FOOD EXPRESS

Capstone Report

EGR 4402

Spring 2019

Student: Zineb Bouayad
Supervised by: Dr. Nasser Assem
FOOD EXPRESS

Capstone Report

**Student Statement:**
I affirm that I have applied ethics to the design process and in the selection of the final proposed design. I have also held the safety of the public to be paramount and has addressed this in the presented design wherever may be applicable.”

_____________________________________________________
Zineb Bouayad

Approved by the Supervisor(s)

_____________________________________________________
Dr. N. Assem
ACKNOWLEDGEMENTS

I would like to thank Dr. Assem for accepting to be my capstone supervisor, for helping me and guiding throughout the accomplishment of my project. I am very grateful for his constant guidance, helpful advice, and availability.

I would also like to thank all the people that have been surrounding me in my life, thanks to whom I would have never been able to achieve this capstone project:

- My parents, who always accepted and supported my decisions, and helped me reach my goals.
- My Friends, for their constant help and support.
- My friend Meriam, for always cheering me up, helping me in moments of doubts, being by my side throughout the years at AUI.
- Al Akhawayn University, the School of Science and Engineering.
CONTENTS

Acknowledgements Page ii
Abstract iv
1. Introduction 9
2. STEEPLE Analysis 11
   2.1. Social 11
   2.2. Technological 11
   2.3. Economical 11
   2.4. Environmental 12
   2.5. Political 12
   2.6. Legal 12
   2.7. Ethical 12
3. Requirement Specification 13
   3.1. Functional Requirement 13
      3.1.1. Registration 13
      3.1.2. Login 13
      3.1.3. Reset Password 13
      3.1.4. Change Password 13
      3.1.5. Map 13
      3.1.6. Select Type of Place 13
      3.1.7. View Place Details 14
      3.1.8. Show on Google Map 14
      3.1.9. View Direction 14
      3.1.10. List 14
      3.1.11. Sign Out 14
   3.2. Non-Functional Requirement 14
      3.2.1. Usability 14
      3.2.2. Efficiency 14
      3.2.3. Performance 14
      3.2.4. Development 15
      3.2.5. Portability 15
      3.2.6. Reliability 15
      3.2.7. Privacy 15
      3.2.8. Ethical 15
      3.2.9. Security 15
4. Feasibility Study 16
4.1. Operational Feasibility 16
4.2. Technical Feasibility 16
4.3. Economical Feasibility 16
4.3. Time Feasibility 17
5. Methodology 18
6. Software Architecture 18
   6.1. Food Express Architecture 18
   6.2. Structure of the data in Firebase 19
7. Software Design 22
   7.1. Use Case Diagram 22
   7.2. Sequence Diagram 27
   7.3. Class Diagram 28
8. Implementation Detail of the Solution 32
   8.1. Front-end 32
      8.1.1. Registration/Login Activity 32
      8.1.2. Change Password Activity 32
      8.1.3. Map Activity 33
      8.1.4. View Place Detail Activity 34
      8.1.5. Display List Activity 35
      8.1.6. View Direction Activity 35
   8.2. Firebase Database 36
   8.3. Google Maps API 37
9. Technologies Enablers 43
10. Conclusion 45
11. Future Perspectives 47
12. References 48
13. SCREENSHOTS OF THE CODE 49
LIST OF FIGURES

Figure 5.1. The Incremental Development Model 19
Figure 6.1. Food Express Architectural Design 21
Figure 6.2. JSON Representation of Markets 22
Figure 6.3. JSON Representation of Products 23
Figure 3.1. Use Case Diagram 24
Figure 6.4. Sequence Diagram 30
Figure 6.5. Class Diagram 31
Figure 6.1. Register Activity 33
Figure 6.2. Reset Password Activity 33
Figure 6.3. Login Activity 34
Figure 6.4. Main Activity 34
Figure 6.5. Change Password 35
Figure 6.6. Map Activity 36
Figure 8.7. View Place Details Activity 37
Figure 8.8. Display List Activity 37
Figure 8.8. View Direction Activity 38
Figure 6.2.1. Linking database to the application 39
Figure 8.2.2. Firebase Dependencies 39
Figure 8.2.3. List of Users in the Database 40
Figure 8.2.4. Authentication in Firebase 40
Figure 8.2.3. Google Maps Dependencies 41
Figure 8.2.4. Google API & Services Console. 41
Figure 8.2.5. JSON File of Nearby Restaurant. 42
Figure 11.2. Screenshot the SignUp Activity. 49
Figure 11.3. Screenshot of the Main Activity. 50
Figure 11.4. Screenshot of the Maps Activity. 50
Figure 11.5. Screenshot of the View Place Activity. 51
Figure 11.6. Screenshot of the View Direction Activity. 51
LIST OF TABLES

