Bilkent University

Department of Computer Engineering

Senior Design Project

Project Analysis Report

Project Name: Nomad

Group Members: Gülnihal Muslu
Berk Ataç
Can Ozan Kaş
Ali Kemal Özkan
Tarık Emin Kaplan

Supervisor: Uğur Güdükbay

Jury Members: H. Altay Güvenir, Özcan Öztürk

Innovation Expert: Veysi İşler

Project Analysis Report
November 11, 2019
This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

Table of Contents

1. Introduction

Nowadays, travelling is an easy and usual activity for people[1]. Everybody wants to explore places that they have never been. However, there are a lot of buildings and places or activities to do in a country/city. Therefore, people have to choose the best of them according to their interests because they usually have a limited time to see a city. Deciding which place is worth seeing or which food is worth eating can be tricky in most times due to people usually preferring to explore cities that they have never seen before. There are some applications and websites for that, but they have never been enough for people that are such travel lovers.

To solve that problem, we thought of a project -which name is Nomad- where people can give trip advice to other people. Nomad means a member of a group of people who move from one place to another instead of living in the same place all the time. Our project idea comes from there. In this project, a person can create a route in a city or country and s/he can give advice on that trip. Users will have a chance to select a route from various list of routes that are created by other users and they can follow that path step by step. Since routes can be seen before and they can be scored, interpreted; people can choose their route easily and more close to their dream trip. With this project travels will become easier and more fun.

This report contains analysis of our project with three subtitles: current system, proposed system and other analysis elements. All system models and user interface included.

2. Current System

Most close application that close to ours is Trip Advisor. TripAdvisor is an American travel and restaurant website company that shows hotel and restaurant reviews, accommodation bookings and other travel-related content. It also includes interactive travel forums. It is wider than our application. It has hotels, restaurants, rental cars, flights, stuff to do. etc., basically all in. They offer places to go in a city. However our app focused on trips and routes. Not only places but also routes of others are important in Nomad. If you don’t like someone’s route, you can create your own route. Our app does not offer places to go, it offer routes to follow. Routes can involve anything such as a street, a museum, a restaurant, a statue etc., does not have limits.

3. Proposed System

3.1. Overview

Nomad is a mobile application, first will be published in Play Store will be coded in Javascript. It will be addressing to people who want to travel and tour companies. With this application, users can create trip routes, which will be held in a database, as traveling by using GPS in their cell phones. While users are creating a route trip, they can place an object with augmented reality, like a coin or a token, when they arrived where they want to go. These objects are checkpoints for other users who will use routes created by other users.

The trip routes created by users are designed to follow a certain path, which means users are able to see where to go first, then second and then third. This way, they will not struggle finding the way as mostly, people travel to places which they have not been there before and Nomad will show the way to the checkpoints. While users are taking a trip created by other users, they can collect objects and tokens, which were put by the creator of that trip, using their phones’ camera. Tokens can be used to create new routes. If user follow more routes, s/he will be able to create more routes. Once, a user completes a trip route, will be able to vote the trip out of 10. Thus, other choose can choose a route according to its rating. Also, all user will have points that increased by travelling distance. Thus, trip routes and trip creators have their own ratings and points that people can see them.

Nomad enables users to give advice to other users as well as take advice from other users. This provides users a better trip experience as every user can have different interests. To illustrate, if a user wants to take a nature trip and all the high rated places are historical places, s/he can still find a trip that s/he can love from the trip route list of the place s/he visit.

3.2. Functional Requirements

- Users will be authenticated by creating an account with an email or with their social media accounts (facebook, twitter, google).
- Every user will have a profile. Their photo, username, name-surname, points, tokens and routes can be seen from their profile. Also users can access others profile from a route info.
- Users can search a route by various filters such as: rating, city, start point, end point, country, count (how many people followed that route before), theme, time (required to finish that route), includes a place.
- Users can create their own routes, and they can set it either public for other people to use it or private if they do not have enough token to share it. User can place checkpoints whatever they want. However there will be minimum limit of checkpoints for different distances. They can set a theme for route they created.
- Users can follow existing routes created by other people and can vote the routes after finishing a route. In each route, there will be checkpoints. In checkpoint there will be items that placed by creators. With collecting that items, users will gain tokens that can be used later to create their own routes.

- Routes may have “themes” if users specify them on creation. For example a route with a “relaxation” theme will have stops at parks, coffee shops etc. whereas a route with a “historical” theme will have stops at historical sightseeing points.
- To gamify the app with AR, while creating a route, the users may drop tokens on the checkpoints using their phone cameras, and the users travelling these routes will be notified of the existence of such tokens while they are near.
- Information about routes such as distance, estimated time to finish, checkpoints, theme of the route, ratings, how many people followed that route will be provided.
- Routes for biking will also be provided (they will have a longer distance, and will have fewer stops overall).
- Existing routes can be updated over time by the creator user.
- Users can comment on routes after completing them (for example, to ask for an update at a specific location in the route).

3.3. Nonfunctional Requirements

Extensibility:

- Nomad should allow adding new functionalities, features, components and so forth.
- The system should be kept up to date.

Availability:

- Nomad will be available 24/7 for the users.
- It will open for every android user who use android 8 or higher.

System:

- Nomad will be an Android application.
- Application will require an Internet Connection to work.
- A smooth, minimalistic and fast user interface will be designed.

Performance:

- Nomad should not take more than 5 seconds to log in.

- Application should not take more than 15 seconds to get ready after app icon is clicked.

Scalability:

- Nomad will support large number of users.

Security:

- Nomad needs to secure user information from any possible threats.

Reliability:

- Nomad needs to be stable and crash free.

Usability:

- Application should be easy to use for all user types because anyone who can use a mobile phone is possible user of this application. Thus, we need to provide user friendly and simple user interface.
- Application should clearly explain what it is aim and why user should use this app.

3.4. Pseudo Requirements

3.4.1. Implementation Requirements

- The application will be a mobile application.
- Github will be used for version control.
- The application will be implemented mainly on Javascript.

3.4.2. Economic Requirements

- Storage of the data can be a problem because there will be no deletion of data. Every route that users produced will be recorded and won't be deleted. Due to that, server costs can be relatively high [2] .
- Our application will use GPS to keep track of the users' traveling path. That will require a good Internet connection. A change for ISP may be needed.
- When a user creates a path, the cost of the way will also be a parameter of our system. Another user who wants to try this path should consider the cost of the path too.

3.4.3. Social Requirements

- This app will help more people to communicate directly. In other applications gives spots based on the top-rated paths but our application will be based on fully user paths.
- Since our user paths include the cost too, it can separate people economically. Some users may not be able to effort the other user’s paths.

3.4.4. Sustainability Requirements

- Our application highly relies on users for developing. Any policy that makes users uncomfortable can blow up our application but at the same time, it becomes so easy to develop if users like our product.
- Since it depends on people, we probably will not face idea problems.

3.4.5. Environmental Requirements

- Our application will include lots of places and if those places will include also natural places, this may cause environmental pollution.
- Since some routes of the users will be more rated than the others, some roads will be used so much that is another cause for environmental pollution in cities.

3.4.6. Professional and Ethical Requirements

- Augmented reality becomes hard when there is a moving object. Because of that, we may not be able to add places that have lots of moving objects such as cars and people. The objects that we add should not touch to them.
- Since the places we choose should be static, some places may not be simulated. Some people may not allow us to simulate the places. Recording those places is possible but it can only be done hiddenly and this creates the ethical issues. We should be careful when implementation comes to that point.
- User account information we hold should be kept secure and should not be shared with third parties. Infos, if needed, should be used anonymously.

3.5. System Models

3.5.1. Scenarios

3.5.1.1 - Signing Up

Use Case: Sign Up

Entry Condition(s):

- User should have a social media account.
- Use should have our application in his/her device.
- The social media account should not have a record in our database.

Exit Condition(s):

- User leaves from our application.

Flow of Events:

1. User selects a social media platform.
2. User enters e-mail and password of her/his social media account.
3. Our application will check the existence of that account and will add it to our database.

Alternative Events:

1. If there is not an account or already existing account,the user will be directed to the same panel.

3.5.2.2 - Signing In

Use Case: Sign In

Entry Condition(s):

- User should have a social account that is validated by our application.

Exit Condition(s):

- User leaves from our application

Flow of Events:

1. When user validates her/his social media account, the account will be saved in user’s device.
2. After sign in, the user will be directed to the main panel of our application.

Alternative Events:

1. User’s account may not be validated and s/he will be directed to login panel again.
2. User can be directed to that page if s/he deletes the save data on her/his device.

3.5.1.3 - Viewing the Profile

Use Case: View Profile

Entry Condition(s):

- User should click the profile button on the screen.

Exit Condition(s):

- User can leave the application without doing anything.
- User can select another menu option.

Flow of Events:

1. User clicks the view profile button.
2. If the user wants to see the followed routes, s/he can click the “Followed Routes” button.

3. The user can create custom routes but publishing them is an option. By using that button, the user can see his/her own routes.
4. User can edit his/her info after clicking edit profile button.

3.5.1.4 - Viewing Routes

Use Case: View routes

Entry Conditions(s):

- User should click the view routes button.

Exit Condition(s):

- User can leave the application without doing anything.
- User can select another menu option.

Flow of Events:

1. In the beginning, all routes in the user's city will be displayed. They will be sorted according to their popularity.
2. If user wants to specify the route details, they can be given in the search areas. There will be options in the filter for that purpose.
3. After search, user can view that particular route in detail. The details contain information such as the creator and checkpoints on route.
4. This particular route can be added to the favorite routes of the user.

Alternative Flow of Events:

1. If the specified route cannot be found, the initial view page will be displayed with an error message.
2. There is also an option to start to follow the route.

3.5.1.5 - Creating Route

Use Case: Create route

Entry Condition(s):

- User should click the create route button.

Exit Condition(s):

- User can leave the application without doing anything.
- User can select another menu option.

Flow of events:

1. User will provide information about route.
2. After providing information, the user will add candidate checkpoints to the route that was recorded using GPS.
3. The user can publish the route after meeting the publishing requirement which is paying tokens. The number of tokens will be calculated by considering the creation and number of checkpoints.

Alternative Flow of Events:

1. User may want to keep the route for himself/herself or could not manage to pay the tokens. In that case, the user can save the route and use it on his/her own.

3.5.1.6 - Following a route

Use Case: Follow a route

Entry Condition(s):

- A route can be followed in the detail panel after a successful search.
- The favorite route panel also includes the start button.

Exit Condition(s):

- User can leave the application without doing anything.
- User may successfully finish the route.
- User may cut the following in between by clicking leave button.

Flow of events:

1. User will start to follow the route by paying the token cost.
2. User may leave early but before leaving s/he will rate the route. In the end, some token will be given to user for future follows.
3. If user successfully finishes the road, there will be a rating procedure too.

3.5.1.7 - Viewing Other Profiles

Use Case: View other user’s profile

Entry Condition(s):

- In the search page, there is a button for looking at the profile of owner of the route.
- There will be a menu option that

Exit Condition(s):

- User can leave the application without doing anything.
- User can select another menu option.

Flow of events:

1. User can view another profile by searching based on username or the result panel of search route panel.

3.5.1.8 - Deleting Route

Use Case: Delete Route

Entry Condition(s):

- User can delete his/her routes and individual routes can be reached from user’s profile.

Exit Condition(s):

- User can leave the application without doing anything.
- User can select another menu option.

Flow of events:

1. User goes to his/her profile.
2. User clicks the “View My Routes” button.
3. User selects one of the routes.
4. User deletes the route.

3.5.2. Use Case Model

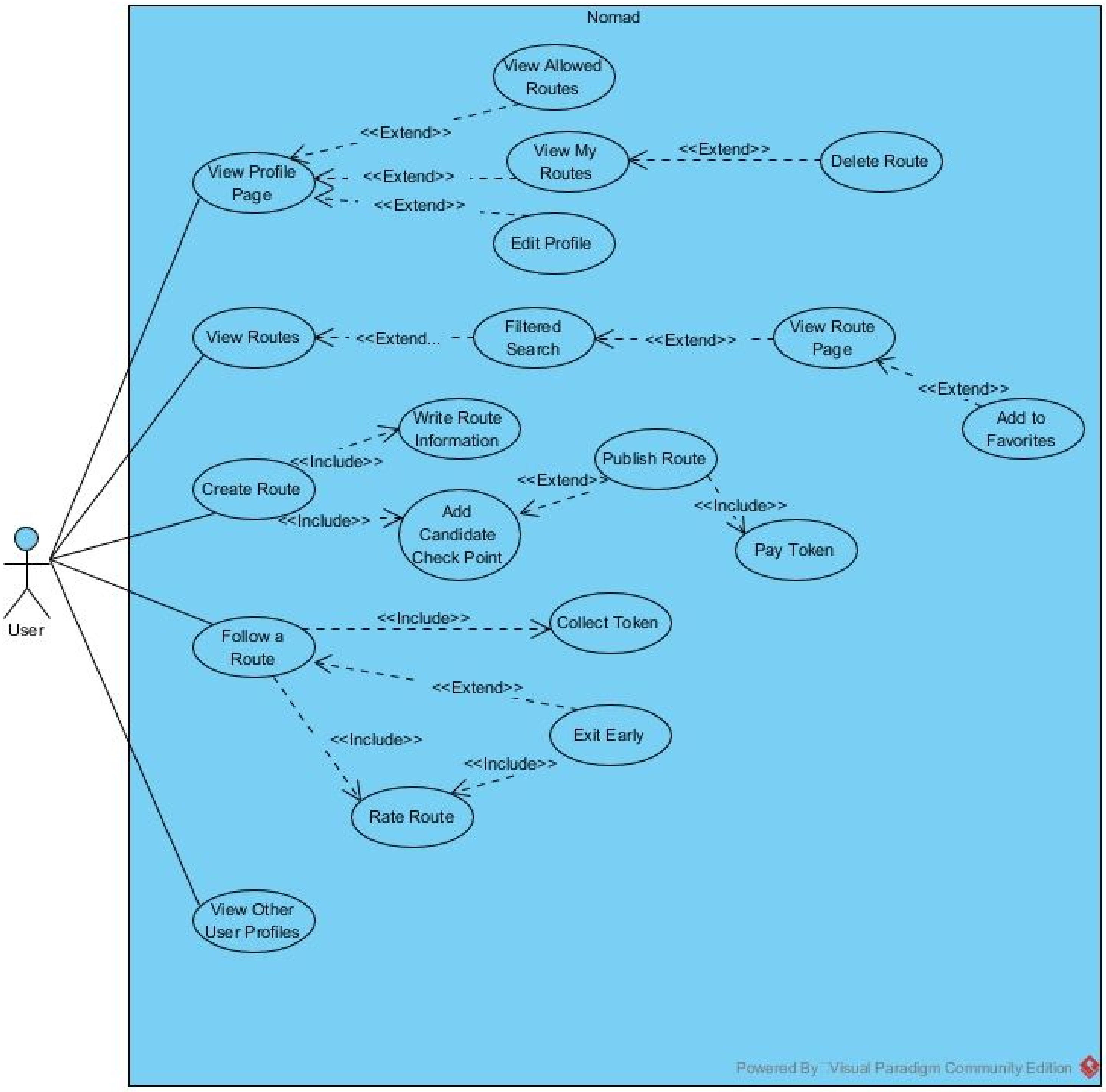


Figure 1: Use case diagram

3.5.3. Object and Class Model

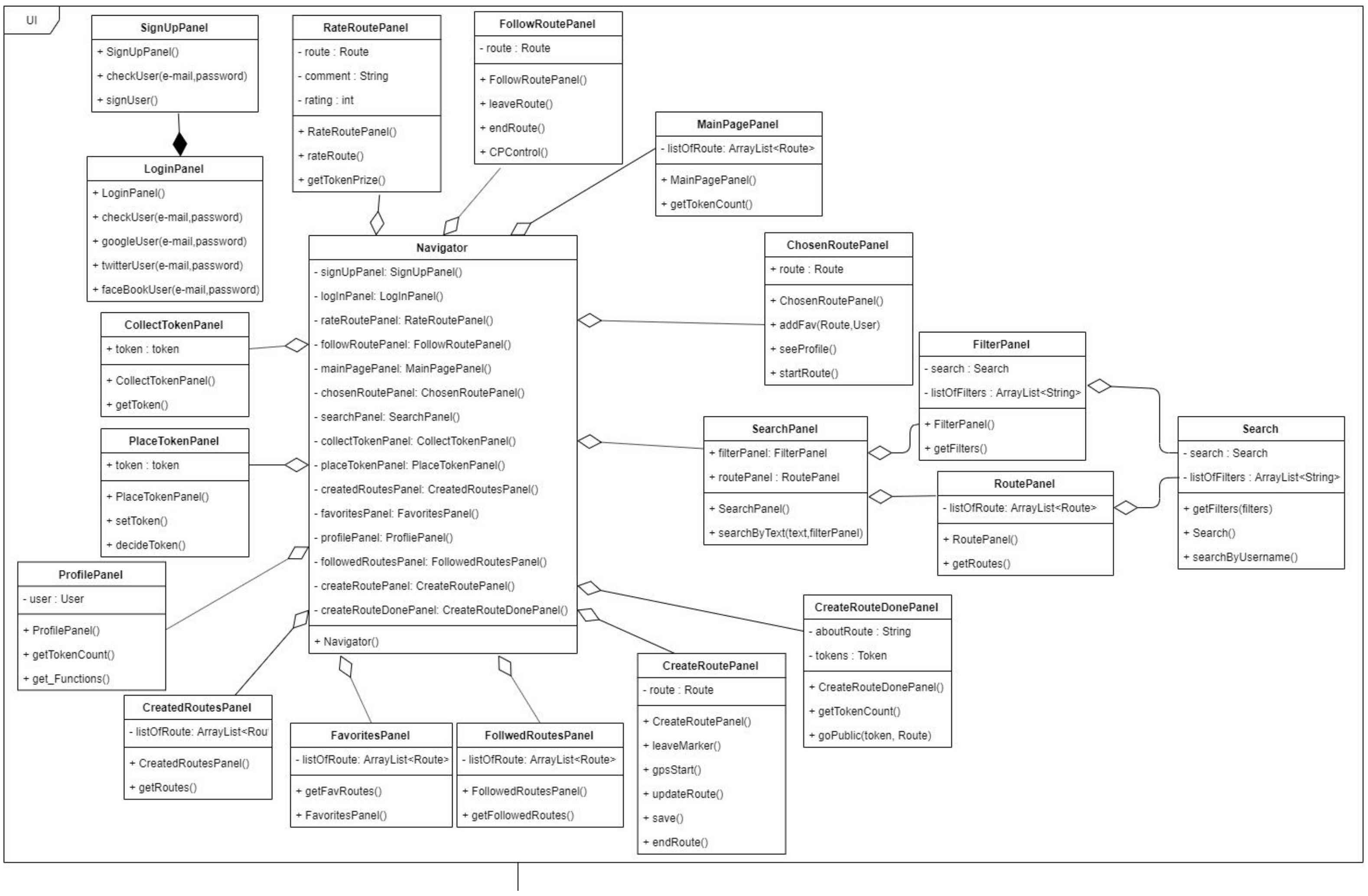


Figure 2: Class Diagram for UI

All attributes and functions are noted in our diagrams with clear names about what they are for. Explanation of these classes also serves as an explanation for mock-up designs in 3.5.5 .

