SWARE PROJECT BACKLOG VERSION 5

CONTENTS

PERSONAS

LATEST CHANGES: MERGED PERSONAS
[PROFESSOR DINA NO LONGER EXISTS]

Professor Vladimir Shostakovich

- ➔ 53 year old male that was raised in Russia
- ➔ Teaches statistics courses, does research on stochastic processes, and is the stats Program Supervisor at the University of Waterloo
- ➔ Lives close to school and has computers with internet both at home and at work
- ➔ Can't speak english fluently, but he can carry on a basic conversation
- ➔ Manages TAs for his research - typically students who are stats majors and specialists
- ➔ Enjoys his work, happy to finish tasks at home if they were not completed at work
- ➔ Students and faculty can easily reach him by popping by his office or sending him an email but he is not familiar with responding from his smartphone
- ➔ Limited experience with modern technology, preferring to show students his steps when solving calculations in class, but is comfortable with the software he has to use regularly
- ➔ Typically prints out any online assignments so that he can hand back papers with corrections and feedback to students
- ➔ Cares a lot about the success of his students and uses statistical data to improve his classes
- ➔ Well known moderator in many discussion boards to discuss improvements to the student experience
- ➔ Enjoys giving unique assessments as opposed to traditional exams, like take home exams, group work projects as a final evaluation and partial final evaluations online (along with physical version)
- ➔ Very organized in his presentation of material and maintains the course content delivered throughout the semester
- ➔ Interested in exploring new technologies to enhance his students' learning and success in their education but he doesn't have time to learn about the technical details of programs and computer interfaces

Professor Karen Chong

- ➔ 30 years old, female
- ➔ 3rd generation immigrant fluent in 3 languages
- ➔ Currently in her 2nd year of teaching Intro to Statistics as an assistant professor
- ➔ Tech savvy and used to dealing with sophisticated applications with tons of features - she loves to find out about everything an application offers
- ➔ Spends 4 hours in commute daily so she prefers to review and revise her online assignments in those times
- ➔ Prefers evaluation applications that return instant feedback to students
- ➔ Praised by her peers as a driven individual with a lot of ambition to enrich the student experience
- ➔ Her PhD in Advanced Statistical Methods of Online Networking was highly regarded and won lots of awards
- ➔ Values statistics of online assignment completions to motivate students in class, and possibly give bonus marks
- ➔ Loves the automation of assignment marking as it gives her lots of time for research!

Professor Adam Smith

- ➔ 42 years old, male
- ➔ Teaches graduate level courses such as statistical consulting, risk theory, and advanced theory of statistics
- ➔ Worked in the industry for 15 years as a Data scientist, currently working as a part time it consultant for a small startup
- ➔ Has worked with complicated applications and is familiar with scripting languages such as Javascript, Perl and PHP
- ➔ Wants to ensure any online evaluation reflects the course material, is fair and different than any past evaluations; hates when past evaluation questions and answers are posted online
- ➔ Hires student interns during the summer to write code to ensure randomized values in online evaluations using problem generating languages
- ➔ Has very high expectations of his students, increases the difficulty of course content slightly if students find work to be too easy
- ➔ Suffers from carpal tunnel syndrome which causes a lot frustrations using some web applications
- ➔ Follows his slides exactly, rarely adding any extra content when presenting, and usually does not answer questions
- ➔ Primarily uses the basic technology provided by the university such as BlackBoard and UofT webmail
- ➔ Rarely changes the software used for class because she would rather spend her time on her business or her research

Student: Jane Doe

- ➔ 18 years old, female
- ➔ First year of university, fresh out of high school
- ➔ Studies mostly in the library when she's free and tries to organize study dates with her friends.
- ➔ Goes to the math aid center only when an assignment is due and she's having difficulties.
- ➔ She doesn't mind having the option of sitting down with a TA or fellow student when she's struggling but prefers problems with readily available solutions to check her answers (especially when the questions are not for credit).
- ➔ Likes a weekly assignment system so she keeps on track with the course, but only does work that counts toward her grade, unless she's practicing for a midterm or final.
- ➔ She's pretty average in her math/stats courses. She much prefers her courses with readings and questions that don't require numbers in their answers.
- ➔ She prefers solving answers on paper and inputting only the final answer she gets.
- ➔ She has great writing skills.
- ➔ When an assignment is marked, she likes to see comments on where she went wrong.
- ➔ Only uses technology for social media.
  Uses Pages on her Mac to write her essays.
  No other technological proficiencies.
- ➔ Has nothing against technology as she sees it, but has no clue about the underlying technologies that exist such as latex.
- ➔ She hasn't really used any online math tools so she doesn't have any feelings towards them in particular, but she does like to give feedback when she tries new things.

