System Requirements Specification for: E-XAMINE

Contents

1 Introduction

1.1 Methodology

Rapid Prototype Model :

The goal of a prototyping-based development is that instead of freezing the requirements before any design or coding can proceed, a throwaway prototype is built to help understand the requirements. This prototype is developed based on the currently known requirements. Development of the prototype obviously undergoes design, coding, and testing, but each of these phases is not done very formally or thoroughly. By using this prototype, the client can get an actual feel of the system, which can enable the client to better understand the requirements of the desired system. This results in more stable requirements that change less frequently.

Prototyping is an attractive idea for complicated and large systems for which there is no manual process or existing system to help determine the requirements. It is also an effective method of demonstrating the feasibility of a certain approach. This might be needed for novel systems, where it is not clear that constraints can be met or that algorithms can be developed to implement the requirements. In both situations, the risks associated with the projects are being reduced through the use of prototyping.

Overall, prototyping is well suited for projects where requirements are hard to determine and the confidence in the stated requirements is low. In such projects where requirements are not properly understood in the beginning, using the prototyping process model can be the most effective method for developing the software.

1.2 Purpose

E-XAMINE is a web application which provides a online portal to conduct online tests based on multiple choice questions. Student users can give a particular test and see their performance whereas teachers has the option to conduct test by adding questions and their corresponding answers along in the system. The questions in this case is single correct multiple choice question with a certain positive score for correct answer and a negative score for wrong answer.

1.3 Definitions, Acronyms and Abbreviations

- E-XAMINE : Web application to conduct the online test

- Teacher : One who set the questions for the test

- Student : One who attempts the test

1.4 Overview

The remaining sections of this document provide a general description, including characteristics of the users of this project, the product's hardware, and the functional and data requirements of the product. General description of the project is discussed in section 2 of this document. Section 3 gives the functional requirements, data requirements and constraints and assumptions made. Section 3 also discusses the external interface requirements and gives detailed description of functional requirements.

2 Overall Description

2.1 Product Perspective

The product has two main interfaces one for student users and one for teachers. Teachers need to register in order to set a test. Teachers can also edit their profile. Teachers can set a test by entering multiple choice questions and respective answers into the database through the interface. Students need to register in order to take the test. Students can also edit their profile. Student can take test on another interface and result of the test is stored in the database. Student can also see their performance subject and topic wise of their previous attempted tests.

2.2 Product Functions

For teacher users :-

- Teacher registers on the system.
- Teacher logs in to the system to set the test.
- Teacher can edit his profile.
- Teacher sets the questions for the test.
- Each questions and theirs respective correct answers are set by the teacher.
- Each questions is assigned a marks and total marks for the test is set.

For student users :-

- Student registers on the system
- Student logs in to the system to take the test
- Student can edit their profile
- Student takes the test.
- At the end of the test, marks obtained and percentage is displayed.
- Student can view his performance subject wise over his last few tests in a graph.

2.3   General Constraints

- GUI is only in English.

- Log in user name and password is used for the identification of users.

- Only registered teachers and students will be authorized to use the web application.

- Limited to HTTP.

3   Specific Requirements

3.1   External Interface Requirement

3.1.1   User Interface

The user interface for the software shall be compatible to any browser such as Internet Explorer, Mozilla, Google Chrome, Opera or Netscape Navigator by which user can access to the system.

3.1.2   Hardware Interface

Since the application must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for e.g. Modem, WAN LAN, Ethernet Cross-Cable.

3.1.3   Software Interface

Various software interfaces used are :-

- PHP : For server side scripting

- Codeigniter : For php framework

- MySQL : For database

- HTML, CSS, JQuery : For front end

- Google Graph : For displaying performance of students graphically.

3.1.4   Communication Interface

The E-XAMINE system shall use the HTTP protocol for communication over the internet and for the intranet communication will be through TCP/IP protocol suite.
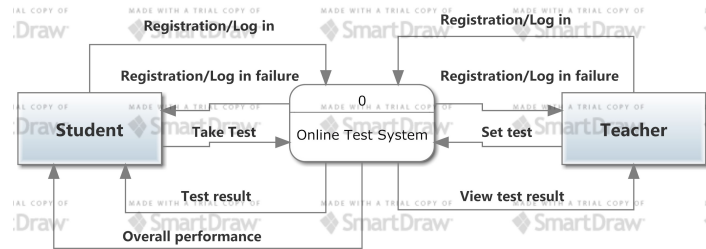
Figure 1: Context Level Diagram for E-XAMINE

3.2 Functional Requirements

3.2.1 Data Flow

The context level diagram and data flow diagram for the E-XAMINE is shown in figure 1 and figure 2 respectively.

Entities :

- Teacher

- Student

Processes :

- 1.0 Registration

- 2.0 Log In

- 3.0 Test

- 4.0 Student Performance

Data Stores :

