MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE

NATIONAL TECHNICAL UNIVERSITY

“KHARKIV POLYTECHNIC INSTITUTE”

Department “Software Engineering & Management Information Technologies”

Report from lab № 1

discipline «Requirement analysis»

Executed by:

Student of group ІF-33zh

Y.D. Bilyk

Checked by:

Melnik K.V.

Kharkiv

2015

Theme: development of requirements specification according to IEEE 830-98

Task: to get familiar with the IEEE 830-98 and to develop requirement specification to software in the given UoD with its help.

Main part:

# 1 Introduction

### Purpose

The purpose of the given SRS is to delineate functional and non-functional requirements for the software for cars audition, concerning UoD “Cars shop”. The intended audience for this SRS is the customer and developer-team.

### Scope

The software product to be produced is a Cars DBMS for the audition of available cars and their components.

The ‘Cars DBMS’ software should be able to work with the cars- database and provide following functions: create, edit, delete, select, sort items of database, generate reports basing on the cars- database.

The goal of developing such DBMS is to provide the easy, safe and clear way of organizing the selling process in the cars shop. The system could be used by ordinary users (sellers), who are not familiar with the database manipulation process.

### Definitions, acronyms, and abbreviations

Definitions, acronyms and abbreviations you can find in the table 1.1.

Table 1.1 - Glossary

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pos. | Full name of term | Short name | Definition | Synonyms/ homonyms | Related words |
| 1. | Database management system | DBMS | System, for database manipulations |  | Cars |
| 2. | Car |  | A self-propelled road vehicle designed to carry passengers. | vehicle, item, machine, automobile |  |
| 3. | User |  | A person (seller), who directly interacts with the DBMS |  | seller |
| 4. | Customer |  | The cars’ shop director. |  | cars’ shop |
| 5. | Operating System | OS | The software system which operates with the hardware. |  |  |

### References

IEEE Std 830-1998, IEEE Recommended Practice f**or Software Requirements Specifications.**

### Overview

Given SRS describes the functional and non-functional requirements to the ‘Cars DBMS’ software in UoD: Cars shop. The SRS has 3 clauses.

Clause 1 explains the scope of the product to be developed. Clause 2 gives overall description of the product. Clause 3 lists the specific requirements to the product.

# 2 Overall description

## 2.1 Product perspective

The product is going to be used directly in order to audit the inventory of the cars shop. The product is independent and totally self-contained, so it’s not the part of the larger system. The product is expected to work with the MySQL database and Microsoft OS.

### 2.1.1 System interfaces

As the given product is not the part of the larger system, it has no system interfaces.

### 2.1.2 User interfaces

The described software solution has a forms-based interface that displays different forms to each user. The DBMS is going to work only with relational databases. Users can fill out all of the form entries to insert new data, or they fill out only certain entries, in which case the DBMS will retrieve matching data for the remaining entries, edit, delete entries. Forms are usually designed and programmed for naive users as interfaces to canned transactions. The system has utilities that define a form by letting the end user interactively construct a sample form on the screen. The developed DBMS also has buttons and forms to create/delete the database, connect/disconnect to the database, to query the database, to create stored functions/procedures to save changes made, will have the interface to view and modify tables of the database. The developed DBMS will not have: possibility to create/delete triggers and views. It also has interfaces to backup/rollback the database. The DBMS also has forms to configure the low-level part connected with the connection to the database.

### 2.1.3 Hardware interfaces

For the correct functioning of the DBMS such hardware devices are needed: Intel Core machine, network card (in order to connect to the database), modem (if needed to connect to the database), keyboard, mouse (or other pointing device), monitor, database server to connect to. To connect to the database, at least one port should be opened and configured by the user.

### 2.1.4 Software interfaces

The following software interfaces are needed:

1. OS Windows 7 (Service Pack 1 (SP1)) (operating environment for DBMS);
2. MySQL database (database to connect and work with);
3. .NET Framework 4.5 (execution environment for DBMS)

### 2.1.5 Communications interfaces

The developed DBMS will be able to use both TCP/IP as well as Named Pipes networking protocol to connect to the database.

### 2.1.6 Memory constraints

To work properly the DBMS needs 50 MB of the disk space and 300MB of available RAM (Random Access Memory).

### 2.1.7 Operations

There’re two modes of operation: user and admin modes.

User can: insert/edit/delete database table entries, query the database.

Admin can: create/edit/delete stored procedures/functions, create user forms for modifying the tables, configure DBMS connection to the database, create/delete databases, connect/disconnect from the database, backup/rollback the database.

### 2.1.8 Site adaptation requirements

In order to adapt the DBMS to the installation site, the connection to the database should be configured by defining the type of connection, port, password and address of connection.

## 2.2 Product functions

So, the system should provide the following functions: create/edit/delete tables, search in the table, query the table, create and use stored functions/procedures, backup/rollback the system, changes saving, connection configuration features.

## 2.3 User characteristics

The user-friendly interface of the DBMS will allow a user, who’s never worked with such systems to work with the database tables after 1 hour of teaching. In order to work more efficient, the user is expected to know the MySQL syntax to create queries to the data table without using the specific GUI features of construction of such queries.

The user can be an administrator of such system, if he/she has knowledge in system architecture, appropriate network protocols, MySQL syntax and has experience in working with such systems.

