

**HIGHER COLLEGE OF TECHNOLOGY**

**Department of Information Technology**

**Software Engineering Specialization**

**Electronic Examination Marking (E-Marker)**

In Partial Fulfillment of the Requirements for B.Tech Degree with Specialization in Software Engineering

**Submitted by:**

Amal Hamood Al-Jabri: 16j1023

Maryam Said Al-Hashmi: 110S091

Salwa Harib Al- Tobi: 16J09564

**Under the guidance of**

Ronette Sosa Rabanal

Semester 2 / 2014



**HIGHER COLLEGE OF TECHNOLOGY**

**Department of Information Technology**

**BONAFIDE CERTIFICATION**

Certified that the project report “Electronic Examination Marking(E-Marker)” is the bonafide work of “Amal Al-Jabri, Maryam Al-Hashmi, Salwa Al-Tobi” who carried out the project work under my supervision for the partial fulfillment of the requirements for “Bachelor ” with specialization in “ Software Engineering ”.

HEAD OF THE DEPARTMENT SUPERVISOR:

DATE:

**2. Table of Contents**

|  |  |  |
| --- | --- | --- |
| **SL. No** | **Chapter Title** | **Page No.** |
| - | Acknowledgement | 4 |
| - | Abstract | 5 |
| 1 | Introduction | 6 |
| 1.1 | Introduction | 6 |
| 1.2 | Problem Statement | 6 |
| 1.3 | Project Objective | 6 |
| 1.4 | Project Description | 6 |
| 1.5 | Action Plan | 6,7 |
| 2 | System Requirements Specification | 8 |
| 2.1 | Introduction | 8 |
| 2.1.1 | Purpose | 8 |
| 2.1.2 | Definition, Acronym's, Abbreviations | 8 |
| 2.2 | Overall description | 8 |
| 2.2.1 | System Perspective | 8 |
| 2.2.2 | System Features | 8 |
| 2.2.3 | Operating Environment | 9 |
| 2.2.3.1 | Hardware Requirement | 9 |
| 2.2.3.2 | Software Requirement | 9 |
| 2.2.4 | Design/Implementation constraints | 9 |
| 2.2.5 | Assumptions and Dependencies | 9 |
| 2.3 | Specific Requirements | 10 |
| 2.3.1 | Functional Requirements | 10 |
| 2.3.2 | Non-Functional Requirements | 10 |
| 2.3.2.1 | Interface Requirements: | 10 |
| 2.3.2.2 | Performance Requirement | 10 |
| 2.3.2.3 | Life Cycle Requirements | 10 |
| 2.3.2.4 | Reliability requirements | 10 |
| 2.3.2.5 | Security requirements | 11 |
| 2.3.2.6 | Safety requirements | 11 |
| 2.4 | External Interface Requirements | 11 |
| 2.4.1 | User Interface | 11 |
| 2.4.1.1 | Welcome Screen | 12 |
| 2.4.1.2 | About E-marker | 12 |
| 2.4.1.3 | E-marker Guidance | 13 |
| 2.4.1.4 | E-marker Exam list | 14 |
| 3 | System Design | 16 |
| 3.1 | Introduction | 16 |
| 3.2 | System Design Overview | 16 |
| 3.3 | Application Design Detail | 16 |
| 3.3.1 | Table Structure | 16 |
| 3.3.2 | Data Flow Diagram | 17 |
| 3.3.3 | Flow Chart Diagram | 18 |
| 5 | Conclusion and Future | 19 |
| 5.1 | Summary of Findings | 19 |
| 5.2 | Conclusion | 19 |
| 5.3 | Future Scope | 19 |
| - | References | 20 |

**3. List of Figures**

|  |  |  |
| --- | --- | --- |
| **SL. No** | **Name of the diagram** | **Page No.** |
| 1.1 | Action plan Gantt Chart | 7 |
| 1.2 | Action Plan | 8 |
| 1.3 | Welcome Screen | 11 |
| 1.4 | About E-Marker | 12 |
| 1.5 | E-Marker Guidance | 13 |
| 1.6 | E-Marker Exam List | 14 |
| 1.7 | DFD level-0 | 16 |
| 1.8 | DFD level-1 | 16 |
| 1.9 | Flow chart Diagram | 17 |

**4. List of Tables**

|  |  |  |
| --- | --- | --- |
| **SL. No** | **Name of the table** | **Page No.** |
| 1.1 | Teacher table | 15 |

**Acknowledgment**

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals. We would like to extend our sincere thanks to all of them.

