Place-Finder

**This application is designed and developed to identify, find and interact with important places in the city of the user. With the help of this application, the user can find and communicate with the place he needs in the chosen city**

Pre-Requisits for usage :

It is required that the user use a web browser to interact and quickly access the program

This software is easy to use and does not require basic scientific learning

This software works with a completely universal and professional M.V.C architecture, and all codes are written as simple, readable, clean and optimized as possible.

Requirement Overview :

A browser that supports web applications and JavaScript.

It does not matter if it is a personal computer or mobile phones or tablets or...

Note: It is recommended to use Google Chrome because it supports all technologies

Also, after providing the first output, the team tries to design the mobile application version

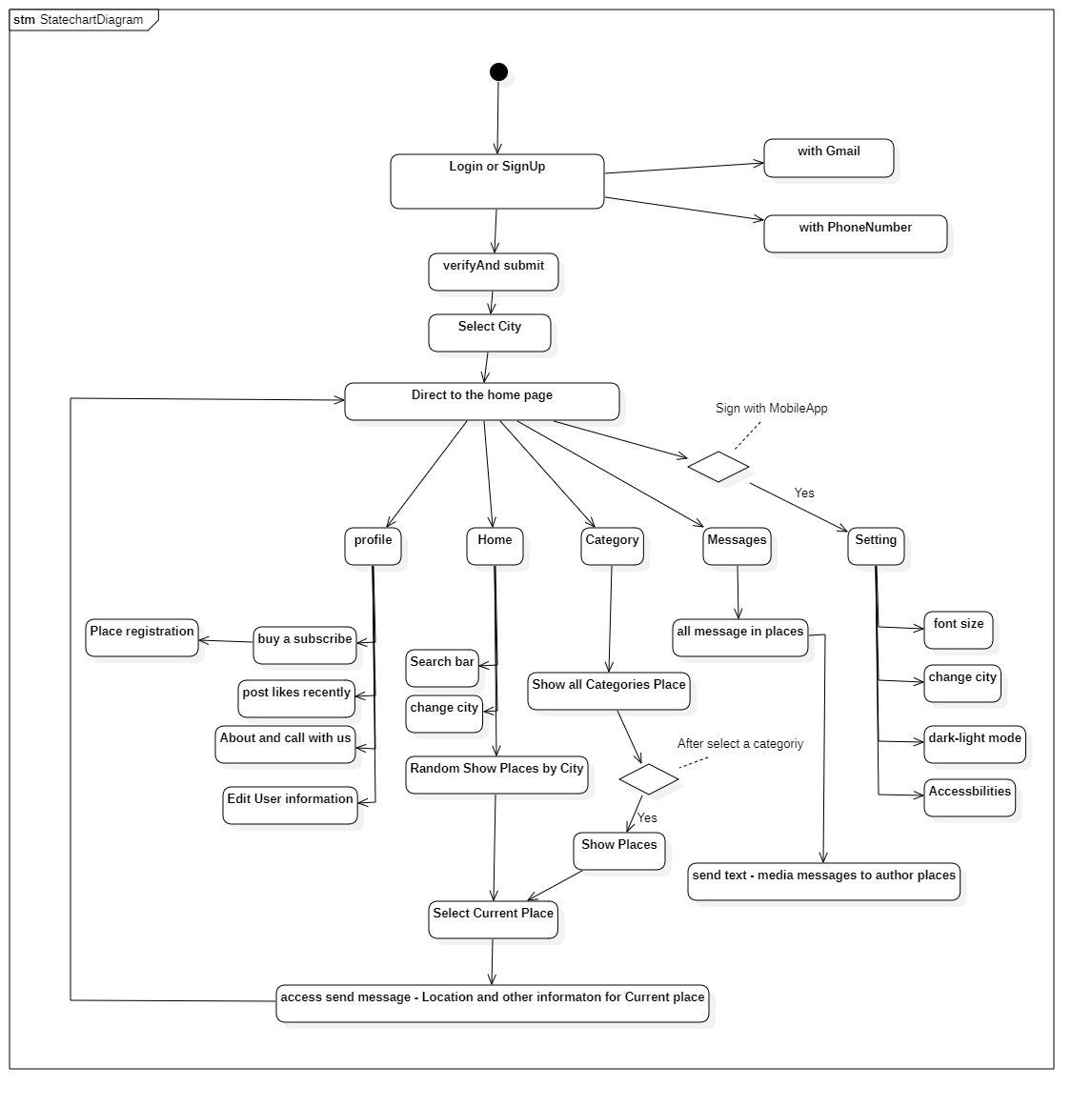
User Interfaces :

