System Design Document

Project Enlightenment
Computer Training For Visually Impaired Automation Tool (CTVIAT)

Prepared By: Project Enlightenment
Date: December 21, 2010
Version No: v1.1

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
1.1	21/12/10	Project Enlightenment	Update for Phase 1 deliverables

Definition

The following are definitions of terms, abbreviations and acronyms used in this document.

Term	Definition
CTVI	Computer Training for Visually Impaired
RAD	Requirements Analysis Document
CTVIAT	Computer Training for Visually Impaired Automation Tool
JFW	JAWS for Windows
SDD	Software Design Document
ODD	Object Design Document
CPU	Central Processing Unit in a Computer

Table of Contents

1.GOALS AND TRADE-OFFS

CTVIAT module will interact with two possible user groups: the candidates and the admins. Since the candidate users are assumed to have no previous experience in using the system and to be visually impaired, the system should provide some features.

User-Friendliness

The system should be user-friendly to attract the candidates with proper help mechanisms to do more practise. Since the candidates are impaired, CTVIAT should be an awesome guide, especially in action-based questions because this question type requires the candidates to do some advanced operations which can get them to wrong places. Moreover, the number of errors the customer may face cause of the tool itself, should be minimized as much as possible so that the candidates do not have the concerns about the system and focus their attention to the practising.

Ease of Use

This feature would make the users more enthusiastic while using the tool. There are some tasks, such as the registration of the candidates or preparing of the test, which may be boring for the users. With well-designed interfaces and easy-to-use components, these tedious processes may be converted into a joyful activity.

Reliability

One of the most important features of the system is reliability. There are two aspects of the reliability. First, the information gathered from the customers (private information) must not be revealed under any circumstances. Moreover, the database in which this valuable information is kept should be secure enough.

High Performance

Since the system talks the database server frequently and keeps the tracks of what the user is doing right now, the tool must respond to the user in a short time. Keeping tracks problem can be solved by intelligent algorithms and the computer that has a primary memory that is large enough. Database connection limitation will be overcomed by a proper database design which is

in BCNF, execution of queries in batches and preprepared batch queries which makes query optimization one times which is in prepreparetion.

Minimum Number of Errors

Like any system, the CTVIAT module should be error-free. The problems that the user may face (in solving for candidate and in preparing for admin) and the corner points and the exceptional cases defined in the RAD should be properly handled. Every necessary test and its correction in code must be done.

Security

Security is an important issue in this system. The security of the information kept in the database is vital. No matter how costly, the database must be protected because it is critical for the system to work properly.

Completeness of Functionality

The completeness of functionality is also important. Every function is necessary of make the CTVIAT module efficient, parctical and useful. They should be carefully designed and implemented.

2. SYSTEM DECOMPOSITION

2.1 System Decomposition

CTVIAT project is composed of following subsystems: LOGIN, USER, AUTOMATION and DAO.

The principal subsystem of CTVIAT is the Login since services are served according to the result of this subsystem. Login subsystem is the entrance of the tool so it is the bodyguard. It directly talks to DAO subsystem to authenticate users.

Another subsystem within the CTVIAT module is the User. Purpose of the this subsytem is to serve functionality of the tool to the user with an user-friendly interface. This subsystem is composed of two subsystems, namely Candidate and Admin, since there are two types of user. If Login system authenticates an user, one of the user systems will be loaded.

The subsystem is used by User systems is Automation system. Automation system is the core of the tool and its functionality can change for the user. For admins, system can be used to add new users, to prepare new questions and new test, to query reports for candidates' progress. For

candidates, system can be used to load completely new test or old test, to ask for help in questions and to guide the user what she should practise. This subsytem uses Hint, Rollback, Test Question and Report subsystems in itself to do these varying tasks. For instance, if a candidate requests a help, Automation system loads Hint subsystem to response.

DAO means database access object, so DAO system includes database and classes to access it. This level is required because DAO will be an abstraction of database and provide query optimization and batch mode execution of queries.

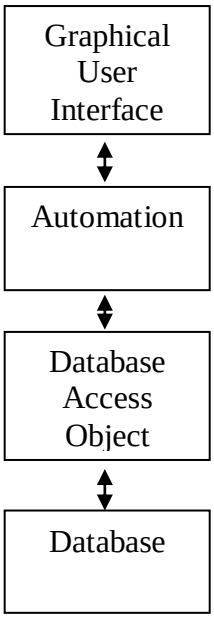
All subsystems requires use of database so does DAO. Main functionality is served by automation but it does not mean anything without user-friendly service.

2.1.1 Layers and Partitions

The CTVIAT module offers a layered architecture with three different layers. The subsystems which include graphical user interfaces, namely Login and User systems, are in the first level. Automation uses three subsystems to serve main functionality so it is the heart of the project and it stays at the mid-level, -second level. Third level is filled by Dao and it is an encapsulation of the database.