Table 4.1. Time Feasibility 18
Table 3.1. First Use Case Diagram 25
Table 3.2. Second Use Case Diagram 25
Table 3.3. Third Use Case Diagram 26
Table 3.4. Fourth Use Case Diagram 26
Table 3.5. Fifth Use Case Diagram 27
Table 3.6. Sixth Use Case Diagram 27
Table 3.7. Seventh Use Case Diagram 28
Table 3.8. Eight Use Case Diagram 28
Table 3.9. Ninth Use Case Diagram 29
ABSTRACT

In this report, I will talk about the different stages throughout the accomplishment of my project Food Express. The goal of this capstone is to create a mobile application for markets, restaurants, schools, and hospitals. This application allows people to find the nearest places depending on their addresses. In order to implement it, I used the software Android Studio as an IDE. This software helped me organize the different classes in an object-oriented manner. This project was divided into two main parts; the first one consist of an interface with a map where the users can see their locations, using their latitudes and longitudes, they will get the nearest stores in the area. The second interface is when the user select a specific market, he/she can have access to all the products of the given store, and he can also add some of them in a basket. The user can also view the direction from his location to the desired place.

I created a database using firebase database that hold the locations of the groceries along with their products. As main language of implementation I used Java. This application is available for android users only for free.
**Résumé**

Dans ce rapport, je parlerai des différentes étapes qui ont précédé à la réalisation de mon projet Food Express. L'objectif de cette synthèse est de créer une application mobile pour les supermarchés, les restaurants, les écoles et les hôpitaux. Cette application permet aux gens de trouver les endroits les plus proches en fonction de leurs adresses. Afin d’être réalisé, j'ai utilisé le logiciel Android Studio en tant qu'EDI. Ce logiciel m'a aidé à organiser les différentes classes de manière orientée objet. Ce projet a été divisé en deux parties principales ; la première consiste en une interface avec une carte où les utilisateurs peuvent avoir accès à leurs emplacements, en utilisant leurs latitudes et longitudes, ils obtiendront les magasins les plus proches de la région. La deuxième interface est lorsque l'utilisateur sélectionne un supermarché spécifique, il peut avoir accès à tous les produits du magasin en question, et il peut également en ajouter certains dans un panier. L'utilisateur peut également afficher la direction de son emplacement à l'endroit souhaité.

J'ai créé une base de données utilisant la base de données Firebase contenant les emplacements des supermarchés avec leurs produits. Comme langue principale de mise en œuvre, j'ai utilisé Java. Cette application est disponible uniquement pour les utilisateurs d’Android.
1 INTRODUCTION

Living here in Ifrane, and studying at AUI we have the chance to meet a lot of exchange students. One thing we learned is that they always complain about not knowing in advance what kind of products a market has to offer. Even if they find a certain market online, they cannot have access to its list of products. Thus, my motivation to create Food express. This application is meant for foreigners and Moroccans that are not from a certain region, they can simply go on the app and start searching for the nearest places. This application is not only intended for markets, Food Express is designed so that it displays the nearest markets, restaurants, schools and hospitals, without wasting time of its users searching for one. Everything is available by just one click of hand. My motivation for the creation of this app came also from my interest for google maps, I have always been curious to learn how they work, and know how to include Google Maps APIs in a mobile application. Working with Google Map Platform, you have the chance to learn a lot about the different libraries available and the API family, for example the API maps, API places, and API routes.

This application will provide the users with the nearest places depending on their locations, if the selected place is a market it will display the list of products of the given store with the price of each product. If the user likes something he can add it to a basket, that way when he comes to the store he can find what he selected ready and waiting for him. This app will also help the owners of the stores, since it will attract more clients to the grocery. Each place available on the map will contain its own specific information. Every time a user clicks on a place; the name, address, rating, opening hours along with the image of the store are displayed on the user’s screen. On the same activity, the user will also have the choice to either display the list
of products (if the selected place is a market), show the selected place on Google Map, or to view the direction leading to the chosen place.

