**MAKEREREUNIVERSITY**

SCHOOL OF COMPUTING AND INFORMATION TECHNOLOGY

BACHELOR OF SCIENCE IN SOFTWARE ENGINEERING

GROUP: BSE20-36

AN INTELLIGENT MEDICAL DIAGNOSTIC TOOL

GROUP MEMBERS

|  |  |  |
| --- | --- | --- |
| **Name** | **Registration Number** | **Email** |
| KEKIRUNGA JEAN | 16/U/5850/PS | jeankekirunga@gmail.com |
| MUGISA BEST | 16/U/7279/PS | bubulicious0@gmail.com |
| NABUUFU ERETH | 16/U/8227/EVE | luckybrave12@gmail.com |
| KISIGA TIMOTHY | 16/U/6173/PS | ktimo550@gmail.com |

**SUPERVISOR**

Dr. STEPHEN ODONGO

Department of Networks

School of Computing and Information Technology

**GROUP: BSE20-36**

**An Intelligent Medical Diagnostic Tool**

**Software Design Document**

**Names:** Kekirunga Jean, Kisiga Timothy, Mugisa Best, Nabuufu Ereth

**Lab Section:**

**Workstation:**

**Date:** 12th February, 2020

**Table of Contents**

[1. Introduction 2](#_Toc34058188)

[1.1. Purpose 2](#_Toc34058189)

[1.2. Scope 2](#_Toc34058190)

[1.3. Overview 3](#_Toc34058191)

[1.4. Reference Material 3](#_Toc34058192)

[1.5. Definitions and Acronyms 4](#_Toc34058193)

[2. System Overview 5](#_Toc34058194)

[3. System Architecture 6](#_Toc34058195)

[3.1. Architectural Design 6](#_Toc34058196)

[3.2. Decomposition Description 7](#_Toc34058197)

[3.2.1. Class diagram 7](#_Toc34058198)

[3.2.2. Sequence Diagram 8](#_Toc34058199)

[3.3. Design Rationale 9](#_Toc34058200)

[4. Data Design 9](#_Toc34058201)

[4.1. Data Description 9](#_Toc34058202)

[4.2. Data Dictionary 10](#_Toc34058203)

[5. Component Design 12](#_Toc34058204)

[6. Human Interface Design 14](#_Toc34058205)

[6.1. Overview of User Interfaces 14](#_Toc34058206)

[6.2. Screen Images 14](#_Toc34058207)

[6.3. Screen Objects and Actions 17](#_Toc34058208)

[7. Requirements Matrix 18](#_Toc34058209)

**List of Figures**

Figure 1 The architectural design of the Intelligent Medical Diagnosis Tool**…………………….**4

Figure 2 The class diagram of the Intelligent Medical Diagnosis Tool**……………………………**6

Figure 3.1 The sequence diagram for making Prognosis**………………………………………….**7

Figure 3.2 The sequence diagram for making a diagnosis**…………………………………………**7

Figure 6.1 Figure showing the screen for Registering New Patient**……………………………...**13

Figure 6.2 A figure showing the screen for Making Prognosis**………………………………….**14

Figure 6.3 Figure showing the screen that will allow entry of Laboratory Results**……………...**15

# Introduction

## **Purpose**

The purpose of this software design document is to present a description of the architecture, preliminary and detailed system design of the Intelligent Medical Diagnosis Tool. The document will be reference to the coding phase.

This document is intended for the project supervisors, the system developers of the Intelligent Medical Diagnostic Tool and any researchers interested in a similar field of study.

## **Scope**

The system to be built is a web based intelligent diagnostic tool, whose operations are based on AI and deep learning as an AI technology. The main purpose of this system is to help doctors and other medical practitioners recognize diseases in people with minimal human interaction.

**Objectives**

* To collect ground-truth data that can be used to map diseases to symptoms.
* To build an intelligent disease recognition system.
* To test and validate the use of both Artificial Neural Networks (ANN) and Convolutional Neural Network (CNN) as a hybrid method for Feature selection and Feature extraction in machine learning.

**Benefits**

* This system will provide a timely reference to medical practitioners while performing patient diagnosis.
* The system will provide differential patient diagnosis to medical practitioners.
* The system will provide recommended treatment for predicted diseases.
* The system will be a localized information store for the different diseases in Uganda.