We are highly indebted to Ms. Ronette Sosa Rabanal for her guidance and constant supervision as well as for providing necessary information regarding the project and also for her support in completing the project. We would like to express our gratitude towards to E-learning administration staff Mr.Nadeem for his kind co-operation and encouragement which helped us in the completion of this project. Our thanks and appreciations also go to our colleagues in developing the project and people who have willingly helped us out with their abilities.

**Abstract**

This report is about Electronic Examination Marking (E-marker). E-marker is an application used for checking multiple choice questions and generates the scored mark. Teachers will be allowed to enter he correct options and scan the student answer sheet. E-marker application will compare the student answers with the correct answers and generates scored mark.

The reason of us creating this application because the teachers had problems in checking the student papers since they have to go through the questions several times to make sure of their correction. The main Objective of our application is to reduce the lectures time and effort.

**Chapter 1: Introduction.**

**1.1 Introduction:**

**1.2 Problem Statement**:

New technologies have been created to make life easy. In education environment, the teachers face some difficulties to correct some questions like multiple choices and true/false questions. They have to go through the questions several times to make sure of their correction, as long as they will correct each paper manually, a lot of time and effort will be needed.

**1.3 project Objectives:**

New generation creates new things. People nowadays prefer using electronic devices due to the ease of use and time consuming. Electronic Examination Marking(E-Marker) will save lectures effort and time on marking students examination sheets.

**1.4 Project Description:**

E-Marker application is developed for marking exam sheets. E-marker will allow the user to enter the correct options and then scan the student answer sheet. Then the scored mark of the student will be generated based on the correctness of the selected options. In addition E-marker application saves all teacher works under his account.

