Software Requirements Specification

for

DoQL

Version 1.0 approved

Prepared by Abdullah Mahrous

DoQL teamwork

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

DoQL is a desktop application created to simplify the task of database creation and designing for all developers, so they can design their ERD or relational schema with simple and easy GUI and convert it into SQL commands.

1.2 Document Conventions

Paragraphs font are 14, paragraphs headings font is16 and subtitles font are 18, font style is italic, heading and subtitles are bold, paragraphs font family is Arial, headings font family is Times.

1.3 Intended Audience and Reading Suggestions

This SRS is mainly for developers and stakeholders, as the whole application is intended for developers.
The best sequence for reading this document is to start with the overview and then the purpose so you can understand the idea and afterwards head for the design and implementation.

1.4 Product Scope

DoQL is created to help the developers to design or build their databases by letting theme design their ERD or relational schema through a simple and easy to use GUI so they can handle the designing smoothly and then convert it into SQL command.

1.5 References

No references yet

2. Overall Description

2.1 Product Perspective

DoQL is a new self-contained software designed to help developers in their struggle with databases and it does not use any vendor API or system to perform its functionality.

2.2 Product Functions

This software will give the user the ability to:

- *Design their database in form of ERD or relational schema.*
- *Save their databases in form of JSON files.*
- *Impot, save and edit the databases.*
- *Encrypt the JSON files with password to secure your work.*
- *Take your databases to any device by sharing the JSON file.*
- *Convert your diagrams to SQL commands.*
- *Toggle between relational schema and ERD.*

2.3 User Classes and Characteristics

There are 4 classes in DoQL: database, table, attribute, reference.
Database is the main class in our system, and it has id, name, type and a list of tables.
The table class has id, name and a list of attributes, and the attribute class has the same properties as the table with datatype, constraints and reference, the reference class has table id, attribute id and actions.

2.4 Operating Environment

DoQL is a desktop application built using windows forms, it uses .Net version 6 and it only operates on windows.

2.5 Design and Implementation Constraints

There are several limitations that could hold the developers and users in using DoQL, such as if they don't have .Net framework on their machine or if they use another operating system than windows such as Lunix or Mac, another limitation is the hardware efficiency as designing the ERD and relational schema through good GUI takes a lot and complex computation which could be a problem if your hardware isn't efficient enough.

2.6 User Documentation

There are no manuals or user documentations yet.

2.7 Assumptions and Dependencies

As a desktop application DoQL requires a specific environment to operate on and that must be windows operating system also the operating system must have .Net on it, and the user must know how to design ERD or relational schema.

3. External Interface Requirements

3.1 User Interfaces

DoQL gives the user an easy GUI to import the databases that he/she already have and display it in form of cards showing the name of the database and its type, you can also create new one by a click of a button and fill the form that appear to create new ERD or relational schema, after that you can drag and drop your components to design your database as you like and define the

relations between the tables so you can export the SQL commands afterwards by choosing which DBMS you want your commands to run on.

3.2 Hardware Interfaces

This software supports only desktop version and windows operating system and good hardware requirements for the optimal experience.

3.3 Software Interfaces

DoQL is built by windows forms and .Net version 6, it also uses JSON files to store databases of the user on the device and to make it portable.

3.4 Communications Interfaces

DoQL does not communicate with any other software or servers as it's a self-contained application and requires no interaction with other services than the described.

4. System Features

4.1 Importing databases

4.1.1 Description and Priority

Importing databases and auto importing so you can visually edit them or secure them with password.

4.1.2 Stimulus/Response Sequences

The user clicks on the import button and select the wanted file so it can be displayed as a card.

4.1.3 Functional Requirements

As the application opens the databases that were created or imported before will be automatically imported as the splash page appears.

the user then can import the wanted database that he/she want and select it from the file explorer and the file will be in form of JSON file, and after selecting the database will appear as a card showing the name of the database and its type and if it has a password the user must enter it.

4.2 Creating new database

4.2.1 Description and Priority

Creating new database and setting its settings.

4.2.2 Stimulus/Response Sequences

The user clicks on the new database button and fill the settings form to create the new database.

4.2.3 Functional Requirements

as the user clicks on the new database button a form appears with the name of the database field and its type and if the user wants to secure it with password so the software will encrypt the JSON file.

4.3 Designing the database

4.3.1 Description and Priority

Designing the ERD or relational schema.

