Final Deliverable

Team: L02_04

Alley Dismay, Ka-Kit Jason Cheung, Pak Sun (Harrison) Fok, Zenan (Will) Ji, Zongyang (Eddie) Li

November 26, 2017

Table of Contents

Product Backlog

Personas

Student Personas:

Sara Bai

- 20 year-old student, female, Canadian-born-Chinese, in 2nd year at UTSC.
- Speaks English fluently.
- Specialist in Sociology with a 2.5 CGPA.
- Loves to read long papers and write essays, but hates math.
- Low technological competence, only uses her computer to watch videos on YouTube, basic usage of MS Word to write essays and uses default web browser on computer to check UTSC Portal.
- Uses a very basic Nokia phone to make phone calls and text.
- She always prints her homework off the computer because she hates reading off a computer screen.
- Takes STAB22 as an elective, since she was told it was an easy A.
- Doesn't want to spend too much time learning software for an elective course, just to complete assignments.
- Wants to just use her uToronto credentials as opposed to making a new account as it would be too much work.
- Likes to divide her assignments and homework into several parts to complete them in smaller chunks, over time.
- Would rather do hardcopy assignments because she believes using software will complicate things.
- Likes to be prepared and caught up for every lecture and before doing every assignment.

Lucas Liu

- 20 year-old student, male, from China, in 3rd year at UTSC.
- Speaks English and Chinese fluently.
- Double major in Statistics and Physical Sciences
- Introvert and autistic, never wants to communicate with people and enjoys staying alone in a room.
- Good computer skills: uses applications like the Microsoft Suite, Dropbox, Skype, computer games
- Typically too lazy to do homework because it's too boring and not engaging enough, would rather play video games like Dota 2
- Very interested in exploring new software and reading about new technologies.
- Works 25 hours a week at a retail store, trying hard to balance work and school.
- Likes to do a lot of practice and do as many problem sets as he can to prepare for tests and understand material
- Hates when professor give limited examples to work on
- Wants to be able to redo assignments to get the best possible mark.
- Appreciates how software looks and loves to use aesthetically pleasing software.
- Doesn't always have internet connection though, so prefers to have a desktop app rather than web app

Alice Smith

- 22 year-old student, female, single Irish-Canadian.
- Lives on residence, with constant access to the school's wifi.
- Speaks English and Irish Gaelic.
- Specialist in Statistics, learns new things quickly and has strong multitasking abilities.
- Passionate about academics, outgoing and spends her free time getting her work done as soon as possible, tries to learn the fastest way of doing everything.
- Loves technology and has access to the cutting-edge of technology.
- Proficient in MS Suite, Photoshop, Final Cut Pro, and numerous other softwares.
- Only likes to do assignments once and disregards them afterwards, not looking at them again.
- Gets frustrated easily when it takes multiple steps to do something simple.
- Is a visual learner, so likes to look at graphs and charts which will help her learn the material better.
- Likes to work on mobile devices like her phone or tablet because its more convenient for her to carry around, rather than a desktop or laptop.
- Likes to keep track of her performance in assignments and other course work she has done.

Professor Personas:

Professor Obi Wan-Too-Thrie

- 65 year-old professor, female
- Speaks English and Hindi
- Works as a professor in the Statistics Department where she mainly teaches introductory statistics courses.
- Prefers to use simple software that is straight forward and uncomplicated.
- She can only perform tasks when someone teaches her and follows the exact same steps
- Little exposure to technology, no cell phone and uses very basic functions in MS Word and Excel to enter grades of her students.
- Teaching style: lets students practice on their own to figure out their own method of approaching problems.
- Doesn't use software that is not engaging or interesting aesthetically
- Currently has students print out exercises to do and hand in at tutorial, but those take really long to be marked (2-3 weeks)
- Teaches classes as large as 1000 students and is usually unable to go over exercises in lecture
- Wants to be able to generate many different questions and types for students through the software, so that students can work them out at home.
- Wants software to be intuitive and familiar to the current software she uses, like MS Word and Excel.
- Believes that gamification(virtual rewards) of exercises/ assignments will increase student's interest in learning and course content.

Professor Bob

- 35 year old stats professor, male with no wife or kids
- works at a the university of Toronto with a degree in both Computer Science and Statistics as a Professor and researcher
- Prof Bob is a full time professor that teaches multiple first and second year courses on Statistics
- Likes to be very organized, likes predictability and things to be done quickly and efficiently
- Speaks English only
- Used to use software for assigning and having students answer questions, but it has become outdated, confusing to use and didn't have sufficient functionality (adding graphs, attachments to questions), which made him frustrated and stopped using it
- good computer skills in the Microsoft Suite, multiple IDEs and terminal
- Has at most 100 students in each class and only a couple of TAs to mark course work since Prof Bob is constantly busy doing research and writing papers
- wants to be able to quickly and easily create and upload questions and distribute them amongst all students
- wants students to be able to provide immediate feedback once students finish an assignment
- Likes to see how the class is doing and what material they are struggling with
- his teaching style is that he prefers students learn by example, rather than memorizing theorems, hoping using the system will allow for abundant question generation to provide many examples to learn and prepare with for exams
- want to give students as much practice / retries on different assignment questions until the deadline of the assignment

