**FOCP II**

**CSL108**

Project Report



Faculty name: **Priyanka Rastogi**

Student name: **Diksha Aggarwal, Dhruvarshi Das, Dev Sharma, Divyanshu**

Roll No.: **20CSU036, 20CSU035, 20CSU032, 20CSU037 (respectively)**

Semester: **Second**

Group: **Group 10**

**Department of Computer Science and Engineering**

**The NorthCap University, Gurugram- 122001, India**

**Session 2020-21**

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **S.No** |  | **Page No.** |
| **1.** | **Project Title:** | **3** |
| **2.** | **Description of Project:**  **Problem Statement** | **3 - 4** |
| **3.** | **Problem Analysis**  **3.1 Hardware Requirements**  **3.2 Software Requirements** | **5** |
| **4.** | **Design**  **4.1 Data/Input Output Description:**  **4.2 Algorithmic Approach / Algorithm / DFD / ER diagram/Program Steps** | **6** |
| **5.** | **Implementation and Testing (stage/module wise)** | **6 - 10** |
| **6.** | **Output (Screenshots)** | **10 - 11** |
| **7.** | **Conclusion and Future Scope** | **11 - 12** |

**Project Report**

**Project Title:** Medical Assistance

**Description of Project:**

Medical assistance is designed for any Hospital to replace their existing manual, paper-based system. The new system is to control the following information; patient id, patient name, list of the symptoms the patient is experiencing, and the possible disease associated with the symptom. These services are provided in an efficient, cost effective manner, with the goal of reducing the time and resources currently required for such tasks.

A significant part of the operation of any hospital involves the acquisition, management and timely retrieval of great volumes of information. This information typically involves; patient personal information, list of symptoms, and the disease of which the patient is suffering from.

**The project contains the following sets of information**

**Symptoms.txt**

Contains all the possible symptoms a patient may experience.

**Diseases.txt**

Contains a list of diseases that may be accompanied with a particular disease.

**medicalDatabase.csv**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Patient ID** | **Patient Name** | **Symptom 1** | **Symptom 2** | **Symptom 3** | **Symptom 4** | **Possible Disease** |

Contains all the patient’s records that were taken from the user as input.

**Functions.java**

Class Functions contains three functions namely **symptomCheck**, **updatingCheck** and **symptomValidity**.

symptomCheck DETERMINES THE DISEASE OF THE PATIENT BASED ON THE SYMPTOMS ENTERED.

updatingMDB UPDATES THE MEDICAL DATABASE ONCE WE HAVE THE NAME OF THE DISEASE

symptomValidity TAKES ALL THE FOUR SYMPTOMS AND CHECKS WHETHER THE SYMPTOMS EXIST IN THE CSV FILE AND IF THE SYMPTOMS DOES NOT EXIST IN THE CSV FILE, THEN IT THROWS AN EXCEPTION.

**WritingToCSV.java**

This class contains medicalDatabase() function in which we have made object of FileWriter and CSVWriter classes. It takes data from GUI and writes it in the CSV File and prints a message “Data entered” when the data got successfully entered.

**DataFromCSV.java**

In it we have made an object of Functions class and called the functions symptomCheck and updatingMDB.

**GUIforDisease.java**

In this GUI a variable “diseaseString” is created which takes the value from DataFromCSV and returns the possible disease.

**GUI.java**

A GUI is been created for making the patient enter his personal details, i.e., Patient ID and Patient Name and symptoms experienced. If the symptoms entered by the user is in the .txt file then it will take further data from the CSV file and if not, an exception is raised.

**App.java**

This is the driver class containing the main function through which the execution starts.

**Problem Statement:**

To design a java application to give medical assistance.

**Problem Analysis:**

**Hardware Requirements:**

A compatible processing device with JVM (Java Virtual Machine) and an executable environment (Shell).