Student: Slacky MacSlackerson

- ➔ Male, 17 years old
- ➔ First year student in the Fine Arts program
- ➔ Works in spaces that inspire him, and as an arts student prefers places and things that have personality/character.
- ➔ Does his readings outdoors if weather permits, and avoids the library at all costs.
- ➔ Personally prefers using websites that draw the eye, which is probably why he avoids most online math related things as he finds them drab. He likes the use of interesting fonts, formatting and a clear design that allows him to differentiate the key parts of the page/app easily.
- ➔ Slacky is really good at sculpting things and drawing. He's got a very arts oriented Instagram page and posts often.
- ➔ He has a smartphone he uses a lot to take photos, post on social media and keep in touch with family and friends.
- ➔ He takes notes on paper as it allows him to doodle when bored and has never tried typing math oriented things because the thought of finding all those special characters makes his head spin.
- ➔ He tends to forget about the small details he learns in class.
- ➔ His lectures interfere with his social habits, so he tends to skip them and read textbooks when he has free time instead.
- ➔ He almost never does practice questions so he prefers learning from knowing how to answer questions from given solutions.
- ➔ He always checks the syllabus for all the coursework that will give his marks and makes sure he completes them. Also prefers notifications when something new is up for completion.
- ➔ He has nothing against technology in general and uses it in all his social media, but doesn't understand the point of using math tools online when it's so much easier to write it out.

Student: Hype McGee

- ➔ Female, 20 years of age
- ➔ Second year pursuing a specialist degree in Math. Just came out of a semester abroad in France, where they had a few advanced math courses she wanted to try.
- ➔ She enjoys exploring new areas of math in her free time, and has joined the math club. She has a relatively light course load this semester, as the math courses she's taking are teaching things she's relatively comfortable with.
- ➔ Has a long commute, where she solves problems on her laptop as she finds this easier than carrying around sheets of paper.
- ➔ Uses mostly online resources, so she rarely ever buys paper textbooks.
- ➔ Finds doing math questions tedious if she already knows it, so prefers to have something that gauges her ability and provides questions accordingly.
- ➔ She's really good at solving number problems and teaching herself things using online tools.
- ➔ Likes helping others and is currently a TA for a first year math course as well.
- ➔ She's worked with online math tools a lot, having practiced a lot of what she knows off them.
- ➔ She prefers auto-marking, as she gets instant feedback.
- ➔ She's somewhat proficient at Latex, as she's had to use it when doing online problems.
- ➔ She is very good at keeping track of her notes, all her notes are sorted by year and category. She likes it this way so she can refer back to anything she had on a subject later on.
- ➔ No problem with technology, and only uses paper when brainstorming solutions to a particular difficult problem.
- ➔ Never asks for answers or any answers as she strongly believes if she can't solve something on her own she won't solve every challenge that comes her way later.

User Stories

LATEST CHANGES: READJUSTING ORDER AND DELETING IRRELEVANT USER STORIES

U01: As Adam Smith (a professor), I would like to make simple questions.

U02: As Jane Doe (a student), I would like to view and answer questions using the app.

U03: As Adam Smith (a professor), I would like to create an online assessment, so that I can start organizing my questions.

U04: As Adam Smith (a professor) I would like to create new courses so I can start organizing my assessments

U05: As Adam Smith (a professor), I want to login as a professor, so I can view courses relevant to me

U06: As Adam Smith (a professor), I want to be able to view the assessments I created.

U07: As Adam (a professor), I want to add and remove students to courses I am teaching, so that I can control who is in my classes.

U08: As Slacky MacSlackerson (a student), I want to have an account with a login and a password, so that I can view questions for courses as professors make them visible.

U09: As Karen Chong (a professor), I would like to create different types of questions.

