|  |  |  |  |
| --- | --- | --- | --- |
| **المملكة العربية السعودية**  **وزارة التعليم العالي**  **جامعة الإمام محمد بن سعود الإسلامية**  **كلية علوم الحاسب والمعلومات** | | **Second term 1441/2020** | **KINGDOM OF SAUDI ARABIA**  **Ministry of Higher Education**  **Al-Imam Mohammad University**  **College of Computer & Information Sciences** |
| **Software Engineering (CS-310)**  **BSCS- Section: 171**  **Project-Phase No: 2**  **Diagnosing Diseases App (Bot)**  **(Design)**  **Submitted By**  **Sultan Mohammed Alzrair (438011441) – Coordinator**  **Anas Wlead Bajunayd (435021105)**  **Abdullah Saad Alswailem (438011158)**  **Aref Mansoor Alotaibi(439027005)**  **Rayan Saud Alshanabah(438012286)**  **Supervisor**  ***Dr.Sultan Saud Alqahtani***  **Date: 021/03/2020** | | | |
|  | | | |

**Revision History**

|  |  |  |
| --- | --- | --- |
| **Date** | **Version** | **Description** |
| 19/03/2020 | Phase.2 | Design |
|  |  |  |
|  |  |  |
|  |  |  |

Table of Contents

[**1.0 Abstract** 4](#_Toc35946775)

[**2.0 Introduction** 5](#_Toc35946776)

[**3.0 High Level and Medium Level Design** 6](#_Toc35946777)

[3.1 System Interaction Model 6](#_Toc35946778)

[3.2 Model-View-Controller 7](#_Toc35946779)

[3.3 Class Diagram 8](#_Toc35946780)

[3.4 Class Method Description 9](#_Toc35946781)

[3.4 Detailed Design 11](#_Toc35946782)

[**4.0 User Interface Design** 12](#_Toc35946783)

[**5.0 Contributions of Teams Members** 16](#_Toc35946784)

[**6.0 Conclusion** 17](#_Toc35946785)

# **1.0 Abstract**

This app will be able determine disease by a bot based on user answers, the questions are related from previous answers of the user during the diagnose session, it will keep asking questions and accessing the database until the bot can determine the disease and it will show it to the user after that.

# **2.0 Introduction**

Recently we have seen an increase adoption of smartphone-based diagnostic tools and one of the most promising directions in medicine is finding new and improving old methods of medical diagnosis, There are many different ways to diagnose and treat diseases, Diagnosis Diseases App designed to help both doctors and patients, Patient can be diagnosed by asking them necessary questions, based in his answers the bot will reducing the other symptoms until determine the disease, By using this method the bot be able to analysing the patient answers,

Before all of you need to know about the database, we have that consists all of the disease name and cures, From aaccessing the database based on the patient answers we can get the name and cure of the disease and show it to the user.

# A close up of a logo Description automatically generated**3.0 High Level and Medium Level Design**

## 3.1 System Interaction Model

As we can see the figure 1 below the use case describe the Diagnose Disease app the User and the System and how can they interact with each other.

Figure 1- Use Case Diagram

## 3.2 Model-View-Controller

A close up of a logo

Description automatically generatedAs we can here figure 2 blow Model-View-Controller the interaction from system data, and how they interact with each other.

Figure 2 - MVC

## 3.3 Class Diagram

We can see Figure 3 below the class diagram is showing the structure of the app.

A close up of text on a white background

Description automatically generated

Figure 3 – Class Diagram

## 3.4 Class Method Description

Class DiagnoseController

Method diagnose

Visibility public

Return type JsonResponse

Parameters, type request: Request

Description this function will handle the whole operation

Class Disease

Method getDrugByDiseaseId

Visibility public

Return type collection

Parameters, type id: int

Description get drug by disease id

Class Disease

Method getDiseaseById

Visibility public

Return type collection

Parameters, type id: int

Description get disease by id

Class Disease

Method getDiseaseBySymptoms

Visibility public

Return type collection

Parameters, type symptoms: int[]

Description get disease by symptoms id's

Class Symptom

Method getMostCommonSymptoms

Visibility public

Return type collection

Parameters, type void

Description get most common symptoms

Class Symptom

Method getMoreSymptomsInCommon

Visibility public

Return type collection

Parameters, type Symptoms: int[]

Description get more symptoms that have inputed symptoms in common

Class Symptom

Method getDiseaseBySymptomId

Visibility public

Return type collection

Parameters, type id: int

Description get disease by symptom id

Class Drug

Method getDiseasesByDrugId

Visibility public

Return type collection

Parameters, type id: int

Description get diseases by drug id

Class Drug

Method getDrugById

Visibility public

Return type collection

Parameters, type id: int

Description get drug by id

## 3.4 Detailed Design

Here we can see Figure 4 how the client gets the results in the sequence diagram.

A screenshot of a cell phone

Description automatically generated

Figure 4 – sequence diagram

# **4.0 User Interface Design**

A close up of a logo

Description automatically generated

Figure 5 – Here we can the user interface consist the logo.

A screenshot of a cell phone

Description automatically generated

.

Figure 6 – Here the user should agree the terms before using the app.

A close up of a logo

Description automatically generated

Figure 7 – Now the user needs to enter his info gender, age, etc.

A picture containing drawing

Description automatically generated

Figure 8 – Here the user should choose what symptom he has to start diagnosing.

A screenshot of a cell phone

Description automatically generated

Figure 9 – As we can see here the Diagnose session is started and the bot start asking and user should answer Yes/No.

A screenshot of a cell phone

Description automatically generated

Figure 10 – And now the user will be able to use the results of his answers names of the disease and treatment and the nearest hospitals.

# **5.0 Contributions of Teams Members**

Sultan Mohammed Alzrair

* + Introductions
  + Use Case Diagram
  + User interface Design
  + Class Diagram

Anas Wlead Bajunayd

* + - Use Case Diagram
    - Class Diagram
    - Introductions
    - User interface Design

Abdullah Saad Alswailem

* + - User interface Design
    - Introductions
    - Use Case
    - Architecture Pattern

Aref Mansoor Alotaibi

* + - Use Case Diagram
    - User interface Design
    - Class Methods
    - Architecture Pattern

Rayan Saud Alshanabah

* + - Architecture Pattern
    - Use Case Diagram
    - User interface Design
    - Introductions

# 

# **6.0 Conclusion**

In this design document we have mentioned many design topics. first, we show the use case diagram and the MCV pattern and we described them and then we did the class diagram with the methods and we fallow it with a sequence diagram descripting how the system interacting with the client to get the results. And also, we showed the interface of the app and how its look like. And the last we provided a description with each member contributions in this phase.