2.1.2 System Topology

The CTVIAT Module system can be represented by the following topology:



3. CONCURRENCY IDENTIFICATION

3.1 Independency Between Objects in the Object Model

The division of the CTVIAT module into different subsystems has been designed so that the dependency among the subsystems is at minimum. However, subsystems are dependent. First of all, many of operations performed by systems require read-only access to the database, which does not bring any dependency between objects since read-only accesses can be performed concurrently. Secondly, write access will be detailed below.

Moreover, the subsystem Login is completely independent from User and Automation subsystems because they will work in sequential flow. Firstly, Login authenticates user and sets up the environment and loads required User subsystem, Candidate or Admin. From concurrency point of view, User and Automation subsystems also follow sequential path in themselves although they will work together. Relation between User and Automation subsystems are that User will watch the user and capture the events and then call Automation to process the request.

Dao is the base of all subsystems and will be used by all subsystems. However, as explained above, from the point view of one user, execution path will be sequential, so nothing is needed by Dao to make requests sequential, requests intrinsically are sequential. Furthermore, multi-client will be connect to database server to request both write and read access operations. This also does not cause a problem because each client will read from and write to its own part which is specified by Login subsystem.

3.2 Identifiable Threads of Control

In the system design of CTVIAT module, usage of threads has been employed for the more responsive graphical user interface and utilize of multi-cpu computers of today for higher performance.

User and Automation subsystems will be loaded and work together. The subsystems have their own workers. Dispatcher is the worker of User subsystem and Backgroundworker(BGW) is the thread of the Automation.

User subsystem is mostly composed of graphical user interface. Graphical user interface will be controlled by just one thread which is called Dispatcher and is also main thread of the system. This design schema is chosen because this will be make interface more compact and there will be no need for communication between multiple Dispatchers and Backgroundworker which

will prevent error which can be thrown while accessing graphical elements. When Dispatcher captured an event, it will call BGW to serve. Meanwhile, Dispatcher can respond user requests which will prevent some sort of frozen interface that may cause the user think that there is a problem and the program does not work properly.

Backgroundworker does the job of the Automation subsystem and is created by Dispatcher via requesting from the thread pool. Backgroundworker will also create threads which help Backgroundworker for Dao usage and io operations. Synchronization between Bachgroundworker and its children will be achieved by AutoResetEvent which allows threads to communicate with each other by signalling. One thread signals wait and blocks, the other thread signals set then blocking thread is released. General usage will be that Backgroundworker will start to serve which requires an io operation so it will create a thread to process this io operation and it will go on its steps, when it came to the step needs the result of the io operation, it will signal wait. If the created AutoResetEvent which is created while the child thread is being created is in signalled state, it will take the result and continue to preparing a valuable result for Dispatcher, otherwise it blocks and waits child thread to complete its task.

To sum up, CTVIAT will follow sequential path at communication between subsystems but in subsystems threads will be used for higher performance via proper implementation.

3.3 User Access to the System

Since the CTVIAT module will provide an interface for both candidates and admins from local area network, the program will inherently be multiuser. Local are network access is not actually a limitation because if computers on which the module is running are connected to Internet, client can be used from anywhere.

From the candidate point of view, there will be two main locations of interaction with CTVIAT module. First, the candidates trying to register to the tool will be confronted with responsive and friendly user interface of CTVIAT. During login procedure, data from the database for password will be fetched and compared with data entered by the user. Since this operation requires a read-only access to the database, it can be performed from different points of access simultaneously. Second, during the registration, filling the necessary information is not a problem, because username and password check operations are read-only and final save operation will be handled proper

primary key choice(username). At rest of the path, every candidate will update her own row in each table so multiple access imposes no problems.

To sum up, there is one point of locking database which is registration at candidate view. After registration, each candidate uses her part to update and read-only operations can inherently be executed concurrently.

From admin view, database requires lock because admin operations updates overall database which can cause indetermined state unless proper synchronization is done.

3.4 Availability of Multiple Queires in a Single Query

For better usage of database, batch mode execution of queires will be used. Queries will be preprepared and will be added a batch and will be sent to database to execute.

Query optimization will be done just one time for preprepared queries, otherwise for each query database runs optimizer. Batch mode execution is like paging in memory structure because bigger page means faster io operations and the bigger batch means faster database operations so actually one query is also a batch.

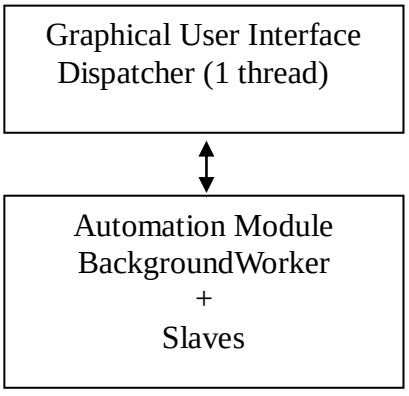
3.5 Parallel Handling of Queries by Different Subsystems

Subsystems execute queries in sequential manner, so this imposes no lock inside of a client.

