**Chapter One**

**Introduction and Purpose**

# Introduction

This Chapter is divided into six Sections: - Section 1.1 represent the introduction of the Chapter, Section 1.2 list of basic definition, Section 1.3 over view about project, Section 1.4 show the problem definition, Section 1.5 represent the project objectives and finally Section 1.6 concludes this chapter.

# Basic definitions:

**Tools:**

**Flutter:** is Google’s UI toolkit for building beautiful natively compiled applications for mobile, web, and desktop.

**Dart**: is the programming language used to code Flutter apps it’s another product by Google and released version2.1, it’s a client-optimized language for fast apps on any platform.

**Android:** is a mobile operating system based on a modified version of the Linux Kernel and other open source software designed primarily for touchscreen mobile devices such as smartphone and tablets.

**IOS:** is a mobile operating system created and developed by Apple Inc. exclusively for its hardware. This operating system powers many of the company’s mobile devices including IPhone, IPod Touch and the IPad.

**API:** is a tool that acts as a gateway that sits between a client and a collection of backend services, it also acts as a reverse proxy to accept all application programming interface calls, aggregate the various services.

**Swift:** is a general-purpose, multi-paradigm, compiled programming language developed by Apple Inc. for IOS. It is designed to work with Apple’s Cocoa and Cocoa Touch frameworks.

**Kotlin:** is a general-purpose, free, open source, statically typed “pragmatic” programming language initially designed for the Java Virtual Machine and Android.

**MySQL**: is an [open-source](https://en.wikipedia.org/wiki/Open-source_software) [relational database management system](https://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS). A [relational database](https://en.wikipedia.org/wiki/Relational_database) organizes data into one or more data tables in which data types may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database.

**Laravel** : is a free, open-source PHP web framework, created by Taylor Otwell and intended for the development of web applications following the model–view–controller (MVC) architectural pattern.

**PHP:** is a **general-purpose scripting language that is especially suited to web development**. It was originally created by Danish-Canadian programmer Rasmus Lerdorf in 1994; the PHP reference implementation is now produced by The PHP Group.

**Bootstrap**: In general, bootstrapping usually refers to a self-starting process that is supposed to proceed without external input. In computer technology the term (usually shortened to booting) usually refers to the process of loading the basic software into the memory of a computer after power-on or general reset, especially the operating system which will then take care of loading other software as needed.

**HTML:** Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript

**CSS:** is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript

# Overview:

Who we are:

We are the first Arabic web site in Egypt to serve three categories in the community: Patient, Donor and Hospital.

What we do:

We provide people with the health care they need and as fast as possible by reducing the time and effort of paperwork and the ambiguity of the medical prescriptions, we also saving lives by enabling ambulances to find patients easily.

# Problem definition:

The problem which faces all of us is to interact with our doctor easily, calling the nearest ambulance, and having our medical history without taking lots of papers, analysis or reports, also we a lot of us cant calculate the insurance ratio by itself to get the cheapest, nearest and the most qualified medical service.

The problem which faces the doctor with his patients is checking the medical history and changing the drugs already taken due to the new case and write a new prescription and record it in an app without making him check lots of papers and reports.

The problem which faces the ambulance is that the patient’s location wasn’t so much clear, so it was forced to call the patient or patient’s companion many times and be late on the patient and of course it is risk on the patient’s life.

The problem which forces the hospital is if any patient forgets to get his insurance card, they can’t calculate the insurance ratio, so it costs the patients so much money

# Project objectives:

1. Allow the patient to connect with the doctor easily.
2. Allow the patient to record all his medical reports without returning back to a lot of papers.
3. Allow the patient to calculate his insurance ratio.
4. Allow the doctor to check the medical history or reports of the patient and record the new reports and the new drug prescriptions.
5. Allow the ambulance to check the patient’s location and arrive faster.
6. Allow the hospital to calculate the insurance ratio anytime the patient went to a hospital even if he forgot his cards.

# References

1. [www.flutter.dev](http://www.flutter.dev)
2. <https://dart.dev>
3. [www.redhat.com](http://www.redhat.com)
4. https://en.m.wikipedia.org/
5. [www.infoworld.com](http://www.infoworld.com)

# 1.7 Conclusion

In this chapter we present an introduction, basic definitions problem definition and project objectives of our project and references that had been used in that chapter

# 