Software Requirements Specification

Project name:

**Database Monraz**

Written by: Christopher Monraz

CIT 284 Spring 2019

**Table of Contents**

**1.Introduction**

1.1 Purpose

1.2 Intended Audience and Reading Suggestions

1.3 Product Scope

1.4 Technologies to be used

1.5 Overview

**2. Overall Description**

2.1 Product Perspective

2.2 Product Functions

2.3 User Classes and Characteristics

2.4 Operating Environment

2.5 Design and Implementation Constraints

2.6 Assumptions and Dependencies

**3. External Interface Requirements**

3.1 User Interface

3.2 Use-Case Reports

**4. System Features**

4.1 Database creation

4.2 Database population

4.3 Database query

**5. Other Nonfunctional Requirements**

5.1 Performance requirements

5.2 Security Requirements

5.3 Software Quality Attributes

Appendix A: Glossary

Appendix B: To Be Determined List

**1.Introduction**

**1.1 Purpose**

Database Monraz is a relational database structured by tables with relations between the information stored in it. This SRS will describe the functionality and the main functions that this project will articulate.

**1.2 Intended Audience and Reading Suggestions**

This SRS is intended for the users, developers and testers of the software. The level of expertise recommended to understand the terminologies used in this document is of moderate knowledge in Computer Science and an entry level in relational databases.

**1.3 Product Scope**

The purpose of the software is to provide a reliable and secure program to store information. This database will stand out from the others thanks to its facility to equip and configurate functions to manipulate the data inside the tables depending on the needs of the user.

**1.4 Technologies to be used**

GUI Framework Qt

Programming language C++

IDE Clion

**1.5 Overview**

This software will be a useful tool to store and consult information saved in structured tables. The database will bring flexibility to the variety of projects or personal uses where the user may decide to implement the software. Nowadays, when information is flooding from everywhere (school, jobs, personal activities, hobbies, internet) people will find this software as an extension of his/her memory. A place to store the information that we don’t need to have wandering in our brains, or that we are afraid to forget.

**2. Overall Description**

**2.1 Product Perspective**

This software is created from the necessity to store medium-sized datasets without the use of the dominant databases such as MongoDB, MySQL, PostgreSQL. The database will be created to accommodate the needs of a college student to organize and to store information. It’s desirable from this software to be scalable and versatile enough to be implemented and used by other programs in order to be useful for future projects that may need data storage.

**2.2 Product Functions**

This software must be capable to create and save multiple databases at the same time. Each database must contain at least one table that will be integrated of rows and columns. The data entered in the rows and columns can be of type integer, character, string, double and boolean. Also, the rows and columns must be able to be added and deleted from each table. By last, the tables will be capable to link between each other using primary and foreign keys.

The database software must be able to generate responses to specific queries from the tables and to save the result in text documents.

The database will be able to handle private information. Therefore, the use of password and usernames is required.

**2.3 User Classes and Characteristics**

-Domestic user. – This class is expected to be the most common. Domestic user refers to any user who will use the database for personal needs. For that reason, the size and complexity of the database is expected to be minimum. The Domestic user is expected to use the software at least once a week, and the user may create a username a1 password to protect the accessibility of the information.

-Third party programs. -This user refers to any software that will implement the code inside its libraries to perform data management using the database. This user will be a class with higher expertise who will be able to access and modify the source code of the program to implement the software in the best way for its needs. The privacy of the data and the speed to make consults is vital for this class of user.

**2.4 Operating Environment**

This software will be developed in Windows 10. Therefore, total compatibility with this operating system is expected.

**2.5 Design and Implementation Constraints**

The software must be able to run in a computer with the next minimum specifications

* OS:Windows Vista, 7, 8 Pro+
* Processor:1.2GHZ CPU
* Memory:1 GB RAM
* Hard Drive:2 GB HD space

**2.6 Assumptions and Dependencies**

The GUI (Graphic User Interface) will be developed with the third-party software called QT 3D Studio. The decision to use this software was made based on the popularity and the high amount of documentation available in the internet of the use of this tool.

**3. External Interface Requirements**

**3.1 User Interface**

The first scene that will appear when opening the program will be the log in screen with the text fields to input the username and the password and the buttons to login and to exit. If the access information is correct the next scene that will appear is the Database list, that will show all the database owned by the user. When the database is clicked the database will be display by tables, each table being on its own sheet. In the Database list scene there will be also a button to create a new database.

If a new database is created the Database management scene will appear. This scene can create new tables, edit existing tables, populate the tables with data, create links using primary and foreign keys and the possibility to perform queries. On the menu of the screen there will be options to save the database, exit the program, open another database, log out of the current session and an option to change the location where the database is being stored.

**3.2 Use-Case Reports**

1.- **Database owner**- is the person who created the database an has full access to it to manipulate it.

* **Create Database**- the user creates a new database with a specific number of tables. The name of the database is entered as well as the path where the files of the database will be stored.
* **Populate Database**- the user inputs data to specific tables. The data entered must be well structured and the type of data must be respected, otherwise the software will pop up a message warning about the error.
* **Edit Database**- the user adds more tables or delete existing ones or certain data is eliminated from the database.
* **Query**- the user request for the data located in certain table and in the specified tables. The result of the consult then is saved in a separate text file is desired.