SignUpPanel: This class is for the panel where user signs up to our system.

LoginPanel: This class is for the panel where user logs in to our system. Either by our system or with google, facebook, twitter accounts.

MainPagePanel: This class is for the panel where user what to do. They can either create or follow a route or choose one of the Top 10 routes listed here, they can see their token count.

RateRoutePanel: This class is for the panel where user rates a route, writes a comment and gets token rewards.

FollowRoutePanel: This class is for the panel where user sees the route he/she is following and tracks progress. They can leave a route early (leaveRoute()) if they wish. CheckPoint controls are done in this panel (CPControl()).

ChosenRoutePanel: This class is for the panel where user sees detailed information about a route they choose.

SearchPanel: Search panel is the class where user searches routes to follow. This panel contains two more panels which are **FilterPanel** and **RoutePanel.**

FilterPanel: This the panel which opens when user wants to filter search results.

RoutePanel: This panel is the show search results (routes). May be one page or more pages.

CreateRoutePanel: This class is for the panel where user creates a route. User can leave markers and code tracks and shows users movements here.

CreateRouteDonePanel: When desired route is drawn, in this panel user enters detailed info about route selects checkpoints and make his/her route public.

FollowedRoutesPanel: This class is for the panel where user routes they followed.

FavoritesPanel: This class is for the panel where user sees their favorite routes.

CreatetRoutesPanel: This class is for the panel where user sees their created routes.

ProfilePanel: This class is for the panel where user sees their user information, edit it, goes to followed or created routes panels or goes to favorites panel.

PlaceTokenPanel: This class is for the panel where user access their mobile phone camera and choses a token to place it on screen.

CollectTokenPanel: This class is for the panel where user access their mobile phone camera and collects a token from screen.

Search: This class is for search functions. For example, searching routes by their creator names, searchByUsername().

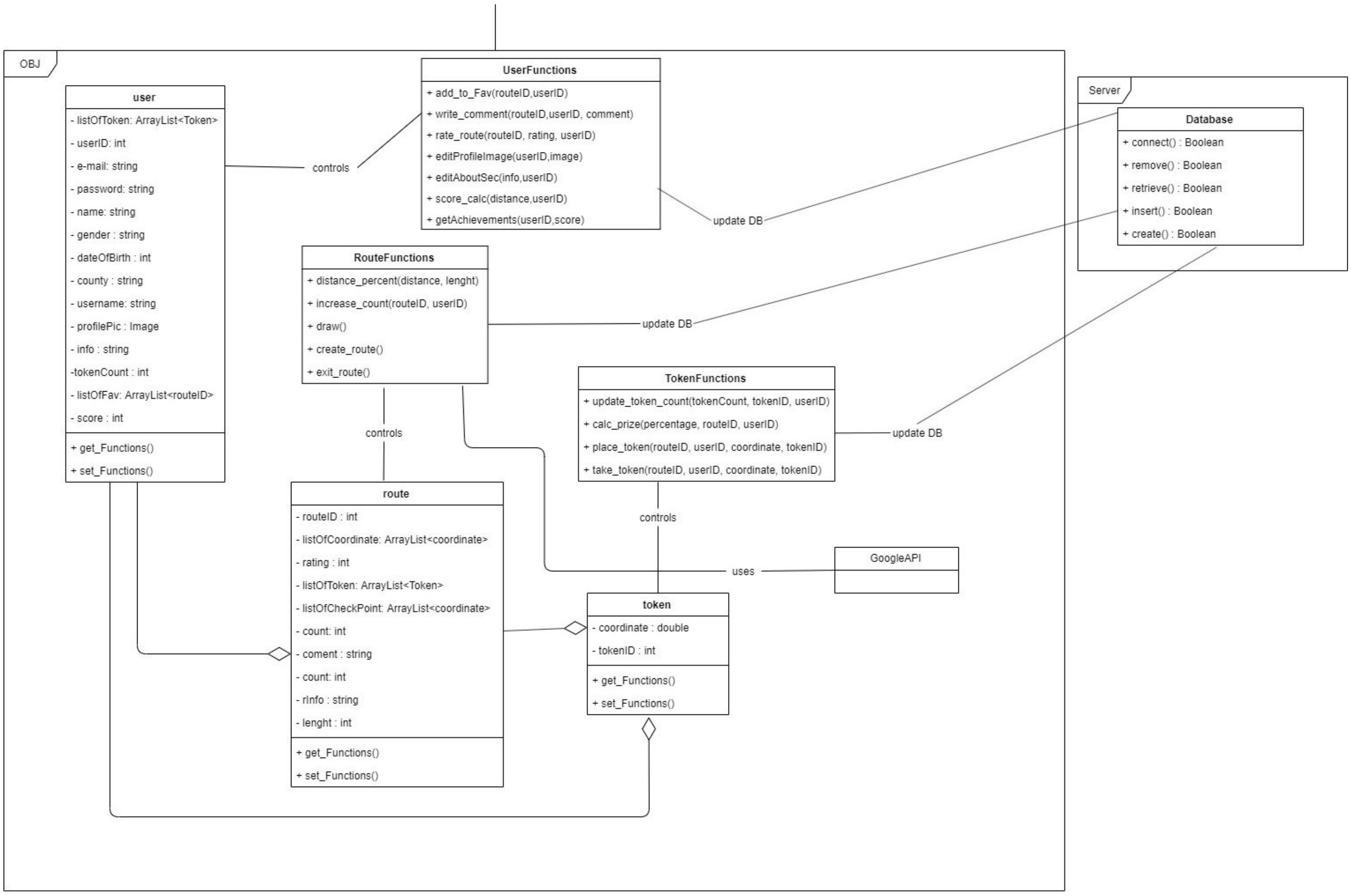


Figure 3: Class Diagram for objects

All attributes and functions are noted in our diagrams with clear names about what they are for.

User : This class keeps information about a user in form of attributes.

UserFunctions: This class controls user class through functions using attributes of both user class and other classes as parameters. For example if a user wants to change their profile picture, UserFunctions class executes editProfileImage() function.

route: This class keeps information about a route in form of attributes.

RouteFunctions: This class controls route class through functions using attributes of both route class and other classes as parameters. For example if a user wants to exit from a route, exitRoute() function executes.

token : This class keeps information about a route in form of attributes.

Token Functions : This class controls token class through functions using attributes of both token class and other classes as parameters. For example if a user finishes a route, prize token is calculated by calc_prize().

Database : This is the class that keeps consistent the data of the system. Its methods are regularly executed by other classes.

GoogleAPI : We will use GoogleAPI and various other APIs and classes for our implementation. GoogleMap, Database, Location and geoPlugin classes etc.

3.5.4. Dynamic Models

3.5.4.1. Sequence Diagrams

3.5.4.1.1. Sign Up

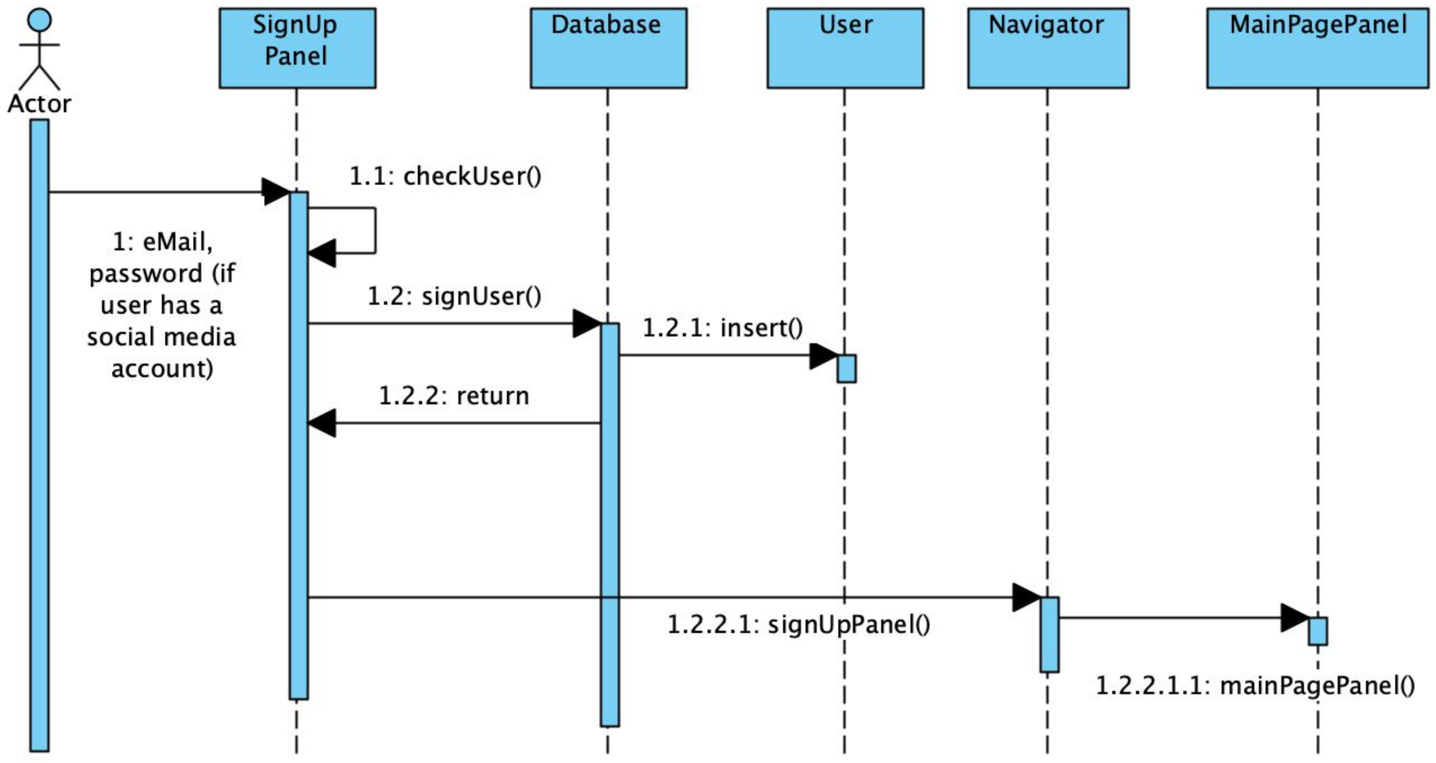


Figure 4: Sign Up Sequence Diagram

When user opens the app, needs to sign up first. With social media account user can sign up easily. This diagram shows the sign up sequence.

3.5.4.1.2. Login

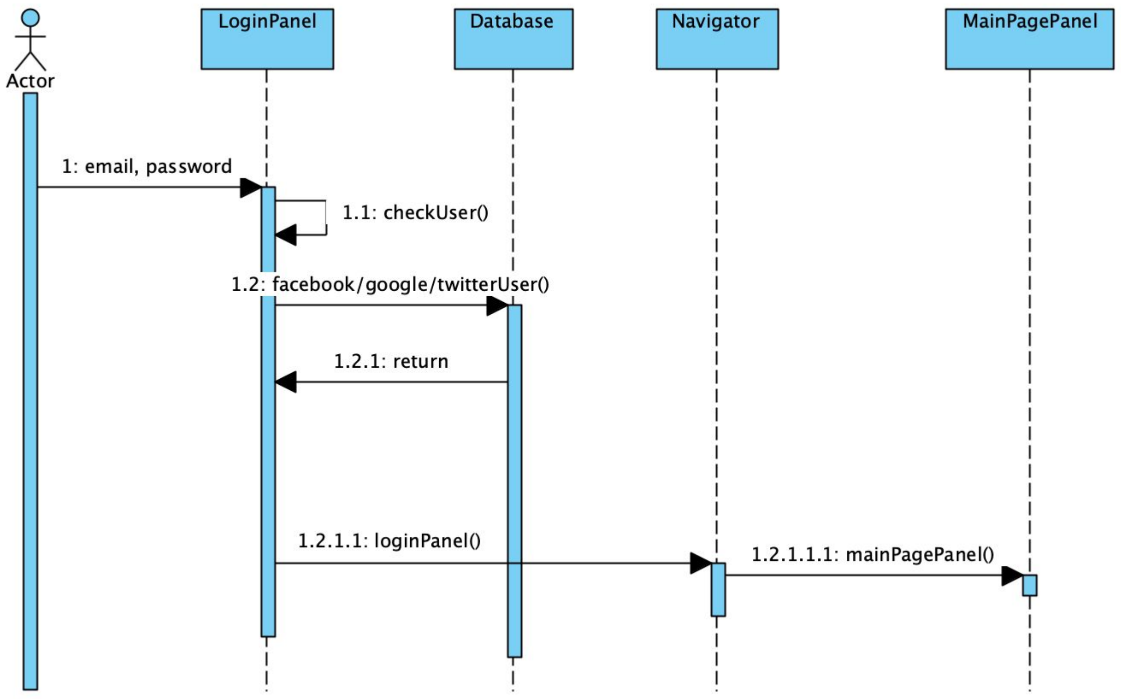


Figure 5: Login Sequence Diagram

User can login with email password or directly with social media accounts. After login user directed to the main page.

3.5.4.1.3. Edit Profile

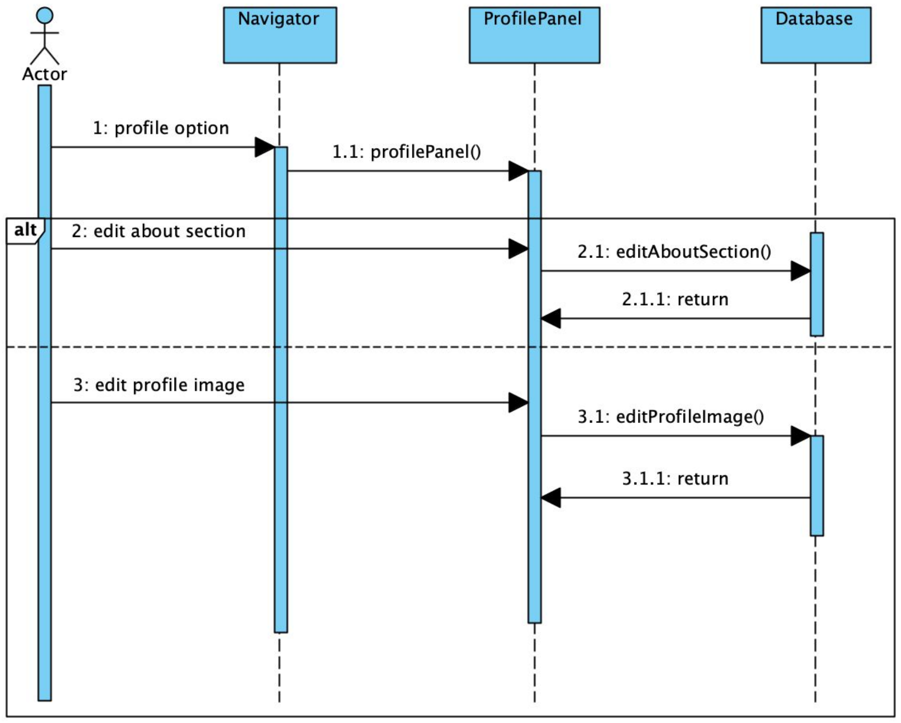


Figure 6: Edit Profile Sequence Diagram

Users can edit their profiles. They can change profile image, about info etc.

3.5.4.1.4. Search Routes and Filter Results

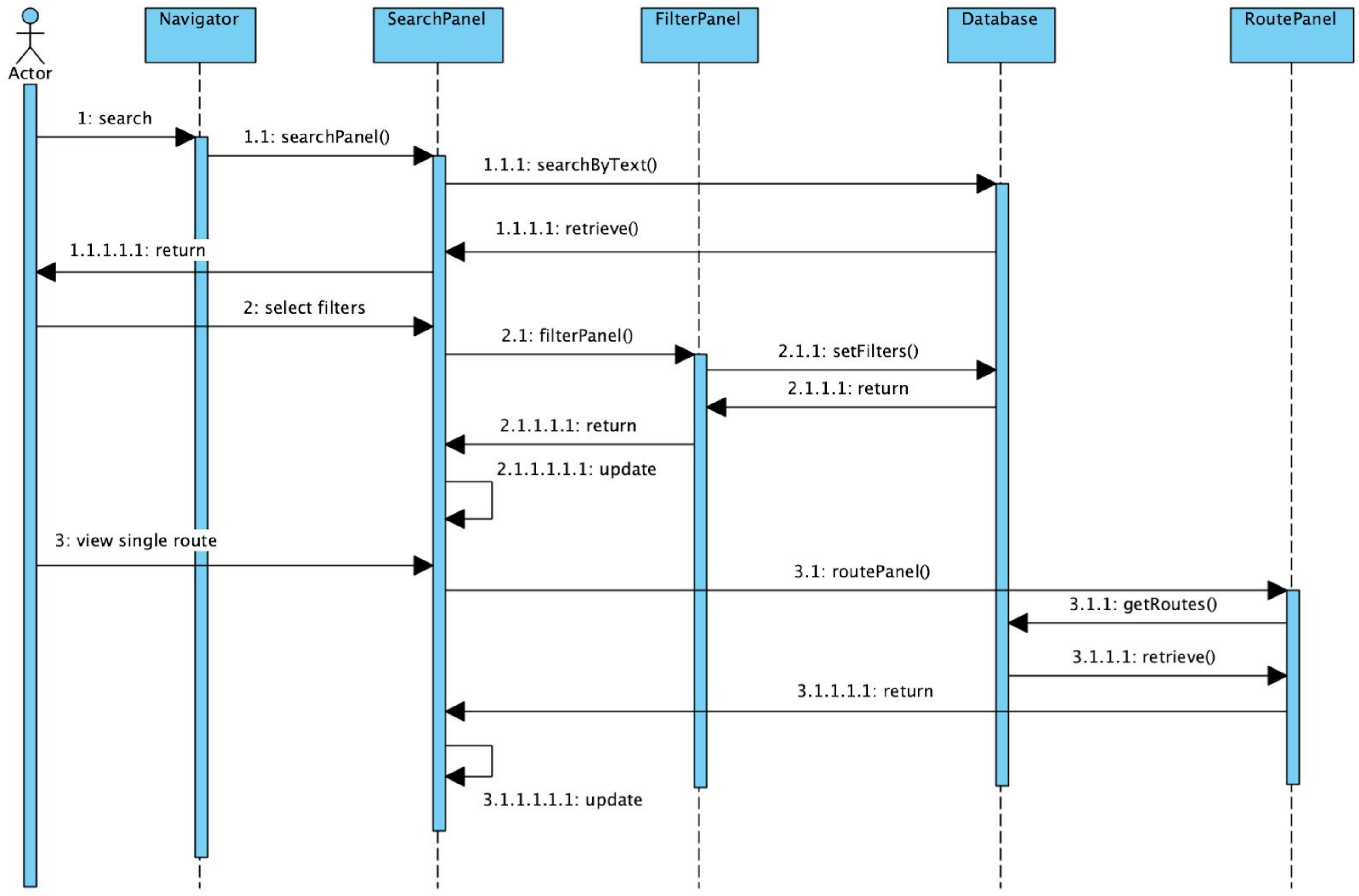


Figure 7: Search Routes Sequence Diagram

Users can search routes by different filters. Searching can be with text or filters. It will show the results immediately.

