جامعة القاهرة الجديدة التكنولوجية
NEW CAIRO TECHNOLOGICAL UNIVERSITY

Faculty of Industry and Energy Technology

Mobile Application Using Flutter and Dart

A capstone project submitted to faculty of Industry and Energy Technology, New Cairo Technological University, In partial fulfillment of the requirements for the Degree of Higher Diploma

Information and Communications Technology

Submitted by

- Ahmed Nabil Mahmoud Mahran
- Hadeer Ahmed Mahmoud
- Kareem Ayman Na'eem
- Mai Farah Hanafy
- Omar Essam Mohamed Aly

Supervised by

Assistant Prof. Ahmed Fares
Information and Communications Technology Department
faculty of Industry and Energy Technology
New Cairo Technological University

Eng. Roula Mohamed
Information and Communications Technology Department
faculty of Industry and Energy Technology
New Cairo Technological University.

جامعة القاهرة الجديدة التكنولوجية
NEW CAIRO TECHNOLOGICAL UNIVERSITY

Faculty of Industry and Energy Technology

Approval Sheet

Mobile Application Using Flutter and Dart

A capstone project submitted to faculty of Industry and Energy Technology, New Cairo Technological University, In partial fulfillment of the requirements for the Degree of Higher Diploma

Submitted by

- Ahmed Nabil Mahmoud Mahran
- Hadeer Ahmed Mahmoud
- Kareem Ayman Na'eem
- Mai Farah Hanafy
- Omar Essam Mohamed Aly

This capstone project has been approved by the examining committee:

Score:----------

Name of the Examiner	Signature
Assistant Prof. Ahmed Hassan Fares Information and Communications Technology Department faculty of Industry and Energy Technology New Cairo Technological University	----------

TABLE OF CONTENTS

ACKNOWLEDGMENT 4

ABSTRACT 5

CHAPTER 1 6

1.INTRODUCTION 7

1.1 Overview 7

1.2 Problem defination 7

1.3 Literature Review 7

CHAPTER 2 9

2.About the project 10

2.1 Introduction 10

2.2 Objectives 10

2.3 Beneficiaries of the application 11

2.4 Scope 12

2.5 Constrains 12

2.6 Project features 14

CHAPTER 3 18

3.Planing & Requirement 19

3.1 Project Tools 19

3.2 Project Functionality 29

3.3 Functional Requirement 31

CHAPTER 4 32

4. Design of Project 33

CHAPTER 5 41

5.Future Improvements 42

Bibliography .....

Arabic Summary .....

First of all, we would like to thank God for granting us success in completing this work. Secondly, all thanks to our supervisors

Assistant Prof. Ahmed Hassan Fares

Eng. Roula Mohamed

for their continuous support and assistance in the capstone project, we very grateful for their strong effort, continuous support, and encouragement during the research study in this capstone project. They really influenced my way of thinking and developing the research ideas adopted in this this is. **And** *I would like also to express my deepest thanks all the members of my colleges for their cooperation during the period I*

spend with them to prepare this work.

ABSTRACT

Mobile app allows providers to effectively streamline communication between patients, providers, and their caregivers and allows for 24/7 management of a patient's condition along with the ability to personalize healthcare per patient.

And presents buying medicines and cosmetics through the app.

Through ML (Machine Learning) we made the prediction of COVID-19 is available which isn’t found in rest of Healthcare applications.

And read about the symptoms and know the common diseases of it.

CHAPTER I

INTRODUCTION

1 Introduction

1:1Overview:

Mobile app allows providers to effectively streamline communication between patients, providers, and their caregivers and allows for 24/7 management of a patient's condition along with the ability to personalize healthcare per patient. By MEDICA, here you can find a doctor by categories or location, buy medicines, know about symptoms, having an online consult and checking for Covid-19 by the Covid-19 Detector.

1:2 Problem Definition:

Generally Mobile apps allow users to have easy, functional access to information, products, services and processes that they need in real-time and are optimized for hands on interaction.

The problems that patients faced before the establishment of medical phone applications are the difficulty of communicating with the treating doctor in a continuous way to consult him.

Especially if the patient lives in a place far from the headquarters of the treating doctor, as well as other problems such as the difficulty of payment and reservation procedures, and all this was largely overcome through This application saves time and effort on the patient.

1:3 Literature Review:

Many medical applications for smart mobile phones have been developed and widely used by health professionals, doctors, consultant and patients.
The use of smart phones is getting more attention in healthcare and medical services day by day.
Medical service provider applications make smart mobile phones useful tools in the practice of evidence-based medicine at the point of care, in addition to their use in mobile clinical communication with a correct reference. Also, smart mobile phones can play a very important role in patient education, disease self-management, and remote monitoring of patients.

Some medical applications:

- Application 1: Dr On Demand
- Application 2: Halodoc
- Application 3: Vezeeta
- Application 4: HealthTap — 24/7 Telemedicine
- Application 5: FALLSAFETY
- Application 6: Amwell: Doctor Visits 24/7
- Application 7: WebMD: Symptoms, Rx, & Doctors

CHAPTER 2

About the Project

2:1 Introduction:

MEDICA MOBILE APPLICATION

Mobile app allows providers to effectively streamline communication between patients, providers, and their caregivers and allows for 24/7 management of a patient's condition along with the ability to personalize healthcare per patient.

And presents buying medicines and cosmetics through the app.

Through ML (Machine Learning) we made the prediction of COVID-19 is available.

And read about the symptoms and know the common disease of it.

2:2 Objectives:

Facilitated the Payment Process.

Meaning that the reservation and payment is made through the application, instead of the patient or one of his family having to go to the clinic of the doctor or hospital, and this will save time and great effort for the patient.

It’s better to Spend more time caring for your patients, and less time billing with healthcare payment processing solutions. Fawry Pay payment systems can improve patient satisfaction and simplify billing.

Connecting Between Patients and Doctors.

The patient appreciates that he communicates online with his doctor through the chat platform in the application through his home without having to go to the doctor so that he can follow up with him the new developments in his condition, his medications and the types of foods he is supposed to adhere to .... etc.

This will reduce the patient's exposure to stress or pollution in the street, which will help him to stabilize his health condition.

Creating Patient's Medical History.

The medical history of the patient is that a dedicated place is created in the application that contains all the information about the patient from personal information about him to information about his medical condition from diseases he suffers from other than the main complaint that he is currently suffering from, which helps doctors to make their therapeutic decisions through this information and refer to the patient's medical history in case of any changes in his health condition.

2:3 Beneficiaries of the application:

Generally, the average user will benefit from this application.

In particular

Elderly people who suffer from chronic diseases such as diabetes and high pressure	Older people with chronic diseases should not be exposed to physical stress through transportation or other things, except in an emergency case, because this will affect the stability of their health status.
Those who suffer from asthma and breathing problems.	as the pollution in the street from car exhaust ... etc. will lead to a worsening of their health condition.
Recently, during the COVID-19 Pandemic	Most of the cases receive treatment from the home ban. The application will help him to follow up his health condition and follow up the symptoms of the virus without going to the doctor, which will lead to reducing the transmission of infection and increasing cases.

2:4 Scope:

Our project is a service that helps people, and it is a medical mobile application that provides various services as mentioned previously. It targets patients, doctors, medicines providers, pharmacies. Not limited to booking with the doctor.

2:5 Constraints:

- MEDICA Patients App:

- Having a Stable Internet Communication.
- The User must be joined in app's data. If he didn't have an account. But if he already has an account, he can login through the app.
- It requires uploading a profile picture for the user to provide a better communication.
- If the user will join with Google Sign in Method, he must have a Google Account or add a new Google Account to join.
- The user's data must be real, because prior appointments and arrangements are made with the treating doctor, and if the data is fake, it will waste the doctor's time and ban the user of this data.
- Logically, The User shouldn't book more than one appointment with the same doctor with the same account Until he goes to the doctor and getting informed with the next appointments.
- The User must upload a real X-RAY Chest Picture for COVID-19 Detector to let the application and its Detector to make their analysis and inform you with the best Prediction.
- Must have Real Payment Method.

