

HIGHER COLLEGE OF TECHNOLOGY

Department of Information Technology

Software Engineering Specialization

FIRST REVIEW REPORT

**For**

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

**Prepared 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

**SYSTEM REQUIREMENTS 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:**

1- user: the person who interact with the system.

2- E-marker: the name of the application.

3- Result answer sheet: the answer sheet with all the correct answers.

4- Student answer sheet: the student's answer paper.

5-Adminstrator: 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 sheets by scanning the result sheet and student answer sheet. The application will compare two sheets according to the shaded options then it will calculate the total mark. Basically comparison function will be used to compare between the two retrieved images. In case of reference all student details will be stored in the database.

**2.2.2 System features:**

1- The system will automatically scan the result sheet and store in the database by clicking a button.

2- The user can add many exams wanted.

3- The system should allow the user to choose specific exam by selecting Exam Id for correction.

4- The system will automatically generate unique exam id for the scanned result sheet to be stored in the database.

5- The system will generate PDF file to retrieve the student details from the database in case of printing or saving documents.

6- The user must log in to the system using his user name and password.

7-The system will turn off when the user enters the user name or password wrongly three times.

8- The student can take more than one exam for the specific subject.

**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: one USB cable for data and power.

2- Personal Computer:

Laptop: Intel Core i5-3317U Processor, 4 GB DDR3L SDRAM, 500 GB Hard Drive, windows 7.

**2.2.3.2 Software Requirement:**

1- Netbeans and Eclipse: software tools which are required for the interface design phase and the implementation phase for the application.

2- Microsoft access: Will be used as a database for the application.

3- Java Development Kit: used to run the application.

4- Java Virtual Machine: used to run the application.

**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 result.

**2.2.5 Assumption and dependencies:**

1- The system will not be able to scan student answer sheet unless there is at least one scanned result sheet.

**2.3 Specific Requirements**

**2.3.1 Functional Requirements:**

* User should be able to login in by username and password.
* User should be able to choose particular exam sheet for the correction.
* The system should be able to scan result and answer sheet.
* The system should be able to compare between two sheets and calculate the total mark based on the shaded options.
* The system should be able to retrieve student details from student table.
* The system should be able to provide the user with guidance.
* The system should be able to allow the users to access the database.
* User should be able to view student marks.

**2.3.2 Non-Functional requirements:**

**1. Interface Requirements:**

* The system should be user friendly.

**2. Performance Requirements:**

* The system should be able to react for each user click within maximum 30 seconds.
* The system should be available for service when requested by end-users.

**3. Life Cycle requirements:**

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

**4. Reliability requirements:**

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

**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 system’s data administrator.

**6. Safety requirements:**

* The system should no longer operate if security attacks have

become obvious.

* The system should not longer operate in case of fire.

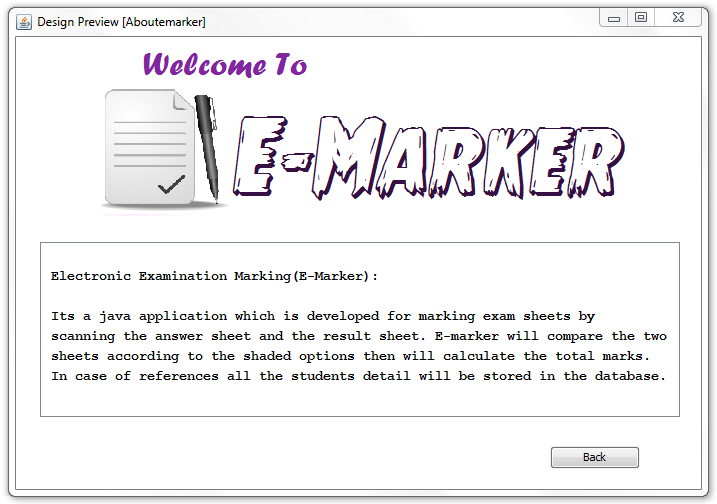
**2.4 External Interface Requirements**

**2.4.1 User Interfaces**



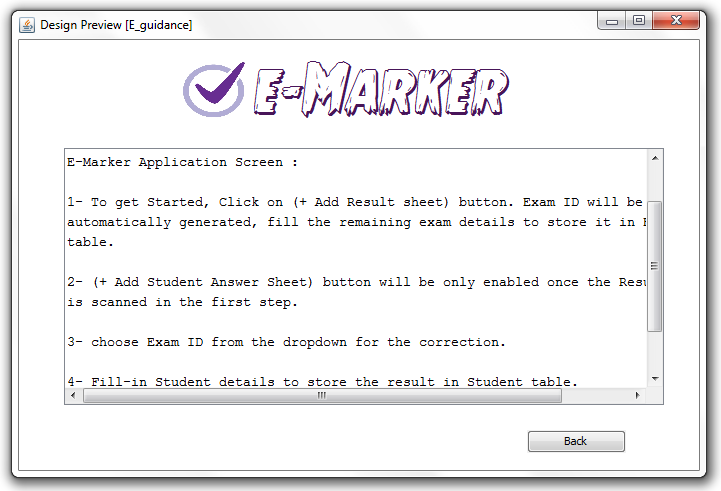
**Welcome screen :**

The startup page where the user name and password must be provided by the teacher to use E-marker application.