3.5.4.1.5. Create Route

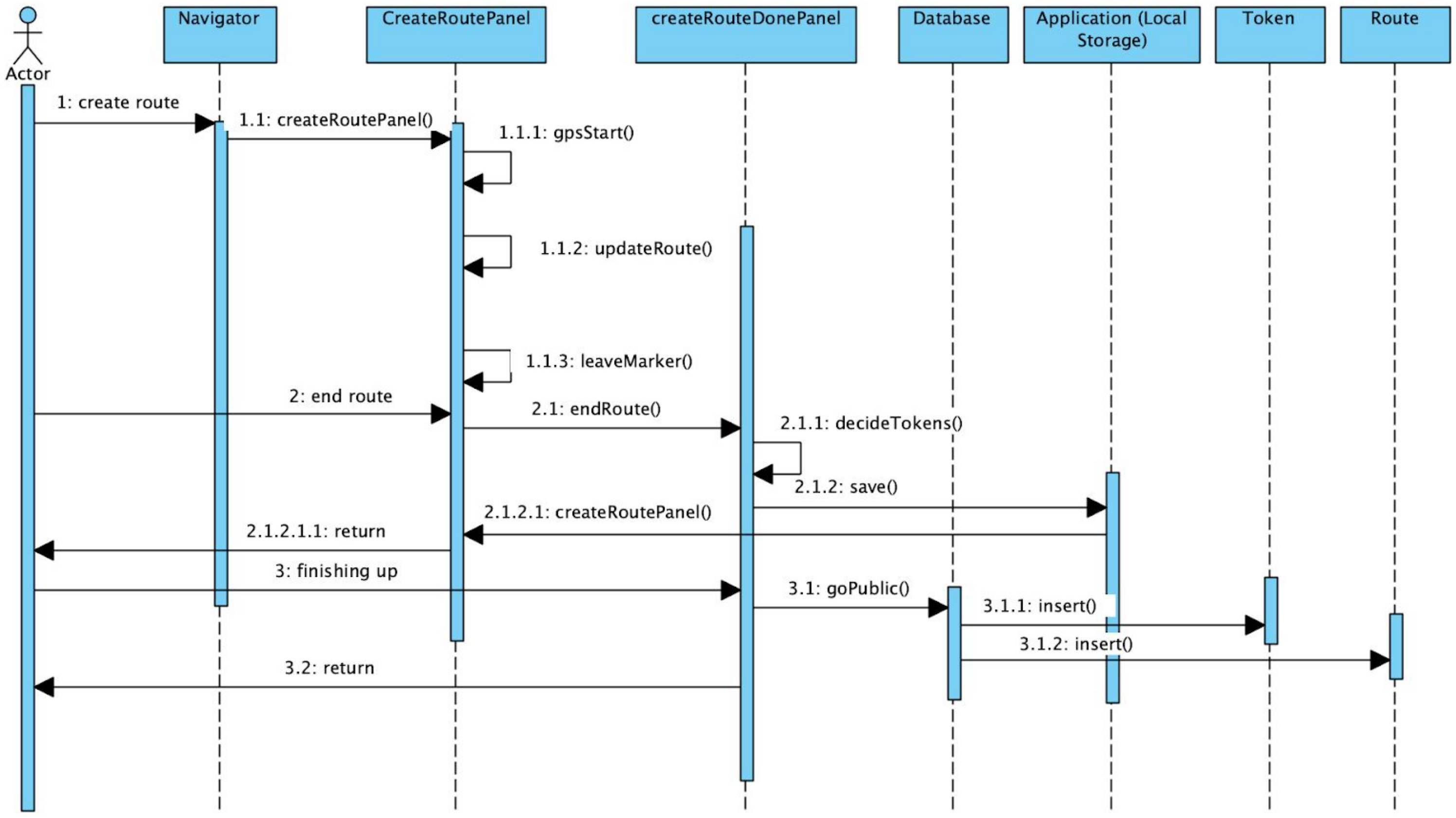


Figure 8: Create Route Sequence Diagram

User can create a route. When creating route, gps starts immediately and update the route continuously. User can leave marker to places for choosing check points later. After ending route user decide tokens and save the route. If user has enough tokens, can share the route to public immediately.

3.5.4.1.6. Follow Route

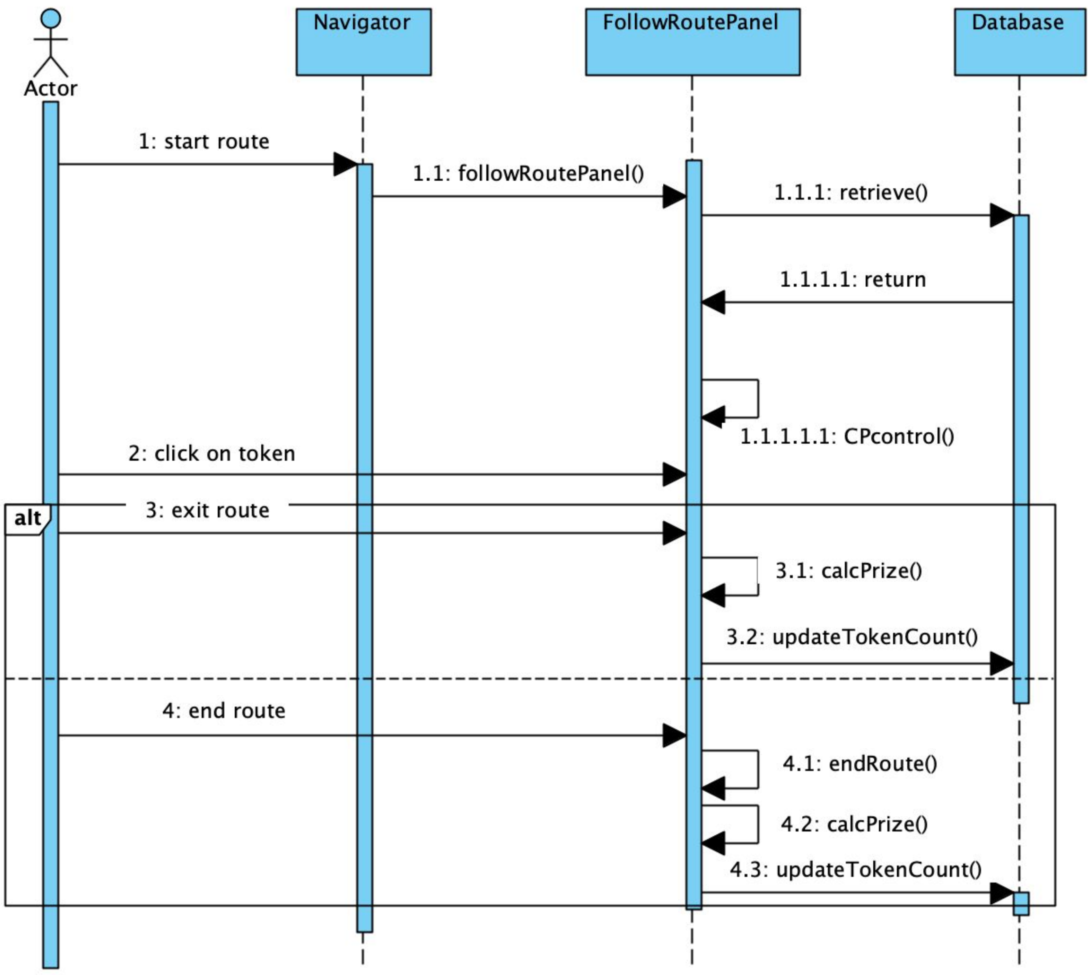


Figure 9: Follow Route Sequence Diagram

User can follow a route from existing routes. When you start a route, checkpoint control will be done continuously. After end route or exit route it will calculate the prize. At the end token count will be updated.

3.5.4.2. Activity Diagram

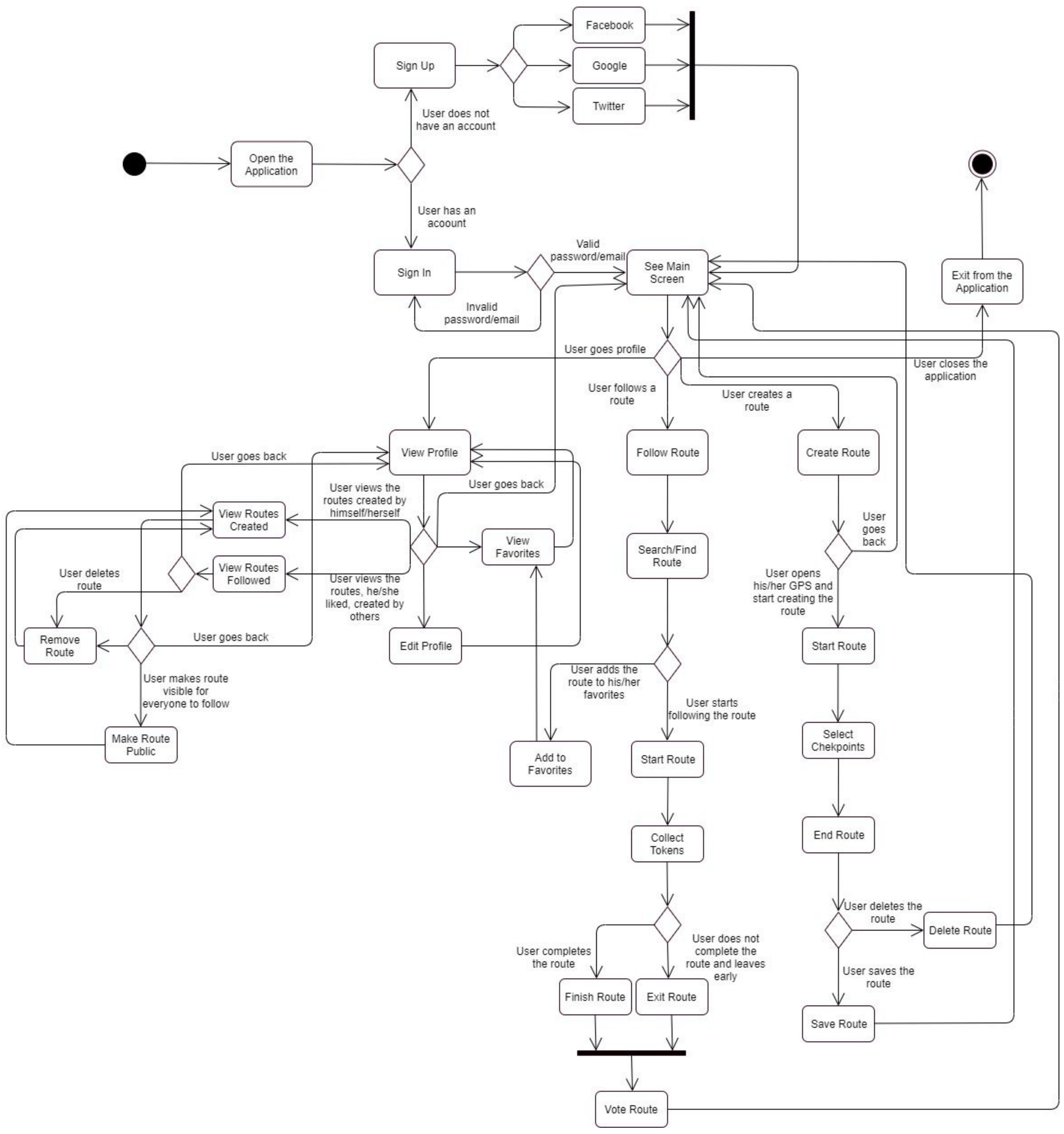


Figure 10: Activity Diagram

User opens the application and signs up first if s/he does not have an account, signs in if s/he already has an account. Sign up action allows users to choose 3 different methods. After successfully signing in, the user will be directed to the main screen where s/he can view his/her profile, follow the routes that have been created before and create his/her own route.

When the user views his profile page, s/he can also view her/her favorite routes, the routes self-created, where the user can remove or publish the routes s/he want and the routes that the user followed, where s/he can remove the routes s/he want. Also the user can edit his/her profile. From all these options, the user can go back to the view profile screen, then the main screen.

When the user chooses to follow a route, s/he will start with searching and finding the route s/he will follow. Once the user selects his/her route, s/he can directly add it to his/her favorite routes or start his/her trip. During the trip, the user will collect tokens and gain points that s/he will use later. If the user wants to end his/her trip early, it will be possible. Yet, the user has to vote the trip, no matter how much s/he completed it. Then, the user will be directed to main screen.

If the user chooses to create his/her own route, s/he will use the GPS on his/her phone and start the trip. The user will select the checkpoints where s/he wants during the trip. After completing the trip, the user either save the route or delete it. Then the user will be directed to the main page where s/he can start over or close the application.

3.5.5. User Interface

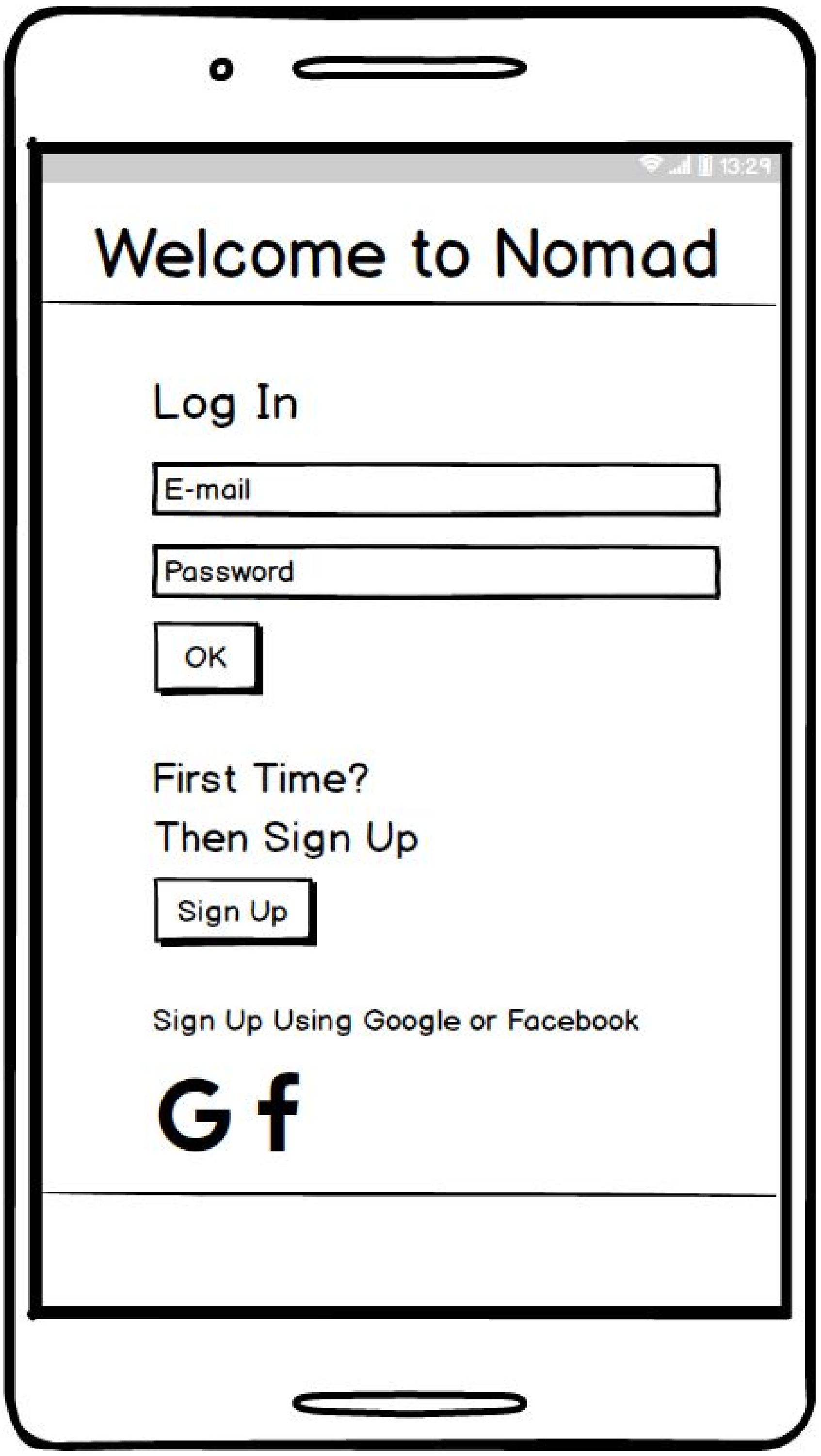


Figure 11: Login Screen

User can login with their e-mail and password. In addition they can log in with social media accounts such as google and facebook.