- ## MEDICA Doctors App:

  - Having a Stable Internet Communication.
  - The Doctor must be joined in app's data. If he didn't have an account. But if he already has an account, he can login through the app.
  - It requires uploading a profile picture for the user to provide a better communication.
  - Must Enter his Specialty in Registration Form. To inform the Users who discovering doctor's Application. Also should enter his Years of Experience, The Price of the Appointment, the number of Certificates he is already had and also we need his location. To give the patients know his location on google maps services. And enter Doctor's Picture to make it easier for the patient to communicate with the Doctor.
  - If the Doctor needs COVID-19 Detector. He will should the instructions of the Detector. The User must upload a real X-RAY Chest Picture for COVID-19 Detector to let the application and its Detector to make their analysis and inform you with the best Prediction.
  - The Doctor's data must be real, because prior appointments are made with people who really need help, and if the data is fake, it will waste the Credibility of the application patient's time and ban the Doctor of this data.

2:6 Project features:

- **Machine Learning COVID-19 Detector.**
    - The COVID-19 pandemic is an ongoing pandemic of coronavirus disease since 2019. Millions of cases and deaths attributed to it have been confirmed in the world.
    - So far the detection of COVID-19 heavily relies on the specialized tests (e.g., based on saliva or respiratory swabs).
    - Some approaches use smart devices (e.g., Whoop) for coronavirus infection detection using respiratory rate.
    - Machine learning (ML) techniques have become a promising approach for the coronavirus infection detection.
    - we introduce just a prototype and simulation based on machine learning to predict infection of COVID-19.
- **Analyze the User's X-RAY Chest Picture Deeply to get the Best Prediction and Helping the User.**
    - **This process is launched by:**
    - Sample an open source dataset of X-ray images for patients who have tested positive for COVID-19.
    - Sample "normal" (i.e., not infected) X-ray images from healthy patients.
    - Train a CNN to automatically detect COVID-19 in X-ray images via the dataset we created.
    - Evaluate the results from an educational perspective.
    - Then, give the user the final result which is processed. But it's the predicted result not a full true result.

- **Booking Appointments System with the best Doctors.**
    - The application is having a package of more than 45+ different specialties and departments. To give every user the flexibility for choosing any suitable doctor for him.
    - Our application provides all the information that you need about the doctor. Full Name and his specialty. Also we provide years of experience, Consulting price, number of certifications that the doctor.
    - The User have all options in choosing in which day he will book the appointment and in which time he need to meet the doctor.
- **Find Near Doctors.**
    - Every doctor in his registration on MEDICA app will enter his clinic's location to give the patient the availability to compare doctor's location with each other to book with the chosen doctor.
- **View the Doctor's Location on Google Maps to help the patient before booking an appointment.**
    - In every doctor's profile, you can click on map icon to switch to google map and it will be adjusted to your doctor's location.
    - In Google Map, you will see a custom marker for MEDICA app logo.
    - This is the current doctor's location. We made this step to be full recognizable by the user.
    - We don't prefer to put the default Google Map Marker. Which is red. Because it looks like the rest of place's Markers.

- **In Patient's Application, The User has a Separated Screen to Check his Requested Appointments.**
    - By default, the user will find My Appointments icon in every screen in MEDICA app to make it easier for the user to show his appointments.
    - If he hasn't appointments, this tab will be empty and will show title "No Appointments found".
    - If the user already has appointments, this tab will be injected by all his booked appointments.
    - This Screen already is connected with the firebase. The user is just booked an appointment. Then, it will be added at the same time in this Screen.
- **In Doctor's Application, The Doctor has a Separated Screen to Check his Booked Appointments.**
    - By default, the doctor will find My Appointments icon in every screen in MEDICA Doctor app to make it easier for the user to show his requested appointments.
    - If he hasn't any requested appointments, this tab will be empty and will show title "No Appointments found".
    - If the doctor already has requested appointments, this tab will be injected by all his booked appointments.
    - This Screen already is connected with the firebase. The patient is just booked an appointment with this doctor. Then, it will be added at the same time in this Screen.

- **Chatting Platform to establish the Communication with the Patient and his Doctor.**
    - Here you can find the list of doctors that you booked with in advance, and you can communicate with them and consult them.
    - All messages between any user or doctor in MEDICA app is secured and is stored already in firebase, there is a separated collection for all messages between users or doctor.
- **Buying Medicines Department.**
    - Here you can find a list of medicines that has been added previously on Firebase, you can add any product to your cart and choose the required quantity, then you can proceed to checkout.
- **Provide Fawry Pay Method for Payment while buying medicines or booking appointments.**
    - It the simplest payment method is available to test on it or working with it now in debug and development stages.
- **Getting Information about Symptoms and Diseases.**
    - If you want to get information about any of Symptoms on any Diseases, there a separated department in MEDICA app, that is provides the common Diseases and it is possible Symptoms to know about it from very simple Screen with some effects to be familiar with the user.
- **Separated apps (Doctor – Patient).**
    - Our team has decided to separate MEDICA app to 2 applications. One to the doctor for arranging their appointments and provide a stable application.
    - And one for the user, to give him the full functionality of MEDICA app.
    - It's be separated trying to get the stability of two applications.

CHAPTER 3

Planning & Requirement

3:1 Project Tools:

- **Flutter:**

  - Flutter is an open-source UI software development kit created by Google. It is used to develop cross platform applications for Android, iOS, Linux, macOS, Windows, Google Fuchsia, and the web from a single codebase.

Flutter Architecture:

- Dart platform.
- Flutter engine (Skia Graphics Engine).
- Foundation library.
- Design-specific widgets.
- Flutter Development Tools (DevTools).

Dart Platform:

- Flutter apps are written in the Dart language and make use of many of the language's more advanced features.

- While writing and debugging an application, Flutter runs in the Dart virtual machine, which features a just-in-time execution engine. This allows for fast compilation times as well as "hot reload", with which modifications to source files can be injected into a running application. Flutter extends this further with support for stateful hot reload, where in most cases changes to source code are reflected immediately in the running app without requiring a restart or any loss of state.

Flutter Engine:

- Flutter's engine, written primarily in C++, provides low-level rendering support using Google's Skia graphics library.

- Additionally, it interfaces with platform-specific SDKs such as those provided by Android and iOS. The Flutter Engine is a portable runtime for hosting Flutter applications. It implements Flutter's core libraries, including animation

and graphics, file and network I/O, accessibility support, plugin architecture, and a Dart runtime and compile toolchain. Most developers interact with Flutter via the Flutter Framework, which provides a reactive framework and a set of platform, layout, and foundation widgets.

Foundation library:

- The Foundation library, written in Dart, provides basic classes and functions that are used to construct applications using Flutter, such as APIs to communicate with the engine.

Design-specific widgets:

- The Flutter framework contains two sets of widgets that conform to specific design languages: Material Design widgets implement Google's design language of the same name, and Cupertino widgets implement Apple's iOS Human interface guidelines.

Gradle:

- Gradle is a build automation tool for multi-language software development. It controls the development process in the tasks of compilation and packaging to testing, deployment, and publishing. Package for deployment on any platform.

MVVM Design Pattern:

- Model–view–viewmodel (MVVM) is a software architectural pattern that facilitates the separation of the development of the graphical user interface (the view) – be it via a markup language or GUI code – from the development of the business logic or back-end logic (the model) so that the view is not dependent on any specific model platform.

- The viewmodel of MVVM is a value converter, meaning the viewmodel is responsible for exposing (converting) the data objects from the model in such a way that objects are easily managed and presented. the viewmodel is more model than view, and handles most if not all of the view's

display logic. The viewmodel may implement a mediator pattern, organizing access to the back-end logic around the set of use cases supported by the view.

- **TensorFlow:**
  - TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks.
  - TensorFlow is a popular term in deep learning, as many ML developers use this framework for various use cases. It provides the ease of implementing machine learning models and inferences for AI applications.
  - But TensorFlow Lite is a deep learning framework for local inference, specifically for the low computational hardware. It allows on-device machine learning by assisting developers in running their models on compatible hardware and IoT devices.
  - A developer needs to select a suitable model depending on the use case. The framework also gives an option of retraining the existing model on a custom dataset as well. As TensorFlow's protocol buffer model comes with a large size and requires advanced computational power, thus it enables the conversion of the TensorFlow model to the TensorFlow Lite model.
  - The customization of optimizing and quantizing parameters allows the reduction in model size and latency.