* Intel Core i3 processor and above
* 256 MB Ram
* 512 KB Cache Memory
* Hard disk 10 GB

**Software Requirements:**

###### **Windows**

* Windows Vista SP2 or Windows SP1 and above
* Windows Server 2008 R2 SP1 (64-bit)
* Windows Server 2012 and 2012 R2 (64-bit)
* Disk space: 124 MB for JRE; 2 MB for Java Update
* Processor: Minimum Pentium 2 266 MHz processor
* Browsers: Internet Explorer 9 and above, Firefox, Chrome

###### **Mac OS X**

* Intel-based Mac running Mac OS X 10.8.3+, 10.9+
* Administrator privileges for installation
* A 64-bit browser (Safari, for example) is required to run Oracle Java on Mac.

###### **Linux**

* Oracle Linux 5.5+1 and above
* Red Hat Enterprise Linux 5.5+1 6.x (32-bit), 6.x (64-bit)2 and above
* Suse Linux Enterprise Server 10 SP2+, 11.x and above
* Ubuntu Linux 12.04 LTS and above

**Design:**

**Data Input/Output description:**

The user enters all the symptoms experienced by them, Patient ID and Patient Name, then click the Submit button which connect all the JFrames and classes together. If the symptoms exist in the records, then it will give out the possible disease the patient is suffering from.

**Algorithm:**

1. The symptoms experienced by the user are entered in the GUI frame which gets written in the CSV file.
2. At another GUI prompt user will enter his Patient ID and Patient Name and clicks the “Submit” button.
3. Then if the symptoms exist in the file, it registers the Patient Name and Patient ID and if the symptoms do not exist it gives an exception.
4. If the symptoms exist it gives out the possible diseases the patient is suffering from.

**Implementation and Testing:**

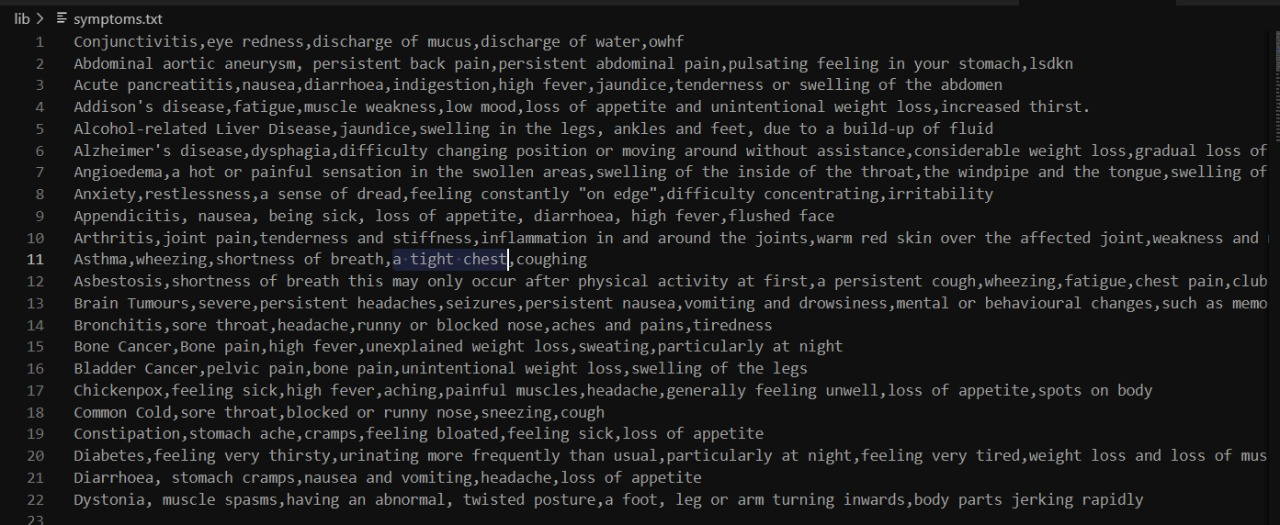
**Implementation**

All the implementation is done accurately and the theoretical design is turned into a working system. The task is completed successfully and it will work for all the users efficiently and effectively.