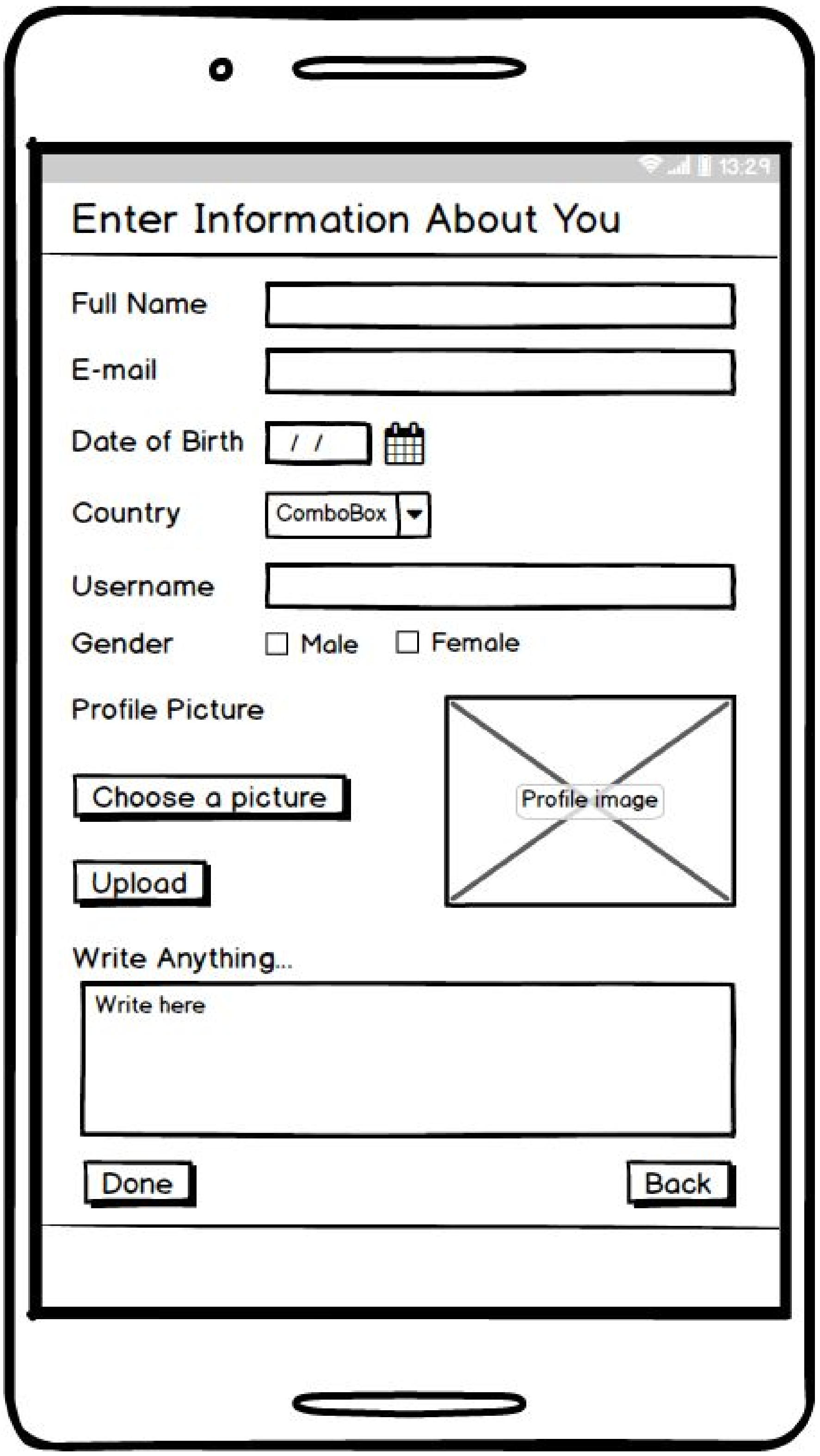


Figure 12: Sign Up Screen

User needs to sign up first. Name, e-mail, password, gender etc. should be entered when signing up. Users can write short info about themselves.

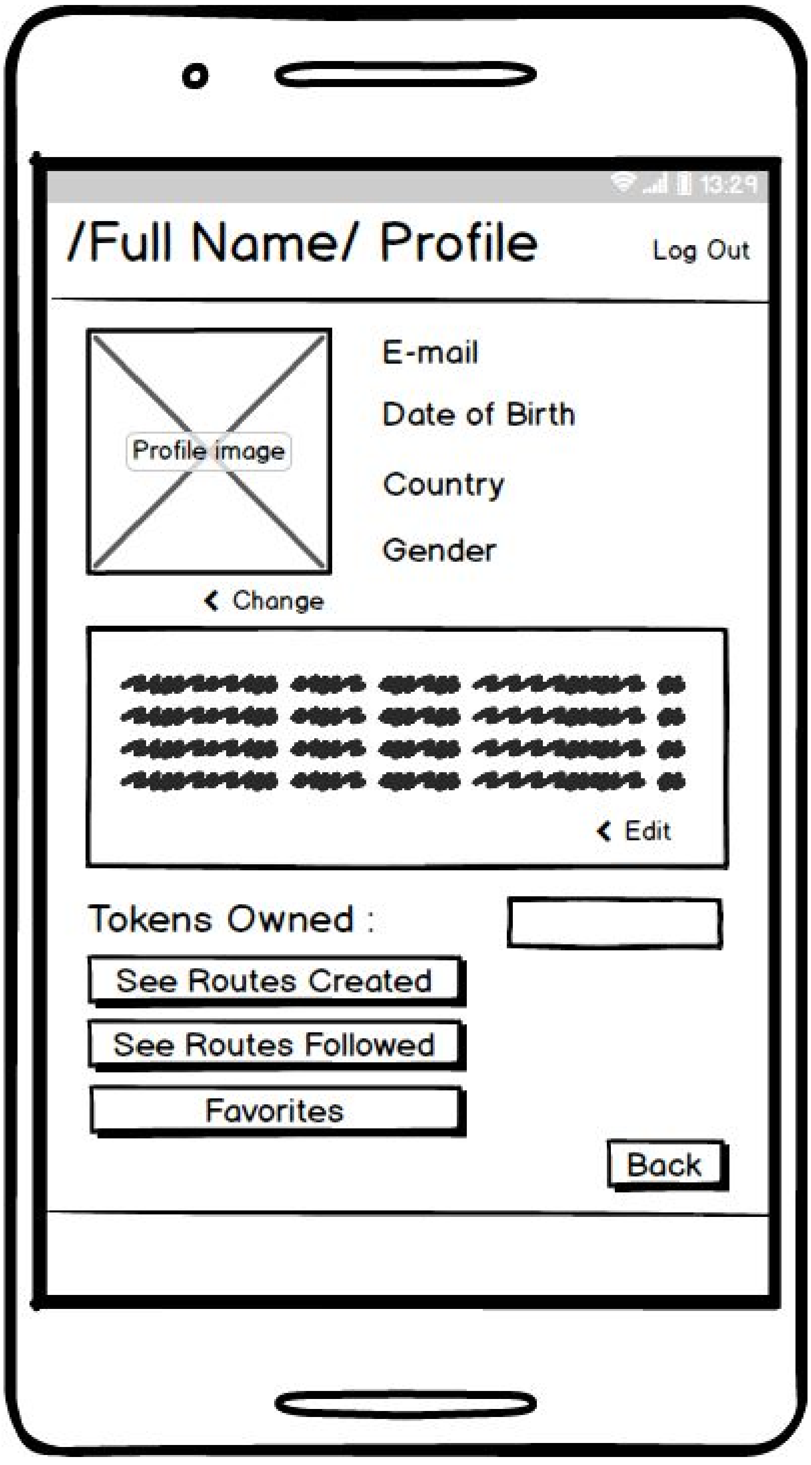
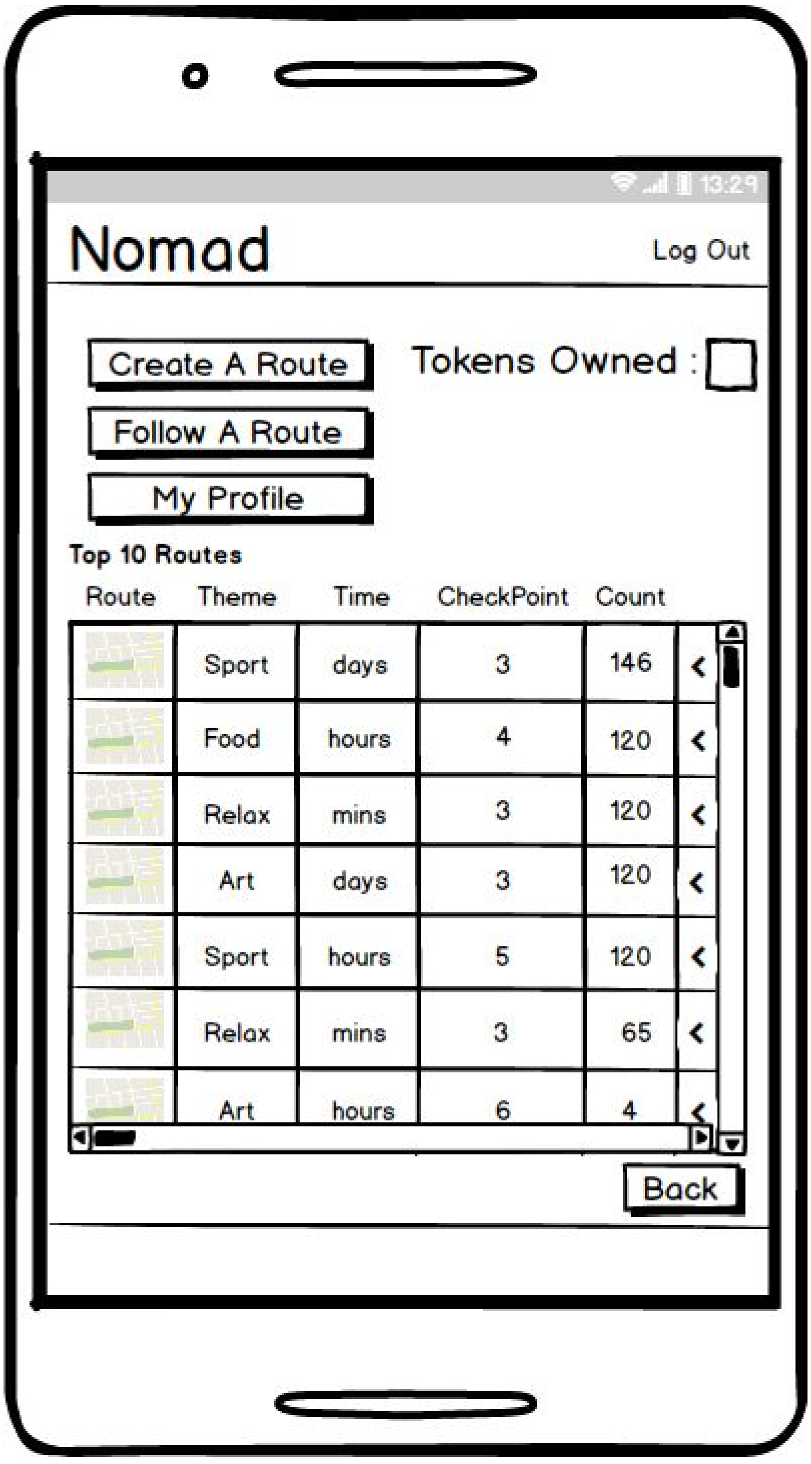


Figure 13: Profile Page

User can see their profile from their profile page. In addition sign up info, they can see their favorite routes, routes followed/created.

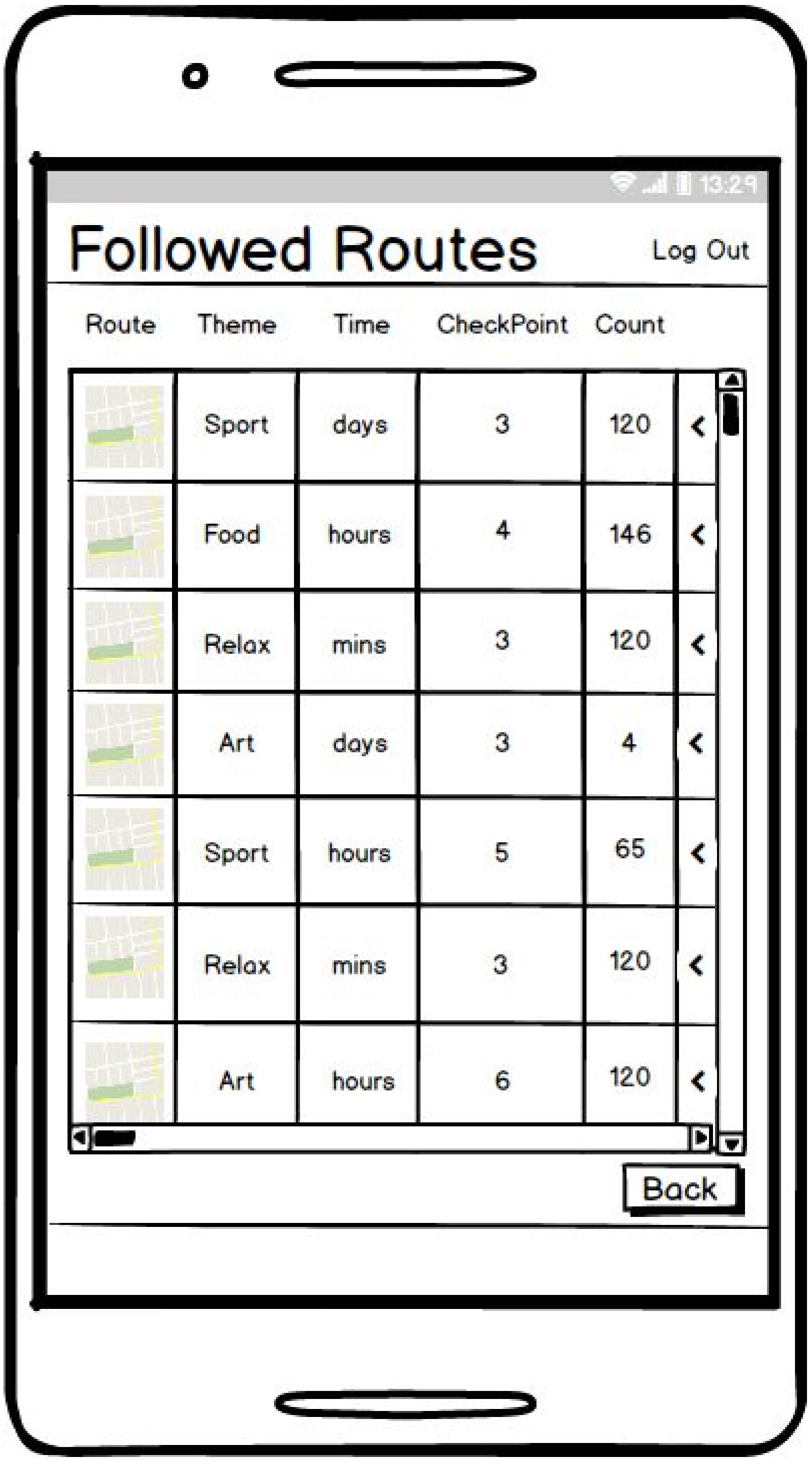


Top 10 Routes

Route	Theme	Time	CheckPoint	Count
	Sport	days	3	146
	Food	hours	4	120
	Relax	mins	3	120
	Art	days	3	120
	Sport	hours	5	120
	Relax	mins	3	65
	Art	hours	6	4

Figure 14: Main Page After Login (Routes)

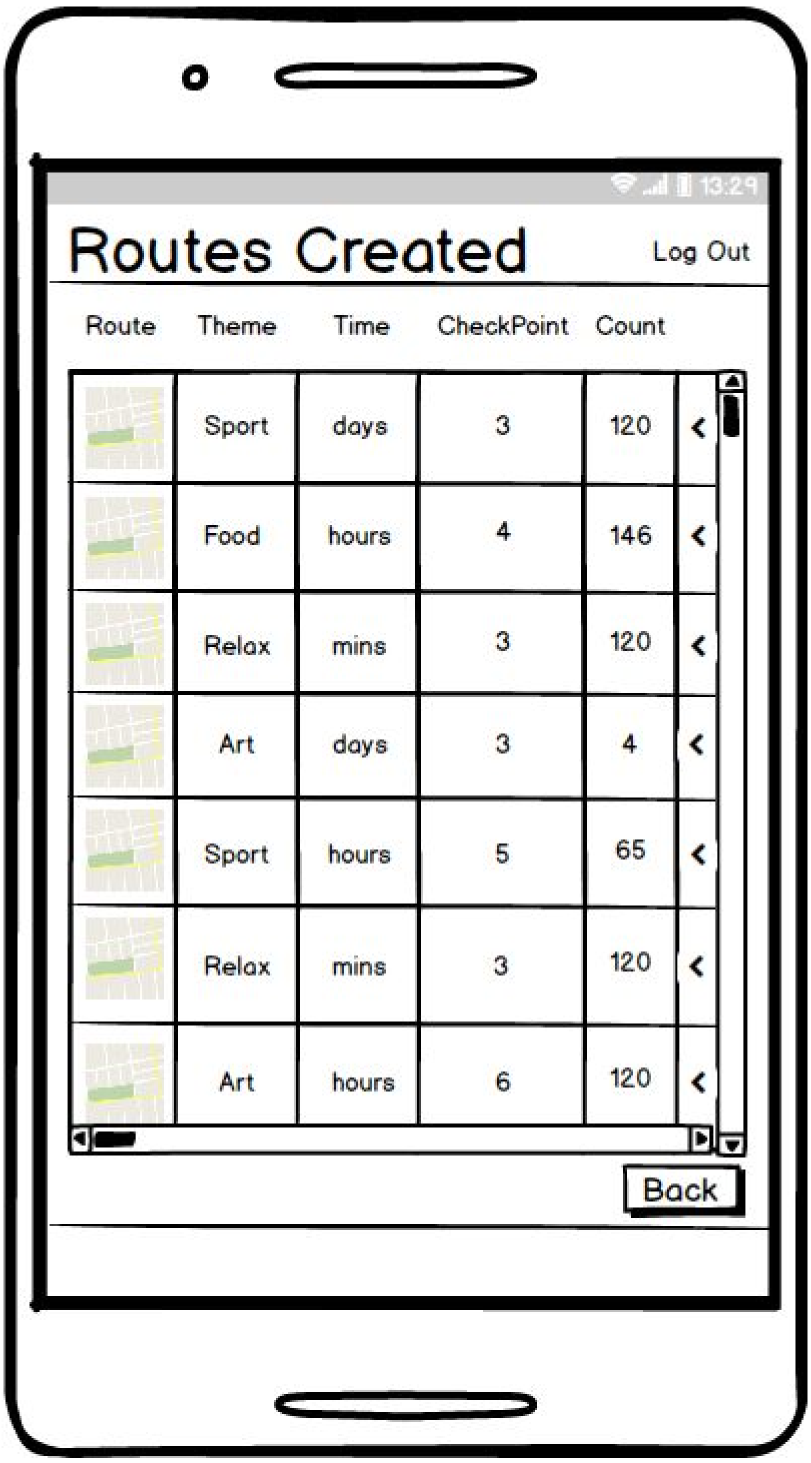
In the main page routes will be showed according to the location. User can search a route, create a route or follow a route from this page.



