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 Mahran
- Kareem Ayman
- Mai Farah
- Omar Essam
- Hader Ahmed

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 Mahran
- Kareem Ayman
- Mai Farah
- Omer Essam
- Hader Ahmed

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

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.

And read about the symptoms and know the common disease 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.

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.

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.
- 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.
- 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.
- Analyze the User's X-RAY Chest Picture Deeply to get the Best Prediction and Helping the User.
- Booking Appointments System with the best Doctors.
- Find Near Doctors.
- View the Doctor's Location on Google Maps to help the patient before booking an appointment.
- In Patient's Application, The User has a Separated Screen to Check his Requested Appointments.
- In Doctor's Application, The Doctor has a Separated Screen to Check his Booked Appointments.
- Chatting Platform to establish the Communication with the Patient and his Doctor.
- Buying Medicines Department.
- Provide Fawry Pay Method for Payment while buying medicines or booking appointments.
- Getting Information about Symptoms and Diseases.
- Separated apps (Doctor – Patient).
- Appointment Reminder and Notification.

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).

- **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. In this respect, 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.

- **Github:**

  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.

- **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.

- **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.

- **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.

- **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:

- Machine Learning analyzer to get the best prediction on the X-RAY Chest Picture. To tell the user if there is a COVID-19 or not.

Booking Appointments System:

- This section is for booking and following up on appointments available to the doctor.

Chatting Platform:

- To communicate with the patient to follow up on his health.

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. Xd

- **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**.

CHAPTER 4

Design of Project

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)

(2021)

(2021)

(2021)