## 2.4 Constraints

The software solution should provide parallel computations (in order user GUI interface not to freeze during background operations), time of an average query performance should be O(n) [where n – the number of queried entries], the DBMS should support MySQL syntax, the system should provide all possible exceptions handling mechanism, the system should provide safe connection to the database.

## 2.5 Assumptions and dependencies

If the Windows operating system is not available for this software solution, the appropriate open source analogs of .NET Framework 4.5 should be used instead to support the system.

## 2.6 Apportioning of requirements

The support for stored procedures/functions, bakup/rollback of the database may be delayed until the future versions of the system.

# 3 Specific requirements

## 3.1 External interfaces

1. Create table. User creates table by defining its name, names of fields and their types, possible initial entries and the DBMS system forms the appropriate MySQL CREATE TABLE query to the database.
2. Delete table. User chooses table to delete and the DBMS forms the appropriate MySQL ALTER TABLE query.
3. Insert entries. User chooses table to insert queries into and the DBMS forms the appropriate INSERT MySQL query.
4. Edit entries. User edits the entry in the database table and the DBMS forms the appropriate UPDATE MySQL query.
5. Delete entries. User selects the entries in the database table to delete and the DBMS forms an appropriate MySQL query DELETE.
6. Database connect. User inputs login, password, address, connection type, port of the database and the DBMS connects to it and returns the result of the connection (Connected OK or ERROR).
7. Database disconnect. User chooses database to disconnect and the DBMS disconnects from the database after verifying the integrity of the database. After disconnection it notifies user with the result of procedure (Disconnected OK or ERROR).
8. Create stored function/procedure. User inputs the stored function/procedure after selecting the appropriate database and the DBMS system compiles and executes it. If the stored function/procedure has errors DBMS notifies user about it.
9. Backup the database. User chooses the database to backup and after the integrity check, the backup of the database is created by the DBMS in the specified location and the result of the procedure is returned (Backup OK or ERROR).
10. Rollback the database. User chooses the database to rollback and chooses the location of the backuped database to use. The system rolls back the database and returns the resulting state of operation (Rollback OK or ERROR).

## 3.2 Functional requirements

### 3.2.1 User class “Ordinary User”

#### 3.2.1.1 Validity checks

The system should check the validity of every input to the data table (field types, unique fields, etc.).

#### 3.2.1.2 Connection state checks

The system should check the state of connection to the database every 5 sec.

#### 3.2.1.3 Exception handling

The system should response to abnormal operations: incorrect input, invalid query, connection errors, stored procedure/function compilation/execution errors, database integrity errors.

### 3.2.2 User class “Administrator”

#### 3.2.2.1 Validity checks

The system should check the validity of the configurations made and propose the solutions for some configuration problems. The system should propose the backup every day.

#### 3.2.2.3 Exception handling

The system should response to abnormal operations: incorrect configuration entry, connection errors, stored procedure/function compilation/execution errors, database integrity errors, deletion of referenced tables, backup/rollback errors.

## 3.3 Performance requirements

Static requirements:

1. the number of databases to be supported: from 1 to 20;
2. the number of users to be supported: from 1 to 50;
3. the number of simultaneous users to be supported: 1;
4. amount of information to be handled at once: 1-20 databases (can be opened at once).

Dynamic requirements:

1. 98% of transactions shall be processed in less than 1s;
2. connection time to database shall be less than 10 s;
3. query time to datatable shall be of order O(n) [where n – number of queried entries];
4. processor workload shall be less than 50% (for Inter Core i3 processors and higher).

## 3.4 Logical database requirements

The DBMS will use its own database to store:

* user name;
* user password;
* user permissions;
* integrity constraints;
* source path of configuration file;
* current license information (license key, license expiration date, etc.).

## 3.5 Design constraints

The system will be used with limited 50 MB/sec network for database access.

The database is going to be placed on external web-server.

### 3.5.1 Standards compliance

All data should be encoded using 16-bit Unicode (UTF-16).

## 3.6 Software system attributes

### 3.6.1 Reliability

The system should handle all possible exceptions in working with database and propose possible solutions to the user (if necessary).

The DBMS should make the backup of all environment configurations changed during the first installation procedure. It also has to backup all configurations after the first configuration of the product by the user.

### 3.6.2 Availability

The system should provide recovery and restart features. It also have to provide rollback of changes function.

### 3.6.3 Security

1. The system should utilize the AES-256 in order to store the user passwords in the database;
2. The system should log the user and network activity;
3. Check data integrity after all changes made to database;
4. The system should restrict ordinary user to change configurations of the system.

### 3.6.4 Maintainability

The system should provide user-friendly GUI, which user can utilize after 1h of training. The system should provide help system in order to help user in program usage. The system should provide possible solutions in the exceptional moments (if necessary).

### 3.6.5 Portability

The system is to be developed using the C# programming language and .NET Framework. The entire GUI environment is to be developed with Windows-dependent code. All other core-modules are operating system independent but are dependent on .NET Framework platform. However all the code is optimized for the Intel machines instruction set.

## 3.7 Other requirements

### 3.7.1 System mode

The system has only one normal operation mode.

### 3.7.2 User class

There’re two types of users: ordinary user and administrator.

### 3.7.3 Objects

The system uses such objects: tables, table entries, stored procedures/functions.

## 3.8 Additional comments

The section 3.2 was organized by the user class as proposed in IEEE Std 830-1998 A.3 template.