

CSE 360

Team Project Phase 2 Report

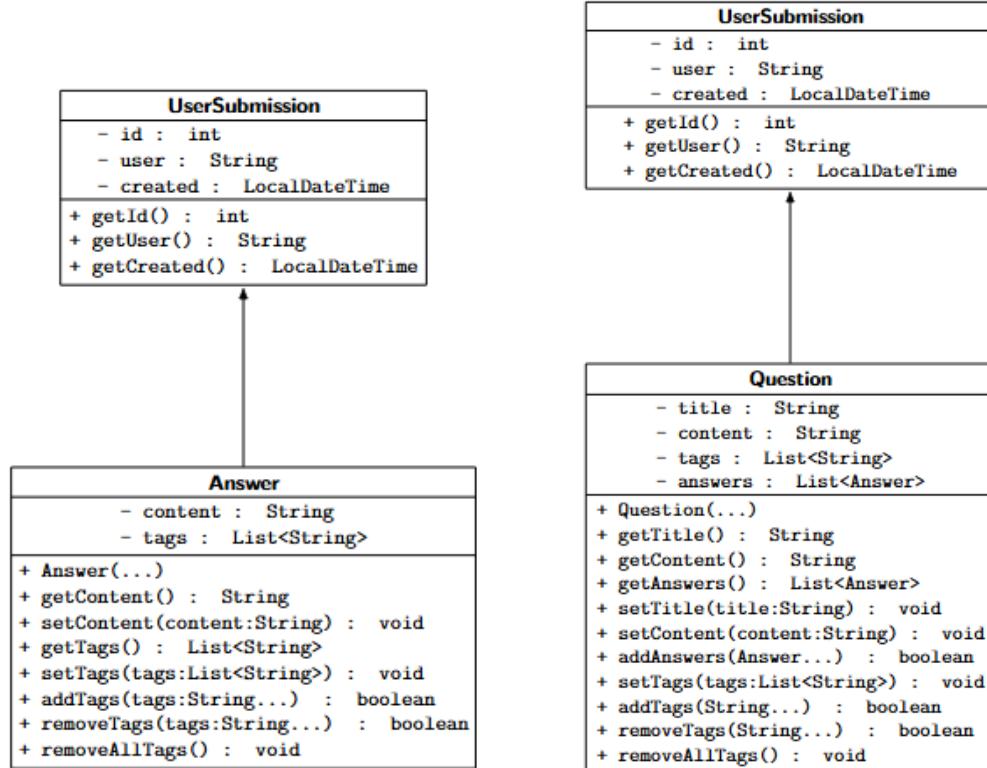
Team 5

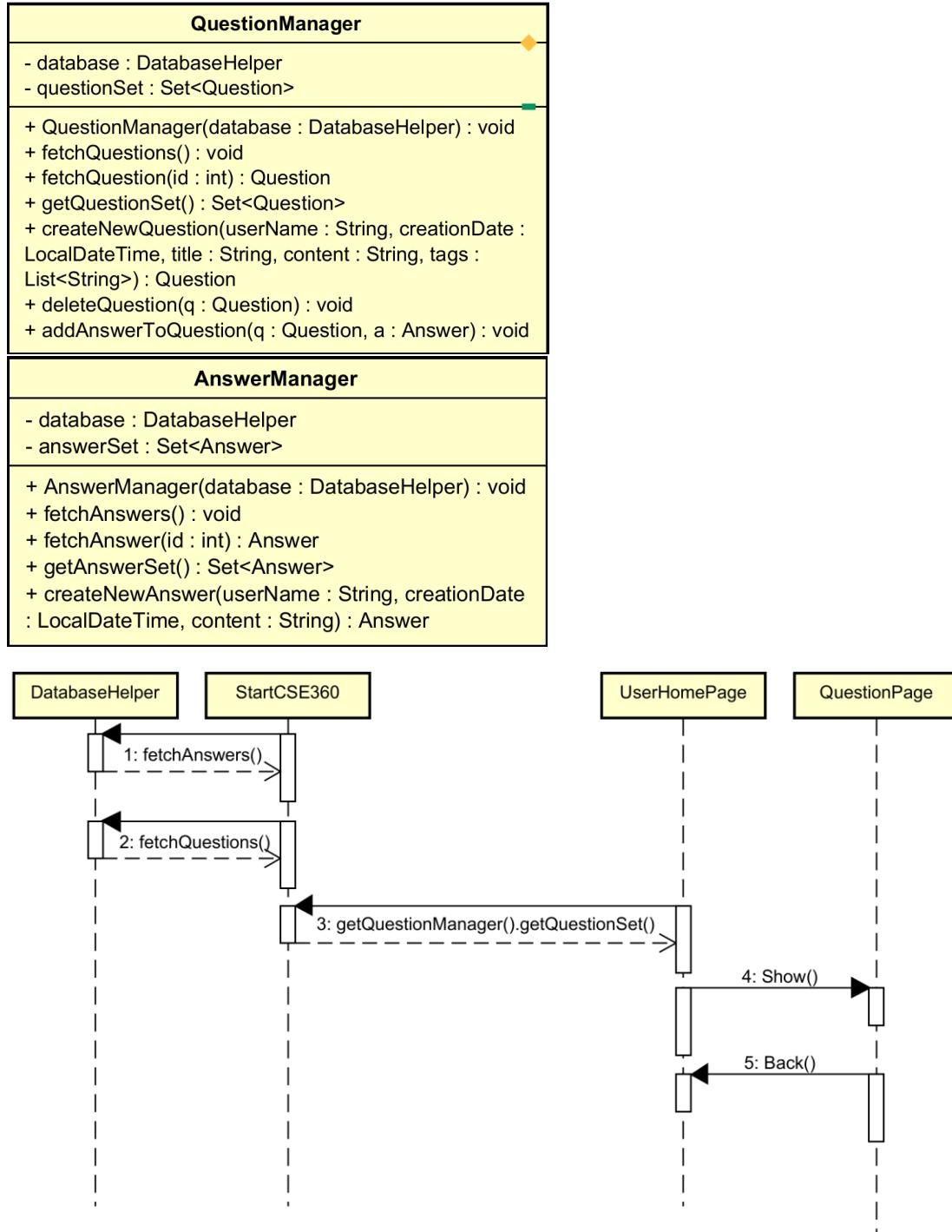
Team Member Names:

1. Jad Khayyati
2. Joshua Wright
3. Kyle Ferolito
4. Steven Grisham
5. Jarod Wagner

1. The Team's agreed upon solution for HW2

1.1. Design Documents





1.2. Description of how the code implements the Design Documents

The code implements the design documents as specified. Each of the classes correspond to their UML diagrams, and the relationships between the main classes are represented in the sequence diagram. StartCSE360 contains the QuestionManager and AnswerManager, which are static members that read from and write to the database. The UserHomePage gets the QuestionSet from the QuestionManager and populates the home page with questions. When

a question is clicked, the UserHomePage creates a QuestionPage that contains the question with its corresponding answers.

1.3. Evidence that the code was tested

Please refer to Section 4 for information on the manual tests. Section 5 also includes a screenshot of the automated test results.

1.4. URL and access to Source Code in GitHub

GitHub Link: <https://github.com/JoshWright22/CSE360Group5/tree/phase2>

2. User Stories

Create (C):

- As a student, I can see a list of questions others have asked that might be related to a question I am about to ask, so I do not waste my time for answers already there, and others don't waste their time reading and answering my question.
- As a student, I can produce a new question based on a previous question to address the feedback I have received and any new insights gained, so I am more likely to get an answer that resolves my issue.

Read (R):

- As a student, I can search the list of questions and answers.
- As a student, I can see my list of unresolved questions and the number of unread potential answers received, so I don't have to scan unrelated messages.
- As a student, I can see a list of all unresolved questions and a list of the current potential answers for each so I can evaluate the potential answers and, if appropriate, propose a new potential answer without duplicating the work of others.

Update (U):

- As a student, I can specify that a specific potential answer resolves my issue so others can benefit from my experience. The system provides a separate access method so others can quickly see the answers that resolve issues without needing to traverse the earlier potential answers that do not resolve the issues.

Delete (D):

- Deletion is implied in question/answer validation.

3. Implementation Plan and Progress Made

3.1. Implementation Plan

The user stories to be implemented, and the team members implementing them, are described in the following table.