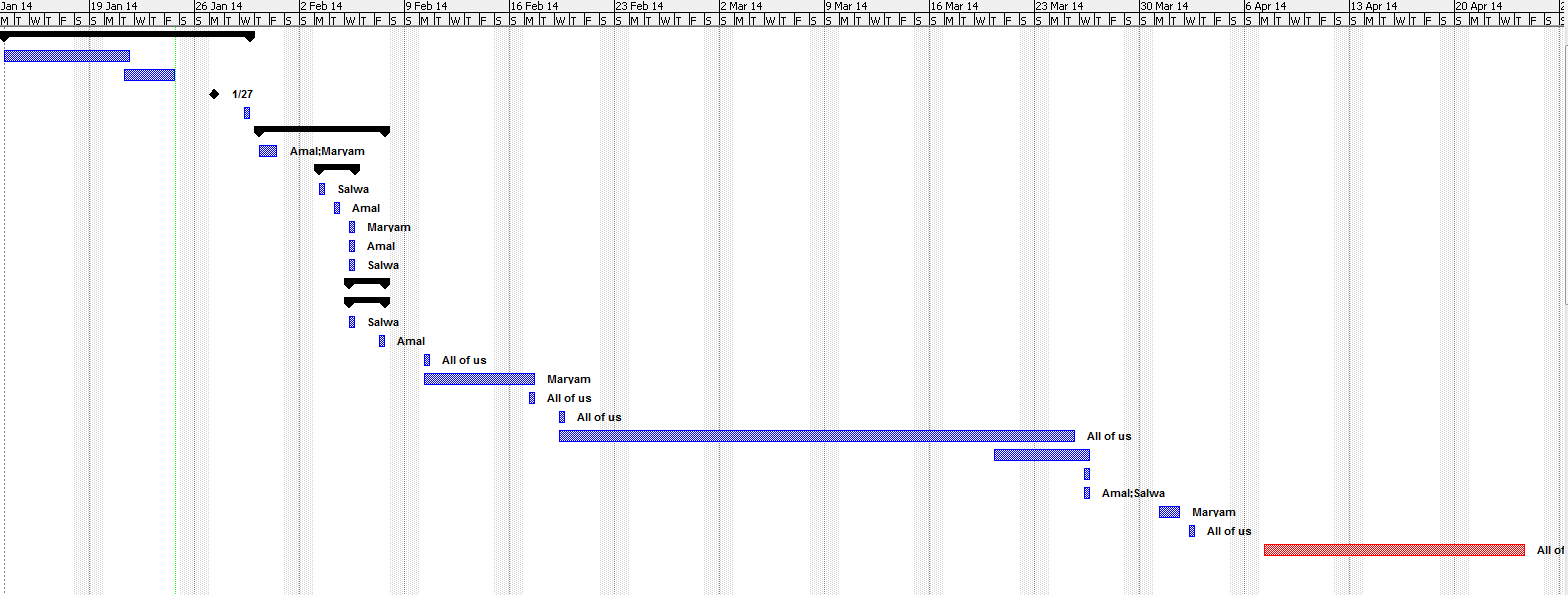
**1.5 Action Plan:**

Figure 1.1 Action Plan Gantt Chart

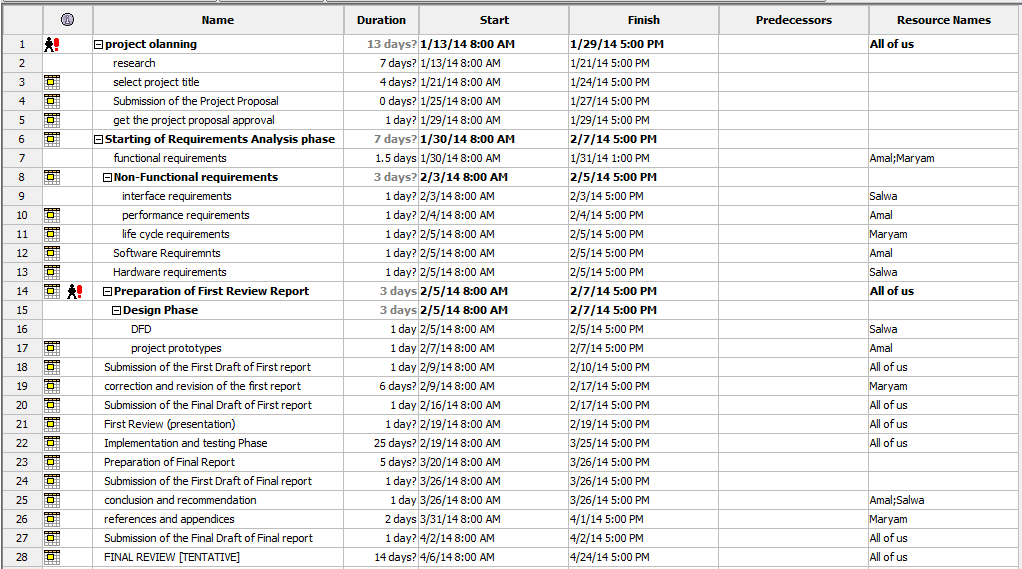


Figure 1.2: Action Plan

**Chapter2: System Requirement Specification**

**2.1 Introduction:**

**2.1.1 Purpose:**

The purpose of this Software Requirement Specification document is to provide a detailed overview of the provided software product, its parameters and goals. This document describes the project's target audience and its user interface, hardware and software requirements. It defines how the client, team and audience see the product and its functionality.

**2.1.2 Definition, Acronym's, Abbreviations:**

* **User**: the person who interact with the system(teacher).
* **E-marker**: the name of the application.
* **Student answer sheet**: the student's answer paper.
* **Administrator**: who will be responsible of administering the system data and the authority to access the application.

**2.2 Overall Description:**

**2.2.1 System Perspective:**

E-marker is a Java application which is developed for marking exam sheet. Teacher will provide the correct options of the exam and will scan the student answer sheet. Comparison functionality will be implemented to compare the student shaded options with the correct options of the exam. Later, the application will calculate student’s score mark and generate it to inform the teacher.

