Database Design Document

Project Enlightenment

Computer Training For Visually Impaired Automation Tool (CTVIAT)

Prepared By: Project Enlightenment

Date: January 4, 2011

Version No: v3.0

Document Change Control

The following is the document control for revisions to this document.

Version Number	Date of Issue	Author(s)	Brief Description of Change
1.0	21/10/10	Project Enlightenment	Phase 1 - Preliminary draft for Phase 1
2.0	24/11/10	Project Enlightenment	Phase 1 & 2 – conform to new framework template, update for Phase 2 deliverables
3.0	04/01/11	Project Enlightenment	Phase 2&3 – conform to new framework template, update for Phase 3 deliverables

1.   INTRODUCTION

In this document, you can find the design of the database of " Computer Training For Visually Impaired Automation Tool". Design decisions that are made during the whole project and the reasons will be explained in the following pages. In the following sections you will find the purpose of this document, system properties, our design approach, trade-offs, interface design standards and tables. Tables include information such as primary keys, foreign keys, referential integrity constraints and normalization decisions that we made.

1.1 Purpose

The purpose of this document is to present the design of the database of " Computer Training For Visually Impaired Automation Tool". In this document you will find the tables that are used in the database and their fields, field types, primary keys, secondary keys and foreign keys that are involved. In the secondary aspect, dependencies among the tables and normalization decisions are explained. We intended to design a secure, efficient, reliable, scalable and compact database.

1.2 System Properties

In this project, we used MS-SQL Server 2008 in order to implement tables, stored procedures primary keys and foreign keys.

1.2.1 Features and Disadvantages

Microsoft SQL Server has the following features such as transaction, selection, insertion, update, stored procedures, triggers, views, referential integrity and foreign keys. The integration with the other Microsoft solutions we will use such as Visual Studio IDE and Windows Presentation Foundation is one of the key factors why we choose Microsoft SQL Server. We preferred SQL Server 2008 since this new version provides a secure, easy and fast way to store and manage data. The main disadvantage of the SQL Server is that it is not an open source system.

1.2.2 Reliability

Microsoft SQL Server 2008 is a stable database 'system. Iis' not an open source system but Microsoft community provides every kind of support. In addition to that every member of the project team is experienced in MS SQL Server and has a positive feedback for the system.

1.3 Design Approach

1.3.1 Investigation

In our design, we have used Requirements Analysis Document and we have contacted with the other group representatives in order to determine the requirements of the database. We updated or changed our design according to the changing requirements. We have tried to update our design efficiently.

1.3.2 Scope

After finishing initial steps of the project, current scope of the design is to implement a efficient database, which can perform fast. We do not have very big tables and we do not use stored procedures. As a next step "Searching" will be more optimal.

1.3.3 Design

We have tried to minimize the usage of join operations in order to improve performance. We preferred not to use stored procedures. We used Visual Studio 2010's database connection and dataset tools which are really fast for implementation and create a reliable database environment. We have also wanted to small tables instead of tables with many columns.

1.3.4 Maintenance

Maintenance of the system will be done by monthly patch controls and system upgrades when needed. Microsoft SQL Server sites will be followed for possible patches and/or advisories on Microsoft SQL Server.

1.4 Interface Design Standards

- Table names start with capital letters and if the table name consists of more than one word other words start with capital letters, too E.g. UserTests
- Field names start with lowercase letters and if the field name consists of more than one word the first letter of the second word is capitalized. E.g UserTests
- We do not use any stored procedures.

2.   TABLES

2.1 Users

2.1.1   Purpose

The purpose of this table is to represent users of the program according to their mode, *admin* or *candidate* through their *userID*

2.1.2   Key Fields

userID is the primary key and it is supposed to be unique.

2.1.3   Referential Integrity Constraints

There are no foreign keys and referential integrity constraints.

2.1.4   Normalization

We tried to make our database design that fits to 3NF. All of the tables are in 3NF form.

2.2 TestsCandidates

2.2.1   Purpose

The purpose of this table is to represent every question of every test that a unique user solved and it stores the answer of the user, too.

2.2.2   Key Fields

candidateID, testID,questionID and *takeOrder* are the keys.

2.2.3   Referential Integrity Constraints

candidateID, testID, questionID must be referenced by *userID, testID* and *questioID* in *User* and *QuestionsTests* tables.

2.2.4   Normalization

We tried to make our database design that fits to 3NF. All of the tables are in 3NF form.

2.3 Tests

2.3.1   Purpose

The purpose of this table is to represent tests that are created by admins and consists of questions that are created by admins.

2.3.2   Key Fields

testID is the primary key and it is supposed to be unique.

2.3.3   Referential Integrity Constraints

There are no foreign keys and referential integrity constraints.