User Stories

- As a Professor (Obi-Wan-Too-Thrie), I want to be able to create different types of questions like multiple choice, fill-in-the-blanks, matching, etc.
- As a Professor (Obi-Wan-Too-Thrie), I want each student's version of each question to be different, by having randomized numerical values.
- As a Professor (Bob), I want to assign graphics to questions.
- As a Professor (Bob), I want to be able to add/edit/delete questions from my csv.
- As a Professor (Bob), I want to be able to select questions from the csv and create an assignment (a collection of questions) for the students to answer.
- As a Professor (Bob), I want to view/edit/remove questions from my csv after I've created them.
- As a Professor (Obi-Wan-Too-Thrie), I want to register a professor profile.
- As a student (Sara), I want to be able to make a student profile.
- As a Professor (Bob), I want toggle the visibility of assignments for students, so that students won't see drafts of the assignments.
- As a Professor (Bob), I want to set a timeframe for when assignments can be completed. (i.e. If it's 1 week of visibility. At the end of 1 week, the assignment's visibility is automatically turned off).
- As a student (Alice), I want to be able to see a list of assignments that I have completed and ones that I have not yet completed.
- As a student (Sara), I want to know the deadline of each posted homework.
- As a student (Sara), I want to view which homework is graded.
- As a student (Lucas), I want to be able to answer the questions using this system and submit the assignment.
- As a student (Sara), I want the online homework system to allow me to save my current answer and quit an unfinished homework.
- As a Professor (Obi-Wan-Too-Thrie), I want a set of questions belonging to an assignment to be automatically marked as soon as a student completes it, so that they get immediate feedback.
- As a professor (Bob), I want to be able to store the student's highest, most recent grade on an assignment in the csv.
- As a student (Sara), I want to know the mark of all the tests and the solution of the last test done.
- As a Professor (Bob), I want to see how many students get each question right or wrong, so that I can gauge the class performance.

- As a Professor (Bob), I want assignment feedback to consist a list of correctly and incorrectly answered questions, the number of questions answered correctly in percentage and also the correct answers for questions that were answered incorrectly.
- As a student (Lucas), I want to be able to redo assignments with new questions until the deadline so that I can get the most practice on a topic.
- As a Professor (Obi-Wan-Too-Thrie), I want one occurrence of each selected question in a single assignment.
- As a Professor (Obi-Wan-Too-Thrie), I want an assignment to re-generate random values each time a student attempts an assignment.
- As a Professor (Bob), I want to be able to create multiple lectures for each class I teach, if I need to.
- As a Professor (Obi-Wan-Too-Thrie), I want to be able to add students to a class, so that they can view and complete any assignment questions in the course.
- As a student (Sara), I want to login to the online homework system using my UTORid and password.
- As a Professor (Obi-Wan-Too-Thrie), I want to create badges and rewards for students depending on their performance on assignments, so that the gasification aspect increases interest.
- As a student (Alice), I want to receive a notification/email indicating when a new assignment has been uploaded, so that I can finish it as soon as possible.
- As a student (Sara), I want to know the related chapter of each posted homework.
- As a student (Sara), I want to be able to save the assignment as a PDF, and other formats and print it so that I can work outside the app.
- As a student (Lucas), I would like to return to where i stopped when reopen the app.
- As a student (Alice), I want to track and see my progress on an assignment or set of questions with a progress bar.
- As a student (Sara), I want the online homework system to have a tutorial to teach me how to use this system when I first use it.

Sprint 5

Instead of completing additional user stories, we decided as a team to organize our finished user stories, polish and refine our final product. We assigned new tasks to different members to ensure the functionality of the final product.

Product Backlog

Task 1: Finish writing unit tests for our associated files (1 story point per member. 5 story points in total)

Description: Every member is designated to finish writing unit tests for functions in files they created for user stories. For sprint 5, this is the top priority because all unit tests should be completed.

Task 2: Separate unit tests into individual files (2 story points)

Description: From our previous meeting with the TA, we realized that we should separate the unit tests so that they were one per Python file. This task was assigned to Eddie.

Task 3: Fix any bugs in the applications (8 story points)

Description: Despite that all current features were connected into one application, bugs still existed within the programs. Therefore, fix any existing bugs to optimize the app. This task was designated to both Harrison as he oversaw the merging of different features into the master branch. Harrison will exhaustively use to app to pick out every possible bug and then implement the necessary solutions.

Task 4: Make sure the UIs don't have extra windows (2 story points)

Description: In the last iteration of the software presented to the TA, there were still instances where random white windows would pop up to the user when using a feature. This is undesired behaviour that will be removed to further polish the software. This task was assigned to Harrison.

Task 5: Write up validation tests for each user story (5 story points)

Description: To ensure that the user stories were accurately and appropriately implemented, validation tests must be performed by potential users and recorded. This task was assigned to Jason.

Sprint Plan

Team of 5 developers: Andy, Eddie, Harrison, Jason, Will:

- Andy is able to complete 10 story points a week
- Eddie - 6 story points a week
- Harrison - 8 story points a week
- Jason - 6 story points a week
- Will - 6 story points a week

Developers will work for 7 days a week and the length of the sprint is 1 week

Team Velocity
The team velocity will be 36 story points a week.

