Individual Homework and Team Project Overview

Product Vision

ASU Students come into CSE 360 through many paths. Some have significant programming experience, while others have just the required course to be admitted to the class. The goal of the class is to provide every member of the class with a solid foundational survey of Software Engineering principles and concepts, practical individual experience applying some of them, and a team experience in producing a realistic and complex application demonstrating a deeper understanding of a set of these principles, concepts, and best practices, as well as the ability to leverage that understanding and use it to work effectively as a member of a team.

One challenge that many face is getting useful answers to their questions. Using an Internet browser typically provides the student with a long list of potentially useful items. While the answer may be in the returned list, finding it is still not easy. Many of the returned items are designed to sell a product or service, and they do not provide the details needed. The items returned are usually sorted by the most popular, and popularity does not guarantee usefulness or even relevance to the question being asked. (E.g., Asking how to do a task in an application may return answers for various versions of the application that may not be useful for the version the student is using.) What is even more troubling is the growth in malware, and what may appear to be just what is needed has the student do things that facilitate opening that student's computer to an attack.

What is needed is a tool that enables those asking questions to shorten the returned list of items to just those recommended by a set of trusted reviewers.

We envision a question-and-answer system that starts by making it easier for CSE 360 students to ask questions, get potential answers to those questions, and specify whether answers work or don't work so others don't waste time trying answers that have not worked for others. Building on that base, we believe the system can be enhanced by adding the role of a trusted reviewer. These reviewers write reviews of answers to accomplish two goals. The first is to help those with questions quickly find a worthy answer. The second is to help those answering to improve their answers.

For the system to work, students with questions need to read reviews, identify reviewers that appear to be worthy of trust, and establish and manage a list of trusted reviewers. Students can then study all the answers to a question or only view those answers that one or more of their trusted reviewers have recommended.

Students writing answers can read the reviews that have been posted, identify reviewers that provide them with feedback that can help them improve the answers they provide, and establish and manage a list of trusted reviewers. Students writing answers can study all the reviews for their answers to a question or only view those reviews from a trusted reviewer.

The instructional team observes the actions of students and reviewers. When issues arise, the instructor coaches and mentors both students and reviewers to help them be successful. Staff identify potential issues to the instructor when they see problematic questions, answers, or reviews.

System administrators manage who has access to the system and the roles each user may play. It may well be that an instructor may also be an admin. A student may also be a reviewer. Different roles allow users to see different things and perform different actions.

The information provided in the homework and team project assignments does not cover all situations. Some of the information elements may be inconsistent with other elements, some elements may be wrong, and others may change over time. This is the nature of systems in the real world. As a software engineer, you are expected to work with an array of stakeholders, find common ground for a quality solution, and produce systems that can adapt to change.

Development Overview

The development of this system is divided into four phases. Each phase consists of a homework assignment followed by a team project assignment. The purpose of each homework is to ensure that each member of the team understands and starts to develop skills using the course's software engineering principles, concepts, processes, tools, and software components that will be used in the team project assignment that follows. You are expected to learn and use the materials covered in the prerecorded lectures and the other provided materials.

Teams are expected to use a Scrum standup **at least twice a week** over the period allocated to each team project assignment. Start on the team project assignment for each phase **immediately after** submitting the homework assignment. What each team member does will likely impact what others on the team are doing. The standups are a communication tool to help everyone understand the areas of overlap and risk. This allows the involved team members to work together, experiment, get help, and produce working solutions to their user stories well before the planned time to integrate all the work into a well-functioning deliverable.

During each development phase, the team **must produce** (and update if needed) the requirements (e.g., user stories), architecture, design, code, test cases, and demonstration to make it clear how the requirements flow gracefully through all the software development stages. Unless special arrangements are made, all four phases of the application development must be written with a Graphical User Interface using JavaFX.

The First Phase

You have been provided with foundational documentation and code that you are expected to understand, use, and update as part of your homework assignment and then for the team project assignment. A provided short screencast introduces you to finite state machines, how they can be used as a practical design tool, and how that design can be easily implemented as Java code. In addition, a working standalone JavaFX program has been provided implementing the finite state machine and demonstrates the kind of internal documentation your projects should provide, as well as some of the features of modern user interface design stakeholders want to see.

The first phase focuses on establishing a foundation for a secure and private identity mechanism for the application.

HW1: Using the provided foundational code and the provided materials, each student is expected to enhance the code by adding input validation for the various input fields, making several other changes to the code, creating a short screencast that shows that the application performs each requirement, and submitting a properly filled out PDF using a provide MS Word template. The detailed requirements for this homework are specified in the Canvas assignment.

