

Milestone 2

Team 18: Kevin Nguyen(009), Yunlong Wang, Faith Lu, Wanquan Zhang, Sai Fu Lui, Zoe Qiao

Risk Analysis:

Risk: Low activity from team members

Description: When a team member/multiple team members are unable to keep up with the schedules and deadlines.

Severity: High

Resolution: Set up meetings to communicate with the team member(s) and identify the problem. After that, there should be communication with the rest of the team so that a plan can be devised to resolve the problem.

Status: Resolved

Risk: Team members leaving

Description: When a team member leaves the team for reasons such as dropping the class or health concerns.

Severity: Low

Resolution: Set up group meetings to discuss how the rest of the team can split and equally handle the work that was assigned by the leaving member.

Status: Resolved

Risk: Malfunctioning devices

Description: When a team member's working device breaks, he or she will not be able to continue working on the project.

Severity: High

Resolution: The team member should immediately try to find a new working device. Continuing work using the CSE lab machines would be the best and most efficient solution.

Status: Resolved

Risk: Inexperienced team members (with Android Development)

Description: If members have little prior knowledge of the Android development process, the development process may become significantly slower.

Severity: Medium

Resolution: The team members can discuss any problems that they encountered together using pair programming to improve understanding. Additionally, the team members can use provided resources such as labs, TA/professor office hours, or the textbook to become more familiar with the process of Android development.

