.class);
            startActivity(intentLoadNewActivity);
        }
    });
}
```

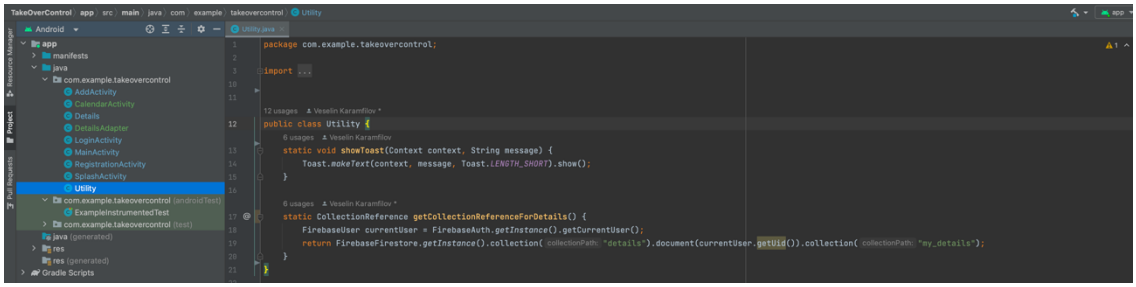


```
void sumUnitsForToday(Date date) {
    firestore = FirebaseFirestore.getInstance();
    CollectionReference collectionReference = Utility.getCollectionReferenceForDetails();
    if (collectionReference != null) {
        Calendar cal = Calendar.getInstance();
        cal.setTime(date);
        cal.set(Calendar.HOUR_OF_DAY, 0);
        cal.set(Calendar.MINUTE, 0);
        cal.set(Calendar.SECOND, 0);
        cal.set(Calendar.MILLISECOND, 0);
        Date startDate = cal.getTime();
        cal.add(Calendar.DAY_OF_MONTH, 1);
        Date endDate = cal.getTime();

        Query query = collectionReference
                .whereGreaterThanOrEqualTo("timestamp", startDate)
                .whereLessThan("timestamp", endDate);
        query.get().addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {
            @Override
            public void onComplete(@NonNull Task<QuerySnapshot> task) {
                if (task.isSuccessful()) {
                    float sum = 0.0f;
                    for (QueryDocumentSnapshot document : task.getResult()) {
                        Double unitDouble = document.getDouble("unit");
                        if (unitDouble != null) {
                            float unit = unitDouble.floatValue();
                            sum += unit;
                        }
                    }
                    float roundedSum = Math.round(sum * 10.0f) / 10.0f;
                    String sumString = String.valueOf(roundedSum);
                    sumUnit.setText(sumString);
                } else {
                    Log.d(TAG, "Error getting documents: ", task.getException());
                }
            }
        });
    } else {
        Toast.makeText(MainActivity.this, "User details reference is null", Toast.LENGTH_SHORT).show();
    }
}
```



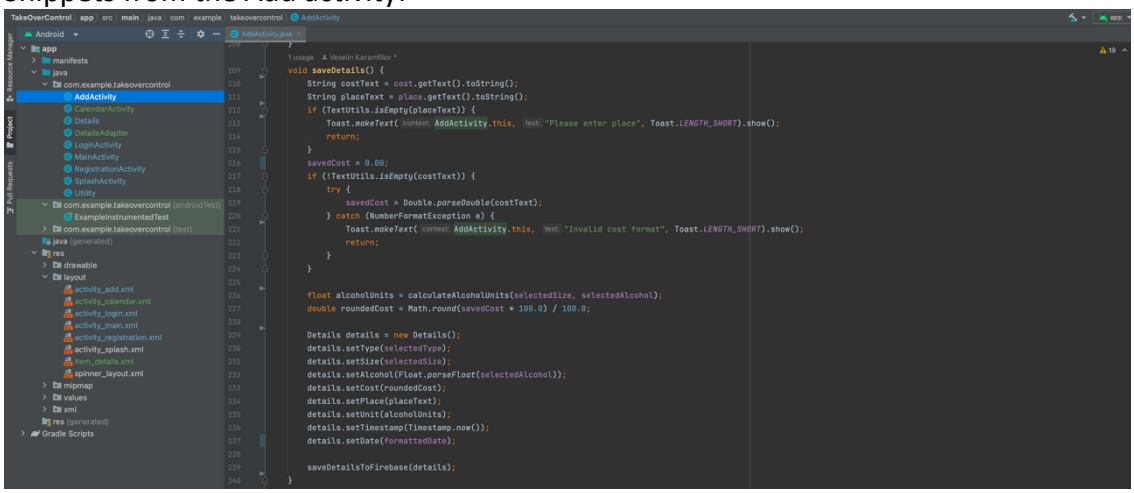
```
void sumUnitsForLastSevenDays() {
    firestore = FirebaseFirestore.getInstance();
    CollectionReference collectionReference = Utility.getCollectionReferenceForDetails();

    if (collectionReference != null) {
        Calendar cal = Calendar.getInstance();
        cal.setTime(new Date());
        cal.set(Calendar.HOUR_OF_DAY, 23);
        cal.set(Calendar.MINUTE, 59);
        cal.set(Calendar.SECOND, 59);
        cal.set(Calendar.MILLISECOND, 999);