3.6 Concurrency Scheme

Operations within the client are mostly sequential. However, there are two main threads, namely Dispatcher and Backgroundworker. Dispatcher calls Backgroundworker and returns user to respond. When Backgroundworker is done, it puts the result to buffer of Dispatcher and Dispatcher shows it. Communication between children of Backgroundworker and it is managed by AutoResetEvents.

Multiuser view requires lock of database in admin operations which may update all database in one query.



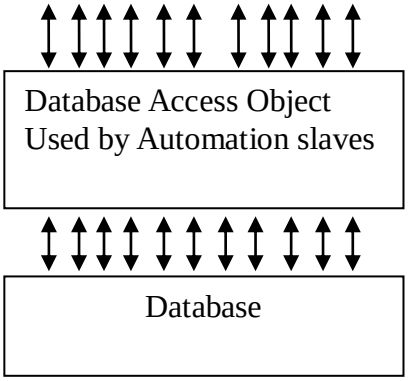


Figure-3.1 General Concurrency Scheme

4. HARDWARE / SOFTWARE ALLOCATION

The CTVIAT module is a desktop application and a client of database server to function. According to place of rhe database server, CTVIAT module may require an Internet connection. If database server is working in LAN, then there is no need overall Internet connection, otherwise needed. There are two types of users to use the module. They can use the tool if it is installed to the computer in which the tool is wanted to run because of its inherent desktop structure.

The hardware allocation of the above described system consists of two main aspects, a closely couple composed of client and server. There is a need of one server to serve database functionality. For client, a computer for mediocre user is sufficient. However, tool is for visually people so for them to communicate the system braille-based keyboard is also required.

CTVIAT module will be designed by and for Windows 7 and .NET technologies. Therefore, at server MS SQL Server will be used and at graphical user interface Windows Presentation Foundation components are used which will reguire .NET 4.0.

4.1 System Performance

The system performance is based on some important measures: information retrieval speed, connection handling performance, data processing capability and memory usage efficiency.

4.1.1 General System Performance

The information retrieval should be as fast as possible for customer satisfaction. To ensure an efficient response time, well-designed database is required. For search queries, selection of primary keys and attributes on which indexing is used, is very important.

Since CTVIAT module platform will work at LAN, which will be a big social foundation, it will have multiple users to access it simultaneously. This multiple user situation will definitely result in a heavy transaction and request traffic. CTVIAT module may be used for examination so database server could be able to serve 100 percent of registered candidates simultaneously. However, there is a remedy because data processing will be in client, server will be used just for data retrieval.

4.1.2 Input / Output Allocation

The CTVIAT module will basically have WPF components as input and outputs. Moreover, it will deal with Word and Excel documents, Internet Explorer connections and Windows Explorer events as input operations. As output, it will use WPF windows but it may use PDF to generate reports of each candidate for admin or mpe3 files to give questions precisely to the candidates.

This allocation is soft. Since expected users are visually impaired, the system will need special hardware for input and output. This special hardware is Braille-based keyboard. However, this is not a bottleneck for system performance. Therefore, again system performance is affected mostly by data retrieval and processing.

4.1.3 Processor Allocation

The system will use multi threads in Automation subsystem because in this subsystem, there will some tasks that can be executed simultaneously. Therefore, managing a multi-processor environment will be a plus.

4.1.4 Memory Allocation

The larger the primary memory, the faster the applications would run. As the system performance is basically related to information retrieval speed and information processing, it would be preferable to have the database server with primary memory in the order of some GB's. Moreover, the size of the secondary storage should be sufficient for input/output files, data swapping, recovery and backup procedures. An acceptable reange for secondary storage should at least cover some tens of GB's.