Status: Resolved

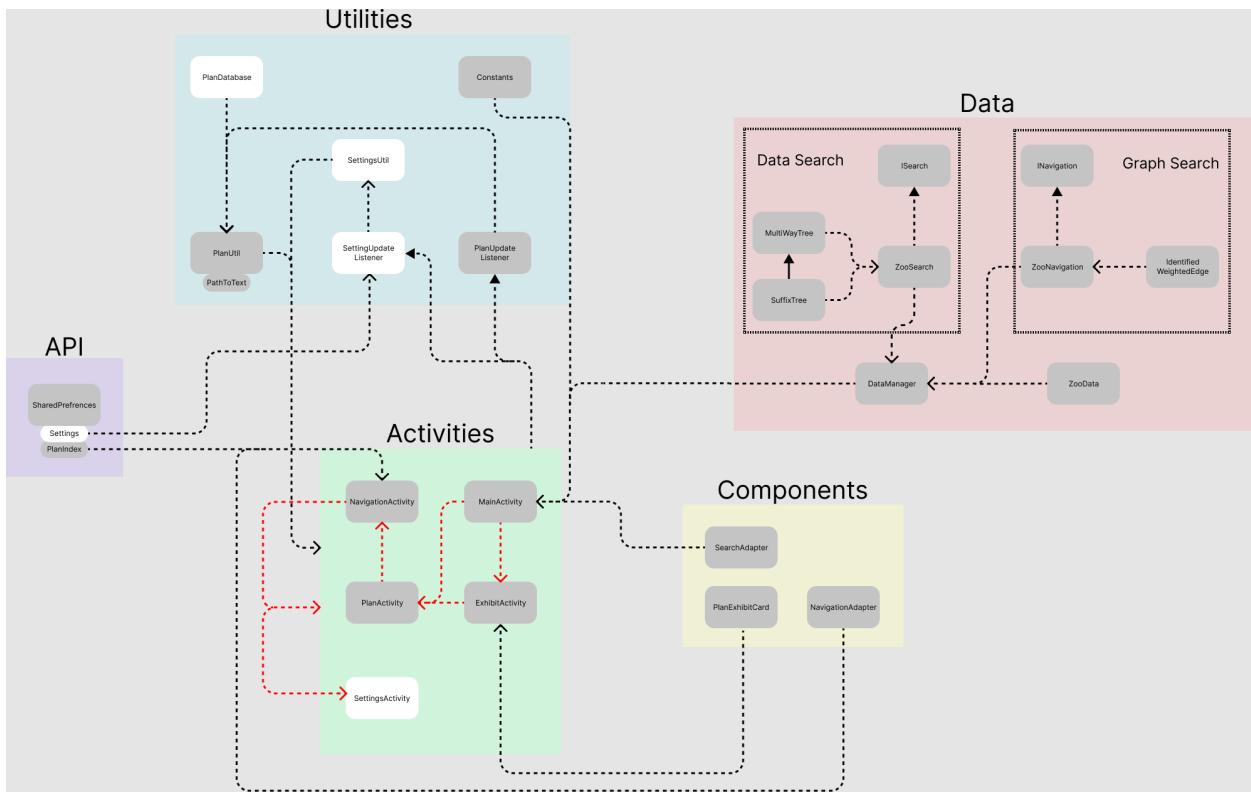
Risk: Unynchronized Code

Description: In the last Milestone, we spent an excessive amount of time refactoring our code for the final integration.

Severity: Medium

Resolution: We are making a outline of a unified class diagram during the plan phase to organize our development, so we would have a similar structure.

Status: Resolved



Risk: Unable to meet due to scheduling conflicts

Description: Due to some schedule conflicts, the team may not be able to collectively meet for major discussions.

Severity: Low

Resolution: Multiple meetings can be set up so we can still maintain general communication/discussion. We will also post general summaries of the meeting either through meeting notes on our Notion (our task assignments and notes platform) or on Discord (our communication platform). Additionally, absent members can share their suggestions after going through them. For meetings where all members have to be present, we can coordinate to be both in person and online, making it more convenient for members that might not be able to make it in person.

Status: Resolved

Velocity Calculation

Milestone 1 Tasks

Task	Initial Estimate	Actual Time Taken
1 - Creating a database to store all the data about the	6	3

Task	Initial Estimate	Actual Time Taken
animals		
2 - Creating a search bar	3	3
3 - Implement the functionality to look up data using the input from the search bar	3	4
4 - Support partial input search	5	3
5 - Implement a UI to display the search results	4	5
6 - Implement an UI for the list	3	4
7 - Implement the functionality to select the displayed exhibitions - UI implementation	3	3
8 - Implement the functionality to add/remove the selected exhibitions - Implement UI corresponding functionalities/behaviors	3	5
9 - Implement a button that allows the user to see the list	2	2
10 - Somewhere near the button, a number with the amount of exhibits on the list	1	1
11 - If the list is open, at the top the number of exhibits on the list should be created	1	0.5
12 - Implement a UI for the text plan with a direction button ready to be used later	6	9
13 - Implement the geographic functionality to look up data about how far each	8	7
14 - Implement the functionality that the exhibits are ordered from closest to the farthest location from the user's location	4	3
Extra helping	0	7
Total	52	59.5

Initial Velocity: .87

Justification: Using the MS1 task completion time tracking chart above, we can apply the time to the velocity formula, *actual hours / estimated hours*, to estimate the velocity for MS2. Since we have actually spent 59.5 hours, and we estimated 52 hours of work, we can estimate our MS2 velocity to be **52/59.5 = .87**

Iteration Length: 7 days

User Stories:

#	Title	As a...	I want...	So that...	Acceptance criteria (BDD scenarios)
9	Settings menu	Visitor	... to be able change how I see certain aspects of the app	...I can tailor the app to better suit my needs	<p>Scenario 1: default “Brief Directions” selected Given that the user opens the app for the first time When the user goes into the settings menu, the “Brief Directions” should be selected by default, and the user starts a plan Then the plan should have all the directions in the brief direction format.</p> <p>Scenario 2: Change level of detail for directions Given that the user is currently on "Brief Directions" When the user goes in the settings menu and selects "Detailed Directions" Then the plan should be automatically adjusted to have more detailed directions</p>
2	Search Bar Autocomplete	Visitor	... to see a list of suggested/possible exhibits based on what I have already typed	... I don't have to type the full name of each exhibit animal	<p>Scenario 1: Autocomplete related inputs Given that the user wants to find exhibits related to the “bear” search term When the user types in the full word “bear” Then a drop down box should automatically update with search terms like “black bear” or “panda bear” that could match the word the cursor is currently on, sorted by relevance. And selecting any of these search terms should display a list of exhibits that relate to these search terms.</p> <p>Scenario 2: Autocomplete incomplete inputs Given that the user wants to search “capybara” When the user types “capyb” into the search bar Then the dropdown page should show possible terms like “capybara,” sorted by</p>