        Date endDate = cal.getTime();
        cal.add(Calendar.DAY_OF_MONTH, -7);
        Date startDate = cal.getTime();

        Query query = collectionReference
                .whereGreaterThanOrEqualTo("timestamp", startDate)
                .whereLessThan("timestamp", endDate);
        query.get().addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {
            @Override
            public void onComplete(@NonNull Task<QuerySnapshot> task) {
                if (task.isSuccessful()) {
                    float sum = 0.0f;
                    for (QueryDocumentSnapshot document : task.getResult()) {
                        Double unitDouble = document.getDouble("unit");
                        if (unitDouble != null) {
                            float unit = unitDouble.floatValue();
                            sum += unit;
                        }
                    }
                    float roundedSum = Math.round(sum * 10.0f) / 10.0f;
                    String sumString = String.valueOf(roundedSum);
                    sum7Unit.setText(sumString);
                }
            }
        });
    } else {
        Toast.makeText(MainActivity.this, "User details reference is null", Toast.LENGTH_SHORT).show();
    }
}
```

A snippet from the Utility activity:



```
package com.example.takeovercontrol;

import ...


public class Utility {
    static void showToast(Context context, String message) {
        Toast.makeText(context, message, Toast.LENGTH_SHORT).show();
    }

    static CollectionReference getCollectionReferenceForDetails() {
        FirebaseUser currentUser = FirebaseAuth.getInstance().getCurrentUser();
        return FirebaseFirestore.getInstance().collection("details").document(currentUser.getUid()).collection("my_details");
    }
}
```

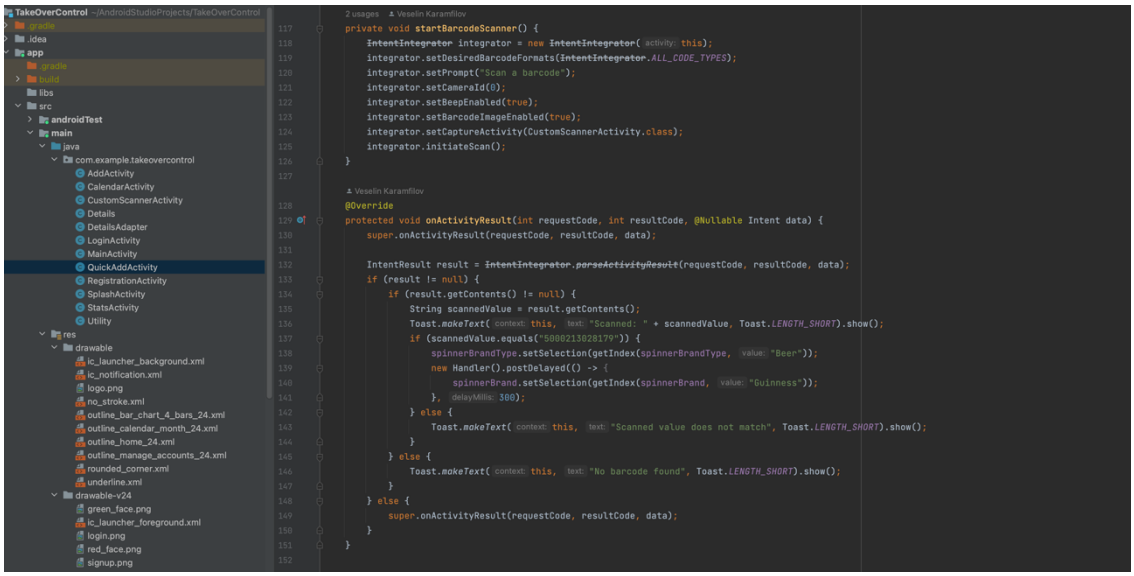
Snippets from the Add activity:



```
1 usage  Veselin Karamfilov *
void saveDetails() {
    String costText = cost.getText().toString();
    String placeText = place.getText().toString();
    if (TextUtils.isEmpty(placeText)) {
        Toast.makeText( context: AddActivity.this,  text: "Please enter place", Toast.LENGTH_SHORT).show();
        return;
    }
    savedCost = 0.00;
    if (!TextUtils.isEmpty(costText)) {
        try {
            savedCost = Double.parseDouble(costText);
        } catch (NumberFormatException e) {
            Toast.makeText( context: AddActivity.this,  text: "Invalid cost format", Toast.LENGTH_SHORT).show();
