**PE SWR302**

**Q1:**

**Software Requirements Specification**

**for**

**FU Lab Pratices website**

**Version 1.0 approved**

**Prepared by**

**Trần Xuân Tiến, SE161471**

**FPT University Campus Ho Chi Minh**

**Monday, July - 24 - 2023**

**Q2:**

1. In this exam paper, the name of the rectangle is: FU Lab Practices

2. The actor may be human, other software systems or devices. In this exam paper, list the name of >= 4 actors is:  
 - Admin  
 - Student  
 - Teacher  
 - System  
 - Guest

3. In this exam paper, list the name of >= 5 use cases are:  
- Lấy 5 cái từ đống use cases trên

4. The actors are inside or outside of the rectangle: Outside

5. The ovals represent the use cases are inside or outside of the rectangle: Inside

6. The primary actor of the use case add course is: Administrator

7. The secondary actor of the use case add course: Student & Teacher

8.Based on the fact you already used coursera. The use cases in this exam paper may have relationship.

8.a List the name of two use cases that have extend relationship: View Course List, View Course Details

8.b The name of base use case (in 8.a) is: View Course List

8.c The name of extends use case (in 8.a) is:  View Course Details

8.d The dashed line with an arrow that points to which use case:  View Course List (tự hiểu là mũi tên extends luôn trỏ về base use case, không phải là extends nên đáp án là View Course List)

9.   Based on the fact you already used Coursera, the use cases in this exam paper may have relationship.

9.a List the name of two use cases that have include relationship: Peer-review, Submit Review Comments

9.b The name of based use case (in 9.a) is: Peer-review

9.c The name of include use case (in 9s.a) is: Submit Review Comments

9.d The dashed line with an arrow that points to which use case?  Submit Review Comments  (tự hiểu là mũi tên includes luôn trỏ về base use case, không phải là extends nên đáp án là View Course List)

**Q4.**

1. Performance:

* **Response Time:** The system should respond to user interactions within 2 seconds.
* **Throughput:** The system should support a minimum of 1000 concurrent users.
* **Latency:** The system's communication latency between components should not exceed 50 milliseconds.

2. Reliability:

* **Availability:** The system should be available 99.9% of the time.
* **Fault Tolerance:** The system should continue functioning properly even in the presence of hardware or software failures.
* **Recoverability:** The system should be able to recover data and resume normal operation within 1 hour after a failure.

3. Security:

* **Authentication:** Users must be authenticated using two-factor authentication before accessing sensitive data.
* **Data Encryption:** All sensitive data should be encrypted both in transit and at rest.
* **Access Control:** Different user roles should have appropriate access privileges to system features and data.

4. Usability:

* **User Interface Consistency:** The user interface should follow the company's branding guidelines and maintain a consistent design.
* **Accessibility:** The system should comply with accessibility standards (e.g., WCAG) to ensure usability for users with disabilities.
* **User Training:** The system should be intuitive and require minimal user training to operate.

5. Scalability:

* **Vertical Scalability:** The system should be able to handle a 50% increase in load by adding more resources (e.g., CPU, memory).
* **Horizontal Scalability:** The system should support adding additional nodes to the cluster to handle increased load.

6. Maintainability:

* **Modularity:** The system's components should be organised into separate modules to facilitate easy maintenance and updates.
* **Code Documentation:** All code should be well-documented with comments and clear explanations of functionality.
* **Extensibility:** The system's architecture should allow for easy integration of new features and functionalities.

7. Compatibility:

* **Browser Compatibility:** The system should be compatible with the latest versions of popular web browsers (e.g., Chrome, Firefox, Safari).
* **Platform Compatibility:** The system should run on Windows, macOS, and Linux operating systems.

8. Performance Efficiency:

* **Resource Utilisation:** The system should use no more than 50% of available system resources under typical load conditions.

