



Ministry of Higher Education and Scientific Research and Computer and Communication

Technologies Directorate-General for Technological Studies

Higher Institute of Technology Studies of Bizerte

Department of Computer Technologies

Initiation Internship

Topic

Upgrade the old application using a restful API and create an event reminder app.

Created by: Merseni Bilel

Supervise by: Iskander Abbessi

Internship period: from 20/08/2021 to 19/09/2021

Host company: CJS-ElAlia

Address: av Habib Bourguiba 7016 El Alia, Tunisie

Tel: 58 197 176

Table of content

Contents

General Introduction	5
Chapter1: Introduction of the company	6
1. Introduction:.....	6
2. Presentation of the company:.....	6
3. Description of the sector concerned by the internship.....	6
4. Presentation of the internship theme.....	6
a. RESTFUL API.....	6
b. event reminder app.....	7
5.conclusion:	7
Chapter2: Preliminary study	8
1. Introduction.....	8
2. Description of the existing.....	8
3. Criticism of the existing.....	8
4. Solution provided.....	9
5. Detailed description of the final solution.....	10
5.1. register new member.....	10
5.2. delete member.....	10
5.3. display members	10
5.4 Add new event	10
5.5 display event	11
5.6 delete event	11
5.7 login member	12
5.8 Add event to favorite	12
5.9 join event.....	12
5.10 architecture used	12
6. diagrams:.....	13
6.1 use case diagram	13
.....	13
6.2 class diagram.....	13

7. Conclusion	14
Chapter 3: Realization.....	15
1. Introduction.....	15
2. Hardware Environment.....	15
3. Software Environment	15
4. Main graphical interfaces.....	19
5. Conclusion	20
General conclusion.....	21

Figure page

Figure 1 : graph represent the number of members which beneficiary Facebook event feature	9
Figure 2 : add new member endpoint	10
Figure 3: delete a member endpoint.....	10
Figure 4: add a new event endpoint	11
Figure 5: delete event endpoint.....	11
Figure 6 : MVC pattern.....	12
Figure 7 : MVCS pattern.....	12
Figure 8 : use case diagram.....	13
Figure 9: class diagram	14
Figure 10: VScode image.....	15
Figure 11 : Postman image	16
Figure 12 : TypeScript image	16
Figure 13 : NodeJS Image.....	17
Figure 14 : Express image.....	17
Figure 15 : MongoDB image	17
Figure 16 : Dart image	18
Figure 17 : Flutter image.....	18
Figure 18: Onboarding Screen images.....	19
Figure 19: Home Screen	20

General Introduction

This rapport is a result of an internship in CJS. The goal of the internship is to create an event reminder cross-platform app for the members so they can get a notification before the event start, also to create an API and connect both the old desktop app and the mobile app

Chapter1: Introduction of the company

1. Introduction:

This chapter covers the introduction of the company and the different tasks I was asked to do as well as the theme of the internship

2. Presentation of the company:

CJS-ElAlia is an association created in 1986. As presented on the logo of the association, the CJS field was astronomy, but with the revelation of the technology, all CSJ branches start focusing on IT,

Nowadays this association provides courses and workshops in Web/mobile development, IoT, cybersecurity, Artificial intelligence. With more than 10 volunteer trainers.

3. Description of the sector concerned by the internship

My task was first to create a restful API to upgrade the old desktop app which was created with electron JS and SQLite, then to create a cross-platform event reminder app for the members, the purpose of the app is to remind them of the date of the events.

4. Presentation of the internship theme

a. RESTFUL API

as we mentioned before the first part of the internship is to create a restful API to upgrade the old app.

The API must contain these endpoints:

- Register a new member
- List all the members
- Login a member (we need this endpoint inside the event reminder app)
- Delete a member and add it to the archive
- Add a new event
- List all events

b. event reminder app

The second task of the internship is to create a cross-platform app to remind the members of the events in which they are interested.