4.2 Connectivity and Network Architecture

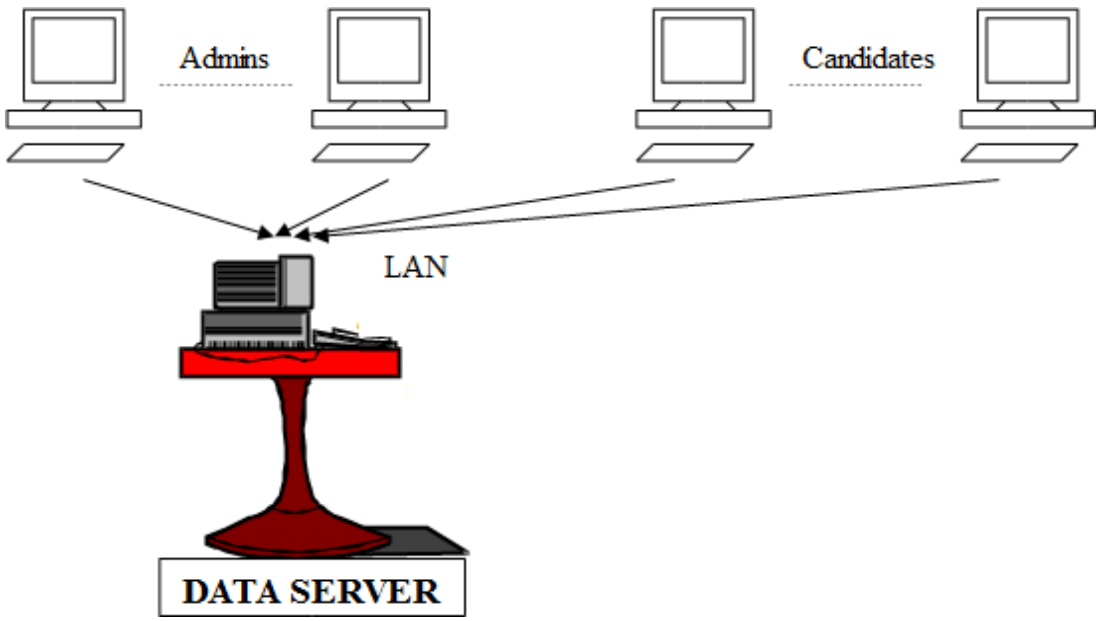


Figure 4.1. Connectivity Diagram

As seen in the connectivity diagram, the two types users, admins and candidates will connect to database server over local area network. However, this schema can be extended to overall IP network with proper design such as a configuration file.

5. DATA MANAGEMENT

The CTVIAT module is the part of Enable India where the information about everything related to candidates is held. One of the main goals of the module is an efficient management of data about examinations, practises and candidates. In order to do that, the CTVIAT module may use main memory, files or databases.

The data that will be held should be volatile as in online games and other online activities, so it cannot be implemented with data structures, using only main memory. To store configuration data and output, files will be employed. However, a simple file would not be sufficient to keep data that will be accessed, modified and erased over and over again. It is more advantageous to apply file structure only when data is needed for a short time, doesn't change frequently and density is low. Therefore, database would be more suitable than a file for the supply of CTVIAT module demands.

5.1 Necessity of Database

Database will be used in order to hold information about hashes of passwords of admins and candidates, questions and answer, examinations and practise tests. Each one of these will form a table of a voluminous database. This will allow multiple users to access at fine levels of details. It is favorable to employ databases whenever data must be ported across multiple platforms in hetegeneous systems as in the case of the CTVIAT. Client parts will work on Windows platform which may be XP, Vista, 7 or over. However, database server can be run on Linux. Furthermore, multiple candidates will also access the data that will held in tables. Since the data management of the CTVIAT module requires a specific infrastructure, database usage is almost an obligation.

5.2 Data Distribution

The data, that the CTVIAT module will hold, will be distributed into different tables of a grand database. Key relations will make access to any data possible. There is no meaning or means of efficiency in holding the main portion of the data in one big table. Modularity will provide more flexibility and higher performance.

5.3 Extensibility of the Database

The database may be thought to be extensible in two different ways. First, database can be selected to be read-only, where one can get data only, or it can be selected as to provide read and write operations, where in the latter case, the database grows meaning that it is extensible. The database that the CTVIAT module will need is going to be extensible in this manner. Since the reason behind using database in this module is to keep information of every candidate that registers, every test solved or exam and of every admin that registers, every question, answer and hint is entered, edited or deleted, every test prepared, it is obvious that a static database would not resolve this problem.

Second, the database can in the format specified during implementation time, or its specification (including field names, new field insertion etc.) can be changed at run-time, which may also be named as extensibility. Extensible databases have a maximum number of fields in tables, where these fields do not have any character (they are dummy values like field 1, field 2 etc.) This is inefficient since SQL does not cover space for empty fields of records. The names of each of these fields are records of another table. By only adding a record to fields table at run-time, a new field is inserted. Unfortunately, this is not the case in the tables of CTVIAT module, since each field is distinct. There is no need for dynamic field allocation.

5.4 Average Request Rate

Assuming to have a maximum of 1000 candidates and admins requesting service in a day, assuming a uniform distribution over an eight-hour period, and assuming candidate will request a question and its answer, hint for this question, the CTVIAT module will have about 36 requests a minute in the average.

5.5 Average Request Size

Assuming a request from the table with the largest number of fields (table questions and it has 11 fields), assuming request deploys each field in the worst case, the request would query all the 11 fields for the worst case.