Provisional Schedule: This sprint plan focuses on polishing and refining all the features we have completed into our final product.

Day 1: November 20th
Day 7: November 27th

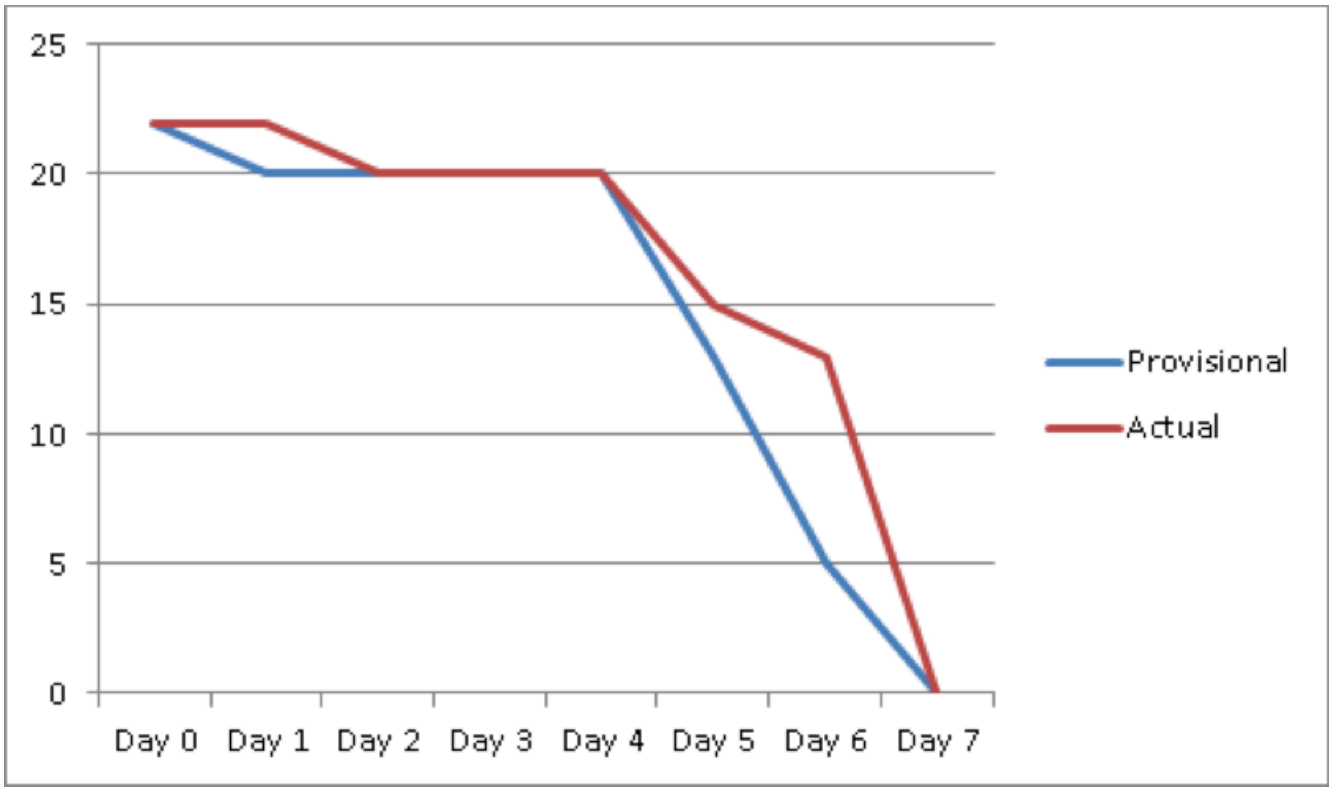
Tasks	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
1	H: 1	E: 1	W: 2		A: 1		
2					E: 2		
3		H: 2	H: 2	H: 2	H: 2		
4	H: 2						
5							J: 5

Sprint Report

By the end of the sprint, even though it took a lot more story points, we managed to burn down all the story points and optimize our product. In some cases such as in Task 3, it required other team members such as Jason, Eddie and William to help Harrison because there were issues with specific code developed by Jason, Eddie and William.

Tasks	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
1	H: 1	E: 1	W: 2		A: 1		
2					E: 2		
3	H: 1	H: 2 E: 2	H: 1 J: 4	H: 4 J: 4	H: 4	W: 2 J: 4 H: 3	J: 4
4	H: 0	H: 2					
5							J: 5

