Sprint 4 Plan

Week of Nov. 6, 2017

User Stories	Tasks	Story Points	Dependencies
U2	T6	4	-
U6	T14	4	-
U7	T15	3	T14
U8	T16	2	T14
U9	T17	2	T16
U10	T18	3	-

Provisional Burndown Chart

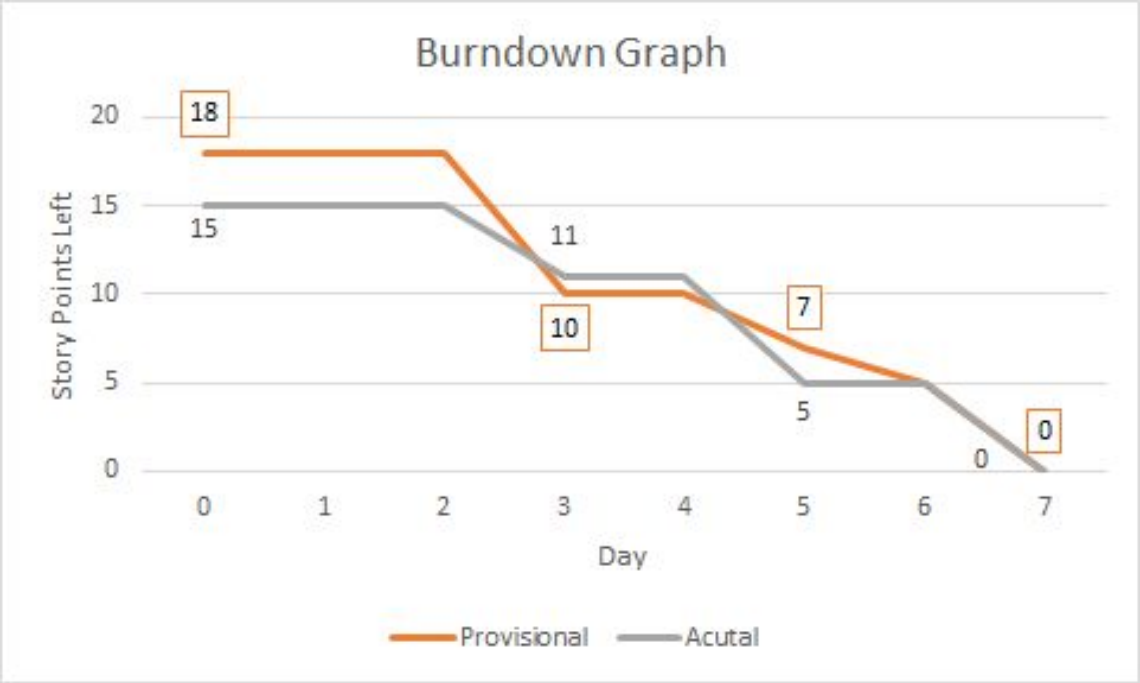
	Task Points	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
T6	4		J:3	J:1				
T14	4		F:3	F:1				
T15	3				O:2	O:1		
T16	2					O:1	O:1	
T17	2							A:2
T18	3							T:3

J - Julian [story points per day = 3]
A - Abhay [story points per day = 3]
F - Felix [story points per day = 3]]
T - Tito [story points per day = 2/3]
O - Olivia [story points per day = 2]

Final Burndown Chart

	Task Points	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
T6	6			J:2	J:1	J:3		
T14	4	F:3		F:1				
T15	3						O:2	O:1
T16	2						O:1	O:1 [AM]
T17	2							A:2 [PM]
T18	3						T:1	T:1

J - Julian [story points per day = 3]
A - Abhay [story points per day = 3]
F - Felix [story points per day = 3]
T - Tito [story points per day = 2/3]
O - Olivia [story points per day = 2]



Backlog

U2: As Karen/Ben (a statistics professor) and Jenny (a TA), I would like to be able to modify or remove an existing assignment question and/or solution.

T6: Create button per assignment listing to edit the corresponding assignment problems. The button ultimately takes the user to the same (or slightly modified) assignment creation UI (UI window 2), in which the assignment's data is retrieved from the file (assignment#.csv), but with "Save" at the bottom instead of "Create".