Users can easily interact with the features of the application after accessing the software (installing or using the website). In the first step - the user can register in the application using his contact number or email. After receiving and entering the verification code, the user must enter the desired city to display the places related to the same city, after selecting the city, the user will be directed to the home page. On the home page, the user has access to do the following: 1. Profile settings 2. Search the desired place by entering the category and name 3. Checking and calling (if the user is registered) all the places that came as a result of the search. 4. Survey to the desired location 5. Communication with the software support team 6. Send Messages to the owner of the place 7. Buy a subscription 8. Place registration after purchasing a subscription

Note : In order to use the location registration page, the user must first go to the subscription purchase page and pay for the corresponding subscription. After that, the user becomes a special member and can go to the place registration page and perform the following activities that are prepared in the user interface: place name, place location registration, place type, contact number, default photo of the place. (More details are in the class diagram)

UserInterface in Pages :

* Home : [ other pages : {change city} , Search bar{ Filtering feature} , Random Show Places by City {Select Current Place{access send message - Location and other informaton for Current place}} , ]
* Messages : [ other pages : {} , all message in places ]
* Profile : [ other pages : {buy a subscribe} , post likes recently , Edit User information , About and call with us , post likes recently ]
* Setting : [ other pages : {change city} , font size - dark-light mode - Accessbilities ]
* Category : [ other pages : {} , Show all Categories Place{Show Places{Select Current Place}} ]

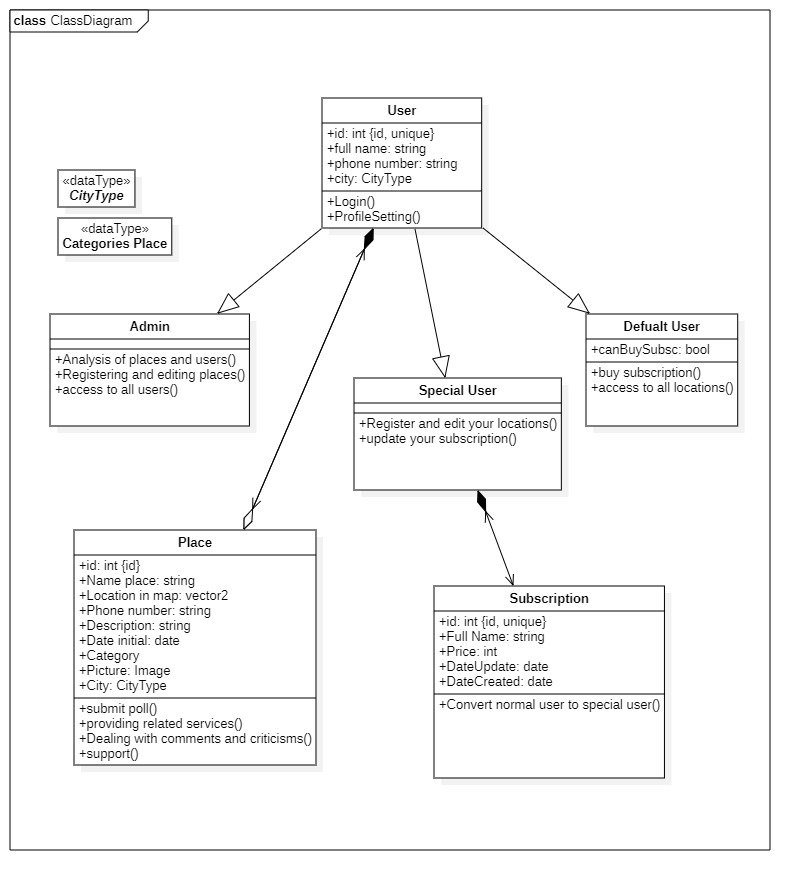


Technical Interfaces :

In the application, each user and Place are our important data on which processes are applied. After registration, each user is stored as new data in its own table in the database, and the administrator has access to it to check users. Places are created by special users (users who purchase subscriptions) and stored in another table in the database. The structure of these data is based on object-oriented structure and each class is separate.