2.3.4   Normalization

We tried to make our database design that fits to 3NF. All of the tables are in 3NF form.

2.4 Questions

2.4.1   Purpose

The purpose of this table is to represent questions created by admins.

2.4.2   Key Fields

questionID is the primary key and it is supposed to be unique.

2.4.3   Referential Integrity Constraints

There are no foreign keys and referential integrity constraints.

2.4.4   Normalization

We tried to make our database design that fits to 3NF. All of the tables are in 3NF form.

2.5 QuestionsTests

2.5.1   Purpose

The purpose of this table is to represent which questions are used in a test by an admin.

2.5.2   Key Fields

testID and *questionID* are the keys.

2.5.3   Referential Integrity Constraints

testID and *questionID* must be referenced by *testID* and *questioID* in *Questions* and *Tests* tables.

2.5.4   Normalization

We tried to make our database design that fits to 3NF. All of the tables are in 3NF form.

2.6 IncompleteTestsCandidates

2.6.1   Purpose

The purpose of this table is to represent incompleted tests of a candidate

2.6.2   Key Fields

candidateID, testID and *takeOrder* are the keys.

2.6.3   Referential Integrity Constraints

candidateID and *testID* must be referenced by *userID* and *testID* in *Users* and *Tests* tables.

2.6.4   Normalization

We tried to make our database design that fits to 3NF. All of the tables are in 3NF form.

2.7 TestsTime

2.7.1   Purpose

The purpose of this table is to represent the total time elapsed for a test of a candidate.

2.7.2   Key Fields

candidateID, testID and *takeOrder* are the keys.

2.7.3   Referential Integrity Constraints

candidateID, takeOrder and *testID* must be referenced by *userID, takeOrder* and *testID* in *TestsCandidates* table.

2.7.4   Normalization

We tried to make our database design that fits to 3NF. All of the tables are in 3NF form.

2.8 TestsTakeOrder

2.8.1   Purpose

The purpose of this table is to represent the take order status for a test of a candidate.

2.8.2   Key Fields

candidateID, testID and *takeOrder* are the keys.

2.8.3   Referential Integrity Constraints

candidateID, takeOrder and *testID* must be referenced by *userID, takeOrder* and *testID* in *TestsCandidates* table.

2.8.4   Normalization

We tried to make our database design that fits to 3NF. All of the tables are in 3NF form.

3.   Design Evaluation

The design and implementation of the database was made by the design subgroup. Also all design decisions were made by them.

However, the design allows the admin to add, update, delete or get the statistics easily, calling some methods from the *Adapters* from dataset tools of the Visual Studio 2010. Since the database type chosen is MS-SQL Server, we created an efficient environment with Visual Studio 2010 and it provided implementation.

Also, there was a discussion on the subject whether using stored procedures but new dataset tools of Visual Studio 2010 were very efficient and we decided to use only adapters of these tools.

4.   Conclusion

This database design document makes use of "Requirements Analysis Document" which is written by the students of CMPE 352. The greatest challenges were those of requirements gathering process when we were trying to understand what the system should really do and this took a lot of time to reach a steady state. During any decision we consulted all the team members as well as the instructors of the course and the supervisor of the project.

5.   References

- Requirement Analysis Document Of The Project
- Database Design Document Examples

6.   Glossary

- **SDD:** A document that formally indicates one or more software components.
- **2NF:** A relation R is in second normal form if and only if it is in 1NF and every nonkey attribute is fully dependent on the primary key.
- **3NF:** A relation R is in third normal form (3NF) if and only if it is in 2NF and every nonkey attribute is nontransitively dependent on the primary key.
- **BCNF:** A relation R is in BoyceCodd normal form if and only if every determinant is a candidate key.
- **Database:** A collection of information organized and presented to serve a specific purpose A computerized database is an updated organized file of machine readable information that is rapidly searched and retrieved by computer.
- **DBMS:** Data base management system
- **Foreign Key:** One or more table attributes that can uniquely identify a record in another table A foreign key is the primary key of another table Foreign keyprimary key relationships define a relational join.
- **MS SQL:** MS SQL is a type of SQL database management tool which is a product of Microsoft and provides a great integration with other Microsoft Solutions.
- **Normalization:** Normalization refers to the process of decomposing base relations using projection to eliminate potential redundancy.
- **Primary Key:** An attribute in an entity that uniquely identifies rows of that entity. For example *the Employee* entity can contain an *empID* attribute that uniquely identifies each employee.
- **Referential Integrity:** It represents real world constraints.
- **Stored Procedure:** Stored procedure is a precompiled program that is stored at the server site It is invoked from the client by a remote procedure call.
- **Trigger:** The activation of an eventdriven process .