# Software requirements and design specifications

FU Kuo-hao          OUYANG Xiating          ZHOU Yi

14112466D          14111773D          14109328D

November 13, 2016

# CONTENTS

# 1 SOFTWARE REQUIREMENT SPECIFICATION

## 1.1 INTRODUCTION

This document will include all requirements and functionalities of Edo, a course management system for the Edward University of Smartness (EUS). This document will summarize the specifications and requirements obtained from Edward, who is the president of EUS, and our proposed design of Edo.

### 1.1.1 DEFINITIONS

The following abbreviations are used throughout this document.

1. Course content: the slides, notes or other course related files

2. Assessment: assignment and quiz

3. User: students and professors

4. File: any document uploaded to Edo, including course slides, note, assignment instruction etc.

### 1.1.2 GOALS AND OBJECTIVES

This project aims at developing an online course management system (Edo) for Edward University of Smartness (EUS). It offers EUS a lightweight platform which is supposed to help teachers distribute their course content and assess the performances of their students, while students can have easy access to course resources and finish assignments or quizzes on this platform.

### 1.1.3 STATEMENT OF SCOPE

1. Description

   Edo is a course management system between professors and students where professors can upload assignments and set up quizzes and students can upload their assignment answers and complete quizzes. After the quizzes and assignments are assessed, the grades can be given to students by professors and students may login their accounts to view their grades.

2. Major inputs

   Major inputs include course resources, such as notes, assignment files and quiz questions from teachers and assessment submissions from students.

3. Processing Functionality

   Edo should be able to store the uploaded file and course assessment information and return the corresponding results, such as the desired course content and attempted grades correctly and efficiently.

4. Outputs

   Course contents are returned in the form of a web link displayed on the screen. The grades for each assignment and quiz are also available. If a user intend to download a file from Edo, the correct file should be returned.

5. Functionality priority

   - Essential

     – Correctness, all files should be returned correctly and no wrong file or content should be returned

     – Basic security: We guarantee the integrity of the users and confidentiality of data processed.

     – Availability, the system should be available 24/7, as the students and professors should be able to access Edo anytime for uploading and viewing files.

   - Desirable

     – More security issues: the system should be immune from DDoS attack; the system should be able to detect malicious accesses. The system allows users to upload files from third-party storage systems like Google Drive, Dropbox, Onedrive, etc.

     – Customizability: The users may choose to customize their personal homepage when displaying the course information.

   - Future

     – Speed: the files should be uploaded and returned at at least 500Kb/s since otherwise a student might miss the deadline due to the low transmission speed.

### 1.1.4 SOFTWARE CONTEXT

Edward is the president of a newly founded university, Edward University of Smartness. EUS has recruited professors and admitted undergraduate students, and it needs a learning platform for professors to post assignments and quizzes, and for students to upload assignments and complete quizzes. Therefore, Edward would like to hire our programming team to develop a course management system Edo, for EUS.

### 1.1.5 MAJOR CONSTRAINTS

The development of Edo is mostly limited by the time of delivery since the EUS will start its first semester in 2017 Fall and the system is in urgent need. Therefore this project will only deliver the core functionality of Edo. After delivery, Edo will be further refined in future endeavors.

## 1.2 USAGE SCENARIO

### 1.2.1 USER PROFILES

- The students of EUS: This user will have the access to his/her registered courses and view all the related course contents. They can also submit the assignment answers and answer quiz questions posted on Edo.

- The professors (and Teaching assistants) of EUS: This user will have access to create the course information in Edo and upload corresponding course materials, including course note, course assignment and quizzes on Edo for students' reference.

### 1.2.2 USE-CASES

- Use-Case Descriptions [1]

  Student/Professor logs in

    1. The user enters his/her username and password, and clicks the enter button.

    2. The user logs in Edo and enter his/her personal homepage.

  Student view course

    1. The student clicks a course displayed on the personal homepage.

    2. The course notes, course assignment and course quizzes are shown.

  Student view course content

    1. The student clicks one course note, and the course note is downloaded.

  Student answer assignment

    1. The student clicks one assignment, and the assignment introduction and guide is displayed.

    2. The student upload his answer to the assignment to Edo.

    3. The student clicks finish button to finish, and returns to course information page.

  Student answer quiz

    1. The student clicks one course quiz, and the quiz questions are displayed.

    2. The student chooses one of the four choices for each quiz question by ticking the box ahead of each choice, and clicks submit button.

    3. The student clicks the completed assignment and quiz, the grades, if available, will be shown to them.

  Professor View course

    1. The professor enters his/her username and password, and clicks the enter button.

    2. The professor enters the personal homepage, and clicks on a course he teaches.

---

[1]The later usecases are presumed to be performed while staying logged in.

Professor create assignment

1. The professor clicks the create content button, and chooses one file from his/her PC for upload.

2. The professor clicks the create assignment button, and enters the assignment information and chooses a file for that assignment.