## **Overview**

This document is written according to the standards for Software Design Documentation explained in “IEEE Recommended Practice for Software Design Documentation” [1]. This SDD contains eight (8) sections which include:

* Section 1 contains the introduction which includes the purpose of the software design document, overview of the document, references, definitions and acronyms, and the scope.
* Section 2 contains a general overview of the system’s functionalities.
* Section 3 contains the architecture of the system.
* Section 4 describes the structures of the data and its properties.
* Section 5 provides the details of each component of the system
* Section 6 contains human interface designs which provide samples of the user interface from the system.
* Section 7 contains a requirements matrix.
* Section 8 contains the appendices that provide more information on the terms used in the document

## **Reference Material**

* Engineering, S., & Committee, S. (2005). IEEE Draft Standard for Software Design Descriptions, (December).
  1. **Definitions and Acronyms**

|  |  |
| --- | --- |
| **Acronym** | **Meaning** |
| AI | Artificial Intelligence |
| ANN | Artificial Neural Network |
| CNN | Convolutional Neural Network |
| SDD | Software Design Document |
| Deep-Learning | It is a subset of machine learning in AI that has networks capable of learning unsupervised from data that is unstructured or un labelled. |
| Diagnosis | This is a process of determining which disease or condition explains a person’s symptoms and signs |
| IEEE | Institution of Electrical and Electronics Engineers |

# System Overview

The challenge of insufficient medical doctors in health facilities in Uganda has increasingly become a problem over the years. This problem has birthed many other challenges that have made it almost impossible for Ugandans to access quality and prompt medical services.

In order to solve this problem, we are have proposed an Intelligent Medical Diagnostic tool that will help medical practitioners to recognize diseases with minimal human interaction. This system shall allow login to only authorized medical practitioners. The medical practitioner will then be able to register a patient via a form that will require the details of the patient e.g. Name, Age, etc. The medical practitioner then will then enter the patient condition and symptoms. The system shall make a prognosis of what the patient may be suffering from in reference to the latter’s signs and symptoms, and recommend the appropriate laboratory tests to be done. The system will also allow entry of the laboratory results to compare with the prognosis.

# System Architecture

## **Architectural Design**

The architecture of the intelligent medical diagnosis of the intelligent medical diagnosis tool shall be a 3-tier client server architecture. The 3 tiers shall be the presentation logic layer, the application logic layer and the data layer.

THE ARCHITECTURAL DESIGN OF THE INTELLIGENT MEDICAL DIAGNOSIS TOOL



Figure 1 The architectural design of the Intelligent Medical Diagnosis Tool

The presentation layer shall house the front-end components of the tool. These shall include the web application user interfaces; this layer is to be used directly by the users. It shall communicate to other tiers through a REST API that will display the predictions of what patients might be suffering from. This layer shall be developed using Bootstrap.

The application tier contains the functional business logic, which drives the application’s fundamental roles e.g. patient registration, capturing patient details, carrying out computations, perform prognosis etc. the application will be based on the Django rest framework of python.

The Data tier shall act as storage for all the useful data generated by the application while in deployment. The data tier will be built using Django rest framework for building web applications.

## **Decomposition Description**

### Class diagram

THE CLASS DIAGRAM OF THE INTELLIGENT MEDICAL DIAGNOSIS TOOL



Figure 2 The class diagram of the Intelligent Medical Diagnosis Tool

### **Sequence Diagram**

SEQUENCE DIAGRAM FOR MAKING A PROGNOSIS



Figure 3.1 The sequence diagram for making Prognosis

SEQUENCE DIAGRAM FOR MAKING A DIAGNOSIS



Figure 3.2 The sequence diagram for making a diagnosis

## **Design Rationale**

The 3-tier architecture style was adopted as the architecture of the Intelligent Medical Diagnosis tool. However, the underlying design pattern to be used is the model, view, controller (MVC) architecture because it supports rapid and parallel development. Below are the reasons to justify the choice.