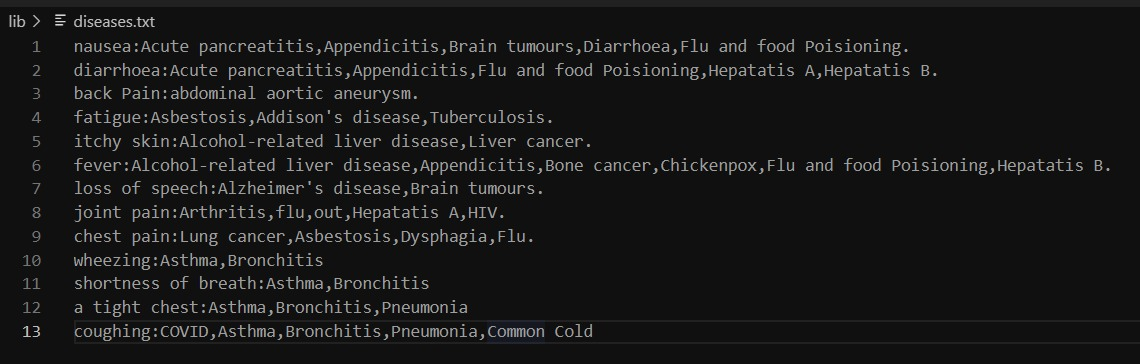
**Testing**

Each module is tested to provide its correctness, validity and also determine any missing operations and to verify whether the objectives have been met. Errors were noted down and corrected.

