Ibn Zohr University

High School of Technology, Agadir

Department of Computer Engineering

Final Year project report

Realization of a mobile application 'Health Maintaining'

By : Mr Mouaad Sadik

& Mlle Mariam Sebbar

& Mlle Sana Er-Rachedy

Framed by : Mr. Mourad Gridach

Academic Year : **2018/2019**

المملكة المغربية
ⵜⴰⴳⵍⴷⵉⵜ ⵏ ⵍⵎⵖⵔⵉⴱ
وزارة التربية الوطنية
ⵜⴰⵎⴰⵡⴰⵙⵜ ⵏ ⵜⵎⴷⵢⴰⵣⵜ ⴰⵏⴰⵎⵓⵔ

Ibn Zohr University

High School of Technology, Agadir

Department of Computer Engineering

المدرسة العليا للتكنولوجيا -أكادير
ⵜⵉⵎⵙⵙⵓⵔⵉⵏ ⵜⴰⵏⴰⵎⵓⵔⵜ ⵏ ⵜⵉⴽⵏⵓⵍⵓⵊⵢⵉⵏ - ⴰⴳⴰⴷⵉⵔ
ÉCOLE SUPÉRIEURE DE TECHNOLOGIE - AGADIR

Final Year project report

Realization of a mobile application 'Health Maintaining'

By : Mr Mouaad Sadik

& Mlle Mariam Sebbar

& Mlle Sana Er-Rachedy

Framed by : Mr. Mourad Gridach

Academic Year : **2018/2019**

Dedications

We would like to dedicate this work and our deep gratitude

To our dear parents for their love and sacrifices.

To our sisters and brothers for their support and encouragement.

To all our friends.

To these caring educators, we dedicate the fruit of our student career.

To those who devote themselves unceasingly to enlighten us The way and vast horizons of knowledge and whose vocation deserves our respect.

To all who have contributed from near or far to the development of this modest work

To all those who love us and love them.

Acknowledgements

Before beginning this report, we would first like to express our sincere feelings and thanks to all those who participated directly or indirectly in the preparation of this Report.

We would like to extend our sincere thanks to our mentor, **Mr.Mourad Gridach**, for his welcome, his help and all his pertinent remarks.

We also wish to express our thanks to the faculty and administration of the School of Technology for the quality of teaching.

Finally, a big thank you to our families and friends for their advice and their unconditional support, both moral, economic and intellectual throughout our internship.

Illustration Index

List of abreviations

Abreviation	Meaning
AI	Artificial Intelligence
CNN / CnvNet	Convelutional Neural Network
ML	Machine Learning
RELU	Rectified Linear Unit

Summary

CHAPTER 3: Realization and implementation of the final project

GENERAL INTRODUCTION

If your health fails, it can overshadow everything else that's going on in your life. From relatively minor health issues such as aches and pains, lethargy, and indigestion to major health problems that can threaten your existence, health can really affect happiness and stress levels. Making a commitment to taking on healthier habits can have a far-reaching payoff : you'll feel better in everything you do.

Our project is realized as part of ending studies project, considered as exploit of enrichments and skills acquired during the two years of DUT and in order to deepen the knowledge gained during our studies in High School of Technology Agadir in a project ; to confront in our specialization and the workplace, that was the key incentive of choosing the topic of final project '**Realization of a mobile application -Health Maintaining-**' .

Sometimes medicines may affect the health revesely by side effects and health disorders. Futhermore, eating unhealthy foods can cause weight gain, food saturated of calories or 'bad' fats can raise blood pressure and cholesterol. This issues will increase chances of gaining weight and having other health problems like heart disease and diabetes. As a solution, and in order to solve dilemma of gaining weight, This project aim fixing this issue implemented on the creation of an Android application using Artificial Intelligence.

Our work is divided into two main parts : The first is the theoretically studies of the requirements specification and the second is the build of the application.

In this document, we will present three chapters:

In the first, we will present the specifications and the different development tools.

In the second chapter, we will develop a detailed use case diagram, sequence diagrams, activity diagrams and classes.

In the last chapter, we will detail the evolvement of our application by presenting the different realization techniques. We will, also, give a view of the application in its final state by presenting its different interfaces.

CHAPTER 1:

Presentation of requirements specification and development tools

Introduction :

In this chapter, we will begin with a presentation of our project. Then we will determine the issues behind realizing this application then, the goals to reach. Recently, we will represent the development tools and the technological choice.

1 Requirements specification :

1.1 Project overview :

The project is about creating an Android application that in general aims suggesting to the user a perfect diet that provides to the body the nutrients it needs to function properly. Moreover, to get the appropriate diet, we offer a smart plan based on the consumption of the majority of daily calories in fresh fruits and vegetables, whole grains, nuts and lean proteins.

The intelligent program is implemented with Artificial intelligence (AI) **-** a branch of computer science that aims to create intelligent and smart machines that mimics human cognition **-** , in general what is called Machine Learning (ML), an application of artificial intelligence that provides systems the ability to automatically learn and improve from experience without being explicitly programmed.

This program is based on the Convolutional Neural Network (CNN, or ConvNet), which is a class of deep neural networks, most often applied to visual imagery analysis.

1.2 Problematic :

The most common reason for going on a diet is to lose weight , also help lower blood pressure levels, improve cholesterol numbers by increasing high-density lipoprotein, make future heart attacks or strokes less likely, help live longer and have more energy. Body uses triglycerides to store any extra calories

consumed, but having high levels can increase risk for heart problems also breast cancers are linked to obesity. Some studies have also reported links between obesity and cancers of the gallbladder, ovaries, and pancreas.

Our project implements a special diet to lose weight and take care of health. When the problem was related to health, several questions arise :

- How a balanced diet will helps to weight loss ?
- How to avoid bad habits that lead to a fat belly ?
- How to balance between diet and diseases ?

1.3 Objectives :

Our application is a mobile application that serves to control the health of the user by proposing a suitable diet and this from several criterias :

- The age and weight of the user.
- The diseases of the user.
- The gender of user (Female or Male).

This diet will be a kind of recipes (dishes, desserts, juicesm glaces ...) offered by the application, of fruits and vegetables in the refregirator of the user classifying using image classification program -process to categorize all pixels in a digital image into one of several land cover classes or themes-, wich aims :

- Consume a variety of nutrient-dense foods within and across the food groups.
- Limit the intake of saturated and trans fats, cholesterol, added sugars, sodium (salt), and alcohol.
- Limit caloric intake to meet caloric needs.
- No need to go to the supermarket, the application provides what the user has at home.