In this report, I will talk about the STEEPLE analysis as well as the feasibility study, which helped me determine the feasibility of some functionalities according to the resources available and the time I had. I will also talk about the methodology followed, and explain briefly the main building blocks of the project, I will create the different diagrams that explains the design of the solution, and finally I will talk about the implementation details of the project.
2 STEEPLE ANALYSIS

The purpose of the STEEPLE analysis is to determine the impact that our application has on the different factors presented below.

2.1 Societal

Food express was implemented in order to help people that are foreign to a city to find not only the nearest grocery stores, but also the nearest restaurants, schools, and hospitals. Nowadays people became extensive users of online services, this is where lies the utility of my application since it will give the users the nearest locations within a perimeter of 1000 meters within a simple click of their hands.

2.2 Technological

With the help of new technologies, in this case google maps APIs I created Food Express an application that will make life of its users easier and improve the quality of their lives by providing them with the nearest locations by just a click.

2.3. Economical

Having an application that provides you with the nearest markets, restaurants and schools is very helpful for the owners of businesses as it will attract more clients to the stores. Food Express was created with the help of Android Studio, an open source free IDE. The application is available for android users for free, and there will not be any kind of payment involved in the application.

2.4. Environmental

Food Express does not use any natural resource, and so it will have a positive impact on the environment by not wasting any of its recourses. In addition, the users of the app would not
have to waste time and gas fuel to go and search for a restaurant or a market all over the city. The users can simply go on the app and start looking for the desired place on the map. This would save them a lot of energy as it would also minimize their intake on gas/fuel making it environmentally friendly.

2.5. Political

This application does not favor any political entity in particular, and is not implicated in any political party.

2.6. Legal

This project has been created using open source software that are available online such as Android Studio and Firebase database. Doing so, this application respects all the regulations and copyrights. By using Firebase as database, I ensure that all the data of the users are kept strictly secure and confidential.

2.7. Ethical

Food Express is conformed to the norms of the code of ethics, as it ensures the privacy of its users and their data.
3 REQUIREMENT SPECIFICATIONS

3.1. Functional Requirements

3.1.1. Registration

The user shall be able to enter a valid email address and password. The password length must be a minimum of 6 characters and it must contain at least one digit.

3.1.2. Login

The user shall enter the exact same email-address and password in order to have access to the application, otherwise an error message will appear.

3.1.3. Reset Password

The user shall be able to reset his password, in case he forgot it, from his webmail. He shall enter a valid password with a length of minimum 6 characters and at least one digit.

3.1.4. Remove User

The user shall be able to remove himself from the database by clicking on the “remover user” button, doing so he will no longer have access to the application.

3.1.5. Change Password

After being logged in, the user shall modify his password from the application by entering a new valid one. The new password shall be different from the old one.

3.1.6. Map

By clicking on the “Map” button, the user shall see the nearest markets, schools, restaurants, and hospitals on the map depending on his location.

3.1.6. Select Type of Place

The user shall choose what type of place he wants to display by clicking on one of the four icons on the menu bar. Each icon represents either markets, restaurants, schools or hospitals.
3.1.7. View Place Details

The user shall be able to see the image, name, address, opening hours and ratings of the selected place.

3.1.8. Show on Google Map

The user shall be able to view the selected place on google map so that he can have access to more information about that place.

3.1.9. View Direction

The user shall get the direction from his current location to the selected place in the form of a red path by clicking on the “view direction” button.

3.1.10. List

The user shall be able to see the list of available products of a given market, the name of each product along with its prices will be displayed.

3.1.11. Sign Out

The user shall be able to sign out from the application anytime he/she wants.

3.2. Non-functional Requirements

3.2.1. Usability

The application shall have a simple interface; it will be easy to use. Every user should be able to use all its functionalities without any ambiguity.

3.2.2. Efficiency

The system shall display the nearest places, their information along with their directions in the shortest amount of time.

3.2.3. Performance

The application shall not exceed the time limit for an operation under normal conditions.
3.2.4. Development

The application shall be created using Android Studio as IDE and Java shall be used as main language of implementation.