U10: As Jane Doe (a student), I would like to view my grades for assessments and questions I have completed.

U11: As Adam Smith (a professor), I would like to control who sees my questions.

U12: As Jane (a student), I would like to only have access to the assessments for the courses I am enrolled in, so that I know what questions and assessments I have to do.

U13: As Karen Chong (a professor), I would like to be able to add hints for students.

U14: As Jane (a student), I would like to have a page with sample solutions to solutions of assessments that are available.

U15: As Adam Smith (a professor), I want to set due dates on online assessments to stop students from answering thereafter.

U16: As Jane (a student), I would like every assessment to be clear about whether it is for marks or not.

U17: As Hype (a student), I would like to save or submit each question in an assessment individually, so that I can resume work on a question or assessment later.

U18: As Slacky (a student), I would like to view hints on questions if available, so that I do not struggle as much on difficult questions.

U19: As Vladimir(a professor), I want to control how many times a student can retry a question

U20: As Adam Smith (a professor), I want to be able to prevent students from seeing the answers to assessments until the due date has passed.

U21: As Karen Chong (a professor), I would like to have the option for a student to receive the result of the question if a student has used up all their attempts and did not get the correct answer.

U22: As Karen Chong (a professor), I would like students to be able to suggest hints after they answer questions.

U23: As Hype (a student), I would like to be able to suggest a hint for a question after submission so that I could be of help to others.

U24: As Vladimir Shostakovich (a professor), I would like to notify students whenever an assessment in their course has been released or modified.

U25: As Adam Smith (a professor), I would like the option to add students to a course when given a csv file of their student details.

U26: As Adam Smith (a professor), I want to add questions that have randomized values.

U27: As Slacky (a student), I would like any math symbols frequently used in the course to be easily accessible so that I do not have to search the symbols up.

-- User Stories identified as extra --

U28: As Karen Chong (a professor), I want to be able to download the online assessments as a pdf.

U29: As Vladimir (a professor), I want to be able to get a copy of questions in a reasonable format for an printed online assessment.

U30: As Karen Chong (a professor), I want to have access to previous problem sets that I can reuse from past years.

U31: As Vladimir Shostakovich (a professor), I want to allow someone else to have professor-like access to a specific class, so that I can get someone else to do some work for me but still control who has access to my classes.

U32: As Adam Smith (a professor), I want to be able to set time limits on each online assessment, after which students are unable to answer questions.

U33: As Vladimir Shostakovich (a professor), I want a timer on each assessment to let students see how long it's taking them to do the assessment.

U34: As Vladimir (a professor), I want to have a frequently asked Q&A for each online assessment.

U35: As Jane (a student), I would like to post feedback and questions on the Q&A board.

U36: As Karen Chong (a professor), I want to post messages for a class, so that students can get updates or encouraging messages from me.

U37: As Vladimir (a professor), I want to get statistics on problems/ problem sets such as

- Percent of students that completed questions on an online assessment
- Percent of students that got a question correct/incorrect
- Average amount of time for each online assessment to be completed (each student, every student in course)
- Average amount of time before each question was submitted. (each student, every student in course)

U38: As Adam Smith (a professor), I would like the option to have students added to a course based on their status on ROSI.

U39: As Slacky (a student), I would like to have a lifeline option to share my question with a friend and solve it together.

U40: As Hype (a student), I would like to answer the questions for an assessment while offline.

SPRINT 01: TASK BREAKDOWN FOR USER STORY 01

U01: As Adam Smith (a professor), I would like to make simple questions.

T01 - Design an object class to hold and display questions and the correct answer

(2 story points, Dependencies: None)

Technological choices: Team decided on a Java based application.

1. QuestionAbstract.java is an abstract class for all future question classes, where question will always be a string, answer will depend on the type of questions, (e.g text based or multiple choice)
2. Name of the question will be a string input
3. Points per question is given an integer input
4. Extend the abstract class to build the TextQuestion class, a basic type of question class where both question, answer, name are string inputs, and points is an integer input.