U6: As Dana (a student), I would like to be able to select an open problem set before its deadline has passed and begin answering questions.

T14: Create an "assignment interface" in which questions are randomly selected from the assignment .csv file (assignment#.csv) and displayed with their options as radio buttons.

U7: As Dana (student), I would like save my progress on my current assignment so I can work on it later.

T15: Create a “save progress” button on the assignment interface that adds the student’s attempted answers into a .csv file (assignment#_submissions.csv) for that assignment that contains one student per row. The format is as follows:

Student id (integer), q1 answer (integer 1-4), … qn answer, average mark (integer 0-100), # tries (integer), time spent in seconds (integer), final mark (integer 0-100)
Where n is the number of questions for that assignment.

U8: As Dana (student), I would like to be able to submit my completed assignment before the deadline has passed.

T16: Create a “submit” button that updates the student’s answers in the submissions file (assignment#_submissions.csv) to reflect their final submission and increments the number of tries.

U9: As Karen/Ben (a Statistics professor), I want every assignment submission to be auto-graded by comparing the saved solutions to the student’s final attempt.

T17: Create a function to parse through the submissions file (assignment#_submissions.csv) and compare each student’s solutions to the one in the assignment file (assignment#.csv) and saving their final mark in the submission file.

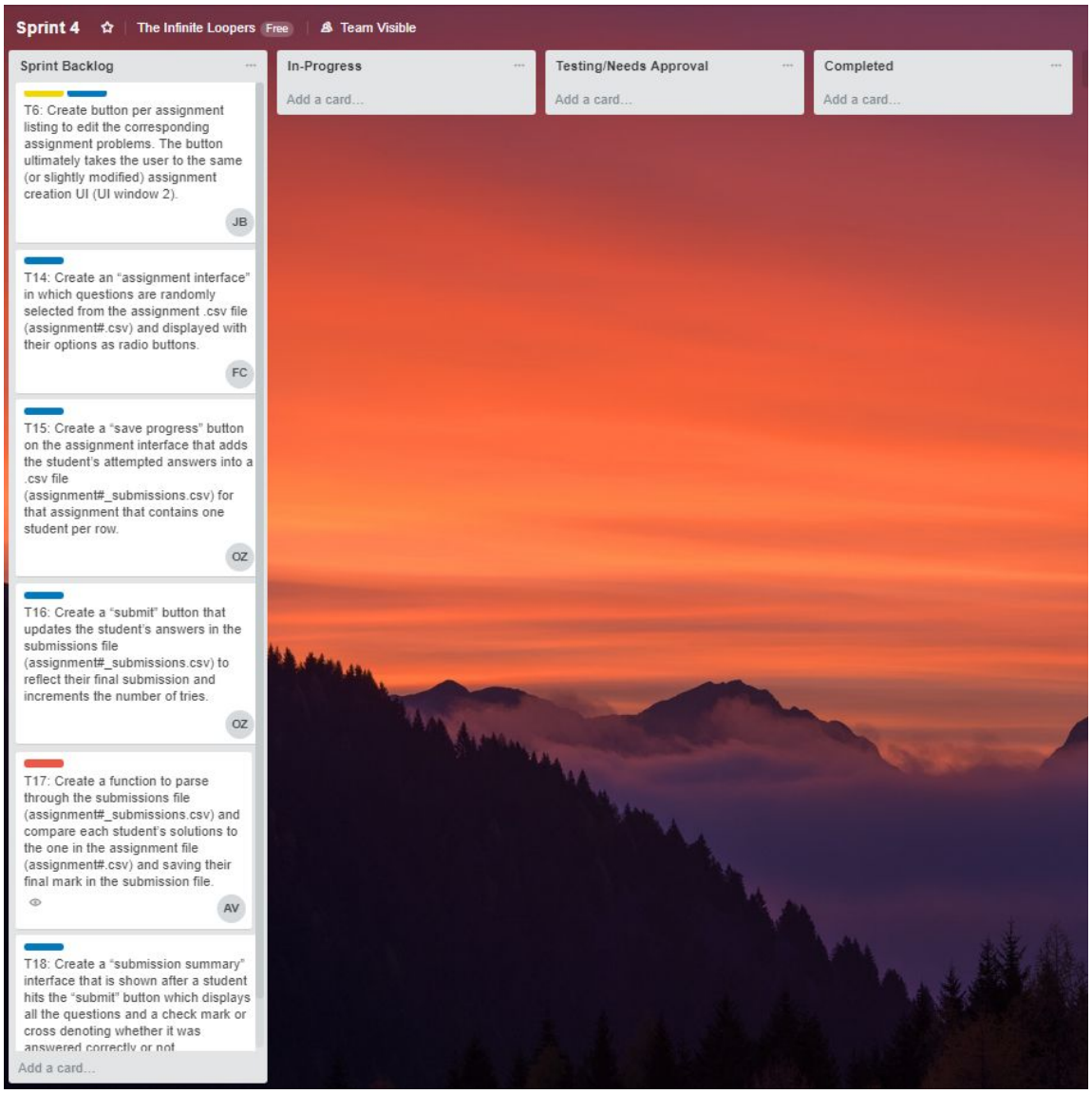
U10: As Dana (a student), I would like to see which questions on an assignment I got right and wrong after a submission so that I can retry the assignment if the deadline has not passed.

