

**Ashesi University College**

**Applied Project Report on the Development of  
the Online Examination System**

By

**Jonathan Asiedu Amos**

Dissertation submitted to the Department Computer Science,

Ashesi University College

In partial fulfillment of

Bachelor of Science Degree in Computer Science.

**APRIL 2010**

## **Declaration**

I hereby declare that this dissertation is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature: .....

Candidate's Name: .....

Date: .....

I hereby declare that the preparation and presentation of the thesis were supervised in accordance with the guidelines on supervision of thesis laid down by Ashesi University College.

Supervisor's Signature: .....

Supervisor's Name: .....

Date: .....

## **Acknowledgement**

I would like to acknowledge the following persons for their help and support in giving me resourceful information, tips and ideas on my project. Dr. Nathan Amanquah and Mr. Aelaf Dafla for keeping me on my toe at all times; Dr. Astrid Larssen for always inquiring and showing a keen interest in my project; my colleagues Edem Morny, Kofi Boateng, Erasmus Larbi and Diabene of Genkey Corporation; Mr. Blaine A. Price, Dr. Linda Price and Mr. Pete Thomas of the Open University who are also working on the same project, and gave me resources that helped in the completion of my project. I am very grateful to all of them for their help.

## **Abstract**

Examination writing in Ghana seems to be a location based process. This is so because most examinations taken, be it school exams or other external examinations, are taken at specific selected locations. If exams are to be taken in a school like Ashesi University for instance, there is a time table drawn which includes a course, date, time and place of the exam. The processes that precede it include typing and photocopying of questions which can use up a lot of paper, time and space even for a relatively small school such as this. For larger societies or institutions where space is an issue, examination writing can get very uncomfortable since people are crammed into one examination room. This can even lead to cheating on a large scale with after effects such as examination cancellation, grade cancellation, and etcetera. A lot of this can be reduced by developing computer based online examination software systems for use by the institutions that need them.

## **Table of Contents**

Declaration.....	II
Acknowledgement.....	III
Abstract.....	IV
Table of Contents .....	V
1. Background .....	1
2. Deliverables.....	4
2.1 Considerations in Project: .....	4
2.2 Major Issues to be considered: .....	5
3. Methodology.....	6
3.1 System Requirements and Requirements Specification .....	6
3.1.1 Functional Requirements.....	6
3.1.2 Non-Functional Requirements.....	9
3.1.3 Software Design Specification.....	10
3.2 Interface design.....	11
3.2.1 Assumptions: .....	11
3.2.2 Constraints: .....	11
3.2.3 Goals: .....	11
3.3 Logical database design.....	11
3.3.1 Constraints.....	11
3.3.2 Goals: .....	12
3.3.3 System Architecture Design.....	12
3.3.4 Three-Tier Architecture Considerations: .....	12

3.4 Detailed System (and Interface) Design .....	14
3.4.1 Flow Chart .....	14
3.4.2 User Interface Design .....	14
3.5 Detailed (Logical) Database Design .....	15
3.5.1 Why the need for a database? .....	15
3.5.2 Choice of Database Management System (DBMS) .....	16
3.5.3 Tables .....	18
3.6 Development tools .....	18
1. Languages .....	18
2. Database .....	19
3. Utilities .....	19
4 Evaluation and Results .....	21
4.1 Creating exams and uploading files .....	21
4.2 Timing the exam: .....	21
4.3 User tests: .....	21
4.3.1 Faculty .....	21
4.3.2 Student .....	22
5 Discussion and Conclusion .....	23
5.1 Constraints .....	23
5.2 Other Considerations .....	23
5.2.1 User Security .....	23
5.2.2 Deletions .....	23
6 Appendix .....	25

6.1 Screens.....	25
6.1.1 Screen 1 – User Login Screen.....	25
6.1.3 Screen 3 – Faculty Create Exam Screen.....	27
6.1.4 Screen 4 – Student Register for Exam Screen.....	28
6.1.5 Screen 5 – Student Register for Exam Confirmation Screen	29
6.1.6 Screen 6 – Student Take an Exam Screen .....	30
6.1.7 Screen 7 – Student Take an Exam Screen .....	31
6.1.8 Screen 8 – Student Take an Exam Screen .....	32
6.1.9 Screen 9 – Student Take an Exam Screen .....	33
6.2 – Question Format .....	34
6.3 - Flowchart .....	35
6.4 - UML Diagram .....	36
6.5 – Entity Relationship Diagram.....	37
6.6 – Enhanced Entity Relationship Diagram.....	38
Endnotes .....	39

## **1. Background**

The project being worked on is an Online Examination System. The motivation to do this came about as a result of information received that the Driver Vehicle and Licensing Authority (DVLA) of Ghana were computerizing their systems to facilitate faster transactions and processing. One of their intents was to develop an online examination system so that people could sit at home and write the theory section of the driving test. This was the birth of the idea for this project. Contact with the authorities at the DVLA was established to obtain more information which enabled for work to begin on designs and strategies for implementation.

Most of the background work done was based on observations of other types of examinations, data collected from the internet, information from other universities trying to explore this same technology and personal experiences. Data obtained from the research conducted led to the realization of some of the things that most examination systems lacked. This enabled the drawing up of necessary requirements for developing a solution easier and more focused.

The issues involved in examination writing in a country like Ghana include the use of a lot of paper, time and space. Due to the small spaces, people get crammed into exam halls making it generally uncomfortable, and making it easier for cheating to occur. These are issues which have been encountered by many other individuals and/or institutions and thus many of them have tried to find ways and means of solving this problem. The main idea is to enable people sit in the comfort of their homes and take examinations at their allocated times. One way by which many institutions

solve this problem (especially schools, thus it does not necessarily entail sitting at home, but a good attempt) is by deploying examinations over a local network.

Some lecturers in Ashesi University do this by setting questions and uploading them to a web server on the network, and allowing the students to login using their school usernames and passwords to retrieve and answer these questions.

Some other examinations like the CCNA (Cisco Certified Network Administrator) and the MCTS (Microsoft Certified Technology Specialist) tend to be standalone applications which are executed and have timers for starting and ending the exam, but these examinations are usually taken at allocated/designated locations which are away from the comfort of the home.

Google on the other hand uses the internet to host their CodeJam competitions which in this case allows users to actually sit in the comfort of their homes and participate. They do this by allowing participants to download the question files and upload their answers when they are done answering the questions. The competition is timed and as such a timer is started from when a question is downloaded to when the answers are uploaded back onto their servers.

From the different types of examinations above, some of the findings were that:

- a. Cheating could not be monitored in examinations where an invigilator was not present. If exams were to be taken online in the comfort a person's home (like that of Google), and the person was reading the answers off a book or being taught by another person, those activities could not be monitored. For the examination types stated above, in most Ghanaian

institutions, one would have physical invigilators monitoring the examinations. The use of human beings as invigilators can also raise concerns.

- i. Human personnel need to be paid and as such tend to incur extra cost.
  - ii. Human personnel can be bribed, blackmailed or forced against their will to allow people cheat during examinations.
- b. Internet speeds are relative in different places of the globe. This meant that if a system was to be built, it did not have to solely rely on internet connectivity in order to work properly. It did also mean that if it the system was to rely on the internet as the main mode of examination writing, then an attempt should be made to make whatever data was being transferred across the networks as little or as light as possible.
- c. A major concern was security. The issue of data being intercepted by people such that they could see the questions before they were actually delivered to the recipient was something to be dealt with. If examinations were deployed on local networks, this wouldn't be much of an issue but the deployment of examinations over the internet is something else altogether. This meant that measures such as security certification and encryption had to be introduced to at least make it difficult for hackers to decrypt whatever information was being sent across.

The above are a few points that I used to draw up my requirements. The intention of this project is to attempt the development of an online examination system which will address these issues.

## **2. Deliverables**

The problem at hand involves taking of examinations online while finding ways of preventing cheating, and making the exam work without lag regardless of the network connectivity and its speed. The aim of this project is to develop a functional online examination system to enable an individual sit in his/her home and take a written online examination regardless of internet speed, while being monitored remotely by systems set up by the examinee as a means of preventing cheating.