					<p>relevance. And clicking on the term autofills the search bar with "capybara"</p> <p>Scenario 3: No autocomplete related exhibit is found for the search term Given that the user wants to input something that does not have an associated search term When the user starts typing in the input Then at the point where the input does not have an associated search term, the drop down box should automatically update to not show any terms and instead inform the user that there are no associated terms</p>
10	Backwards Button	Visitor	... to be able to see the instruction to go back to the previous exhibit from my current location	... I can find my way back to the previous exhibit	<p>Scenario 1: Go back to the previous exhibit Given that the user wants to head back to a previous exhibit "Arctic Foxes" When the user presses back while at "Elephant Odyssey" Then they should be given directions that lead to "Arctic Foxes". Specifically the instructions will be in reverse order of what was displayed for going from "Arctic Foxes" to "Elephant Odyssey"</p> <p>Scenario 2: Attempting to go back on the first exhibit Given that the user is currently on the first exhibit in the plan When the user attempts to go back to the previous exhibit Then the app should let the user know that there is no previous exhibit</p>
11	Skip Button	Visitor	... to be able to skip the current exhibit	... I can skip the exhibit that I no longer wanna visit even if it was selected to the list before.	<p>Scenario 1: Skip Exhibit Given that the user is not heading towards the last destination (gate) When the user presses the skip button on the Navigation page Then the next exhibit will be removed permanently and the app will navigate towards the destination after.</p> <p>Scenario 2: Attempt Skip Gate Given that the user is heading towards the last destination (gate)</p>

					<p>When the user presses the skip button on the Navigation page Then the next exhibit will not be removed and the user notified they are already heading towards the exit</p> <p>Scenario 3: Graying out Skip button in reverse order Given that the user is currently heading backwards (backtracking US 10) Then the skip button will be grayed out and not interactable.</p>
12	Display selected exhibits	Visitor	... to be able to show the selected exhibit on the list screen	... I can add or remove any exhibits on the list if things change	<p>Scenario 1: No selected exhibit Given that the user has not added any exhibit When the user opens up the list Then the list should be empty with 0 items</p> <p>Scenario 2: See Selected Exhibits Given that the user has added the "Arctic Foxes" and "Elephant Odyssey" exhibits to their plan When the user opens up the list Then the exhibits should be labeled as selected</p>
13	Route plan summary	Visitor	... to be able to see the simplified version of my route	... I can have a better idea of the overall route	<p>Scenario 1: Viewing route summary with existing exhibits Given that the user has selected at least one exhibit When the user presses on the Plan Summary button Then the user should see all the exhibits that they want to visit in order of the route plan and their distance.</p> <p>Scenario 2: View route summary with no existing exhibit Given that the user haven't selected any exhibit When the user presses on the Plan Summary button Then the user should see a text/message stating that no exhibit has been selected</p>
14	Track User's Current Location	Visitor	... to be notified of when I deviated from	... I know I'm following	<p>Scenario 1: Cannot access to the user's current location Given that the user opens the app</p>

			the plan	the plan correctly	<p>When the user starts the plan and the app cannot access to the user's current location Then a warning message displays "Cannot access your current location!"</p> <p>Scenario 2: User off route Given that the user is supposed be between Gorillas and Foxes When the user is currently at Elephants Then the app notifies the user they are off route and suggests to replan</p>
--	--	--	----------	--------------------	---

Story Priority:

Story 12: High
 Story 14: High
 Story 9: Medium
 Story 2: Medium
 Story 10: Medium
 Story 11: Medium
 Story 13: Low