The app must contain:

- Onboarding screen to present the association
- Login screen
- Register screen (optional because the HR can add a member)
- See future events and old event
- Add event to favorite
- Get reminded with notification

5.conclusion:

CJS is a grey-haired association, its purpose is to teach and train the new fields of technology and helps youths upgrade their hard and soft skills

Chapter2: Preliminary study

1. Introduction

In this chapter, we will explain the situation of the company before the development of the app. Also, it will introduce the problem facing the company and the solution proposed.

2. Description of the existing

CJS el Alia accepting new members every summer and organizes countless events in many themes.

The association has a Facebook group that contains the events forms, date, and the date of recruitment of new members.

3. Criticism of the existing

- If a new member is accepted, the HR of the association adds the information of the new member into the database; CJS uses a desktop application created with electron JS and SQLite. Sometimes the Deputy director of the association needs to check the number of members, trainers, and events, so he needs access to the app. But as we know, SQLite is a local Database. So, the information is located only inside the HR laptop; to access the information, the Deputy Director of the association needs to get into the HR laptop or get a DB copy which is not that easy for a not tech person.

- new events are posted using the Facebook event feature inside the association Facebook group. We've created a pool which we asked all member how much they use the Facebook event feature, as you can see in the graph below only 33.3% of members uses the Facebook event feature this means the events posted are not reached by 66.7% of the members.

DO YOU USE FACEBOOK EVENT FEATURE?

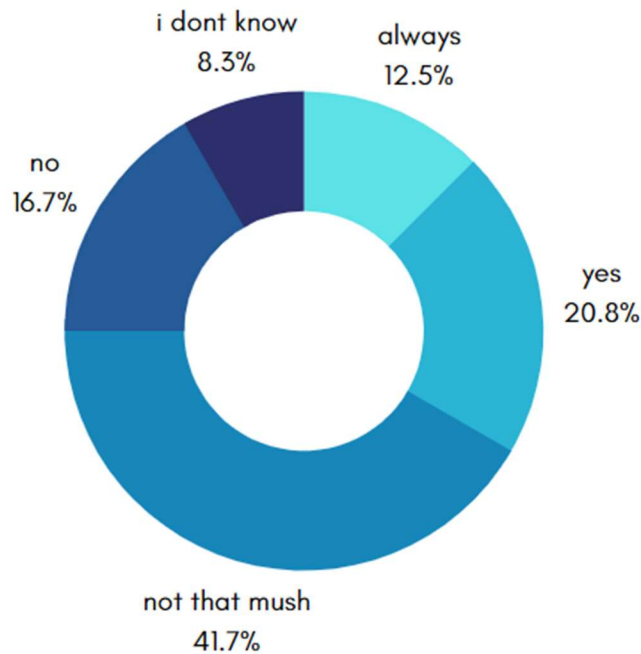


Figure 1 : graph represent the number of members which beneficiary Facebook event feature

4. Solution provided

For the first problem, the best practice is to separate the DB from the desktop application; to do this we need to upgrade electron JS so it connects to an API that communicates directly with the DB.

To solve the second problem, we decided to create an event reminder app which connects to the same API and display all the future event.

By doing this a member can join an event using this app, and they will get a notification before the event starts. Otherwise, members are not obliged to complete a google form before each event, because all the information needed are located inside the DB

5. Detailed description of the final solution

5.1. register new member

To add a new member, all we need to do is to send a post request to the API with the new user information. The server will return a detailed error if an error is triggered.