4.3.2 Stimulus/Response Sequences

When the user clicks on a card or create new database a workspace opens in front of him so he could design the database

4.3.3 Functional Requirements

When the user clicks on a card or create new database a workspace opens in front of him so he could create new entity and attributes and relations and define each type of them and define the relations between all of them, and he/she could select the

datatype of each attribute and define the action type and the constraints, and if wanted you could toggle to relational schema and create tables and columns inside of theme and create the relations between the columns.

4.4 Exporting your database to SQL commands

4.4.1 Description and Priority

Export your database as SQL commands to 3 different DBMS.

4.4.2 Stimulus/Response Sequences

As the user finish the database design, he/she could export it into SQL commands by clicking the export button and selecting the wanted DBMS.

4.4.3 Functional Requirements

DoQL offers the user to export their database ERD or relational schema to SQL commands that can run on 3 different DBMS as the user chooses, DoQL supports: MySQL, SQL Server, SQLite. The commands come out as a string that could run on the chosen DBMS.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

DoQL requires heavy computations as the framework redraw the components when dragging and dropping them and when the components are too many the dragging of the screen makes the computation for all those components a lot so it requires sufficient hardware.

5.2 Safety Requirements

DoQL is mainly made for documenting your databases, so the commands could be wrong if the design is not accurate, also if the user secure their database with a password and forgot it won't be decrypted without it.

5.3 Security Requirements

As DoQL secure the databases with password it encrypts those databases so none could access them without the owner permission.

5.4 Software Quality Attributes

The software is portable, and the files are portable and flexible.

5.5 Business Rules

DoQL has no Business Rules as it's an open-source project created to help developers document their databases.

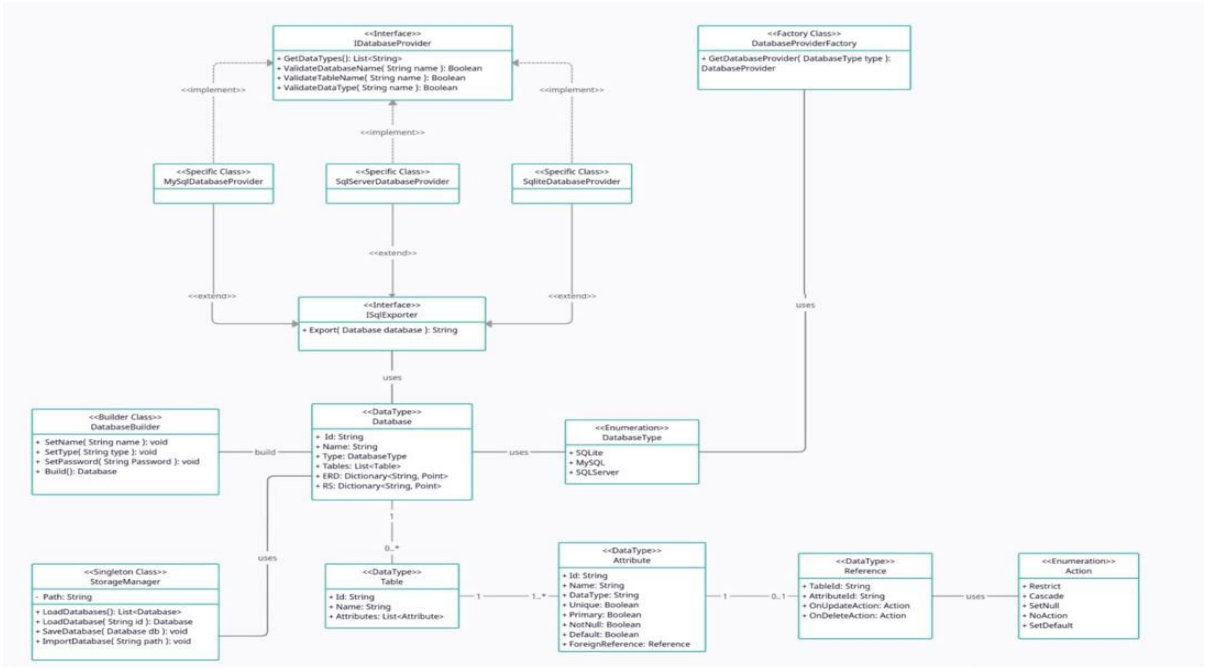
6. Other Requirements

There are no other requirements.

Appendix A: Glossary

There is no Glossary or acronyms.

Appendix B: Analysis Models



Appendix C: To Be Determined List

This list will be written later.