3.2.5. Portability

Food Express shall be available on all devices working with an android OS.

3.2.6. Reliability

The application shall not fail, or crush under any circumstances for the whole period of time used by the user.

3.2.7. Privacy

The application shall keep the data of its users private from the public.

3.2.8. Ethical

The application shall conform to the norms and code of ethics of the “Association for Computing Machinery”.

3.2.9. Security

The application shall be implemented in a secure way, such that no other user can have access to the information of someone else’s profile.
4 FEASIBILITY STUDY

The goal of this capstone is to create a mobile application to help customers find the nearest places, and shop at the grocery stores.

4.1 Operational Feasibility

Concerning the operational feasibility of the project, once the users register to the app, they will have access to the nearby markets, restaurants, schools and hospitals based on their addresses. When the user clicks on one of the icons of the menu bar he will have all the results displayed on the map view as coloured pins. Once the user chooses a specific place, information as address, rating, and opening hours will be displayed. The user will also have the option to show that specific place on the actual google map. When the user chooses to click on a market, he/she can view a list of its different items. If the user likes the item, he/she can add them to a basket (shopping card). Once the order has been confirmed, a person in the grocery will be responsible of gathering the different items of the basket. This project intends to solve the problem of people not having enough time to go and search for all their necessary groceries over different stores, so they can go to the selected store and they will find all the products that they need in it. This application will also allow markets to attract more clients to the store.

4.1 Technical Feasibility

For the technical feasibility, I will be using the software Android Studio as an IDE. In order to get the nearest places of the area I will need certain requited parameters. First, in order to identify my application, I will need the API key of the application. Then, I will need the latitudes
and longitudes in order to get the location of the desired place. Finally, we will move on to the radius, this parameter will determine the maximum distance (which usually does not exceed 50,000 meters) within which to return the final results. [1]

Concerning the database, I chose Firebase Database, this database enables the users to synchronize and store data in real time using a NoSQL, cloud hosted database. As main language of implementation I will be using Java. [2]

4.3. Economical Feasibility

Concerning the economical part of the feasibility, the application will be available for Android users for free, and there will not be any kind of payment involved in the application. When the client gets its order, he will pay as he would normally do at the store. The stores using the services of the app would have to pay a monthly fee.

4.4. Time Feasibility

The Schedule feasibility is as follow; this schedule was made in order to help me finish the project within the time required, and to stay up to date with my work.
<table>
<thead>
<tr>
<th>Week</th>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Select the project</td>
</tr>
<tr>
<td>1</td>
<td>Generate the initial Specification</td>
</tr>
<tr>
<td>2</td>
<td>Install the required software</td>
</tr>
<tr>
<td>3</td>
<td>Feasibility Study</td>
</tr>
<tr>
<td>3-5</td>
<td>Implement the first part of the project</td>
</tr>
<tr>
<td>5-6</td>
<td>The Interim report</td>
</tr>
<tr>
<td>7-11</td>
<td>Continue with the implementation of the project</td>
</tr>
<tr>
<td>12</td>
<td>Final Report</td>
</tr>
<tr>
<td>13</td>
<td>Software validation and testing</td>
</tr>
<tr>
<td>14</td>
<td>Presentation of the capstone</td>
</tr>
<tr>
<td>15</td>
<td>Make modification to the final report.</td>
</tr>
</tbody>
</table>

Table 4.1. Time Feasibility
5 METHODOLOGY

The methodology used in the implementation of this project is the Incremental Development Model. This methodology allowed me to break down the delivery and the development into many increments, instead of delivering the final product as a single release. Each of the delivered increment would be responsible of a specific functionality that would then be assembled with the other increments to form the final product. Doing so, customers feedbacks are available with the release of each increment, which speeds up the release of the project. This gives to the final project more priority and reduce the risk of failure. The advantages that comes with using this method is that the functionalities with the highest priority receive more testing. [3]

![Incremental Development Model](image)

Figure 5.1 The Incremental Development Model
The first increment developed for this application included the Registration, Login, Main, and Reset Password Activities. The Registration Activity prompts the user to enter a valid email and password that would then be saved in the firebase database. After successful registration, the user can be able to login, in the same activity the user has the option to reset his password in case he/she forgot it. After login in the user is directed to the main activity.