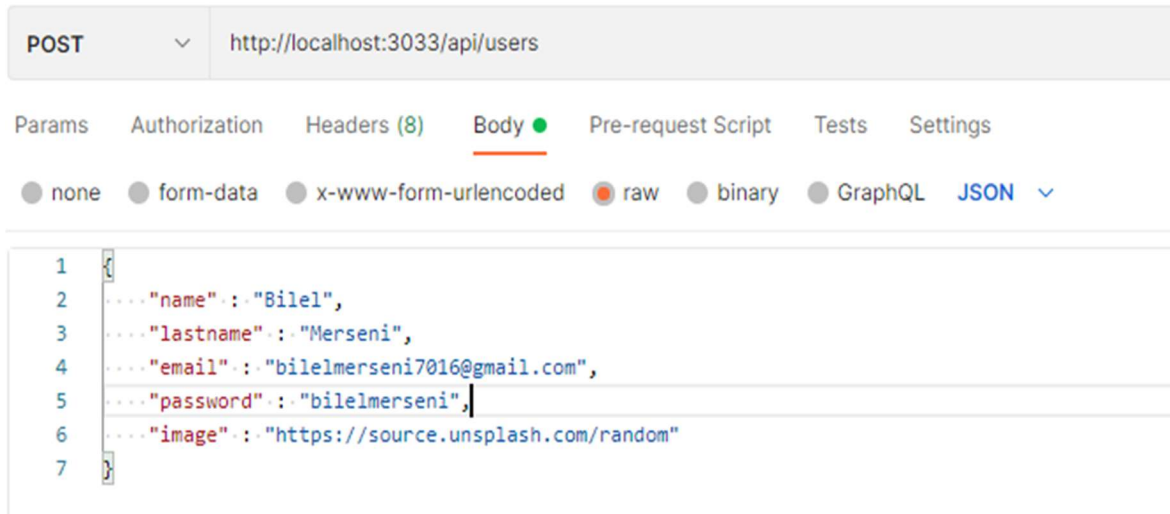


Figure 2 : add new member endpoint

5.2. delete member

A member can update his membership every year, if not, he will be deleted from the member list and added to the archive. To delete a member, we need to send a delete request to the users' endpoint with the user-id

- An archived member can't log in to the app.



Figure 3: delete a member endpoint

5.3. display members

We can get the list of all members by hitting the user's endpoint API.

5.4 Add new event

As we mentioned before the association organize countless events, to add an event, the event leader, create a post request to the API with the necessary information;

If an error is triggered the response will be a detailed error response so we can easily know where the error is coming from.

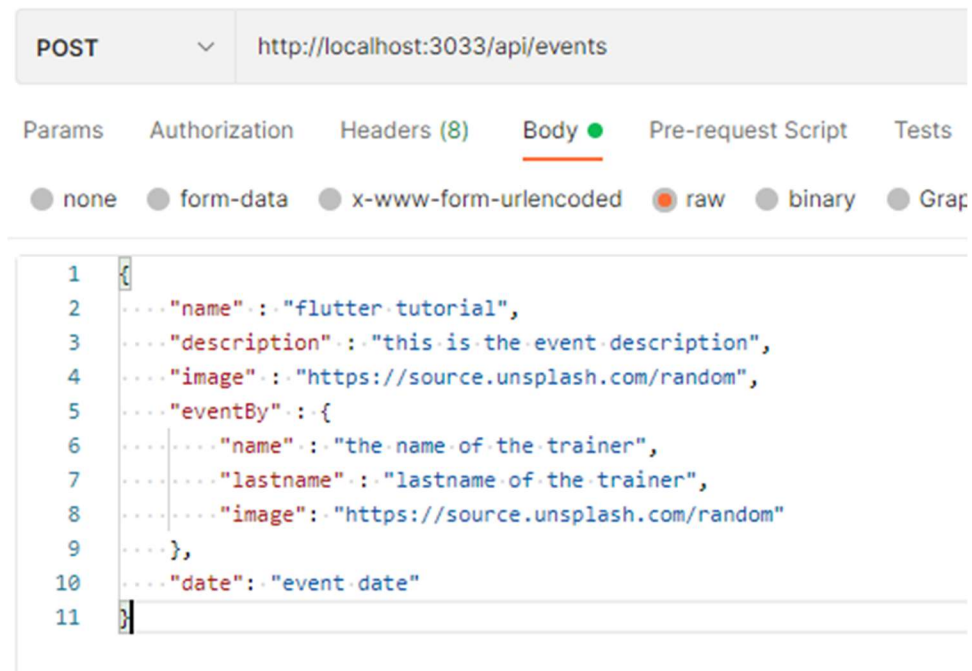


Figure 4: add a new event endpoint

5.5 display event

After login, the users are redirected to the homepage where all events are displayed.