**Energy Efficiency:** The system should minimise energy consumption to reduce environmental impact.

**Q3.**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and Name** | **UC-1: Submit Project** | | |
| Created by | Nguyen Van Tu Cuong | Date Created | 12/11/2023 |
| Primary Actor | Teacher | Secondary Actor | Teacher |
| Description | Students have the ability to utilise the FU Lab Practices platform to submit their project to their teachers, categorised under relevant subjects such as Java, Python, C#, and more. Subsequently, teachers can review the submitted code and assess the students' results. | | |
| Trigger | A student indicates to submit a project | | |
| Preconditions | PRE1 - Student is logged in FU Lab Practices  PRE2 - The project is either stored on the student's local computer or has already been uploaded to GitHub. | | |
| Postconditions | POST1 - The status of the student submission is marked as "Done."  POST2 - The teacher is able to access and review the student's project. | | |
| Normal Flow | **1.0. Submit project from local computer**  1. Student clicks on Submit button (see 1.0 E1)  2. FU Lab Practices presents a modal that enables students to choose a specific file path.  3. The student selects the project path (e.g., /D/Cuong/PRU211) and clicks the Confirm button. (see 1.0 E2)  4. FU Lab Practices displays an alert to the student, asking for confirmation.  5. The student clicks on "Yes."  6. The project is successfully submitted. | | |
| Alternative Flow | **1.1. Submit project from GitHub**  1. Student clicks on Submit button (see 1.0 E1)  2. FU Lab Practices presents a modal that enables students to input a GiHub repository URL.  3. The student enters a URL that points to their project.  4. FU Lab Practices will find the repository, then display an alert to the student, asking for confirmation.  5. Return to step 5 of normal flow. | | |
| Exceptions | **1.0 E1: The deadline has passed.**  1. FU Lab Practices will notify the student that they have missed the deadline and are unable to submit.  **1.0 E2: Project exceeds 10MB.**  1. FU Lab Practices will inform the student that the file size exceeds 10MB and submission is not possible.  2a. If the student cancels the submission process, the use case is terminated.  2b. Otherwise, if the student requests another file, FU Lab Practices restarts the use case. | | |
| Priority | **High** | | |
| Business Rule | **BR-1.** The project cannot exceed 10MB. | | |
| Other informations | The student has the ability to cancel the upload process at any time of their choosing. | | |
| Assumption | The student's internet connection remains stable while uploading the file. | | |

1.1 UC-01-Register

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | **UC-01-Register** | | |
| Created By: | PhucLNH | Date Created: | 11/11/2023 |
| Primary Actor: | Guest | Secondary Actors: | None |
| Trigger: | Actor decides to register on the website | | |
| Description: | The process of a actor registering on the website | | |
| Preconditions: | * The actor has access to the website * The actor has a valid email address | | |
| Postconditions: | * Actor register the system successfully * The system tracked successful register into the Activity Log | | |
| Normal Flow: | 1. The user navigates to the registration page. 2. The system presents a registration form. 3. The user enters their details, including username, password, email address, and other required information. 4. The system validates the entered information. 5. The system validates the entered information. 6. If validation is successful, the system creates a user account and notifies the user of successful registration. 7. The user can now log in using their registered credentials. | | |
| Alternative Flows: | None | | |
| Exceptions: | * If the entered username already exists, the system prompts the user to choose a different one. * If the entered email address is not valid, the system prompts the user to enter a valid email address. * If there are any server errors during registration, the system displays an error message and prompts the user to try again. | | |
| Priority: | Must have | | |
| Frequency of Use: | Undeveloped | | |
| Business Rules: | * A user must have a unique username within the system. * Passwords must meet minimum security requirements. * Email addresses must be valid and unique within the system. | | |
| Other Information: | None | | |

2.1 UC-04-Login

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | **UC-04-Login** | | |
| Created By: | PhucLNH | Date Created: | 11/11/2023 |
| Primary Actor: | Registered User | Secondary Actors: | None |
| Trigger: | User clicks Login button from the page header, or  User accesses an authenticated feature (from a link or type the page URL directly into the address bar) | | |
| Description: | As a actor, I want to be able to log into the system so that I can use the system’s authenticated features and access my personalized account. | | |
| Preconditions: | Actor account has been created & authorized | | |
| Postconditions: | * Actor logs in the system successfully * The system tracked successful login into the Activity Log | | |
| Normal Flow | **2.0 Login System**  1. User accesses the User Login screen  2. User types in the login details or choo other login options  3. User clicks the Login button  4. System validates the login details  5. System allows user to access  6. System tracks user’s success login to the Activity Log  7. System accesses the Home Page (or the previous calling page if any) | | |
| Alternative Flows: | ***2.1 Google Login***  1. User chooses to login system using Google account  2. System redirects the user to the Google’s Login screen  3. User types in the Google account details and chooses to login  4. Google validates user’s login information successfully and redirect him/her back to the system  5. Return to step 5 of normal flow. | | |
| Exceptions: | * If the entered username or password is incorrect, the system notifies the actor and prompts them to enter valid credentials. * If the actor's account is locked, the system informs the actor and provides instructions for unlocking the account. * If there are any server errors during the login process, the system displays an error message and prompts the actor to try again. | | |
| Priority: | Must Have | | |
| Frequency of Use: | Undeveloped | | |
| Business Rules: | * Actors must have a registered account to log in. * Passwords must meet minimum security requirements. * After a certain number of unsuccessful login attempts, the account may be temporarily locked for security purposes. | | |
| Other Information: | None | | |
| Assumptions: | None | | |