In addition, users with an account have free access and a better service.

1.4 Actors :

The visitor: An individual who is searching on Play Store, looking for a good program to start his daily diet. Until this stage it's an unknown user.

The user: A visitor who already has an account in the application, so he can start his program for losing weight and forgot about obesity.

1.5 Requirements analysis :

1.5.1 Functional requirements :

The functional requirements are presented in four main parts:

- User registration by giving his information, the most important is his age, his gender, and his diseases.
- Authentication of the user by an email and password.
- Establishment of the diet from fruits in his refrigerator.
- Edit profile.

1.5.2 Non-functional requirements :

The non-functional requirements are presented in four main parts:

- **Reliability:** The application must work consistently without errors and must be satisfactory.
- **Errors:** Ambiguities must be indicated by well organized error messages for good guide the user and familiarize them with our application.
- **Ergonomics and good interface:** The application must be adapted to the user without it providing no effort (clear and easy use) of navigation point of view between different pages, colors and setting in texts used.
- **Maintenance and reutilisability:** The system must conform to a standard architecture and clear for its maintenance and reutilisability.

2 Development tools :

Softwares:

- **Anaconda :** Anaconda is a free and open source distribution of python programming language for large-scale data processing, predictive analytics, and scientific computing, that aims to simplify package management and deployment.

- **Jupyter Notebook :** Is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

- **Android Studio (Virtual device manager) :** The AVD Manager provides a graphical user interface in wich you can create and manage Android Virtual Devices (AVDs), wich are required by the Android Emulator.

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

- **Enterprise Achitect :** Is a visual modeling and design tool based on the OMG UML. The platform supports: the design and construction of software systems; modeling business processes; and modeling industry based domains.

- **Adobe Photoshop :** Adobe Photoshop is the predominant photo editing and manipulation software on the market. Its uses range from the full-featured editing of large batches of photos to creating intricate digital paintings and drawings that mimic those done by hand.

- **Adobe Illustrator** : Adobe Illustrator is a professional vector-based design and drawing program. Used as part of a larger design workflow, Illustrator allows for the creation of everything from single design elements to entire compositions. Designers use Illustrator to create posters, symbols, logos, patterns, icons, etc.

2.1 Programming languages :

- **Python 3.7 :** Python is an interpreted language & high-level programming language widely used in general programming purpose especially big-data and machine intelligence development. It has a large and comprehensive standard library.

- **Dart :** Dart is a general-purpose programming language originally developed by Google and later approved as a standard by Ecma. It is used to build web, server, desktop, and mobile applications.

2.2 Frameworks :

- **Tensorflow :** TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks.

- **Keras :** Keras is an API designed for human beings, not machines. It puts user experience front and center. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs.

- **Tensorflow Lite :** TensorFlow Lite is TensorFlow's lightweight solution for mobile and embedded devices. It enables on-device machine learning inference with low latency and a small binary size. TensorFlow Lite also supports hardware acceleration with the Android Neural Networks API.

- **Flutter :** Flutter is an open-source mobile application development framework created by Google. It is used to develop applications for Android and iOS, as well as being the primary method of creating applications for Google Fuchsia.

- **Firebase :** Firebase is a Backend-as-a-Service -BaaS- that started as a YC11 startupand grew up into a next-generation app-development platform on Google Cloud Platform, and gives functionality like analytics, databases, messaging and crash reporting .

- **ML Kit :** Machine learning kit can make apps more engaging, personalized, and helpful, and provides solutions that are optimized to run on device. ML Kit offers the technologies that have logn powered Google's own experiences on mobile. Use out-of-the-box solutions (base APIs) or custom models, running on device or in the Cloud, depending on your specific needs.

Conclusion :

In this chapter we have presented the project overview , as well as it fonctionnalites, the tasks we will realize, the presentation of tools and language used in the realization of our mobile application .

CHAPTER 2:

Analysis and concept

Introduction :

In our project, design is a crucial and decisive phase for produce a high quality application. It is in this stage that we first need to clarify the view, by describing the general architecture that we will follow in the realization part of our project.

Then, in a second place we will detail our conceptual choice through several types of diagrams.

1 The model :

In order to develop our model, we used the Keras library with a Tensorflow backend , the keras library helps us build our convolutional neural network (CNN) ,so we followed many steps :

1.1 Prepare data :

- Download dataset of Fruits from **Kaggle.com,** which are divised into two folders , training data and testing data.
- Specify output classes, we have 53 classe of fruits **[** Apple**,** Apricot**,** Avocado**,** Banana**,** Cactus fruit**,** Cantaloupe**,** Carambula**,** Cherry**,** Chestnut**,** Clementine**,** Cocos**,** Dates**,** Granadilla**,** Grape**,** Grapefruit, Guava**,** Hazelnut**,** Huckleberry**,** Kaki**,** Kiwi**,** Kumquats**,** Lemon**,** Limes**,** Lychee**,** Mandarine, Mango**,** Mangostan**,** Maracuja**,** Melon Piel de Sapo**,** Mulberry**,** Nectarine, Orange**,** Passion Fruit**,** Peach**,** Pear**,** Pepino, Physalis**,** Physalis with Husk, Pineapple**,** Pitahaya Red**,** Plum**,** Pomegranate**,** Pomelo Sweetie**,** Quince, Rambutan**,** Raspberry**,** Redcurrant**,** Salak**,** Strawberry**,** Tamarillo**,** Tangelo**,** Walnut **]**.
- Import training and testing images from the dataset file using a path and name of class.

- Resize images from the original size (100x100) to (60x60).
- Create Traing images and Testing images (features and labels), features contains array of images, labels contains number of classes (from 0 o 52).
- The data must be preprocessed before training the network. If you inspect the first image in the training set you will see that the pixel values fall in the range of 0 to 255, we scale these values to a range of 0 to 1 before feeding to the neural network model. For this we divide the by 255. It's important that the training set and the testing set are preprocessed in the same way.
- Display the first 25 images from the training set and display the class name below each image to verify that the data is in the correct format and we are ready to build and train the network.