### **2.1 Considerations in Project:**

- a. Application to deliver most question types: The application should be able to deliver multiple-choice like and drag-drop question types.
- b. Correct user and question management: The users of the system must be who they say they are. Examinees should not have the ability to log in as their examiners to view or edit questions. If an examinee logs in, they should be noted as such.
- c. Database optimization: The examination application will have a lot of users i.e. both examiners and examinees. It would also have a lot of questions and possible answers stored in it. This would mean that if for some reason all the users were taking an exam at the same time, the database should be able to handle everything with little or no lag at all thus the need for optimization techniques.
- d. Communication: Less server side and more client side communication will be adopted. This will enable the client application perform faster. If transactions were to go to the server and back all the time, the application

would react very slowly, even worse on slow network connections. If the information needed is brought to the client side, then everything will be done there even if the network fails. This will enable the client be more effective and make the application more responsive.

**2.2 Major Issues to be considered:**

- a. Security
- b. Client and server side technologies to use.
- c. Timing of the exam
- d. Speed of application from client point of view
- e. Database management
- f. Database optimization

### **3. Methodology**

In solving the problem of examination writing, there needs to be requirements and designs drawn in order to get a good grasp of the challenges that will be faced, and enable us zero in on the solutions to the problem. This chapter will thus focus on the project planning, functional and non-functional requirements gathering and analysis, design specifications, implementation of the application and its integration. It will also look at the development tools and type of system architecture employed

#### **3.1 System Requirements and Requirements Specification**

##### **3.1.1 Functional Requirements**

A functional requirement is a requirement that specifies a function that a component or system must perform [1]. They capture the intended behavior of the system including inputs, outputs, calculations, external interfaces, communications, and special management information needs. Functional requirements are also called behavioral requirements because they address what the system does [2]. This behavior may be expressed as services, tasks or functions the system is required to perform [3]. There are some tasks that the system should perform and are thus presented below.

Functions the system must perform:

###### **3.1.1.1 Operations**

Users of the system should be able to do the following operations based on their authority/permissions:

- i. Upload exam: This option or operation would be useful so that a faculty member who is part of the system can upload an examination to be taken by students of that course.
- ii. Retrieve results: The results of the students should be sent to the examiner after an examination. This should be done to enable grading.
- iii. Get examination report: If necessary, an examination report should be sent to the examiner with information such as the number of students who took the exam, cheating and impersonation information if any, and so on.
- iv. Register for exam: A student should be allowed to register for an exam which has been put up by a lecturer. The student should be allowed to register for more than one examination if necessary and if they have been put up by the lecturer.
- v. Take an exam: The student should be able to log in to the system from anywhere and take the examination. The examination should be one which was pre-registered by the student.
- vi. View grades: The students should be allowed to view their grades for a particular examination and/or all of the examinations.

#### *3.1.1.2 User Management*

User information tends to be very dicey. If user information should into the wrong hands, it could be misused and as such, the correct users should be able to access the appropriate content necessary to them.

- i. Restrictions should be made on which users can access what content. For example, a student should not be allowed to upload examinations

or retrieve the examination report since this is the role of the faculty member.

- ii. Sessions [4] should be used to enable the system and the user know who exactly is logged in and working on the system. This session management should cater for the fact that some users will write their examinations using un-trusted networks such as connecting to the internet through someone's computer, or the use of public computers where browsing history and sessions are not deleted automatically from the browser. The application should encrypt delicate information such as usernames and passwords, clear all sessions and session variables after they have been used, and make sure to leave no traces of delicate information.

#### *3.1.1.3 Check Impersonation*

- i. The system is to allow a user write an examination once and only once, unless the user is administratively allowed to retake the exam for any reason.
- ii. This is one step to enable the system know whether the right person wrote the examination or not. If an impersonator wrote an exam unknowing to the right person, if a report is made, the impersonator can be traced and whatever sanctions necessary are applied.
- iii. The system should have an embedded camera function used for viewing the examinee live so that we can know that the right person is taking the exam. This will also serve as means of eliminating impersonation.

### **3.1.2 Non-Functional Requirements**

A non-functional requirement is a requirement that does not relate to functionality, but to attributes of the system such as reliability, efficiency, usability, maintainability and portability [5]. As described by Ruth Malan and Dana Bredemeyer of Bredemeyer Consulting, These properties, emerge from the combination of the parts of the components. They go on to explain that these emergent properties come about as a matter of accident, not design i.e. if the system qualities are not specified in advance [6].

#### **Non-functional Requirements of the system:**

##### **3.1.2.1 Browser Display**

The system should display and work well on all the major browsers that exist.

##### **3.1.2.2 Speed**

The system should be fast especially with operations such as the main examination writing module. Each exam has a specified duration and as such during the examination writing, moving from question to question should not be slow regardless of the internet connection speed, System Memory (RAM) size, processor speed and so on. Slow page navigation would defeat the purpose of the examination module. The idea is for the examination is to be as seamless as possible like it were being written on paper.

##### **3.1.2.3 Availability**

The system should always be available. The idea of a web based system is to enable any individual with internet access to access the system from

anywhere in the world at any time through any medium and have the required content available.

#### 3.1.2.4 Client protection

The system should have measures against issues such as power fluctuations, computer hacks, virus attacks and so on. Users of the system should be made aware of using equipment such as voltage stabilizers to protect their computers. This will be made known to them before they take the examination. Encoding and encryption will be implemented as an attempt to secure privileged information and deter hackers who would want to sniff and steal data from the network. The users should also be asked to protect themselves with antivirus, antispymware and firewall applications. Other protective measures on the user side might include data backups, use of Redundancy disks (RAID) in storing data, and so on.

### **3.1.3 Software Design Specification**

This part of the document is a presentation of the Software Design Specification for the 'Online Examination System'. The purpose of a Software Design Specification (SDS) is to design the software that is to meet the functional requirements for the project. It is the stage at which the supplier specifies the detailed design of the software system, produces the program code to realize that design, tests the individual programs and integrates them into the complete software system [7].

The functional requirements of the project have already been defined in the above sections. From these functional requirements, the design specifications that are needed for the project to progress have been drawn.

## **3.2 Interface design**

### **3.2.1 Assumptions:**

Every user of the system should have a role/privilege. No user can be created without a privilege.

### **3.2.2 Constraints:**

This main examination module is constrained to the desktop, but the rest of the application is optimized for both desktop and mobile platforms.

### **3.2.3 Goals:**

- i. Simplicity of design and user interface elements.
- ii. A user should have Knowledge of his/her exact location in the application.
- iii. Speed of sending and retrieving data has to be very good, close to real-time.

## **3.3 Logical database design**

The database would run on a database server.

### **3.3.1 Constraints**

1. Slow network: This may not cause the database to perform at peak performance.
2. Database detail: There is a possibility that certain high level details of the database may not be captured and as such it may lack some richness in detail.
3. Anomalies: Not all anomalies can be identified and removed. Most of them may be removed, but not necessarily all of them.

### 3.3.2 Goals:

1. Normalization of the various entities to reduce the chance of anomalies occurring in our database.
2. Primary key, foreign key, indexing and relationship definition to increase optimization of the database.

### 3.3.3 System Architecture Design

The software system will be implemented based on the model of the Three-Tier Architecture. 3-Tier client-server architectures have 3 essential components:

1. A Client PC(s)
2. An Application Server
3. A Database Server

### 3.3.4 Three-Tier Architecture Considerations:

- Client program contains presentation logic only
  - Less resources needed for client workstation
  - No client modification if database location changes
  - Less code to distribute to client workstations
- One server handles many client requests
  - More resources available for server program
  - Reduces data traffic on the network

The other architectures that exist like the one-tier architecture where as an example, dumb terminals are connected to mainframe computers; and the two-tier architecture where the client computer/program accesses the database directly tend to be either extremely database dependent (as in the case of the mainframes) or extremely client side dependent (as in the case of the two-tier architecture).