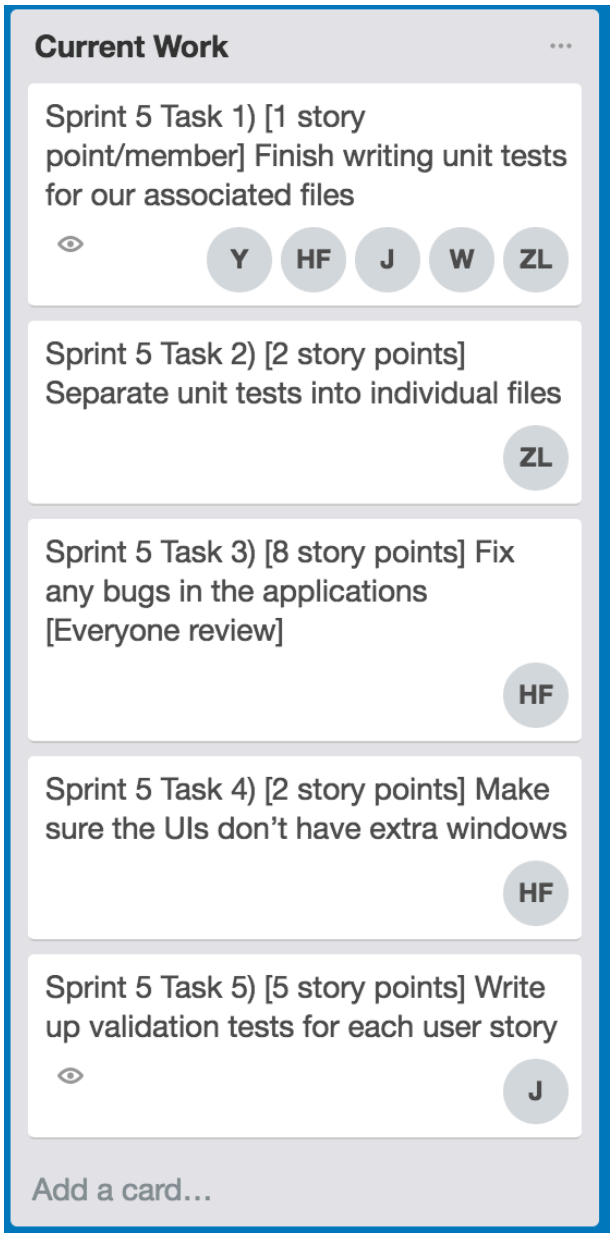
Burndown chart



Snapshots of Taskboard

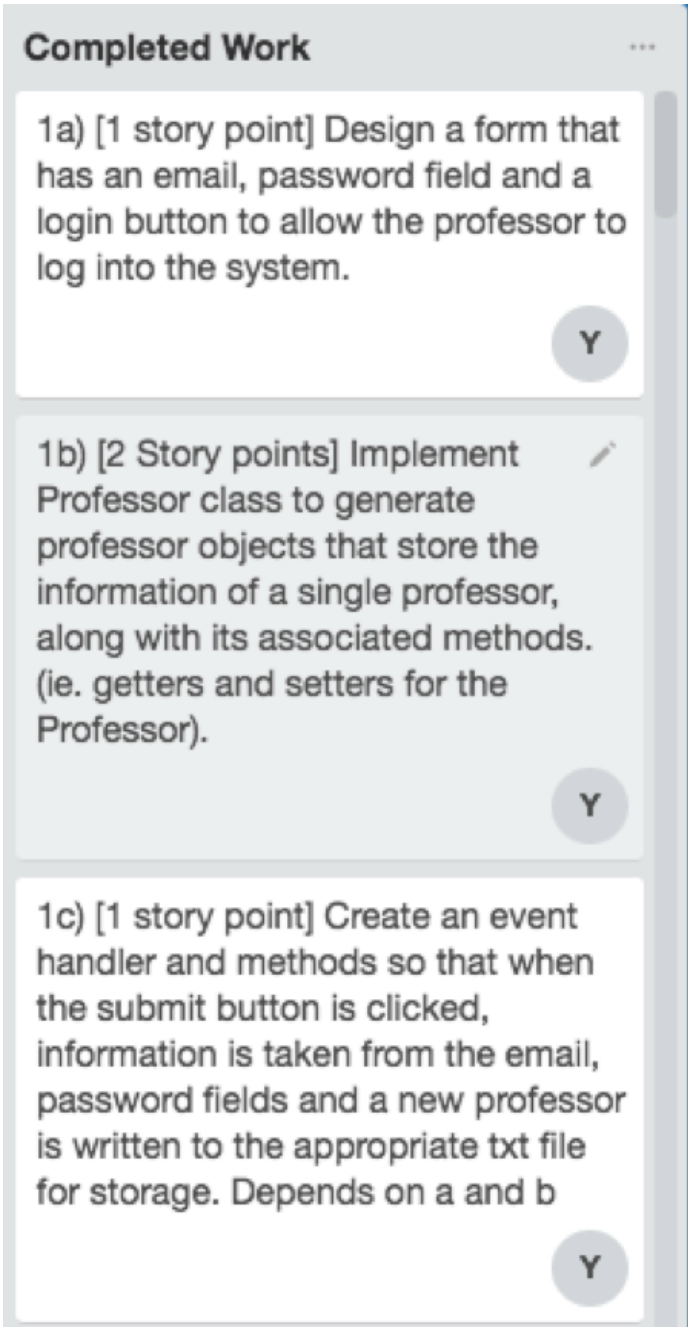
Start of Sprint

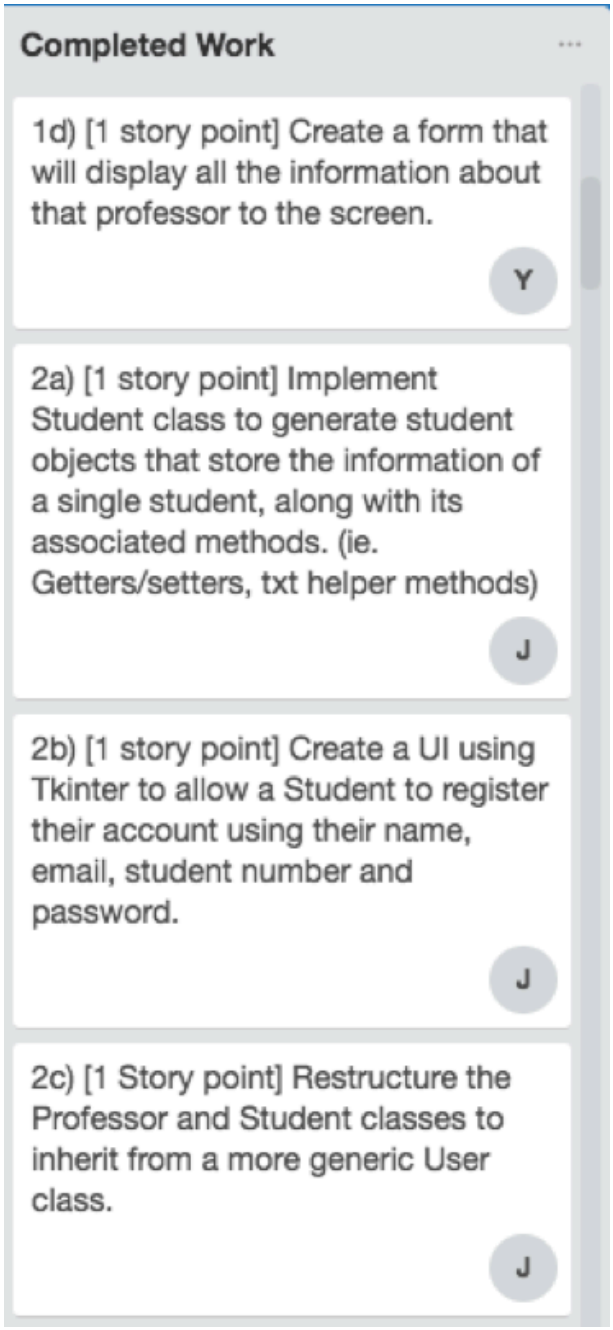
Each list has the task number, number of story points and the person responsible for the task. For example, the first one is task 6b, which requires 2 story point and is allocated to Eddie.
Y = Andy, J = Jason, ZL = Eddie, W = William, HF = Harrison
We all worked on the five different tasks for optimizing our final product.