5.6 Frequency of Database Access

Database is accessed whenever there is a login of admin or candidate, and enter, edit or deletion a question, hint or answer by admin, and addition new candidate, deletion old candidate or check for candidate performance by admin, change in passwords, test preparation by admin, practise of prepared tests by candidate which includes access for questions, answers and hints.

5.7 Query Format and Interface

The query format of the database is SQL, but this format is hidden from the user (both of candidate and admin) with the help of easy-to-use interface. Query is made by clicking mouse and filling text boxes and execute some domain specific commands such as give me a hint for this question.

5.8 Privacy Information Handling

There will be a login at the beginning to get client authenticated and then authentication and authorization information information will be held in the main memory. This information will be send to database server in queries.

5.9 Usage of Relational Database

The database is selected to be relational since it is based on relational algebra. Data is presented as 2-dimensional tables. Tables have a specific number of columns and arbitrary number of rows to dynamically keep records.

Relational database allows uniquely identifying a row in a table with primary key and reference to another primary key in another table (foreign key). Moreover, SQL can be used since it is the standard language defining and manipulating tables in relational databases.

5.10 Selection of Archival Data

User id, user name, user password and user type are stored in the Users table after the registration of the user.

Question id, question type, question text, and if question is objective, 4 options and true answer or if question descriptive, one descriptive answer are stored in Questions table after each question creation.

Test id and test type are stored in Tests table after each test creation.

Test id and question id are stored in QuestionsTests after test creation to map questions to the tests.

Candidate id, test id, question id, answer type, take order and one type answer (objective or descriptive) are stored in the TestsCandidates after one question is answered by the candidate.

Candidate id, test id, take order, question id are stored in the IncompleteTestsCandidates table after each close of the incomplete tests.

5.11 Tables

userID	*userName*	*userPassword*	*userType*

Table-5.1 : Users Table

questionID	*questionType*	*question*	*objectiveChoiceFirst*	*objectiveChoiceSecond*
objectiveChoiceThird	*objectiveChoiceFourth*	*objectiveAnswer*	*descriptiveAnswer*	

Table-5.2 : Questions Table

testID	*testType*

Table-5.3 : TestsTable

testID	*questionID*

Table-5.4 : QuestionsTests Table

testID	*questionID*	*candidateID*	*takeOrder*	*answerType*

objectiveAnswer	*descriptiveAnswer*			

Table-5.5 : TestsCandidates Table

testID	*candidateID*	*takeOrder*	*questionID*

Table-5.6 : IncompleteTestsCandidates Table

6. GLOBAL RESOURCE HANDLING

Since the whole project has some common components that are shared by different modules, there should be some constraints in using them. The following are the list of the global resources and their usage.

6.1 Memory Management

Since the system provides simultaneous services to many different users, the memory must be used efficiently to make the system fast and reliable. The data should not be retrieved from the database at once, since huge amount of data may block the memory, causing different systems to crash.

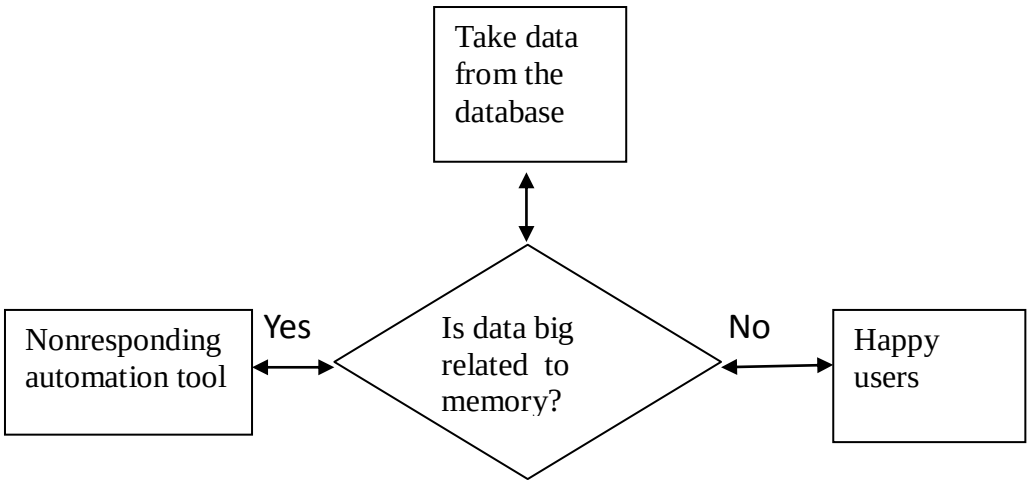


Figure-6.1 : Memory relation to user satisfaction

6.2 Database Management

The CTVIAT module has six main tables, namely users, tests, questions, questions of tests, tests of users and incomplete tests of users. Login subsystem

interacts with users table. Authentication requires just read access and registration requires read and write access. Only admins can update tests, questions, question of tests. Users, Tests of users and incomplete tests of users tables can be updated by candidates, actually the update is started by a candidate activity but transaction is transparent to the candidate.