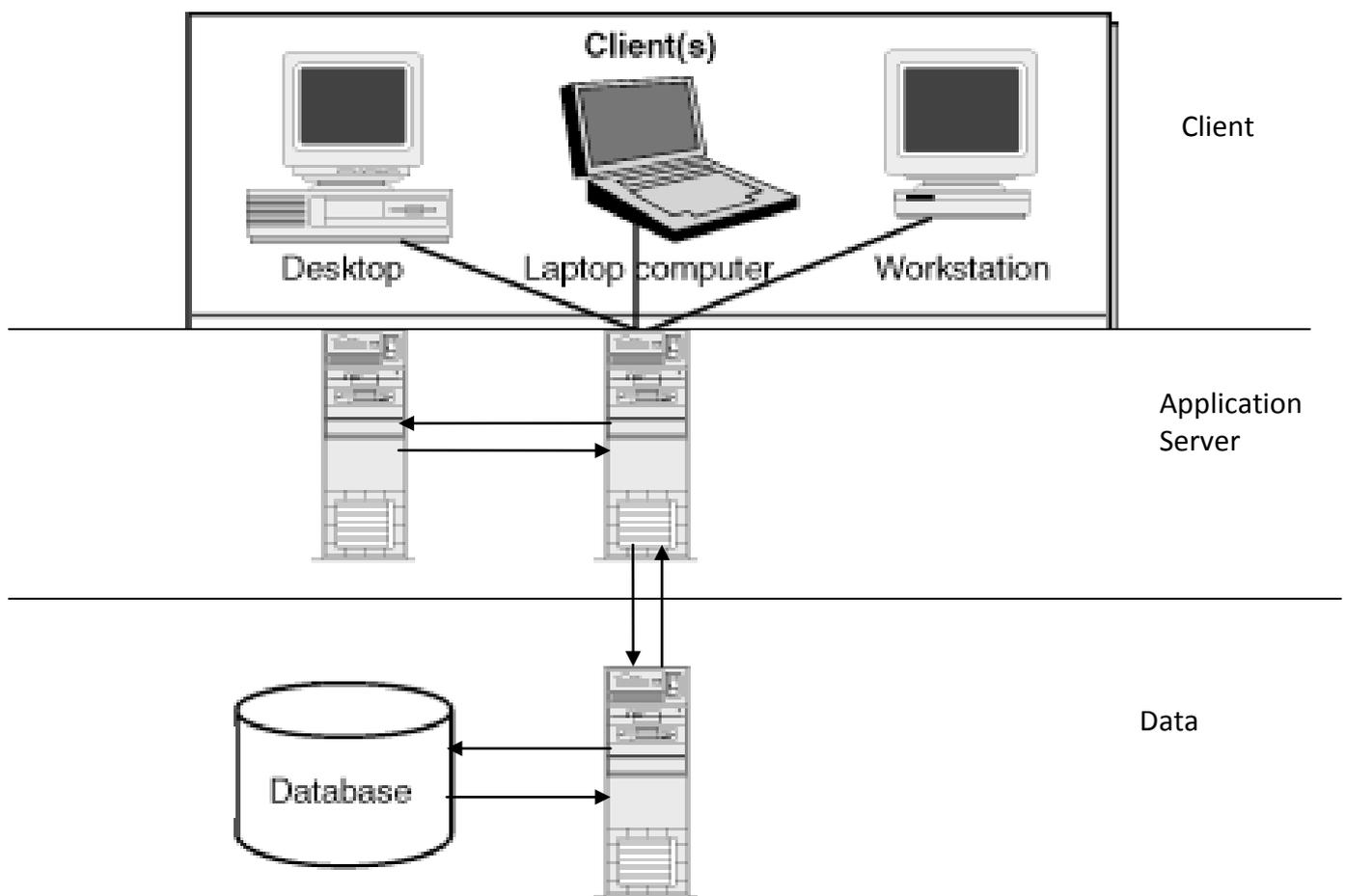


Figure1: Model of Three-Tier Architecture of the processes of the 'Online Examination System'. Image Sources: <http://www.webbasedprogramming.com/Java-Unleashed-Second-Edition/f25-3.gif>

Either ways, the addition of the third layer brings about efficiency. The application server would enable the database perform its key function as the data store and respond to queries when necessary. It would also enable the

client to serve its main purpose of data presentation to the user. The use of the three-tier architecture thus enables the key functions, processes and business logic to be performed on the application server where it has the resources it needs to function properly.

### **3.4 Detailed System (and Interface) Design**

The application would be based on a school scenario where the examiners would be lecturers and the examinees would be the students of a school. This scenario is to enable the design throw more light on specific details of the application.

#### **3.4.1 Flow Chart**

The flowchart is a means of visually presenting the flow of data through an information processing system, the operations performed within the system and the sequence in which they are performed [8]. It can also be defined as a graphical representation for the definition, analysis or solution of a problem in which symbols are used to represent operations, data flow and so on [9].

The flowchart shows the flow of events of the Online Examination System, and can be found in the appendix.

#### **3.4.2 User Interface Design**

The design of the user interface was done by taking each privilege or role i.e. the student role and the faculty role, and creating a screen for each

of them. Each of these users has different privileges and different roles to play. The various screens that therefore exist are:

### **Login Page**

Page for the user to login

### **Faculty Main Page**

Exam creation and upload page

Results retrieval page

Examination report page

### **Student Main Page**

Register for exam page

Examination confirmation and key retrieval page

Take exam page

Exam Information Page

Exam rules and regulations page

Main Exam module

## **3.5 Detailed (Logical) Database Design**

### **3.5.1 Why the need for a database?**

The Online Examination System can be described to be in itself more a less a database application. The main operations that are carried out rely extensively on the availability of data which is either yet to be stored in one location, or already stored in that location. Modern trends have shown that most reliable forms of data storage depend on a database of a sort. These

are evident even from data storage giants such as Google, Yahoo and MSN (including Bing). All these entities use Databases in their daily transactions and still use it till this very day because it works for them.

In relation to data storage for the Examination System, the Software, also being data dependent, and seeking a need to be secure, fast, efficient and reliable, uses a database. The majority of operations carried out involve data insertion and data retrieval. These are all operations that are best maintained by a database. A company such as Google serves as one of the best examples when it comes to databases and their usefulness. With reference to their CodeJam competition, the applicants of the competition, the questions that are asked, and the results of the examination are all stored in a database. Evidence of this is seen in the redirection of their servers when accessing their portal from different countries. Accessing Google services from Ghana will redirect to [www.google.com.gh](http://www.google.com.gh). Accessing from Germany will redirect to [www.google.de](http://www.google.de). These databases are optimized for quick retrieval and insertion of data. These are the same kind of processes that need to be implemented, and as such it is necessary to use one.

### 3.5.2 Choice of Database Management System (DBMS)

There are many reasons for which the database selected was used. One of the reasons is that the software application being developed is implemented in the PHP programming language. According to the developers of applications of this platform, the speed of the PHP server is optimized when used in unison with MySQL as its DBMS. Other DBMS's could have been

used including Microsoft SQL, postGre SQL, Oracle and so on, but the level of integration that can be issued when PHP is used with MySQL makes it a good choice for use as the DBMS.

The commercial Enterprise version has no up-front licensing costs, which means lower capital outlay. It is much lower especially when compared to high-end proprietary databases like Microsoft and Oracle, who have a per-CPU license cost anywhere from \$25,000 to \$40,000. Support for the Enterprise Edition also costs less, leading to savings of up to \$5000 annually.

Hardware requirements for MySQL are also very modest and as such can be run a variety of systems. The one caution however is to ensure the availability of to enable data processing to be fast and perform at its peak.

MySQL is also very well supported with a dual-licensing scheme that provides access to support directly from the creators of the database. MySQL AB provides support and maintenance services as part of it's roughly \$3000 annual subscription fee. A separate fee gives customers priority 24 hours a day, seven days a week support.

The scalable nature of MySQL also makes it possible to easily adapt the system to the current needs of the Online Examination System. The system can be run from the very large systems to very small systems. This scalability is achieved through feature called "stored procedures" which represents mini, precompiled routines that reside outside of the application.

MySQL also comes prepared to support a range of languages such as Ruby, JavaScript and PHP.