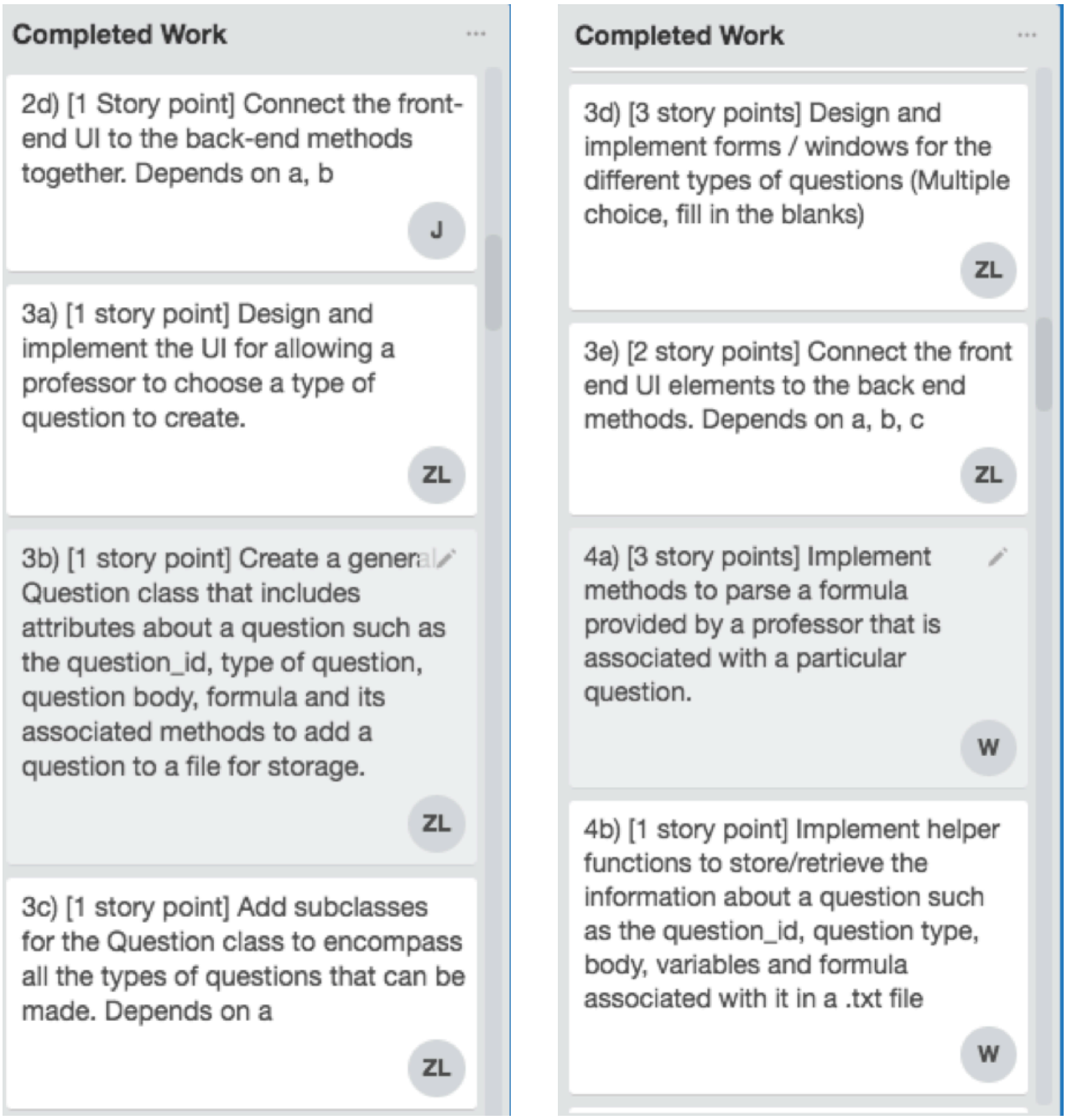


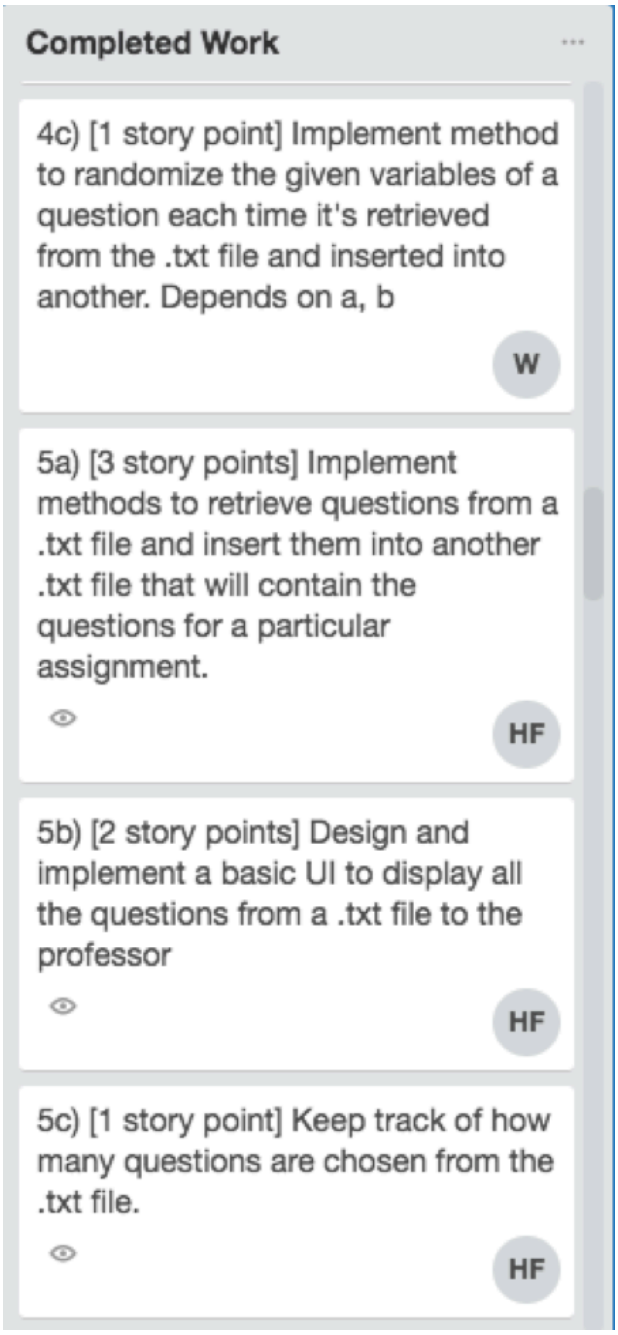
End of Sprint

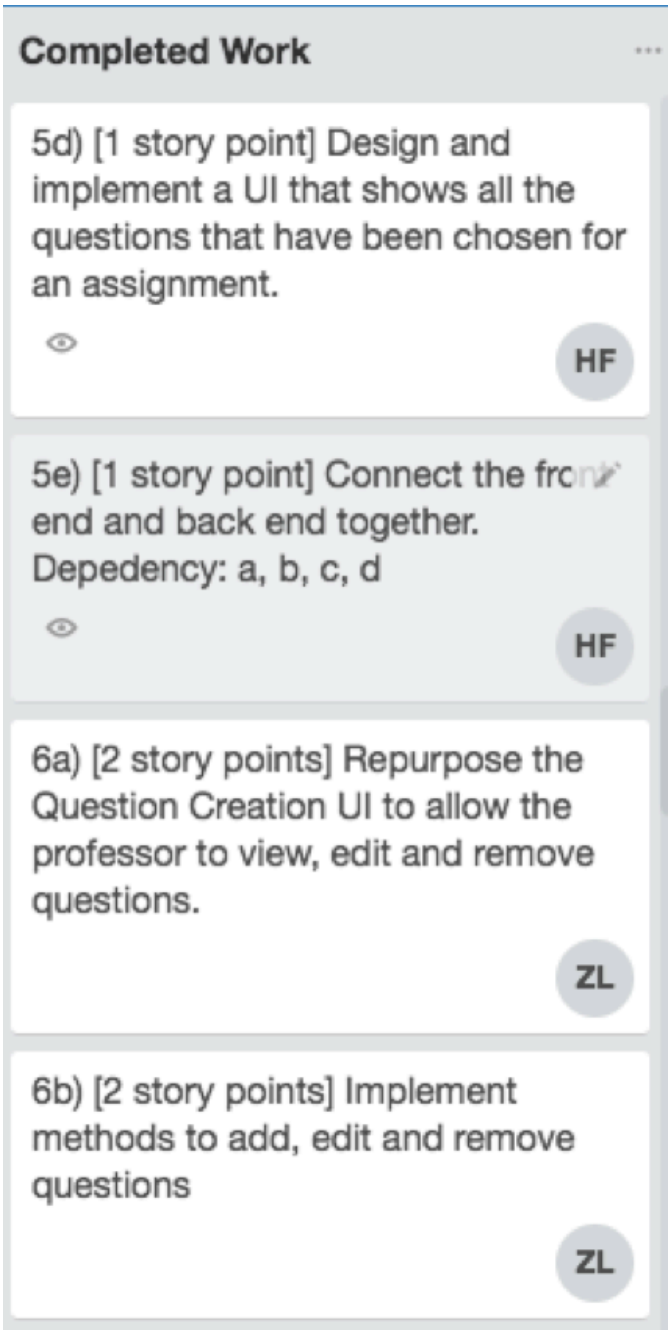
By the end of the sprint, all tasks of this sprint are finished and our final product is ready to be released.






Completed Work
2d) [1 Story point] Connect the front-end UI to the back-end methods together. Depends on a, b
J
3a) [1 story point] Design and implement the UI for allowing a professor to choose a type of question to create.
ZL
3b) [1 story point] Create a general Question class that includes attributes about a question such as the question_id, type of question, question body, formula and its associated methods to add a question to a file for storage.
ZL
3c) [1 story point] Add subclasses for the Question class to encompass all the types of questions that can be made. Depends on a
ZL
Completed Work
3d) [3 story points] Design and implement forms / windows for the different types of questions (Multiple choice, fill in the blanks)
ZL
3e) [2 story points] Connect the front end UI elements to the back end methods. Depends on a, b, c
ZL
4a) [3 story points] Implement methods to parse a formula provided by a professor that is associated with a particular question.
W
4b) [1 story point] Implement helper functions to store/retrieve the information about a question such as the question_id, question type, body, variables and formula associated with it in a .txt file
W


Completed Work
4c) [1 story point] Implement method to randomize the given variables of a question each time it's retrieved from the .txt file and inserted into another. Depends on a, b
W
5a) [3 story points] Implement methods to retrieve questions from a .txt file and insert them into another .txt file that will contain the questions for a particular assignment.
HF
5b) [2 story points] Design and implement a basic UI to display all the questions from a .txt file to the professor
HF
5c) [1 story point] Keep track of how many questions are chosen from the .txt file.
HF


Completed Work
5d) [1 story point] Design and implement a UI that shows all the questions that have been chosen for an assignment.
HF
5e) [1 story point] Connect the front end and back end together. Depedency: a, b, c, d
HF
6a) [2 story points] Repurpose the Question Creation UI to allow the professor to view, edit and remove questions.