1.2 Build the model :

We used a sequential model which we :

- **Setup the layers:** We implemented 4 layers :
  - The first one in this network provide the keyword argument input_shape (60x60x3), this layer extract features from an input image and preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filter or kernel, it has 64 nodes (or neurons).
    Relu stands for Rectified Linear Unit for a non-linear operation. The output is $f(x) = max(0,x)$.

model.add(Conv2D(64,Kernel_size=3,activation='relu',input_shape(60,60,3)))

  - The second layer is the same as the first but we don't have an input shape because it's already montionned , it has 32 nodes .

model.add(Conv2D(32,Kernel_size, activation='relu'))

- The third layer transforms the format of the images from a 2d_array , to a 1d-array of 100352 pixels.

model.add(Flatten())

- The last layer is densely-connected, or fully-connected, neural layers, it is a 53 node softmax layer, this returns an array of 53 probability scores that sum to 1. Each node contains a score that indicates the probability that the current image belongs to one of the 53 classes .

model.add(Dense(53, activation = 'softmax'))

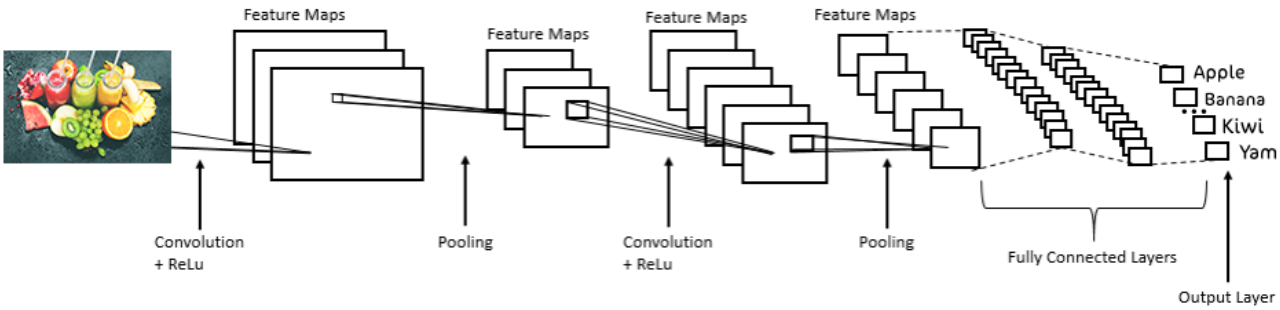

Figure 1: Complete CNN Architecture

- **Compile the model :** Before the model is ready for training, it needs a few more settings.These are added during the model's compile step:
  - **Loss Function :** This measures how accurate the model is during training. We want to minimize this function to "steer" the model in the right direction.
  - **Optimier :** This is how the model is updated based on the data it sees and its loss function.

- **Metrics :** Used to monitor the training and testing steps.The following example uses accuracy, the fraction of the images that are correctly classified.

Model.compile(optimizer = 'adam', loss = 'sparse_categorical_crossentropy' , metrics=['accuracy'])

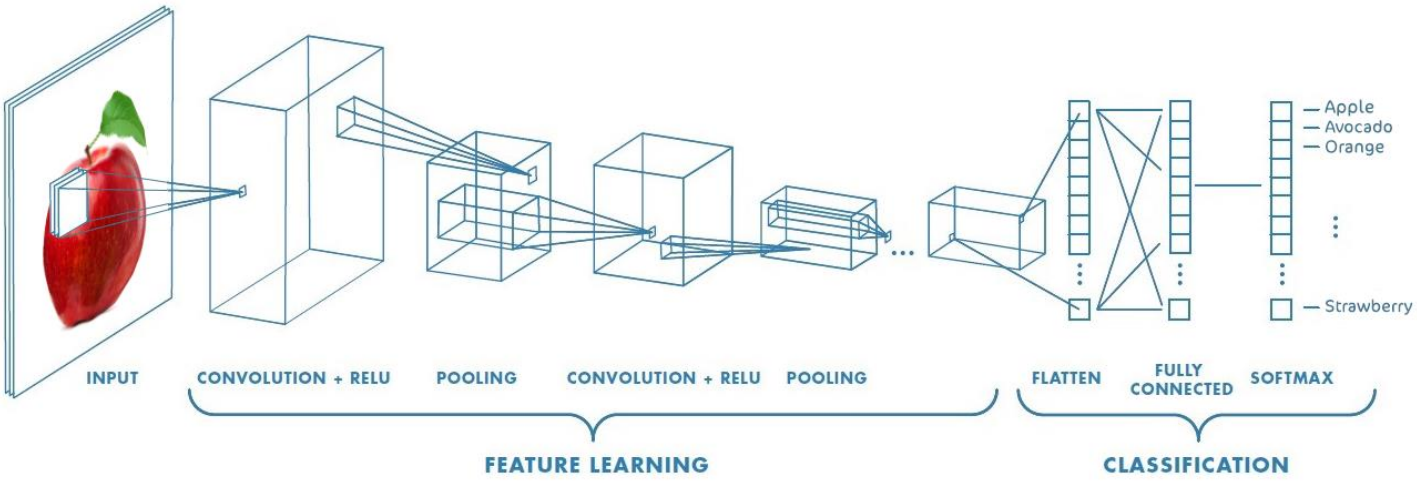

Figure 2:Neural Network with many convolutional layers

- **Train the model :** Training the neural network model requires the following steps:
  - Feed the training data to the model (the train_images(X) and train_labels(y) arrays).
  - The model learn to associate images and labels.
  - We ask the model to make predicitions about test set :the test_images(test_X) array.
  - We verify that the predictions match the labels from the test_labels(test_y) array.
  - To start Training, we call the model.fit method.

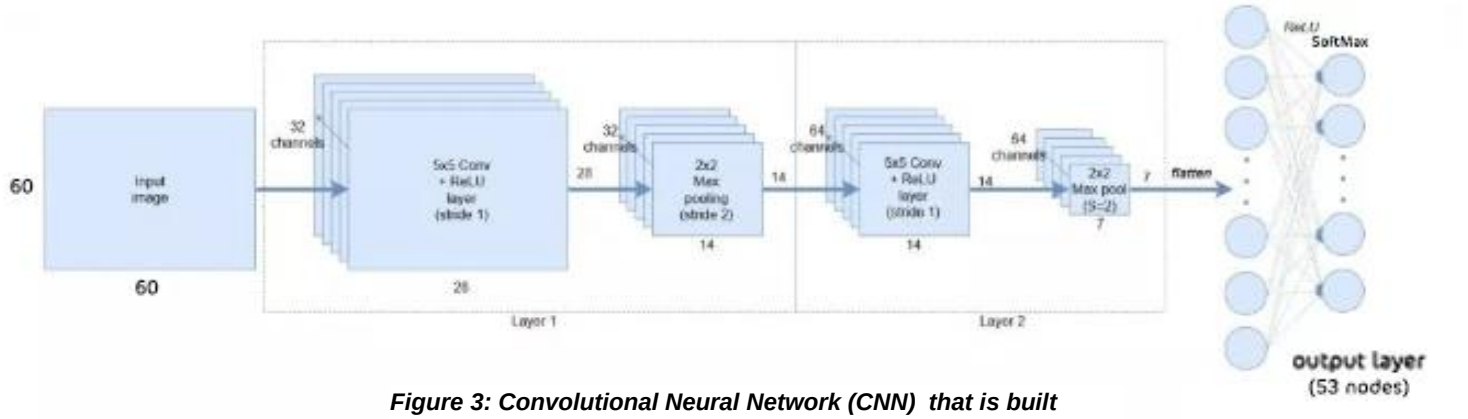

Figure 3: Convolutional Neural Network (CNN) that is built

history = model.fit(X, y, validation_data = (test_X, Test_y) , epoch=10 , batch_size = 128)