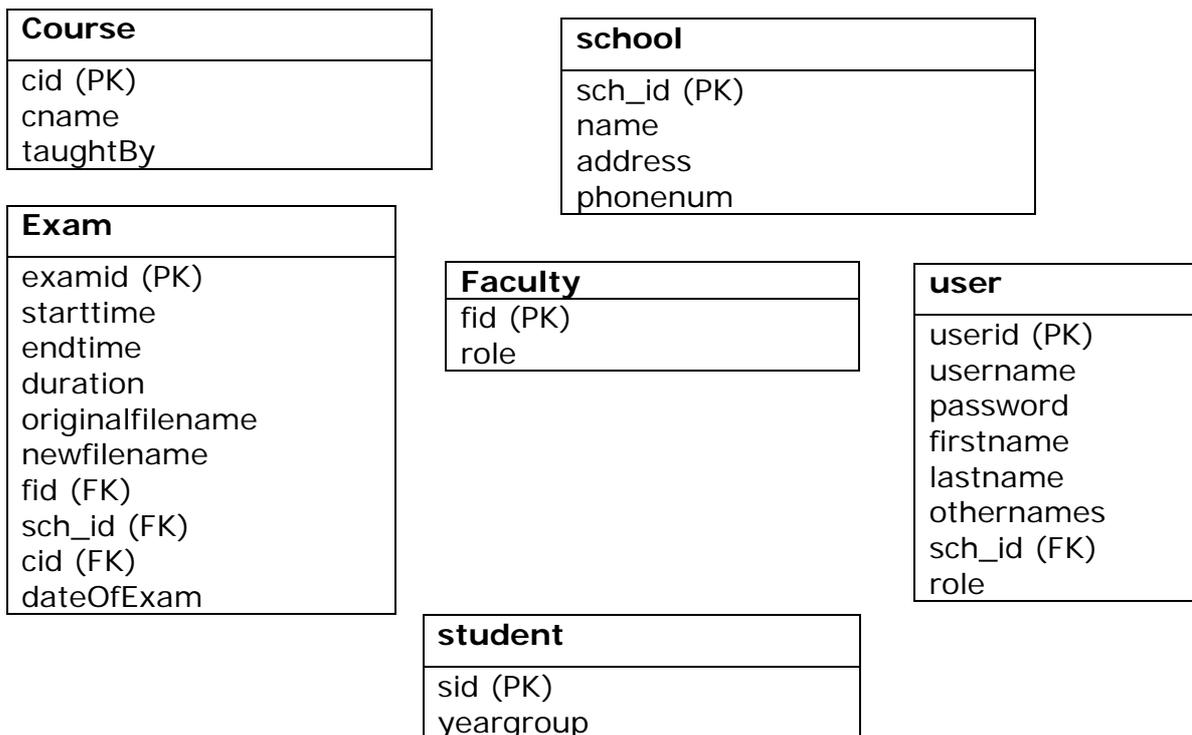
All these features within this comparably inexpensive software make it ideal to run the application on.

### 3.5.3 Tables

From the functional requirements and design specifications, relationships were drawn which were used to build the database. The tables that evolved are:

- a. course
- b. exam
- c. faculty
- d. school
- f. student
- g. user

Each of the tables above have their own attributes. The diagrams below show the tables and their attributes as it is in the database.



See appendix for UML, Entity Relationship and Enhanced Entity Relationship Diagrams.

### **3.6 Development tools**

#### 1. Languages

PHP, JavaScript, HTML 4.0, CSS, MXML, XML

#### 2. Database

MySQL

#### 3. Utilities

Apache web server, Notepad++, PhpMyAdmin, Sqlyog, Microsoft Paint, Adobe Flex, Builder CS4, Adobe Dreamweaver CS4, Flash enabled web browser, Microsoft Office Visio 2007.

The various development tools listed above were used in the design and implementation of the Online Examination System. These tools served various purposes in the project. The PHP, JavaScript, HTML and CSS languages were used in developing the modules before the main examination module. They were used to create pages such as the faculty exam upload page, student exam registration page, the student take exam page and the login page. For the main exam module, MXML, XML and JavaScript were used extensively.

The application also connected to the database when necessary to query it for information, insert information, update information, or remove information from the database. The application connected to the MySQL database to do that.

The Apache Web Server served as the applications web server. All drawings and images including the flowchart and the screens were

implemented using Microsoft Paint and Microsoft Office Visio. For the code writing, Notepad++, Adobe Flex, Builder CS4 and Adobe Dreamweaver CS4 were used. These provided tools for making code writing easier than traditional code writing approaches. PhpMyAdmin and SQLYog were used to manage and manipulate the database. Finally, the Flash enabled browser was used for testing purposes. After every module was developed, it was tested on the browser to see how it functioned.

|

## **4 Evaluation and Results**

Many results were obtained during the testing and validation phase of the project. Testing in particular covered issues such as uploading exam files (i.e. files which contain exam questions), creating an exam, registering for an exam and taking an exam. These were the modules that were completely done and fully functional.

### *4.1 Creating exams and uploading files*

This module worked very well. The requirement for this to work well was that the lecturer selected had to match with the course that he/she taught. If the lecturer had a mismatched course, the exam creation process would not work.

### *4.2 Timing the exam:*

Secondly, setting the time for the duration of the exam did not work as planned. This field was thus left blank.

### *4.3 User tests:*

Testing was done for the different types of users of the systems i.e. faculty and students. The results are as follows.

#### **4.3.1 Faculty**

- i. The faculty members were able to login.
- ii. The examination questions had to be set in a particular format. If this was done, they were able to upload the examination for each course they were teaching.

#### 4.3.2 Student

- i. The students were able to login.
- ii. They were able to register for an examination they wanted to take.
- iii. They were able to take their examinations.
- iv. They were able to see their grades immediately.
- v. They were not allowed to reset and retake the examination unless administratively allowed to do so.
- vi. One person was allowed to log into the system more than once, but was not allowed to have that one person take the exam at two different locations at the same time.
- vii. The embedded camera function was included in the system. The camera though just existed but neither took snapshots nor video content.

From the above, the results that were obtained from the system can be said to be quite satisfactory. More work may need to go into this project to bring out a very good solution. More research into the area may be required, and more time to work on the different components would also be necessary if better results are to be expected.

|

## **5 Discussion and Conclusion**

This chapter presents final discussions and conclusion to the Online Examination System project documentation. This includes the constraints that were included in the project, and other considerations of the project.

### **5.1 Constraints**

Some of the tables had foreign keys. These Foreign Keys were referenced from their base tables and thus if the values were not existent in the base tables, then it would be incorrect for them to exist as Foreign Key values in other tables. In the application, the tables that have the Foreign keys therefore have their values deleted and/or updated when a delete or update operation occurs on the base table.

### **5.2 Other Considerations**

#### **5.2.1 User Security**

The database contained a table called `user` which had information on the users of the system. This table took in information such as the full name, username, and password. The application was designed to encrypt this information using the md5 hashing function before storing it the hashed information into the database.

#### **5.2.2 Deletions**

The effects of deletion on information, especially delicate information can be very disastrous if done without right permissions, or if done accidentally. This issue was of great concern to me, so I decided that an extra field be added to all the tables so that from the application, if a deletion

occurs, the data will not actually be deleted, but will have the extra field set to yes so that if a search query is executed, that particular row of information is not sent back to the user so that it looks as though it has been deleted. This will enable the administrator recover supposedly deleted information.

The ability to sit home in a comfortable environment and write examinations like it was being written and invigilated in an exam room or hall is definitely not impossible. This project has a lot of potential benefits and should be carried on further.

## 6 Appendix

### 6.1 Screens

#### 6.1.1 Screen 1 – User Login Screen



Figure 2: Online Examination System Login Screen.

This is the main login screen of the application. Every user of the system has to go through this secure login process before being allowed to use the application. If incorrect details are entered, an error message is displayed, and the user stays on the login page until he/she is able to provide correct login details.

## 6.1.2 Screen 2 – Faculty Activity Selection Screen

On this screen, we use the example of a Faculty as our user who has logged in correctly. The Faculty has the system privileges to add an exam, retrieve the results of an exam, and get the examination report.