- **Tflite (Package):**
  - A Flutter plugin for accessing TensorFlow Lite API. Supports both iOS and Android. Supports image classification, object detection (SSD and YOLO), Pix2Pix and Deeplab and PoseNet on both iOS and Android.

- TensorFlow Lite is a deep learning framework for low size and low computational device and allows on-device machine learning for edge AI applications.

- **Teachable Machine:**

  - At first, learning AI and related concepts were very difficult and time taking. Many people feared to take a look at this interesting field due to the hardcore coding and statistical knowledge to create the AI models. However, with the creation of this amazing tool called Teachable Machine, work became easier and fast.

  - Teachable Machine is a web-based tool that makes creating machine learning models fast, easy, and accessible to everyone.

  - Teachable Machine is an AI, Machine Learning, and Deep Learning tool that was developed by Google in 2017 and it runs on top of Tensorflow.js that was also developed in the same company. This is a very powerful and user-friendly tool that helps in creating your Machine Learning and other AI models without having any prior knowledge of the same.

  - Teachable Machine is so advanced that it already does the training, testing, and validating part by itself. One just needs to upload his/her files that he wants to train like mp3, mp4, images, etc. and just specify the names of the file in this tool.

- **Github:**

  - GitHub is a social network for programmers. Git is a distributed software management program created by Linus Torvalds, originally for the Linux Kernel Development. GitHub is a hosted Git repository. Github allows you to take part in collaboration by forking projects, sending and pulling requests, and monitoring development.

  - It is used for manage the Project and make the Source Code is shareable with all the Team Members. To make the Coding is easier.

- **State Management (GET X):**
    - GetX is an extra-light and powerful solution for Flutter. It combines high-performance state management, intelligent dependency injection, and route management quickly and practically.
    - Generally, the developer should be concerned with removing controllers from memory. With GetX this is not necessary because resources are removed from memory when they are not used by default. If you want to keep it in memory, you must explicitly declare "permanent: true" in your dependency. That way, in addition to saving time, you are less at risk of having unnecessary dependencies on memory. Dependency loading is also lazy by default.
    - GetX has 3 basic principles. This means that these are the priority for all resources in the library: PRODUCTIVITY, PERFORMANCE AND ORGANIZATION.
    - PERFORMANCE: GetX is focused on performance and minimum consumption of resources. GetX does not use Streams or ChangeNotifier.
    - PRODUCTIVITY: GetX uses an easy and pleasant syntax. No matter what you want to do, there is always an easier way with GetX. It will save hours of development and will provide the maximum performance your application can deliver.
    - ORGANIZATION: GetX allows the total decoupling of the View, presentation logic, business logic, dependency injection, and navigation. You do not need context to navigate between routes, so you are not dependent on the widget tree (visualization) for this. You don't need context to access your controllers/blocs through an inheritedWidget, so you completely decouple your presentation logic and business logic from your visualization layer.

- With GetX you know where to find each feature of your application, having clean code by default. In addition to making maintenance easy, this makes the sharing of modules something that until then in Flutter was unthinkable, something totally possible. BLoC was a starting point for organizing code in Flutter, it separates business logic from visualization. GetX is a natural evolution of this, not only separating the business logic but the presentation logic. Bonus injection of dependencies and routes are also decoupled, and the data layer is out of it all. You know where everything is, and all of this in an easier way than building a hello world. GetX is the easiest, practical, and scalable way to build high-performance applications with the Flutter SDK. It has a large ecosystem around it that works perfectly together, it's easy for beginners, and it's accurate for experts. It is secure, stable, up-to-date, and offers a huge range of APIs built-in that are not present in the default Flutter SDK.

- **Firebase:**
    - Firebase is a platform developed by Google for creating mobile applications. It was originally an independent company founded in 2011. And it works with FlutterFire to merge Flutter with Firebase. FlutterFire is a set of Flutter plugins that enable Flutter apps to use Firebase services.

- **Firebase_core (Package):**
    - A Flutter plugin to use the Firebase Core API, which enables connecting to multiple Firebase apps.
    - The Firebase_core plugin is responsible for connecting your Flutter app to your Firebase project. The plugin must be installed and initialized before the usage of any other FlutterFire plugins. It provides basic functionality such as:
        - Initializing FlutterFire.
        - Creating Secondary Firebase App Instances.

- **Firebase_auth (Package):**

  - Flutter plugin for Firebase Auth, enabling Android and iOS authentication using passwords, phone numbers and identity providers like Google, Facebook and Twitter.

  Easy sign-in with any platform:

  - It aims to make building secure authentication systems easy, it provides an end-to-end identity solution, supporting email and password accounts, phone auth, and Google, Twitter, Facebook, and GitHub login, and more.

  Fast implementation:

  - It can take months to set up your own auth system. Now, set up the entire authentication system of your app in under 10 lines of code, even handling complex cases like account merging.

- **Google_sign_in (Package):**

  - Flutter plugin for Google Sign-In, a secure authentication system for signing in with a Google account on Android and iOS.

- **Cloud_firestore (Package):**

  - Flutter plugin for Cloud Firestore, a cloud-hosted, noSQL database with live synchronization and offline support on Android and iOS.

  - Cloud Firestore is a flexible, scalable database for mobile, web, and server development from Firebase and Google Cloud. Like Firebase Realtime Database, it keeps your data in sync across client apps through realtime listeners and offers offline support for mobile and web so you can build responsive apps that work regardless of network latency or Internet connectivity.

- **Firebase_storage (Package):**
    - Flutter plugin for Firebase Cloud Storage, a powerful, simple, and cost-effective object storage service for Android and iOS.
    - Cloud Storage for Firebase is a powerful, simple, and cost-effective object storage service built for Google scale. The Firebase SDKs for Cloud Storage add Google security to file uploads and downloads for your Firebase apps, regardless of network quality.

- **Shared_preferences (Package):**
    - Flutter plugin for reading and writing simple key-value pairs. Wraps platform-specific persistent storage for simple data (NSUserDefaults on iOS and macOS, SharedPreferences on Android, etc.). Data may be persisted to disk asynchronously, and there is no guarantee that writes will be persisted to disk after returning, so this plugin must not be used for storing critical data.

- **Google_maps_flutter (Package):**
    - A Flutter plugin for integrating Google Maps in iOS and Android applications through API Specific Key.

- **Dio (Package):**
    - A powerful Http client for Dart, which supports Interceptors, FormData, Request Cancellation, File Downloading, Timeout etc.

- **Geolocator (Package):**
    - Geolocation plugin for Flutter. This plugin provides a cross-platform (iOS, Android) API for generic location (GPS etc.) functions.

- **Flutter_polyline_points (Package):**
    - A flutter plugin that decodes encoded google polyline string into list of geo-coordinates suitable for showing route/polyline on maps.
- **Provider (Package):**
    - It is a wrapper around InheritedWidget to make them easier to use and more reusable.
- **Date_picker_timeline (Package):**
    - Flutter Date Picker Library that provides a calendar as a horizontal timeline.
- **Image_picker (Package):**
    - Flutter plugin for selecting images from the Android and iOS image library, and taking new pictures with the camera.
- **Equatable (Package):**
    - A Dart package that helps to implement value based equality without needing to explicitly override == and hashCode.
- **Flutter_staggered_grid_view (Package):**
    - Provides a collection of Flutter grids layouts (staggered, masonry, quilted, woven, etc.).
- **Carousel_slider (Package):**
    - A carousel slider widget, support infinite scroll and custom child widget.

- **Flutter_native_splash (Package):**
    - Customize Flutter's default white native splash screen with background color and splash image. Supports dark mode, full screen, and more.
- **Url_launcher (Package):**
    - Flutter plugin for launching a URL. Supports web, phone, SMS, and email schemes.
- **Flutter_svg (Package):**
    - An SVG rendering and widget library for Flutter, which allows painting and displaying Scalable Vector Graphics 1.1 files.
- **Material_design_icons_flutter (Package):**
    - The Material Design Icons designed by the community for Flutter.
- **Cupertino_icons (Package):**
    - Default icons asset for Cupertino widgets based on Apple styled icons.
- **Flutter_launcher_icons (Package):**
    - A package which simplifies the task of updating your Flutter app's launcher icon.
- **Fluttertoast (Package):**
    - Toast Library for Flutter, easily create toast messages in single line of code.
- **Flutter_lints (Package):**
    - Recommended lints for Flutter apps, packages, and plugins to encourage good coding practices.