1.3 Evaluate accuracy :

Compare how the model performs on the test dataset:

It turns out, the accuracy on the test dataset is a little less than the accuracy on the training dataset. This gap between training accuracy and test accuracy is an example of overfitting. Overfitting is when a machine learning model performs worse on new data than on their training data.

We achieved in : **Train Accuracy : 100%** and in **Test Accuracy : 96.50%** .

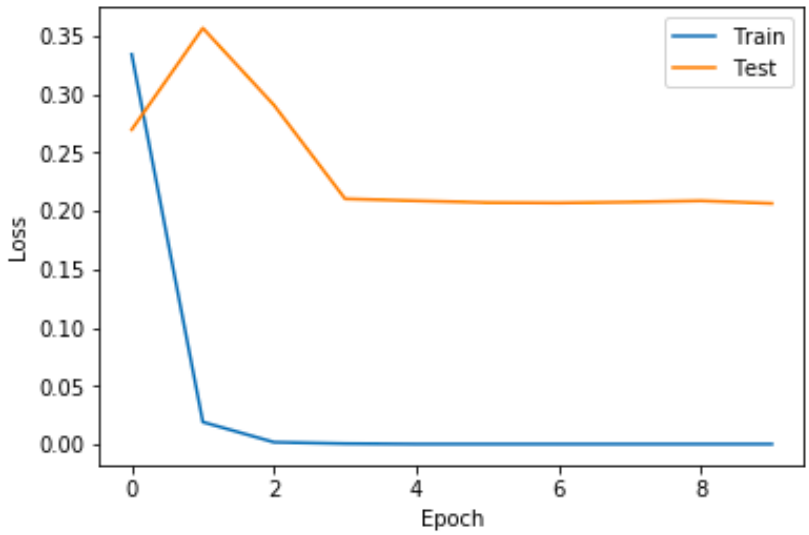

Figure 4: Plot of Model Accuracy on Train and Validation Datasets

From the plot of accuracy, we can see that the model has comparable performance on both train and validation datasets (labeled test). The test accuracy of the model is high which shows that the model is good.

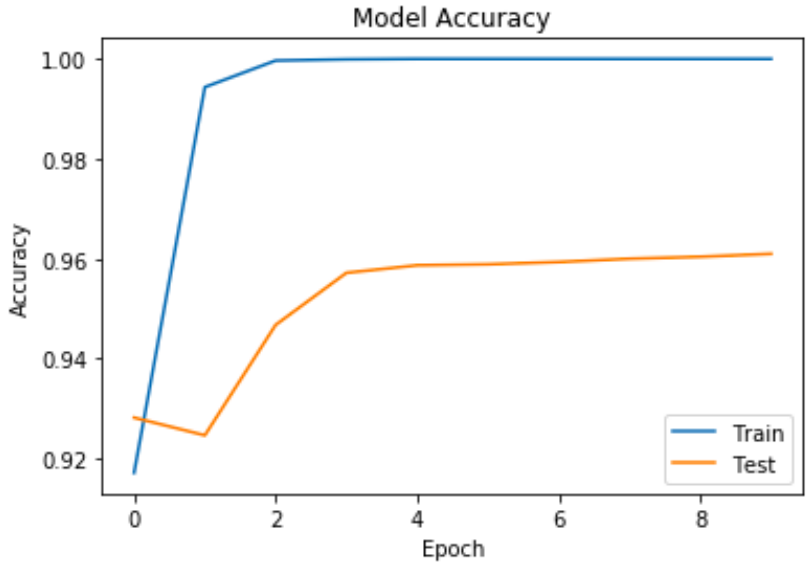

Figure 5: Plot of Model Loss on Training and Validation Datasets

From the plot of loss, we can see that the model has comparable performance on both train and validation datasets (labeled test). If these parallel

plots start to depart consisntently, it might be a sign to stop taining at an earlier epoch .

2 The application :

2.1 Use Case diagram :

The role of use case diagrams are to collect, analyze, and organize needs, as well as that the census So it was made of the first stage UML for the design of a system.

The case diagram consists of three main elements:

- **Actor:** it is the idealization of a role played by an external person, a process or a choice that interacts with a system.
- **Use case:** it is a coherent unit representing a visible functionality .
- **Relationships:** Three types of relationships are supported by the UML standard and are graphicallyrepresented by particular types of these relationships.