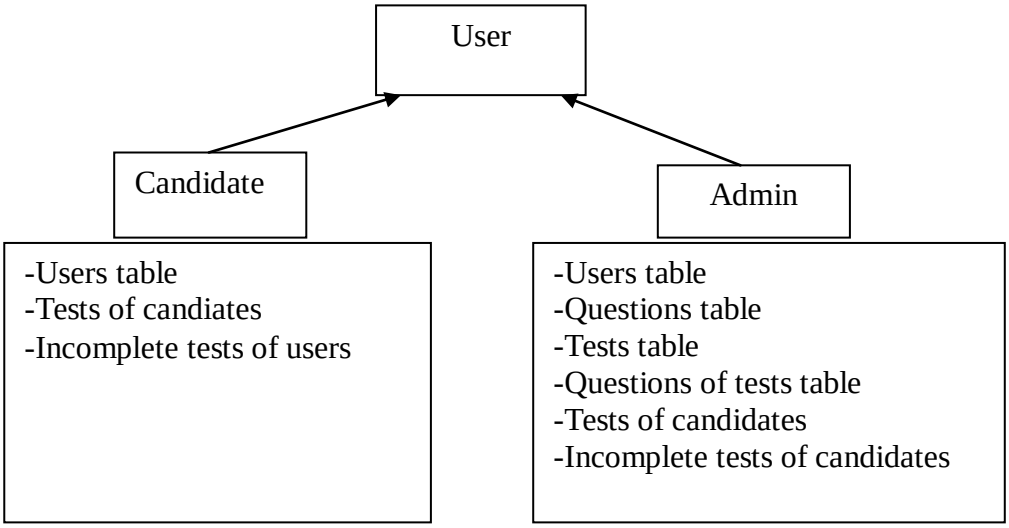


Figure-6.2 : Users, database tables and update permissions

6.3 Authentication and Security Issues

There will be an authentication scheme both for the candidates and the admins. The usernames and hash of passwords will be stored in the users table. After authentication user is not needed to take any further action. Storing hash of passwords in the users table will increase security because if database is revealed or comprimised, passwords would not be revealed since hash function works only one-way. It means original password cannot be produced by using its hash. Hash function can be MD5 or SHA1 which are also implemented by database management system.

Hash	**Supported**
Adler32	-
CRC32	-
GOST	-
MD2	-
MD4	-
MD5	✓

SHA1	✓
SHA256	-
SHA384	-
SHA512	-
RIPEMD128	-
RIPEMD160	-
TIGER128	-
TIGER160	-
TIGER192	-

Table-6.1: Supported hash algorithms to store user passwords

7. SOFTWARE CONTROL IMPLEMENTATION

7.1 External Control Flow(Between Subsystems)

Control flow of the CTVIAT module has the simple characteristics defined by desktop applications. Window keeps track of user via backgroundworkers and captured events and processes the event and produces a response and starts to wait new events. In process phase of the event, Automation subsystem is active which can query database, do file operations or cpu-based operations.

7.2 Concurrent Control

Since the application is Lan based, all subsystems and components can run concurrently for different users in the CTVIAT module. This means each user can use her own copy of the application independent from others. Different copies of module has one relation which is database. Logical design and execution, integrity of database provides different clients to connect simultaneously.

7.3 Internal Control(Within a Single Process)

Process control will mostly be implemented by the designed windows on the desktop. The system is based on the request a service and show response structure. This makes the desired procedures to be simple and mostly linear. However, there can be procedure calls to other subsytems. For instance, user is prompted by login page, user must enter her username and password, actually action that can be taken are chosen by window content/context. Then, there is a procedure call in DAO subsystem to get the user authenticated and learn type of user. Finally, result/response is shown which is candidate or admin main window according to the type of the user.

7.4 User Interface

The user interface of the system will be done through WPF. Control of next steps is up to the user. In addition to this, the flow is implemented within

the WPF windows. Login and User subsystems have different windows and also according to steps, different windows are designed in User subsystems. Automation and Dao subsystems compose the core of CTVIAT module. Due to the event driven design of the system, subsystems cannot be thought to have their own event loop. However, the events are controlled by the WPF windows and are considered to form a global event system.

8. BOUNDARY CONDITIONS

8.1 Initialization

8.1.1 Dynamic Model of the System Startup

The CTVIAT module is an event-driven program working on the LAN without explicit startup and termination. However, there are two points in time when startup is necessary. First is the installation time and the second is the startup after backup operation.