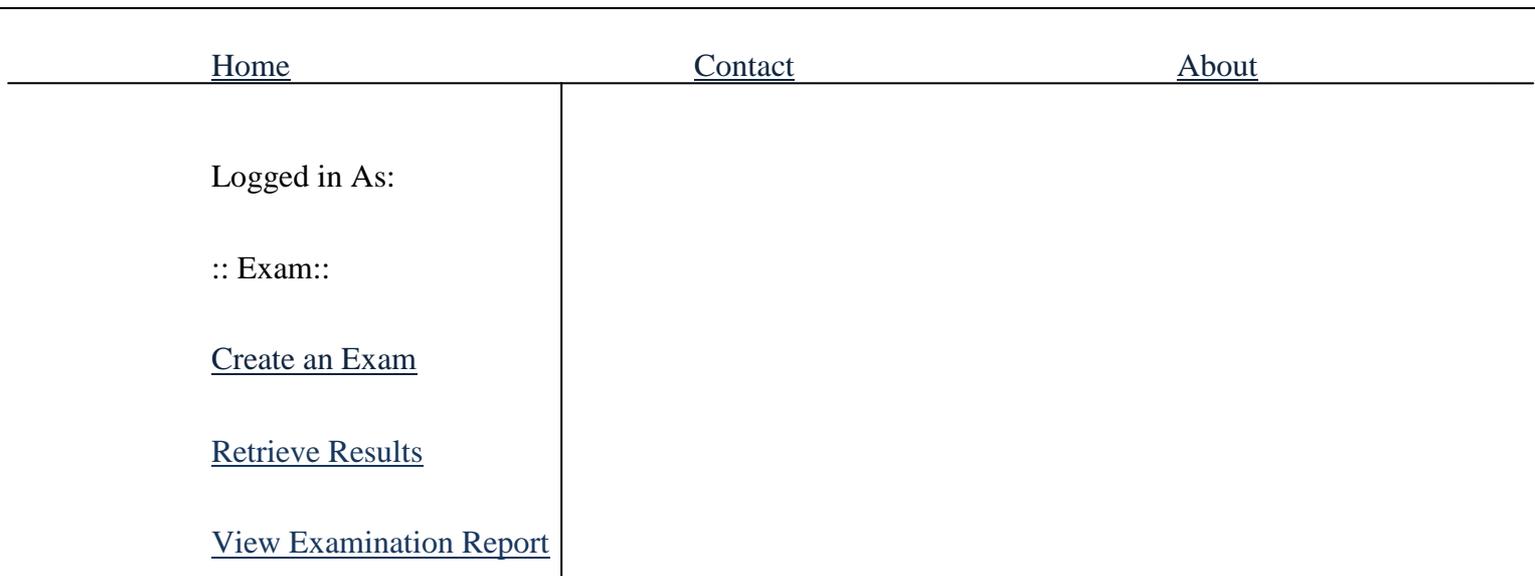


Figure 3: Screen after faculty login

\* The mobile version of the application does not use frames as the web version does, but contains all the links that are seen here.

### 6.1.3 Screen 3 – Faculty Create Exam Screen

<a href="#">Home</a>	<a href="#">Contact</a>	<a href="#">About</a>
----------------------	-------------------------	-----------------------

---

Logged in As:	Course	<input type="text"/>
:: Exam::	Date of Exam	<input type="text" value="12/01/2012"/>
<b><u><a href="#">Create an Exam</a></u></b>	Start Time	<input type="text" value="0"/> <input type="text" value="0"/>
<u><a href="#">Retrieve Results</a></u>	End Time	<input type="text" value="0"/> <input type="text" value="0"/>
<u><a href="#">View Examination Report</a></u>	Duration	<input type="text"/>

**Click browse, go to the location of the exam file and select it.**

Filename:

Figure 4: Clicking on create exam brings up a course select box, date of exam input box, start time and end time drop downs, duration text field, and a browse button to browse for the question file. Clicking finish creates the exam and uploads the exam file to the server.

## 6.1.4 Screen 4 – Student Register for Exam Screen

[Home](#)                      [Contact](#)                      [About](#)

---

Logged in As:  
:: Exam::  
**[Register for an Exam](#)**  
[Take an Exam](#)  
[View Grades](#)  
[View Progress](#)

Select Faculty :

Select Year :

Select Course :

Figure 5: Clicking on register for an exam brings up a faculty, year and course select box. A confirm button also exists. Clicking 'Confirm' sends the user to a confirmation page.

From figure 5 and figure 6, the screens show that there needs to be a registration for an exam. For the school scenario used, and with Ashesi University as an example, the lecturers are given a list of the names of students expected to take their class/course for a particular semester. They can use this list to register their students so that the students do not have to register manually (as we have in this case). But there are cases where the numbers of students are large, and they keep taking up and dropping courses. It would not be wise in such a case to use the list to register the student since at the time of the examination, it might be outdated (due to taking up and dropping of courses). Therefore, the students themselves

## 6.1.5 Screen 5 – Student Register for Exam Confirmation Screen

<a href="#">Home</a>	<a href="#">Contact</a>	<a href="#">About</a>						
Logged in As: :: Exam:: <b><u>Register for an Exam</u></b> <a href="#">Take an Exam</a> <a href="#">View Grades</a> <a href="#">View Progress</a>	<table border="1"><tr><td>Faculty</td><td>Ato Yawson</td></tr><tr><td>Year</td><td>2010</td></tr><tr><td>Course</td><td>Calculus</td></tr></table> Examination code: <input type="text"/>	Faculty	Ato Yawson	Year	2010	Course	Calculus	
Faculty	Ato Yawson							
Year	2010							
Course	Calculus							
<b>*Please keep Examination code for future reference. You will be asked to provide it to take the exam.*</b>								

Figure 6: This page contains the details entered on the previous page. It also has an 8 digit examination code which must be copied from the text field and kept to write the exam.

should be made to register for the exams that they are taking. This will also serve as a good reminder to them of their pending examination. When a student registers with correct details, he/she is given an examination code which will be required before they can write the examination.

## 6.1.6 Screen 6 – Student Take an Exam Screen

<a href="#">Home</a>	<a href="#">Contact</a>	<a href="#">About</a>
<p>Logged in As:</p> <p>:: Exam::</p> <p><a href="#">Register for an Exam</a></p> <p><b><a href="#">Take an Exam</a></b></p> <p><a href="#">View Grades</a></p> <p><a href="#">View Progress</a></p>	<p>Hello Ato Yawson,</p> <p>Please enter your 8 digit Examination Code into the box below to proceed with the examination.</p> <input type="text"/> <input type="button" value="Proceed"/>	

Figure 7: Clicking Take an Exam prompts the user to enter their 8 digit Examination code before they are allowed to write the examination.