Team Member	User Stories
Jad Khayyati	<ul style="list-style-type: none"> As a student, I can search the list of questions and answers. As a student, I can see my list of unresolved questions and the number of unread potential answers received, so I don't have to scan unrelated messages.
Joshua Wright	<ul style="list-style-type: none"> As a student, I can see a list of questions others have asked that might be related to a question I am about to ask, so I do not waste my time for answers already there, and others don't waste their time reading and answering my question. As a student, I can update questions or answers.
Kyle Ferolito	<ul style="list-style-type: none"> As a student, I can produce a new question based on a previous question to address the feedback I have received and any new insights gained, so I am more likely to get an answer that resolves my issue.
Steven Grisham	<ul style="list-style-type: none"> As a student, I can specify that a specific potential answer resolves my issue so others can benefit from my experience. The system provides a separate access method so others can quickly see the answers that resolve issues without needing to traverse the earlier potential answers that do not resolve the issues.
Jarod Wagner	<ul style="list-style-type: none"> As a student, I can see a list of questions others have asked that might be related to a question I am about to ask, so I do not waste my time for answers already there, and others don't waste their time reading and answering my question.

3.2. Progress-Made Descriptions and Screencasts

3.2.1. Standup Meeting 1 (10/4/2025)

- Built a general understanding of TP2
- Shared our HW2 codes with each other to be able to analyze our code and see what we'd like to implement
 - Search function for example
- Determine user stories that focus on CRUD operations

3.2.2. Standup Meeting 2 (10/8/2025)

- Decided whose HW2 to build off of: Kyle's classes and Joshua's database operations
- Created the codebase for Phase 2: a branch called phase2 in the CSE360Group5 repository
- Discussed and added some design documentation

- Clarified screencast-related requirements for TP2
- Assigned user stories to team members

3.2.3. Standup Meeting 3 (10/15/2025)

- Reviewed progress since previous meeting
 - Established Question, Answer, QuestionManager, and AnswerManager objects
- Discussed whether to use a terminal interface or UI
- Discussed implementation of MySQL functions
 - Opted to build off of Joshua's base from HW2
- Discussed the user stories to be implemented further

3.2.4. Standup Meeting 4 (10/15/2025)

- Established the Comment and CommentManager objects
- Wrote most of the remaining backend
- Discussed refactoring the code into more separate packages (obj, eval, etc.)
- Discussed code conventions for consistent readability across codebase

3.2.5. Standup Meeting 5 (10/15/2025)

- Discussed the UI implementation further
- Clarified object relationships amongst development team
- Discussed automated and manual tests
- Discussed updated README file and who will be writing it
- Clarified schedule for the final day

3.2.6. Standup Meeting 6 (10/15/2025)

- Final hoorah: quickly went over everything thus far
- Discussed who will be submitting the final document
- Discussed pull request to merge 'phase2' into 'main'
- Established finishing touches on TP2 prior to submission
- Scheduled the next meeting to look at TP3

4. List of Tests

4.1. Automated Tests

- Username validation
 - Reject blank input
 - Reject input that doesn't begin with an alphabetical character
 - Reject input that contains consecutive special characters
 - Reject input that is too long
 - Reject input that is too short
 - Reject input that contains invalid characters
- Password validation
 - Reject blank input
 - Reject input that lacks an uppercase character
 - Reject input that lacks a lowercase character
 - Reject input that lacks a special character
 - Reject input that lacks a numeric character
 - Reject input that is too long
 - Reject input that is too short
- Name validation (first and last name)
 - Reject blank input
 - Reject input that contains special characters or whitespace
 - Reject input that contains numbers
 - Reject input that is too long
- Email validation
 - Reject blank input
 - Reject input that is missing a domain name
 - Reject input that is missing the '@' symbol
 - Reject input that is too long

4.2. Manual Tests

- Ensure that questions can be submitted
- Ensure that answers can be submitted
- Ensure that questions and answers are correctly retrieved from the database
- Ensure that the search feature is functioning as expected

5. GitHub Repository

5.1. The URL to access the GitHub Repository with a ReadMe file

- GitHub URL: <https://github.com/JoshWright22/CSE360Group5/tree/phase2>

5.2. Access to the Team's Solution for HW2

- Joshua's HW2 solution: <https://github.com/JoshWright22/CSE360HW2>

5.3. Access to Screencasts

5.3.1. Screencast testing the Team's HW2

- Code review: <https://youtu.be/BsA6Mftx-48>
 - A lower-quality version of the same video can be found in the repository.
- The program demo is in the GitHub repository.

5.3.2. Screencasts from each Standup Meeting

- Screencasts will be available in the GitHub repository for TP2 (listed above).

5.3.3. Plan for the manual tests

- Please see Section 4 of this document for plans regarding the manual tests. Below, however, are screenshots of the results of automated tests.

5.3.4. Screencasts showing the manual tests

- Screencasts will be available in the GitHub repository for TP2 (listed above).