Route	Theme	Time	CheckPoint	Count	
	Sport	days	3	120	<
	Food	hours	4	146	<
	Relax	mins	3	120	<
	Art	days	3	4	<
	Sport	hours	5	65	<
	Relax	mins	3	120	<
	Art	hours	6	120	<

Figure 15: Followed Routes Screen

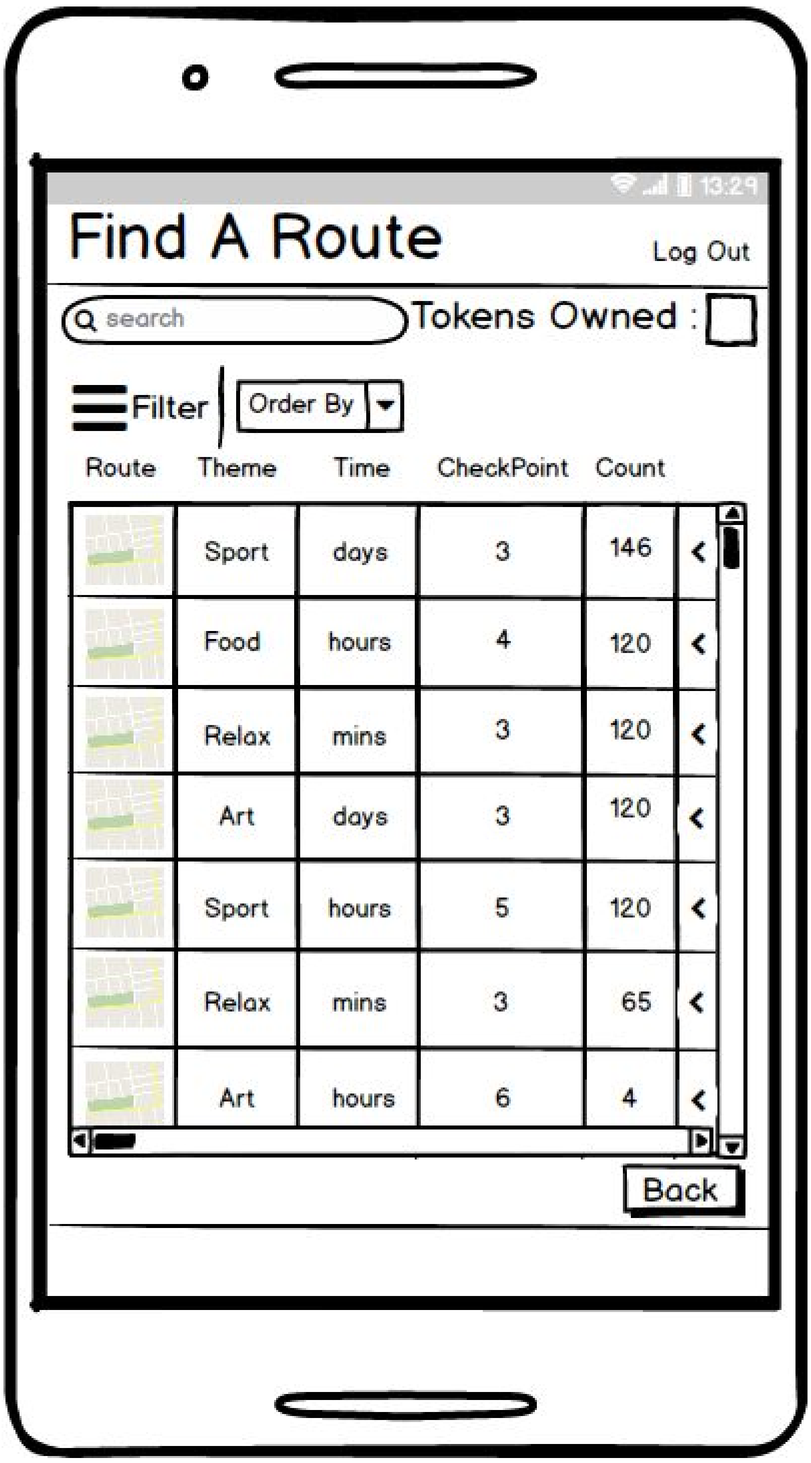
Followed routes can be seen from profile page. Every route has common info such as theme, time, checkpoints and count (how many people followed them).



Route	Theme	Time	CheckPoint	Count
	Sport	days	3	120
	Food	hours	4	146
	Relax	mins	3	120
	Art	days	3	4
	Sport	hours	5	65
	Relax	mins	3	120
	Art	hours	6	120

Figure 16: Routes Created Screen

Created routes can be seen from profile page. User can edit them later.



Route	Theme	Time	CheckPoint	Count	
	Sport	days	3	146	<
	Food	hours	4	120	<
	Relax	mins	3	120	<
	Art	days	3	120	<
	Sport	hours	5	120	<
	Relax	mins	3	65	<
	Art	hours	6	4	<

Figure 17: Search Route Screen

User can search a route by filter or by a word. Routes can be sorted by according to filters. In this page tokens owned will be shown to user.

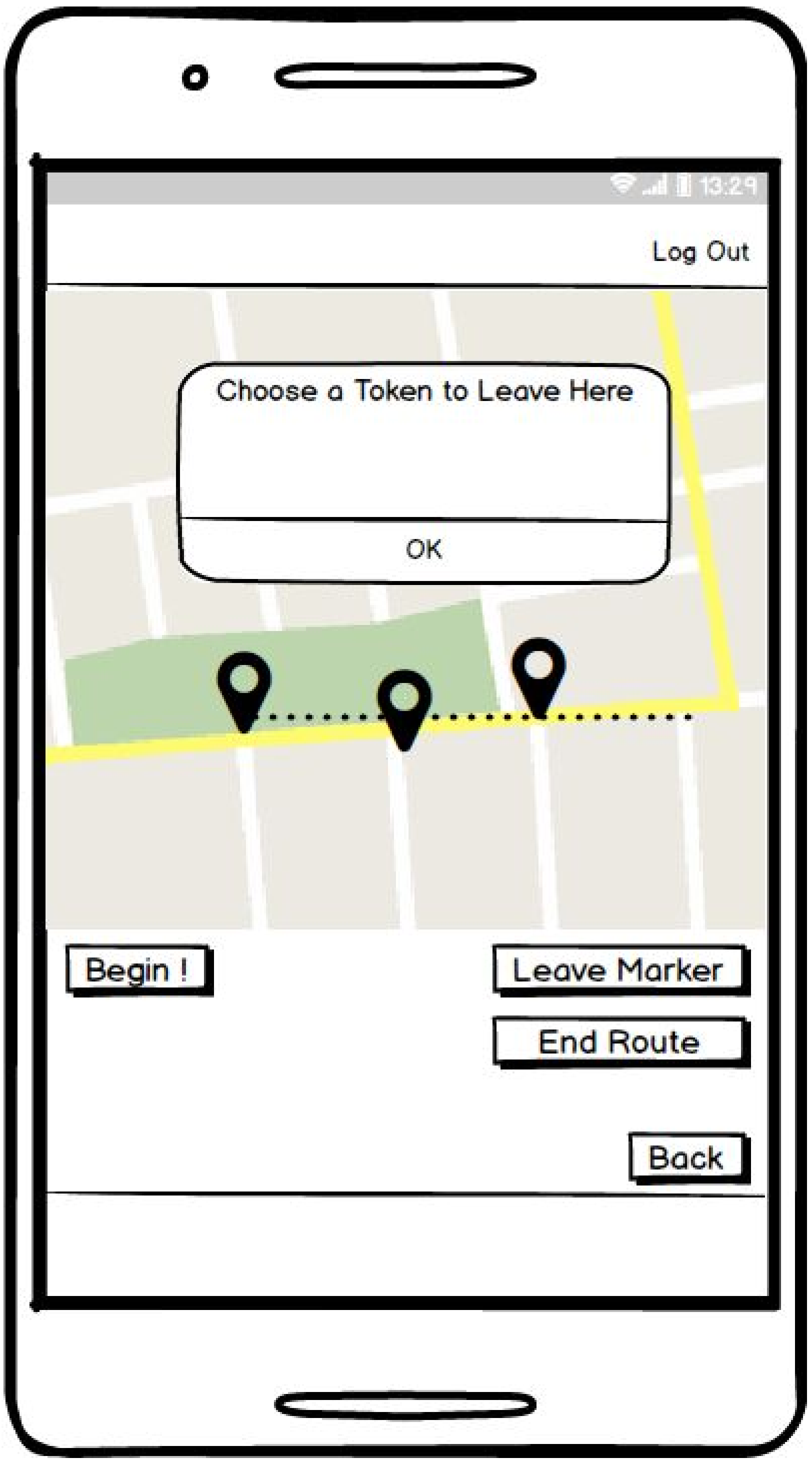


Figure 18: Creating route screen

When user create a route, s/he can leave markers to some places. These places can be checkpoints later, according to user decision.

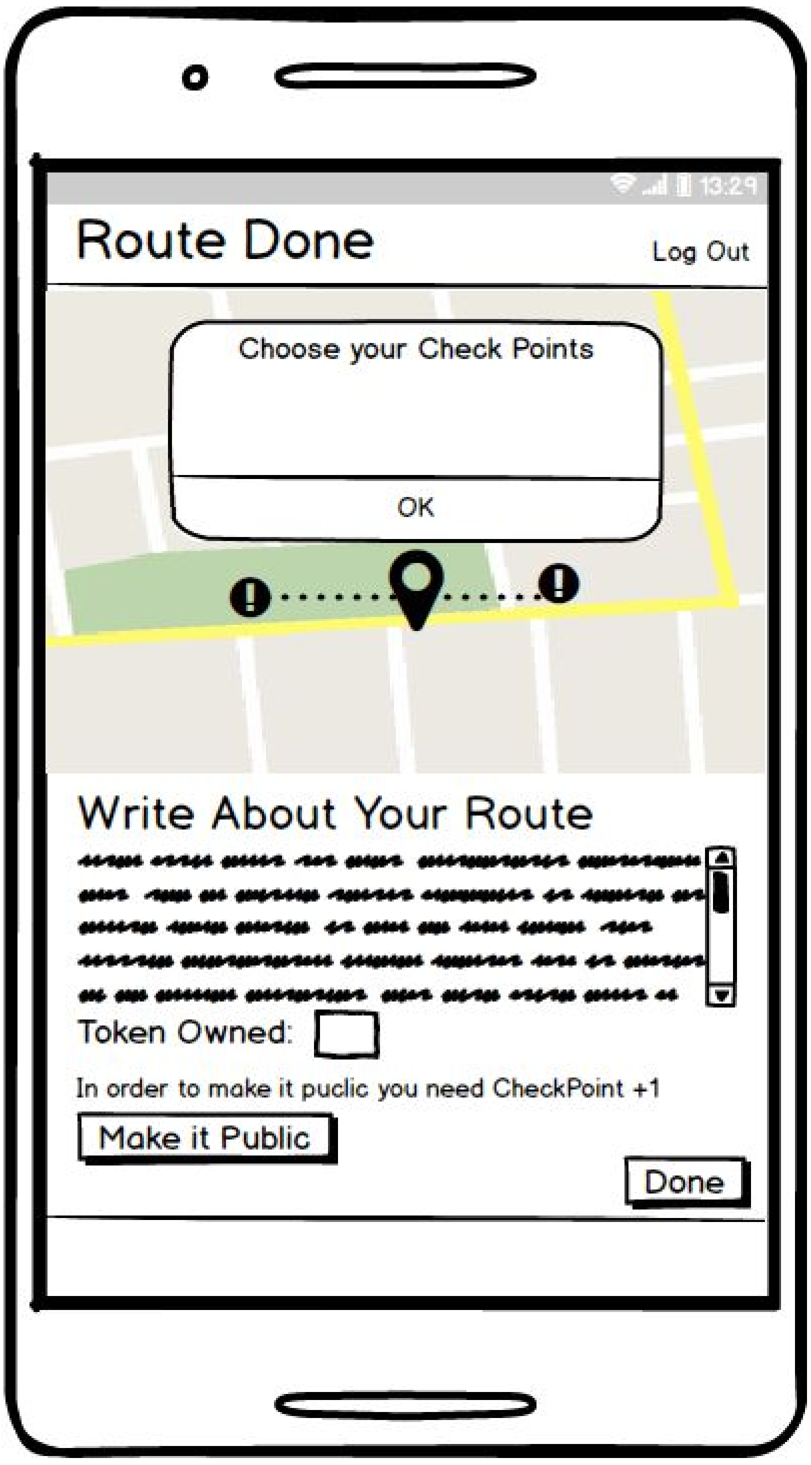


Figure 19: Route Done/Submit Screen

When route creation is done, user can write info about that route and can make it public by spending tokens.

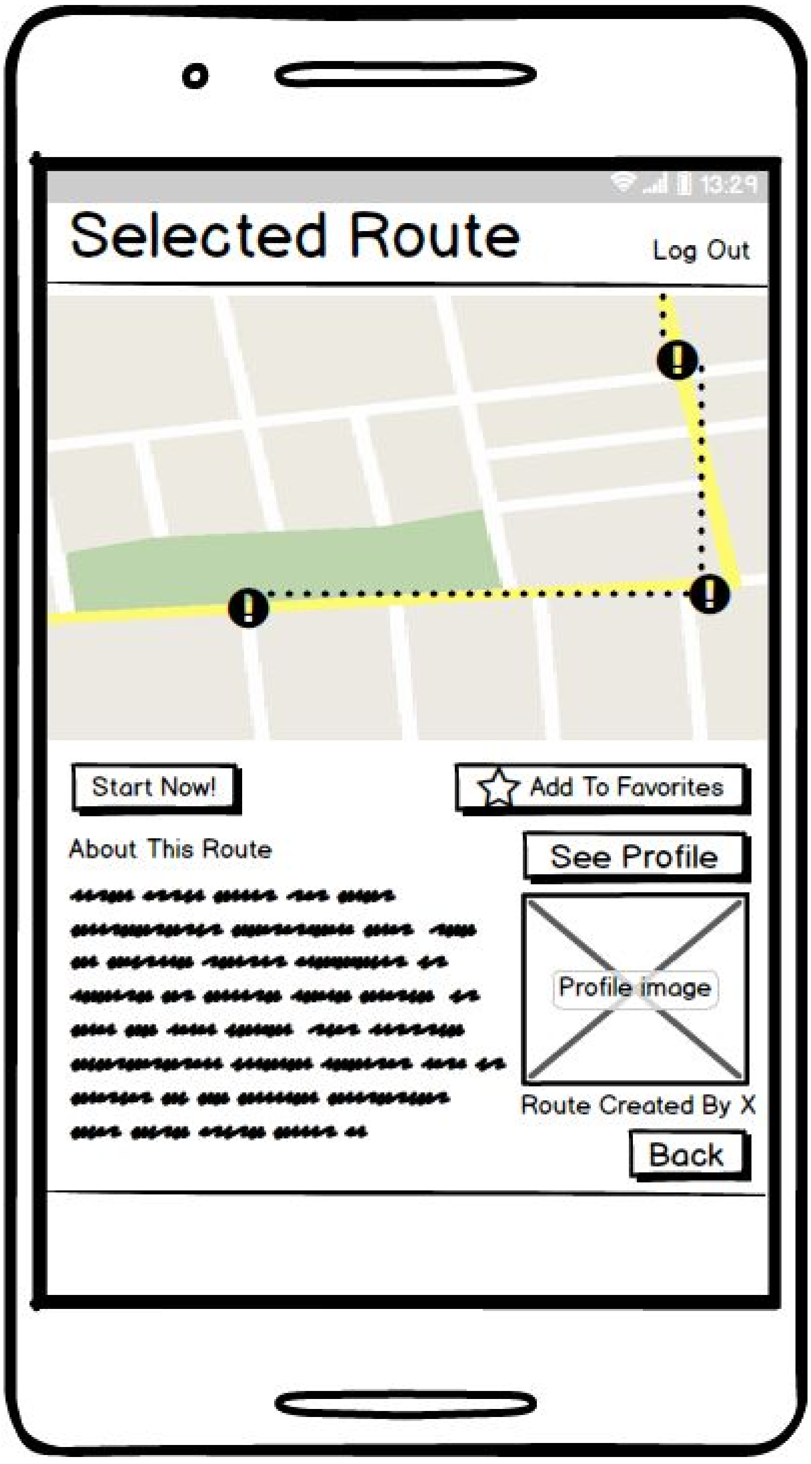


Figure 20: Route Info Screen

Users can follow a route. This is the route info screen. User info, route info, add to favorites option, start option can be seen from this page.

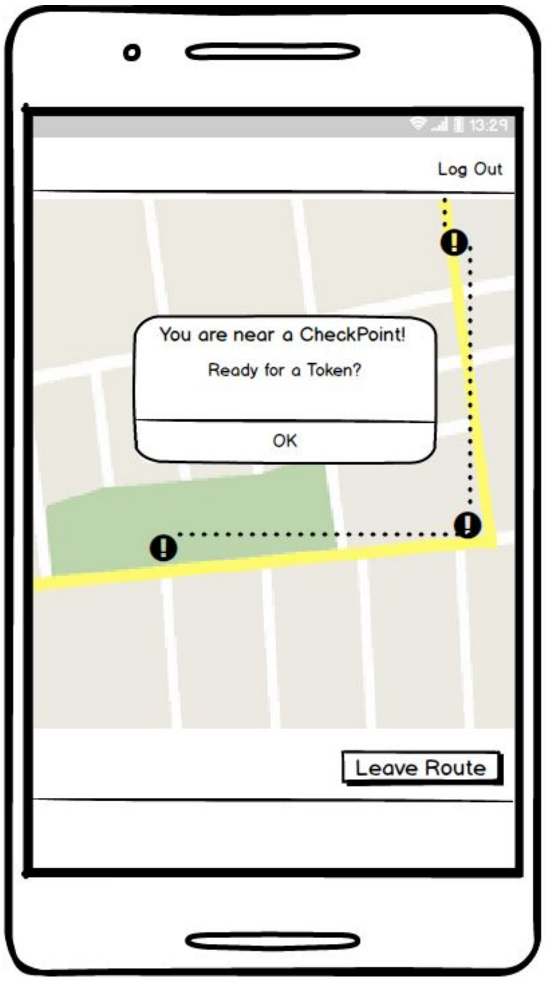


Figure 21: Warning for a checkpoint screen

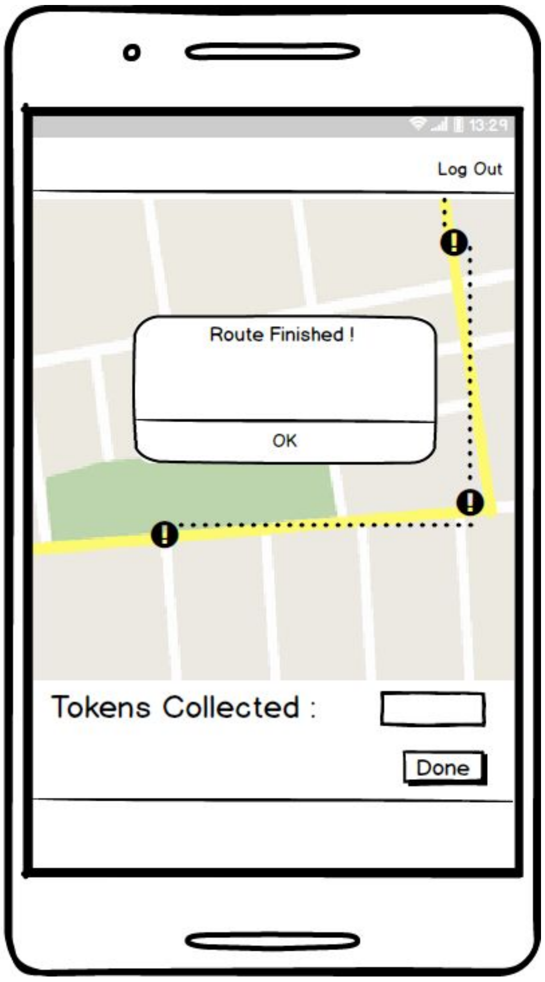


Figure 22: Route Finished Info Screen

When a user following a route, nearest checkpoint will be seen by warning in the app. If a route is finished, pop-up message and collected tokens will be shown.