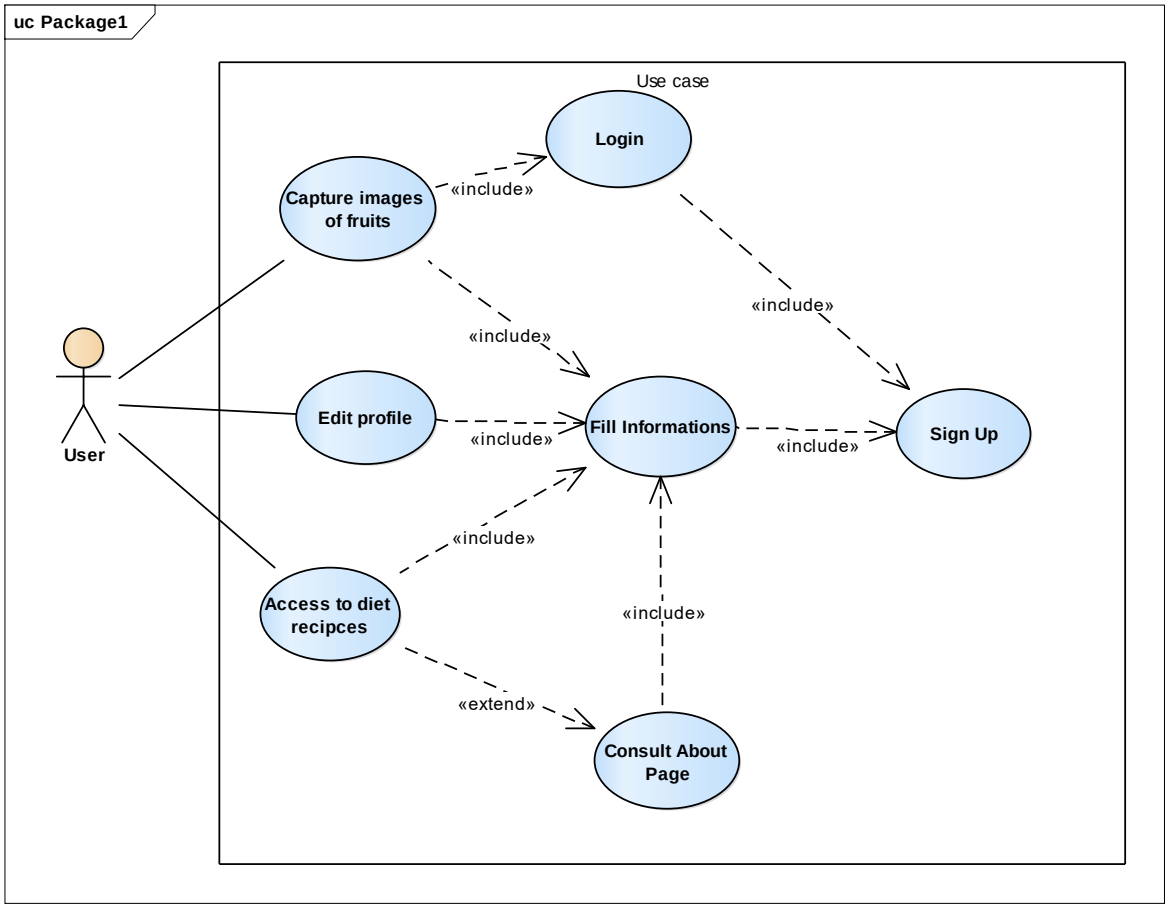

Figure 6: Use Case diagram 'User'

2.2 Class diagram :

The class diagram is a schema used in software engineering to introduce classes, interfaces systems and the different relationships between them. This diagram is part of the static part of UML because it ignores temporal and dynamic aspects.

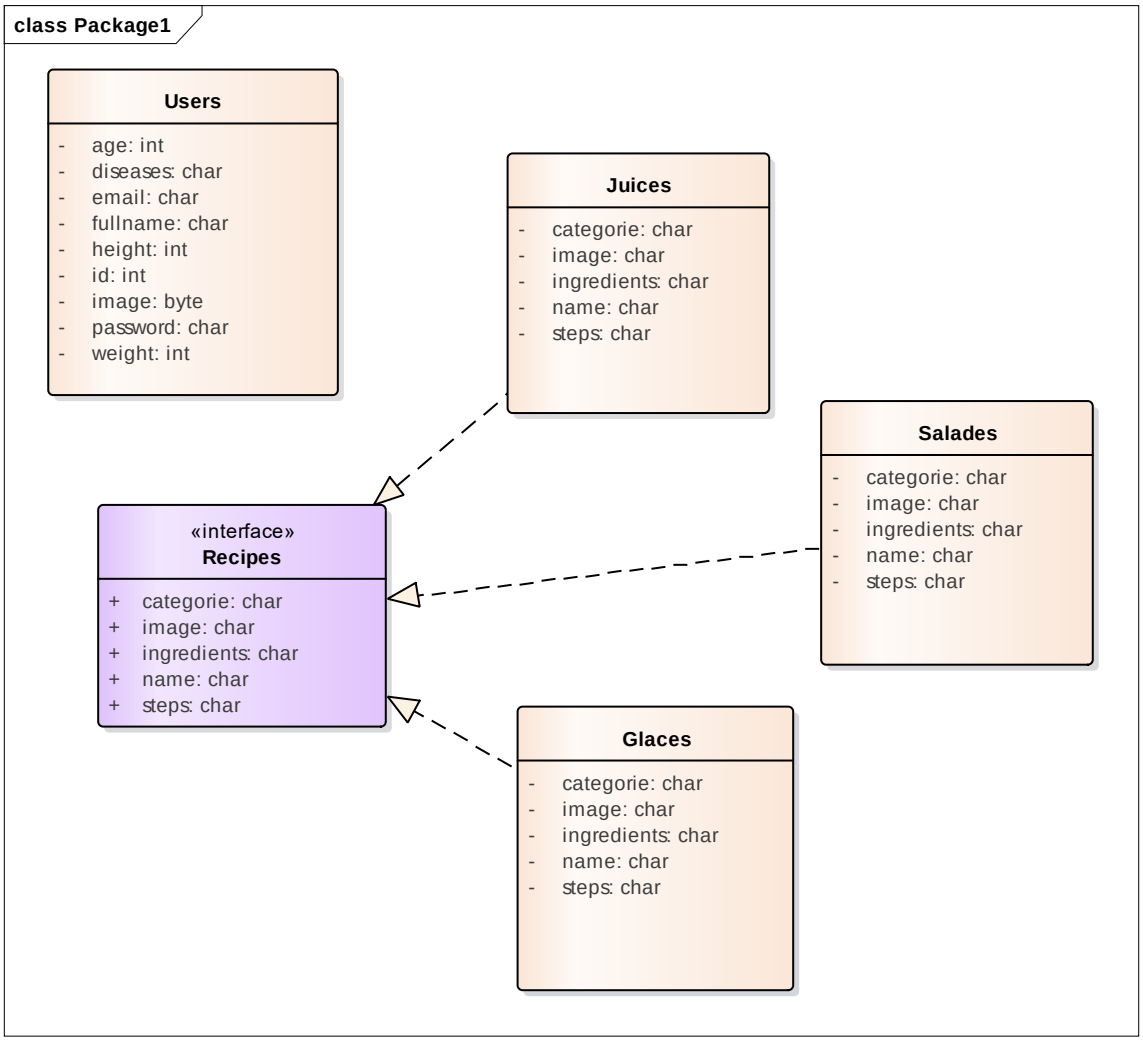

Figure 7: Class Diagram

2.3 Sequence diagram :

A sequence diagram is an interaction diagram that details how operations are done: what messages are sent and when they are.