**2.2.2 System features:**

* The user must log in to the system using his user name and password.
* The system will automatically generate random passwords for new users.
* The user can add many exams wanted.
* The student can take more than one exam for the specific subject.
* The system allows the teacher to enter the exam correct options.
* The system saves all the teacher exams under his account.
* The system calls the scanner window to scan the required sheets.
* A progress bar is shown when the scanning is processed.
* The system shows a list of student names with their marks.
* There is no specific quality for scanned papers as long as the shaded options are clear.
* The system will be able to accept images that have been taken with any suitable devices (phones, cameras… etc.).
* The system is safe.
* The system is easy to use.
* The system is fast.

**2.2.3 Operating environment:**

This program will operate in different types of operating systems such as: Windows, Apple Mac OSX, Linux and Solaris. As long as they have the Java Virtual Machine (JVM) on the system. They will be able to run the application.

**2.2.3.1 Hardware Requirement**:

1. Scanner: Canon CanoScan LiDe 110, USB connection.
2. Personal Computer: Laptop: Intel Core i5-3317U Processor, 4 GB DDR3L SDRAM, 500 GB Hard Drive.

**2.2.3.2 Software Requirement:**

1. Visual Studio and Eclipse: Software tools which are required for the interface design phase and the implementation phase for the application.
2. SqlLite: Will be used as a database for the application.
3. Java Development Kit: Used to compile and debug.
4. Java Virtual Machine: Used to run the application.
5. Canon canoScan LiDe 110 software driver: To get the executable file to run the scanning window.

**2.2.4 Design/ Implementation Constraint:**

1. The system must be connected to the scanner to scan the sheets.
2. The system must be connected to the database to store the username and password.
3. The user should first scan the student sheet and then browse the scanned picture from its location.

**2.2.5 Assumption and dependencies:**

1. The system will generate score mark without decimal points.
2. The system will detect the shaded options in case the student shaded two options at a time, a reduced mark will be given for the particular question.
3. The Exam ID that’s given from the teacher must match the Exam ID written in student answer sheet.

**2.3 Specific Requirements**

**2.3.1 Functional Requirements:**

* User should be able to login in by username and password.
* The system should be able to generate random passwords for new users.
* User should be able to add/view/delete exams.
* The system should be able to call the auto-scanner window.
* The system should be able to scan student answer sheet.
* The system should be able to compare between student options and the exam correct options.
* User should be able to view student scored mark.
* The system must be able to save all User’s works under the user account.

**2.3.2 Non-Functional requirements:**

**2.3.2.1 Interface Requirements:**

* The system should be user friendly.

**2.3.2.2 Performance Requirements:**

* The system should be able to react for each user click within maximum 1 seconds.
* The system should be available for service when requested by end-users.

**2.3.2.3 Life Cycle requirements:**

* The system should be maintainable.
* The system should be portable across different computer system.

**2.3.2.4 Reliability requirements:**

* The system shall have strategy for error detection and correction.
* The system should be able to handle any failure occurrence.

**2.3.2.5 Security requirements:**

* The system should ensure the integrity and components from accidental or malicious damage.
* The access permissions for system data may only be changed by the systems data administrator.

**2.3.2.6 Safety requirements:**

* The system should no longer operate if security attacks have become obvious.
* The system should no longer operate in case of fire.

**2.4 External Interface Requirements**

**2.4.1 User Interfaces**

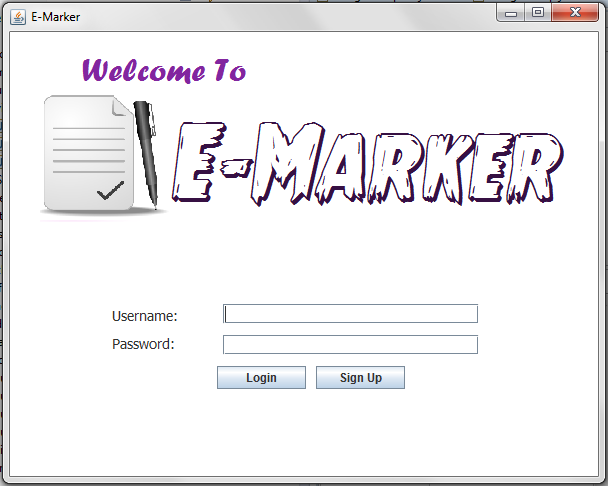


Figure 1.3 Welcome Screen.