* The architecture supports the ability for each tier to evolve and change independently, this adds robustness to the system. It encourages high cohesion and low coupling between the tiers.
* Scalability since it allows the distribution of the application components across multiple Servers.
* All the tiers can be developed independent of one another in terms of technology, platform implementation and language meaning any tier can be modified, maintained without impacting the other tiers. This speeds up development since different developers work on different tiers.
* Flexibility and portability is another advantage in that the system can be usable for a variety of user interfaces and devices.
* Suitable for handling large volume of complex and varied operations since the presentation tier doesn’t interact with the data access layer, the data doesn’t need to handle multiple operations at the same time.

# 4. Data Design

## **4.1. Data Description**

The medical diagnostic tool will run on SQL lite that will store information about patients and medical practitioners. This database will also store prognoses and real lab results.

The medical diagnostic tool database a will have entries like Doctor, Patient, Prognoses and Lab results which will be represented as tables in the database.

This kind of information will be used in the event that a doctor needs to login into the medical diagnostic tool to register a new patient, or to retrieve patient information.

## **4.2. Data Dictionary**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Fields** | **Type** | **Null** | **Description** |
| **Medical practitioner** | MedId | VARCHAR [5] | No | Unique identifier for each medical practitioner |
| Username | VARCHAR [30] | No | Name used by each person to access the system |
| Password | VARCHAR [30] | No | Authentication |
|  | | | | |
| **Differential diagnosis** | Signs | VARCHAR [20] | yes | Signs available for the different diseases. |
| Symptoms | VARCHAR [20] | yes | Signs available for the different diseases. |
| Age | INT [5] | No | This will store the age groups that will be covered by the system. |
| prognosis | VARCHAR [20] | yes | This is the predicted possible diseases. |
|  | | | | |
| **Patient** | Age | INT [2] | No | Age provided by the patient when he is being registered by a medical practitioner. |
| Signs | VARCHAR [30] | yes | These are the signs of the patient. |
| Symptoms | VARCHAR [30] | yes | These are the symptoms of the patient. |
| Name | VARCHAR [30] | yes | This is the name provided by the patient when he is being registered by a medical practioner. |
| PatientId | VARCHAR [20] | No | This is a unique id given to each patient on registration. |

# Component Design

**Login**

*READ Username, Password*

*IF (Username == EnteredUsername && Password == EnteredPassword)*

*Login Successful ELSE*

*Login Failed. ENDIF*

**Register Patient**

*READ details*

*IF details!= NULL*

*User registered successfully Else*

*Error “fill in all details”*

*END IF*

**Perform differential diagnosis**

*WRITE patient symptoms and signs*

*READ symptoms and signs*

*IF(majority of signs & symptoms match those of diseases in the database)*

*RETURN the probable diseases to medical practitioner*

*END*

*ELSE*

*Exception “Signs & symptoms not known”*

*ENDIF*

*END*

**Recommend Tests**

*IF (disease ==probable disease)*

*Suggest recommended tests for this disease*

**Recommend treatment and management**

*IF (disease==probable disease)*

*Show treatment and management*

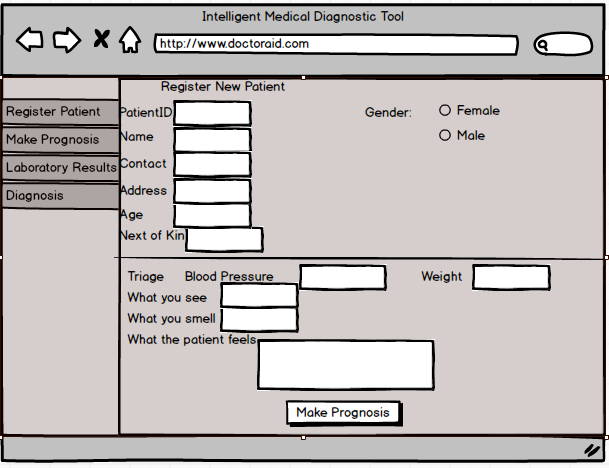
# Human Interface Design

## **Overview of User Interfaces**

This system shall allow login to only authorized medical practitioners. The medical practitioner will then be able to register a patient via a form that will capture the patient’s bio data. The medical practitioner then enters the patient’s presented signs and symptoms. A prognosis shall then be performed by the system basing on the signs and symptoms presented and recommend the appropriate laboratory tests to be done. The system will also allow entry of the laboratory results to compare with the prognosis. The system shall recommend treatment and management for all the differential diagnosis

## **Screen Images**

THE REGISTER PATIENT SCREEN

Figure 6.1 Figure showing the screen for Registering New Patient

The system shall make a prognosis from the information availed by the patient.