5.6 delete event

As we mentioned before, CJS create a countless number of events event, that's why we don't need to show members the very old events, so if an event date exceeded 30days it will be automatically added to the archive, we can also delete an event manually by sending a delete request to events endpoint adding the event id.



Figure 5: delete event endpoint

5.7 login member

Login is like registering a new member, to log in, a member needs to access the app, after login, a list of events will be displayed on the home screen as we mentioned before. The server will return a detailed error description if any error is triggered.

5.8 Add event to favorite

As we said only association members are allowed to download and get access to the app, after login, a list of events will be displayed on the home screen as we mentioned before. The user can add the event to his favorite. This will create a local favorite list

5.9 join event

As we said only association members are allowed to download and get access to the app, after login, a list of events will be displayed on the home screen as we mentioned before. The user can join the event. When clicking the join button, a request will be sent to the server which will update the events table and add a new participant. The app also will remind the user of the event date using local notifications

5.10 architecture used

the MVCS - Model View Controller Service pattern.

MVC - Model View Controller, is a design pattern used to separate the source code in Model (Data model and business logic), View (the user view of the model, it's the GUI, Ex: webpage), and Controller (connects the view with the model, acting as a bridge).

MVCS - Model View Controller Service, similar to MVC plus Service layer that encapsulates the business logic, remaining the Model clean.

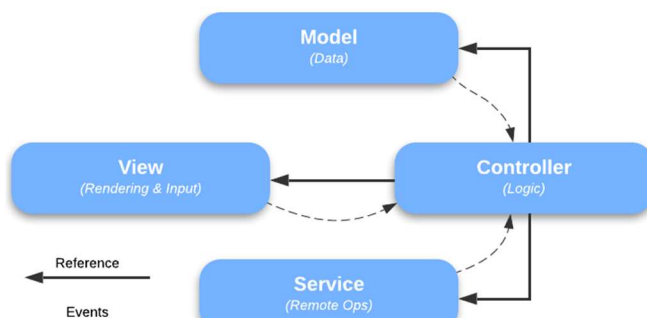


Figure 7 : MVCS pattern

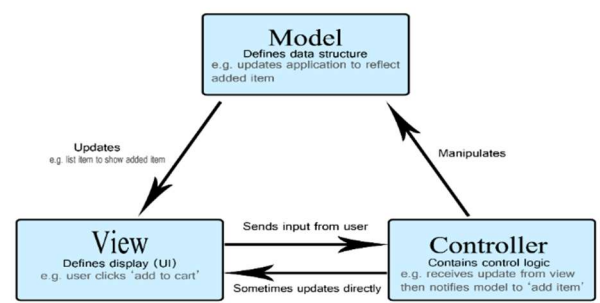


Figure 6 : MVC pattern

6. diagrams:

6.1 use case diagram

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

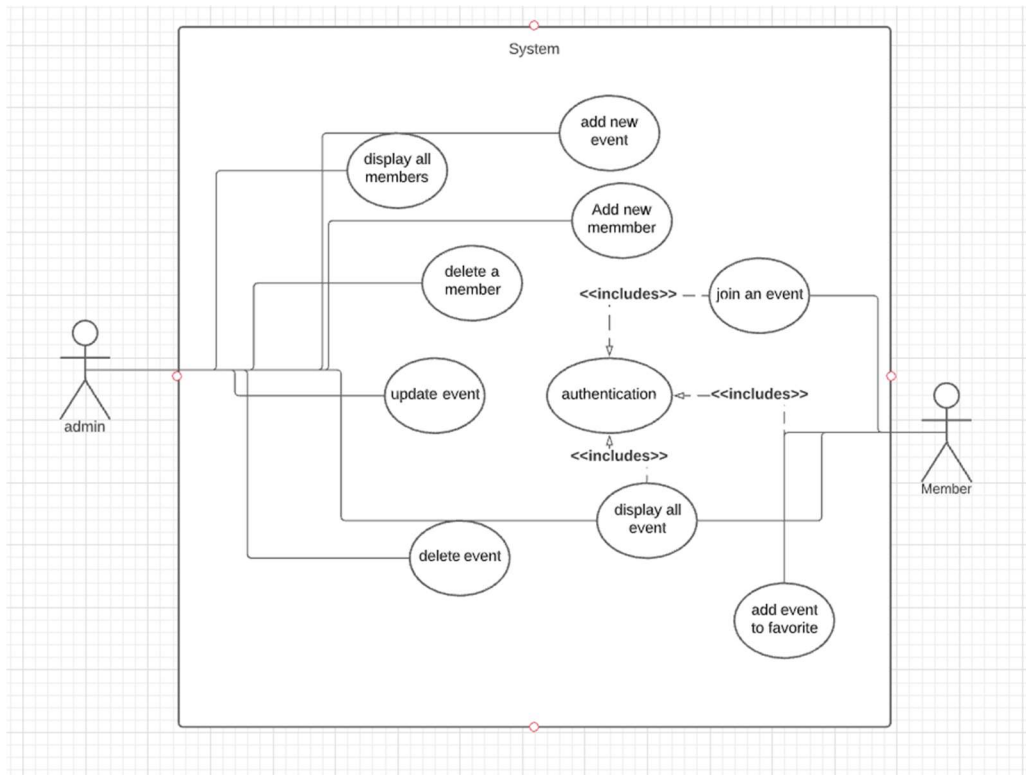


Figure 8 : use case diagram

6.2 class diagram

In software engineering, a class diagram in the Unified Modeling Language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among objects.