3:2 Project functionality:

COVID-19 Detection:

- First, Image Picker Package it takes your uploaded X-RAY Chest Picture for sending it to the Analyzing Model.
- Then, it will be sent to the model with specific coordinates to check if this X-RAY Chest is affected by COVID-19 or not. By Flutter, the application is requesting from the model to work on this X-RAY Chest Picture and compares it with more than 7000+ X-RAY Chest Pictures Positive and negative.
- The Model works on this picture by applying this coordinates as it is trained before to get the best analyzing process to check and provide the prediction.
- This Model that is integrated and merged with the application is working by .tflite extension and then takes the labels form another text file to send the result to the user on the screen.
- It works by Machine Learning and Deep Learning technologies to take the chosen X-RAY Chest Picture and in a many seconds it gets into the training set which is injected by similar X-RAY Chest Picture and return the best prediction for the user with its label from the text file.

Booking Appointments System:

- This Section is working asynchronously between the two MEDICA application (Patient – Doctor).
- It started with patient is viewing all doctors in patient's application and choose what he need according to the department or the disease that he wants to consult about. Then he will find many of doctors that may help him in his consulting case.
- Then, the patient will get a specific doctor and go to his profile, now he will see data about the doctor and it

contains the days that are available from doctor's clinic and also the available times in these days.

- While he clicking on "Book an Appointment" button, all the chosen data will be stored and it will be sent to the firebase server to store it for the doctor and also for the patient and in their collections. To syncing the important screen "My Appointments" in two applications.
- Important Point which is logically must the user take care of this point. The user shouldn't book more than one appointment with the same doctor by the same account until he arranges and go to the first appointment.

Chatting Platform:

- Chatting and Communication Platform for the two MEDICA applications. That is based on the booking appointment system. To enable the communication with the doctors which in the application and the users.
- Chatting Consult is available also for the patients to get connected with the doctor if he need really help just right now. He doesn't need to wait to book an appointment.
- There are a doctors enabling the online consult or chat consult and that is separated part from the appointments.

Collecting patient's Data:

- Gathering all the patient's information in order to make a medical history for the patient that can be referred to during the treatment trip to make appropriate decisions.

Buying Cosmetics and Medicines:

- The User can browse the products which is available in MEDICA application. And can add it to the Shopping Cart to buy it.

Online Payment Methods:

- It's available now for the users to pay their appointment's fees online. Also their orders or their medicines. To avoid wasting time and give the user full functionality from e-healthcare application.

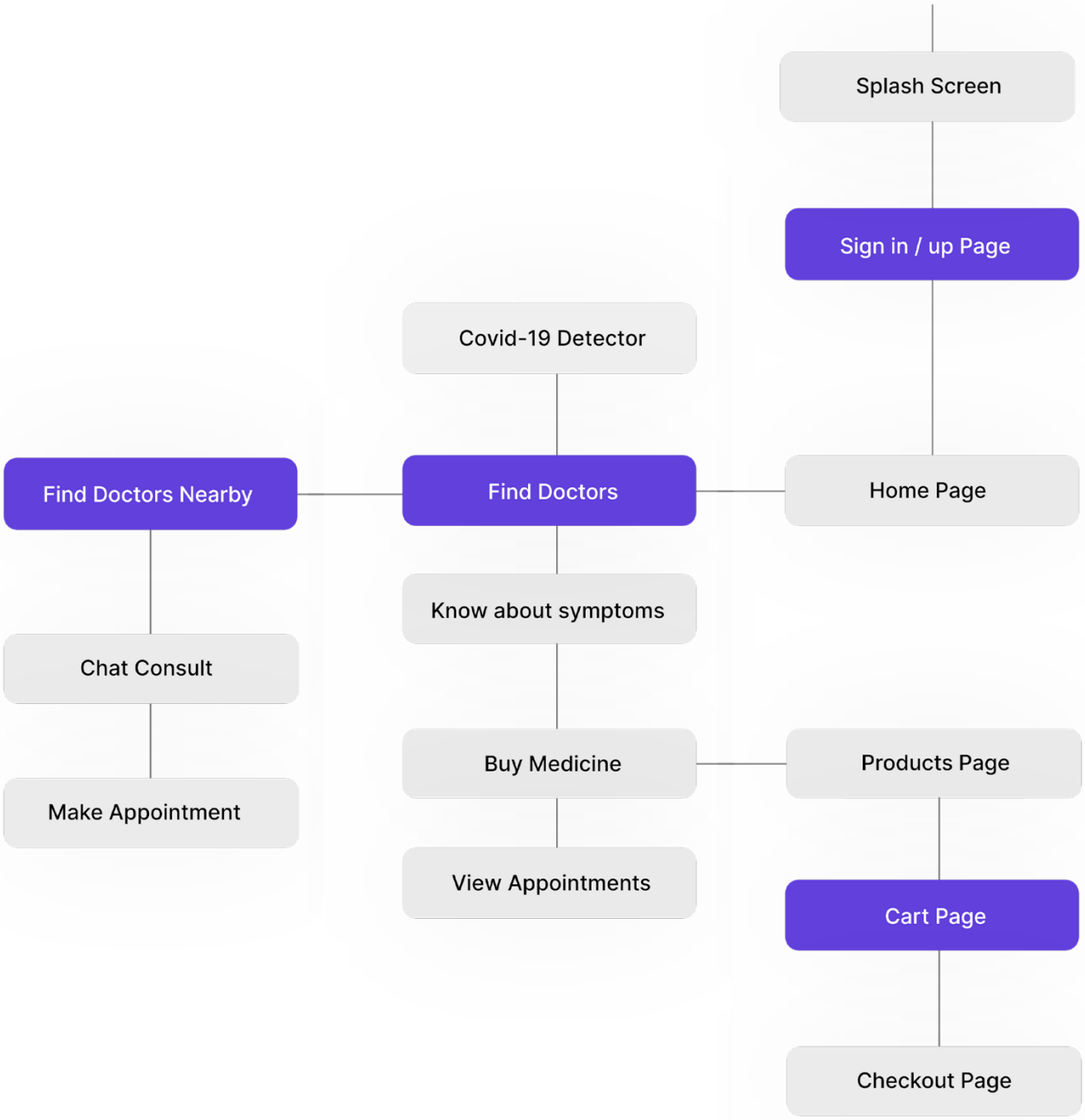
3:3 Functional Requirement:

- Graphic designer for building Professional and advanced UI Design. For making the application screens are easy for using and interacting with it.
- Adobe XD to build this UI Design on it and make it editable and can see it on a Mockup to help in designing.
- Dart Programming Language for developing the Application, Controlling the Front-end screens, Converting the UI Design to UX Design to source code to implement it on Application Screens.
- Firebase Platform for creating the application's database. And enabling the Authentication and Storing the data on Cloud like Credentials, Pictures, Doctor's Data, Patient's Data.
- Developers with experience to interact, control, manage the Back-end and make it related and communicated with Front-end Parts.
- Some other auxiliary programs such as VS CODE for testing and implementing the Source Code.
- Tensorflow to use Machine Learning to train and inject the dataset with the X-RAY Chest Pictures.
- Teachable Machine to take the dataset and convert it to Model is ready to use and analyzing the inputs to predict the true result.