Planning Poker documentation

Story	Task	Hand	Assumption	Final
User Story 9 Settings menu	19 - Create Gear Icon on each Activity	4/2/1/1/2/3	Assumed that we can just use Android's UI designer to create the icon	2
	20 - Create a dropdown box with options "Brief Directions", "Detailed Directions"	2/3/3/2/1/2	Assumed that the logic is already implemented	2
	21 - Modify Display Logic on Navigation Activity to consider instruction details	4/5/5/5/3/6	Assumed that both detailed directions and brief logic works properly	5
	22 - Implement SettingsUtil	4/5/4/4/2/4	Assumed that SettingsUtil focuses solely on updating the settings	4
User Story 2 Search Bar	15 - Search Bar Autocomplete	N/A	Backlog Story	4

Autocomplete				
User Story 10 Backwards Button	23 - Create Back Button	1/1/2/2/1/2	Assumed that the activity and its corresponding xml file already exist	2
	24 - Implement Backtracking Logic	3/5/4/4/2/4	Assumed that the logic is just to implement switch page when onClick	4
User Story 11 Skip Button	25 - Create Skip Button	1/1/2/2/1/2	Assumed that the skipping logic is already completed	2
	26 - Implement Skip Exhibit Logic	4/4/3/5/2/5	Assumed that it would be similar to the next button that we already had created	4
User Story 12 Display selected exhibits	27 - Display list of exhibits	4/3/3/3/3/4	Assumed that list of exhibits doesn't exist.	3
User Story 13 Route plan summary	29 - Generating the plan summary item	2/5/4/3/3/4	Assumed that the paths are generated	4
	30 - Implement UI using Recycler View showing the summary	3/4/5/3/2/3	Assumed that the summary has already been generated and it just needs to be properly displayed	4
	31 - Update the summary after user interactions	5/5/4/2/2/2	Assumed that the Next button works when the user progresses	3
User Story 14 Track User's Current Location	32 - Get user location from API	5/3/3/3/3/5	Assumed that the API is available and it works	4
	33 - Detects when user is off track and prompt to replan	8/3/4/6/3/4	Assumed that the user must be detected in real time	6
Bug Fixes and Refactoring	34 - Rework TextToPath Algorithm to fit Detailed Directions & Brief Directions Specification	4/3/4/3/3/6	Assumed that TextToPath is fixed when traveling in both directions and detailed directions already implemented	4

	35 - Fix TextToPath Algorithm to consider direction of travel on an edge	4/3/5/4/1/5	Misunderstanding in what the task was asking for	2
	36 - Double Tap to add Exhibit	4/2/3/1/2/3	Assumed that double tap is detectable on the device	3
	37 - Refactor Class Structure	5/4/4/2/4/4	Assumed that we would just use the diagram we created to help refactor the code.	4
	38 - Remove Toasts	4/2/3/1/1/3	Assumed that the information displayed by the toasts is provided elsewhere	1
	39 - Implement PlanDatabase for PlanUtil	5/5/4/3/4/4	Assumed that the database knowledge from the lab is sufficient	4
Total				70

Photo of you playing planning poker (Zoom screenshots are OK)