            return;
        }
    }

    float alcoholUnits = calculateAlcoholUnits(selectedSize, selectedAlcohol);
    double roundedCost = Math.round(savedCost * 100.0) / 100.0;

    Details details = new Details();
    details.setType(selectedType);
    details.setSize(selectedSize);
    details.setAlcohol(Float.parseFloat(selectedAlcohol));
    details.setCost(roundedCost);
    details.setPlace(placeText);
    details.setUnit(alcoholUnits);
    details.setTimestamp(Timestamp.now());
    details.setDate(formattedDate);

    saveDetailsToFirebase(details);
}
```

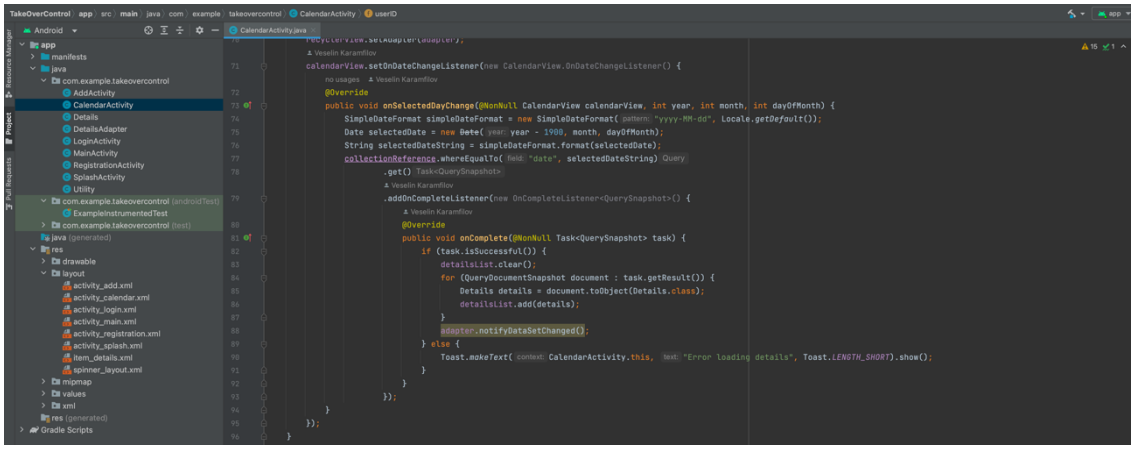
```
1 usage  Veselin Karamfilov *
void saveDetailsToFirebase(Details details) {
    DocumentReference documentReference;
    documentReference = Utility.getCollectionReferenceForDetails().document();
    Veselin Karamfilov *
    documentReference.set(details).addOnCompleteListener(new OnCompleteListener<Void>() {
        Veselin Karamfilov *
        @Override
        public void onComplete(@NonNull Task<Void> task) {
            if (task.isSuccessful()) {
                Utility.showToast( context: AddActivity.this,  message: "Details saved successfully");
                setResult(RESULT_OK);
            } else {
                Utility.showToast( context: AddActivity.this,  message: "Details couldn't be saved");
                setResult(RESULT_CANCELED);
            }
            finish();
        }
    });
}
```

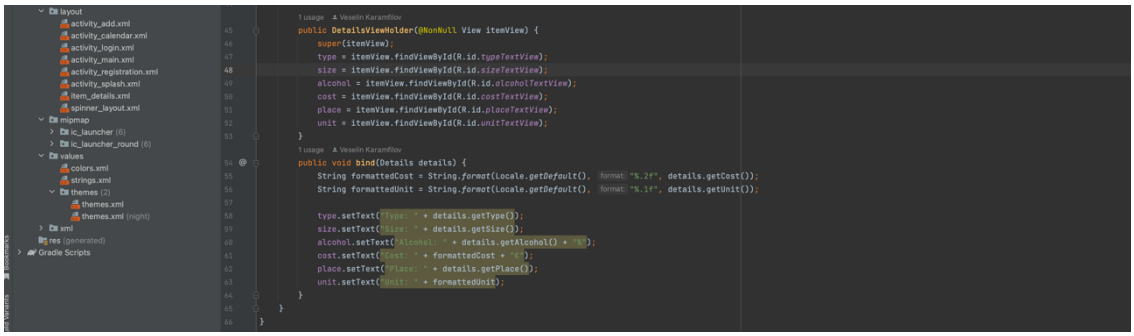
Snippets from the Quick Add activity:



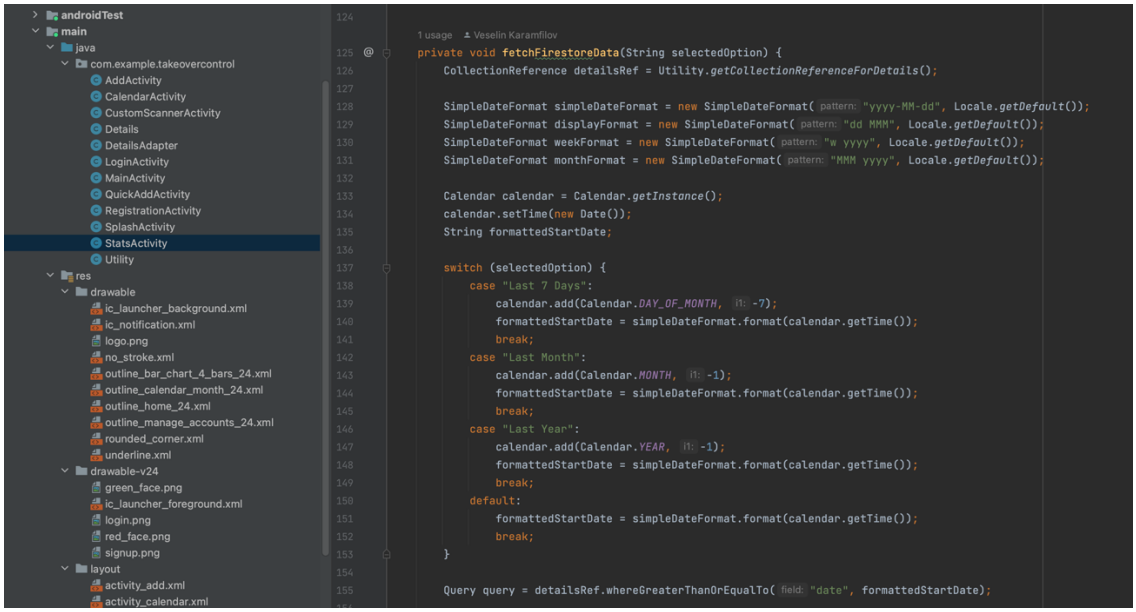
```
2 usages  Veselin Karamfilov
private void startBarcodeScanner() {
    IntentIntegrator integrator = new IntentIntegrator( activity: this);
    integrator.setDesiredBarcodeFormats(IntentIntegrator.ALL_CODE_TYPES);
    integrator.setPrompt("Scan a barcode");
    integrator.setCameraId(0);
    integrator.setBeepEnabled(true);
    integrator.setBarcodeImageEnabled(true);
    integrator.setCaptureActivity(CustomScannerActivity.class);
    integrator.initiateScan();
}

Veselin Karamfilov
@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    IntentResult result = IntentIntegrator.parseActivityResult(requestCode, resultCode, data);
    if (result != null) {
        if (result.getContents() != null) {
            String scannedValue = result.getContents();
            Toast.makeText( context: this,  text: "Scanned: " + scannedValue, Toast.LENGTH_SHORT).show();