T02 - Implement code that takes string input and creates a question object, which has a question and an answer

(2 story points, Dependencies: T01)

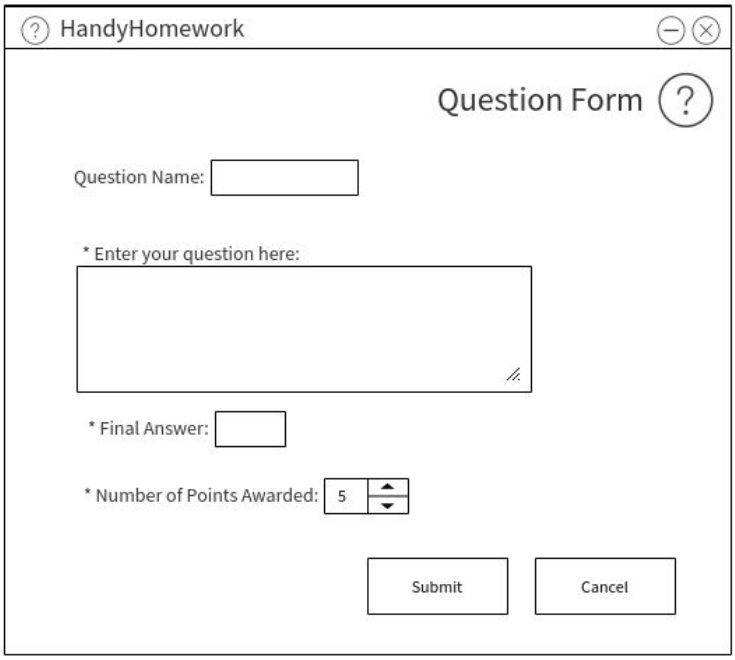
The main file runs an infinite loop till user types enter. Continuously reading inputs from users to gather questions and answers to put into the database. Used for testing.

T03 - Design wireframe for user interface to make questions

(4 story points, Dependencies: None)

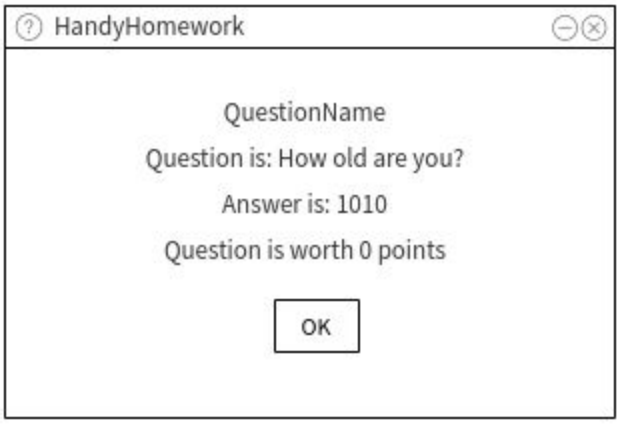
UI Design Sketches using mockflow.com

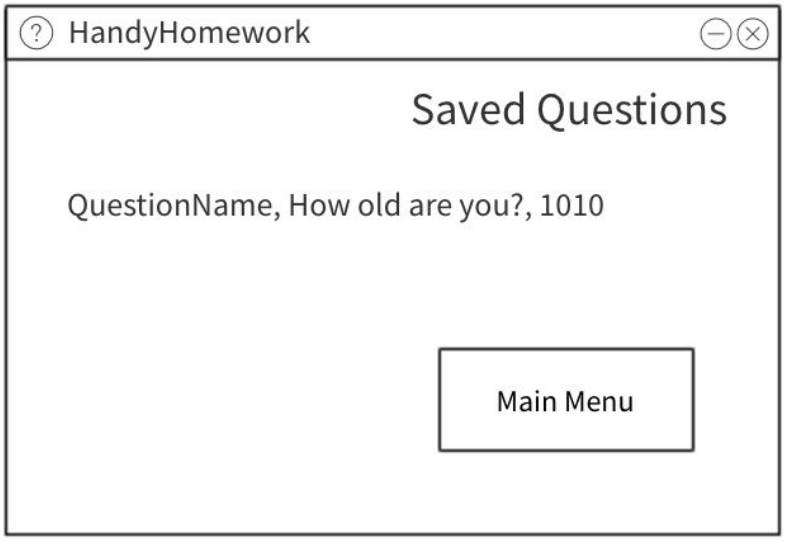
Home Page of the application →



← Question Form

Successful question save message →





← Saved Questions page

T04 - Design a schema to hold saved questions

(2 story points, Dependencies: None)

