**Software Requirement Specification** for

Web-based Assignment Submission System

**Author:** Uttam Yadav

**Table of Content**

1. Requirement Analysis
   1. Requirement analysis and Functional requirements
      1. Student
      2. Teacher
   2. Non-functional Requirements
2. Requirement Specifications
3. Hardware Requirements
4. Software Requirements

**Requirement Analysis for an Online Assignment Submission System**

**1. Requirement analysis and Functional requirements**

The functional requirements will be detailed as following: **1.1 Student**

o The student requests the OSS site from web server using internet browser.

o The student inserts his username and password at log in area in the login Page.

o If the student inserts a valid username and password he or she will see the main menu that contains the following items:

1. Name information.

2. Personal information: personal number, name, email and password.

3. Course information: course name, number, point, semester and related assignments for each course

**Functional requirements**:

o Student can update his personal data.

o Student can upload the solution of any of his assignments before the deadline (if strict).

o Student will receive a message from the system after uploading the solution (accept or reject).

o Student gets feedback from the teacher.

o Student can see the information about the assignment (description and instruction, start-time, end-time, motivation, how to download and upload the assignment and type of work: individually or as groups)

**1.2 Teacher**

o The teacher requests the OSS site from web server using internet browser.

o The teacher inserts his username and password at login area in the login Page.

o If the teacher inserts a valid username and password he will see the teacher main menu that contains the following items:

* Name details.
* Address information: personal number, name, and email

**Functional Requirements:**

* Teacher can administer (add, delete, and update) courses and oversee students using a simple XML (Extensible Markup Language) file.
* In this XML file the teacher can add
  + Assignment information (description and instruction, start-time, end-time, motivation, how to download the assignment and upload it, type of work: individually or as groups).
  + Login information (accounts for the students
* Teacher can see the assignments submitted by students. He can assess the students by browsing a report of the assignments and students.
* Teacher can manually send feedback, marks and notes to students.

**1.2 Non-Functional Requirements**

* Safety, the system must have the ability to prevent illegal or incorrect operations from teachers or students by using certain tools such as validation control.
* Understandability, which makes it easy for students to use and deal with, user-friendly by developing a good interface and data accessibility, must be easy.
* Secure and private.
* The system developed for the purpose of supporting integration between existing and future systems.
* Accessibility, the system is available via the Internet and can be accessed any time and any place by the internet.
* Performance, the system must be fast.

**2. Requirement specifications**

This section explores all the project functions in details.

**1. The user requests the OSS home page from web server.**

**Function:** User requests the page from web server.

**Description:** This function provides ability to browse the OSS.

**Input:** The OSS URL.

**Source:** User.

**Output:** OSS home page**.**

**Destination:** Web server.

**Require:** Insert correct site address.

**Pre-condition:** Availability of internet service.

**Post-condition:** Displaying the OSS home page.

**2. Login.**

**Function:** Login.

**Description:** Enabling the student, teacher and administration to access his account,see his data and use menu by using valid username and password. **Input:** Username and password.

**Source:** Student, teacher.

**Output:** The student or teacher home page**.**

**Destination:** Web server.

**Require:** Valid login by correct username and password and having an account onthe OSS.

**Pre-condition:** OSS Home Page.

**Post-condition:** Connect student or teacher to OSS home page.

**3. Display Student profile*.***

**Function:** Displaying the student information.

**Description:** By clicking the profile link from the main menu a new page will bedisplayed containing the student information.

**Input:** Clicking my profile link from student main menu.

**Source:** The student and student menu web form.

**Output:** The student profile**.**

**Destination:** Web server.

**Require:** Valid log in and single click on my profile item.

**Pre-condition:** Home Page and no student displayed in main menu.

**Post-condition:** The student can see his profile.

**4. Update student profile.**

**Function:** Update student data

**Description:** The student can update his information (name, email and password). **Input:** By clicking my profile link from student main menu and entering somespecific information to the OSS and saving them in the web server. **Source:** The student and student menu web form.