Tasks for the Stories in your first Iteration

Story #12: Display selected exhibits

Task 27: Display list of exhibits

Story #14: Track user's current location

Task 32: Get user location from API

Task 33: Detects when user is off track and prompt to replan

Story #2: Search bar autocomplete

Task 15: Search bar autocomplete

Bug Fixes and Refactoring

Task 34: Rework TextToPath Algorithm

Task 35: Fix TextToPath Algorithm

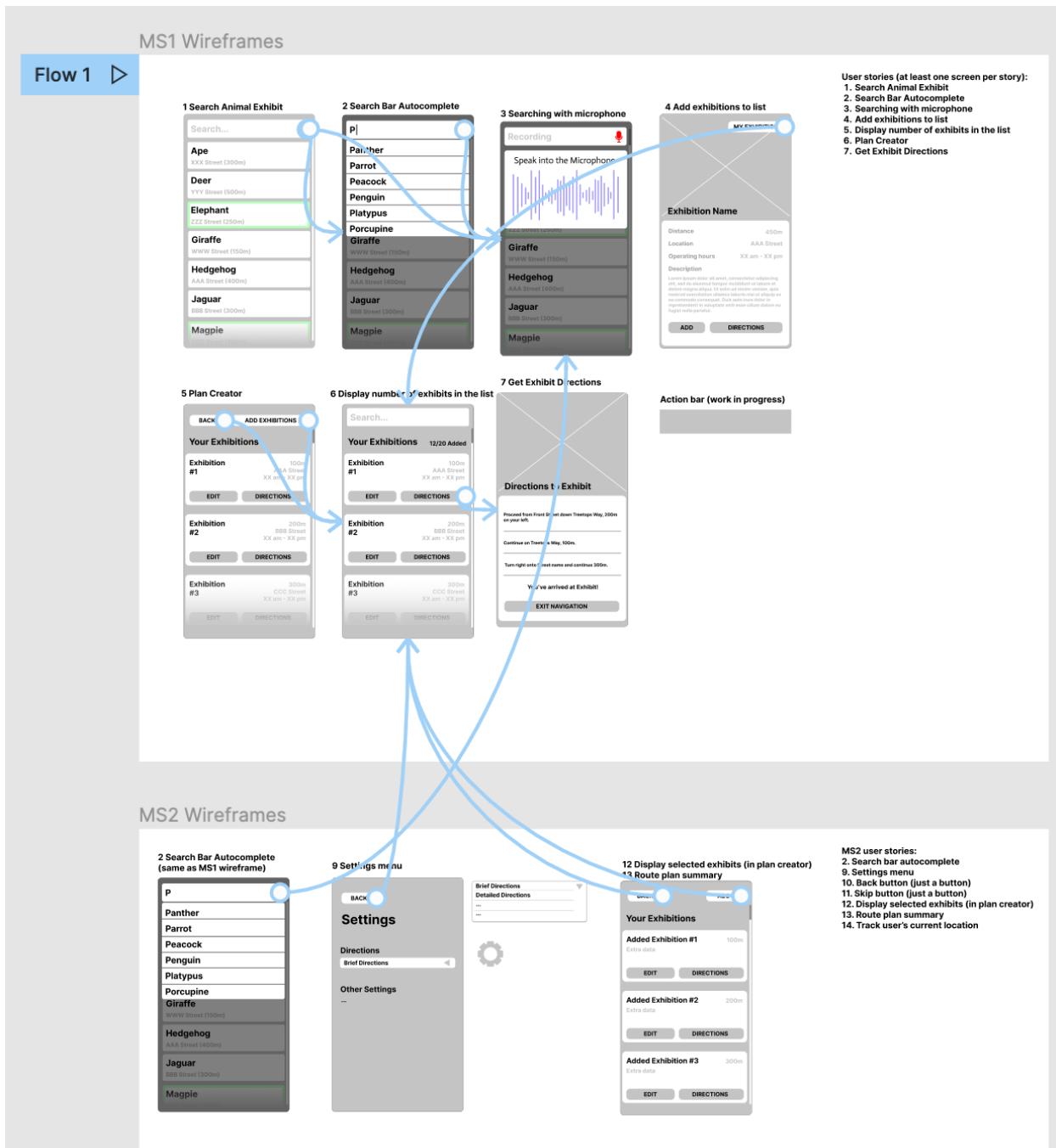
Task 36: Double Tap to add Exhibit

Task 37: Refactor Class Structure

Task 38: Remove Toasts

Task 39: Implement PlanDatabase for PlanUtil

Wireframes



Scenario-Based System Tests [4%]

SBST 3: Detailed Directions and Next, Previous, and Skip buttons

1. Open the app by tapping on the app icon, there should be a search bar at the top of the screen. (US#1)
2. Type in an animal/exhibit name and a list with the relevant animals/exhibits should appear below the search bar. Example: Search Bear, Different types of bears show up on the list. (US#2)