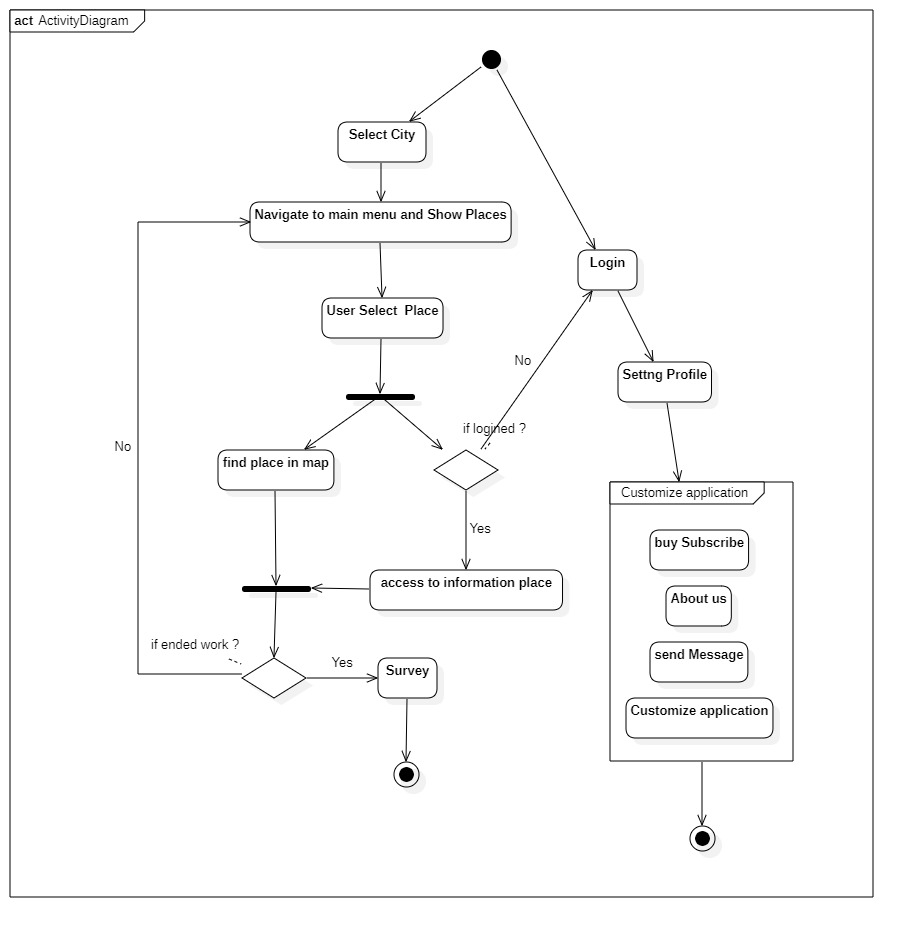
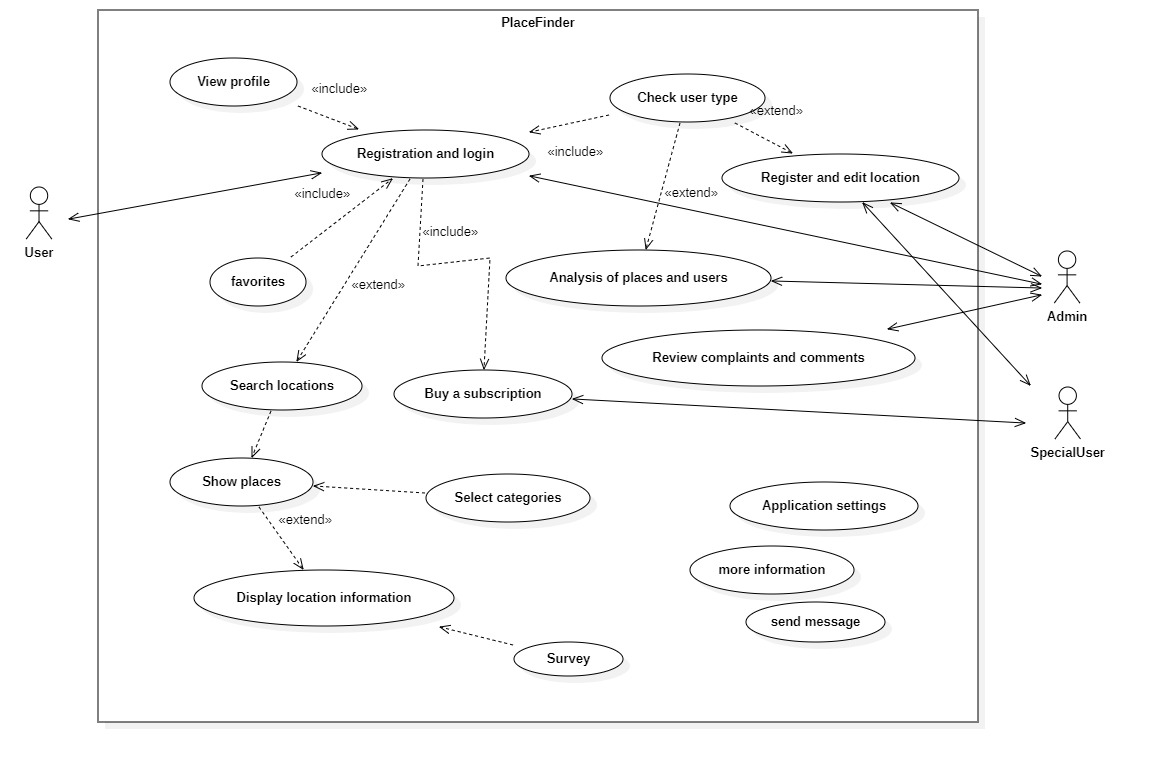
The user class as a parent and normal and special users inherit from the parent class and each has its own characteristics and behaviors that are designed in the class diagram.

