Remario Richards | Shervain Barrett

1303783| 1303991

MOH PROLOG DoCUMENTATION

**User Manual**

The following contains all the necessary instructions needed to utilize the program. The software was design from the perspective of an administrator, which is to say that the administrators controls everything from login, to querying of data. The only exception is data input from the user.

There are five scenes or components in the program, firstly the login for the administrator default username and password is [admin]. Credentials are recommended to change, this is only for demonstration purposes only. After successful login, the user is posed with two options, either they query the knowledge base or input data. Only the administrator can query the system, and the user can only input the relevant data which must be administered for transparency purposes.

The following includes all the fields required for user information input.

|  |  |
| --- | --- |
| Name | User name |
| Age | User age |
| Ethnicity | User Ethnicity |
| Feet (height) | User height in feet |
| Inches(height) | User height in inches |
| Weight(kg) | User weight in kilograms |
| Waist Circumference | User Waist Circumference in centimeters |
| Gender | M – male | F - female |
| The following below are true and false genial questions for previous family history. When selecting you’re in the program hover over the radio button to see a short description. |

**Have any of the members of your immediate family or other relatives been diagnosed with diabetes (type 1 or type 2)?**

(0 - No)

(1 - Yes (grandparent, aunt, uncle or cousins))

(2 - Yes (parent, sibling, and child))

The following are general questions regarding health status.

Do you exercise at least 30 minutes every day (Yes/No)?

Do you eat vegetables or fruits every day (Yes/No)?

Have you ever taken medication for high blood pressure on regular basis (Yes/No)?

Have you ever been found to have high blood glucose (Yes/No)?

After successful input for the aforementioned various information set. The user will be greeted with a list of specific recommended steps to enhance their lifestyle. Thus alleviating their chances of type 2 diabetes. The user will also be shown their current status in developing type 2 diabetes within 10 years. Ranging from status such as low, slightly elevated and of course highly elevated.

**Administrator section.**

The following details all the necessary instruction’s needed to query the knowledge base. After successful login, the admin can query the database by selectively clicking the query action. The following are all the current query actions and returned results.

|  |  |  |  |
| --- | --- | --- | --- |
| stat\_min\_age | Gets the minimum age in database | EXAMPLE USAGE | |
| stat\_max\_age | Gets the maximum age in database |
| stat\_family\_history | Gets count of total family history |
| stat\_avg\_age | Get average age of all persons |
| stat\_num\_high\_risk | Counts number of high risk persons |
| stat\_user\_all | Get all info for person with Name | stat\_user\_all->nick |
| stat\_user\_weight | Get weight for person with Name | stat\_user\_weight->nick |
| stat\_user\_height | Get height for person with Name | stat\_user\_height->nick |
| stat\_user\_bmi | Get bmi for person with Name | stat\_user\_bmi->nick |
| stat\_user\_age | Get age for person with Name | stat\_user\_age->nick |
| stat\_user\_ethnicity | Get user ethnicity with Name | Stat\_user\_ethnicity->nick |
| stat\_height\_filter\_below | Filter height below threshold | stat\_height\_filter\_below->6 |
| stat\_height\_filter\_above | Filter height above threshold | stat\_height\_filter\_above->11 |
| stat\_weight\_filter\_below | Filter weight below threshold | stat\_weight\_filter\_below->134 |
| stat\_weight\_filter\_above | Filter height above threshold | stat\_weight\_filter\_above->123 |
| stat\_family\_history\_filter | Filter family history | stat\_family\_history\_filter->0|1|2 |
| stat\_gender\_filter | Filter gender | stat\_gender\_filter->male|female |
| stat\_risk\_filter | Filter various risk | stat\_risk\_filter->High etc |
| stat\_age\_filter\_below | Filter age below value | stat\_age\_filter\_below->20 |
| stat\_age\_filter\_above | Filter age above value | stat\_age\_filter\_above->23 |

**Notes:**

* The query scene contains a clear button at the administrator discretion.
* When inputting user information in the input scene, if by any chance the user did not provide the necessary data an error will be thrown.
* Certain query displays a dialog
* For stat methods that specify a magnitude such as below or above. Format returned [*gender,age,weight,height(m),waist-circumference,exer-amt,veg-fruits,high-bp,high-bg,risk*]
* If a query was submitted and no result was returned, it simply means no data was found.
* (Do you exercise at least 30 minutes every day ) = exer-amt.
* (Do you eat vegetables or fruits every day) = veg-fruits.
* (Have you ever taken medication for high blood pressure on regular basis) = high-bp.
* (Have you ever been found to have high blood glucose) = high-bg.
* Risk = [low, slightly-elevated, moderate, high, very-high]

**Project Group Report**

The project was built by Remario Richards and Shervain Barrett. They were two sub dividing roles, firstly the creation of the prolog knowledge base system(Shervain) and the design and implementation of the graphical user interface and connection interpreter between java and prolog(Remario).