CHAPTER 4

Design of Project

USER JOURNEY

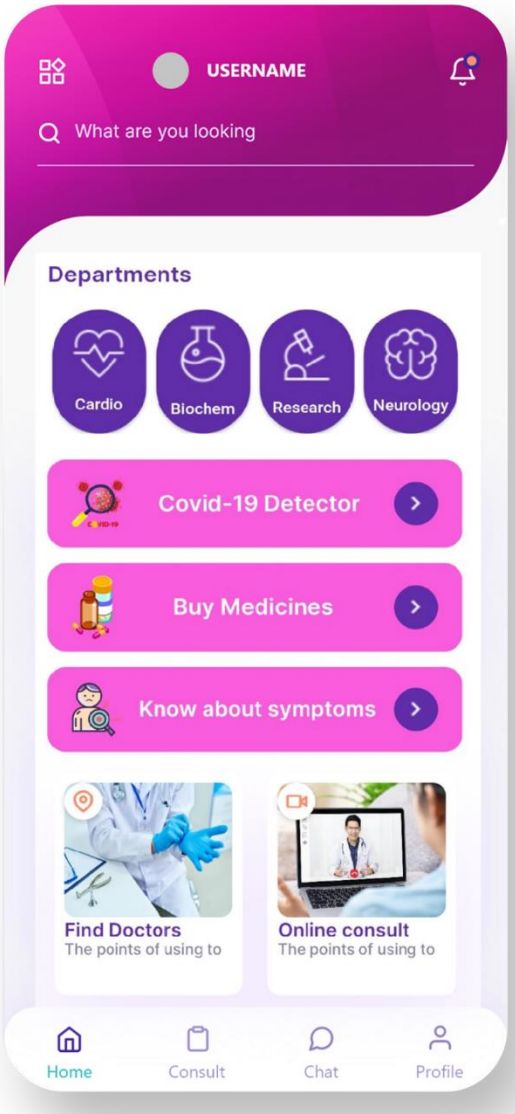


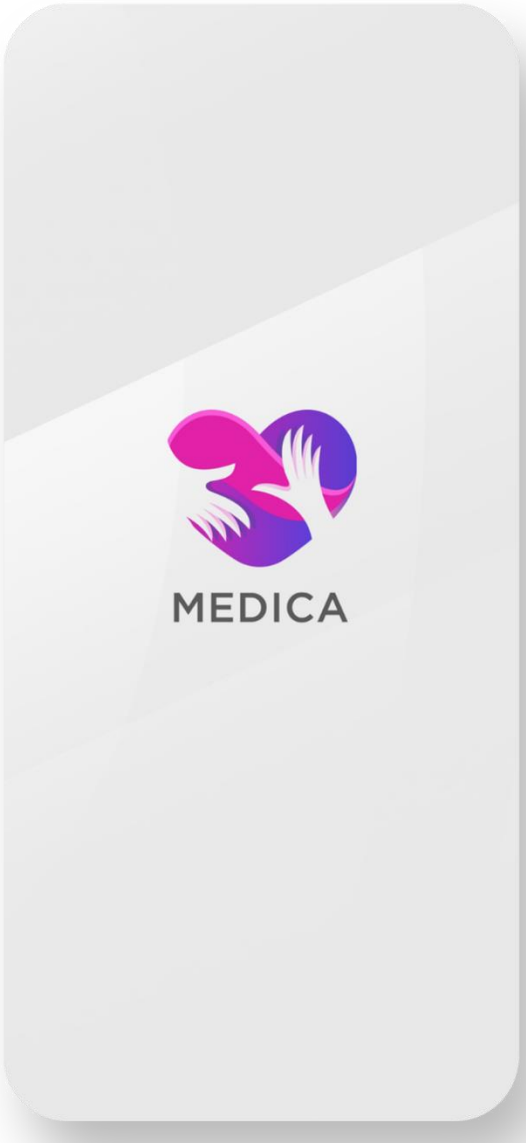
Description

This is the user journey from the begining of the application to making appointments and buying medicines.

HOME SCREEN

Main Features



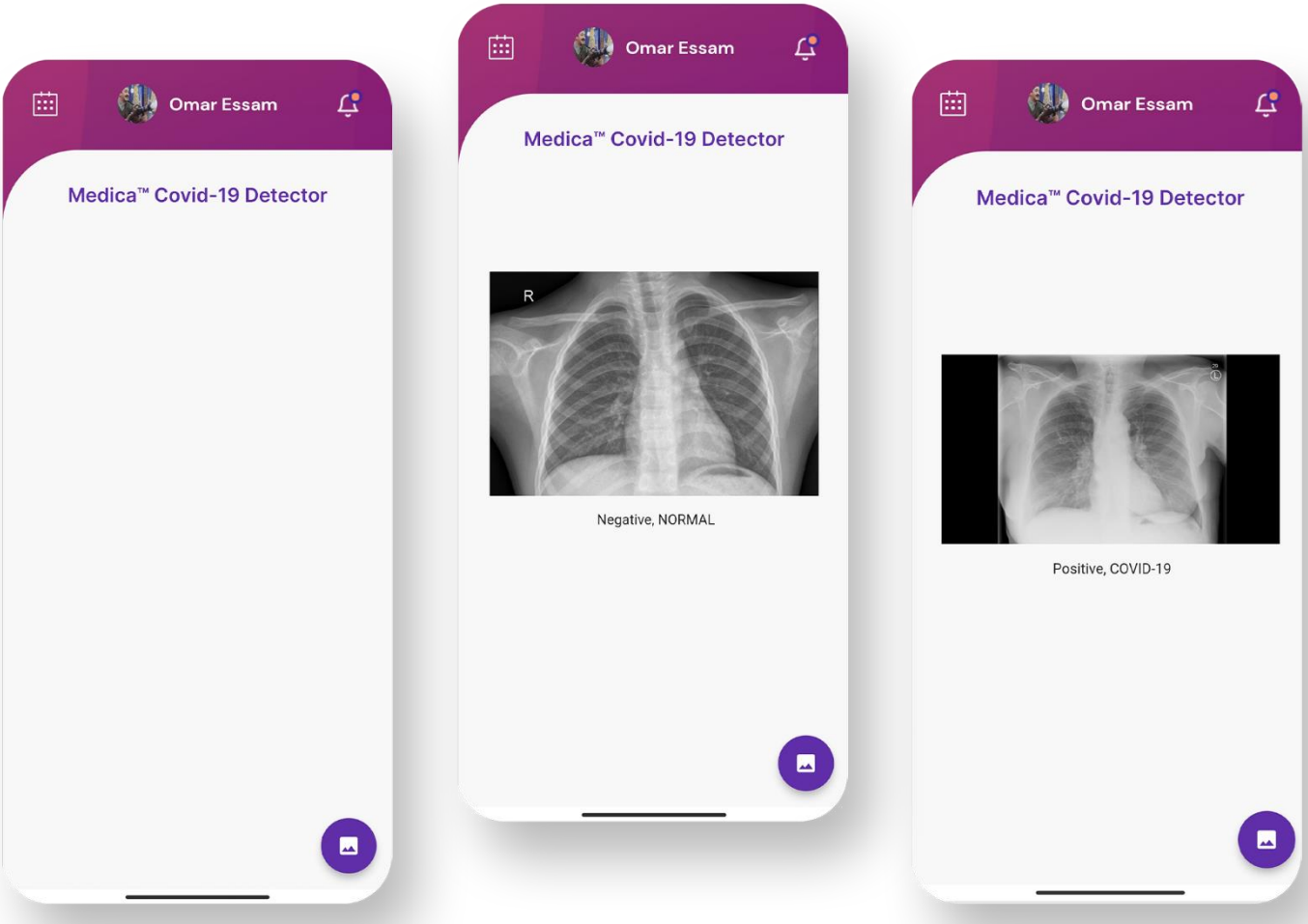


Screen Description

Here you can find a doctor by categories or location,
Buy medicines, Know about symptoms,
Having an online consult and checking for Covid-19 by the Covid-19 Detector.