The second increment consists of the Map, ViewplaceDetail, and DisplayList Activities. The Map activity displays your location in a map where you can select the type of place desired. Once you click on a pin the details of that place will be displayed. If the place selected is a market you can see the list of products available.
6 SOFTWARE ARCHITECTURE

6.1. Food Express Architecture

For the architecture of the application, Food Express make use of firebase database which allows you to synchronize data over all of its users using its real-time database. The data is synchronized and stored using a cloud database and NoSQL. [4]

Using the authentication service provided by the database, each time a user registers, the application sends a REST API request to the database in order to save their information. Once the users are registered, they can access the map that displays the nearby services. This is possible thanks to the google maps APIs platform. When a user clicks on the “MAP” button, a HTTP request will be processed by the firebase, and the data stored will be displayed on the screen of the user.
6.2. Structure of the data in firebase

The data stored in the real-time database is represented as JSON objects. As said earlier firebase database is based on NoSQL and so, the data is not stored in the form of tables as it is the case with SQL, but is rather represented in the form of a JSON tree. Once a user is registered and its information is added, the data turns into a node in the JSON tree and each node is associated with a specific key. [5] As follow are examples of data representation in a firebase database.

![JSON Representation of Markets](image)

**Figure 6.2. JSON Representation of Markets**

In this screenshot, you can see the JSON representation of the nearby markets in Ifrane. Each market contains the list of products, the name of the store, its opening hours, a picture, and its rating.
Figure 6.3. JSON Representation of Products

In this second screenshot, you can see the list of products of a market. Each product contains a description, an image, the name of the product and its price.
7 SOFTWARE DESIGN

7.1. Use Case Diagram

Figure 3.1. Use Case Diagram
<table>
<thead>
<tr>
<th>Use Case Id</th>
<th>001</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Case Name</td>
<td>Register</td>
</tr>
<tr>
<td>Description</td>
<td>The user registers must enter a valid email and password.</td>
</tr>
<tr>
<td>Actor</td>
<td>User</td>
</tr>
<tr>
<td>Precondition</td>
<td>The mobile app must be active</td>
</tr>
</tbody>
</table>
| Main flow     | Click on the icon of the app.  
|               | Start the application.  
|               | The Register Activity will show up. |
| Post Condition| The user will be added to the database |
| Alternatives  | Quit the app.             |

**Table 3.1. First Use Case Diagram**

<table>
<thead>
<tr>
<th>Use Case Id</th>
<th>002</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Case Name</td>
<td>Login</td>
</tr>
<tr>
<td>Description</td>
<td>The user enters his email and password in order to access the functionalities of the app.</td>
</tr>
<tr>
<td>Actor</td>
<td>User</td>
</tr>
<tr>
<td>Precondition</td>
<td>The user must be registered in the app</td>
</tr>
<tr>
<td>Main flow</td>
<td>Once the user clicks on login, he will be directed to the main activity.</td>
</tr>
<tr>
<td>Post Condition</td>
<td>The Main activity is displayed</td>
</tr>
<tr>
<td>Alternatives</td>
<td>Click on the button “Forgot Password”</td>
</tr>
</tbody>
</table>

**Table 3.2. Second Use Case Diagram**
<table>
<thead>
<tr>
<th>Use Case Id</th>
<th>003</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Case Name</td>
<td>Forgot Password</td>
</tr>
<tr>
<td>Description</td>
<td>The user can change his password by entering his email.</td>
</tr>
<tr>
<td>Actor</td>
<td>User</td>
</tr>
<tr>
<td>Precondition</td>
<td>The user must be registered in the app.</td>
</tr>
<tr>
<td>Main flow</td>
<td>Once the user changes his password, he will be able to login.</td>
</tr>
<tr>
<td>Post Condition</td>
<td>The password will be changed.</td>
</tr>
<tr>
<td>Alternatives</td>
<td>Quit the app.</td>
</tr>
</tbody>
</table>