5.3.5. Plan for showing that each requirement has been satisfied

- Log in with an already-established Admin account and create an invite for a new user
- Create a new user named "CrashDummy" whose password is "Live2Learn!"
- Restart the program to sign in as "CrashDummy" and create a new question
- Restart the program to sign in as a pre-existing user named "SmugDummy"
- As "SmugDummy", answer the question created by "CrashDummy"
- Restart the program to sign in as "CrashDummy"
- View questions page and the answer left by "SmugDummy"

5.3.6. Screencasts showing the satisfaction of the requirements

- Code review: <https://youtu.be/BsA6Mftx-48>
- The program demo is in the GitHub repository.
 - A lower-quality version of the same video can be found in the repository.

5.4. Console Evidence from the Automated Tests

Finished after 0.449 seconds		
Runs: 4/4	Errors: 0	Failures: 0
✓ DatabaseHelperTest [Runner: JUnit 5] (0.000 s)		
└ testDuplicateUserRegistration() (0.000 s)		
└ testLoginSuccess() (0.000 s)		
└ testLoginFail() (0.000 s)		
└ testRegisterUser() (0.000 s)		
Runs: 6/6	Errors: 0	Failures: 0
✓ EmailEvaluatorTest [Runner: JUnit 5] (0.030 s)		
└ tryMissingDomain() (0.020 s)		
└ tryInvalidCharacter() (0.001 s)		
└ tryNotAnEmail() (0.001 s)		
└ tryEmptyEmail() (0.001 s)		
└ tryValidEmail() (0.001 s)		
└ tryEmailWithSubdomain() (0.001 s)		
Runs: 5/5	Errors: 0	Failures: 0
✓ NameEvaluatorTest [Runner: JUnit 5] (0.022 s)		
└ tryNameWithNumber() (0.016 s)		
└ tryTooLongName() (0.001 s)		
└ tryNameWithSpecialChar() (0.001 s)		
└ tryEmptyName() (0.001 s)		
└ tryValidName() (0.001 s)		
Runs: 6/6	Errors: 0	Failures: 0
✓ PasswordEvaluatorTest [Runner: JUnit 5] (0.041 s)		
└ testUpperCaseRequirement() (0.026 s)		
└ testLengthRequirement() (0.004 s)		
└ testLowerCaseRequirement() (0.003 s)		
└ testNumericDigitRequirement() (0.002 s)		
└ testEmptyPassword() (0.001 s)		
└ testSpecialCharacterRequirement() (0.002 s)		
Runs: 6/6	Errors: 0	Failures: 0
✓ UserNameRecognizerTest [Runner: JUnit 5] (0.038 s)		
└ testEmptyUserName() (0.025 s)		
└ testFirstCharacterIsAlphabetical() (0.002 s)		
└ testMaxLength() (0.002 s)		
└ testSpecialCharactersAreNotConsecutive() (0.003 s)		
└ testMinLength() (0.001 s)		
└ testInvalidCharacters() (0.003 s)		

6. Appendix A: Credit Sheet

This appendix lists the members of the team and includes a description of the contribution that team members have made to this submission. Each team member must provide text for their contribution at a team meeting just prior to the submission of this deliverable and the entire team must agree. If a team member fails to provide this information and/or does not participate in the agreement process to fill out this table, that member will receive no credit for the submission of this deliverable.

Team Member Name	Contributions
Jad Khayyati	Designed search function Write plans for manual tests Participated in team design and planning meetings Created .gitignore file to declutter repository Added finishing touches to submission document
Joshua Wright	Integrated database operations from HW2 Developed update functionality for questions/answers Helped establish foundational backend structure Contributed to question similarity feature Reviewed and tested team code for consistency Participated in team design and planning meetings
Kyle Ferolito	Provided design documents Created the user interface for questions and answers Added database functionality for questions and answers Handled Section 1 of this document Provided initial UML diagrams Participated in team design and planning meetings
Steven Grisham	Created collaborative documents on Google Drive Created 'phase2' branch and helped add TP1 requirements Wrote backend code for numerous objects Wrote scripts for Screencasts 3-6 Added documentation and consistent formatting to code Implemented automated tests for TP1 fields
Jarod Wagner	Created final UML diagrams Created final pull request Participated in team design and planning meetings Added comments to code Wrote updated README file Wrote plans for automated tests

7. Appendix B: Team Norms

No changes have been made to the Team Norms.