            if (scannedValue.equals("5000213028179")) {
                spinnerBrandType.setSelection(getIndex(spinnerBrandType,  value: "Beer"));
                new Handler().postDelayed(() -> {
                    spinnerBrand.setSelection(getIndex(spinnerBrand,  value: "Guinness"));
                },  delayMillis: 300);
            } else {
                Toast.makeText( context: this,  text: "Scanned value does not match", Toast.LENGTH_SHORT).show();
            }
        } else {
            Toast.makeText( context: this,  text: "No barcode found", Toast.LENGTH_SHORT).show();
        }
    } else {
        super.onActivityResult(requestCode, resultCode, data);
    }
}
```

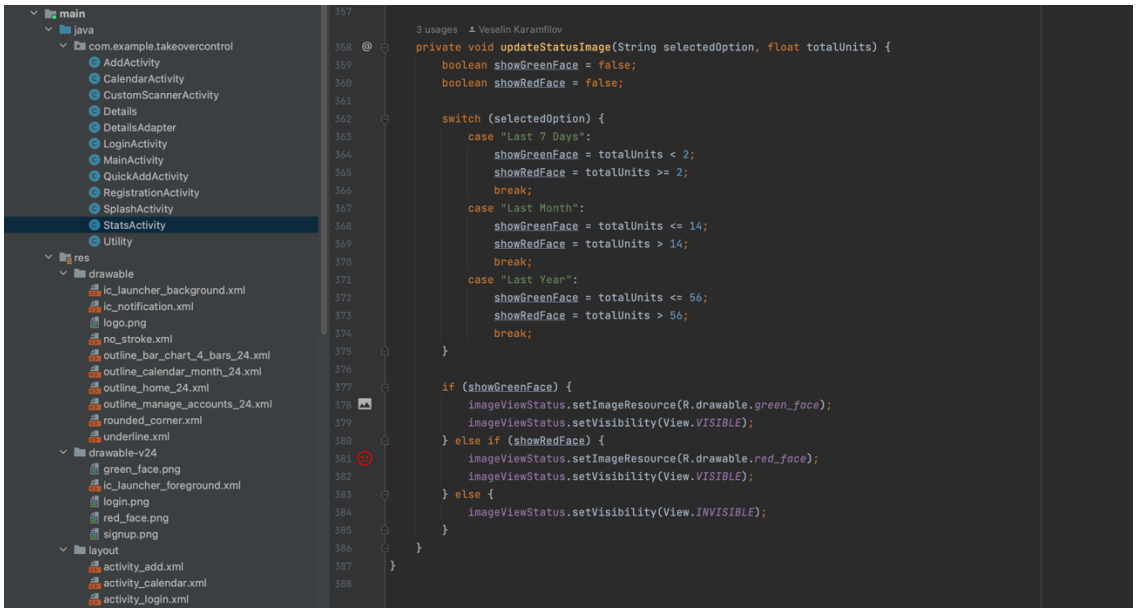
Snippets from the Calendar activity:





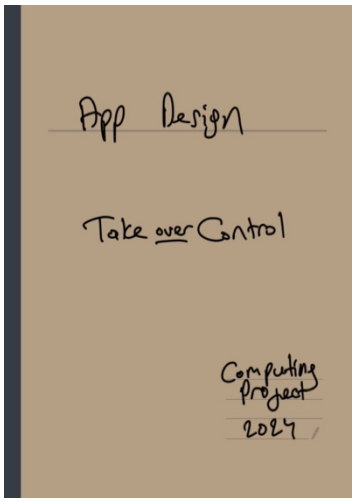
Snippets from the Stats activity:

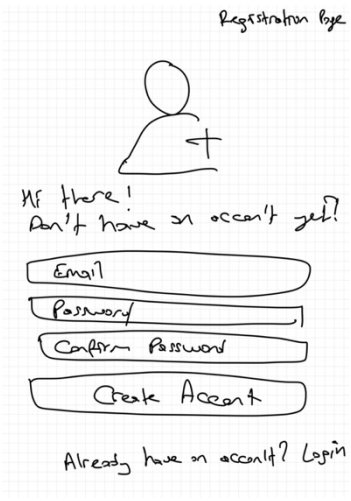


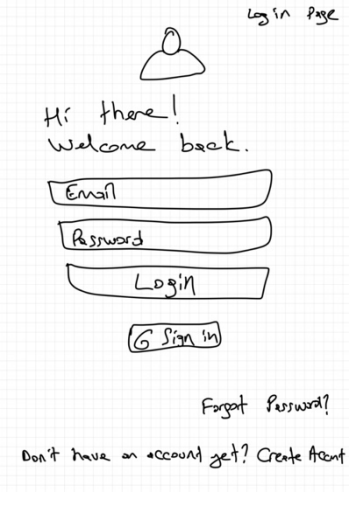


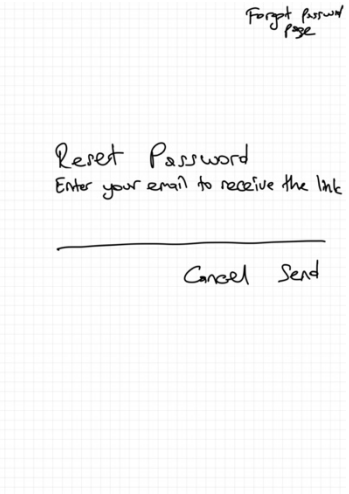
2.4. Graphical User Interface (GUI)

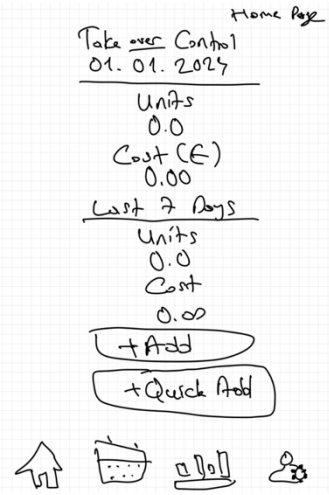
Wireframe Design of the Application

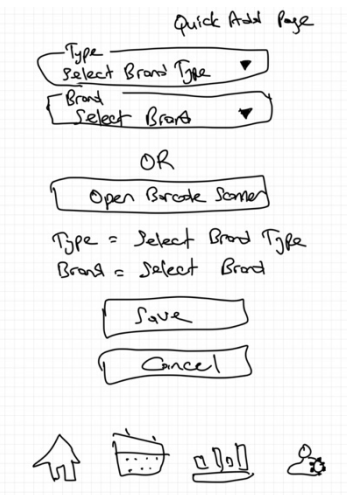


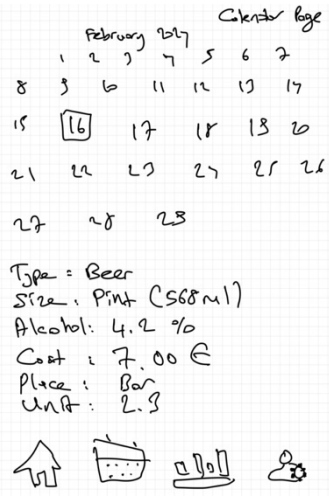


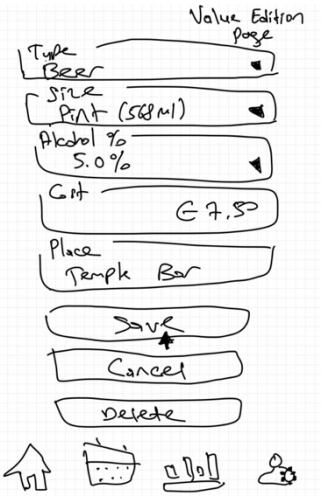


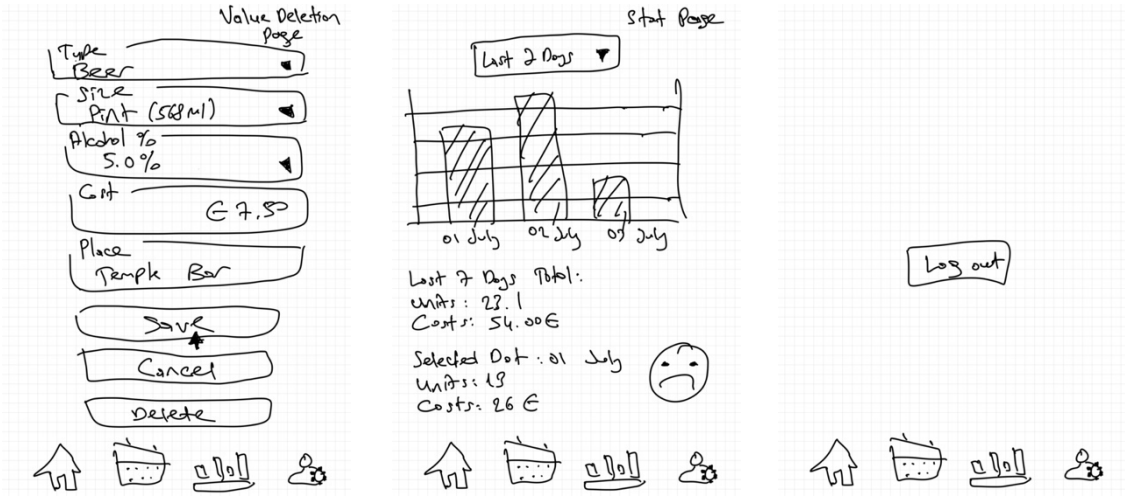




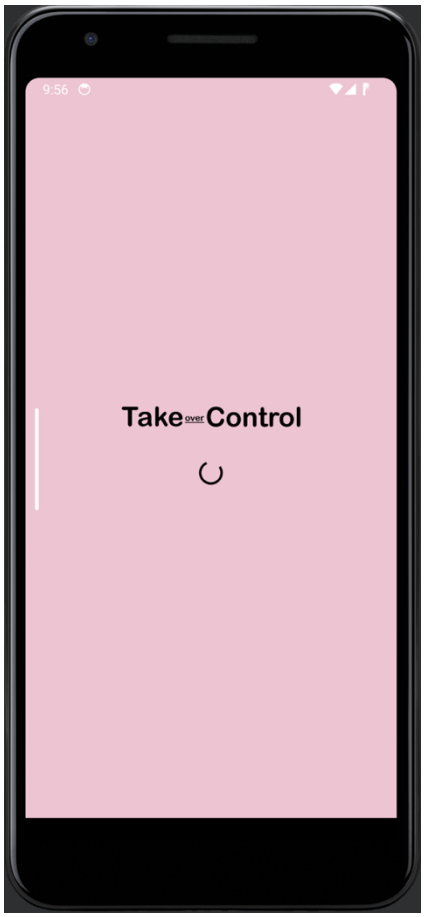


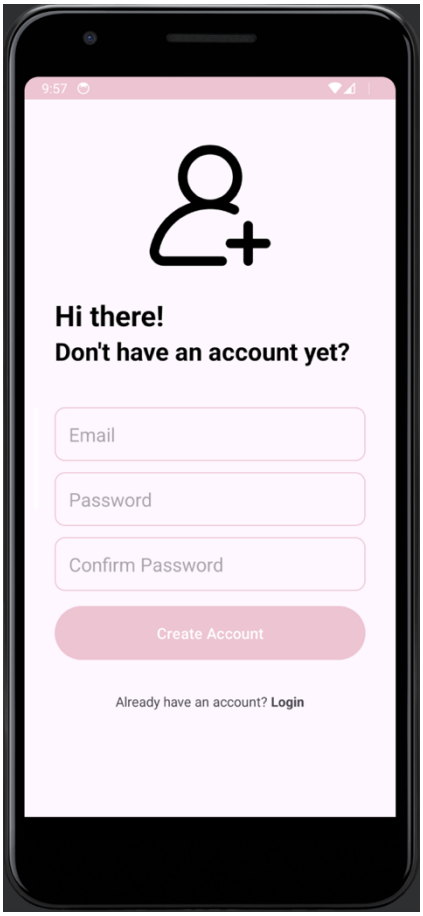


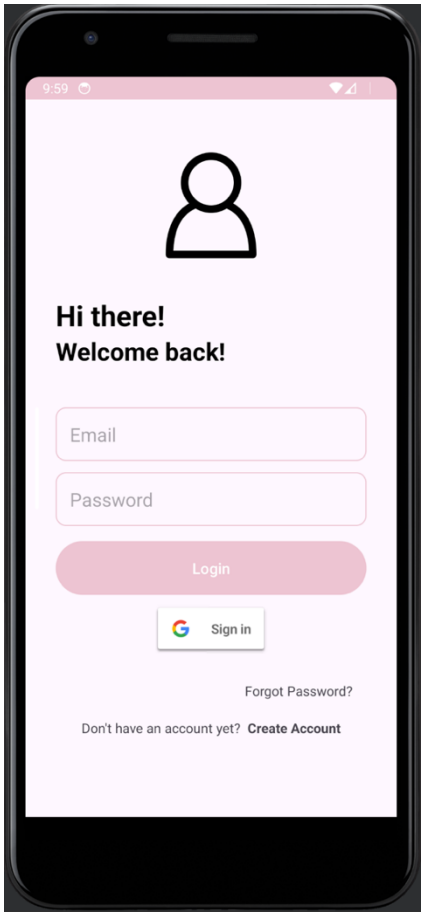


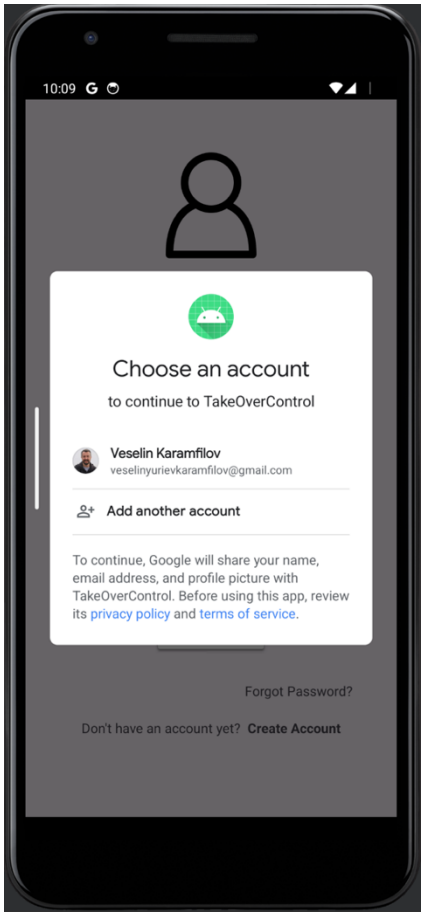


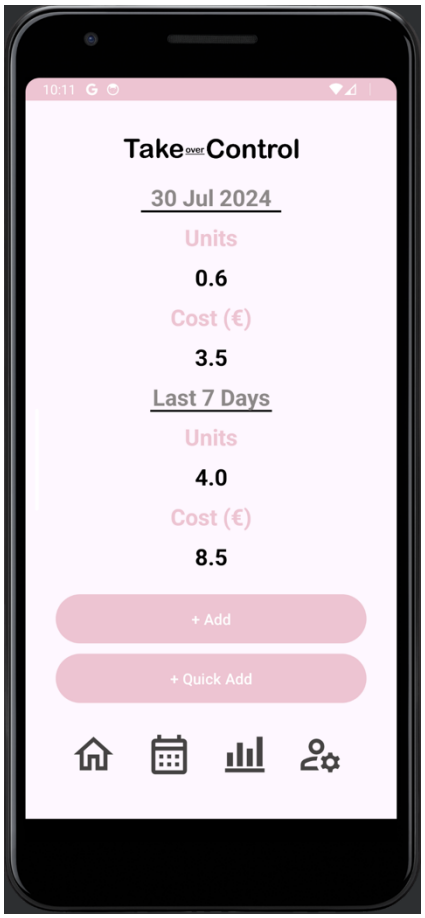
Actual Design of the Application

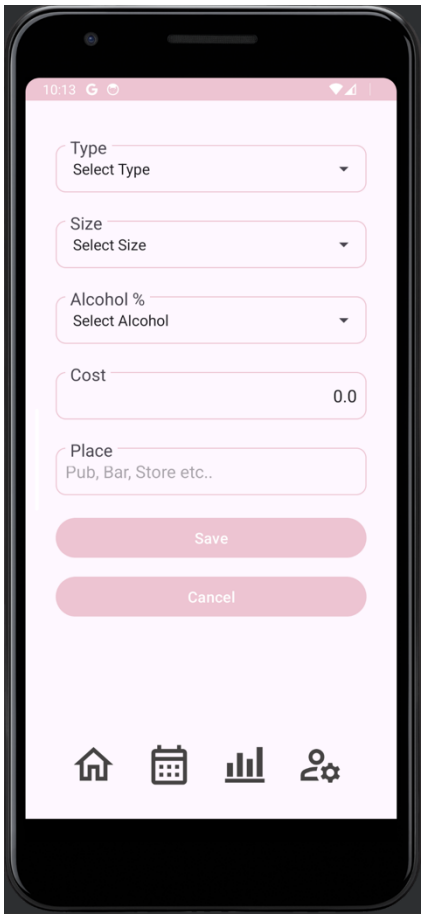


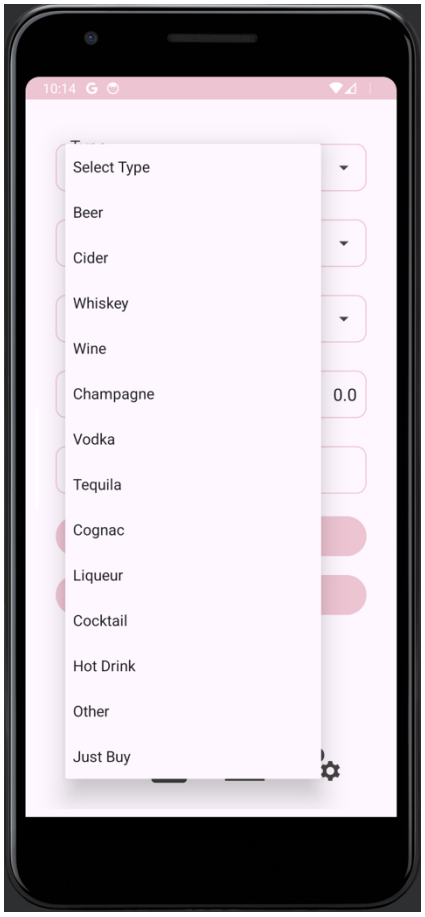


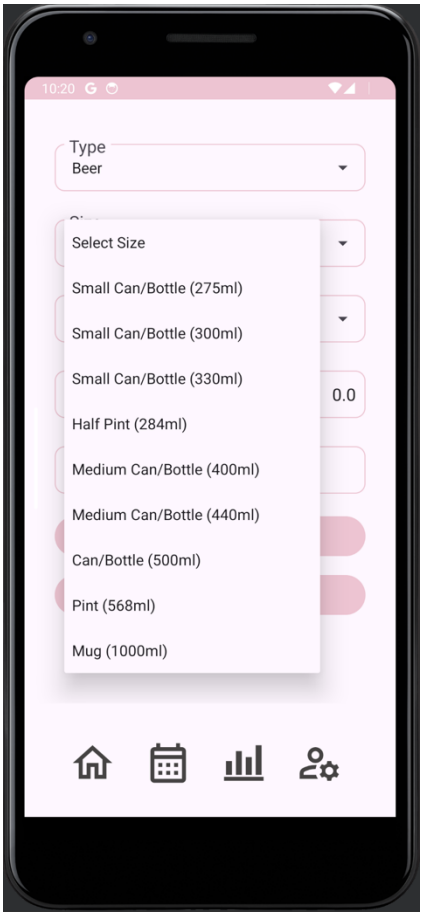

9:59
Hi there!
Welcome back!
Email
Password
Login
Sign in
Forgot Password?
Don't have an account yet? Create Account

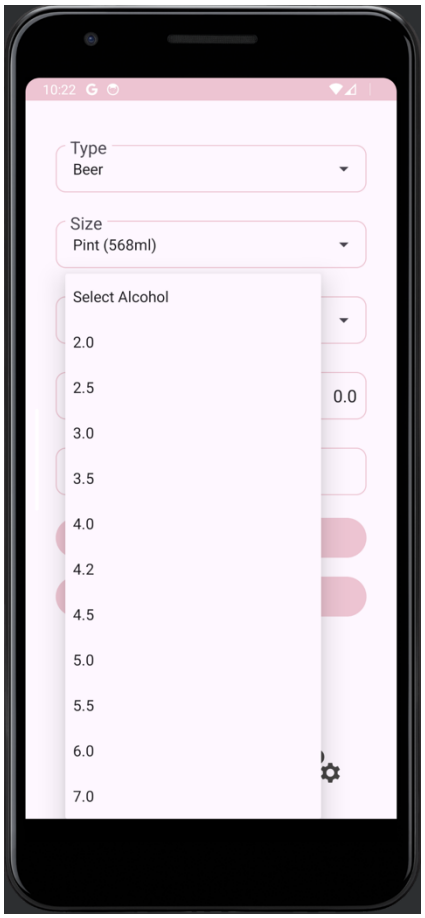

10:09
Choose an account
to continue to TakeOverControl
Veselin Karamfilov
veselinyurievkaramfilov@gmail.com
Add another account
To continue, Google will share your name, email address, and profile picture with TakeOverControl. Before using this app, review its privacy policy and terms of service.
Forgot Password?
Don't have an account yet? Create Account

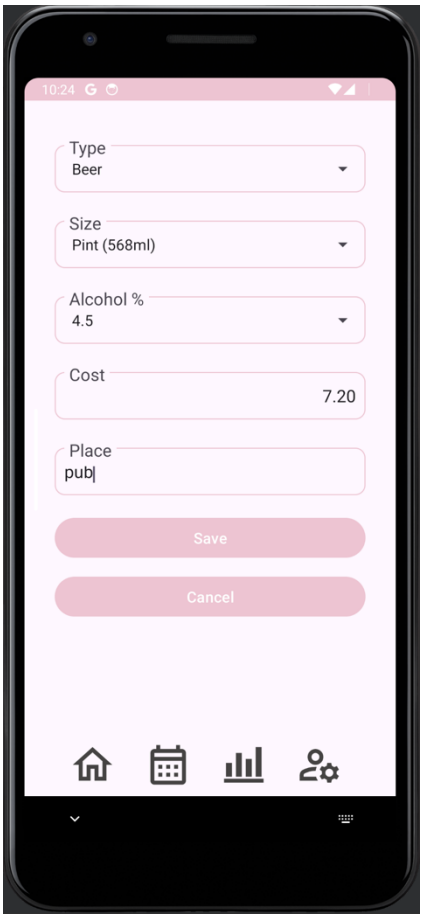

10:11
TakeoverControl
30 Jul 2024
Units
0.6
Cost (€)
3.5
Last 7 Days
Units
4.0
Cost (€)
8.5
+ Add
+ Quick Add

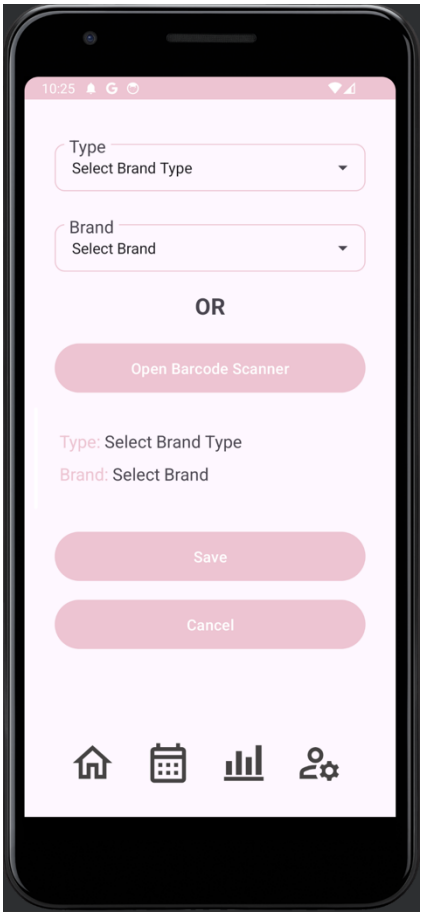

10:13
Type
Select Type
Size
Select Size
Alcohol %
Select Alcohol
Cost
0.0
Place
Pub, Bar, Store etc..