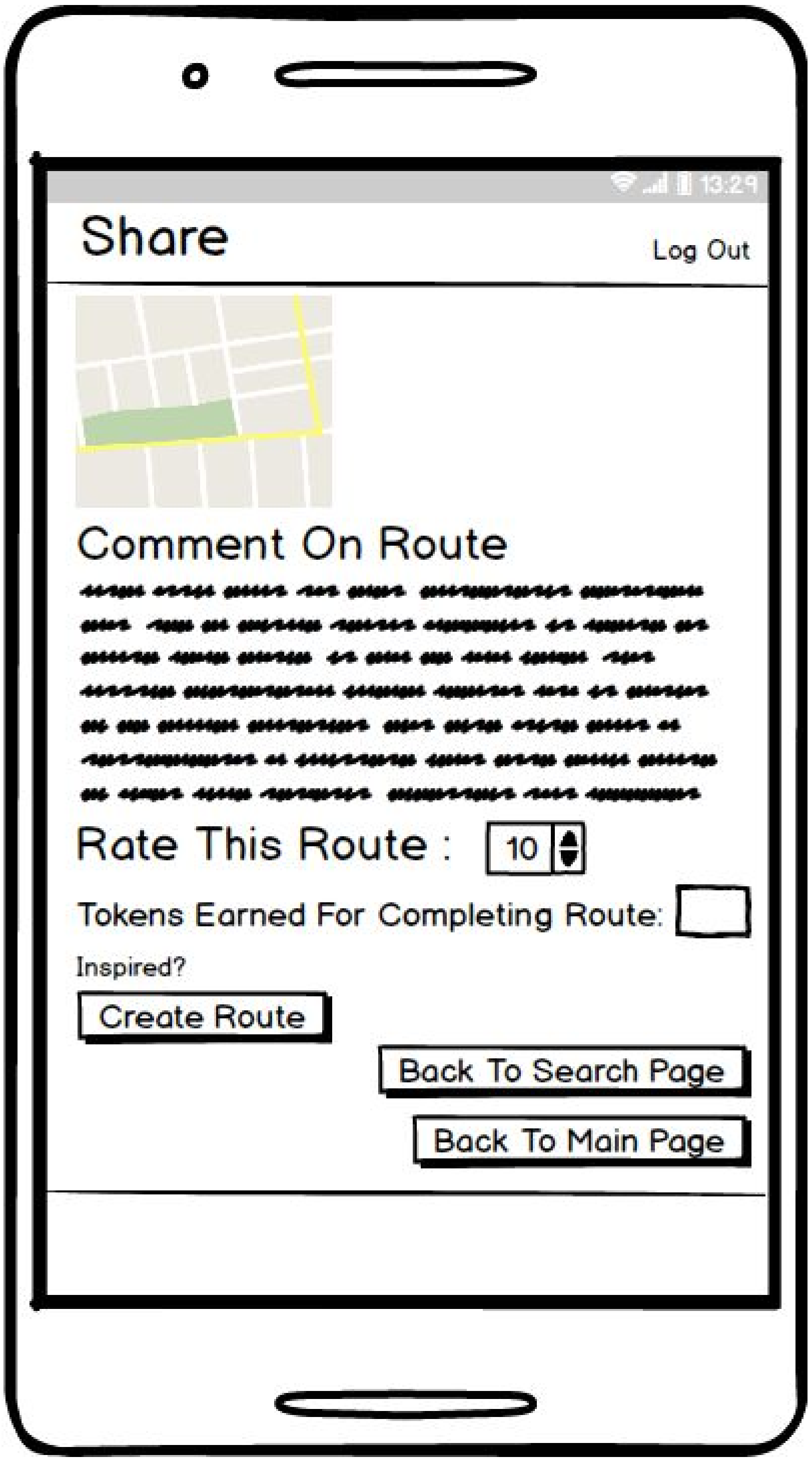


Figure 23: Comment & Rate for a route screen

After a route is finished, user can comment on route and can give vote out of 10. Tokens earned will be shown after completing a route.

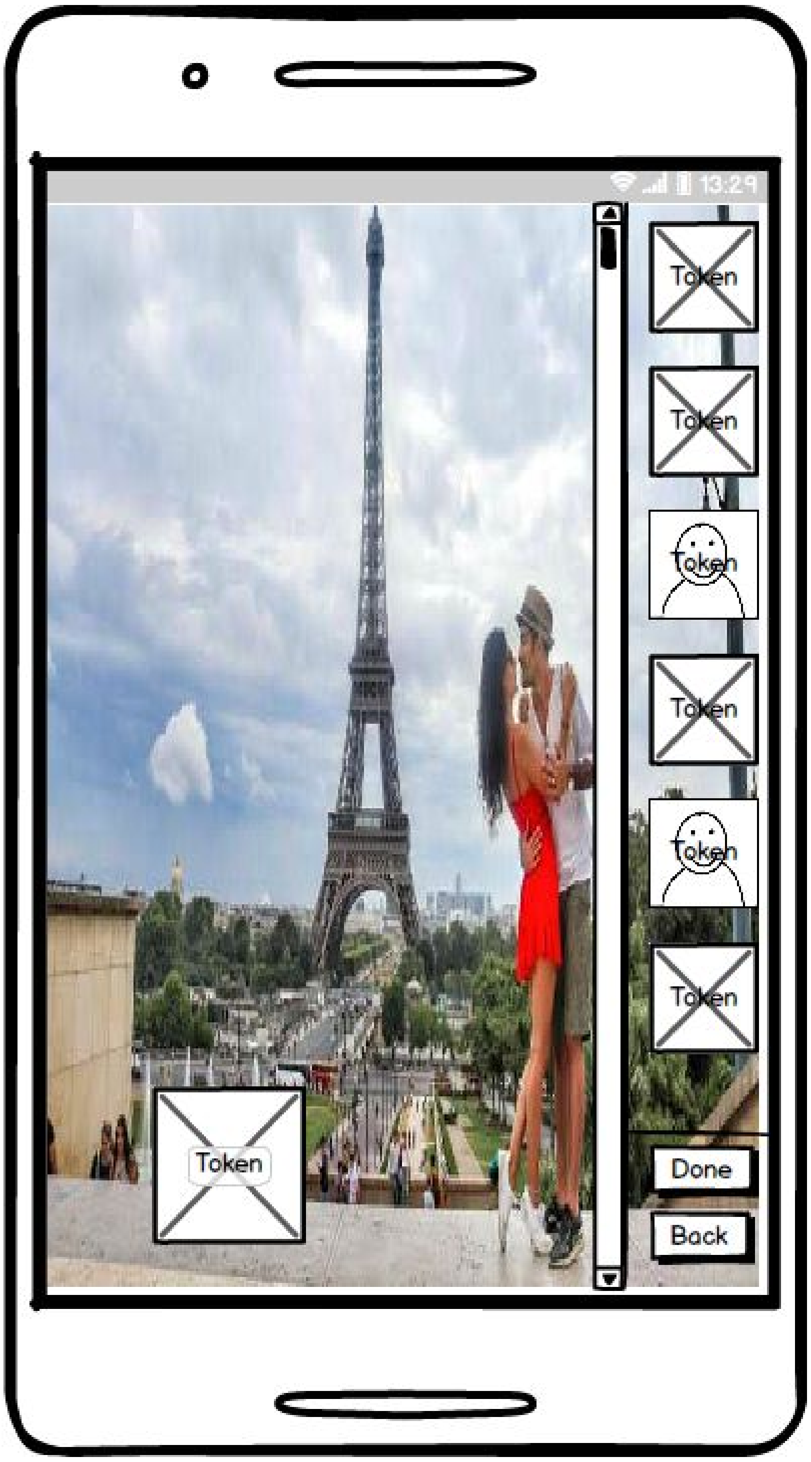


Figure 24: Leave Token Screen

When route is created, user can leave tokens to places. These tokens will be collected later by people following that route. This is the page that AR is used.

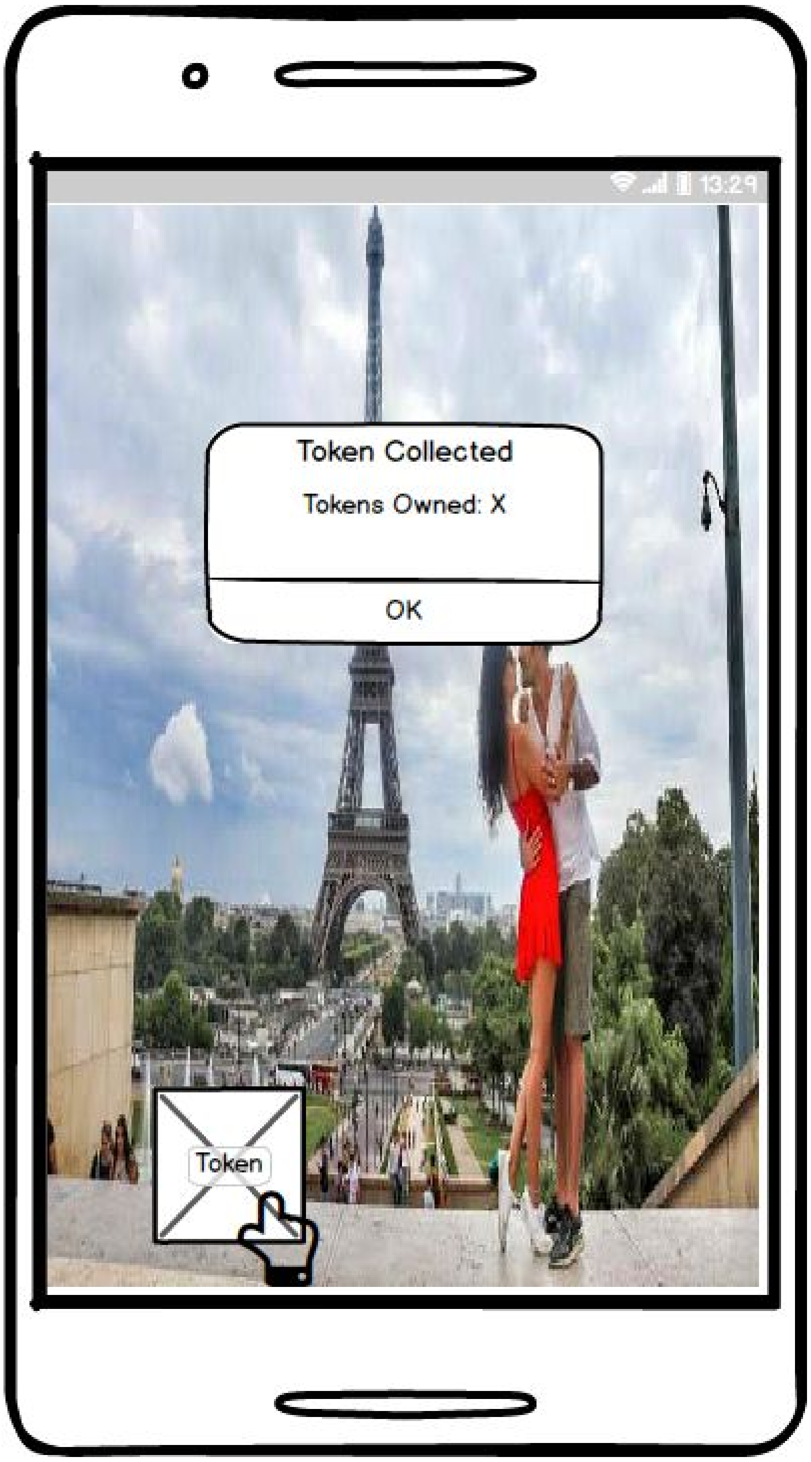


Figure 25: Collect Token Screen

User can collect token from places when following a route. AR will be used to collect tokens to make real life experience.

4. Other Analysis Elements

4.1. Consideration of Various Factors

4.1.1. Adaptation to massive data

Our application can reach a very large scale. Since the created routes can be published, the users can see all of the routes in the world. This large scale creates the problem of storage. It affects both maintainability and sustainability.

4.1.1.1. Safety

This constraint is very important. When they are creating routes, their GPS will always open during that time and people may feel suspicious about that since they can be known by someone else in that time. It can be really hard to get user's trust. In order to give that feeling, we may need to take further security for our application. If the increasing amount of data causes such security problems, we may lose trust. In order to keep it safe, we may need to think of optimization for data storage.

4.1.1.2. Global

Since our search criteria can include other nations too, it can be used globally. If a tourist wants a good experience, s/he may want popular routes of that country and our application will give these routes based on popularity. After searching among those options, the tourist can decide on the best routes for her/him. The details such as cost, time and various touristic places can be given in the information section of the routes. In that sense, our application can be used globally. The results can be good for maintainability but may not be good for sustainability.

4.1.1.3. Economic

Adaptation to data will require money for storage. It becomes harder to control something that is growing because the parts cannot be treated equally. Any new additional feature will be needed for maintenance but its sustainability can be affected if the effect of maintenance will be less than sustainability. If we try to restrict the sustainability, in that case, our changes may not be that satisfying for users.

4.1.2. Efficient usage in terms of time while searching and downloading the path

The data to store a route is relatively high for other objects in our system. If the amount of data increases, the cost of searching increases too. Creating a route is relatively simple. In the user side of view, it is just a path and some details about route such as notes, checkpoints, and location information. However, we planned to store the path by sampling the path with GPS. By that information, we can recreate the route in user's device.

4.1.2.1. Economic

The amount of data storage unit will be highly dependent on the sampling rate of the paths. If we sample them in half of the previous frequency, the data is doubled so the potential storage units will be doubled. In that sense, economy will be a powerful constraint.

4.1.2.2. Global

If the application serves everyone from just one server, the efficiency will be dramatically fall. Downloading the content is problematic already and if the people from all around the world try to use our application, this problem is getting bigger and bigger.

4.1.3. Unfairly taking advantage of collecting tokens

Our application should be bug free and take precautions to eliminate the unfair advantage of collecting tokens. It is important because it can affect the trust of users. In that situation, the decrease in the number of users would probably be high.

4.1.3.1. Social

If the trust of society is lost just one time, it would be very hard to regain it. However, sometimes no matter how hard you try to eliminate it, they can happen.

4.1.3.2. Environmental

If a bug happens and there are some routes that are more advantageous during that bug, these roads can be used excessive amounts. Unfortunately, if there is people, it means that there is also pollution. These people may pollute those routes by throwing garbage in the simplest sense.

4.1.3.3. User’s tolerance

One possible solution that comes to my mind is to block the entrance to the application or make the tokens valueless. But this solution can change the perception of the users to our application before the bug and after the solution. They can think that the application is too amateur and decide not to use it.

Judgement Description: **Adapting to massive data can be problematic.**		
	Impact Level	**Impact Description**
Impact in Safety	8	Safety in bigger systems become harder to establish.
Impact in Global	5	Such a great scale will make sustainability harder.
Impact in Economic	7	Purchasing new data storage units can be costly.

Judgement Description: **Time efficiency for searching and downloading**		
	Impact Level	**Impact Description**
Impact in Global	5	Number of servers is important for that problem.
Impact in Economic	8	The cost of new servers within a country can be a problem.

Judgement Description: **Bugs about collecting tokens may occur**		
	Impact Level	**Impact Description**
Social	9	People may lose their trust on our application.

Environmental	8	If they use a beneficial bugged route, people may pollute there.
User’s tolerance	9	Users may lose their tolerance because of this bug.

4.2. Risks and Alternatives

At any point during our project’s lifecycle, some problems that will have an effect on our project’s progress may occur, but we can foresee some of those problems before they occur, which we call them risks. We have identified and analysed some risks that may occur in our project process, and we also proposed some techniques to deal with such risks. To meet significant requirements and to provide a satisfactory end result no matter what happens, we also proposed an alternative “B plan” to carry out in the case that we might not overcome the risks that we have foreseen, or unexpected problems that significantly impact our project’s progress occur.

Risk Analysis

We analysed our risks regarding to three main criteria: probability, impact, and priority. Probability criterion tells about the risks likeliness to occur, impact criterion tells about the negative effect of the risk on our project’s progress in the case that it occurs, and the priority criterion tells about how much should we consider dealing with a risk above others.

Risk Prevention

To deal with risks we will follow three main techniques. We may avoid risks to basically prevent them from occurring in the first place, we may mitigate the effects of risks to cause them to have lower impact on our progress in the case that they occur, or we may transfer the risks to another source so that it’s no longer a risk concerning us.

Following are the risks that we have identified.

1. **Wrong Scoping and Scheduling**
   Probability: High
   Impact: High, end result will not meet some of the requirements, and therefore may be deemed unsatisfactory
   Priority: High
   Prevention (Mitigate): Monitoring other similar existing projects and inspect their scheduling to have a better idea and understanding of a realistic estimation, and come up with a new project plan regarding the remaining progress to be completed.