Sequence diagrams are organized according to the time that passes as we go through the page. The objects involved in the operation are listed from left to right depending on when they take part in the sequence.

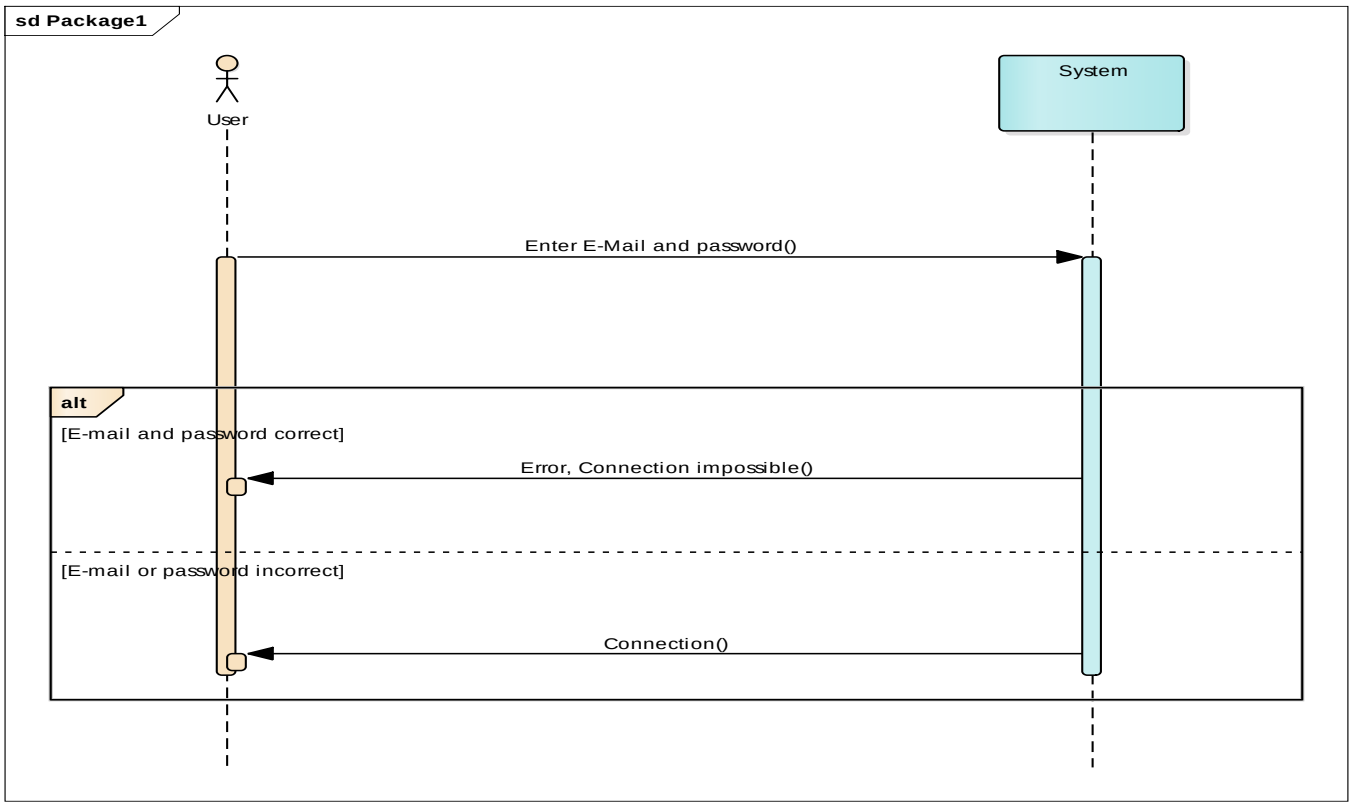

Figure 8: Sequence diagram "Login"

2.4 Activity diagram :

An activity diagram is a diagram associated with a particular object or set of objects, which illustrates the flows between activities and actions. It allows to graphically represent the progress of a use case.

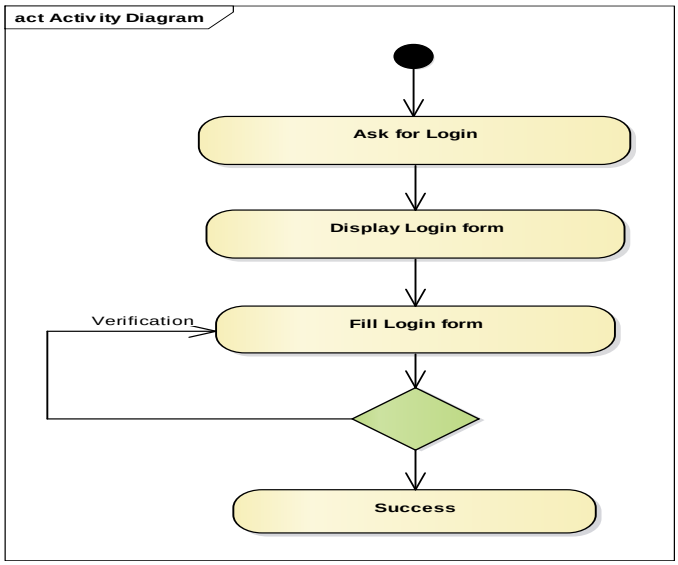

Figure 9:Activity diagram "Login"

Conclusion :

The conceptual phase is a fundamental step for the realization of any project. She allows to facilitate the information system and realize the implementation of the database and processing. Over there then we need to look for ways and tools to develop the application, what we will present in the next chapter.

CHAPTER 3:

Realization and implementation of the final project

Introduction :

This chapter aims to present the final product. In this part, we will present some mobile interfaces in order to locate you in the project.