COVID-19 Detector

Main Features

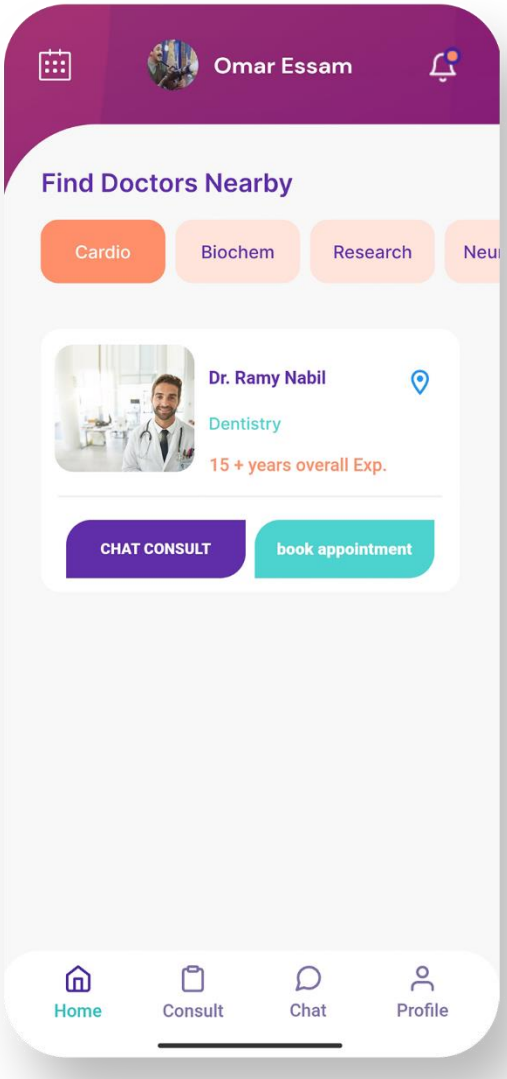


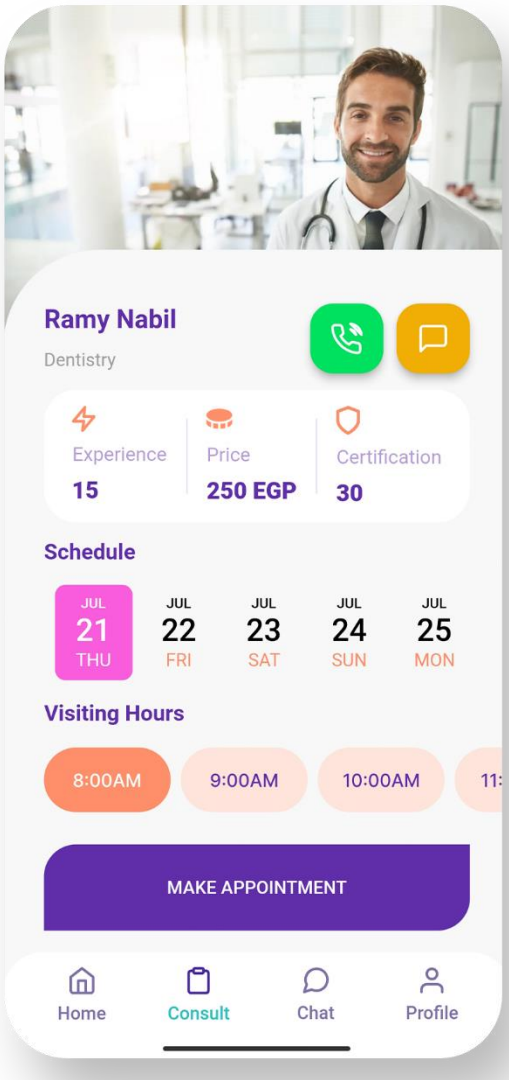
Screen Description

Here you can upload a X-RAY chest picture to let the detector analyze your X-RAY by machine learning, then inform you with the best prediction about being infected by COVID-19.

DOCTORS SCREEN

Main Features



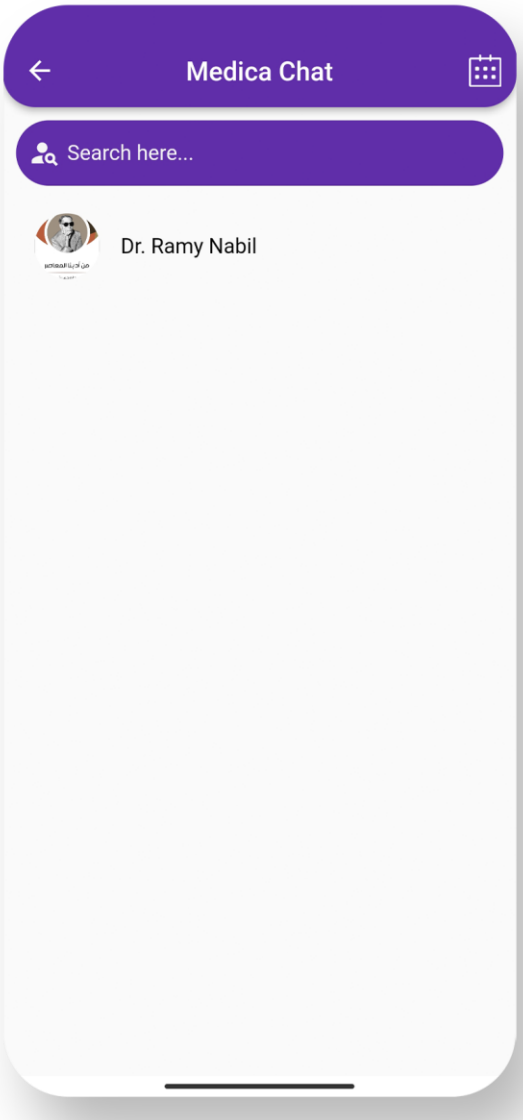


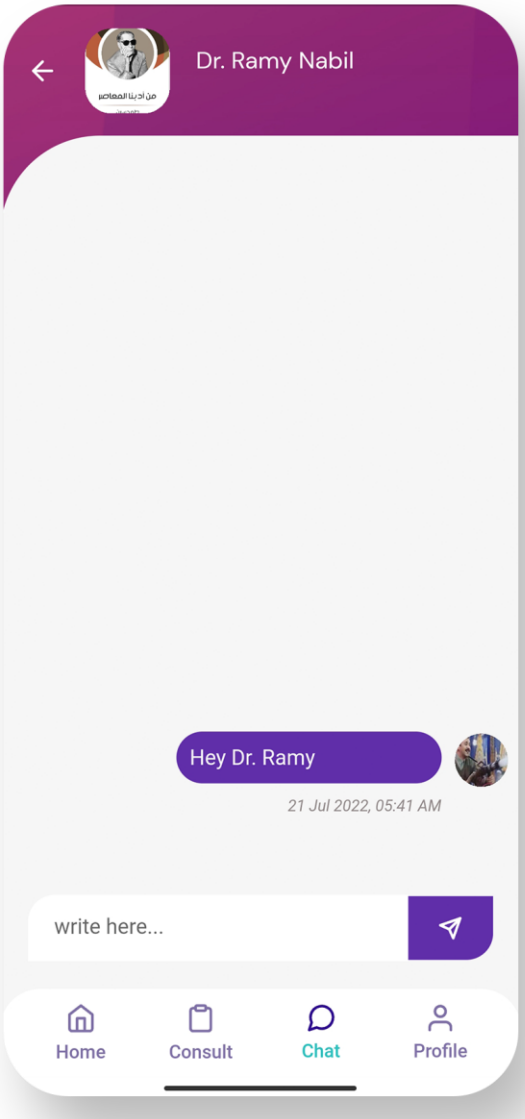
Screen Description

Here you will find a list of doctors registered on the application, You can make appointment with them or consult them.
You can choose the day and time that you want to book.