T18: Create a “submission summary” interface that is shown after a student hits the “submit” button which displays all the questions and a check mark or cross denoting whether it was answered correctly or not.

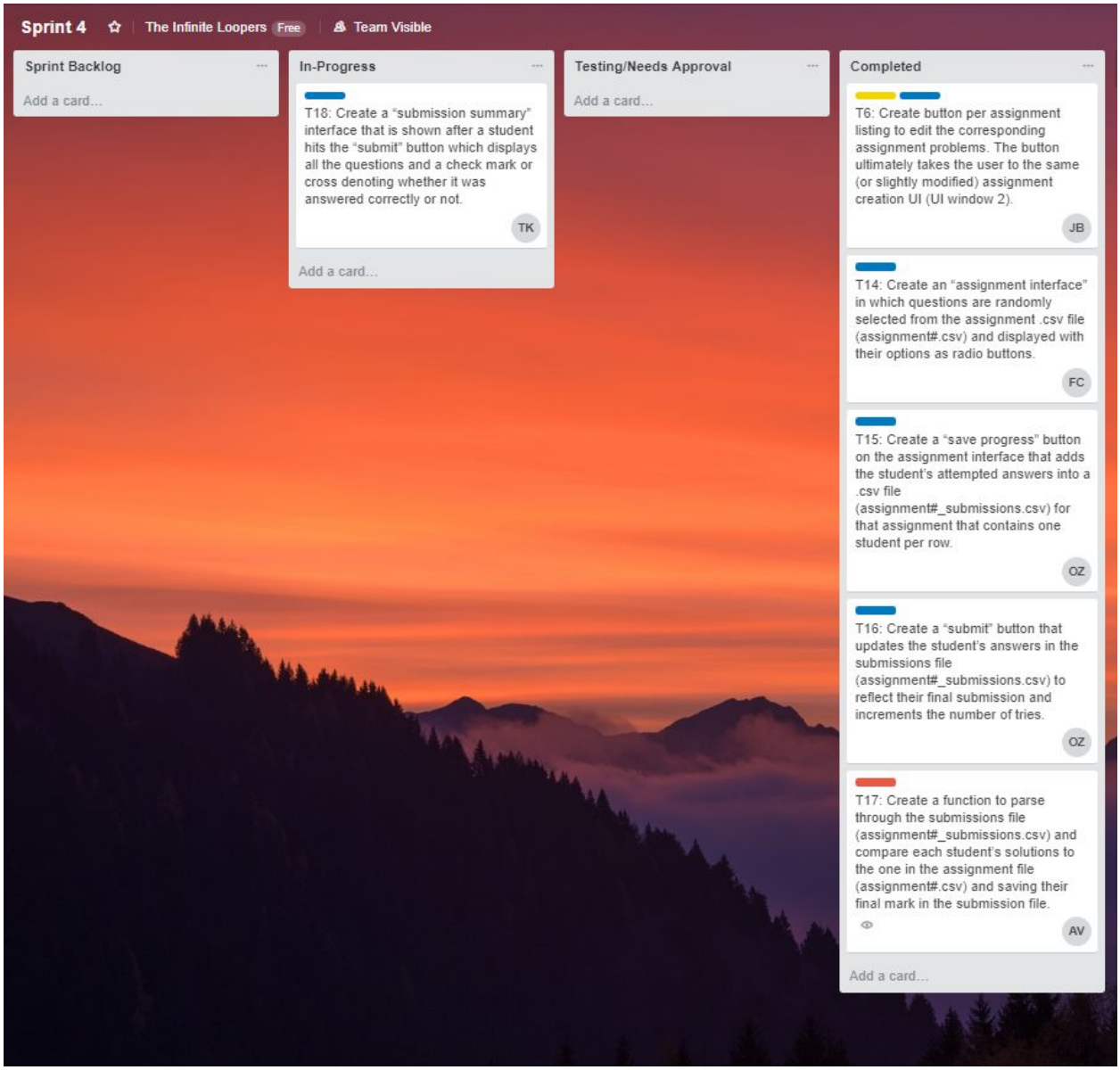
Report

- Julian created a GUI of the assignment editing feature.
- At run time, Julian implemented AssignmentEditingGUI to preset the default state of the file into the GUI of the editing window, i.e. If the assignment was created on January 15th, 2017, then the assignment editing window would display january 15th, 2017 and enough users to edit that.
- Julian added a JComboBox<String> to list every single assignment and give instructor users the option to manually modify each question, along with their multiple choice options and the solution. These components are are also preset to the current state of the assignment (see previous bullet).
- Felix created the initial interface for students to be able to complete an assignment
    - Initially made each multiple choice option a regular button, changed it to radio button
- Abhay initially wrote gradeSubmission() method in the AssignmentCompletionGUI.java to auto-grade a student's submission by comparing it with the last saved entries in the submissions file
    - Found bug in this logic and rewrote the method to use the student's answers straight from the radio buttons rather than read the file
- Tito wrote part of the submission summary GUI, but is incomplete currently
- After pass on T6 from Tito to Julian, worked together for a little bit on the editingGUI
- Olivia created a "save progress" button on the assignment interface that adds the student's attempted answers into a .csv file (assignment#_submissions.csv) for that assignment that contains one student per row.
- Olivia created a "submit" button that updates the student's answers in the submissions file (assignment#_submissions.csv) to reflect their final submission and increments the number of tries.