## 6.1.8 Screen 8 – Student Take an Exam Screen

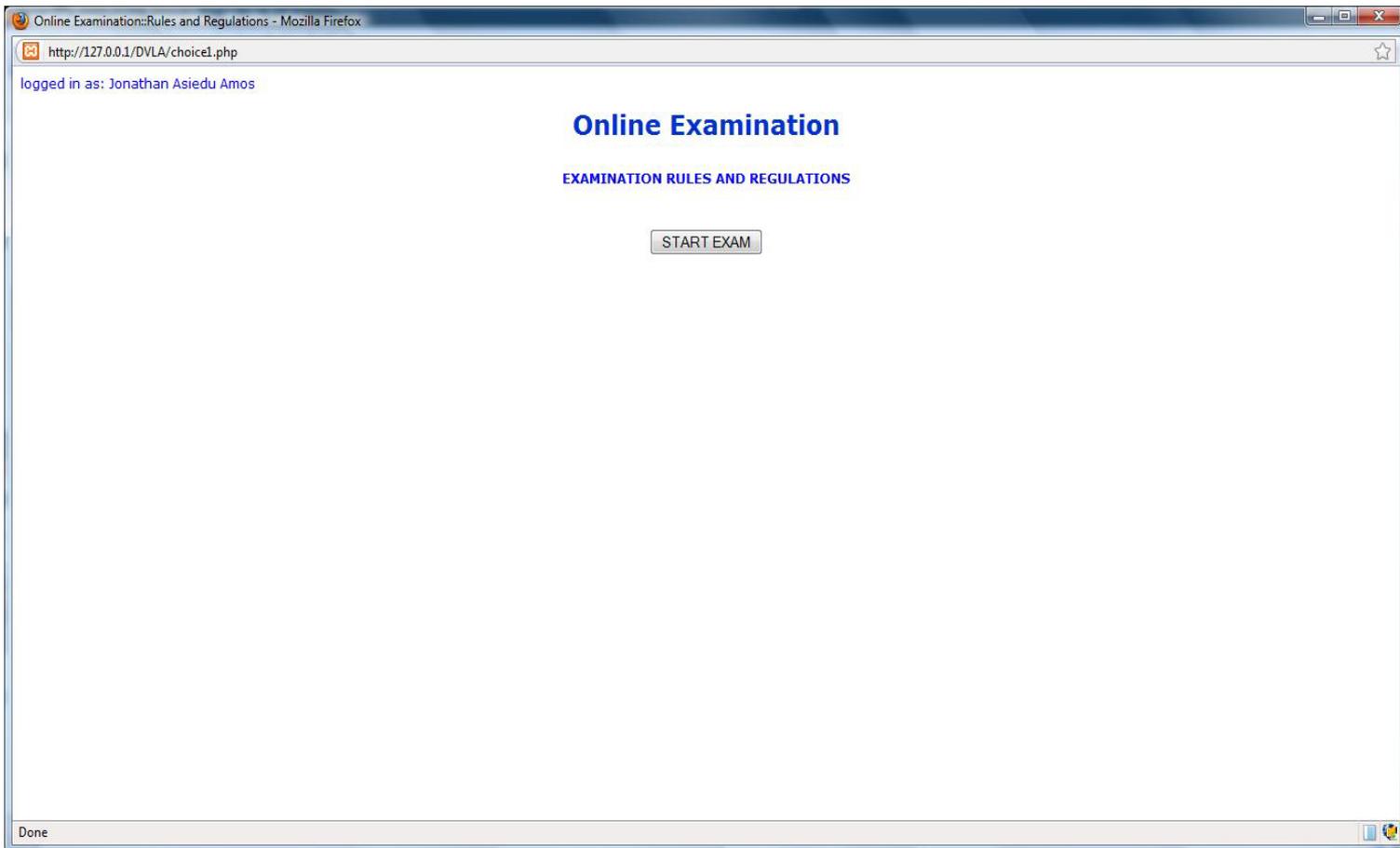


Figure 9: This screen displays the examination rules and regulations.

## 6.1.9 Screen 9 – Student Take an Exam Screen

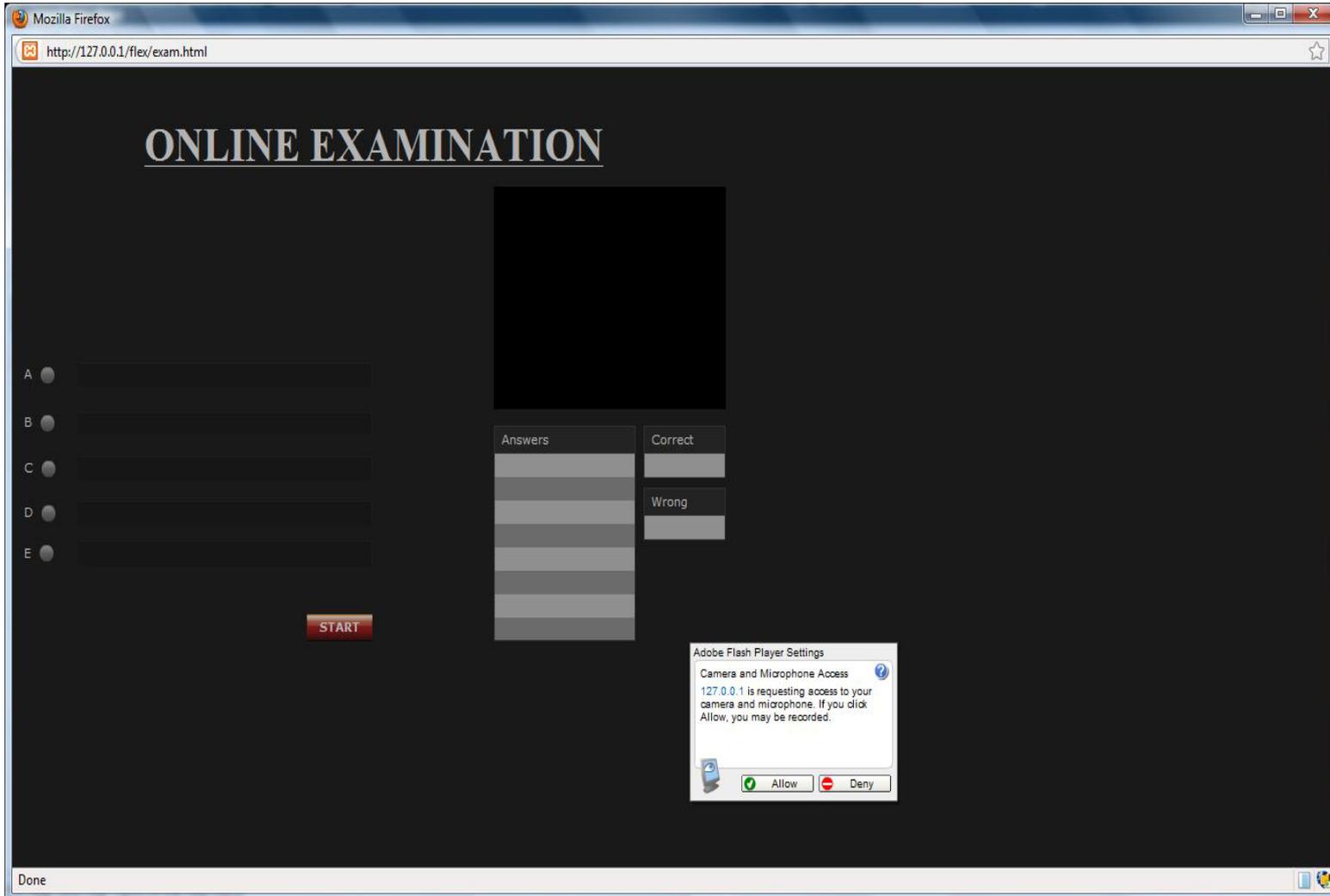


Figure 10: This screen contains the main exam module.

This module is developed in Flash. The user must allow access to their camera and microphone to enable video and audio capture during the examination.

## 6.2 – Question Format

```
1 ;;
2 Question::How old are you?:#A.@10::B.@20::C.@24::D.@40::E.@50:::&Correct::C;;
3 Question::Where are you from?:#A.@Accra::B.@Kumasi::C.@Takoradi::D.@Sekondi::E.@Wa:::&Correct::A;;
4 Question::What are you doing?:#A.@Eating::B.@Drinking::C.@Bathing::D.@Dancing::E.@Banking:::&Correct::E;;
5 Question::What is your name?:#A.@Kwadwo::B.@Kwaku::C.@Kwame::D.@Joshua::E.@Kwasi:::&Correct::D
```

---

```
1 ;;
2 Question::Why am i here?:#A.@To eat::B.@To drink::C.@To present::D.@To show off::E.@To sleep:::&Correct::C;;
3 Question::Do i deserve to be here?:#A.@Maybe::B.@Maybe not::C.@Yes::D.@No::E.@Definately:::&Correct::E;;
4 Question::Am i getting a for my applied project?:#A.@Maybe::B.@Maybe not::C.@Yes::D.@No::E.@Definately:::&Correct::E;;
5 Question::Nathan, Dafla, KG and Edmund would have the answer, wont they?:#A.@Maybe::B.@Maybe not::C.@Yes::D.@No::E.@Definately:::&Correct::A;;
```

Figure 11: The question format that can be uploaded by a lecturer.

Lecturers are allowed to upload questions into the Online Examination System Application. The questions must be written in a particular format for the application to be able to read. The two files above show two different sets of questions. The questions must follow this format in order for the application to work.