**Output:** Save student data in the OSS system.

**Destination:** Web server.

**Require:** Valid login and single click on update **–**my- profile item.

**Pre-condition:** Student Home Page + having an account on the OSS.

**Post-condition:** Save data in web server (Student XML file).

**5. See the assignment.**

**Function:** Student can see the information about the assignment (description andinstruction, start-time, end-time, motivation, how to download the assignment and upload it, type of work individually or as groups).

**Description:** By clicking the courses link from the main menu a new page will bedisplayed, containing courses and related assignments.

**Input:** Clicking my courses link from student main menu.

**Source:** The student and courses link from menu web form.

**Output:** Courses and related assignments**.**

**Destination:** web server.

**Require:** Valid log in and single click on courses link.

**Pre-condition:** Student Home Page and registered student courses.

**Post-condition:** Courses assignments displayed.

**6. Registration courses for students.**

**Function:** Registration of courses for students.

**Description:** This function Provides administrators with the ability to registercourses for students that offered at the beginning of current semester.

**Input:** Click on registration link from administrator menu and choose the coursenumber and student number and saving them in the web server.

**Source:** Administrator and registration of web form.

**Output:** Save student registration.

**Destination:** Web server.

**Require:** Valid login and single click on the registration link**.**

**Pre-condition:** Select course and student number.

**Post-condition:** Displaying student and registered courses then save it in registrationxml file.

**7. Teacher can add, delete and update courses.**

**Function:** Add, delete and update courses.

**Description:** This function Provides administrator the ability to insert, delete andupdate courses.

**Input:** Click on insert course link from administrator main menu and fill coursename, course number, course credit, course level and the course type text, or click on update courses link to update course or delete it from courses xml file. **Source**: Administrator and insert new course web form or update course.

**Output:** Save the course and its information in courses xml file.

**Destination:** Web server.

**Require:** Valid login and single click on courses link (insert or update).

**Pre-condition:** Admin Home Page and fill all textboxes with valid data type.

**Post-condition:** Save courses in simple course xml files or update it.

**8. Teacher can add, delete and update Students.**

**Function:** Add, delete and update Students.

**Description:** This function Provides administrator the ability to insert, delete andupdate Students.

**Input:** Click on insert Students link from administrator main menu and fill Studentsname, ID, password and the email text, or click on update Student link to update Students or delete students from courses xml file.

**Source**: Administrator and insert new Students web form or update Students.

**Output:** Save the Student and his information in Students xml file.

**Destination:** Web server.

**Require:** Valid login and single click on Students link (insert or update).

**Pre-condition:** Administration Home Page and fill all textboxes with valid data type.

**Post-condition:** Save Student in simple student xml files or update it.

**9. Teacher can add, delete and update assignments.**

**Function:** Add, delete and update assignments.

**Description:** this function Provides administrator the ability to insert, delete andupdate assignments.

**Input:** Click on insert assignments link from administrator main menu and fillassignments number, Course number, description, instruction, start time , end time, motivation, type of work, is strict and the URL text, or click on update assignments link to update assignments or delete students from assignments xml file.

**Source**: Administrator and insert new assignments web form or update assignments.

**Output:** Save the assignments and its information in assignments xml file.

**Destination:** Web server.

**Require:** Valid login and single click on assignments link (insert or update).

**Pre-condition:** Admin Home Page and fill all textboxes with valid data type.

**Post-condition:** Save assignment in simple assignment xml files or update it

**10. Generate reports**

**Function:** Generate reports

**Description:** Teacher can see the assignments submitted by students. He can assessthe students by browsing a report of the assignments and students.

**Input:** Click on report link from administrator menu.

**Source:** administrator and report web form.

**Output:** Report on student data and evaluation of home assignments.

**Destination:** Web server.

**Require:** Valid login and single click on reports link**.**

**Pre-condition:** Admin Home Page.

**Post-condition:** Displaying students

**4. Hardware Requirements**

Processor - Intel Core i3,i5,i7

RAM - 4GB(64-bit OS) minimum

Hard Disk - 256 GB minimum

Input Devices - Mouse, Keyboard

**5. Software Requirements**

Windows/Linux

Browser supporting JavaScript

Command Line

A Word Editor specifically VS Code.

**6. Technologies Used**

**For Front End -** HTML, CSS and JavaScript.

**For Back End -** Node.js,Express.js

**For DataBase -** MongoDB