**Name of Use Case** – Create Database.

**Description** – a new database is created by the user.

**Pre-Condition** – if a path to save the database is given, it must exist in the system.

**Normal Flow of Events**-

* A name for the database is given.
* If desired, a path to save the database is given.
* The database is created.

**Name of Use Case** – Populate Database.

**Description** – a new row is added to a column of a table.

**Pre-Condition** – The table must exist, and it must have enough space for the new entry. Also, the data entered must be of the appropriate type.

**Normal Flow of Events**-

* A table from a database is selected.
* The user clicks the “EDIT” button
* The user clicks the “ADD” button.
* The data is entered in the specified format given by the database.
* The changes are stored.

**Name of Use Case** – Edit Database

**Description** – A database is edited by the user to add, delete or change one or more tables.

**Pre-Condition** – the database must exist in the system and the user has the permits required to edit the database.

**Normal Flow of Events**-

* The user logs in.
* The user selects the database.
* The “EDIT” button is pressed.
* The user selects a table to delete it or to change it, or a table is changed (add or delete a row or a column).

**Name of Use Case** – Query

**Description** – a specific consult is generated by the user.

**Pre-Condition** – a database must exist, and it must have at least a table with some data.

**Normal Flow of Events**-

* The user logs in.
* The user selects the database.
* The “QUERY” button is pressed.
* The way to output the query is selected (console, window view or in a text file).
* The consult is entered using the appropriate syntaxis.
* The user receives the answer of the consult in the format selected.

2.- **Viewer**- is a user who has been given permission to access the data of a database, but he/she can’t manipulate it.

* **Query**- the user request for the data located in certain table and in the specified tables. The result of the consult then is saved in a separate text file is desired.
* **View tables**- the software display a table selected by the user to display it in the specified output (window, console or printed in a new text file).

**Name of Use Case** – Query

**Description** – a specific consult is generated by the user.

**Pre-Condition** – a database must exist, and it must have at least a table with some data.

**Normal Flow of Events**-

* The user logs in.
* The user selects the database.
* The “QUERY” button is pressed.
* The way to output the query is selected (console, window view or in a text file).
* The consult is entered using the appropriate syntaxis.
* The user receives the answer of the consult in the format selected.

**Name of Use Case** – View Tables

**Description** – a specific table is displayed to the user.

**Pre-Condition** – a database must exist, and it must have at least a table with some data.

**Normal Flow of Events**-

* The user logs in.
* The user selects the database,
* The user clicks the “VIEW” button.
* The table to view is selected by the user.
* The method to display the table is selected (window view or console).
* The table is displayed in the format selected by the user.

**4. System Features**

*Priority scale from 1(Low priority) to 10 (high priority)*

**4.1 Database creation**

This feature has a priority of 10, is the main characteristic of the software and the one that must be implemented first. It must be possible to create a Database as long as the space in memory where the software has been installed is greater than 1 MB. The database creation will be able to add up to 100 tables, this limit has been placed just to determine the scale of the projects it will contain, the performance issues related to database with more than 100 is TBD. Every data entry to create tables of certain number of rows and columns must be an integer value. Decimal values must be omitted, and the user must reenter the value.

**4.2 Database population**

This feature has a priority of 9. The database must be able to be populated with data entered by the user. The data entered in each column must be of the same type. The type will be decided when the table is created, and it is immutable. Every time a field is filled the software must validate the information to make sure the type is correct. An existing information can be changed or deleted by the owner of the database.

**4.3 Database query**

This feature has a priority of 7. Specific data from the database must be able to be consulted by the owner of the database and Read Only users (users with permit from the creator of the database). The consult system must have these commands:

* Select.- Selects the specified columns from a table
* From.- Specifies the table where the information will be taken from.
* Where.- Selects the rows where the column specified has the entered value.

**5. Other Nonfunctional Requirements**

**5.1 Performance requirements**

Every action performed in the database must last up to a second. Any response time longer than a second must be listed for revision by the developer to find the causes and to solve it. The only process that can last up to 2 seconds is the consult of databases with tables populated with 5,000 values.

**5.2 Security Requirements**

Every value entered in a database must be considered private. What this means is that no one without permit of the database’s owner can access it. The log in and the requirement of a password to enter the database is a must in this software.

**5.3 Software Quality Attributes**

The software must be:

**Adaptable**– If under the petition of an user and approval of the developer, a new command or function needs to be implemented the program must be adaptable to this changes. The new command should be able to coexist with existing ones without generating errors or exceptions.

**Maintainable**– The software must be able to be upgraded easily, without major changes in many classes. To achieve this, the Object-Oriented paradigm must be followed scrupulously. The software must generate a text document with the errors and exceptions generated by the user. This record will allow the developer to find and to fix any problem related to the software.

**Usable**-- One of the main qualities of this program must be usability: the capacity to be implemented in different scenarios and projects with few changes in configuration.

**Appendix A: Glossary**

SRS -> Software Requirements Specification

OS -> Operative System

**Appendix B: To Be Determined List**

* Compatibility of the software in IOS
* The implementation of more query commands such as Union, Except and Intersect
* Implementation of encryption algorithms to protect the databases and the user passwords.