### 6.3 - Flowchart

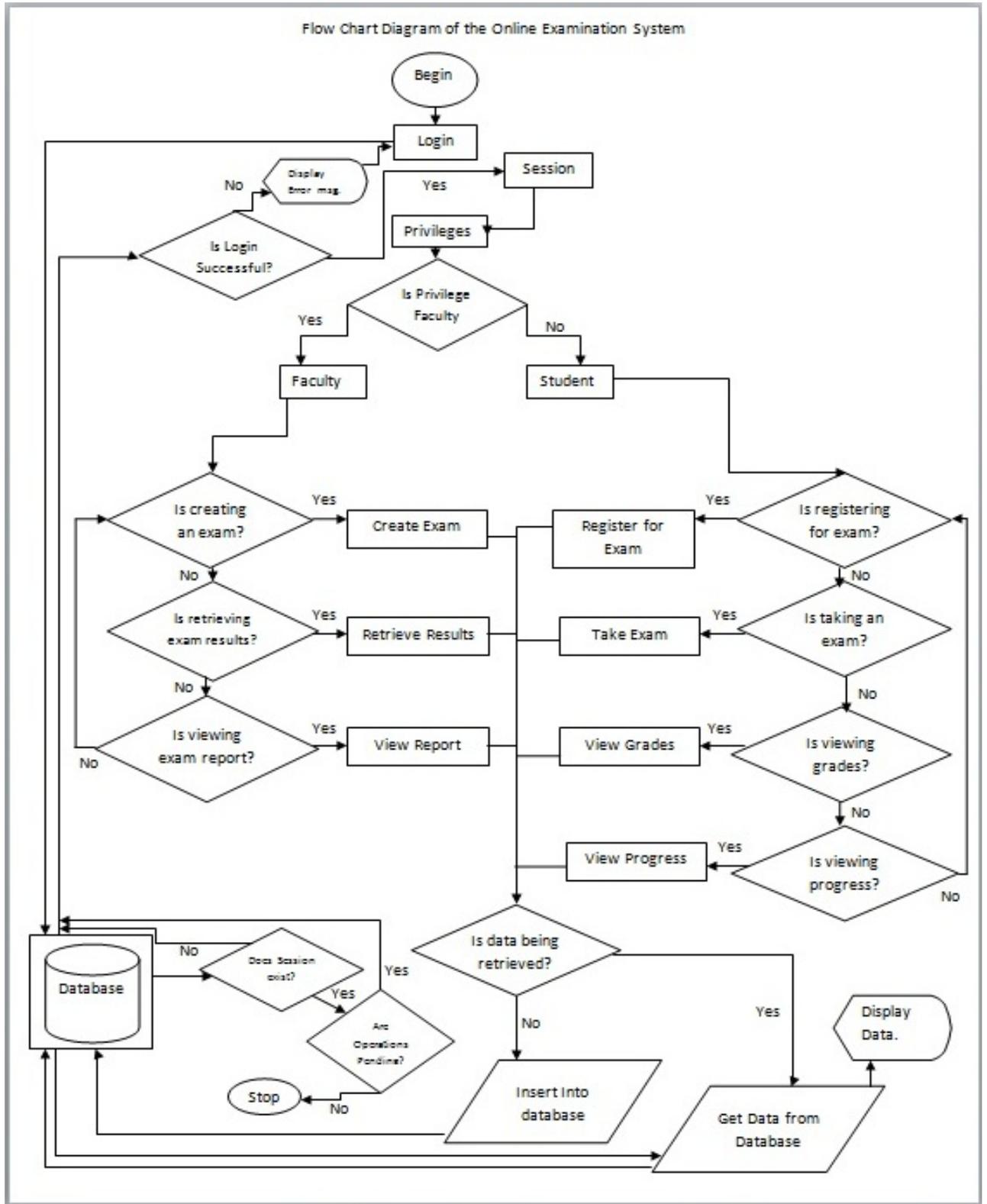
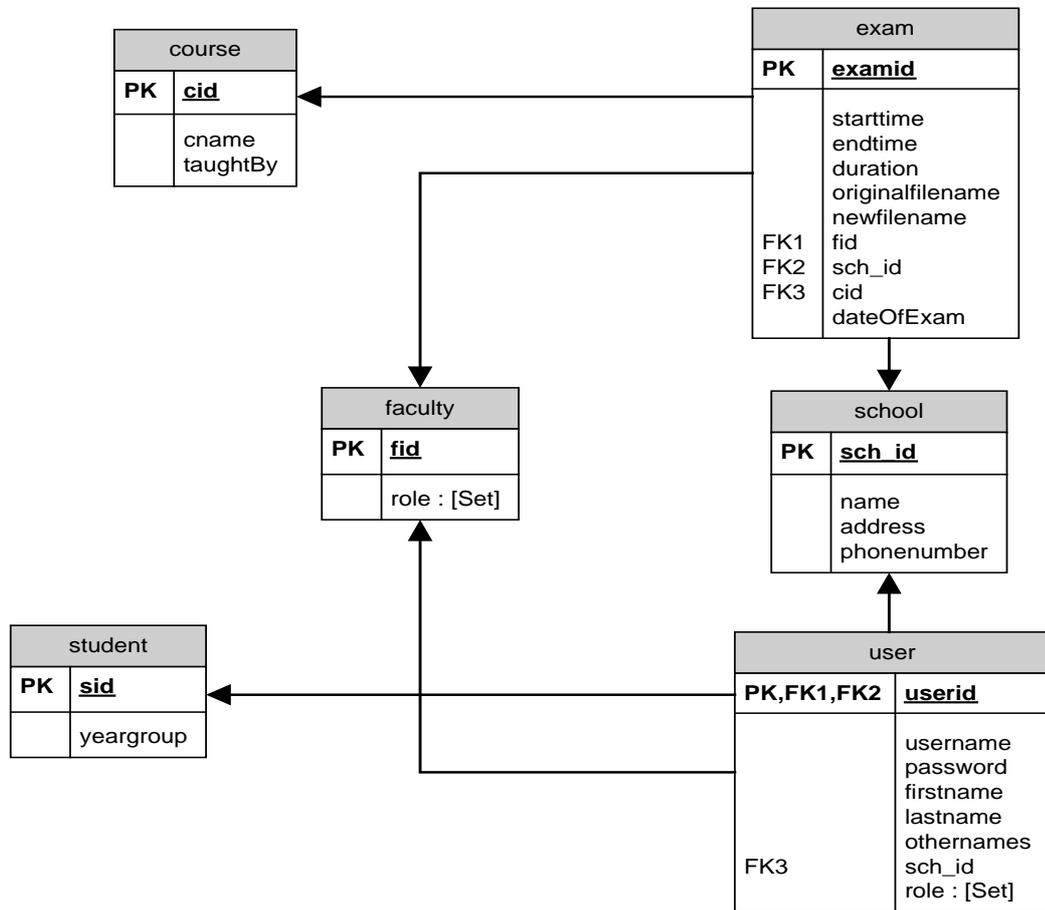


Figure 12: Flowchart of the online examination system.