For the CTVIAT module to start working properly, the program should be installed on the computers of the candidates and the database should be installed on the Data Server. Since all subsystems depend on the DAO to retrieve data from Data Server, DAO subsystem should be functioning before other subsystems to use application on the candidate computers, both in the case of installation and after backups. In short, startup of the CTVIAT module requires functioning database.

8.1.2 Description of Data Accessed at Startup

Since the CTVIAT module has operations related with the time and date, these should be gathered at the startup. This is necessary for automation module to generate report to admin for performance of the candidate.

Furthermore, user must be loged on the system to utilize of CTVIAT functionality. Therefore, user should provide an username and a password, then these entered values will be compared via getting user info from users table at startup. Login mechanism is particularly important for the robustness of the program, and keep the application secure from third parties.

8.1.3 User Interface at Startup

Startup requires the main window of the Login subsystem to be displayed for both candidate and admin users. If the user is authenticated, she will be redirected to main window of User subsystem, which is actually composed of two subsystems, namely Candidate and Admin subsystems. Selection of the subsystem is determined according to the type of the user which was taken

from database via DAO subsystem while authentication data is being received.

8.1.4 Presentation of System to the User

The system is presented to the user through WPF windows which come to screen according to the type of the user and steps taken. For example, at startup Login subsystem is loaded and authentication is needed. By entering username and password to edit box, user takes steps. If authentication is successful, new windows go on to be loaded.

Admin windows include more editing components than candidate windows because admins can update database where an update requires a plenty of data. On the other hand, candidates usually will usually query select statements.

Each window is self-explanatory to perform necessary operations so that ensure user-friendness.

8.2 Termination

The CTVIAT module is tried to be designed to be used all the time without interruptions. However, sequantial communication between two subsystems results in dependency. As a result of dependency, if one of the module is down, then whole system can go down. For example, DAO is the only subsystem that can talk to database whereas all subsystems need data from database. Therefore, the most essential part of the system is database and backups will be taken at the time when the user access is statically at its lowest and downtime will not damage the user satisfaction. Therefore, the exact time of backup taking will be performed after the first statistics are taken following the opening of the CTVIAT module on the Lan.

Since CTVIAT module works on the Lan and users has line of sight, users are expected not to use system while backuping.

8.3 Failure

8.3.1 Behaviour of System in Failures

Failures can be classified as internal and external communication failures. External communication failure is a communication failure between Dao subsystem and Data server and may affect the consistency of the database. However, internal communication failure is the failure inside of a subsystem which is shown in deviation from the normal usage and execution path of core subsystems. This failure is a result of programming errors and may also cause external communication failure. In short, internal failure is more important and includes external failure. Internal failure can be produced by unexpected usage

of commands or concurrency. When this type of failure is detected, developer team will come together and resolve thew problem very rapidly. External type failure may be caused by network structure which can be easy issue to address but can take more time to fix.

8.3.2 Availability of Backup Communication Links

The CTVIAT module does not deal with most vital pieces of information such as the credit card number, and the operations do not have as much urgency. However, since exam results will be held in database, backups are needed and should not be delayed much. As a result, no additional backup communication links will be provided but backups will be taken on regular basis, in the least crowded time of the CTVIAT.

8.3.3 Recovery From Failure

Externeal failure can be caused by network structure or database itself. Therefore, recovery from this type of failure will be handled by the Database Management System, which is Microsoft SQL Server. The routines for recovery will be handled by the DBMS and no special handling will be performed by the CTVIAT module.

However, internal failure can be caused by programming error, concurrency or unexpected usage of subsystems. When this failure is detected, developer will prepare patch that fixes the problem.

8.3.4 Differences Between Recovery and Initialization

Recovery is different from initialization, such that after recovery, the consistency of the database has to be checked using the synchronization points in time and logs of database operations, whereas database can immediately start functioning after startup. This contol over the status of the database is performed by the DBMS, Microsoft SQL Server.

9. DESIGN RATIONALE

There are mainly two important design issues to discuss about the CTVIAT module and subsystems of the module.

The first one is the visual part of the CTVIAT consisting of windows by which the users, candidates and admins, will interact with the system. As the CTVIAT is a desktop application and excessively talks to a data server which runs at Lan, it is decided with the accord of the project team that the proposed WPF programming methodology is the best solution among alternatives such as WinForms.

Since platform is Windows, integration with Windows is the most

important criteria that brings two options for desktop applications, namely WPF and WinForms. However, Microsoft advertised that there will be no support for WinForms in upcoming version of Windows. Moreover, WPF is the Microsoft's gun to compete with Flash so Microsoft provides good documentation. Finally, implementation in WPF is simple and projet members are familiar with this framework.