ZL
6b) [2 story points] Implement methods to add, edit and remove questions
ZL

Completed Work

6c) [2 story points] Connect the repurposed front-end UI to the newly implemented back-end method. Depends on a, b

ZL

7a) [2 story points] Implement a new class OptionsMenu.py that will check whether a user is a professor or student.

Y

8a) [2 story points] Design and implement a UI using Tkinter to allow the student to see the questions as a list and have a corresponding text field to type their answer.

J

8b) [2 story points] Implement the back end methods to pull data from the csv file that stores the questions of an assignment and compares the student's answer with the one in the csv when they hit the submit button.

J

Completed Work

8c) [2 story points] AssignmentAttempts.csv will hold all the results of an assignment, overwriting the respective row with the most recent attempt.

J

9a) [2 story points] create a button in the AssignmentDashboard GUI to be displayed beside an assignment name that will take you to the 'visibility options menu', with radio buttons for visibility on and visibility off.

Y

9b) [2 story points] implement the back end methods to pull an assignment from the .csv that stores all the assignments and change it's attribute for visibility depending on what the professor chose.

Y

Completed Work

9c) [3 story points] connect the front and back end together. Dependent on a, b

Y

10a) [2 story points] Add a button that takes the AssignmentDashboard GUI menu for the professor beside each assignment.

W

10b) [3 story points] Implement back-end methods so that when the time length is set in the SetTimer GUI, and the timer is up, the visibility will be automatically turned off for an assignment.

W

10c) [2 story points] Connect the front end to the back end. Dependent on a, b

W

Completed Work

11a) [2 story points] Implement methods to iterate through the .csv file to find the particular assignment the student is working on.

HF

11b) [3 story points] Implement methods to extract the student's answers from AssignmentAttempts.csv, and compare these answers with the calculated answer in part a). Dependent on a)

HF

11c) [1 story point] Design and implement a UI using Tkinter that will display the results of a students answers when compared to the actual answer.

