Final Report

for

Automated Software Engineering Student Subject Evaluations (ASESSE)

Version 1.0

Prepared by Tanaka Madyara, Deepthi Tummala, Samuel Weems



December 07, 2018

Table of Contents

Table of Contents 1

1. Introduction 2

1.1. Purpose 2

1.2. Overview of Document 2

2. Overview of Project Error! Bookmark not defined.

3. Outcomes 3

4. Lessons Learned 3

Summary 4

1. **INTRODUCTION**

**1.1 Purpose**

The purpose of this document is to provide an official detailed description of the outcomes for the “Automated Software Engineering Student Subject Evaluation” (ASESSE) software. This document seeks to fully describe the exact behavior of the software including all high level functions, capabilities and constraints. This document will be used as a reference to the customer.

**1.2 Overview of Document**

This document is divided into 4 major sections including the introduction to the document, overview of project, outcomes, and lessons learned. The introduction to the document provides the purpose of this document, and overview of this document. The overview of project provides the detailed description of the “Automated Software engineering Student Subject Evaluation” software. The outcomes section describes the problems occurred and their solutions, and problems remaining in the system. The lessons learned section describes what the team has learned by developing this project.

1. **Overview of Project**

The “Automated Software Engineering Student Subject Evaluation” (ASESSE) is a web-based application which automates the student peer review system for Software Engineering Students at Midwestern State University located in Wichita Falls, Texas. The main purpose is to allow students to review other students in their course group and to collate that information automatically for the professor.

Each student in a group should be able to connect to the website, enter a password specific to their course, and then provide the information requested with regards to their specific group members through a graphical user interface. The submitting user name and their review of their group members are collected, but are stored separately and disconnected from one another to maintain anonymity.

The professor, or administrator, should be able to setup new classes with their own specific passwords, change the passwords if needed, delete or clear a class that has been setup, check to see the progress of all students in completing their reviews, request an report for an individual student that aggregates the reviews of that student, and request a course aggregate report which puts all individual student reviews into one printable PDF document.

1. **OUTCOMES**

During the implementation phase we encountered several issues, as each team member is implementing different parts of the project. As the database used is same for all the team members, while developing the code for the project, we encountered an issue of inserting the data and retrieving the data from the database, as database connection is more important for the web based application.While communicating with other team members and creating the neat and clear database form will solve the issues that occurs regarding the database usage. Communication between the team members will rectify more issues as each member of the team member may be specialized in some part of the code or the language they are writing. During the development phase we have encountered several issues like, while entering the student data in the form and clicking the submit button the form is not inserting into the database, by correcting the database columns and table names the form is getting inserted into the database.

Each and every error or the problem encountered during the development or implementation phase will be rectified by communicating with the other team members and with the customer. But still we have some problems in the system, which we have not yet rectified. Some of them are, all the student side web pages are not password protected, we can directly enter into the student web page directly by entering the login page. Whenever the form is submitted the database first enters the comma and the form results into the table. There is no limit for the student submission, every student report can be submitted as many times as the students can submit. Styling and appearance of the web pages is also not good.

1. **LESSONS LEARNED**

Every member of the team has learned something different from developing this project. Lessons learned by every member of the team will be explained below.

While working on a team to create a software project, the team environment requires a lot of communication. Communication between the team members should be good and accurate. Lack of communication creates some disturbances between the team members and that may lead to failed project. Must make customer as our team member, as if we are making the project for the customer only. Communicating with the customer will help us in developing the project in correct manner and make the project success. Trust is needed in between the team members to develop the project by meeting the deadline. This project developed ability of each and every member of team to work with other people with different nature and interests.

As team contains the people of different personalities and talents. Each and every phase required for development of the project is divided equally in between the team members. Based on the interests of the team members, our team leader samuel weems divided the phases, so that every team member can involve in every phase of the development process.

A well designed database application is needed for developing the web based application. From tables you can design classes to create objects. From fields you can derive object member data types, object sizes, and operations to act on these members. A well designed database acts like a blueprint and provides structure for a program task.

During the development of this project we learned to use tools like slack, Visual Studio Code, and Git bash. We learned how to create schedule for which milestones needed to be reached, a project plan, program specifications, use case diagrams, testing plans, test report, final reports, and user manual. Also learned more about the database management, PHP, Java script.

If we have more time to develop the project, we would add reset button to the course information, add icons to the buttons instead of text, spend more time on styling, appearance, functioning on various browsers, devices. Fix the issues where the first review should not include a comma at the front of the entries, use a prototype model and begin implementation sooner, protect the student side pages, right now we can directly access the pages by typing the URL.

Creating the “Are you sure” danger message while trying to delete or remove the course, and adding more admin users capability. As in the student side all the groups in the database all listed instead of listing the groups that are in the course which is selected by the student to login.

**SUMMARY**

This report provides an official and detailed description about the development of the project. It actually explains what the project is used for, and it describes about the problems that were occurred during the development phase and testing phase, and the solutions we have used to rectify those errors will be explained. Also describes the problems that are still remaining in the system, and the lessons what we have learned while developing this project will be explained clearly, either the individual or the group experiences. Each and every member of the team will explain their experience regarding the development of the project in teams, and the changes they want to make on the project when they have given more time after the deadline or asked the team to redo the project in their own manner. Team communication and collaboration will be explained in their words.