3. The professor clicks the submit button, and the assignment is created.

4. The professor clicks the assignment submission button to view all assignment submissions, and give grades to those submissions by entering the grades in the grade bar.

Professor create quiz

1. The professor clicks create quiz.

2. The professor enters the name of the quiz.

3. The professor sets up one MC question.

4. The professor enters the description of the MC question.

5. Then, the teacher enters the four options for the MC question.

6. The professor sets up the correct answer.

7. The professor continues 9-12, until all questions are set.

8. The professor clicks "finish" button, and the quiz is created and henceforth posted.

- Use-Case Diagram

Figure 1.1: Use case diagram

### 1.3 DATA MODEL AND DESCRIPTION

#### 1.3.1 DATA OBJECTS

The users will have username and password for authentication. After login, their first name and last name will be displayed on the homepage. For professors, they additionally have titles and institutions, which are useful in future refinements.

A course has course contents, assignments and quizzes. Contents will be simple .doc or .pdf files, and an assignment will contain a description and a supplementary file, if deemed necessary. Quizzes will have multiple MC questions with four choices each. Each quiz question will have one correct answer. Students will submit their quiz or assignment submission, which may contain a supplementary file if deemed necessary. Grade is either a floating point value or A, B, C, D, F (and their subgrades), and is associated with assignment and quiz submissions.

One student may take multiple courses, and one professors may teach multiple courses. For each course, it has multiple course contents, courses assignments and course quizzes. Each quiz has multiple questions and each question will have four choices. Each assessment will have multiple submissions from students, and each student submission will have only one grade.

### 1.3.3 COMPLETE DATA MODEL

Figure 1.2: Data Model of Edo

## 1.4 BEHAVIORAL MODEL AND DESCRIPTION

### 1.4.1 DESCRIPTION FOR SOFTWARE BEHAVIOR

- Events
  - Student/professor login: this event will guide the students or professors to their homepages.
  - Click on course: click on the course, content, assignment, quiz and grade.
  - Click on course content
  - Click on course assignment
  - Click on course quiz
  - Click on assignment/quiz grade
  - Submit assignment
  - Submit quiz
  - Create assignment
  - Create quiz
  - Create content

- States
  - Logged off state
  - Student logged in state
  - Professor logged in state
  - View course content state: course content including notes, assignments and quizzes are displayed.
  - View notes, assignment, quiz states: the corresponding content is displayed.

- Do assignment, quiz states: the answers are stored by the system and ready for upload.

- Upload content, assignment, quiz state: the content, assignment and quiz is being prepared and completed by professors and is ready for upload.

- Grade: the professors are giving grades to the submitted assignment. [2]

- View grade: the students may view their attempted grades of assignments and quizzes.

### 1.4.2 STATE-CHART DIAGRAM

Figure 1.3: State Chart Diagram of Edo

## 1.5 RESTRICTIONS, LIMITATIONS, AND CONSTRAINTS

The size of files uploaded should not exceed the maximum size (5MB so far), otherwise the server will not be able to store all of the files.

The students are expected to complete the course and they cannot drop the registered course half way.

The course list will be retrieved from the course registration system. Edo only serves as a platform to share teaching materials and post assessments.

Edo is highly dependent on network performance, and thus the response time is not well guaranteed.

# 2 SOFTWARE DESIGN SPECIFICATION

## 2.1 ARCHITECTURAL AND COMPONENT-LEVEL DESIGN

This section will focus on the design of Edo in terms of its components, and it is shown through its class diagram.

### 2.1.1 CLASS DIAGRAM

Figure 2.1: Class Diagram of Edo

---

[2]Quizzes are auto graded since each quiz question is set with a correct answer.

## 2.2 Description for Components

The following components together perform the essential functionalities of Edo. All components rely on and are triggered by the user event handler, which is a widely used web component. Therefore web components are omitted for the component interface part.

### 2.2.1 Login Component

1. Processing narrative for component

   This component will be responsible for user login. The user enters the username and password, and if they match, the user is thereafter logged in.

2. Components interface description

   This component will have an interface with the student account system of EUS to validate the usersname and password since the student account information is only stored in the student account system.

3. Components processing detail

   - Restrictions/limitations for component

     This component does not hold the specific login information as duplication of student accounts may bring security issues.

   - Performance issues for component

     This component should be correctly implemented and run securely since it involves privacy issue. Therefore the performance overhead might be large since it involves security and network transmission.

   - Design constraints for component

     Login is largely dependent on the transmission between Edo and the account system, and thus the responsiveness may not be guaranteed. The username and password will be limited to a maximum of 32 characters.