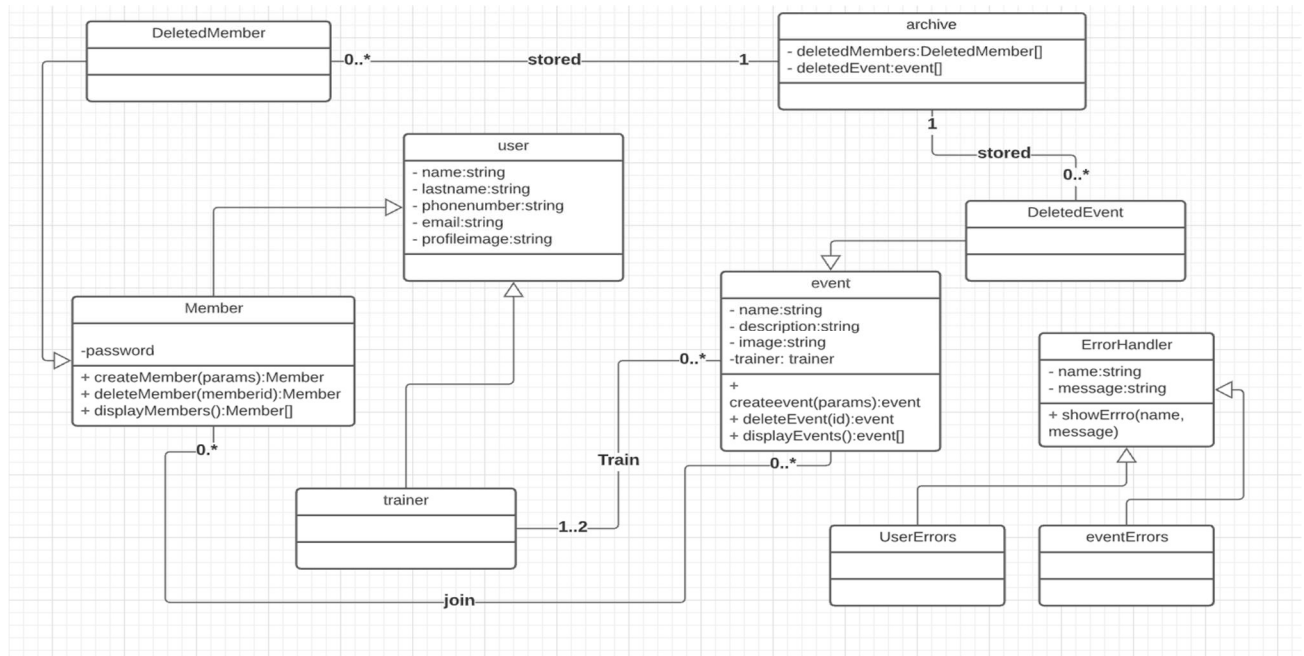


Figure 9: class diagram

7. Conclusion

The work demanded is to create a clean architecture restful API, and then a cross-platform application that helps members get reminded of the date events

Chapter 3: Realization

1. Introduction

this chapter will contain the hardware and software used in the process of creating the application, including the programming languages and the technology used to achieve the goal of the internship.

2. Hardware Environment

the project was made with my pc:

- CPU: Ryzen 5 3600x 4.5ghz
- GPU: RTX 2060 super OC 6GB
- RAM: 16Gb 3200 MHz
- Hard Drive: 1TO HDD 128SSD
- OS: Arch Linux

3. Software Environment

The project was made using:

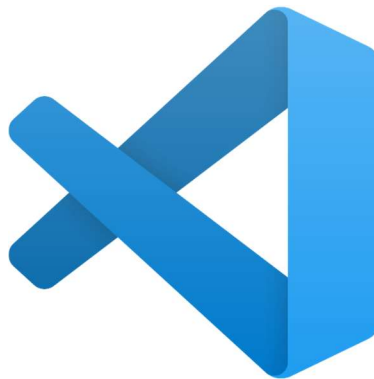


Figure 10: VScode image

Visual Studio Code is a source code editor developed by Microsoft for Windows, Linux, and macOS. It includes support for debugging, embedded Git control and GitHub, syntax highlighting, intelligent code completion, snippets, and code refactoring



Figure 11 : Postman image

Postman is a collaboration platform for API development. Postman's features simplify each step of building an API and streamline collaboration so you can create better APIs—faster.



Figure 12 : TypeScript image

TypeScript is a programming language developed and maintained by Microsoft. It is a strict syntactical superset of JavaScript and adds optional static typing to the language. TypeScript is designed for the development of large applications and trans compiles to JavaScript.



Figure 13 : NodeJS Image

Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser.



Figure 14 : Express image

Express.js, or simply Express, is a back-end web application framework for Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs. It has been called the de facto standard server framework for Node.js.



Figure 15 : MongoDB image

MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like

documents with optional schemas. MongoDB is developed by MongoDB Inc. and licensed under the Server-Side Public License.



Figure 16 : Dart image

Dart is a programming language designed for client development, such as for the web and mobile apps. It is developed by Google and can also be used to build server and desktop applications. Dart is an object-oriented, class-based, garbage-collected language with C-style syntax.



Figure 17 : Flutter image

Flutter is an open-source UI software development kit created by Google. It is used to develop cross-platform applications for Android, iOS, Linux, Mac, Windows, Google Fuchsia, and the web from a single codebase. First described in 2015, Flutter was released in May 2017.