1. Setup Postgres database and host it (on amazon ads server)
2. Create and test DDL with Postgres

T05 - Implement the database

(4 story points, Dependencies: T04)

1. Implement schema with tables
2. Allow for inputting question with a generated unique id (using postgres sequences)

T06 - Implement the UI

(10 story points, Dependencies: T03)

1. Used Java Swing
2. Used wireframes as a guide and connected the pages using ActionListeners on buttons

T07 - Implement the API to push questions to the database

(7 story points, Dependencies: T06)

1. Install required jar files and create a database connection class
2. Implement method to connect to database
3. On creation of a question, store values
4. When viewing saved questions, query the question table and and print stored questions

SPRINT 02: TASK BREAKDOWN FOR USER STORIES 02 AND 03

U02: As Jane Doe (a student), I would like to view and answer questions on the application.

T08 - Query saved questions from the db, create question objects on retrieval

(1 story point, Dependencies: None)

Schema stored in a Google Drive folder.

T09 - Implement a way to check answer for questions

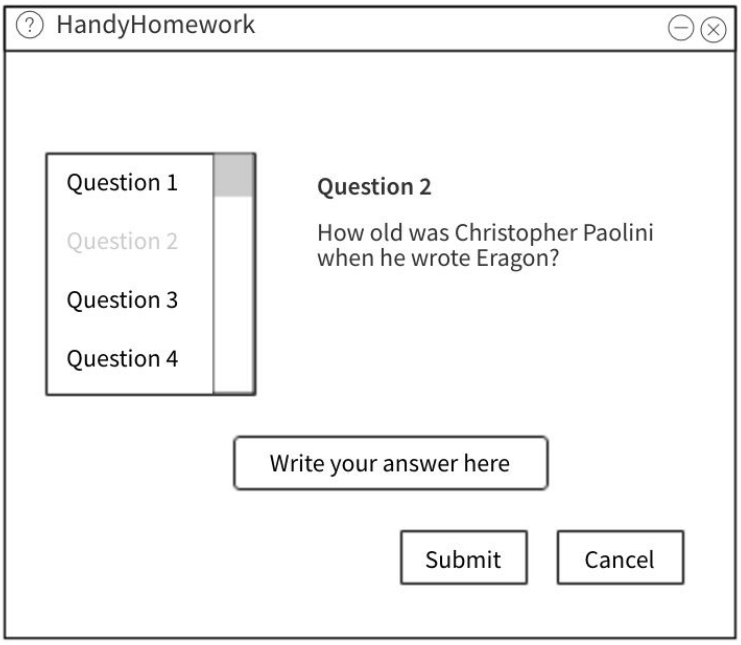
(1 story point, Dependencies: None)

Used database implementation to retrieve the correct answer and compare.

T10 - Design gui to display questions based on their name, content, and points

(3 story points, Dependencies: None)

Wireframe:



T11 - Extend gui to get answers to questions

(2 story points, Dependencies: T10)

T12 - Connect UI to check answers with database

(2 story points, Dependencies: T8, T11)

U03: As Adam Smith (a professor), I would like to create an online assessments, so that I can start organizing.

T13 - Create online assessment classes that store questions in some order

1. Assessment class that will be used to add question objects and delete question objects.
2. Questions are stored in Hashmaps with name of the question as key and object as return value.

(2 story points, Dependencies: None)

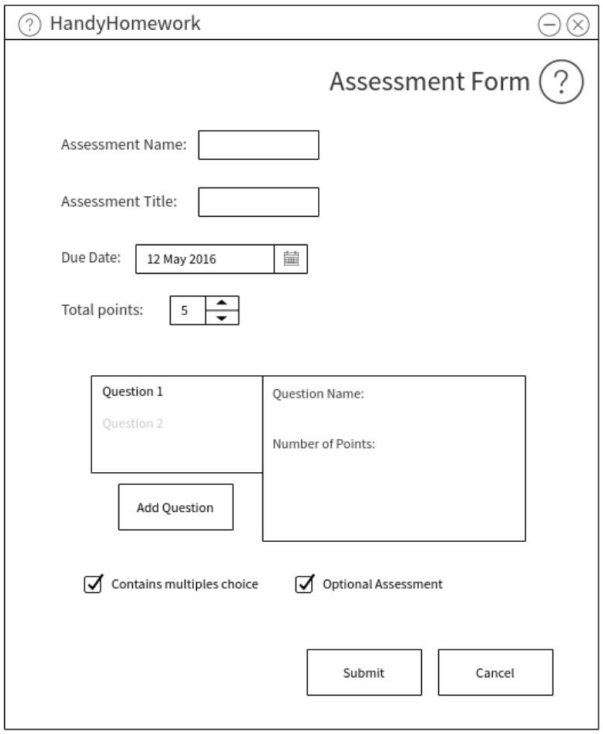
T14 - Extend the schema to hold online assessments