1 SignUp Page :

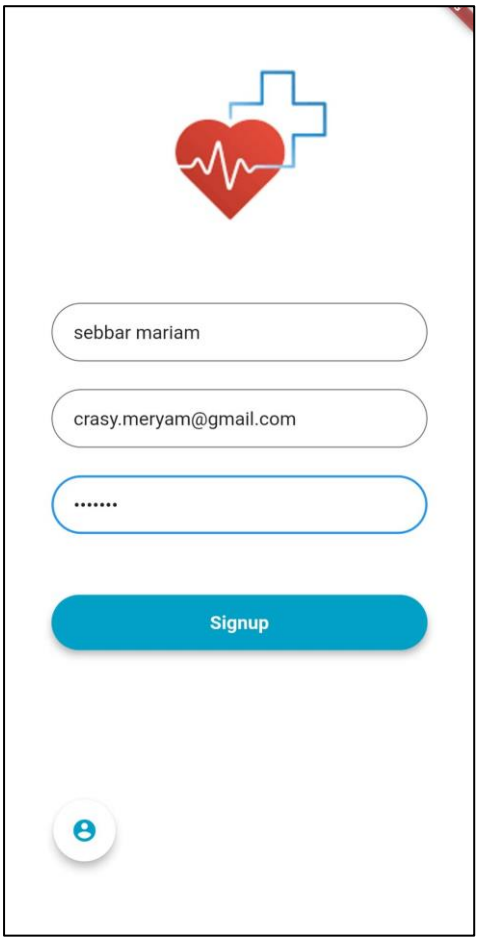

Figure 10: SignUp Page (2)

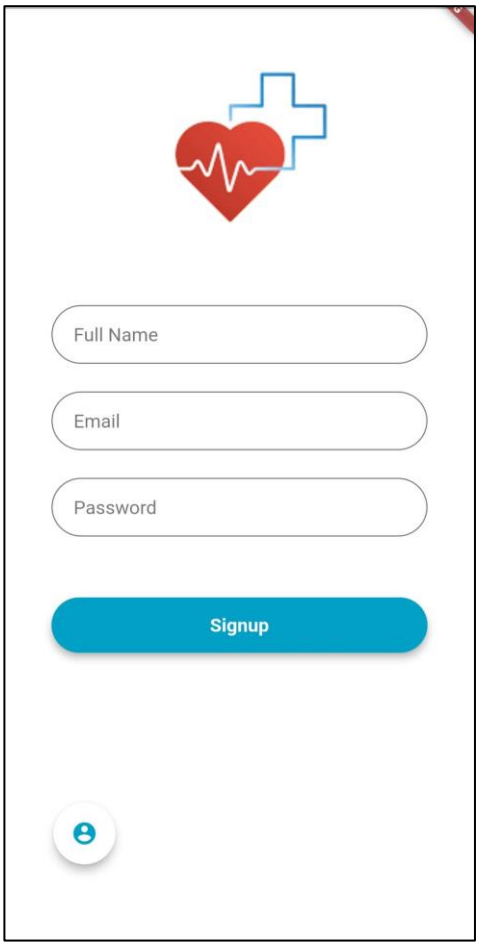

Figure 11: SignUp Page (1)

This page allows the user to subscribe to our application and benefit from its services, if the fields are empty or the email already exists, an error message is displayed.

2 Login Page :

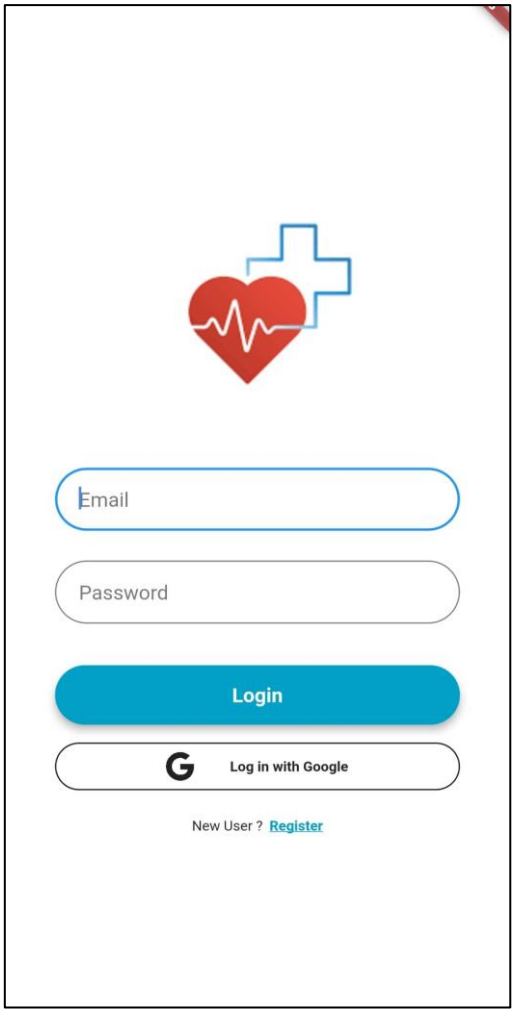

Figure 13: Login Page (1)

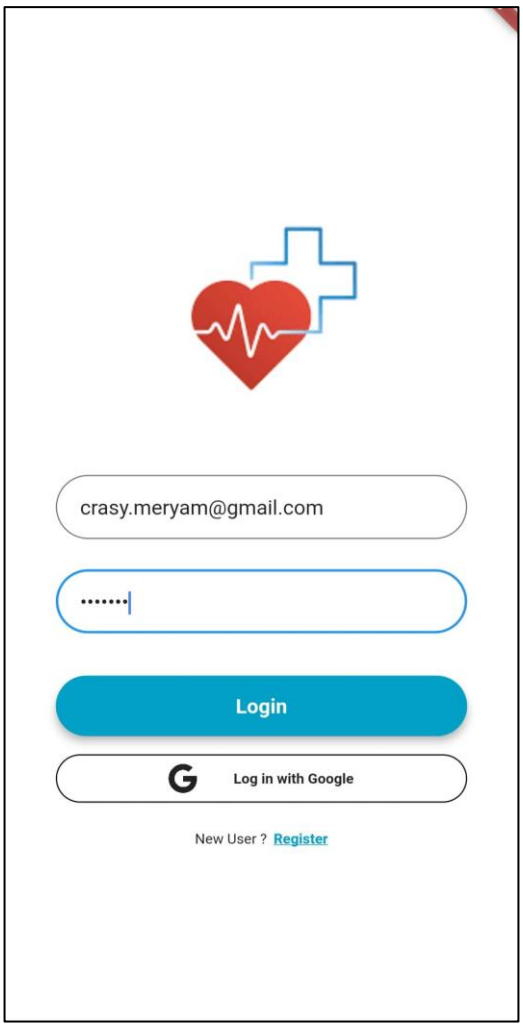

Figure 12: Login Page (2)

This page allows the use to authentificate via an email and password , to benefit from services of application, if email or password are wrong or the fields are empty , an error message is displayed.