Important functions in the application: Place search: This function is used when the user enters the name of the desired Place in the search input and displays each Place with the same name as the input. Place display: This function is used to display Place. All the desired Places are sent from the server to the client in the form of Json depending on the filtering and search. User registration: After entering the email or contact number, this function is called and checks whether this user exists or not. If not, it will be sent to the database as new data, otherwise, the desired user information will be sent to the client. Place registration: This function is only used when the special user enters information about his Place in the form and clicks the registration button. This function checks whether the entered location is allowed to be registered or not, if it is allowed, it is stored in the database and sent to the client so that it can be displayed to all users.



Information Application :

* Build App View : ( use case diagram ) :



* Conventions :

Platform : web application, Mobile Application {

To run the application, web and mobile platforms are used. In the first stage, the application is developed in the website environment (web application). After debugging and debugging for the new update, the mobile application is also designed

}

Server Side : Node js ( express ) }

In order to implement the logic and behaviors of the server side, JavaScript and NodeJS programming languages are used in this project because they have good speed, security and data handling capabilities.

<https://nodejs.org/en/>

}

DataBase : [ if the test fails : Sql Server ] – [ my sql for test ] {

The database used in the application is as follows:

In the first stage: mySql is used and the necessary tests are performed. In case of a drop in speed and hard access to data, the database will be changed and used Sql Server

}

Client Side : Reactjs , Next js {

For the user side and external design, react and next js are used. Because these libraries are very popular and have a high development speed, these libraries are very popular and have a high development speed, NextJS is also used to increase the SEO value of the website. As mentioned earlier, the structure MVC is used.

}

Project Management : Git – Github {

All the files of this project are stored in detail in Git Hub for the access of all the developers who work in this project, with the help of Git commands.

}

Team Management ( scrum ) : Trello {

Each developer in the project has duties that all these plans and duties are recorded and retrieved in Trello. Scrum method is also used to manage the project and use the agile process

}

Api’s : Google map , firebase {

In order to use services such as receiving location, saving data and synchronizing users when necessary, the mentioned interfaces are used

}

* Architecture : M.V.C pattern & Client-server pattern :

In order to further develop this application, developers should use the structure: M.V.C and Client-Server pattern for better management and easier access to the codes. The Client-server pattern is used to connect clients with the server in the project. Developers must comment on every activity they do for the project, whether it is the server side, the client side, or the rest of the developers. Server-side developers should fully explain the reason for using APIs when they use them.

The point that all the developers of this project should understand is that the optimality and high speed of this application is the main pillar of this structure that must be observed.

* Test Strategy : System Tests - Integration Tests - Unit Tests :

// 1. Unit Testing : Testing the correct operation of small blocks of the software source code, based on which the correct operation of each part of the code is evaluated

// 2. Integration Testing : Integrating small blocks of code with each other and forming a series of modules, then integrating several modules with each other and testing their overall performance, so that finally it is ensured that the components or modules that are developed by different developers in a software team work with each other correctly. They communicate.