(8 story points, Dependencies: None)

T15 - Adapt the GUI to create online assessments

(7 story points, Dependencies: None)

Wireframe:



T16 - Adapt the GUI to display online assessments

(6 story points, Dependencies: None)

T17 - Pushing and retrieving assessments to/from database

(4 story points, Dependencies: T14, T16)

SPRINT 03: TASK BREAKDOWN FOR USER STORIES 03

Continued User Story 3 from Sprint 02, with the same tasks aside from T13 which was completed in Sprint 02.

U03: As Adam Smith (a professor), I would like to create an online assessments, so that I can start organizing.

T14 - Extend the schema to hold online assessments

(8 story points, Dependencies: None)

T15 - Adapt the GUI to create online assessments

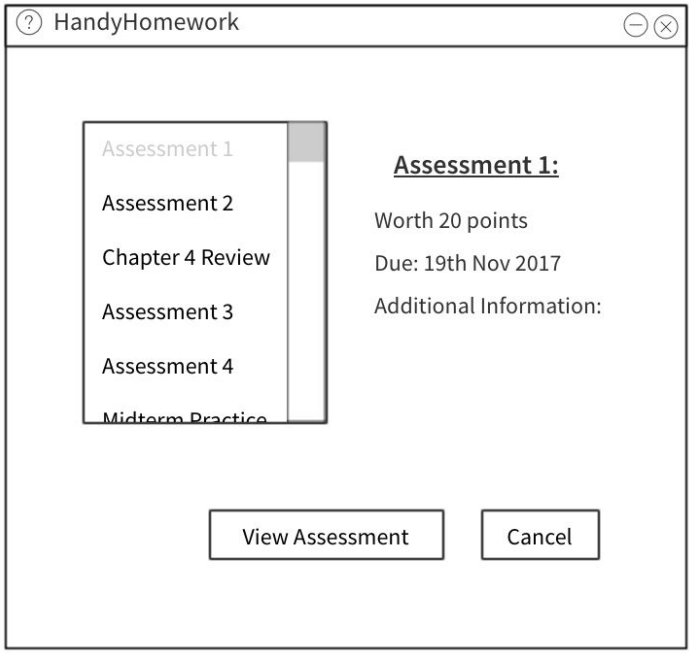
(7 story points, Dependencies: None)

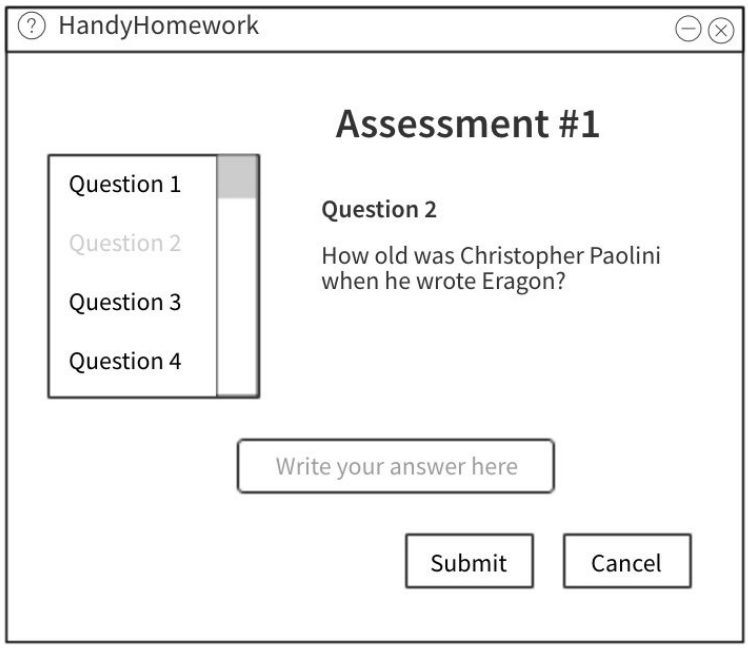
Wireframes as above.