**Table 3.3. Third Use Case Diagram**

<table>
<thead>
<tr>
<th>Use Case Id</th>
<th>004</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Case Name</td>
<td>Map View</td>
</tr>
<tr>
<td>Description</td>
<td>The user will be able to see his location on a map.</td>
</tr>
<tr>
<td>Actor</td>
<td>User</td>
</tr>
<tr>
<td>Precondition</td>
<td>The user must be logged in.</td>
</tr>
<tr>
<td>Main flow</td>
<td>Once the user clicks on the “Map” button, the map activity will be displayed.</td>
</tr>
<tr>
<td>Post Condition</td>
<td>The map view will be displayed on the screen.</td>
</tr>
<tr>
<td>Alternatives</td>
<td>The user goes back to the main activity.</td>
</tr>
</tbody>
</table>

**Table 3.4. Fourth Use Case Diagram**
<table>
<thead>
<tr>
<th>Use Case Id</th>
<th>005</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Case Name</td>
<td>Choose Place Type</td>
</tr>
<tr>
<td>Description</td>
<td>The user must click on one of the icons displayed on the menu bar.</td>
</tr>
<tr>
<td>Actor</td>
<td>User</td>
</tr>
<tr>
<td>Precondition</td>
<td>The user must be on the map view</td>
</tr>
<tr>
<td>Main flow</td>
<td>Once the user clicks on the desired service, the nearby places will be displayed.</td>
</tr>
<tr>
<td>Post Condition</td>
<td>Colored pins will be displayed on the map.</td>
</tr>
<tr>
<td>Alternatives</td>
<td>The user can select one of the three other place types.</td>
</tr>
</tbody>
</table>

**Table 3.5. Fifth Use Case Diagram**

<table>
<thead>
<tr>
<th>Use Case Id</th>
<th>006</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Case Name</td>
<td>See Place Details</td>
</tr>
<tr>
<td>Description</td>
<td>The name, address, opening hours and rating of the selected place will be displayed.</td>
</tr>
<tr>
<td>Actor</td>
<td>User</td>
</tr>
<tr>
<td>Precondition</td>
<td>The user must click on a specific place.</td>
</tr>
<tr>
<td>Main flow</td>
<td>Once the user clicks on a specific place the details of that place will be displayed.</td>
</tr>
<tr>
<td>Post Condition</td>
<td>ViewPlace activity will be displayed.</td>
</tr>
<tr>
<td>Alternatives</td>
<td>The user can select another place.</td>
</tr>
</tbody>
</table>

**Table 3.6. Sixth Use Case Diagram**
### Table 3.7. Seventh Use Case Diagram

<table>
<thead>
<tr>
<th>Use Case Id</th>
<th>007</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Case Name</td>
<td>View List of Products</td>
</tr>
<tr>
<td>Description</td>
<td>The list of products of the selected store will be displayed.</td>
</tr>
<tr>
<td>Actor</td>
<td>User</td>
</tr>
<tr>
<td>Precondition</td>
<td>The user must click on the “List” button.</td>
</tr>
<tr>
<td>Main flow</td>
<td>Once the user clicks on the List button the products along with their prices will be displayed</td>
</tr>
<tr>
<td>Post Condition</td>
<td>The user will be directed to the DisplayList Activity</td>
</tr>
<tr>
<td>Alternatives</td>
<td>The user can click on “Show on Map” or “View Directions” buttons.</td>
</tr>
</tbody>
</table>

### Table 3.8. Eight Use Case Diagram

<table>
<thead>
<tr>
<th>Use Case Id</th>
<th>008</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Case Name</td>
<td>View Directions</td>
</tr>
<tr>
<td>Description</td>
<td>The direction from your current location to the selected place will be shown.</td>
</tr>
<tr>
<td>Actor</td>
<td>User</td>
</tr>
<tr>
<td>Precondition</td>
<td>The user must click on the “Show Direction” button.</td>
</tr>
<tr>
<td>Main flow</td>
<td>Once the user clicks on the Show Direction button the path to the selected place will be displayed</td>
</tr>
<tr>
<td>Post Condition</td>
<td>The user will be directed to the ViewDirection Activity</td>
</tr>
<tr>
<td>Alternatives</td>
<td>The user can click on “Show on Map” or “ListDisplay”.</td>
</tr>
<tr>
<td>Use Case Id</td>
<td>009</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Use Case Name</td>
<td>Show on Google Map</td>
</tr>
<tr>
<td>Description</td>
<td>The selected place will appear on the actual google map</td>
</tr>
<tr>
<td>Actor</td>
<td>User</td>
</tr>
<tr>
<td>Precondition</td>
<td>The user must click on the “Show on Map” button.</td>
</tr>
<tr>
<td>Main flow</td>
<td>Once you click on the Show on Map button, the user will be directed to google maps where he can see the chosen place.</td>
</tr>
<tr>
<td>Post Condition</td>
<td>The place will be displayed on google map</td>
</tr>
<tr>
<td>Alternatives</td>
<td>The user can click on “View Directions” or “ListDisplay”</td>
</tr>
</tbody>
</table>