4. Main graphical interfaces

When a member opens the app for the first time he will get an Onboarding screen that describes the app and the association in general

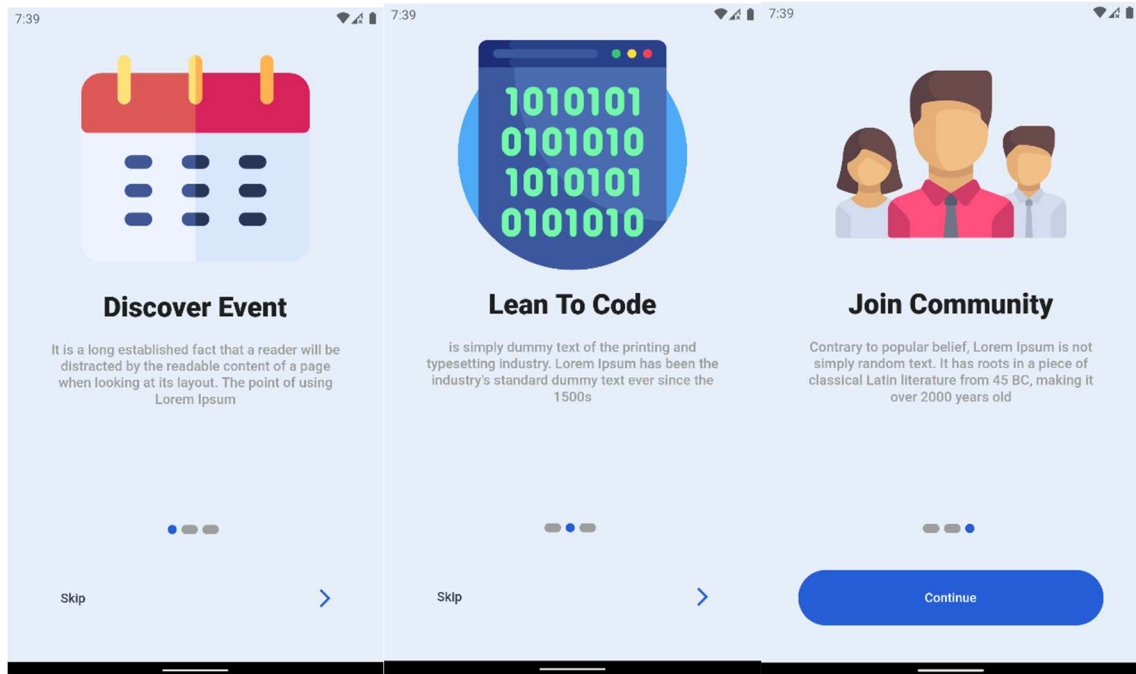


Figure 18: Onboarding Screen images

After the onboarding screen a login will popup, the default credentials of a new member are, his email, and a default password. After login, the user will find a top event list and a for you event list. He can add an event to his favorite or click on it and get more information.

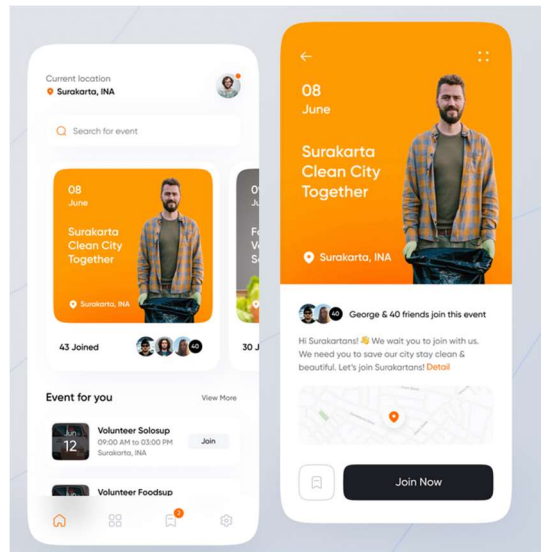


Figure 19: Home Screen

5. Conclusion

The project was made using many different tools and software that was well put together to create a fully functional backend and mobile application that can satisfy the needs of the association.

General conclusion

On the whole, this internship was a useful experience. I have gained new knowledge, skills and meet many new people; I achieved several of my learning goals. Not only but also, I've created a fully functional useful backend with a fully lightweight mobile app with fancy animation and a good level of security which helps members get reminded of events.

The internship was also good to find out what strengths and weaknesses are, and it has given me new insights and motivations.