T16 - Adapt the GUI to display online assessments

(6 story points, Dependencies: None)

Wireframe:





T17 - Pushing and retrieving assessments to/from database

(4 story points, Dependencies: T14, T16)

SPRINT 04: TASK BREAKDOWN

U03: Tasks 15-7 from previous sprint carried forward.

U04: As Adam Smith (a professor) I would like to create new courses so I can start organizing my assessments

T18. Adjust DB to include courses

a. Design and Add course table and key relations between assessment table

(2 story points, Dependencies: None)

T19. Extend Db connection API

a. Add functions to retrieve assessments from course table

b. Refactor assessments table to retrieve answers for certain courses

(4 story points, Dependencies: T18)

T20. Create a class for courses for front end purposes

(2 story points, Dependencies: None)

T21. Create the course creation and view courses forms

a. Wireframes for viewing follows that of saved assessments

b. Course creation must include course code, name and semester.

(4 story points, Dependencies: T20)

T22. Connect DB api and course form front end

(4 story points, Dependencies: T19, T20, T21)

U05: As Adam Smith (a professor), I want to login as a professor, so I can create and add to courses I am a part of.

T23. Create backend classes to support the front end

(2 story points, Dependencies: None)

T24. Adapt GUI for login page as professor

a. Differentiate student from professor

b. Check if input fields username and password are empty

c. Direct user to different pages depending on student or professor

(6 story points, Dependencies: None)

SPRINT 05: TASK BREAKDOWN

U05: As Adam Smith (a professor), I want to login as a professor, so I can create and add to courses I am a part of.

T24. Adapt GUI for login page as professor (continued)

a. Differentiate student from professor

b. Direct user to different pages depending on student or professor

c. Create register form

(4 story points, Dependencies: None)

T25. Refactor GUI for all pages

a. Change every GUI page so the placement of the GUI makes sense

(4 story points, Dependencies: None)

T26. Configure DB to reflect users

(6 story points, Dependencies: None)

U06: As Adam (a professor), I want to add and remove students to courses I am teaching, so that I can control who is in my classes.

T27. Modify database schema to include students

(Shadman, 6 story points, Dependencies: None)

T28. Extend gui to add students to a course

(4 story points, Dependencies: None)

a. Using a csv file

b. Add student course to add them individually

T29. Add form to view all students in a course as a prof

(2 story points, Dependencies: None)

U07: As Slacky MacSlackerson (a student), I want to have an account with a login and a password, so that I can view questions for courses as professors make them visible.

T30. Update login screen to support students

(2 story points, Dependencies: None)

T31. Revise UI for students to answer questions where professors see the answer.

(2 story points, Dependencies: None)

SPRINT 06: TASK BREAKDOWN

U06 (remainder): As Adam (a professor), I want to add and remove students to courses I am teaching, so that I can control who is in my classes.

T28. Extend gui to add students to a course

(4 story points, Dependencies: None)

c. Using a csv file

d. Add student course to add them individually

T29. Add form to view all students in a course as a prof

(2 story points, Dependencies: None)

U08: As Karen Chong (a professor), I would like to create different types of questions.

T30. Update DB to hold MC questions (1 story points)

T31. Update DB to hold Latex Questions (1 story points)

T32. Create tests for retrieving and deleting questions (1 sp)

T33. Add dropdown menu option to create MCQ in view assessments (1 sp)

T34. Create forms to create a MCQ (4 story point)

T35. Create forms to create/import Latex questions (4 story points)

T36. Update GUI to display questions on a separate page, with an assessment outline on the side to navigate to different questions within the assessment (2 story points) James

T37. Update GUI to present MC questions to both students and professors (1 story point) James

T38. Update GUI to present Latex questions to both students and professors (2 story points)

T39. Create class for opening new forms (2 sp)

T40. Test database classes (4 sp)