3 Home Page :

Figure 14:Home Page (1)

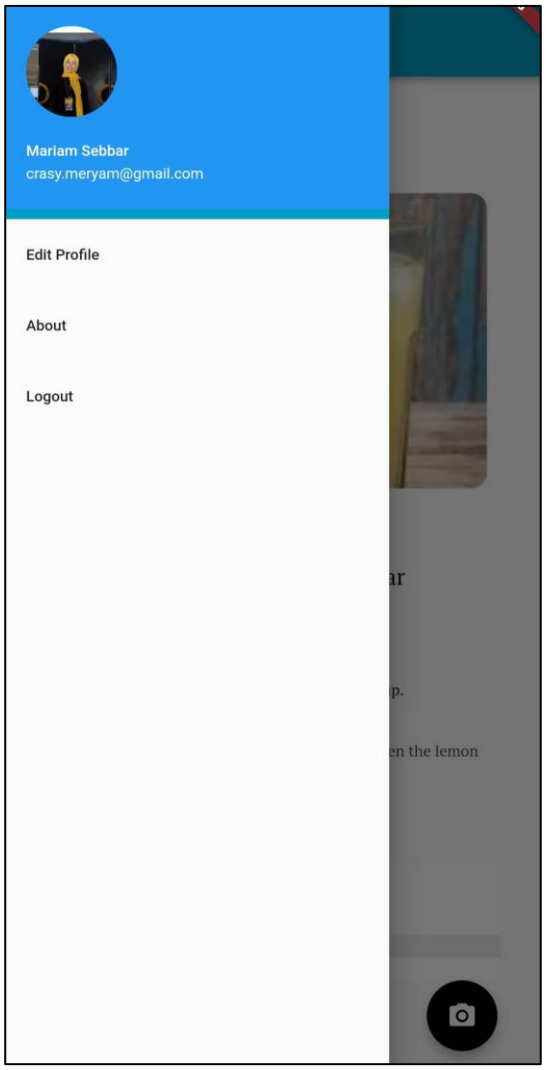

Figure 15: Home Page (2)

Home page is the page that appear once the suer is logged in , it contains a lider that shows how application work, and a button of a camera that allows taking shoots, it contains also a toogle menu bar that shows the image, email and name of the user , and possibility to access About Page, edit profile and log out.

4 Recipes Page:

Figure 16: Recipes Page

This page is a result of taking picture and classify the fruit in the picture, so for example if the system classify a Lemon , the recipes where Lemon is an ingredient will appear.

5 About Page :

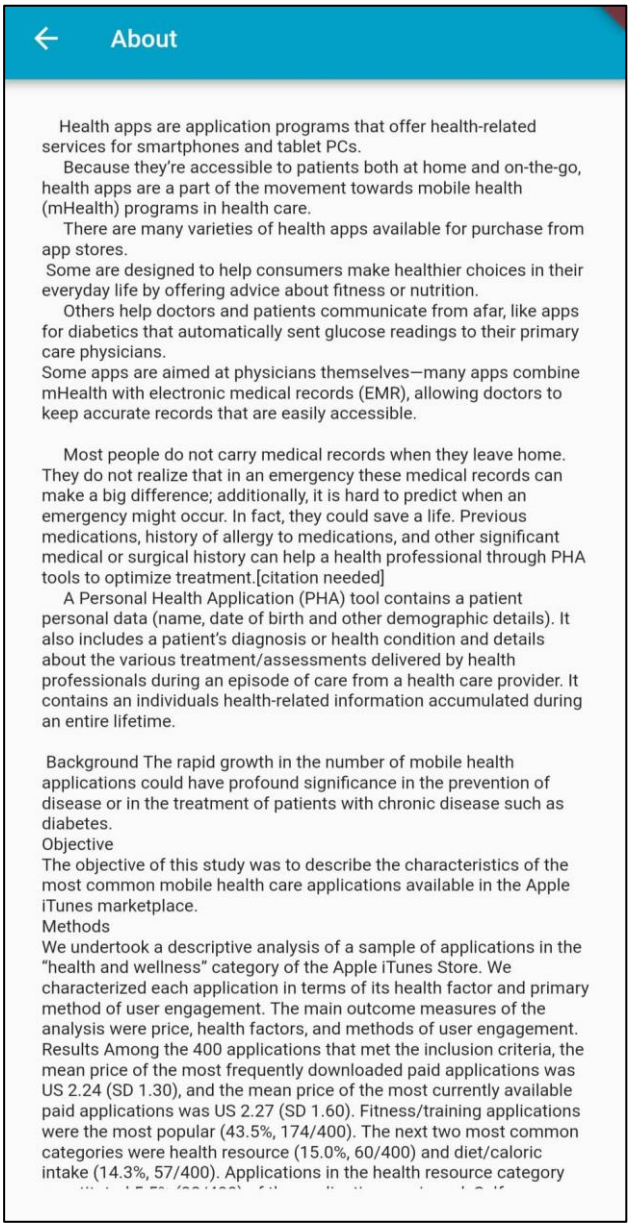

Figure 17: About Page

This page is an article about the application .

Conclusion :

In this chapter we called to present the interfaces realized in our mobile application for clarify the steps of using our application.

GENERAL CONCLUSION

In our Final Project : '**Realization of a mobile application -Health Maintaining-**' , we realized a program which implements a special diet to lose weight and take care of health.

The goal of our project was, first, to consume a variety of nutrient-dense foods within and across the food groups. Second to limit the intake of saturated and trans fats, cholesterol, added sugars, sodium (salt), and alcohol. And third, to limit caloric intake to meet caloric needs.

Indeed, for all these requirements, we have designed as an alternative solution : the mobile application that is based on Machine Learning and image classification, so the users can begin their diet which will speed up access to the desired goals which is a balanced health .

However, during our realization of this modest work, we were confronted at the beginning with a certain number of difficulties related to the documentation, the constraint of the time, the lack of space to work with the whole trinomial and the ignorance of computer languages not yet taught and which were essential for the realization of our project like building a model using Tensorflow Keras.

Finally, to improve the component, we could have integrated vegetable classification. But, for lack of time, and the complexity of this task were the major factors that put us off.

However, would it not be a motivating force that would allow us to fill our gaps in order to follow up this component in our future projects?

Webographie:

http://alexlenail.me/NN-SVG/LeNet.html

https://medium.com/@sidereal/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5

https://www.photopea.com/

https://www.quora.com/What-tools-are-good-for-drawing-neural-network-architecture-diagrams

https://www.chachatelier.fr/latexit/

https://about.draw.io/draw-io-training-exercise-7-create-a-diagram-with-layers-and-images/

http://www.androiddocs.com/tools/help/avd-manager.html

https://www.healthypeople.gov/2020/topics-objectives/topic/nutrition-and-weight-status

https://www.eatthis.com/weight-loss-questions/