   - Processing detail for each operation of component

     First the user will enter the user name and the password, and then this component will forward the username to the account information system for password. When the reply from the account information system is received, this component will compare the passwords to decide if this authentication is successful or not. If successful, the user will be directed to his/her homepage, otherwise the user will be prompted to enter the password again.

### 2.2.2 Course retrieval

1. Processing narrative for component

   This component will retrieve the course registration information from the registration system as the semester begins and initialize the database of assignments, notes and quizzes for each course.

2. Components interface description

This component will retrieve the courses taken/taught by the users and use that information to initialize the database.

3. Components processing detail

- Restrictions/limitations for component

This system is run once per day during the add/drop period, and will be disabled after the add/drop period. Therefore the students are strictly not allowed to drop any course from the course registration system after the add/drop period.

- Performance issues for component

This system also greatly rely on the network performance, and the data to be transmitted is large in size. Therefore the students may not get timely updates.

- Processing detail for each operation of component

Once per day during the add/drop period, this component will retrieve the course registration information from the registration system and compare it with its current course database to delete those out-of-date course-student relationships stored in the resource management component.

### 2.2.3 Resource Upload

1. Processing narrative for component

This component will handle the uploading request made by professors and students.

2. Components interface description

This component will send the uploaded file to the resource management system for storing purpose.

3. Components processing detail

- Restrictions/limitations for component

This component is sensitive to the input file size, and the size of each file should be limited to maximum 5 MB.

- Performance issues for component

The performance might decline as the number of files to be submitted grow.

- Processing detail for each operation of component

The user chooses one file to be uploaded, and clicks the submit button. This component will forward the file to the resource management component, and redirect the user back to the course information page.

1. Processing narrative for component

   This component stores all the files and information that a professor uploads and a student submits, and maintains the link between the students and their registered courses.

2. Components interface description

   This component will have an interface with the user interaction component to exchange data with the users.

3. Components processing detail

   - Restrictions/limitations for component

     This component is sensitive to the input file size, and the size of each file should be limited to maximum 5 MB.

   - Performance issues for component

     The performance might decline as the number of files grow.

   - Processing detail for each operation of component

     When the user uploads a file to assignment or content, this component will record the label of that origin and associate the uploaded file with the fingerprint of that origin. Later when the document is requested by the user, this component will retrieve the file using the origin and return it to the user.

## 2.3 INTERFACE DESIGN

### 2.3.1 EXTERNAL MACHINE INTERFACE

Edo will be an online platform and thus the users are required to use the browser and enter the URL or Edo to access this system. The physical machine could be either PC or tablets.

### 2.3.2 EXTERNAL SYSTEM INTERFACE

Edo will be exchanging data with the student account system and the web system that carries out the data transmission.

### 2.3.3 DESCRIPTION OF THE USER INTERFACE

This prototype of the Edo user interface is shown as below.

Figure 2.2: User interface

### 2.3.4 Interface design rules

The interface should have high clarity to facilitate the use of Edo. In each page, only the navigation bar and the corresponding content shall be visible. E.g. in course information page, only the course notes, course assignment and couse quiz will be displayed, but in course note, course assignment and quiz will not be shown.

### 2.3.5 Interface Components available

Edo will be put online and thus all the web based components, such as bootstrap, javascript and angular js are all available for development.

## 2.4 Restrictions, limitations, and constraints

The design of Edo is very compact that the interface may not look quite fancy. It only supports a few core functionalities so far, and more can be added in the future evolution process.

This physical design does not include the network layer components, and is dependent on the network performance.

## 2.5 Testing Issues

### 2.5.1 Classes of tests

1. Login: Entering the username and password wrongly.

2. Creation: create an assignment or quiz by entering the name and description.

3. View: view an assignment by clicking on the assignment title.

4. Download file: Edo should return the correct file to be downloaded to the students.

5. Submission of assignment and quiz: submit a file or quiz response to the system.

6. Invalid upload file: The file to be uploaded to Edo is not supported.

### 2.5.2 Expected software response

1. Login: An error message shall pop up and the user to be directed to login again.

2. Creation: the assignment, content and quiz should be created and viewable in both professors' and students' homepages.

3. View: The students can see the courses and the related materials in their homepages.

4. Download file: The correct file that the student desires should be downloaded to the client side.

5. Submission of assignment and quiz: the file is submitted to Edo and professors can see the submission from Edo.

6. Invalid upload file: An error message shall pop up and redirect the user to upload again.

### 2.5.3  PERFORMANCE BOUNDS

Since most of the functionalities are dependent on network performance, quick response is not guaranteed. Efforts will be spent on ensuring the turnaround time of each functionality is bounded by 2 seconds.

### 2.5.4  IDENTIFICATION OF CRITICAL COMPONENTS

The critical components are the login component and resource management component. Login component should ensure the overall security of the system, and the resource management component is responsible for efficient material retrieval and delivery process. The resource management component shall index the uploaded materials properly to achieve high efficiency.