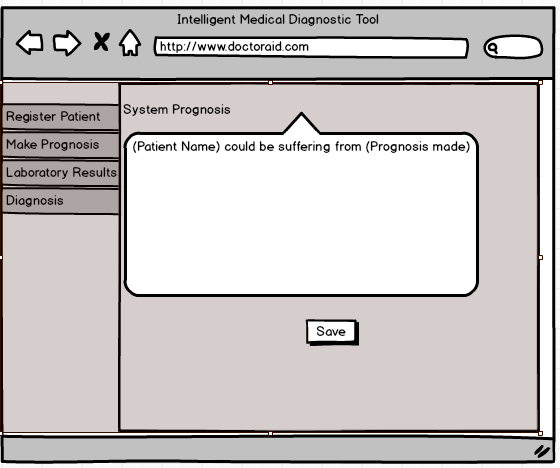


Figure 6.2 A figure showing the screen for Making Prognosis

The system shall allow entry of laboratory results carried out to confirm the presence or absence of the disease predicted.

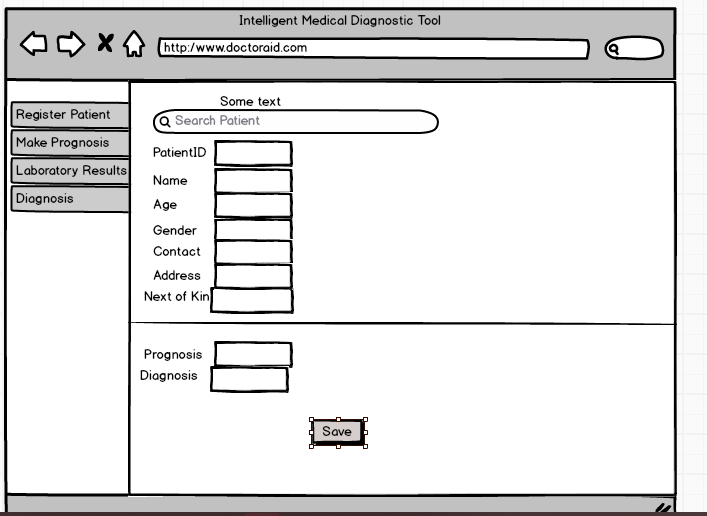
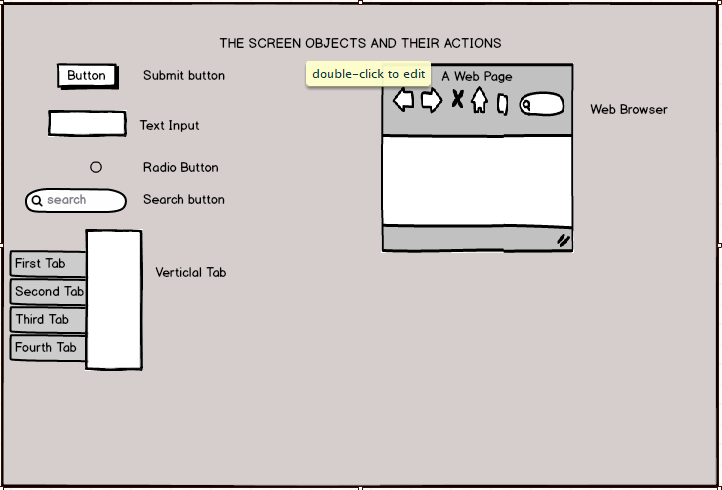


Figure 6.3 Figure showing the screen that will allow entry of Laboratory Results

## **Screen Objects and Actions**



# Requirements Matrix

FR-01: Perform Different diagnosis

FR-02: Recommend tests

FR-03: Recommend treatment and management

FR-04: Register Patients

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Subsystems** | | |  |  |
| **Functional Requirements** | Web application | Prognosis | Tests recommendation | Database | System management |
| FR-01 | X | X |  | X |  |
| FR-02 | X |  | X | X |  |
| FR-03 | X |  |  | X |  |
| FR-04 | X |  |  | X |  |