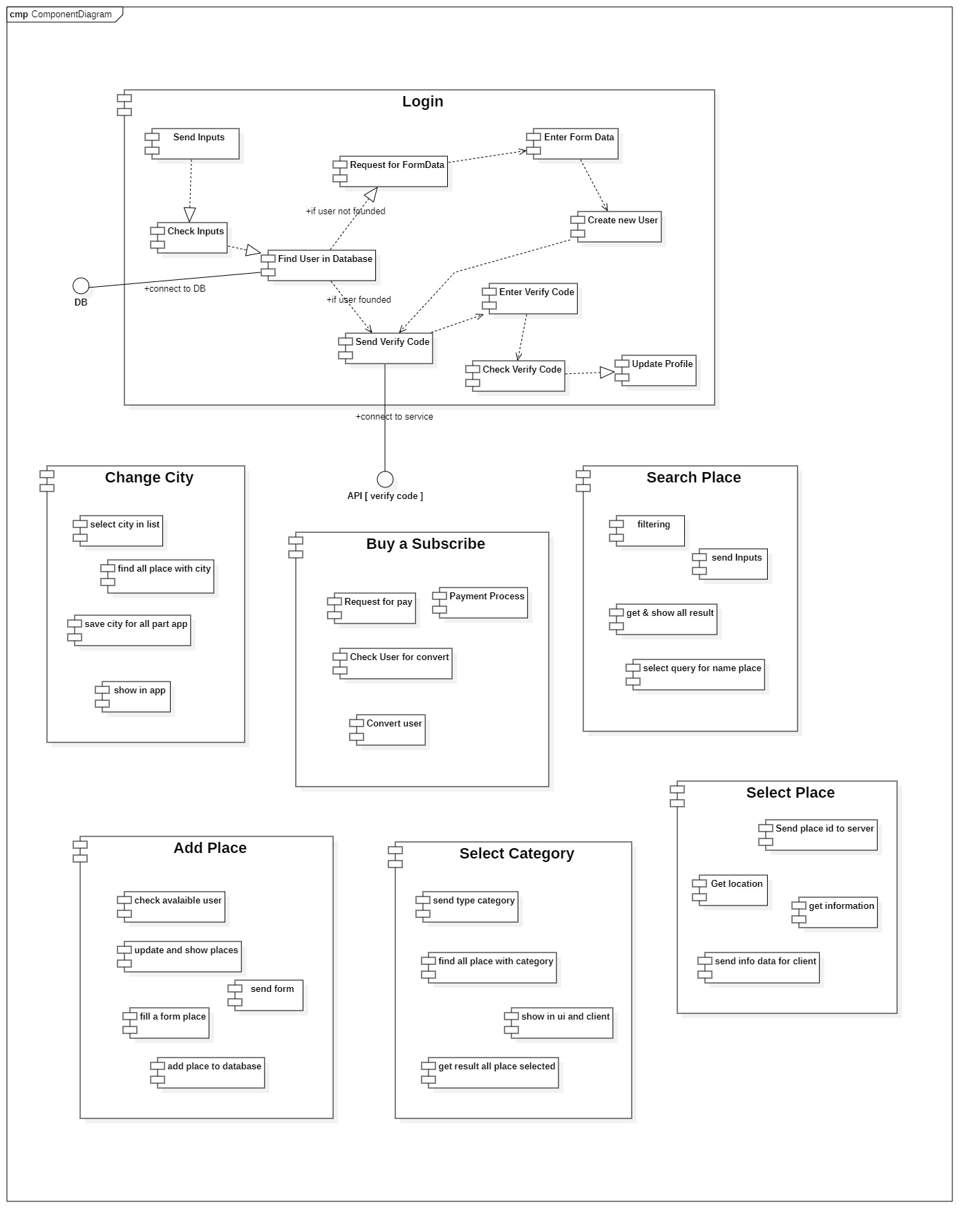
//3. System Testing : By it, a complete software or application is under review so that it will be checked whether our software will work properly and meet their needs when it is available to its users or not.

This process will be completed in the development phase

* Components : // check components in project

In the application, due to the fact that the object-oriented structure is followed, various components have been designed, each of which performs its own operations. These components are designed as a UML diagram and can be seen below

All a important component’s using in project :



* RunTime :

This section contains the details and evolution of the application with the help of diagrams.