Save
Cancel

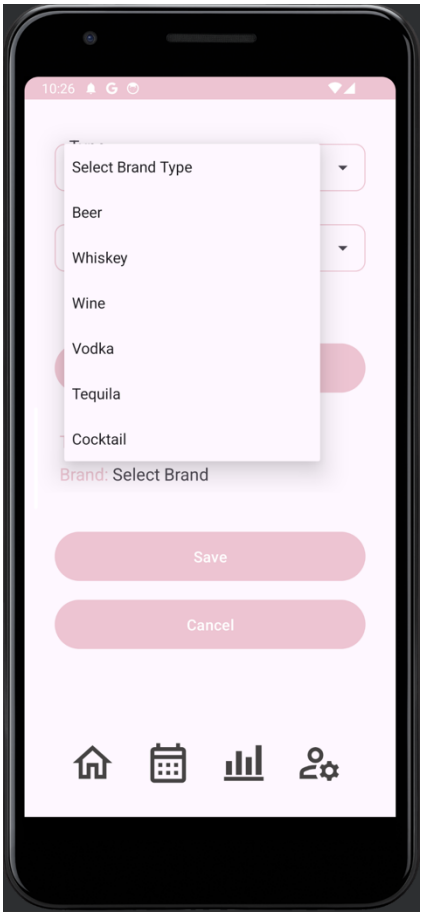

10:14
Select Type
Beer
Cider
Whiskey
Wine
Champagne
Vodka
Tequila
Cognac
Liqueur
Cocktail
Hot Drink
Other
Just Buy
0.0

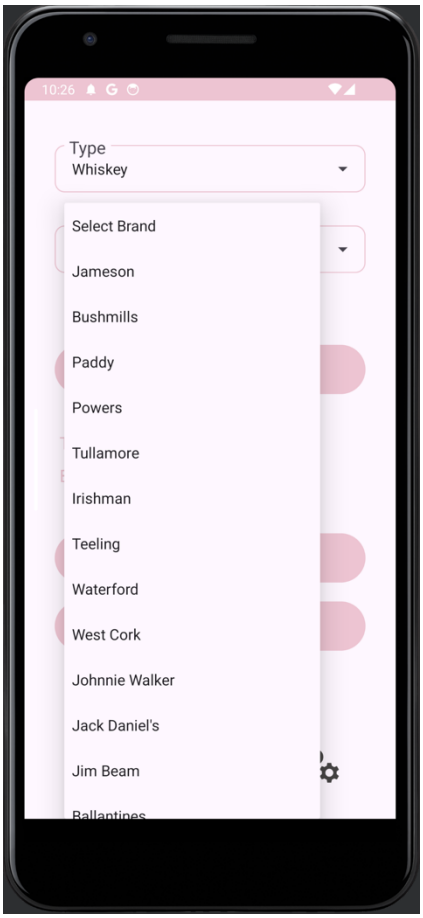

10:20
Type
Beer
Select Size
Small Can/Bottle (275ml)
Small Can/Bottle (300ml)
Small Can/Bottle (330ml)
Half Pint (284ml)
Medium Can/Bottle (400ml)
Medium Can/Bottle (440ml)
Can/Bottle (500ml)
Pint (568ml)
Mug (1000ml)
0.0

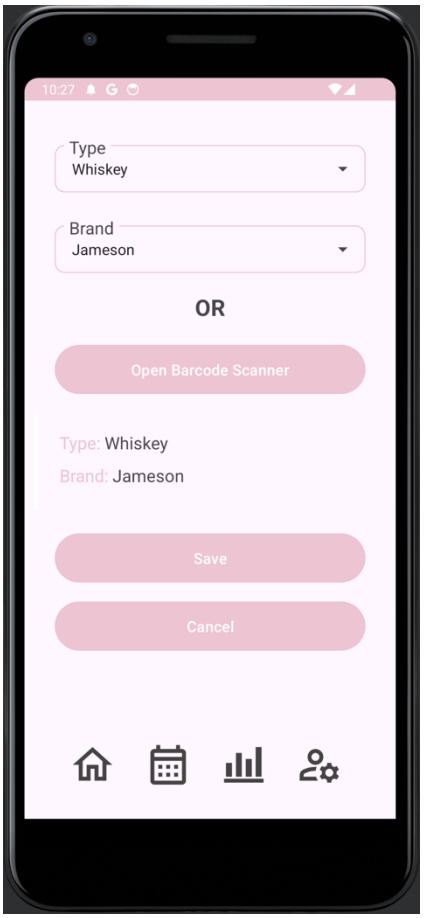

10:22
Type
Beer
Size
Pint (568ml)
Select Alcohol
2.0
2.5
3.0
3.5
4.0
4.2
4.5
5.0
5.5
6.0
7.0
0.0

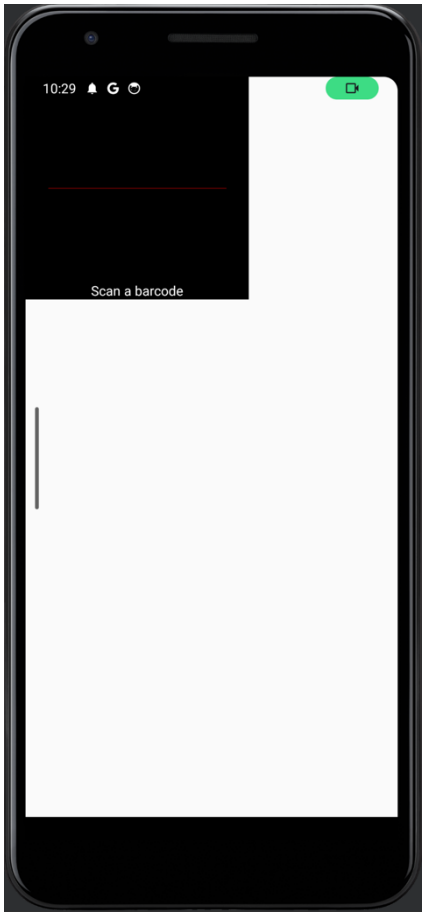

10:24
Type
Beer
Size
Pint (568ml)
Alcohol %
4.5
Cost
7.20
Place
pub
Save
Cancel

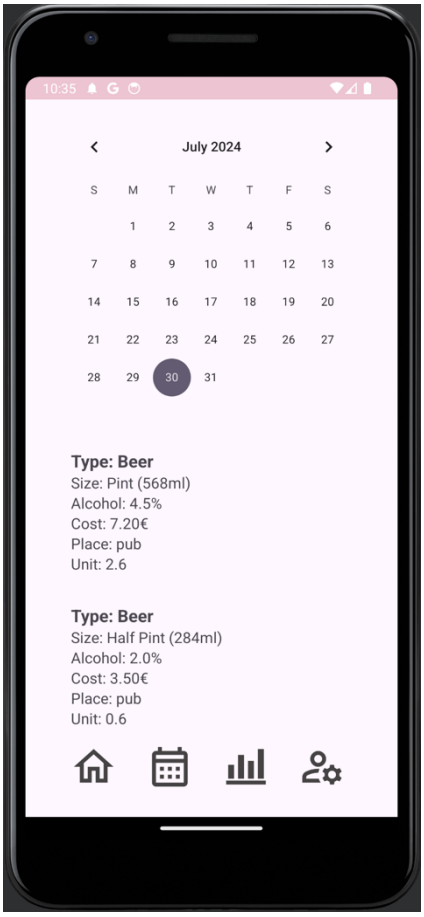

10:25
Type
Select Brand Type
Brand
Select Brand
OR
Open Barcode Scanner
Type: Select Brand Type
Brand: Select Brand
Save
Cancel

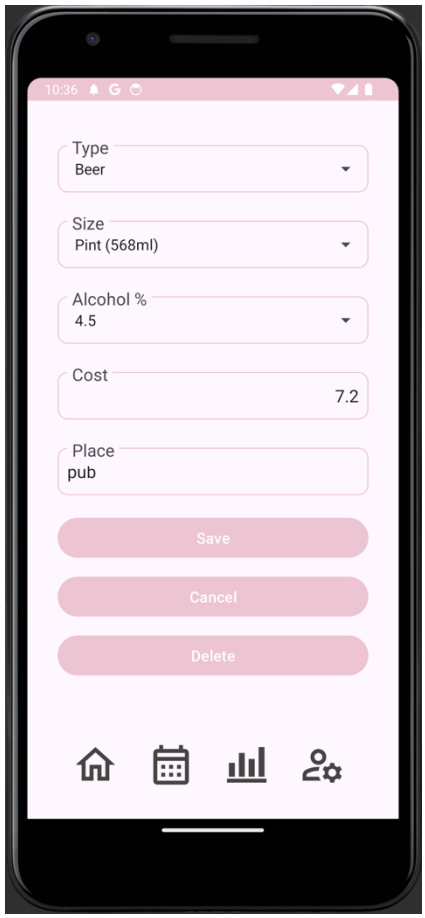

10:26
Select Brand Type
Beer
Whiskey
Wine
Vodka
Tequila
Cocktail
Brand: Select Brand
Save
Cancel


10:26
Type
Whiskey
Select Brand
Jameson
Bushmills
Paddy
Powers
Tullamore
Irishman
Teeling
Waterford
West Cork
Johnnie Walker
Jack Daniel's
Jim Beam

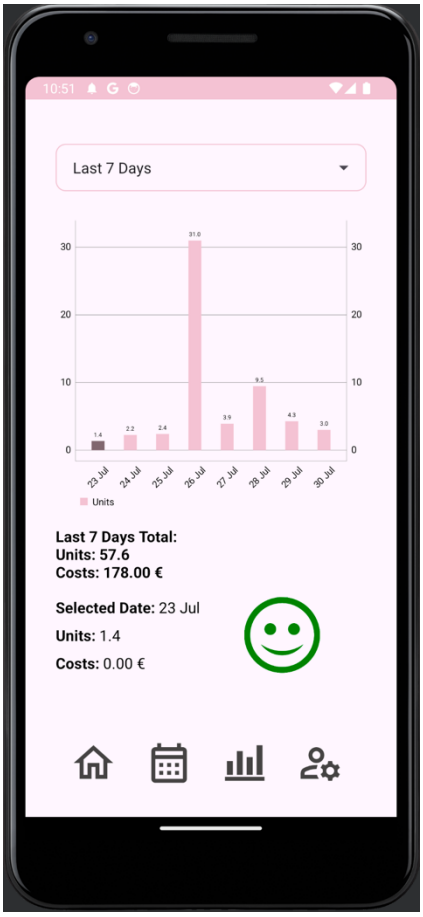

10:27
Type
Whiskey
Brand
Jameson
OR
Open Barcode Scanner
Type: Whiskey
Brand: Jameson
Save
Cancel

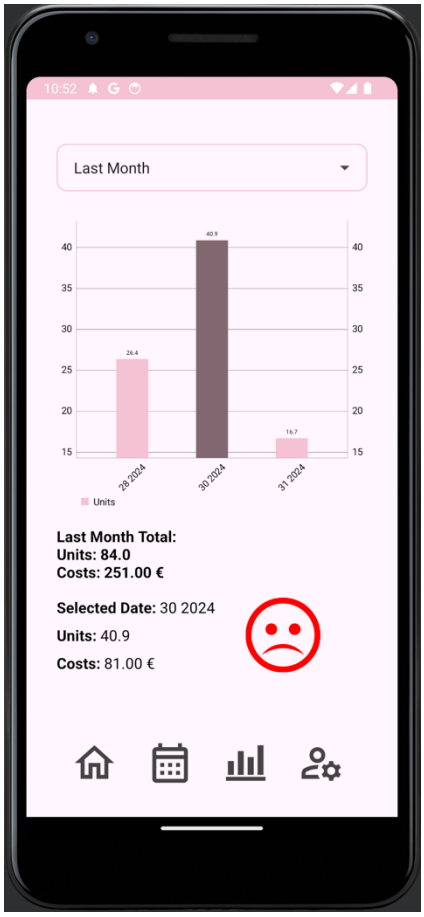

10:29
Scan a barcode

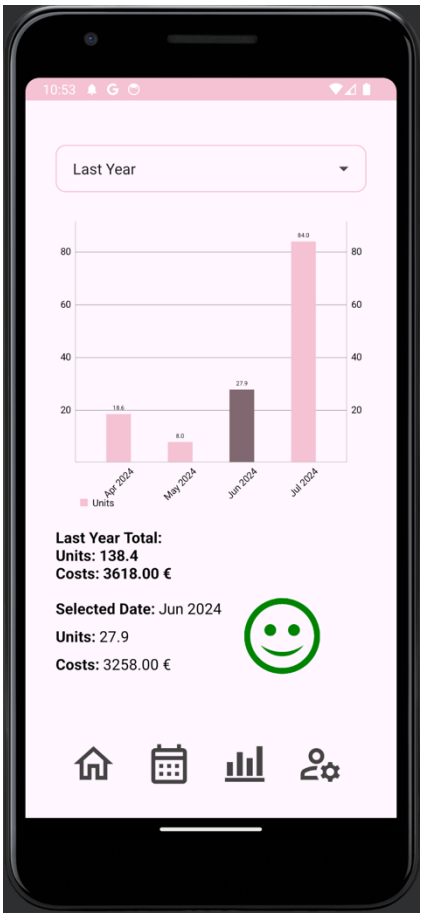

10:35
July 2024
S M T W T F S
1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31
Type: Beer
Size: Pint (568ml)
Alcohol: 4.5%
Cost: 7.20€
Place: pub
Unit: 2.6
Type: Beer
Size: Half Pint (284ml)
Alcohol: 2.0%
Cost: 3.50€
Place: pub
Unit: 0.6

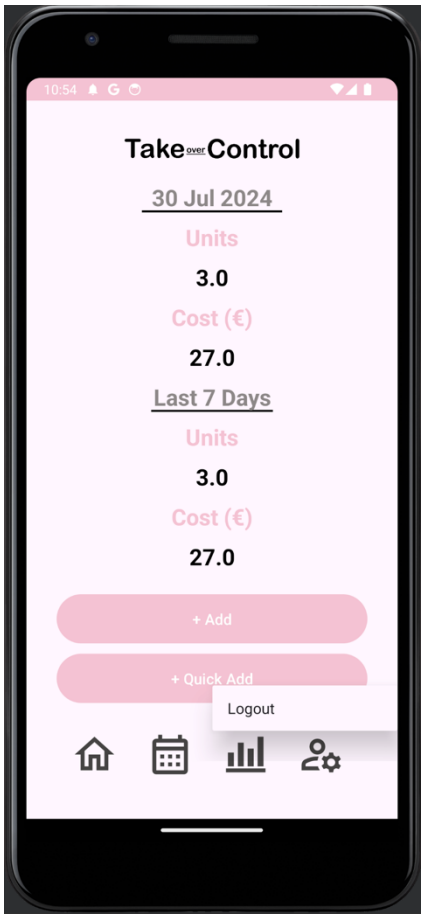

10:36
Type
Beer
Size
Pint (568ml)
Alcohol %
4.5
Cost
7.2
Place
pub
Save
Cancel
Delete


10:50
Last 7 Days
30
20
10
0
24 Jul
25 Jul
26 Jul
27 Jul
28 Jul
29 Jul
30 Jul
Units
Last 7 Days Total:
Units: 57.6
Costs: 178.00 €
Selected Date: 26 Jul
Units: 31.0
Costs: 10.00 €


10:51
Last 7 Days
30
20
10
0
31.0
1.4
2.2
2.4
3.9
9.5
4.3
3.0
23 Jul
24 Jul
25 Jul
26 Jul
27 Jul
28 Jul
29 Jul
30 Jul
Units
Last 7 Days Total:
Units: 57.6
Costs: 178.00 €
Selected Date: 23 Jul
Units: 1.4
Costs: 0.00 €


10:52
Last Month
40
35
30
25
20
15
40.9
26.4
16.7
28 2024
30 2024
31 2024
Units
Last Month Total:
Units: 84.0
Costs: 251.00 €
Selected Date: 30 2024
Units: 40.9
Costs: 81.00 €


10:53
Last Year
80
60
40
20
18.6
8.0
27.9
84.0
Apr 2024
May 2024
Jun 2024
Jul 2024
Units
Last Year Total:
Units: 138.4
Costs: 3618.00 €
Selected Date: Jun 2024
Units: 27.9
Costs: 3258.00 €


10:54
TakeoverControl
30 Jul 2024
Units
3.0
Cost (€)
27.0
Last 7 Days
Units
3.0
Cost (€)
27.0
+ Add
+ Quick Add
Logout

2.5. Testing

Test	Description	Test Data	Expected Result	Actual Result	Pass/Fail
1	Log in	Email:veselin.karamfilov@mail.com Password:654321	User should log in	User is able to login	Pass
2	Sign in with Google account	A Google account (veselinyurievkaramfilov@gmail.com)	User should log in	User is able to login	Pass
3	Registering for an account	Email:i.vatansever@mail.com Password:123456 Confirm Password:123456	The account should be created and sent activation email	The account is created and received activation email	Pass
4	Adding values	Type: Beer Size: Pint(568ml) Alcohol:5.0% Cost:8€ Place: Bar	The values should be added to database	The values are added to database	Pass
5	Quick Add	Type: Whiskey Brand: Jameson	The values should be added to database	The values are added to database	Pass
6	Quick Add with Barcode Scanner	A Guinness can with barcode on it	The scanner should detect the brand from its barcode	The brand is detected via its barcode	Pass
7	Fetching and listing the values from database	6 May is selected from calendar view	The values with date of 6 May should be listed	The values with 6 May are listed	Pass