**Table 3.9. Ninth Use Case Diagram**
7.2. Sequence Diagram

Figure 6.4. Sequence Diagram
7.3. Class Diagram

![Class Diagram Image]

Figure 6.5. Class Diagram
8 IMPLEMENTATION DETAILS OF THE SOLUTION

8.1. Front End

8.1.1. Register/Login Activity

As a first step for my capstone project I created a register/login platform. When a user first uses the application, he has to enter a valid email and password in order to get registered. Once successfully registered he is added to the firebase database. Then, the user is directed to the Main Activity where he has the possibility to either change his password, remove himself from the database, go to the MapView, or sign out.

On the login Activity side, the user must enter the exact same email and password entered during the registration in order to have access to the functionalities of the app. In case the user forgot his password, no need to worry, Food Express has the option to allow its user to reset their passwords. That’s when an email is going to be sent to his email address where he can modify it. After successful login, the user is then going to be directed to the Main Activity.
Figure 6.1. Register Activity

Figure 6.2. Reset Password Activity
8.1.2. Change Password Activity

After being logged in, the user is directed to the Main Activity where he has the option to modify his password. The difference between modify password and reset password is that in reset password the user forgot does not have access to the app, and so an email has to be sent to his email address to reset it. But for change password, the user is already logged in and having access to the app he can modify it through application without having to go through his email box.
8.1.3. Map Activity

Once the user decides to click on the “Map” button, he will be directed to the map Activity where he will be able to see his current location as a green pin in a map fragment. In the bottom of the activity there is a menu bar with four icons, each one representing markets, restaurants, schools and hospitals. When the user clicks on one of the icons, the places that are in a perimeter of 1000 meter form the user’s location will be displayed on the map fragment. Each type of place is characterized by a color, for the markets multiple pins in pink will be displayed, for restaurants it would be red pins, violet for schools, and finally green for hospitals.
8.1.4. View Place Details Activity

This activity is designed so that it displays information such as the image, address, rating and opening hours of the selected place. When the activity is shown on the screen three buttons are displayed along with the information, the user has the choice to show this place on google map, to view the direction from his location to the selected place. If the place is a market he can display its list of products.
8.1.5. Display List Activity

Once the user has selected a specific market, he can click on the “List” button, this will direct him to the List Display Activity. This Activity return a list view with all the products of the chosen store along with their prices.

8.1.6. View Direction Activity

If the user chooses the “View Direction” button, the map fragment will be displayed once again on the user’s screen. This time only two pins will appear, one green indicating the location of the user and another yellow indicating the location of the desired destination. Between the two pins is a red path showing the direction to follow.
Moving on to the implementation of the database, first I had to connect the database to my application. To be able to do that, I had to download a google-services.json file and add it to the src package of my application, then to complete the connection to the database I added the dependencies for the Firebase SDKs to the build.gradle, and app/build.gradle files of the project.
After completing these steps, the database should be successfully linked with the android application. But one last step is required which is to enable the “Email/Password” in the sign-in method. From now-on every time a user registers to the app his email, password along with a unique user UID key are automatically added to the database. As seen below is a screenshot of the different users available in the database.
8.3. **Google Maps APIs**

The next step in the implementation of this application consists of including the required google maps dependencies that are necessary for the app to function correctly. As you can see in the screenshot below, I used Google maps API location, nearby, and maps.
With the help of google API & Services console, I have access to an API key, in this case API key 5 as seen in the screenshot below, that sends an HTTP request that results in a JSON file containing the nearby locations depending on what type of service was chosen. [6] In the image below you can see an example of the JSON file that was returned containing the restaurants near to Ifrane.