**2.4.1.1 Welcome screen :**

The startup page of the application where the user name and password must be provided by the teacher to use E-marker application. New users can register to the application by clicking the Sign Up button.

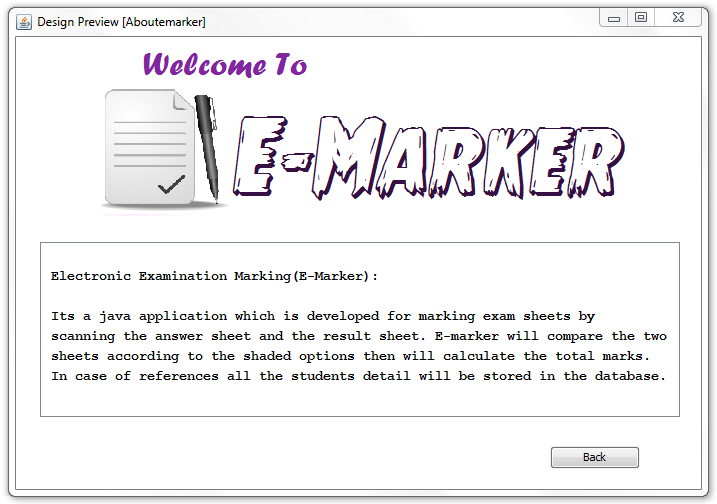
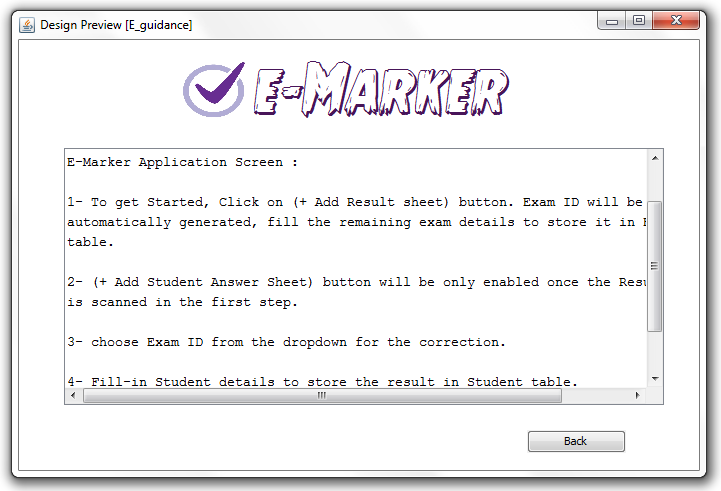


Figure 1.4 About E-Marker

**2.4.1.2 About E-marker:**

This page will hold brief information about E-marker application. User can open this page by clicking the file on menu bar and select About E-marker.

Figure 1.5 E-Marker Guidance



**2.4.1.3 E-marker Guidance :**

E-marker provides the users to view the guidance of how to use E-marker application. Users can click on file on the menu bar and select E-marker guidance to view this page.

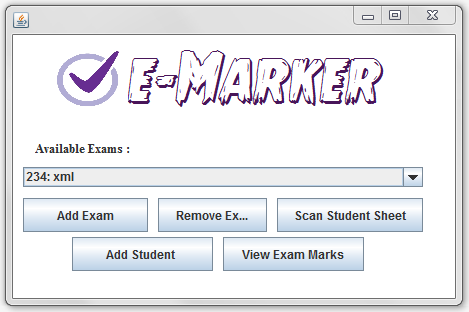


Figure 1.6 E-marker Exam List

**2.4.1.4 E-marker Exam List :**

This screen will pop up once the login process is successful. The teacher must add new exam with the exam name and exam ID. Fill-in form will be provided for the teacher to fill the exam correct options which later will be compared with the student answers. They can also delete unwanted exams from the dropdown list.