3. Select one of the options on the list, and it should get added to the list of exhibits. (US#4)
4. Double tap on an existing exhibit on the list, and it should get removed from the list. (US#4)
5. A number of added exhibits should be displayed at the top of the list (US#5)
6. Tap plan and a plan is created from the list of exhibits, with each exhibit having the street/trail name and distance below it. The exhibits should be ordered from closest to farthest. (US#6)
7. Tapping the directions buttons should display simple directions to the first exhibit on the list from the user's current location. (US#6,7)
8. Tap a gear icon to open a menu to display all the settings.(US#9)
9. An option that displays simple directions should already be selected. Select detailed directions and click on the X to close the menu.(US#9)
10. Detailed directions should now be displayed where the simple directions were displayed originally.(US#6,7,13)
11. Moving away from the specified directions, the detailed directions should be updated in relation to where the user is currently located.(US#12,13,14)
12. Once at the new exhibit, click on next to get the detailed directions to the next exhibit.(US#7,8)
13. Now press previous to get detailed directions to the previous exhibit from the current location.(US#7,10)
14. Now press skip to skip over the next exhibit and get directions to the exhibit after that. (US#7,11)
15. Keep pressing the skip button until it reaches the second to last exhibit; this should display directions to the exit gate. If it is the very last exhibit in the list, then the button should be grayed out and should do nothing. (US#7,11)
16. Click the end plan button to end the plan and to return to the search menu screen. (US#7)

SBST 4: Replan Option

1. Open the app by tapping on the app icon. There should be a search bar at the top of the screen. There should also be a microphone on the right side of the search bar. (US#1,3)
2. Tap on the microphone icon, it should turn red and a drop-down should appear with a sound wave showing if you are speaking or not. The search bar should also display the text "Recording". (US#3)
3. Speak an exhibit name into the microphone. The sound wave should be reacting to your voice. Words of what you said should also appear in the search bar. (US#3)
4. Double tap on an existing exhibit on the list, and it should get removed from the list. (US#4)
5. Tap on the microphone again when done speaking the exhibit name. The microphone should turn gray again and the sound wave should disappear. (US#3)
6. The exhibits relating to the term in the search bar should be displayed below the search bar. (US#1)

7. Select one of the options on the list, and it should get added to the list of exhibits. (US#4)
8. A number of added exhibits should be displayed at the top of the list (US#5)
9. Tap plan and a plan is created from the list of exhibits, with each exhibit having the street/trail name and distance below it. The exhibits should be ordered from closest to farthest. (US#6)
10. Tapping the directions buttons should display simple directions to the first exhibit on the list. (US#6,7)
11. Detailed directions should now be displayed where the simple directions were displayed originally.(US#6,7,13)
12. Move away from where the directions are pointing to the next exhibit, so that you are closer to a different exhibit on the list. Once you are closer to the other exhibit, a prompt should appear saying “Off track, Replan?” with an option saying yes or no. (US#13,14)
13. Tap on yes, and the plan should be changed to the nearest exhibit on the plan. The detailed directions should be changed to display the new detailed directions to the new closest exhibit. (US#13,14)
14. Go to the exhibit on the directions and then click on the next button to get directions to the next exhibit on the new adjusted plan. (US #7)
15. Move away from the next exhibit so that you are closer to a different exhibit on the plan. (US #14)
16. The same off track prompt should pop up again, but this time select no. The prompt should disappear and the directions should have not changed. (US#13,14)
17. Keep pressing next until the last exhibit. (US #7)
18. Press end plan to end the plan, it should return back to the search menu. (US #7)

Two Iterations for Milestone (and Backlog for overflow) [5%]

Prioritized User Stories and Developer Stories assigned to named Iterations

- **Iteration1:**
 - Story #12: Display selected exhibits
 - Story #14: Track user's current location
 - Story #2: Search bar autocomplete
 - Bug Fixes and Refactoring
- **Iteration2:**
 - Story#9: Settings menu
 - Story#10: Backwards button
 - Story#11: Skip button
 - Story#13: Route plan summary

Discussion of why Backlog Stories didn't fit into Milestone (based on estimates and Velocity):

Backlog Stories are the risks we could potentially have during the development process. We are unable to accurately fit them into our initial velocity estimate for this Milestone because they are not predictable at the beginning of the planning phase. User Story #2 is the only story from

Milestone 1 planning that we might consider as a backlog story (although it was postponed, so we aren't completely sure if it counts as one).

Population of ZenHub with the above [3%]

<https://app.zenhub.com/workspaces/cse-110-team-18-project-625f89b811672c0019432e02/board>