Figure 8.2.3. Google Maps Dependencies

Figure 8.2.4. Google API & Services Console.
Figure 8.2.5. JSON File of Nearby Restaurant.
9 TECHNOLOGICAL ENABLERS

Java is an object-oriented, class-based programming language. This language was created specifically to help developers decrease the use of dependencies as much as possible. According to Oracle, Java is one of the most secure, reliable and fast programming languages available. [7] This language was used as main programming language for the creation and implementation of the application Food Express.

Android Studio is an open source integrated development environment created mainly for the development of android applications and websites. [8] This software was used as an IDE for the creation of this project.

XML stands for eXtensible Markup Language; it is very similar to HTML. XML was created mainly to transport and store data. [9] In android Studio XML files are used to create and edit activity layout files.
Firebase database is a cloud-hosted real time database based on NoSQL. This database enables the users to synchronize and store their data in real time, it also allows you to have access to the user’s data even if the app is offline. [4]

Google Maps platform covers approximately 99% of the world, makes more than 25 million updates a day, and counts more than 1 billion users. Google maps APIs offers Places APIs, Routes APIs, and Maps APIs all of which were used in the implementation of this application. Maps APIs allowed me to display a map fragment, Routes APIs allowed me to get the direction to a specific place, and finally places allowed me to get the nearby places within perimeter of a 1000 meters. [10]
10 CONCLUSION

To conclude, Food Express was a great opportunity for me to put into practice what I have learned these past four years. Most of the computer science related classes that I took were of much help for me for the accomplishment of this project, especially the Java, Software Engineering and Database classes.

So far I have achieved the most important aspects that application was supposed to do, Food Express successfully registers and adds the users into the database, once they are logged-in they are able to go the Map View where a fragment of a map is displayed on the user’s screen. Using its latitude and longitude, and with the help of Google maps APIs, I was able to display the current location of the user onto the map. At the bottom of the screen there is a menu with different icons, each one intended to display either markets, restaurants, schools, or hospitals. Each time a user clicks on one of the icons the nearby places of the user’s current location are displayed on the screen within a perimeter of 1000 meters. Once the user clicks on a specific place, information containing the name, address, opening hours, rating, and a picture are displayed to the user. Along in the same activity there are three buttons; “Show on Map” where the user can open the selected place on the actual google map, “View Direction” where the user can see the path leading to that specific place, and finally “List” this button is meant only for markets as it will display the list of available products along with their prices of a given store. What is yet to be added, is to have a basket for every user where he can add all of his desired products and he can modify it as he wishes. This basket will also be seen by the owner of the market, so that he can go and gather all of
the chosen products so that when the customer comes he will find all of his products already gathered for him, all what is left for him to do is to pay.

During the implementation of this project I have faced a lot of constraints and difficulties, most of them were errors in the form of null pointers exceptions that required a lot of debugging. The error that was the most frustrating for me and that made me waist a lot amount time was an Over Query Limit in Google Maps API, what was basically happening was that google Maps platform changed their policies allowing users to have only one request per day so when I clicked on the icon to display the nearby restaurants that was it, I could only see that. If I clicked on another icon I would have the over query limit problem and so I would have to wait till the next day for another request. After extensive researches, I found that you needed to enter your credit card information and that google maps API gives you a year for free to have unlimited requests per day which allowed my project to function correctly.
11 FUTURE PERSPECTIVES

Due to time constraint, I was not able to accomplish all the functionalities that I wanted. For the time to come, as future work I would add the basket option where every user would have his own basket with the different product he has chosen and he would be able to modify it by adding or removing items as he wishes. I would also add another activity that would display specific information for each product, whenever a user clicks on a product available on the list a new activity will appear containing a picture, the price, a brief description, and the rating, there would also be a review section where customer could leave a comment describing their experience with the product. Another aspect of my future work is to make Food Express accessible to IOS users as well.
12 REFERENCES


13 SCREENSHOTS OF THE CODE

Figure 11.1. Screenshot of the SignUp Activity.

Figure 11.2. Screenshot the Login Activity.
Figure 11.3. Screenshot of the Main Activity.

Figure 11.4. Screenshot of the Maps Activity.
Figure 11.5. Screenshot of the View Place Activity.

Figure 11.6. Screenshot of the View Direction Activity.