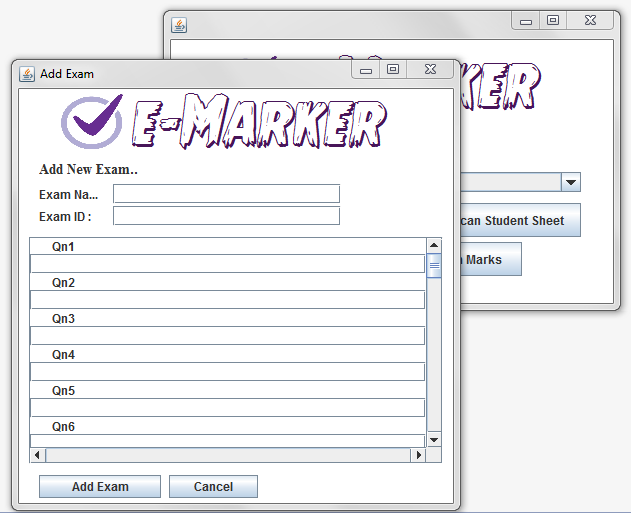


Figure 1.7 Add Exam Screen

**2.4.1.5 Add Exam Screen :**

This screen will pop up to the User once (add Exam button) is clicked. The User must enter the Exam name, Exam ID and the correct options of the particular exam. If the user skipped one question the application will break and take the total number of questions as the maximum score mark of the particular exam.

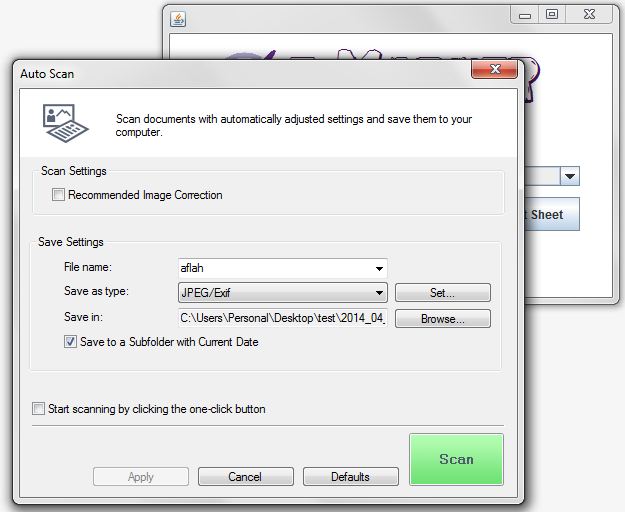


Figure 1.8 Scan Student Sheet

**2.4.1.6 Scan Student Sheet :**

E-marker application calls the executable file of the scanner to allow the users to scan student sheets. Users can specify the path where to save the student scanned sheet.

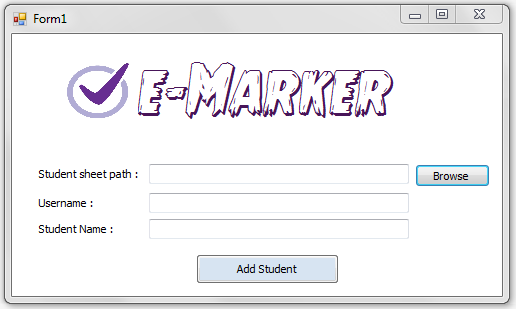


Figure 1.9 Add Student sheet

**2.4.1.7 Scan Student Sheet :**

Once the (add student button) is clicked from the (Exam View screen) the application will allow the user to browse the scanned sheet. User must enter username and student Name and click add student. The application will read the shaded options from the particular student sheet.

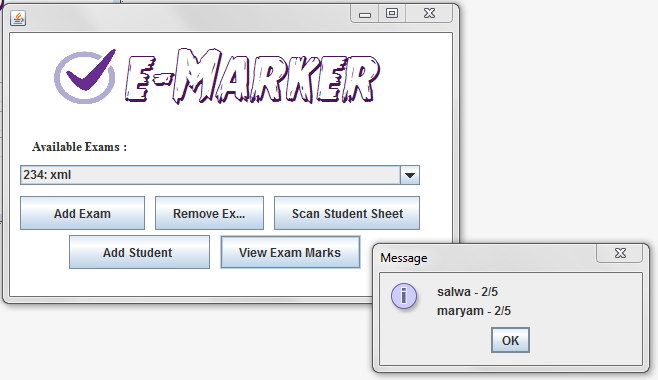


Figure 1.10 View Marks.

**2.4.1.8 View Marks:**

Once the application finishes checking the student sheet, all the students result will be shown when the user clicks (View Exam Marks). It will show the student name and his score.