Summary :

In general, all the factors of this application include the following:

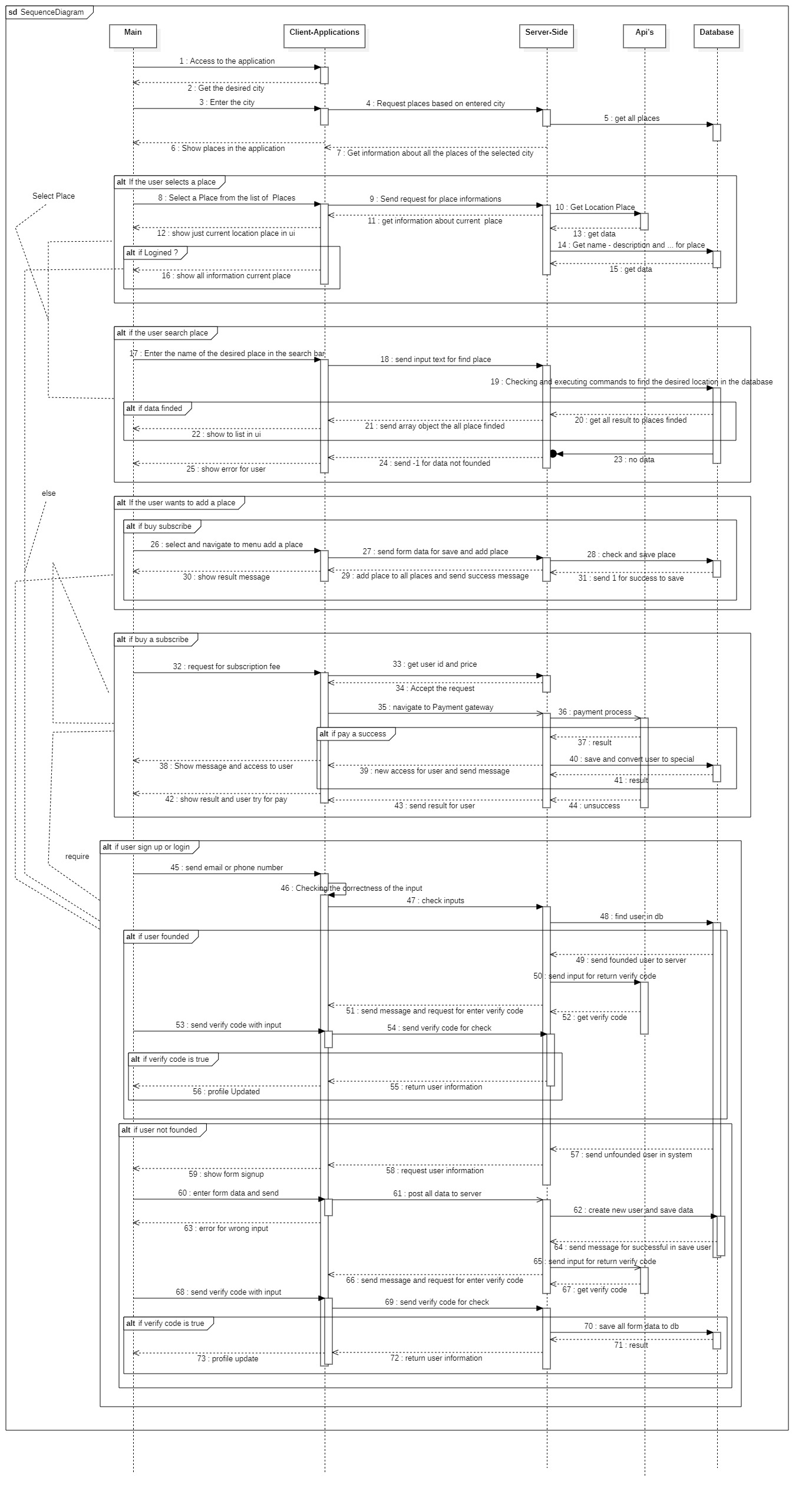
- User behavior: the real person for whom the application is designed. With his choices in this application, the user is trying to solve his needs, which includes finding the places that are useful and needed by the user

- Client application: This factor includes the platforms of this application (mobile app and web app) that the user can interact with.

- Server: The server that includes checking, logical calculations and communication between the database, client application and APIs.