2. **A Group Member Leaving the Project**
   Probability: Medium
   Impact: Medium to High, workload of remaining members will significantly increase
   Priority: Medium to High
   Prevention (Avoid): Redistribution of workload is a solution to prevent the member from leaving if the member's reason for wanting to leave is about the work they are assigned.
   Prevention (Mitigate): Redistribution of workload between the remaining members or coming up with a new Project Plan are some solutions to keep the project going to its end.

3. **Technologies Used are Changed Significantly**
   Probability: Low
   Impact: Very High, after we have started implementing our project, if the technologies we decide to use are changed so much that they will no longer serve any use to us, it will cause us to take extreme

measures and will significantly change the course of our project's progress.
Priority: Low to Medium
Prevention (Avoid): Prior to our implementation, identifying technologies that are either not going through a lot of changes, or allow use of old versions, and later using them is a very optimal solution.
Prevention (Transfer): We may decide to comply with changes and update our requirements accordingly. As long as the customer's will be happy with the updated functionalities of our new product, this is a potential solution.

4. **Change in Requirements**
   Probability: Very High
   Impact: Low, since we have adopted a project process that has significant agile characteristics, we can adapt to changes. (More on this in the next section, Project Plan).
   Priority: Low
   Prevention (Mitigation): We will adapt to the changed requirements by analysing them as soon as possible and we will plan our next iteration according to these updated requirements so our design and implementation will comply to these changes.

5. **Loss of Critical Information**
   Probability: Low
   Impact: Low to High, Depends on the size of lost info
   Priority: Very Low
   Prevention (Avoid): Since we will upload our project to Github, this issue won't be a problem very much, because Github already has a well established distributed version control system. In the event that we lose some of our progress, we can easily rollback to a recent version.

6. **<u>Excess Creation of Accounts by Specific Users</u>**

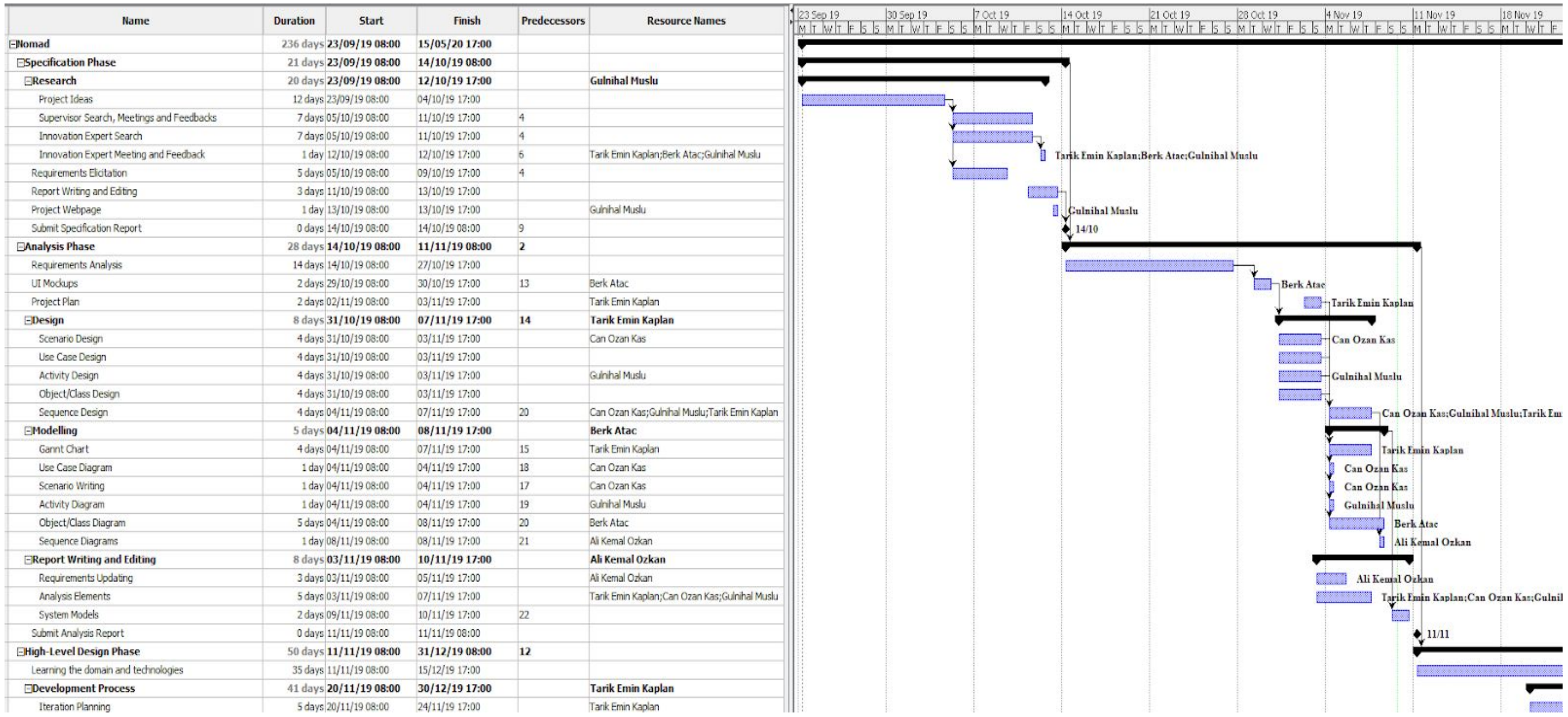
   Probability: Very Low (at this point)

   Impact: Medium to High,

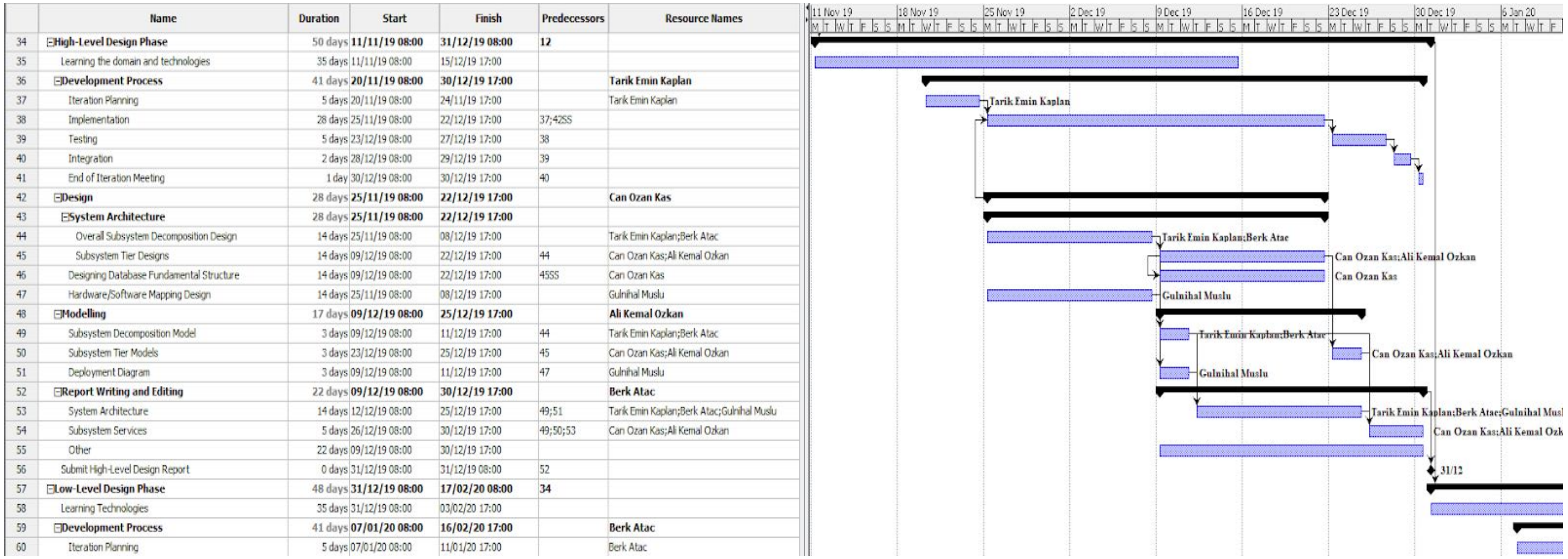
   Priority: Very Low

   Prevention (Transfer): For first time users, we will allow logging in to the app (and therefore, creation of an account) via social media platforms (google, facebook, and potentially twitter). Therefore, a user that would want to create many accounts in our app will have to first create many social media accounts, which is a hassle for most users.

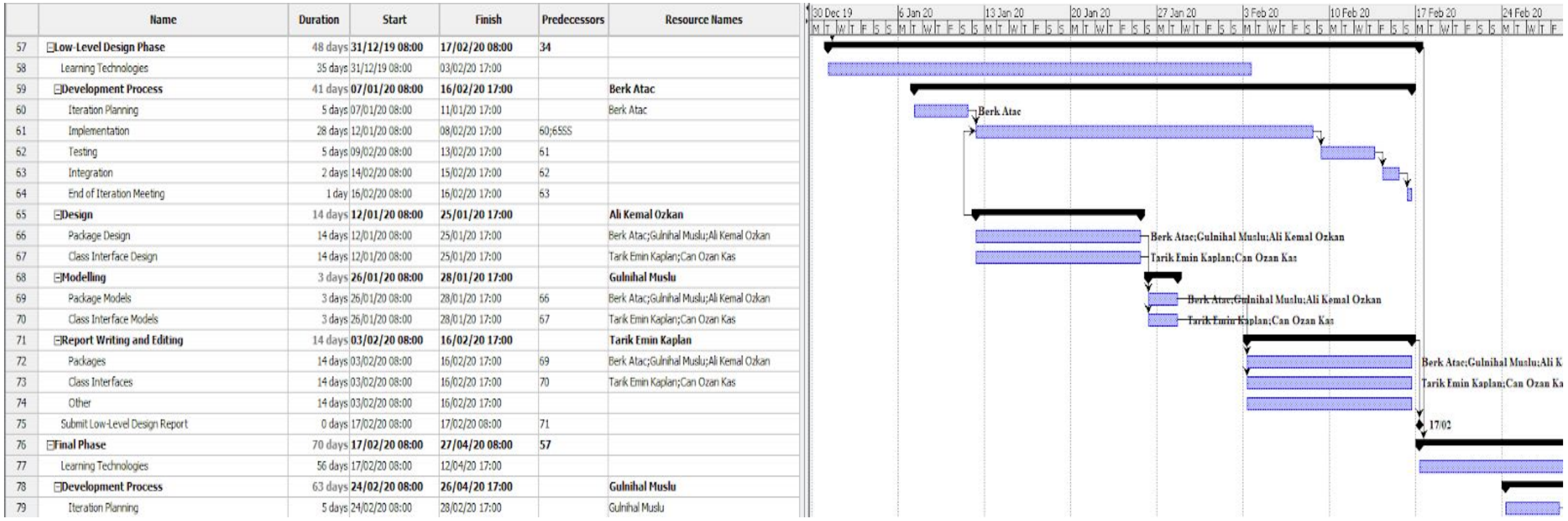
4.3. Project Plan



Name	Duration	Start	Finish	Predecessors	Resource Names
Nomad	236 days	**23/09/19 08:00**	**15/05/20 17:00**		
Specification Phase	21 days	**23/09/19 08:00**	**14/10/19 08:00**		
Research	20 days	**23/09/19 08:00**	**12/10/19 17:00**		**Gulnihal Muslu**
Project Ideas	12 days	23/09/19 08:00	04/10/19 17:00		
Supervisor Search, Meetings and Feedbacks	7 days	05/10/19 08:00	11/10/19 17:00	4	
Innovation Expert Search	7 days	05/10/19 08:00	11/10/19 17:00	4	
Innovation Expert Meeting and Feedback	1 day	12/10/19 08:00	12/10/19 17:00	6	Tarik Emin Kaplan;Berk Atac;Gulnihal Muslu
Requirements Elicitation	5 days	05/10/19 08:00	09/10/19 17:00	4	
Report Writing and Editing	3 days	11/10/19 08:00	13/10/19 17:00		
Project Webpage	1 day	13/10/19 08:00	13/10/19 17:00		Gulnihal Muslu
Submit Specification Report	0 days	14/10/19 08:00	14/10/19 08:00	9	
Analysis Phase	28 days	**14/10/19 08:00**	**11/11/19 08:00**	**2**	
Requirements Analysis	14 days	14/10/19 08:00	27/10/19 17:00		
UI Mockups	2 days	29/10/19 08:00	30/10/19 17:00	13	Berk Atac
Project Plan	2 days	02/11/19 08:00	03/11/19 17:00		Tarik Emin Kaplan
Design	8 days	**31/10/19 08:00**	**07/11/19 17:00**	**14**	**Tarik Emin Kaplan**
Scenario Design	4 days	31/10/19 08:00	03/11/19 17:00		Can Ozan Kas
Use Case Design	4 days	31/10/19 08:00	03/11/19 17:00		
Activity Design	4 days	31/10/19 08:00	03/11/19 17:00		Gulnihal Muslu
Object/Class Design	4 days	31/10/19 08:00	03/11/19 17:00		
Sequence Design	4 days	04/11/19 08:00	07/11/19 17:00	20	Can Ozan Kas;Gulnihal Muslu;Tarik Emin Kaplan
Modelling	5 days	**04/11/19 08:00**	**08/11/19 17:00**		**Berk Atac**
Gannt Chart	4 days	04/11/19 08:00	07/11/19 17:00	15	Tarik Emin Kaplan
Use Case Diagram	1 day	04/11/19 08:00	04/11/19 17:00	18	Can Ozan Kas
Scenario Writing	1 day	04/11/19 08:00	04/11/19 17:00	17	Can Ozan Kas
Activity Diagram	1 day	04/11/19 08:00	04/11/19 17:00	19	Gulnihal Muslu
Object/Class Diagram	5 days	04/11/19 08:00	08/11/19 17:00	20	Berk Atac
Sequence Diagrams	1 day	08/11/19 08:00	08/11/19 17:00	21	Ali Kemal Ozkan
Report Writing and Editing	8 days	**03/11/19 08:00**	**10/11/19 17:00**		**Ali Kemal Ozkan**
Requirements Updating	3 days	03/11/19 08:00	05/11/19 17:00		Ali Kemal Ozkan
Analysis Elements	5 days	03/11/19 08:00	07/11/19 17:00		Tarik Emin Kaplan;Can Ozan Kas;Gulnihal Muslu
System Models	2 days	09/11/19 08:00	10/11/19 17:00	22	
Submit Analysis Report	0 days	11/11/19 08:00	11/11/19 08:00		
High-Level Design Phase	50 days	**11/11/19 08:00**	**31/12/19 08:00**	**12**	
Learning the domain and technologies	35 days	11/11/19 08:00	15/12/19 17:00		
Development Process	41 days	**20/11/19 08:00**	**30/12/19 17:00**		**Tarik Emin Kaplan**
Iteration Planning	5 days	20/11/19 08:00	24/11/19 17:00		Tarik Emin Kaplan

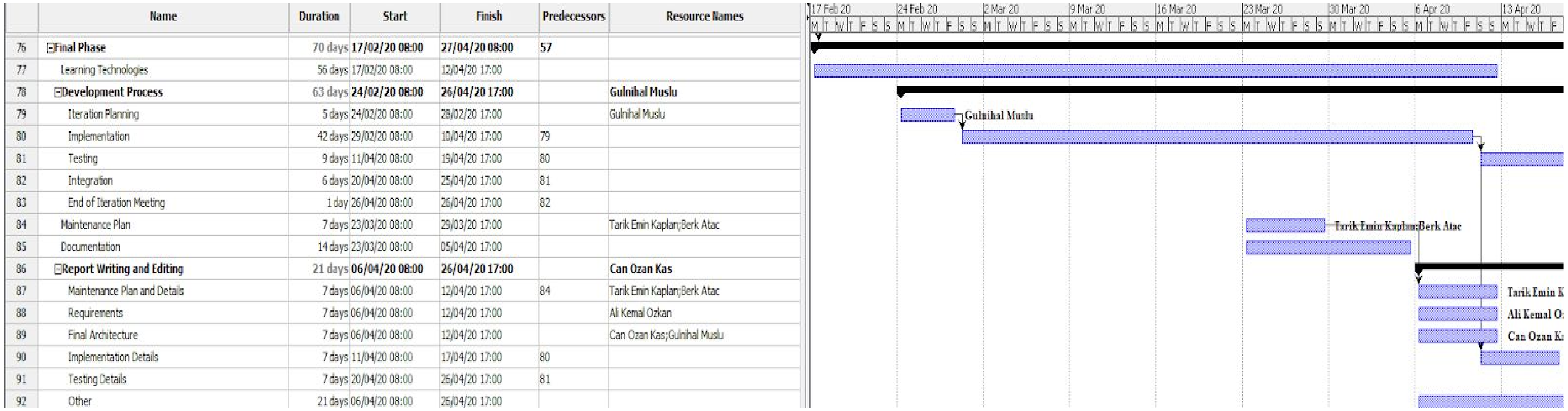

11 Nov 19
18 Nov 19
25 Nov 19
2 Dec 19
9 Dec 19
16 Dec 19
23 Dec 19
30 Dec 19
6 Jan 20
Tarik Emin Kaplan
Tarik Emin Kaplan;Berk Atac
Can Ozan Kas;Ali Kemal Ozkan
Can Ozan Kas
Gulnihal Muslu
Tarik Emin Kaplan;Berk Atac
Can Ozan Kas;Ali Kemal Ozkan
Gulnihal Muslu
Tarik Emin Kaplan;Berk Atac;Gulnihal Musl
Can Ozan Kas;Ali Kemal Ozk
31/12