HF

11d) [1 story point] Connect the front-end and the back-end. Dependent on a, b, c

HF

Completed Work

11) Create Assignment.py and Question.py to make the whole program object-oriented

HF

12) Merge the assignment making feature into the master to incorporate the changes made in the format of the questions and assignments

HF

12) Connect all the current classes

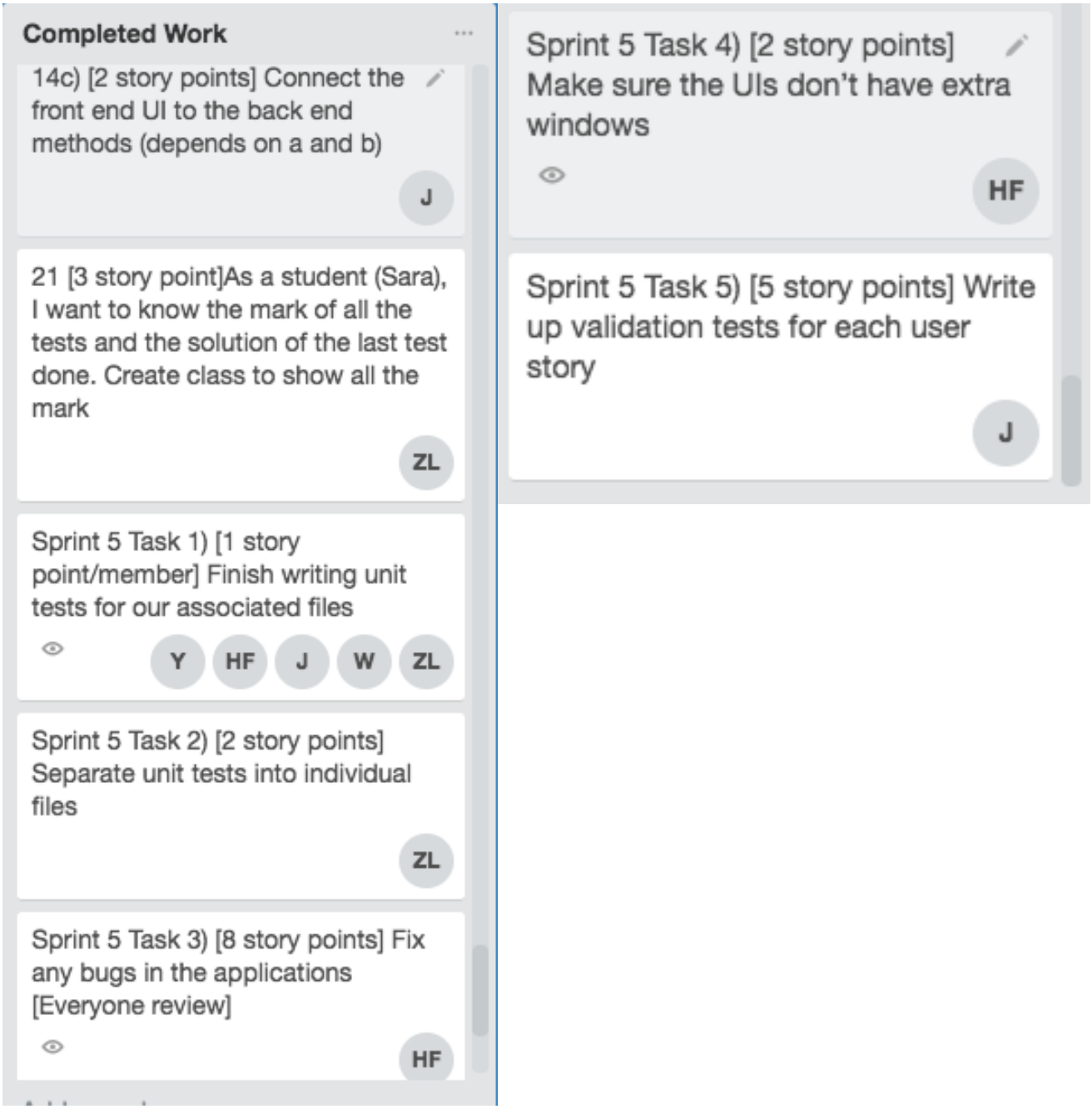
HF

14a) [2 story points] Design and implement a list layout using Tkinter to display all the assignments, with it's name, due date, number of questions and assignment ID for a specific student.

J

14b) [2 story points] Implement back end method to retrieve data from the csv file, format and display this data to the implemented UI. (dependent on a)

J


Completed Work
14c) [2 story points] Connect the front end UI to the back end methods (depends on a and b)
J
21 [3 story point]As a student (Sara), I want to know the mark of all the tests and the solution of the last test done. Create class to show all the mark
ZL
Sprint 5 Task 1) [1 story point/member] Finish writing unit tests for our associated files
Y HF J W ZL
Sprint 5 Task 2) [2 story points] Separate unit tests into individual files
ZL
Sprint 5 Task 3) [8 story points] Fix any bugs in the applications [Everyone review]
HF
Sprint 5 Task 4) [2 story points] Make sure the UIs don't have extra windows
HF
Sprint 5 Task 5) [5 story points] Write up validation tests for each user story
J