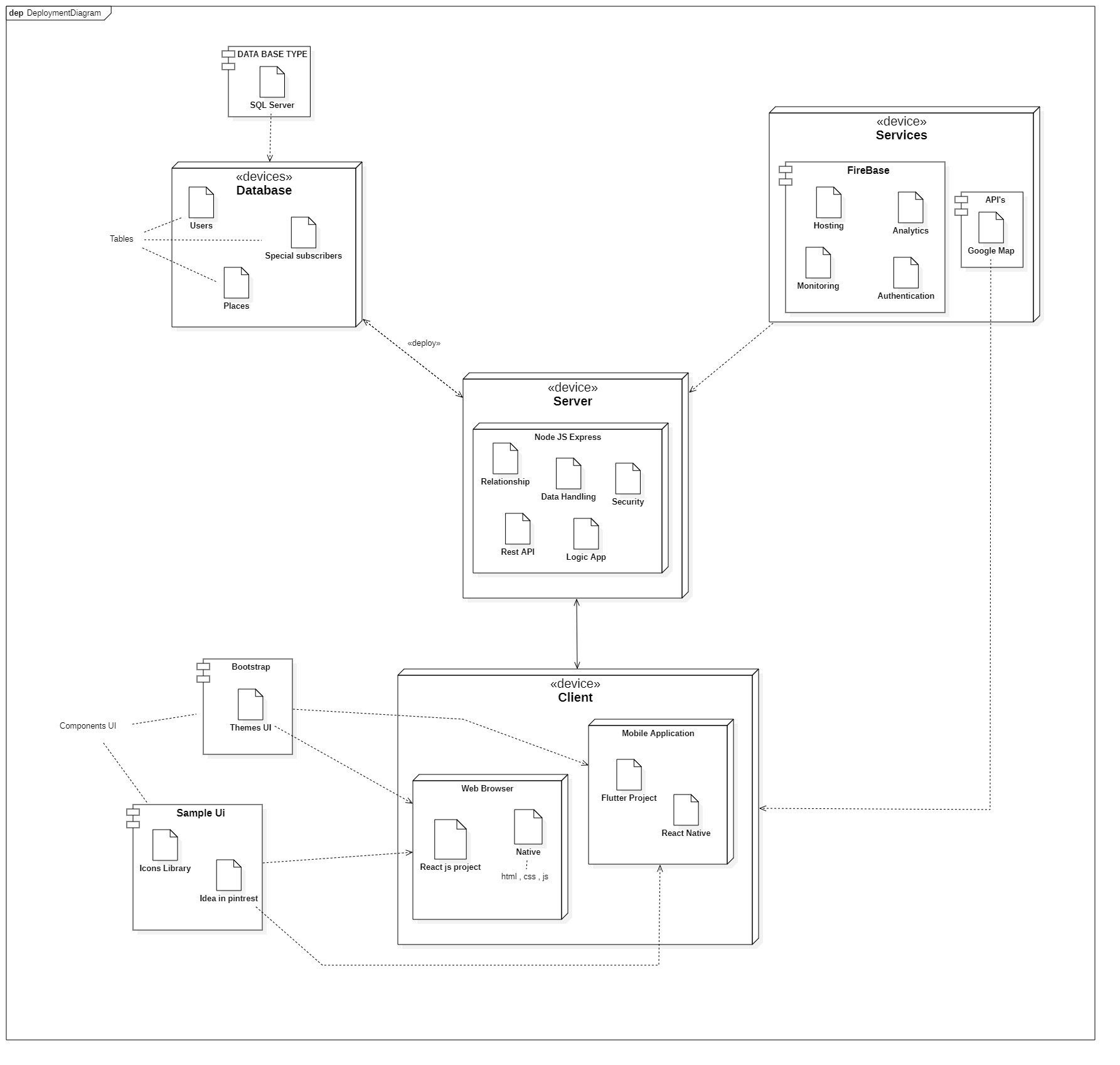
- Interfaces and APIs: external web services that are developed to improve and increase quality and speed

- Database: Application data storage place



* Deployment:

How to install the application and its different parts in the physical structures is designed in the diagram below. The written and tested codes on the server side + client side codes are deployed on the rented virtual server after completion. External components and APIs are connected to the server through interfaces and added to the project. The pre-designed database is also installed inside the virtual server and finally, with the help of the platform defined for the application, users can communicate with it and solve their needs.



* Marketing : // by dr.jalili

Collaborators:

* Product & Project Manager : Dr.Jalili
* Back-end & Software Designer : Mehrab Aghaee
* Front-end : Iliya Khoshnoodi