## 6.4 - UML Diagram

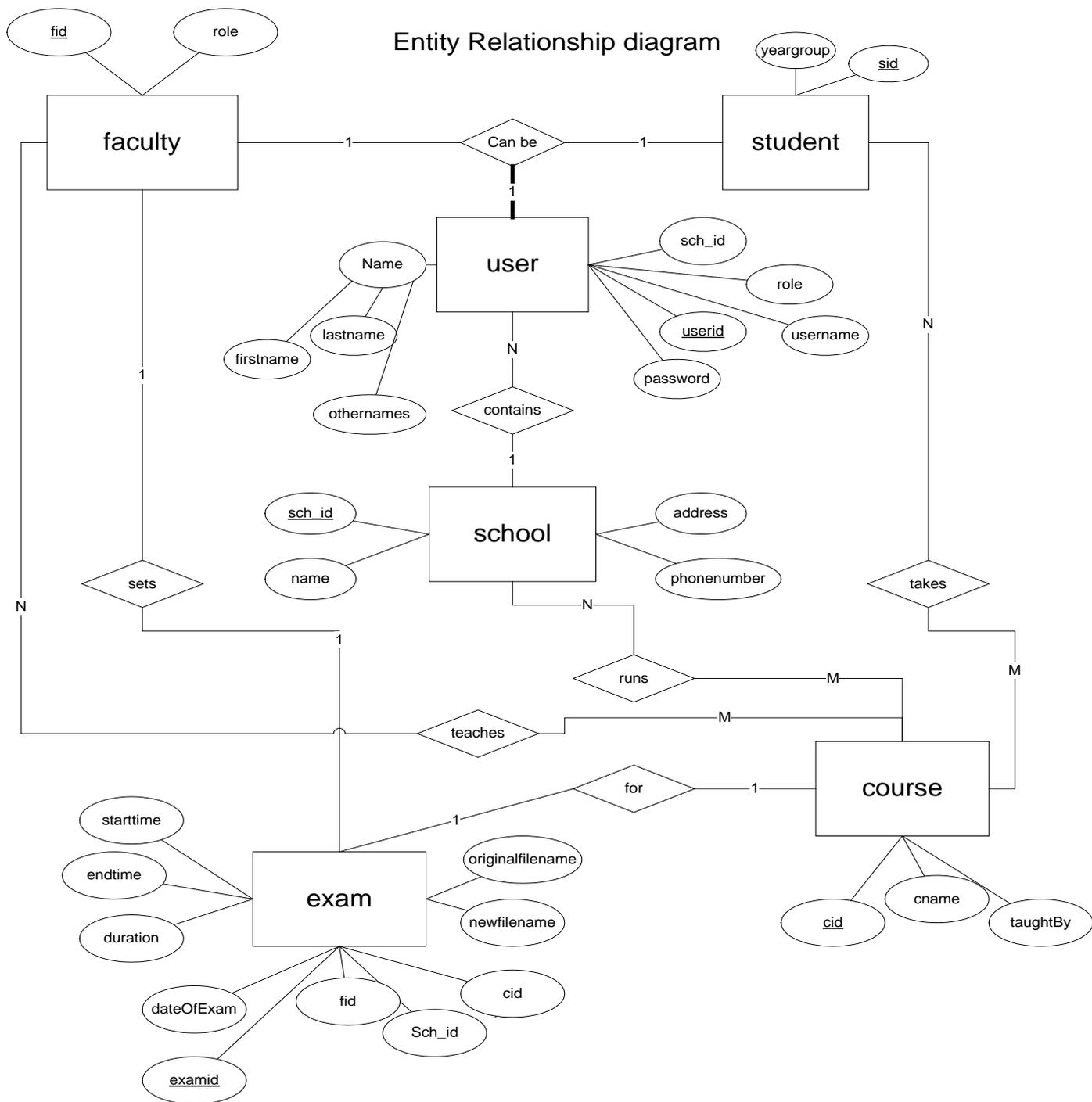
UML Diagram



The UML diagram above shows the difference entities in the application, and a model of the relationship that exist between them.

Figure 12: UML Diagram of the Online Examination System

## 6.5 – Entity Relationship Diagram

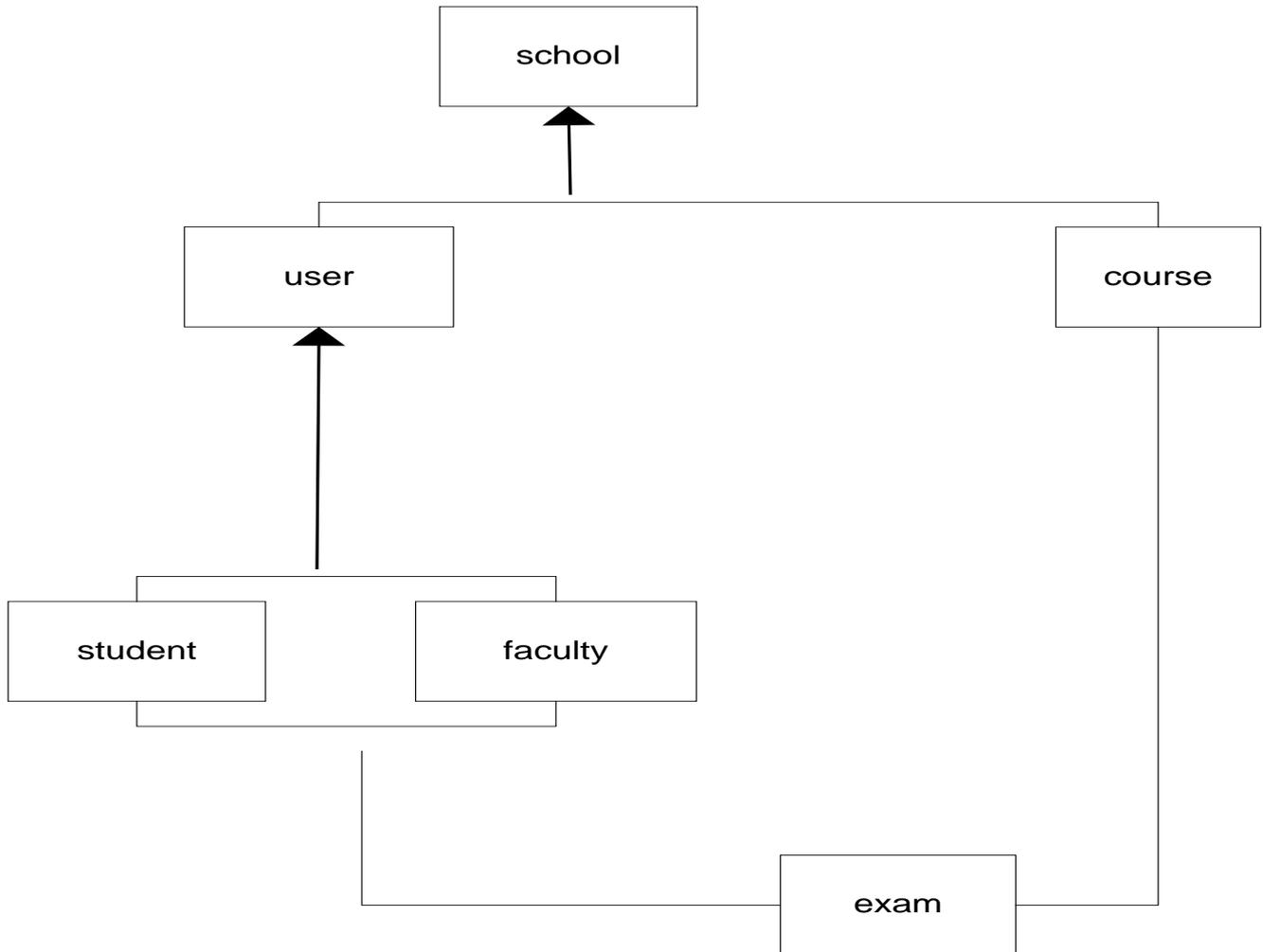


Thick line represents double line

Figure 13: Entity Relationship Diagram of the Online Examination System

## 6.6 – Enhanced Entity Relationship Diagram

### Enhanced Entity Relationship diagram



1. The big arrow shows inheritance with the arrow pointing the parent table
2. The respective attributes of the entities were not shown

Figure 14: Enhanced Entity Relationship Diagram of the Online Examination System

## Endnotes

[1] T. Veluchamy. "Glossary<<Thiyagarajan Veluchamy's Blog". Internet: <http://thiyagarajan.wordpress.com/glossary>, October 2009 [November 3, 2009].

[2] L. Ludwig Consulting Services. "Requirements Terms and Definitions". Internet: <http://www.jiludwig.com/Definitions.html>, August 22, 2008 [November 1, 2009].

[3] R. Malan and D. Bredemeyer. "Functional Requirements and Use Cases - Architecture Resources for Enterprise Advantage". Internet: [http://www.bredemeyer.com/pdf\\_files/functreq.pdf](http://www.bredemeyer.com/pdf_files/functreq.pdf), [November 1, 2009]

[4] Personal. "Session Definition". October 2009.

[5] T. Veluchamy. "Glossary<<Thiyagarajan Veluchamy's Blog". Internet: <http://thiyagarajan.wordpress.com/glossary>, October 2009 [November 3, 2009].

[6] R. Malan and D. Bredemeyer. "Functional Requirements and Use Cases - Architecture Resources for Enterprise Advantage". Internet: [http://www.bredemeyer.com/pdf\\_files/functreq.pdf](http://www.bredemeyer.com/pdf_files/functreq.pdf), [November 3, 2009]

[7] ControlDraw Ltd. "Software Design Specification". Internet: <http://www.urswriter.com/index.html?SoftwareDesignSpecification>, 2009 [November 2, 2009].

[8] EdrawSoft. "Flow Chart Definition - Types of flowcharts, How to draw flow chart and guideline for drawing a flowchart". Internet: <http://www.edrawsoft.com/Flowchart-Definition.php>, 2009 [November 6, 2009].

[9] 4hb.com. "Business Glossary - F: Flowchart". Internet:  
<http://www.4hb.com/25f.html>, [October 2009].