The second part of the design needs a management tool for the database purposes. Since the CTVIAT module will work on Windows platform, for better integration data server should also work on Windows. Therefore, there are mainly two alternatives on the subject. The first one the usage of MS Access and the other one the implementation on SQL Server.The arguments for MS Access were the availability and the familiarity of the software, but to ensure a secure multi-connection handling, the best solution was the choice of the SQL server although its usage will require the installation of the software on the project members' PC's for development purposes. At this point, it will also be worthwhile to mention about some disagreements on database tables and fields. Some module teams have argued that a “username & password” entry in database would cause an extra burden for the designers and a dissatisfaction for the customers. This issue has also been resolved by a positive decision on the problem saying that those entries were crucial to build a security barrier since the functionality of the system involves monetary transactions.

The main tool to achieve a proper software design and development is the Rational Rose case tool whose usage has been proposed by the Project Manager and been accepted by all members due to its visual representation capabilities with UML features.

The system seems to be quite responsive to technological improvements for the moment. WPF is used which supports hardware acceleration. Multi-threading is used to utilize of modern cpus of nowadays. Main performance issues are the efficient handling of database operations and required io operations by questions. Database operations are handled via proper design and indexing, while io operations via multithreading.

For the moment, the biggest problem seems to be the integration of different modules in order to create the compact final system. As the CTVIAT is composed of different modules and departments implemented separately, an efficient coordination between project teams is mandatory. If not resolved, this problem will lead to the overall failure of the project and also individual

modules won't work because of sequantial logical flow.

10. GLOSSARY

Analysis An activity during which developers ensure that the system requirements are correct, complete, consistent, unambiguous, and realistic.

Authentication The process of associating a person with access rights.

Boundary condition A special condition the system must handle. Boundary conditions include startup, shutdown, and exceptions.

Client/server architecture A software architecture in which user interactions are managed by simple client programs and functionality is delivered by a central server program.

Completeness Property of a model indicating whether all relevant phenomena are modeled or not. A model is incomplete if one or more relevant phenomena do not have a corresponding concept in the model.

Component A physical and replaceable part of the system that complies to an interface. Examples of components include class libraries, frameworks and binary programs.

Control flow The sequence of execution of operations in the system.

Customer Relations Management An information industry term for methodologies, software, and usually Internet capabilities that help an enterprise manage customer relationships in an organized way.

Design Rationale *See Rationale*

Error State of the system such that further processing will lead to a failure.

Exception An unexpected event that occurs during the execution of the system.

Extensibility Quality of a system indicating how easily the system can be changed to accommodate new functionality.

Failure Deviation of the observed behaviour from the specified behaviour.

Goal High level principle that is used to guide the project. Goals define the attributes of the system that are important. For a transportation vehicle, safety is a goal. For shrinkwrapped software, low cost is a goal.

Installation An activity during which the system is installed and tested in its operating environment. Installation can also include the training of users.

Layer Subsystem in a hierarchical decomposition. A layer can only depend on lower level layers and has no knowledge of the layers above it.

Login procedure used to get access to an operating system, or application, usually in a remote computer.

RAD *See* Requirements Analysis Document

Rationale The justification of decisions. For example, selecting myDBMS as a database management system is a system design decision. Stating that myDBMS is reliable and responsive enough to attain the project's goals is part of the rationale for this design decision. Also called design rationale

Reliability Property of a yatem indicating the probability that its observed behaviour conforms to the specification of its behaviour.

Requirements Analysis Document A document describing the analysis model.

SDD *See* System Design Document.

Security Property of a system indicating its ability to protect the resources against unauthorized use.

Service A set of related operations offered by a class.

Subsystem In general, a smaller, simpler part of a larger system; in system design, a well defined software component that provides a number of services to other subsystems. Examples of subsystems include storage subsystems (managing subsystem data), user interface subsystems (managing the interaction with the user), networking subsystems (managing the communication with other subsystems over a network).

Subsystem decomposition Division of the system into subsystems. Each subsystem is described in terms of its services during system design and its API during object design. The subsystem decomposition is part of the system

design model.

System Organized set of communicating parts designed for a specific purpose. For example, a car, composed of four wheels, a chassis, a body, and an engine, is designed to transport people. A watch, composed of a battery, a circuit, wheels, and hands, is designed to measure time.

System design An activity during which developers define the system design model, including the design goals of the project and decompose the system into smaller subsystems that can be realized by individual teams. System design also leads to the selection of the hardware/software platform, the persistent data management strategy, the global control flow, the access control policy, and handling of boundary conditions.

System Design Document A document describing the system design model.

System design model High level description of the system, including design goals, subsystem decomposition, hardware/software platform, persistent storage strategy, global control flow, access control policy, and handling of boundary conditions. The system design model represents the strategic decisions made by the architecture team, which allows subsystem teams to work concurrently and cooperate effectively.

Thread Control flow paradigm in which the system creates an arbitrary number of threads to handle an arbitrary number of input channels.

User A role representing the persons who interact directly with the system when accomplishing their work.