TP1: The team must work together to find a current professionally produced application or webpage that requires users to log in to use the application and allows users to update their account information (e.g., name, email address, and password). Using it as a user interface design inspiration, the team must update the provided foundational code so it is clear to a third party that the team's submission was influenced by that application or webpage. The team is then expected to implement input validation, selecting the best from the HW1 work of the members of the team. The team will also implement a set of new user stories specified in the Team Project 1 assignment, integrate them into a well-functioning application, and document the application in a manner similar to the provided code. Two screencasts must be produced: one showing that the functional requirements have been implemented. The second specifies how the requirements flow to the architecture, the detailed design, and the code. The team will then submit a properly filled-out PDF using a provided MS Word template. The detailed requirements for this homework are specified in the Canvas assignment. **Each team member** is required to verify that the submission was successful and works as required **before** the deadline.

The Second Phase

The project's second phase will focus on the data at the heart of the question-and-answer system. In addition to a list of users, each with one or more roles, the system must support lists of questions and answers for each question. The user roles of focus for this phase are admins and students.

HW2: Using the provided materials and either HW1 or TP1, each student must design, document, and then create and test a question class, an answer class, a questions class that supports storing all current questions as well as returning a subset of the questions, and an answers class that supports a set of potential answers to a question as well as a subset of potential answers to a question. With the requirements satisfied, each student must create a short screencast that shows that the application performs each requirement and submit a properly filled-out PDF using a provided MS

Word template. The detailed requirements for this homework are specified in the Canvas assignment.

TP2: Building on the team's TP1 submission, issues and problems pointed out by the grader must be addressed. The team is then expected to implement questions and answers, selecting the best from the HW1 work of each member. The heart of this phase is the addition of functionality that builds on HW1 components to enable students to ask questions, propose answers, read questions and proposed answers, ask for or suggest clarifications, update questions or answers, and announce that a specific answer successfully addressed the issue that prompted the initial question. The team will also implement and provide basic automated testing of a set of new user stories specified in the Team Project 2 assignment, integrate and test them into a well-functioning application, and document the application in a manner similar to the provided code. The team will then submit a properly filled-out PDF using a provided MS Word template. The detailed requirements for this homework are specified in the Canvas assignment. **Each team member** must verify that the submission was successful and works as required **before** the deadline.

The Third Phase

The third phase extends the application by adding the reviewer and instructor user roles and updating the rest of the application as required to support and benefit from these two new roles.

HW3: Using the provided materials and the team's TP2 submission, each student must design, document, create, and then test a set of JUnit tests, explore using Javadoc, find a professional-looking Javadoc example (e.g., Oracle's documentation of the **Java Collections Framework (https://docs.oracle.com/javase/8/docs/technotes/guides/collections/overview.html)**), and update the internal documentation to produce similar Javadoc output for the updated TP2 submission. With the requirements satisfied, each student must create a short screencast that shows that the application performs each requirement and submit a properly filled-out PDF using a provided MS Word template. The detailed requirements for this homework are specified in the Canvas assignment.

TP3: Building on the team's TP2 submission, issues and problems pointed out by the grader must be addressed. The team must implement reviewer and instructor roles, update the testing using JUnit and the internal documentation using Javadoc, selecting the best from the HW3 work of the members of the team. The team will also implement and provide JUnit testing of a set of new user stories specified in the Team Project 3 assignment and integrate and test them into a well-functioning application and documented application. The team will then submit a properly filled-out PDF using a provided MS Word template. The detailed requirements for this homework are specified in the Canvas assignment. **Each team member** is required to verify that the submission was successful and works as required **before** the deadline.

The Fourth Phase

The fourth phase has a focus on reflection and improvement of the documentation, code, testing, and screencasts, as well as the addition of the staff user role as well as any other enhancements that will transform this submission into something each team member will be excited to have and use as a concrete example of what that member and the team can do when applying for internships and future jobs.

HW4: Using the provided materials, the team member's HW3 submission, and/or the team's TP3 submission, each student must expand the staff role epic into a set of User Stories consistent with the epic and the other User Stories. The student must then use those User Stories to design, document, create, and JUnit test their CRUD (create, read, update, and delete) functionalities. With these requirements satisfied, each student must produce a short screencast that shows that the application performs each requirement and submit a properly filled-out PDF using a provided MS Word template. The detailed requirements for this homework are specified in the Canvas assignment.

TP4: Building on the team's TP3 submission, issues and problems pointed out by the grader must be addressed. The team must collaborate to find consensus on the staff roles' User Stories. They must then implement those User Stories and update the testing using JUnit and the internal documentation using Javadoc. The team must produce a plan for who will do what and by when to produce the final pair of presentations and ensure their quality. The team will then submit a properly filled-out PDF using a provided MS Word template. The detailed requirements for this homework are specified in the Canvas assignment. **Each team member** is required to verify that the submission was successful and works as required **before** the deadline.

Frequently Asked Questions

▶ What do we turn in?

▶ Will we receive a paper that tells us everything that the customer wants?

▶ Can our team use C# or some other language, and not Java?

▶ We have some ideas that the customer didn't ask for – but we think that they would be great additions to the project. Can we receive extra credit?

▶ Our grade was low because one of our team members did a bad job on his/her part of the work. Can our grade be higher because we did more work?