Taskboard Snapshots

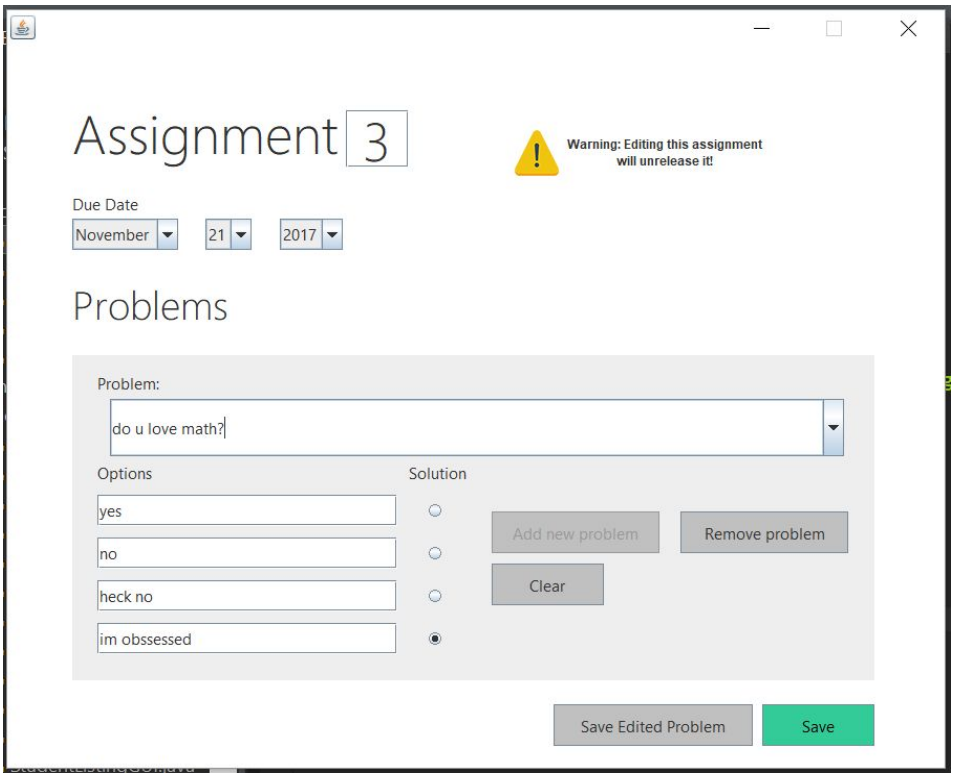


Before

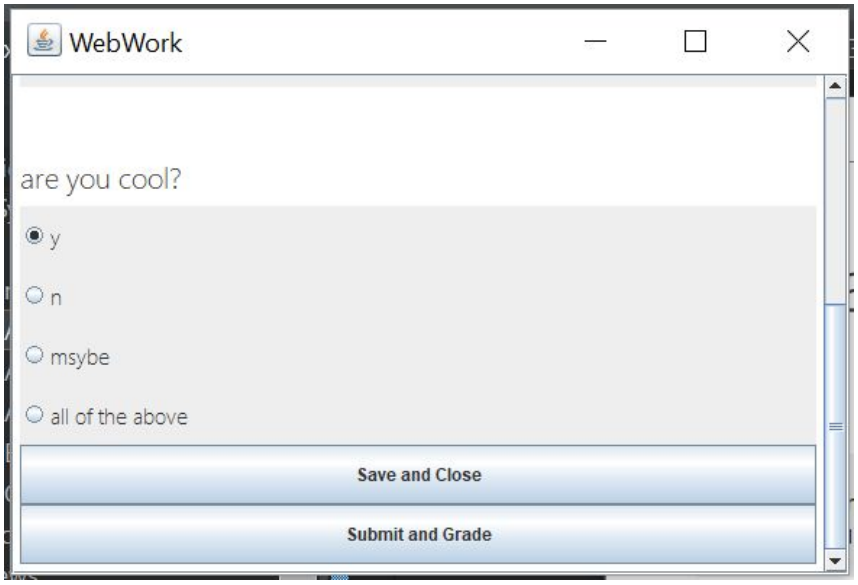

Sprint 4
The Infinite Loopers
Free
Team Visible
Sprint Backlog
Add a card...
In-Progress
T18: Create a "submission summary" interface that is shown after a student hits the "submit" button which displays all the questions and a check mark or cross denoting whether it was answered correctly or not.
TK
Add a card...
Testing/Needs Approval
Add a card...
Completed
T6: Create button per assignment listing to edit the corresponding assignment problems. The button ultimately takes the user to the same (or slightly modified) assignment creation UI (UI window 2).
JB
T14: Create an "assignment interface" in which questions are randomly selected from the assignment .csv file (assignment#.csv) and displayed with their options as radio buttons.
FC
T15: Create a "save progress" button on the assignment interface that adds the student's attempted answers into a .csv file (assignment#_submissions.csv) for that assignment that contains one student per row.
OZ
T16: Create a "submit" button that updates the student's answers in the submissions file (assignment#_submissions.csv) to reflect their final submission and increments the number of tries.
OZ
T17: Create a function to parse through the submissions file (assignment#_submissions.csv) and compare each student's solutions to the one in the assignment file (assignment#.csv) and saving their final mark in the submission file.
AV
Add a card...

After

Screenshots



Assignment editing window, where instructors have the option to edit existing assignments.



Window that enables students to complete assignments, and submit or save their answers.