8	Value edition	Alcohol % : 35 Place: Testing value edition	The record should be updated with new value	The record is updated	Pass

9	Value deletion	Selected a record from calendar view with a date	The record should be deleted	The record is deleted	Pass
10	Stats (Last 7 Days)	The Stats page's default loading value is "Last 7 Days"	The graph and information should be displayed with last 7 days values	The expected graph and information is displayed	Pass
11	Stats (Last Month)	The "Last Month" is chosen from drop-down menu	The graph and information should be displayed with last month values	The expected graph and information is displayed	Pass
12	Stats (Last Year)	The "Last Year" is chosen from drop-down menu	The graph and information should be displayed with last year values	The expected graph and information is displayed	Pass
13	Log out	Logout is clicked under setting menu	The logged in user should be logged out	The user is logged out	Pass

2.6. Evaluation

Although I tested every function of the application during the development phases, I also performed functional and non-functional tests in this section. Using the values specified in the previous section, I tested functions such as login, registration, logout, some CRUD operations, and data retrieval from the database. I also conducted non-functional tests, including performance, scalability, and usability testing. All of the test results were successful, and the application responded as expected.

3.0 Conclusions

I did not encounter any problems with the project that I could not overcome by the end. However, I had to do research on integration, value storage, and queries regarding the Firestore database because I had not had the opportunity to work in such detail with a non-SQL database before. In conclusion, I enjoyed stepping out of my comfort zone, learning new technologies, and developing new things. As a part-time student, I had to make many

sacrifices to spend time on college and this project. Considering the time constraints, I believe I did a good job.

4.0 Further Development or Research

I divided the project into two parts. The first part consisted of the main functions of the project, while the second part included some additional and advanced features. I conducted detailed research for each implementation with which I had no prior experience and put them into practice by repeatedly testing until the expected result was achieved.

5.0 References

Android Developers. *AndroidX Overview*. [online] Available at: https://developer.android.com/jetpack/androidx. [Accessed 18 Dec. 2023].

Android Developers. *Get the last known location | Sensors and location*. [online] Available at: https://developer.android.com/develop/sensors-and-location/location/retrieve-current#java [Accessed 18 Dec. 2023].

Android Developers. *Health Connect | Android health & fitness*. [online] Available at: https://developer.android.com/health-and-fitness/guides/health-connect. [Accessed 18 Dec. 2023].

Android Developers. *Health Connect comparison guide | Android health & fitness*. [online] Available at: https://developer.android.com/health-and-fitness/guides/health-connect/migrate/comparison-guide [Accessed 18 Dec. 2023].

App.diagrams.net. *Flowchart Maker & Online Diagram Software*. [online] Available at: https://draw.io/. [Accessed 18 Dec. 2023].

Firebase. (2019). *Add Firebase to your Android project | Firebase*. [online] Available at: https://firebase.google.com/docs/android/setup. [Accessed 18 Dec. 2023].

Firebase. *Get started with Cloud Firestore*. [online] Available at: https://firebase.google.com/docs/firestore/quickstart [Accessed 18 Dec. 2023].