**Chapter3: SYSTEM DESIGN**

* 1. **Introduction**

The System Design Document describes the system requirements, operating environment, system and subsystem architecture, files, input formats, output layouts, human-machine interfaces, detailed design, processing logic, and external interfaces. This document will be produced with a brief description of the Electronic Examination Marking system.

* 1. **System Design Overview**

Electronic Examination Marking (E-marker) is an application which allows user to scan student answer sheet. The application will process the comparison between student options and the exam correct options then generates student scored mark. The following displays show the structure of E-marker application.

**3.3 Application Design Detail**

**3.3.1 Table Structure**

i. Table Name: Teacher

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Field Size** | **Constraints** | **Description** |
| Teacher\_ID | Number | 4 | Not Null | Primary Key |
| Username | Character | 30 | Not Null |  |
| Password | Number | 10 | Not Null |  |

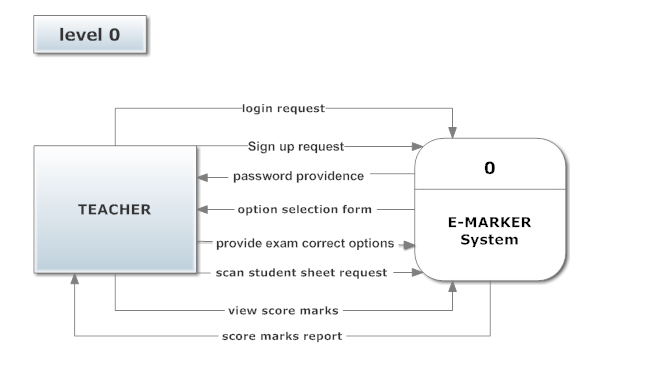
**3.3.2 Data Flow Diagrams**

Figure 1.7: DFD level-0

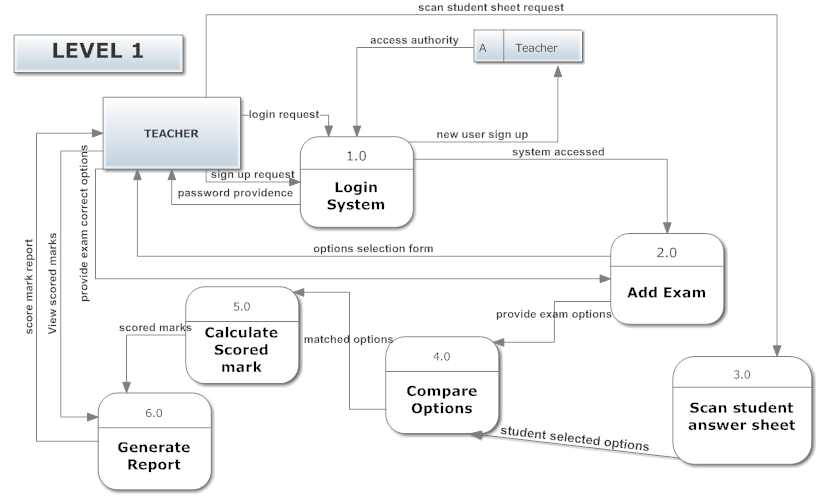


Figure 1.8: DFD level-1

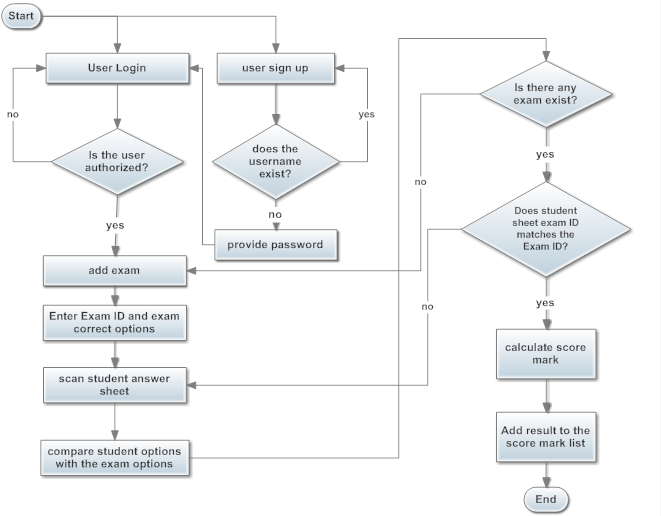
**3.3.3 Flow chart Diagram**

Figure 1.9 : Flow Chart Diagram

**Chapter 4: SYSTEM IMPLEMENTATION:**

**4.1 Implementation**

4.1.1 System Implementation Description

The system starts up by showing the main screen which contains the login and the sign up buttons for the teacher to enter and access to the application. E-marker application allows the user to add the correct exam options and scan student answer sheet. Later, it will read the values from the specified student sheet and store it in array. The students options will be compared with the correct exam options and the application will calculate the scored marked.

4.1.2 Coding

4.1.3 Validation Checks

**4.2 Testing**

**Black Box Testing:**

|  |  |
| --- | --- |
| **Black Box Test case** | **Expected output** |
| * If user enter new user name * If user entered existing username * If user enter wrong password * When user click on Add exam button:   + User should enter three digits for Exam ID.   + User should enter the Exam name.   + User should enter the correct answer for each question. * If user click on remove button. * When user click on scan student sheet. * When user click on add student button:   + User should browse the image of the scanned sheet   + User should enter the username   + User should enter the student name * When user click on view exam mark button. | * System will auto generate the password * Error message will pop up (user is existing) * (Password not found) message will pop up * When user click on Add Exam button:   + Exam ID and name will be added into drop down list. * Added exam name and ID will be deleted. * The student answer sheet will be scanned and saved in a folder which the user specified. * (sheet added ) message will appear if the selected image match with Exam ID and   ( Exam ID not found) Error message will appear if the selected image is not match with Exam ID.   * Student’s mark will be listed down. |