*Lessons learnt by Remario.*

**GUI**

In implementing the graphical user interface, I learnt about multiple scenes for which the user had to interact with. Learnt how to seamlessly transition from one scene to the next. Scenes are the actual user interface any person can interact with at any given moment in the program.

**JPL**

In connecting jpl with java, this proved a very daunting task, they   were a lot hurdles ranging from path environment variables settings. To jpl initialization and class diagrams. The jpl journey was a complete learning experience. I learnt to mimic logic statements from prolog to java using symbolic classes.

**System Design Abstraction**

The design includes the java programming language for graphical user interface implementation, particularly javafx. This framework was primarily chosen because, JavaFX is designed to provide applications with such sophisticated GUI features as smooth animation, web views, audio and video playback, and styles based on Cascading Style Sheets (CSS). Prolog was utilized for the knowledge base implementation, mainly due to the fact that prolog excels at declarative formal logic statements. The main java application comprised of three main classes. First the main class, this class includes all the constructs that was used to build the graphical user interface. Secondly the model class contains the models to represent the prolog logic statements. For the aforesaid the prolog/java interface (JPL) was incorporated into the application. JPL is a library using the SWI-Prolog foreign interface and the Java jni interface providing a bidirectional interface between Java and Prolog that can be used to embed Prolog in Java as well as for embedding Java in Prolog. In both setups it provides a re-entrant bidirectional interface. And finally a cascading style sheet for the overall look and feel of the program.

*Lessons learnt by Shervain*

**Determining Type II Diabetes Risk**

Diagnosis of Type II diabetes can be dependent of a multitude of different factors. As such, the approach was taken to ask users 10 questions which their responses being waited based on the importance that factor plays in a patient developing Type II diabetes. After the short questions are answered, a user **score** is calculated and their risk for developing the disease is assessed. The maximum number of points that a user may accumulate from the questions is 26. Diabetes diagnosis is then selected based on the number of points actually scored based on the following:

* 0 – 6: Low
* 7 – 11: Slightly Elevated
* 12 – 14: Moderate
* 13 – 20: High
* > 20: Very High

**Factors Considered**

Although, multiple questions were used to determine the risk that users faced with developing Type II Diabetes. Some factors were more important than others. For example, persons who had close family members who had been diagnosed with any form of diabetes were at a greater risk of developing same. Additionally, the risks associated with persons who have a large waist circumference, high blood glucose levels as well as being over the age of 65 are usually high/very high.

Another import factor that was considered was the Body Mass Index (BMI) of a user. This is a simple traditional metric of determining whether or not a person is overweight and can be calculated using the following formula:

Classification according to BMI is done using the following rules:

* < 18.5 – Underweight
* 18.5 – 24.9 – Normal Weight
* 25 – 29.9 – Overweight
* > 30 – Obese

**Persisting the Knowledge Base**

Although **assert** is used to add new rules to a prolog database within a given session. When this session has closed, all rules that were contained in the database are discarded of. As such, the information needed to be persisted across different sessions to ensure that the system operated as a true expert system. As such, whenever a new record related to patient data is added to the in memory database, this information is also stored within a Comma Separated Values (CSV) file that would have contained all prior records. Then when a new session is started, the values from this file is fetched and asserted back to the knowledge base.

Although, a user enters his weight in **pounds** and his height in **feet and inches**. Based on the calculations that need to be made, these are the converted to their equivalent metric values (pounds -> kilograms, feet/inches -> meters), in the data persistence process, only the metric values are kept for these records. Conversion formulae are below:

The reading and writing of this CSV file is handled by using built in prolog modules that were designed to read csv files easily and have additional predicates to assert these records to the database without the need for a lot of code.

**Statistics associated with the Knowledge Base**

As listed above, there are various statistical predicates that may be called to get prior data from the knowledge base. Additionally, the knowledge generates are prompt to the user alerting him/her when over 75% of records in the knowledge base has been diagnosed as having a **High/Very High** risk of developing Type II diabetes in the next 10 years. Other general information that can be queried, as records above or below any age, weight (kg) and height (m). Database records can also be filtered by Risk level, shared family history and a user’s gender.