	Name	Duration	Start	Finish	Predecessors	Resource Names
34	**High-Level Design Phase**	50 days	**11/11/19 08:00**	**31/12/19 08:00**	**12**	
35	Learning the domain and technologies	35 days	11/11/19 08:00	15/12/19 17:00		
36	**Development Process**	41 days	**20/11/19 08:00**	**30/12/19 17:00**		**Tarik Emin Kaplan**
37	Iteration Planning	5 days	20/11/19 08:00	24/11/19 17:00		Tarik Emin Kaplan
38	Implementation	28 days	25/11/19 08:00	22/12/19 17:00	37;42SS	
39	Testing	5 days	23/12/19 08:00	27/12/19 17:00	38	
40	Integration	2 days	28/12/19 08:00	29/12/19 17:00	39	
41	End of Iteration Meeting	1 day	30/12/19 08:00	30/12/19 17:00	40	
42	**Design**	28 days	**25/11/19 08:00**	**22/12/19 17:00**		**Can Ozan Kas**
43	**System Architecture**	28 days	**25/11/19 08:00**	**22/12/19 17:00**		
44	Overall Subsystem Decomposition Design	14 days	25/11/19 08:00	08/12/19 17:00		Tarik Emin Kaplan;Berk Atac
45	Subsystem Tier Designs	14 days	09/12/19 08:00	22/12/19 17:00	44	Can Ozan Kas;Ali Kemal Ozkan
46	Designing Database Fundamental Structure	14 days	09/12/19 08:00	22/12/19 17:00	45SS	Can Ozan Kas
47	Hardware/Software Mapping Design	14 days	25/11/19 08:00	08/12/19 17:00		Gulnihal Muslu
48	**Modelling**	17 days	**09/12/19 08:00**	**25/12/19 17:00**		**Ali Kemal Ozkan**
49	Subsystem Decomposition Model	3 days	09/12/19 08:00	11/12/19 17:00	44	Tarik Emin Kaplan;Berk Atac
50	Subsystem Tier Models	3 days	23/12/19 08:00	25/12/19 17:00	45	Can Ozan Kas;Ali Kemal Ozkan
51	Deployment Diagram	3 days	09/12/19 08:00	11/12/19 17:00	47	Gulnihal Muslu
52	**Report Writing and Editing**	22 days	**09/12/19 08:00**	**30/12/19 17:00**		**Berk Atac**
53	System Architecture	14 days	12/12/19 08:00	25/12/19 17:00	49;51	Tarik Emin Kaplan;Berk Atac;Gulnihal Muslu
54	Subsystem Services	5 days	26/12/19 08:00	30/12/19 17:00	49;50;53	Can Ozan Kas;Ali Kemal Ozkan
55	Other	22 days	09/12/19 08:00	30/12/19 17:00		
56	Submit High-Level Design Report	0 days	31/12/19 08:00	31/12/19 08:00	52	
57	**Low-Level Design Phase**	48 days	**31/12/19 08:00**	**17/02/20 08:00**	**34**	
58	Learning Technologies	35 days	31/12/19 08:00	03/02/20 17:00		
59	**Development Process**	41 days	**07/01/20 08:00**	**16/02/20 17:00**		**Berk Atac**
60	Iteration Planning	5 days	07/01/20 08:00	11/01/20 17:00		Berk Atac

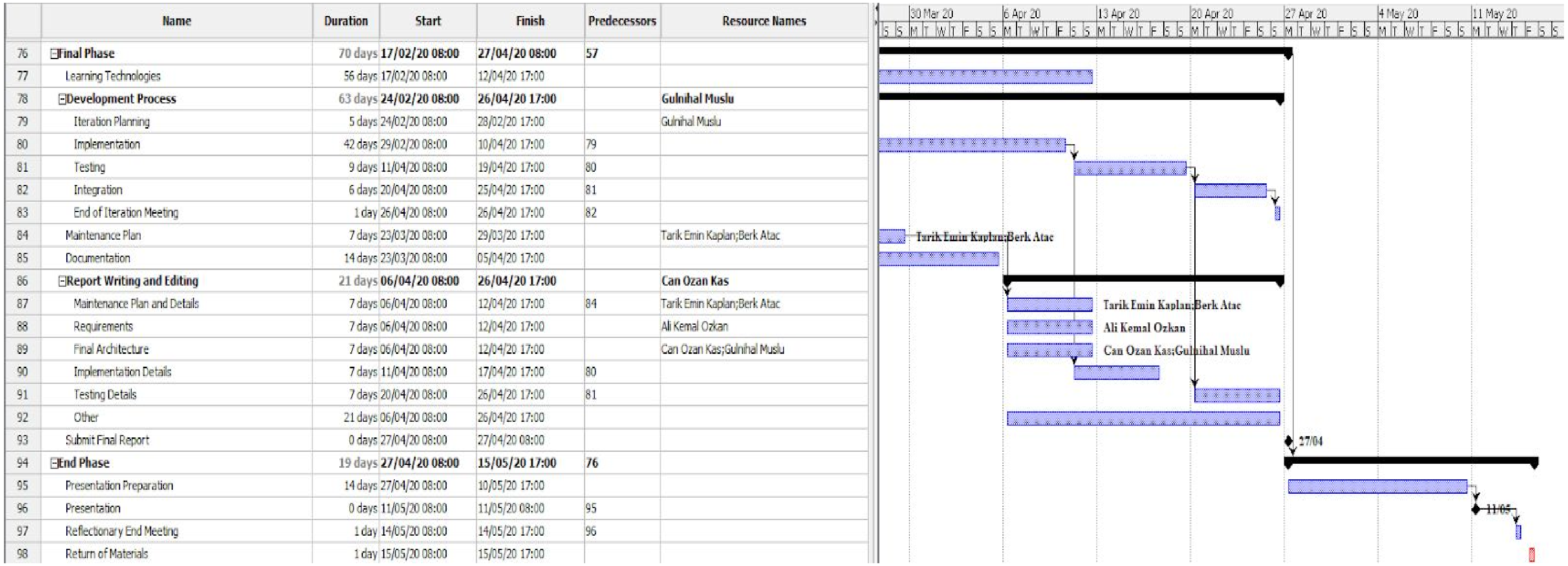


	Name	Duration	Start	Finish	Predecessors	Resource Names
57	**⊟Low-Level Design Phase**	48 days	**31/12/19 08:00**	**17/02/20 08:00**	**34**	
58	Learning Technologies	35 days	31/12/19 08:00	03/02/20 17:00		
59	**⊟Development Process**	41 days	**07/01/20 08:00**	**16/02/20 17:00**		**Berk Atac**
60	Iteration Planning	5 days	07/01/20 08:00	11/01/20 17:00		Berk Atac
61	Implementation	28 days	12/01/20 08:00	08/02/20 17:00	60;65SS	
62	Testing	5 days	09/02/20 08:00	13/02/20 17:00	61	
63	Integration	2 days	14/02/20 08:00	15/02/20 17:00	62	
64	End of Iteration Meeting	1 day	16/02/20 08:00	16/02/20 17:00	63	
65	**⊟Design**	14 days	**12/01/20 08:00**	**25/01/20 17:00**		**Ali Kemal Ozkan**
66	Package Design	14 days	12/01/20 08:00	25/01/20 17:00		Berk Atac;Gulnihal Muslu;Ali Kemal Ozkan
67	Class Interface Design	14 days	12/01/20 08:00	25/01/20 17:00		Tarik Emin Kaplan;Can Ozan Kas
68	**⊟Modelling**	3 days	**26/01/20 08:00**	**28/01/20 17:00**		**Gulnihal Muslu**
69	Package Models	3 days	26/01/20 08:00	28/01/20 17:00	66	Berk Atac;Gulnihal Muslu;Ali Kemal Ozkan
70	Class Interface Models	3 days	26/01/20 08:00	28/01/20 17:00	67	Tarik Emin Kaplan;Can Ozan Kas
71	**⊟Report Writing and Editing**	14 days	**03/02/20 08:00**	**16/02/20 17:00**		**Tarik Emin Kaplan**
72	Packages	14 days	03/02/20 08:00	16/02/20 17:00	69	Berk Atac;Gulnihal Muslu;Ali Kemal Ozkan
73	Class Interfaces	14 days	03/02/20 08:00	16/02/20 17:00	70	Tarik Emin Kaplan;Can Ozan Kas
74	Other	14 days	03/02/20 08:00	16/02/20 17:00		
75	Submit Low-Level Design Report	0 days	17/02/20 08:00	17/02/20 08:00	71	
76	**⊟Final Phase**	70 days	**17/02/20 08:00**	**27/04/20 08:00**	**57**	
77	Learning Technologies	56 days	17/02/20 08:00	12/04/20 17:00		
78	**⊟Development Process**	63 days	**24/02/20 08:00**	**26/04/20 17:00**		**Gulnihal Muslu**
79	Iteration Planning	5 days	24/02/20 08:00	28/02/20 17:00		Gulnihal Muslu

30 Dec 19
6 Jan 20
13 Jan 20
20 Jan 20
27 Jan 20
3 Feb 20
10 Feb 20
17 Feb 20
24 Feb 20
Berk Atac
Berk Atac;Gulnihal Muslu;Ali Kemal Ozkan
Tarik Emin Kaplan;Can Ozan Kas
Berk Atac;Gulnihal Muslu;Ali Kemal Ozkan
Tarik Emin Kaplan;Can Ozan Kas
Berk Atac;Gulnihal Muslu;Ali K
Tarik Emin Kaplan;Can Ozan Ka
17/02



	Name	Duration	Start	Finish	Predecessors	Resource Names
76	**Final Phase**	70 days	**17/02/20 08:00**	**27/04/20 08:00**	**57**	
77	Learning Technologies	56 days	17/02/20 08:00	12/04/20 17:00		
78	**Development Process**	63 days	**24/02/20 08:00**	**26/04/20 17:00**		**Gulnihal Muslu**
79	Iteration Planning	5 days	24/02/20 08:00	28/02/20 17:00		Gulnihal Muslu
80	Implementation	42 days	29/02/20 08:00	10/04/20 17:00	79	
81	Testing	9 days	11/04/20 08:00	19/04/20 17:00	80	
82	Integration	6 days	20/04/20 08:00	25/04/20 17:00	81	
83	End of Iteration Meeting	1 day	26/04/20 08:00	26/04/20 17:00	82	
84	Maintenance Plan	7 days	23/03/20 08:00	29/03/20 17:00		Tarik Emin Kaplan;Berk Atac
85	Documentation	14 days	23/03/20 08:00	05/04/20 17:00		
86	**Report Writing and Editing**	21 days	**06/04/20 08:00**	**26/04/20 17:00**		**Can Ozan Kas**
87	Maintenance Plan and Details	7 days	06/04/20 08:00	12/04/20 17:00	84	Tarik Emin Kaplan;Berk Atac
88	Requirements	7 days	06/04/20 08:00	12/04/20 17:00		Ali Kemal Ozkan
89	Final Architecture	7 days	06/04/20 08:00	12/04/20 17:00		Can Ozan Kas;Gulnihal Muslu
90	Implementation Details	7 days	11/04/20 08:00	17/04/20 17:00	80	
91	Testing Details	7 days	20/04/20 08:00	26/04/20 17:00	81	
92	Other	21 days	06/04/20 08:00	26/04/20 17:00		



	Name	Duration	Start	Finish	Predecessors	Resource Names
76	**Final Phase**	70 days	**17/02/20 08:00**	**27/04/20 08:00**	**57**	
77	Learning Technologies	56 days	17/02/20 08:00	12/04/20 17:00		
78	**Development Process**	63 days	**24/02/20 08:00**	**26/04/20 17:00**		**Gulnihal Muslu**
79	Iteration Planning	5 days	24/02/20 08:00	28/02/20 17:00		Gulnihal Muslu
80	Implementation	42 days	29/02/20 08:00	10/04/20 17:00	79	
81	Testing	9 days	11/04/20 08:00	19/04/20 17:00	80	
82	Integration	6 days	20/04/20 08:00	25/04/20 17:00	81	
83	End of Iteration Meeting	1 day	26/04/20 08:00	26/04/20 17:00	82	
84	Maintenance Plan	7 days	23/03/20 08:00	29/03/20 17:00		Tarik Emin Kaplan;Berk Atac
85	Documentation	14 days	23/03/20 08:00	05/04/20 17:00		
86	**Report Writing and Editing**	21 days	**06/04/20 08:00**	**26/04/20 17:00**		**Can Ozan Kas**
87	Maintenance Plan and Details	7 days	06/04/20 08:00	12/04/20 17:00	84	Tarik Emin Kaplan;Berk Atac
88	Requirements	7 days	06/04/20 08:00	12/04/20 17:00		Ali Kemal Ozkan
89	Final Architecture	7 days	06/04/20 08:00	12/04/20 17:00		Can Ozan Kas;Gulnihal Muslu
90	Implementation Details	7 days	11/04/20 08:00	17/04/20 17:00	80	
91	Testing Details	7 days	20/04/20 08:00	26/04/20 17:00	81	
92	Other	21 days	06/04/20 08:00	26/04/20 17:00		
93	Submit Final Report	0 days	27/04/20 08:00	27/04/20 08:00		
94	**End Phase**	19 days	**27/04/20 08:00**	**15/05/20 17:00**	**76**	
95	Presentation Preparation	14 days	27/04/20 08:00	10/05/20 17:00		
96	Presentation	0 days	11/05/20 08:00	11/05/20 08:00	95	
97	Reflectionary End Meeting	1 day	14/05/20 08:00	14/05/20 17:00	96	
98	Return of Materials	1 day	15/05/20 08:00	15/05/20 17:00		

30 Mar 20
6 Apr 20
13 Apr 20
20 Apr 20
27 Apr 20
4 May 20
11 May 20
S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S
Tarik Emin Kaplan;Berk Atac
Tarik Emin Kaplan;Berk Atac
Ali Kemal Ozkan
Can Ozan Kas;Gulnihal Muslu
27/04
11/05

As you can see, we have divided the work into 6 phases, each having a milestone deliverable to signify the end of a phase (only exception to that is the last phase named “end phase”, which has it’s milestone deliverable slightly earlier than it’s end). A phase is made of various tasks and work packages (WP). All WPs has a leader assigned to it. Through the whole project all members will get to be the leader of multiple work packages of various types. Tasks, which are part of WPs (and sometimes directly a part of phases), are assigned to any number of group members, depending on the difficulty, workload and collaboration need of each task (a task that is not assigned to specific members are assigned to all members). For example, a specific modelling task is usually assigned to only one member, however long lasting tasks which require more effort like implementation, or complex tasks which require all members to be on the same page such as class designs, or meetings which require all members to be present are assigned to all members.

Another important thing to mention is that we will follow a development process with agile characteristics. We will start the development process in our high-level design phase, where we will create our design and implement our project simultaneously. In each phase, where a development process is present, we will first plan the development process (iteration) for that phase, then we will carry out the necessary development steps and we will end the iteration with a reflectionary meeting. These development processes are present in the two design phases and the final phase. The development process in the final phase is longer than the ones in design phases, and we expect to carry out most of our implementation work in there. Each development process is also a WP, which is also assigned to different leaders in each different phases.

4.4. Ensuring Proper Team-work

We will try to estimate the length of a work and assign the number of group members accordingly. By that way, we distribute the work uniformly. When a WP is assigned to a group member, that group member will be the leader and take its responsibility. Every group member will have a chance to have that opportunity. Assigning different numbers to WPs helps us to make it uniform too. When there are problems between WPs, the leaders will communicate. To ensure that everyone is contributing, we will regularly check everyone's commits on github.

4.5. Ethics and Professional Responsibilities

The object that will be placed by the users may cause accidents. There are places that people should be careful when they visit, like Grand Canyon, as the slightest distraction can cause people to get hurt or fall. When we consider it in a more general sense, this is also valid for any street or building depending on the location of the object. If an object placed by a user can be collected only by standing on a street or top of a building, searching for this object poses a risk for human life. Thus, we will take this situation into consideration while implementing the project.

Creating or completing routes may require visiting places like museums. If those museums are also checkpoints, meaning that the traveler needs to collect or place an object there, the traveler should use the camera on his/her phone, which is generally illegal. If we allow placing objects in such places, we would be forcing people to break the laws. In implementation phases, we will consider this issue.

We should also keep account information of the users secure, meaning that it cannot be shared and it should be protected against third party software. When there is a need of statistical data, the account information that we keep can be used. Yet, it cannot contain the personal information of the users.

4.6. New Knowledge and Learning Strategies

We have to decide carefully which framework and SDKs we should use as Nomad is a web based AR application. For this purpose, we will be learning about frameworks React Native and Titanium and Google's ARCore SDK.

React Native is an open source mobile application framework and it is widely used since it supports cross platform development and has an active community which creates many libraries for various uses. Therefore, there are a lot of resources and tutorials for new starters to learn. It also requires high level of JavaScript knowledge. Since we will be working on a web based application, we will also use JavaScript. During the learning process, we will be reading the tutorial guide of React Native's official site [3] and other well known useful sites like tutorialspoint [4] to see more examples. Appcelerator Titanium is also one of the frameworks that is widely used. It also is an open source application development platform. It allows us to create both mobile and desktop applications using web technologies [5]. We will be using its official site to watch tutorials [6] [7]. We will do exercises with easy practises and decide which framework is more suitable for our project.

In order to include augmented reality to our project, we will be using Google's ARCore. It allows motion tracking, detecting the surfaces and estimating the lighting conditions of the environment. Since the application allows users to place and collect objects with AR in anywhere around the world, ARCore will be the SDK that will provide us the most convenience for the project. We will be reading Google's ARCore documents [8] and searching other sites like HeartBeat [9] that gives basic examples to practise first. We will be also in contact with our supervisor and innovation expert, who have experience with AR, to consult when it is necessary.

5. References

[1] J. Fund, “Travel Is So Much Better Than It Was,” National Review, 02-Jan-2017. [Online]. Available: www.nationalreview.com/2017/01/international-travel-today-much-easier-cheaper-it-was/. [Accessed: 12-Oct-2019].

[2] Ramakrishnan, Raghu, and Johannes Gehrke. Database Management Systems and Solutions Manual. 3rd ed., vol. 6, Cornell University, 2011. pages.cs.wisc.edu/~dbbook/openAccess/thirdEdition/solutions/ans3ed-oddonly.pdf [Accessed: 12-Oct-2019].

[3] Facebook, “React-Native,” [Online]. Available: facebook.github.io/react-native/docs/javascript-environment.html [Accessed: 10-Nov-2019].

[4] Tutorialspoint, [Online]. Available:]https://www.tutorialspoint.com/react_native/index.htm [Accessed: 10-Nov-2019].

[5] Value Coders, “Mobile Frameworks to Know in 2019,” [Online]. Available: https://www.valuecoders.com/blog/technology-and-apps/top-javascript-frameworks-for-mobile-app-development/ [Accessed: 10-Nov-2019].

[6] Appcelerator, [Online]. Avaible: tutorials.axway.com/search/Titanium [Accessed: 10-Nov-2019].

[7] Appcelerator, [Online]. Avaible: https://www.appcelerator.com/Titanium/ [Accessed: 10-Nov-2019].

[8] Google Developers, “Developer Guides”, [Online]. Available: developers.google.com/ar/develop/java/quickstart [Accessed: 10-Nov-2019].

[9] HeartBeat, “Build your first Android AR app with ARCore and Sceneform in 5 minutes.” [Online]. Available: heartbeat.fritz.ai/build-you-first-android-ar-app-with-arcore-and-sceneform-in-5-minutes-af02dc56efd6 [Accessed: 10-Nov-2019].