**White box testing:**

1-Test if the entered user name in sign up button does exist or not.

2-Test if the user name and password is matched.

3-Test if the exam id is unique and is matches the one in the student sheet.

4-Test if there is blank text field in the entered options of the question.

5-Test if java opens the executable file of c# project.

6- Test if C# will pass the arguments through the command line.

7-Test if the entered values in exam options are saved in an array.

8- Test if array of student options match the array of exam options.

9-Test if the application calls the executable file of the scanner.

10-Test if the application calculates the score mark of the student.

11-Test if the application will display all students' marks which have been marked by the teacher.

12- Test if the entered Exam name and Exam id will be shown in the dropdown list.

**Chapter 5: Conclusion and Future**

**5.1 Summary of Findings:**

Our project is about marking multiple choice questions by scanning the student answer sheet. Based on the shaded circles the application will identify the options selected and calculate the scored mark of a particular student.

* The problems that we found during that sometimes:
  + Finding the executable file of the scanner.
  + Read the exact selected options from student answer sheet.

**5.2 Conclusion**

As a group, we start working on our project which is Electronic Examination Marking aiming that we will achieve our goal. Our project is to find the score mark of the student by marking it through our application. The marking can be done by scanning the student answer sheet. Then the application will retrieve the scanned sheet and compare it with the exam correct options based on the entries that the teacher made and later it will provide the teacher with the student score mark.

**5.3 Future Scope**

We are aiming to improve E-marker application by increasing some features which will be very helpful to the users:

* Recommendations :
  + Provide a template for the correct answer sheet based on the number of questions and options that have been provided by the teacher.

**References**

[1] Kathy Sierra and Bert Bates, Head First Java, O'Reilly Media, USA, February 2005.

[2] Eric Freeman, Head First Design Patterns, O'Reilly Media, USA, Nov 2004.

[3] Donald G. Drake, Introduction to Java threads, JavaWorld, USA, Jan 1996, <http://www.javaworld.com/jw-04-1996/jw-04-threads.html>

# [4] Umar.techBOY, Optical Marks Recognition (OMR) Engine 1.0, USA, 20 Aug 2013,

# <http://www.codeproject.com/Articles/451169/Csharp-Optical-Marks-Recognition-OMR-Engine>

# [5] [Jeff Friesen](https://today.java.net/pub/au/174), Acquire Images with TWAIN and SANE

# , USA, November 18, 2004,

# <https://today.java.net/article/2004/11/16/java-tech-acquire-images-twain-and-sane-part-1>