Teach-Me Application

Aayushee Dave
School of Computer Science
University of Windsor
Windsor, Canada
dave71@uwindsor.ca

Manan Parmar
School of Computer Science
University of Windsor
Windsor, Canada
parmar6@uwindsor.ca

Rahul Pandya
School of Computer Science
University of Windsor
Windsor, Canada
pandya51@uwindsor.ca

Richa Gupta
School of Computer Science
University of Windsor
Windsor, Canada
gupta14h@uwindsor.ca

***Abstract*—This document gives an overview about the progress made in the second phase - Elaboration phase of the project including the objectives, deliverables, issues along with its mitigation plan and test strategies.**

I. Objectives

- To create an online system aimed to teach students a particular subject.
- The application should be compliant to student's ability and should evaluate them accordingly.
- The application should give feedback about the student's performance.
- The system should be robust and durable.
- It should easily accessible to authorized personnel and restrict access to unauthorized personnel.

II. Deliverables

- Source code of project containing Controllers, APIs, Services, Entities.
- Source code of project containing various UI modules: Login, Student/Tutor Dashboard, Student/Tutor Test page.
- Modified Database Design Document
- Test Plan document Version 2.0

III. Issues and their mitigation plan

The issues and risks examined during the Elaboration phase along with their mitigation plan are highlighted below:

A. JSON data parsing

One of the concern raised while integration of front end and back end was JSON data parsing. It was due to format of the data returned from back-end was differing to the format of data that was required in front-end. To mellow down this issue, Data Transfer Object have been used to bridge the gap between data obtained from back-end and the data required in front-end.

B. Security

Security is the prominent factor while using any application. As there were number of options available to ensure security, we analysed all the approaches and their drawbacks. Considering approach and their concerned drawbacks, we found that Spring Security would be best to implement. However, it has 3 ways to implement - Spring Security Kerberos, Spring Security OAuth and Spring Security SAML. Hence, decision is pending yet to finalize the approach from above 3 mentioned ways.

C. API for parsing Excel file

Parsing of excel file is required to upload any test and this impacts the performance of the application with the varying volume of excel file size. Hence, after empirical research, we decided to use Apache POI to parse the excel file, that will follow stream-based processing and is best fit with all kind of file size, also consumes comparatively less memory.

IV. Test Strategies

Test strategy for Teach-Me application is as follows:

- Creating a test plan that includes all important entities i.e. scope, testing methodologies to be followed, test deliverables and resource and environment needs.
- Creating/developing the test cases/test scripts as per the defined scope.
- Execute the tests and generate reports based on test results.
- Log bugs if any found with detailed information.
- Defect management: It will include fixing the logged bugs and their verification after the fix is done. Parallelly, tracking the overall status of bug until it is resolved.
- Delivering the feature.

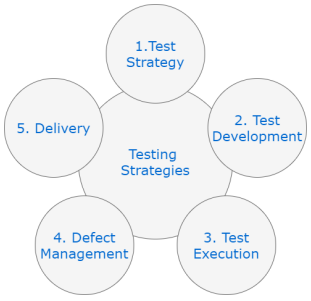

Fig. 1. Testing Strategies

Based on the changes in this iteration, modifications in the test plan document were also required. Please refer below links for respective test plans of this project:

- Test Plan Document Version 1.0.
- Test Plan Document Version 2.0.

V. Inception Phase Review

The following lists of tasks have been achieved during the Inception phase.

#	Status	Priority	Subject
3268	Closed	High	UML Diagrams
3131	Closed	Normal	Decide testing strategies
3128	Closed	High	Database definition
3127	Closed	High	Environment setup
3126	Closed	Normal	Miscellaneous tasks
3125	Closed	Normal	IEEE Manuscript creation
3124	Closed	Normal	Project architechture discussion
3123	Closed	High	Requirement Gathering
3122	Closed	Normal	Inception Phase Document Finalization
3121	Closed	Normal	Inception phase task - 2
3120	Closed	Normal	Inception phase task - 1

Fig. 2. Inception Phase Tasks

VI. Elaboration Phase

Following are the tasks that we have determined to complete at the end of Elaboration Phase.

- Front End: Login, Student and Tutor Dashboard, Student and Tutor Test page,
- Back End : Creating Entities, repositories, Services, Controllers and APIs.
- Integration of front and back end
- Testing - User Interface Testing, API Testing, Unit Testing of Services, Repositories and Controllers

#	Status	Priority	Subject
3436	In Progress	Normal	Testing - Unit Test of Services
3435	In Progress	Normal	Testing - Unit Test of Controllers
3434	In Progress	Normal	Testing - Unit Test of Repositories
3433	In Progress	Normal	Testing - User Interface
3421	Closed	Normal	IEEE Manuscript Updation
3414	New	Normal	Performance Testing
3411	New	Normal	Add unit tests for APIs
3410	New	Normal	API Testing
3370	New	Normal	Integration of Front end and Back end
3369	New	Normal	Front end - Report Module
3368	Closed	Normal	Front end - Login Module
3367	Closed	Normal	Front end - Test Module
3366	Closed	Normal	Front end - Upload Test Module
3365	Closed	Normal	Front end - Student Dashboard Module
3364	Closed	Normal	Front end - Tutor Dashboard Module
3363	Closed	Normal	Front end - Discussion of UI screens
3362	New	Normal	Back end - Creating APIs
3361	Closed	Normal	Back end - Creation of Controllers
3360	Closed	Normal	Back end - Creation of Services
3359	Closed	Normal	Back end - Creation of Repositories
3358	Closed	Normal	Back end - Creation of Entities

Fig. 3. Elaboration Phase Tasks

VII. Product Documentation Draft

Below is the Product Document Draft for Elaboration Phase where we have finalized the Inception Artifact Document, Project Architecture, Database Design, Environment/VCS Setup, Project Management/Tracking tool setup.

A. Inception Artifact Document

For Artifact Document, please visit Inception Artifact Document

B. Project Architecture

A good architecture can cover most of the non functional requirements. Designing it was the most crucial task of this phase. We have finalized monolithic design for our application where the back-end application consists of REST APIs with proper authorization and the front-end service will make request to these APIs. Please refer our Architecture Diagrams for more understanding.

The finalized technology stack for the project is given below:

- Back-end service: Java (Spring Boot)
- Front-end service: Angular 9
- Database: MySQL
- IDE: Eclipse, IntelliJ Idea, Visual Studio Code
- Database Client: MySQL Workbench
- Other Tools: Postman, SourceTree, Github

C. Database Design

For further references see Database Design

D. Environment/VCS Setup

For Source Code, please visit TeachMe App page for Back End Application, and UI Service page for Front End Application.

E. Project Management

For Project Management, please visit our RedMine page.

References

- Java Documentation: https://docs.oracle.com/javase/8/docs/api/overview-summary.html
- Baeldung Documentation: https://www.baeldung.com/spring-boot
- Spring Documentation: https://docs.spring.io/spring/docs/current/spring-framework-reference/overview.html
- Angular Documentation: https://angular.io/docs
- Testing Strategies:https://www.guru99.com/how-to-create-test-strategy-document.html