GitHub (2023). *GitHub*. [online] GitHub. Available at: https://github.com/. [Accessed 18 Dec. 2023].

Google Developers. *Scan Barcodes with ML Kit on Android*. [online] Available at: https://developers.google.com/ml-kit/vision/barcode-scanning/android. [Accessed 18 Dec. 2023].

Trello (2023). *Trello*. [online] trello. Available at: https://trello.com/. [Accessed 18 Dec. 2023].

Visual Paradigm (2019). *Ideal Modeling & Diagramming Tool for Agile Team Collaboration*. [online] Visual-paradigm.com. Available at: https://www.visual-paradigm.com/. [Accessed 18 Dec. 2023].

Adobe (2019). *Stock photos, royalty-free images, graphics, vectors & videos*. [online] Adobe Stock. Available at: https://stock.adobe.com/ [Accessed 16 Feb. 2024].

A Standard Drink in Ireland: *A Health Service Executive Report. (2009).* Available at: https://www.drugsandalcohol.ie/12374/1/HSE_Hope_Standard_drink_in_Ireland.pdf. [Accessed 16 Feb. 2024].

www.drinkaware.co.uk. (n.d.). *Unit and Calorie Calculator | Drinkaware*. [online] Available at: https://www.drinkaware.co.uk/tools/unit-and-calorie-calculator. [Accessed 16 Feb. 2024].

www.youtube.com. (n.d.). *Android Beginner Tutorial #26 -CalendarView [Getting the Date and Displaying in a TextView]*. [online] Available at: https://www.youtube.com/watch?v=hHjFIG0TtA0&ab_channel=CodingWithMitch [Accessed 26 Apr. 2024].

6.0 Appendices

6.1. Project Proposal

6.1.1. Objectives

This project aims to create a secure and scalable Android Application by using Java Programming Language and Google Services, Tools, and Programs such as Android Studio, Firebase, Firestore and other ML Kits. Apart from programming and functionality, it also aims to design application GUI with Adobe Photoshop and visualise the development and interaction phases using draw.io, Lucidchart, and visual-paradigm diagrams.

6.1.2. Background

I chose this project topic because, recently I was asked how many units of alcohol I drink per week upon a request for quotes from insurance companies for home and car insurance. I realised I didn't have an answer to this question. I don't really have a regular drinking habit, but I thought it wouldn't be bad to know how much alcohol I generally consume in a given week, month or year. I thought this would not only visualise my alcohol consumption from a health perspective, allowing me to take preventive steps if necessary but also help me see the money I was spending on alcohol. The result was a desire to find out how much alcohol I consumed. I am already familiar with some of the objectives mentioned in the first question, but additional research is required for those I have yet to use before or know only in theory. I will overcome what I don't know and need to learn with good planning, so I

created a rough workflow plan using Trello. I plan to update this plan by breaking it down into details when necessary, but for now, following the current headings at the specified times will be enough to ensure the objectives are met.

6.1.3. State of the Art

There are already a few similar applications out there. However, when I examined these, they all run locally; they save the manual user input data to the local SQLite database and return information to the user with the generated SQL queries. On the other hand, my application differs from others; It will benefit from cloud services, not local ones. For example, while it offers advanced user authentication using Firebase, it will also eliminate the need to manage a traditional backend server by taking advantage of its serverless architecture. Plus it provides real-time synchronization as well as speed and scalability by using the Firestore database. Moreover, usability is enhanced thanks to Machine Learning implementations from Google Cloud ML Kits. These are differences from the developer's perspective. Let's list the differences that the users will have; In addition to simple data entry, the user will also have advanced features such as data entry with barcode scanning and image detection, location sharing, precaution alert notifications, advanced statistics and interaction with Google Health.

6.1.4. Technical Approach

I will stick to the agile methodology because I plan to add or upgrade some features to the application; these additions will vary on my timeline depending on how I progress. As mentioned in section 2.0, I planned the application development process using Trello Sprint Board. I will also use user stories and use case diagrams to visualize that the functionality complies with the documentation and to facilitate implementation. I will follow continuous integration and testing best practices to ensure everything is progressing as expected. So that I will have developed an organised iterative method for project development by adhering to the listed guidelines above. In addition, my supervisor and I already meet once a week to assess the project. Thus, maintaining open communication lines and working with him will be another essential to the project's success. Because depending on the requirements and the feedback I receive, project development will continue taking shape.

In addition, since I will be using some of the tools and programs I will use, especially many of the Google Cloud services, for the first time, I will review in detail the official documentation and other 3rd party videos necessary to understand and apply their implementation in practice. This situation will most likely cause me to get stuck at some points and even waste a lot of time, so I am aware of the importance of working according to the timeline.

6.1.5. Technical Details

First of all, Java will be used as the application language on Android Studio IDE. Since the application will be an Android Application, the principal library will be AndroidX. Other libraries like Firebase and Firestore SDK will be used for user authentication and database integration purposes. Along with these principal libraries, user authentication, database integration, restful API communication, error handling and some security consideration algorithms and approaches will be implemented.

Since the application will be delivered in two stages, the basic features, manual data entry, and statistical results will be implemented in the first stage. Subsequently and depending on the timeline status, advanced features such as barcode scanner, image detection, location sharing and advance notifications will be included into the application. So for that advanced future, there will be some libraries used in the application like ML Kit Android libraries and Google Play Services.

As for other technical details, since Android Studio provides a device emulator, running and testing the code will be done by running the emulator. Adobe Photoshop will be used to design the application's graphical interface for the pages. One of the indispensable parts, GitHub for software development and version control, will be used. In addition, other different technologies and adaptations will be made as needed during application development, depending on the requirement.

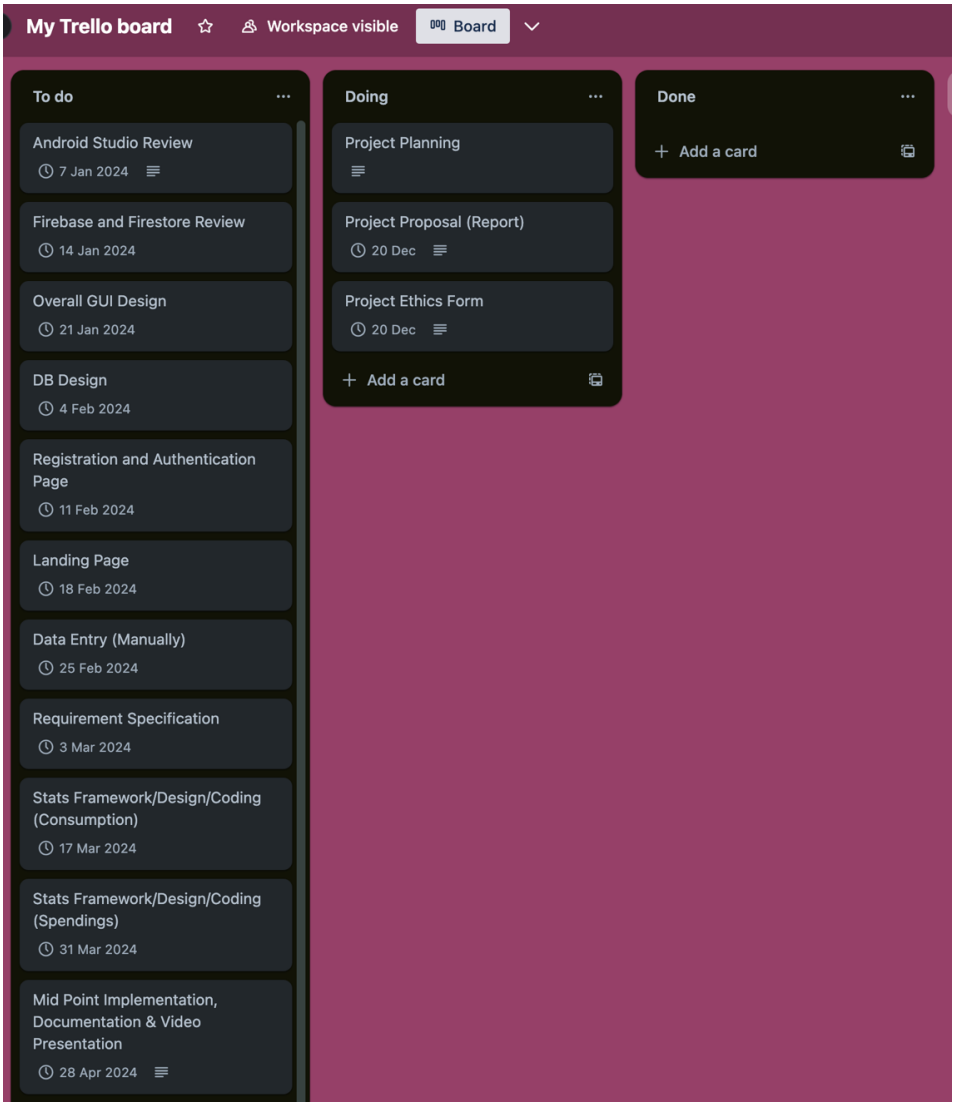
6.1.6. Special Resources Required

Special main resources required to develop this project are as follows:

- Android Studio
- Firebase Account
- Google Play Developer Account
- Cloud Services (Google Cloud)
- Android Device or Emulator
- GitHub Account
- Adobe Account

Apart from these main requirements, other technical online tools, such as drawing diagrams, creating plans, or preparing documentation, will also be needed.

6.1.7. Project Plan



I mentioned in previous sections that I use Trello for planning. As can be seen from the screenshot above, I have determined the project development stages until the mid-point submission date. I divided the project into two main parts. The first part is the fundamental part of the application, and the second part is the extra features to be added after midpoint submission. I am literally starting to develop the application with the new year. The first week will be to review the necessary documentation and videos about Android Studio. I will make the required program and plugin updates. I will determine which version of the library and dependencies I want to use will work most smoothly and make similar determinations, and I will set the project accordingly. Then, I devoted the second week to learning Firebase and Firestore, doing a general documentation review, and reviewing existing examples. Also, since a Firebase account is required for use and implementation, I will ensure that my current account is still active for usage. Then, when it comes to creativity, which is the part I enjoy most, I will make design determinations for the general layout of the application, the login page, the main page, and others. I may even design a logo if there is extra time during the allotted time for design. The following week, I will

move on to the database section and determine the database design, model and relationships. I will also need to review the implementation documentation. I will be using the Firestore (document database) database for the first time; it will be one of the challenges of this project. Afterwards, I will handle the user registration and authentication part with Firebase, another Google Cloud product. Again, I will need to review some documentation for this. Since I have used Firebase once before, I am more comfortable in this regard. After completing the registration and authentication part, the design and coding of the main page to which I will direct the user will be next. After completing the registration and authentication part, it will be time to design and code the home page, to which I will direct the user to where the data entry will be possible. Finally, I allowed the database queries to display the user-requested statistics about a month before the Midpoint submission date. The steps and timelines I have outlined are up to the Midpoint submission.