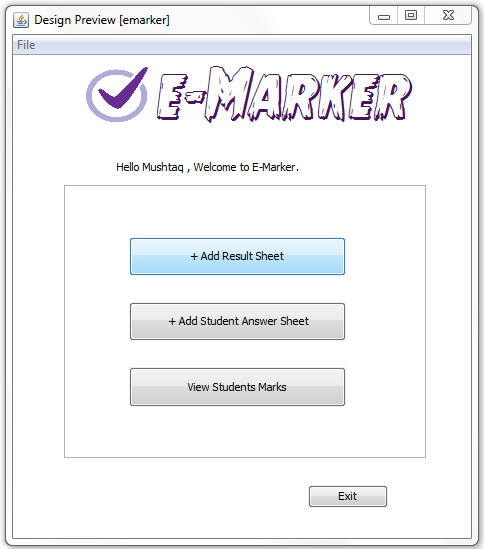
**About E-marker screen:**

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 or by clicking the shortcut (Ctrl+A).



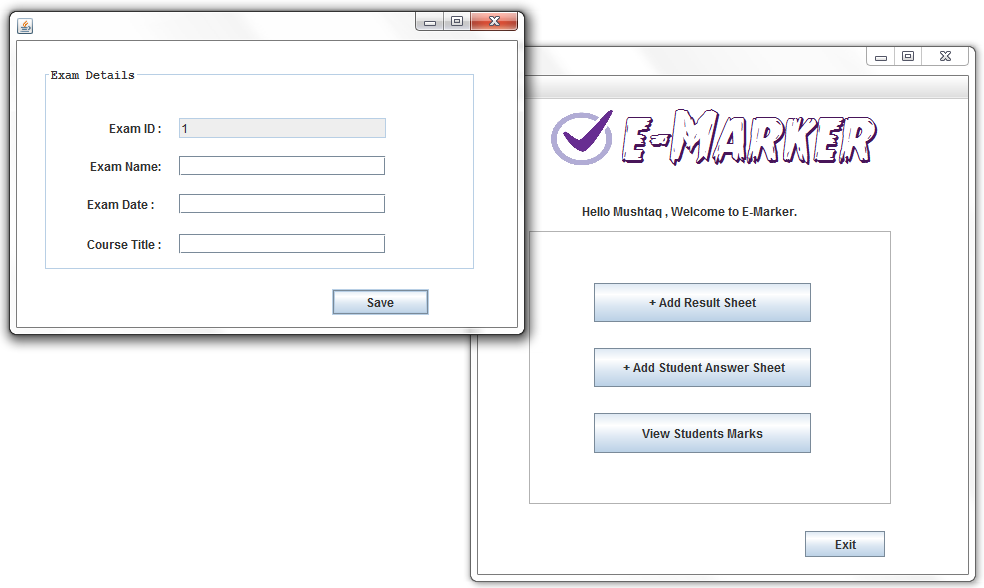
**Guidance Screen :**

E-marker provides the users to view the guidance of how to use E-marker application. User can click on file on the menu bar and select E-marker guidance to view this page or by pressing the shortcut (Ctrl+E).