CHAT SCREEN

Main Features



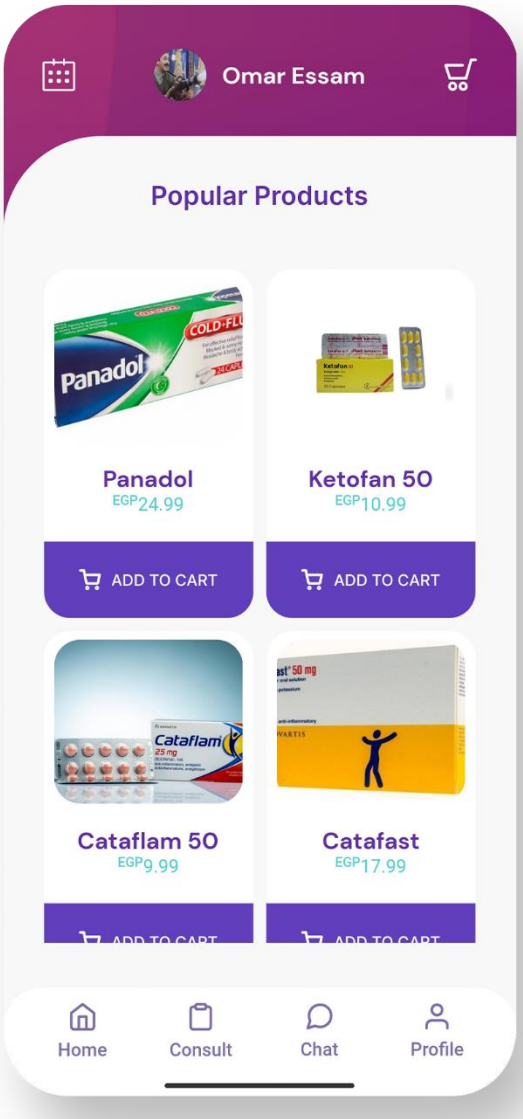


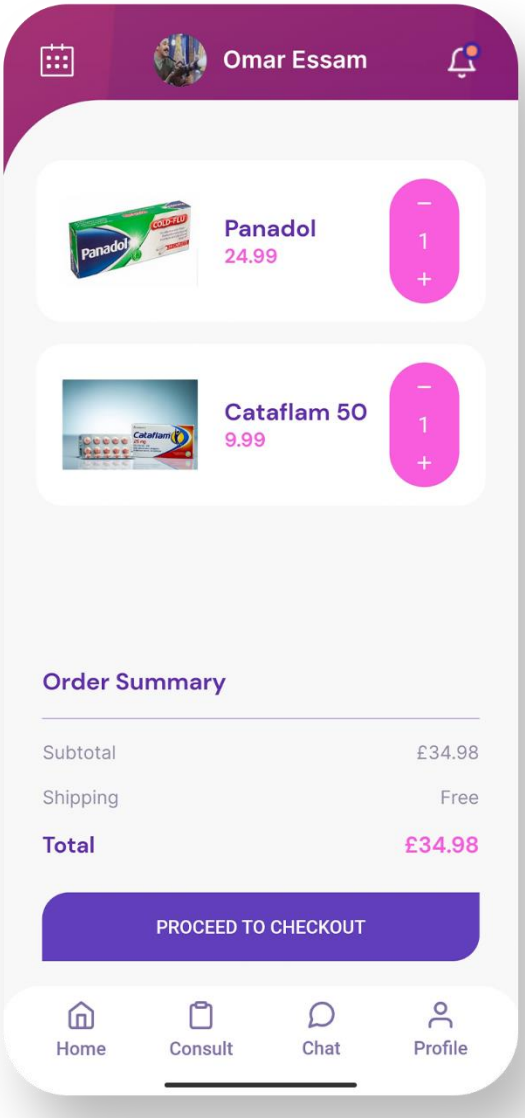
Screen Description

Here you can find the list of doctors that you booked with in advance, and you can communicate with them and consult them.

PRODUCTS SCREEN

Main Features

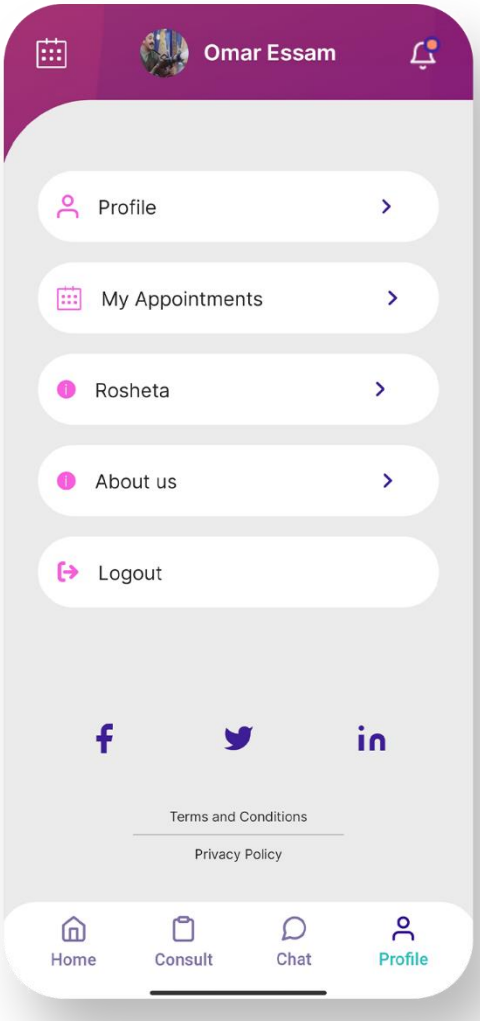


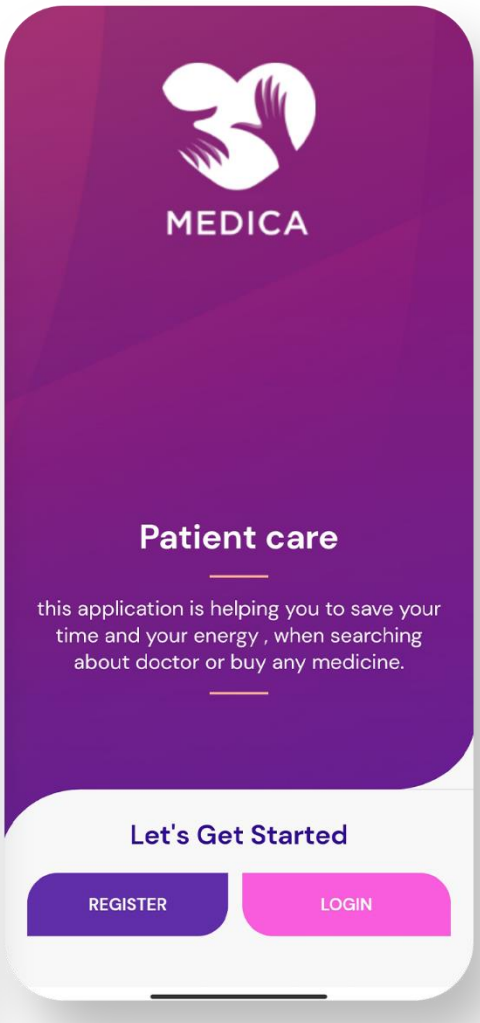


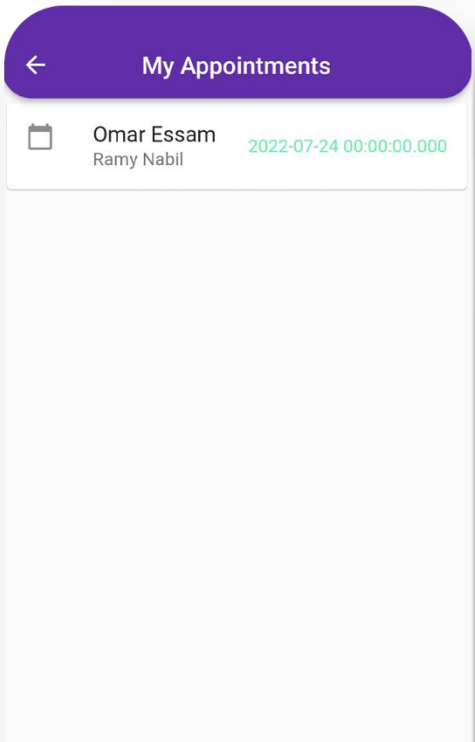
Screen Description

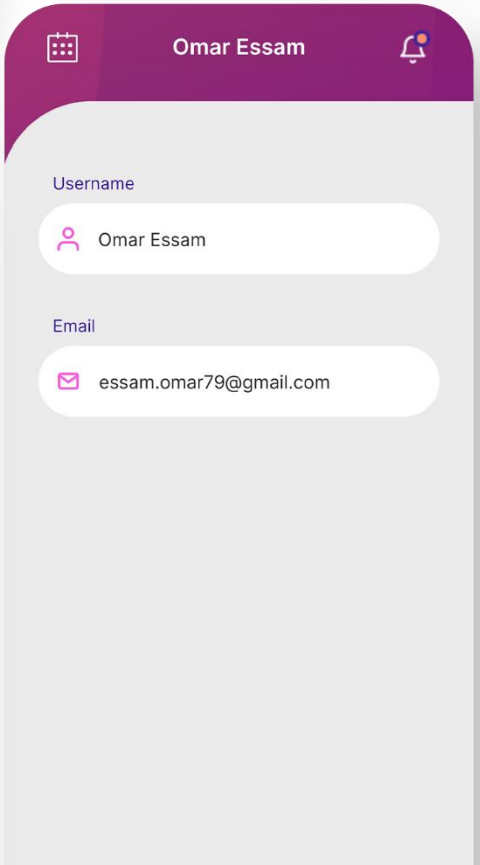
Here you can find a list of medicines that has been added previously on Firebase , you can add any product to your cart and choose the required quantity , then you can proceed to checkout.