- D1 User Database

- D2 Student Profile

- D3 Teacher Profile

- D4 Test Database
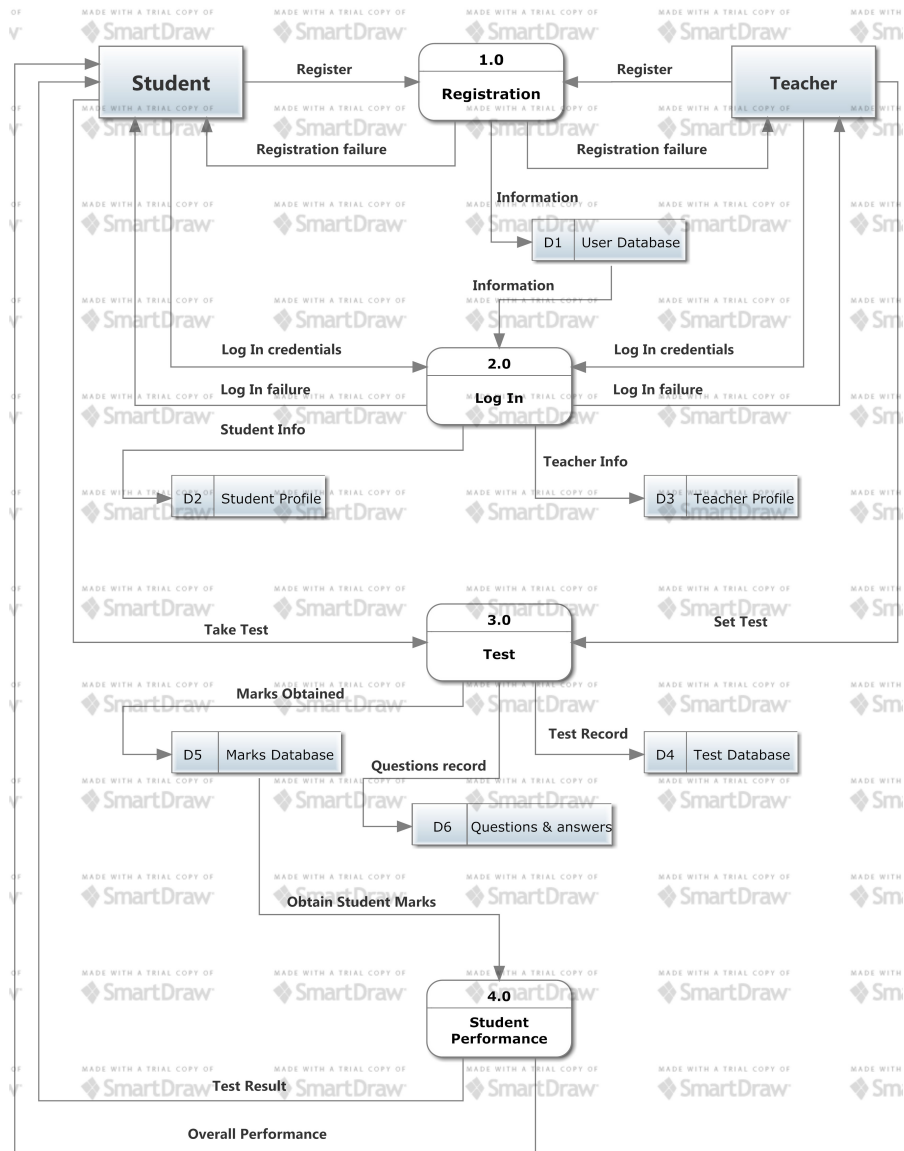
- D5 Marks Database

- D6 Questions and answers

Figure 2: Data Flow Diagram for E-XAMINE

3.2.2 Functional Requirement 1: Log In

UC1: Log in for users
Primary Actor: Student/Teacher
Precondition: Student/Teacher has already registered
Main Success Scenario:
User validates the email id and password correctly
Exception Scenarios:
If the user used wrong user name or password or user have not registered yet
(System tells the user about it.)

3.2.3 Functional Requirement 2: Set a test

UC2: Set a test
Primary Actor: Teacher
Precondition: Teacher has already logged in
Main Success Scenario:
1. Teacher selects the subject he want to set the test for
2. Teacher inputs the topic of the test
3. Teacher sets the questions and the corresponding correct option for each
4. Teacher assign a certain marks for correct answer and certain marks for wrong answer
5. System accepts the test set up and posts it
Exception Scenarios:
If the teacher entered negative marks instead of positive marks for each question
(System tells the user about it.)

3.2.4 Functional Requirement 3: Take a test

UC3: Take a test
Primary Actor: Student
Precondition: Student has already logged in
Main Success Scenario:
1. Student selects the test from the list which he wants to take
2. Student attempts each question
3. System takes the answers and calculate the marks scored
4. System posts the marks and percentage obtained and displays it to the student
Exception Scenarios:
None

3.2.5 Functional Requirement 4: View overall performance

UC1: View
Primary Actor: Student
Precondition: Student has already logged in and has given atleast a test on the

selected subject
Main Success Scenario:
1. Student selects the subject which he wants to view his performance
2. System accepts the subject and displays a graph showing the performance of his previous test on that particular subject
Exception Scenarios:
If the student has not given a single topic on the selected subject
(System tells the user about it.)

3.3   Performance Requirements

The application shall be based on web and has to be run from a web server. The application shall take initial load time depending on internet connection strength which also depends on the media from which the application is run. The performance shall depend upon hardware components of the users.

3.4   Security

- The users web browser shall never display a users password. It shall always be echoed with special characters representing typed characters.

- The users shall automatically log out all customers after a period of inactivity.

- The users back-end databases shall be encrypted.

3.5   Design Constraints

3.5.1   Standard Development Tools

The system shall be built using a standard web page development tool that conforms to either IBMs CUA standards or Microsofts GUI standards.

3.5.2   Web Based Product

There are no memory requirements. The computers must be equipped with web browsers such as Internet explorer. The application must be stored in such a way that allows the user easy access to it. Response time for loading the application should take no longer than five minutes. A general knowledge of basic computer skills is required to use the application.

4   Future Scope

The web application can be further modified depending on customer requirement on which other subject tests will be conducted and also the option whether to conduct multiple correct MCQ questions.