6.1.8. Testing

I aim to use Unit, Integration and System tests for system-side tests. I plan to observe the behaviour of some methods through Unit Testing by writing code for them (JUnit). I am considering simulating the integration of Firebase, FireStore and other modules with Integration Testing. With System Testing, I desire to perform tests covering all operations, from user authentication to data acquisition and storage. I will test the application regarding the End User Evaluation angle since no actual data is required; I plan to mock up plenty of different accounts and mock-up data for the test. Therefore, there is no consideration in terms of ethics requirements. However, If the participation of a different person or group is required, I will ensure that I meet all needs in terms of Ethics.

6.2. Reflective Journals

Supervision & Reflection Template

Student Name	Veselin Karamfilov
Student Number	x20120885
Course	Computing Project
Supervisor	-

Month: October

What? At the start of the semester this year, which is our last year, we were informed about what we should pay attention to, what we should do and another necessary documentation submission regarding the graduation project. We were asked to determine the project topic in line with this information and to shoot and upload a pitch video that briefly introduces the project.
So What?

I completed the first stage, determining the project topic and shooting a pitch video. I didn't go ahead and do research on the technical details yet because the project pitch video will be reviewed by the supervisor(s), and after it is approved whether it is suitable or not, I will move on to the next stage. Also, this is my first monthly reflective journal. As I mentioned, my project idea is still at the approval stage, so for that reason, this month's reflective journal is general and short.
Now What? I am currently waiting for a supervisor to be assigned for my project, and I will start the project work based on the approval.

Student Signature	Veselin Karamfilov

Supervision & Reflection Template

Student Name	Veselin Karamfilov
Student Number	x20120885
Course	Computing Project
Supervisor	William Clifford

Month: November

What? The first progress of last month was the assignment of a supervisor for my project. In our first meeting, I explained the features I wanted to develop regarding the project and he shared his thoughts with me. We decided to hold a meeting every week on Monday so that we will have frequent meetings about the progress of the project.
So What? We made determinations with my supervisor about the ten features I had identified that I could implement in the project and their degree of difficulty and priority. These points we determined should have been in the task at least until Midpoint. Other less prioritised features will be taken into consideration later again, depending on time and project development.
Now What? Before starting the project, I will organise my tasks, activities, and goals using Trello, a project management tool.

Student Signature	Veselin Karamfilov

Supervision & Reflection Template

Student Name	Veselin Karamfilov
Student Number	x20120885
Course	Computing Project
Supervisor	William Clifford

Month: December

What? I did research on cloud service providers for the authorisation, authentication and database services that I will use in the application. As a result, I decided that Google Cloud is the most suitable for my application. I also prepared and submitted the Project Proposal and Ethic Forms regarding documentation that I had to submit.
So What? The project progress continues in accordance with the planning and dates I determined before; in other words, everything is going well for now. Since I am still in the preparation phase of the project, I have yet to encounter any obstacles. Still, on the other hand, It is apparent that I will face challenges when I move into the development phase.
Now What? As I mentioned, I have not encountered any difficulties yet, and in order to avoid encountering any problems, I am doing preliminary research on possible issues that I foresee before moving on to the development step.

Student Signature	Veselin Karamfilov

Supervision & Reflection Template

Student Name	Veselin Karamfilov
Student Number	x20120885
Course	Computing Project
Supervisor	William Clifford

Month: January

What? This month, I followed my Trello project plan again, gaining a general overview of the implementation of Firebase and Firestore. I also drew the wireframe design for each page of the application. My studies were generally theoretical since I reserved January for general knowledge and research. However, this theoretical information clarified many questions since I will move on to the implementation process starting next month.
So What? My project management continues in accordance with the dates I planned without any negative effects.
Now What? There is no outstanding challenge that I cannot solve right now.

Student Signature	Veselin Karamfilov

Supervision & Reflection Template

Student Name	Veselin Karamfilov
Student Number	x20120885
Course	Computing Project
Supervisor	William Clifford

Month: February

What? This month was a bit busy regarding the project as I moved to the practical implementation phase. Since I acted according to my Trello planning as usual, this month, I carried out the database design, account registration and authentication, main page, and data addition sections in design and code. At the same time, I prepared and submitted the necessary documentation for the Requirement Specification submission.
So What? The project progresses as it should; so far, I have not encountered any difficulties that would challenge or put me behind my plans. However, I am aware that the difficulty level increases as I progress, so I need to spend a little more time and gain preliminary knowledge for future implantations, and if I encounter any difficulties during the implantation, I am aware that I need to spend more time to overcome them.
Now What? Since I can foresee the difficulties and learn the problems I may encounter in advance, there is no difficulty that I have not overcome yet.

Student Signature	Veselin Karamfilov

Supervision & Reflection Template

Student Name	Veselin Karamfilov
Student Number	x20120885
Course	Computing Project
Supervisor	William Clifford

Month: March

What? This month, I took a break from implementing registration and authentication, the main page, and the data addition sections, which I started last month and almost completely completed and focused on documentation since I have to make preparations for the mid-Point Implementation, Documentation, and video Presentation, which was due on April 28th.
So What? Everything is going well so far, but I have to admit that since the CAs for my other three courses are very busy and their due dates are before Interim Submission, I needed to focus more on them than on this project. However, I have completed at least 3 use cases completely, which is not bad for Midd Point submission, which requires at least one complete use case diagram to be completed.
Now What? Other than changing the prioritization, I have no issues with the technical progress of this project.

Student Signature	Veselin Karamfilov

Supervision & Reflection Template

Student Name	Veselin Karamfilov
Student Number	x20120885

Course	Computing Project
Supervisor	William Clifford

Month: April

What? This month was the last month before the middle submission. That's why, as soon as I finished the CA and TABA of other subjects, I continued working on the project from where I left off. Unfortunately, the CA and TABAs of other subjects were quite advanced and made me lose a lot of time.
So What? I continued with my planning until this month, but I fell behind my plans for the first time. Also, I will probably have to submit a little behind in the middle of the submission. Apart from this, on a positive note, I successfully completed the database integration.
Now What? By allocating a little more time after the middle submission for the chapters that are behind the planning, I will first complete the missing parts and then work on the implementations I have planned for the second phase.
Student Signature Veselin Karamfilov

Supervision & Reflection Template

Student Name	Veselin Karamfilov
Student Number	x20120885
Course	Computing Project
Supervisor	William Clifford

Month: May

What? I started this month by completing the missing parts of the midpoint submission. Thus, I implemented the statistics page and its functions. Also, I made improvements and enhancements on CRUD. Thus, the add, edit and delete functions has become fully functional.
So What? I completed the remaining parts planned for the mid-point submission and met the rescheduled schedule. I can say that I am satisfied with the project's progress.
Now What? I will continue to work on the project based on the feedback I receive from Midpoint Submission.
Student Signature Veselin Karamfilov

Supervision & Reflection Template

Student Name	Veselin Karamfilov
Student Number	x20120885
Course	Computing Project
Supervisor	William Clifford

Month: June

What? I received a good enough grade from the midpoint submission, which I thought was worth the effort. I also reviewed the feedback from the examiners in detail and modified the progress of my project slightly in accordance with this.
So What? In addition to the base "Add" functionality, in line with feedback, I added a feature for the "Quick Add" functionality. In this way, users can easily add value by simply selecting a brand. Again, based on feedback, I added a feature for signing in with a Google account. In this way, users with a Google account will be able to log in to the application using their Google account without registering. Moreover, I added a notification feature to the application, so that the user will be notified when daily and weekly limits are exceeded.
Now What? I will complete the project and bring it to its final form next month.
Student Signature Veselin Karamfilov

Supervision & Reflection Template

Student Name	Veselin Karamfilov
Student Number	x20120885
Course	Computing Project
Supervisor	William Clifford

Month: July

What? Since this month is the last month before the final submission, I checked the existing functionalities and made some modifications that I thought were missing or could be better. I also added some new functionalities.
So What? I added a barcode scanner feature to the "Quick Add" page. This way, the user can add data quickly using their mobile device's camera. I implemented the "Forgot Password?" function on the login page and made it work. I made the statistics page more interactive with some emojis. Finally, I did a general code review, made some corrections, and pushed the final version to GitHub.
Now What?

I hope to get an excellent grade on this project and am looking forward to graduating.	
Student Signature	Veselin Karamfilov

6.3. Other Resources – Midpoint Submission

6.3.1. Presentation Video Link

https://studentncirl-my.sharepoint.com/:v:/g/personal/x20120885_student_ncirl_ie/EdHBhpZ_vlhHifLF3imxmY8BQR0ibXAje36BEO5VG85raQ?e=xIbFXG

6.3.2. GitHub Link

https://github.com/veselinkaramfilov/TakeOverControl

6.3.3. Alternative Access Link To All Content

https://studentncirl-my.sharepoint.com/:f:/g/personal/x20120885_student_ncirl_ie/EpjLsCYPgTBBsBSIfKCjRmoBNE0JUM3N34tDKOat8EoNbQ?e=FIasnl

6.4. Other Resources – Final Submission

6.4.1. Presentation Video Link

https://studentncirl-my.sharepoint.com/:v:/g/personal/x20120885_student_ncirl_ie/EYW1-SDe3ZRAtkAujXOD1UYBR3tDJ1CybdX5KFQo7scbGw

6.4.2. GitHub Link

https://github.com/veselinkaramfilov/TakeOverControl

6.4.3. Alternative Access Link To All Content

https://studentncirl-my.sharepoint.com/:f:/g/personal/x20120885_student_ncirl_ie/Ek-7LW1dwXtJiE0krYMlNCMBzK6gyVHl3RHTr0MqCYj-4Q?e=7quc5S