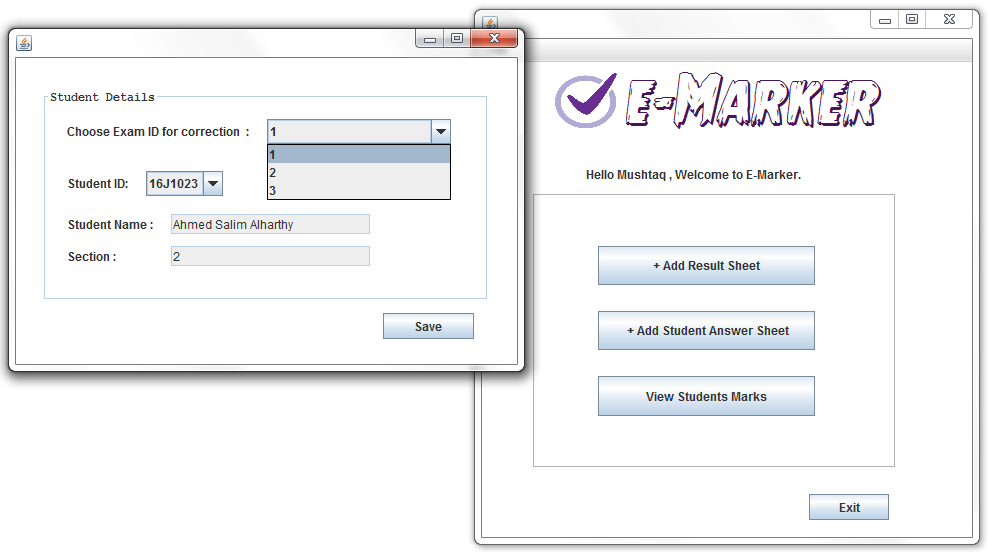
**E-Marker Application Screen:**

This page will provide the user with 3 main button. The first button will allow the user to scan result sheet. Users can add as many result sheets they want. Second button will allow the user to scan student answer sheet. The total mark will be stored in student database so once the third button is clicked E-marker will generate a table to show all students mark.



**Exam Details Screen:**

This page will pop up once the first button is clicked. E-marker will generate unique Exam IDs to be saved in exam’s table. User will fill-in the remaining exam details and will click save button.



**Student details screen:**

(+ Add Student Answer Sheet) button won’t be enabled if there is no scanned result sheet. Once it’s enabled student details screen will pop up for the user to select the specific exam id for the correction. User will select student id from the dropdown list and the remaining details will be automatically retrieved from student table. Once the save button is clicked the comparison of two sheets will be processed and the total mark will be generated.

**SYSTEM DESIGN DOCUMENT**

* 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 exam result sheet and student answer sheets. The application will process the comparison between result sheet and student answer sheets and calculates total mark of the students. The following displays show the structure of E-marker application.

**3.3 Application Design Detail**

**3.3.1 Table Structure 3.3.1 Table Structure**

1. Table Name: Student

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Field Size** | **Constraints** | **Description** |
| Student\_ID | Text | 8 | Contain 1 letter | Primary Key |
| Student\_Name | Text | 30 | Not Null |  |
| DOB | Date/Time |  |  |  |
| Email | Text | 30 |  |  |

2. Table Name: Teacher

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

3. Table Name: Exam

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Field Size** | **Constraints** | **Description** |
| Exam\_ID | Number | 4 | Not null | Primary Key |
| Exam\_name | Text | 20 |  |  |
| Exam\_Date | Date/Time |  |  |  |
| Result\_sheet |  |  |  |  |
| Total\_mark | Number | 3 | Not Null |  |
| Section\_ID | Number | 4 | Not Null | Foreign Key |

4. Table Name: Section

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Field Size** | **Constraints** | **Description** |
| Student\_ID | Text | 8 | Contain 1 letter | Primary Key |
| Section\_Name | Text | 30 | Not null |  |
| Teacher\_ID | Number | 4 | Not Null | Foreign Key |

5. Table Name: Answer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Field Size** | **Constraints** | **Description** |
| Answer\_ID | Number | 4 | Not Null | Primary Key |
| Student\_ID | Text | 8 | Contain 1 letter | Foreign Key |
| Exam\_ID | Number | 4 | Not Null | Foreign Key |
| Student\_sheet | Number | 3 |  |  |
| Student\_mark | Number | 3 |  |  |