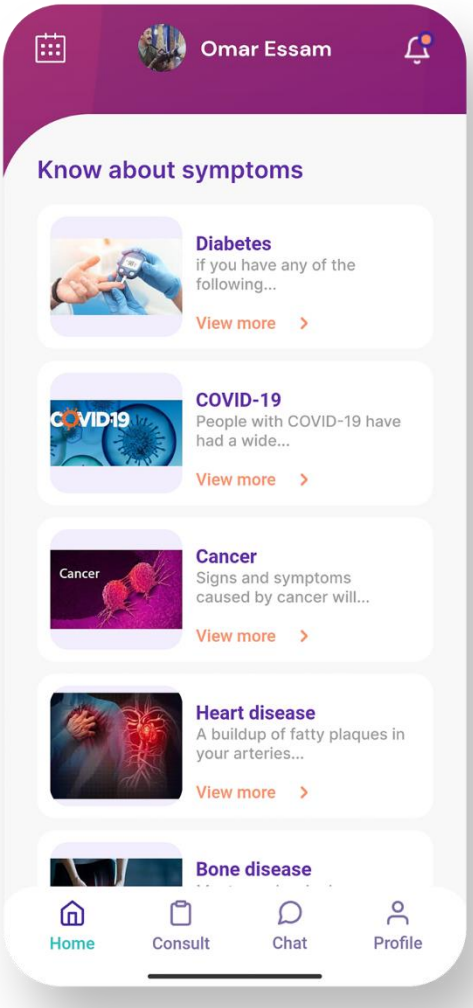
APP SCREENS

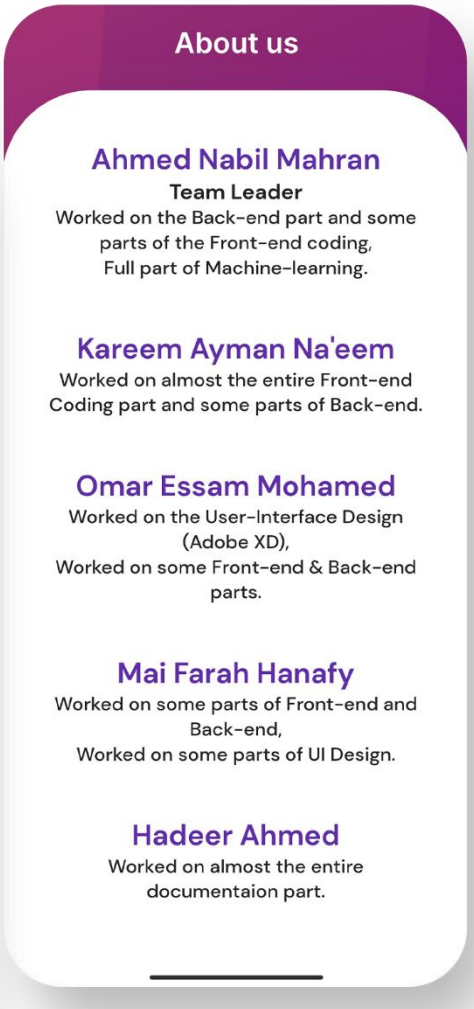

Omar Essam
Profile
My Appointments
Rosheta
About us
Logout
Terms and Conditions
Privacy Policy
Home
Consult
Chat
Profile

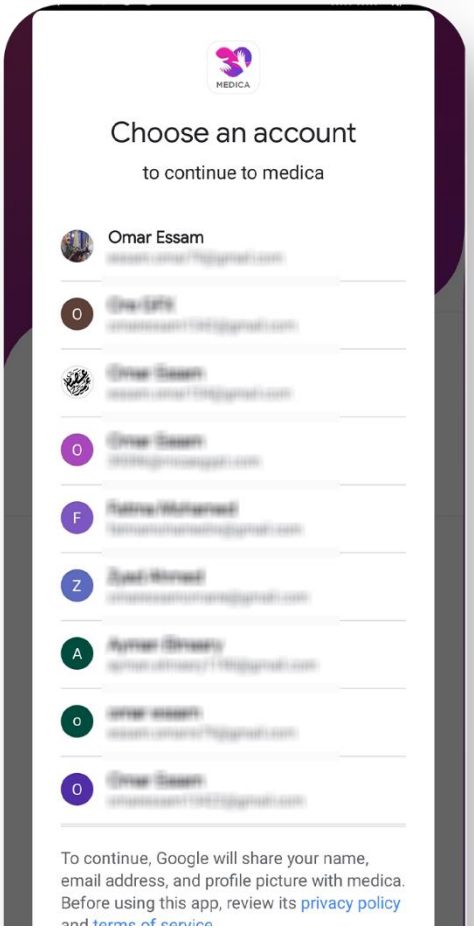

MEDICA
Patient care
this application is helping you to save your time and your energy , when searching about doctor or buy any medicine.
Let's Get Started
REGISTER
LOGIN

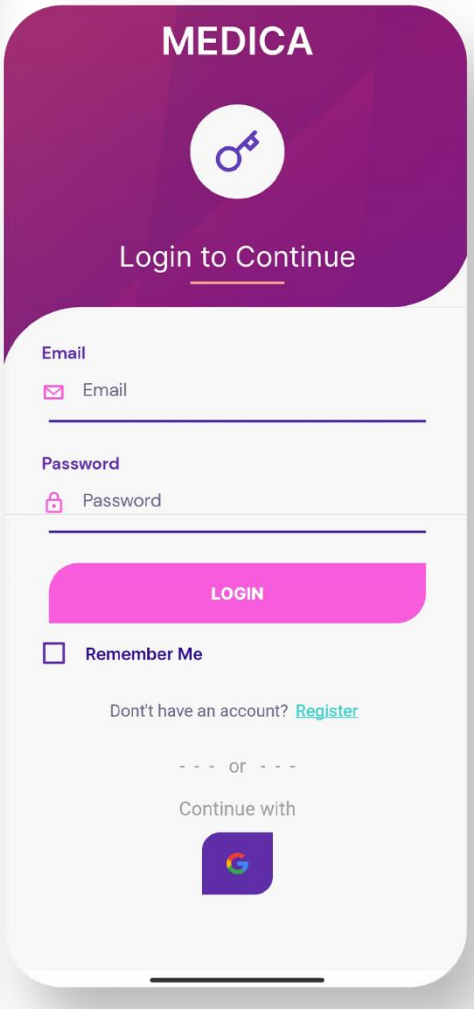

My Appointments
Omar Essam
Ramy Nabil
2022-07-24 00:00:00.000


Omar Essam
Username
Omar Essam
Email
essam.omar79@gmail.com


Omar Essam
Know about symptoms
Diabetes
if you have any of the following...
View more
COVID-19
People with COVID-19 have had a wide...
View more
Cancer
Signs and symptoms caused by cancer will...
View more
Heart disease
A buildup of fatty plaques in your arteries...
View more
Bone disease
Home
Consult
Chat
Profile


About us
Ahmed Nabil Mahran
Team Leader
Worked on the Back-end part and some parts of the Front-end coding,
Full part of Machine-learning.
Kareem Ayman Na'eem
Worked on almost the entire Front-end Coding part and some parts of Back-end.
Omar Essam Mohamed
Worked on the User-Interface Design (Adobe XD),
Worked on some Front-end & Back-end parts.
Mai Farah Hanafy
Worked on some parts of Front-end and Back-end,
Worked on some parts of UI Design.
Hadeer Ahmed
Worked on almost the entire documentaion part.


MEDICA
Choose an account
to continue to medica
Omar Essam
To continue, Google will share your name, email address, and profile picture with medica. Before using this app, review its privacy policy


MEDICA
Login to Continue
Email
Email
Password
Password
LOGIN
Remember Me
Dont't have an account? Register
- - - or - - -
Continue with

CHAPTER 5

Future improvements

Improvements that may be added in the future:

- **Emergency medical service.**
- **Multi language user interface.**
- **Integration with smart wearables.**
- **Health educational messages service through notifications that reach users carrying awareness messages about useful health habits.**

(2021)