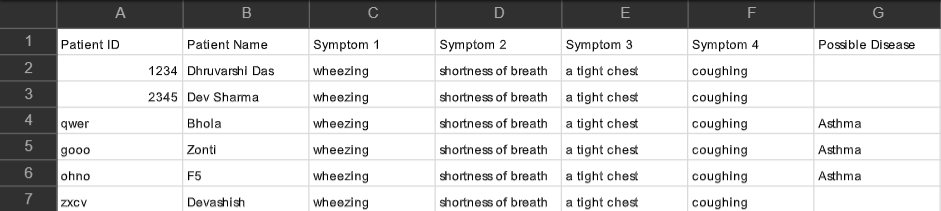
**Symptoms.txt**



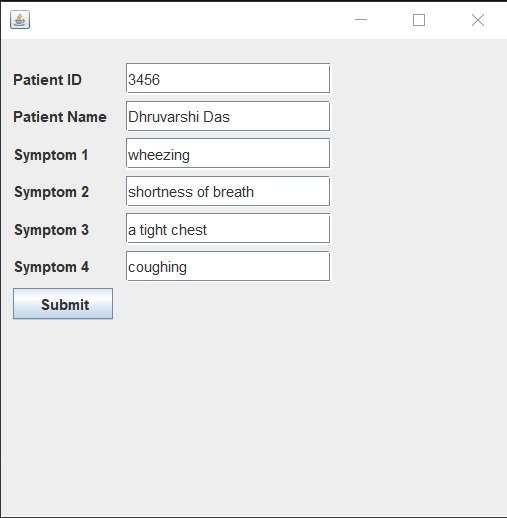
**Diseases.txt**



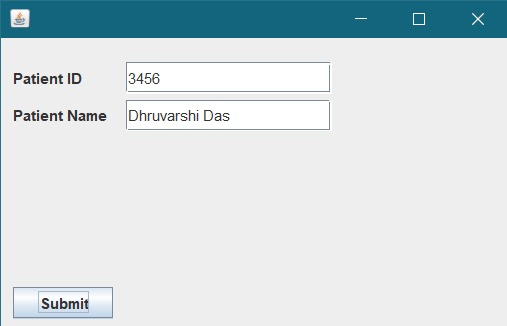
**medicalDatabase.csv**

****

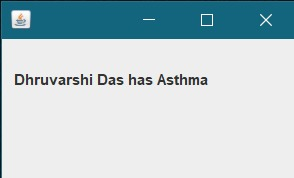
**GUI.java**



**DataFromCSV.java**



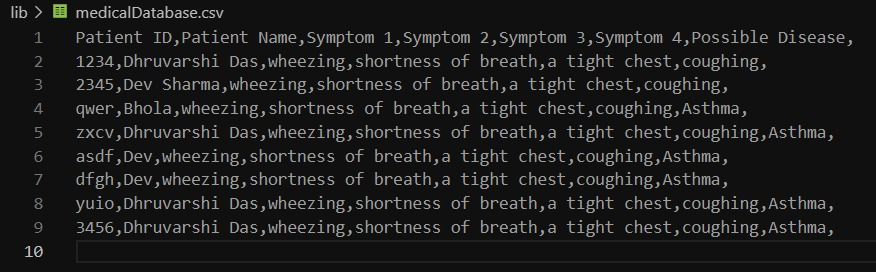
**GUIforDisease.java**



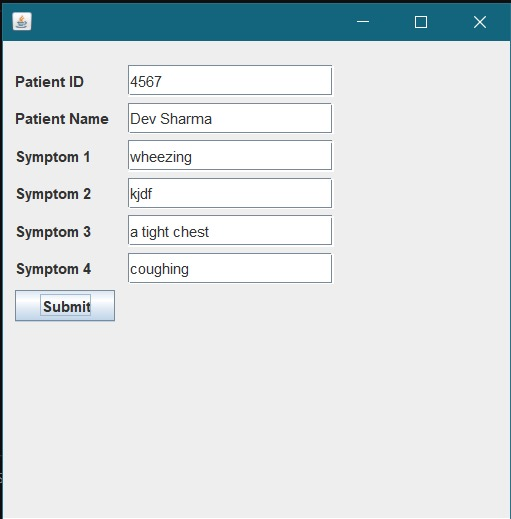
**Console Output**



**WritingToCSV.java**

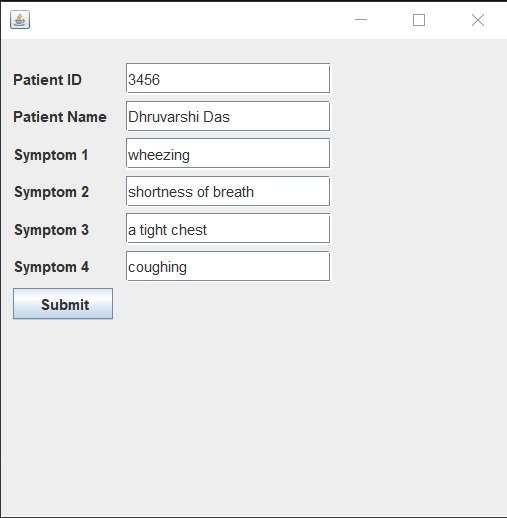


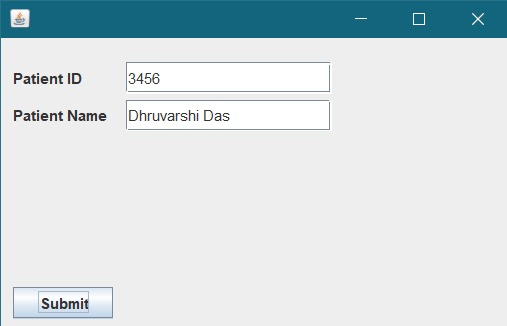
**WrongSymptomsException**

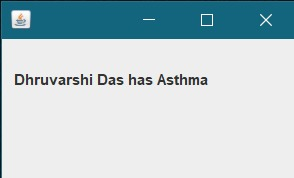




**Output (Screenshots):**









**Conclusion and Future Scope:**

**Conclusion:**

The following conclusion can be deduced from the development of the project.

* Automation of the entire system improves the efficiency
* It provides a friendly graphical user interface which proves to be better when compared to the existing system.
* It gives appropriate access to the authorized users depending on their permissions.
* It effectively overcomes the delay in communications.
* Updating of information becomes so easier.
* System security, data security and reliability are the striking features.
* The System has adequate scope for modification in future if it is necessary.

**Future Scope:**

This application avoids the manual work and the problems concern with it. It is an easy way to obtain the information regarding the diseases a person might be suffering from based on the symptoms.

The system offers reliability and ease - of - access. It can be used as a base for creating and enhancing applications for viewing health records by patients and doctors.

Me and my team members have worked hard in order to present an improved application.

The next enhancement can be that we can develop online services. That mean, if patient have any problems, he can send his problem to the doctor through internet from his home, then doctor will send reply to him. In this patients have the same ID and Name.