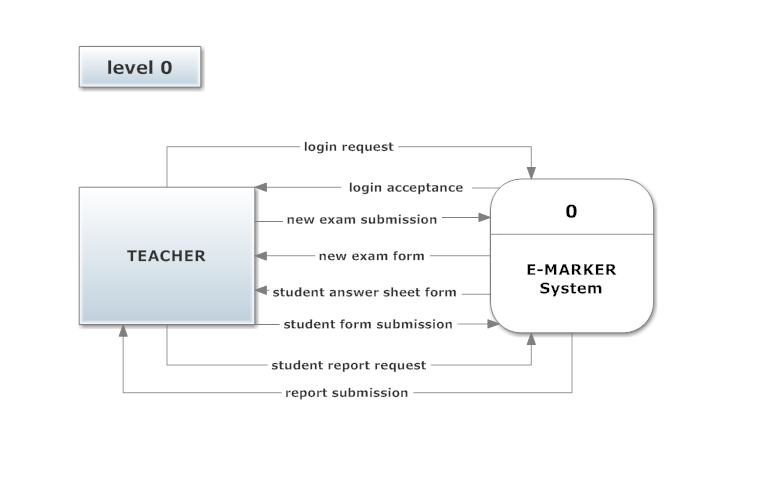
iv. Table Name: Student\_Section

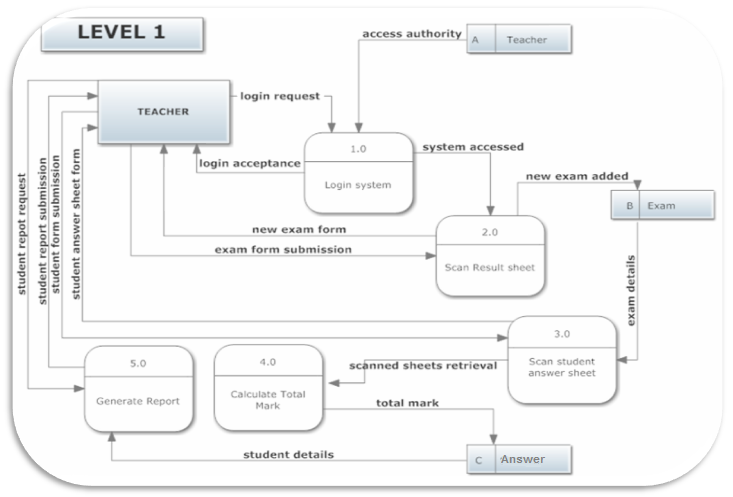
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Field Size** | **Constraints** | **Description** |
| Student\_ID | Text | 8 | Contain 1 letter | Foreign Key |
| Section\_ID | Number | 4 | Not null | Foreign Key |

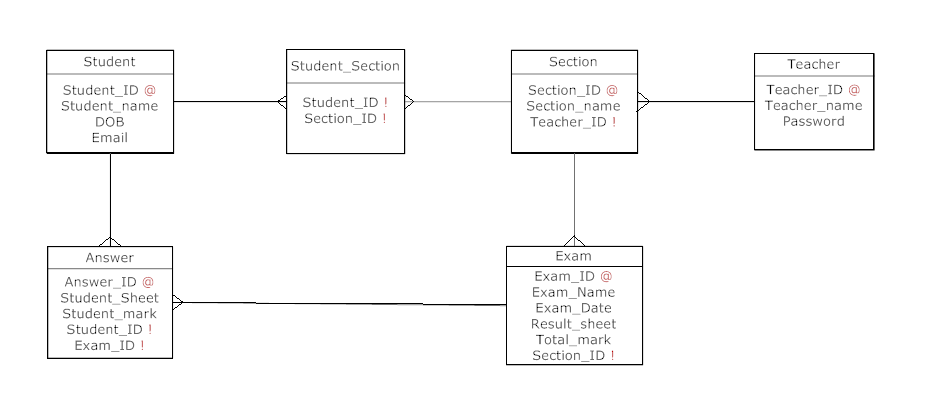
**3.3.2 Normalized Table Layout**

|  |  |  |  |
| --- | --- | --- | --- |
| **UNF** | **FNF** | **SNF** | **3NF** |
| Student\_ID @  Student\_name  DOB  Email  Student\_Sheet  Student\_mark  Teacher\_ID  Teacher\_name  Password  (Section\_ID  Section\_name  Exam\_ID  Exam\_Name  Exam\_Date  Result\_sheet  Total\_mark)\* | **Student** | **Student** | **Student** |
| Student\_ID @  Student\_name  DOB  Email  Student\_Sheet  Student\_mark  Teacher\_ID  Teacher\_name  Password | Student\_ID @  Student\_name  DOB  Email  Student\_Sheet  Student\_mark | Student\_ID @  Student\_name  DOB  Email |
| **Section** | **Teacher** | **Answer** |
| Student\_ID !  Section\_ID@  Section\_name  Exam\_ID  Exam\_Name  Exam\_Date  Result\_sheet  Total\_mark | Teacher\_ID @  Teacher\_name  Password | Answer\_ID @  Student\_Sheet  Student\_mark  Student\_ID !  Exam\_ID ! |
| **Student\_Section** | **Teacher** |
| Student\_ID !  Section\_ID ! | Teacher\_ID @  Teacher\_name  Password |
| **Section** | **Student\_Section** |
| Section\_ID@  Section\_name  Exam\_ID  Exam\_Name  Exam\_Date  Result\_sheet  Total\_mark  Teacher\_ID! | Student\_ID !  Section\_ID ! |
| **Section** |
| Section\_ID@  Section\_name  Teacher\_ID! |
| Exam |
| Exam\_ID @  Exam\_Name  Exam\_Date  Result\_sheet  Total\_mark  Section\_ID ! |

**3.3.3 Data Flow Diagrams**





**3.3.4 Entity – Relationships**