Milestone 1

Team 18: Kevin Nguyen , 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: In progress

Risk: Team members leaving
Description: When a team member leaves the team for reasons such as dropping the class or health concerns.
Severity: High
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: In progress

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: In progress

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: In progress

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

Initial Velocity: 0.7
Justification: Given that we are still in the planning phase of the project and have not worked on actual hands-on development, we have no other basis for velocity besides an initial value of 0.7. Once we have assigned development tasks and started productive work on the actual product in our first iteration, we will adjust our velocity accordingly.

Iteration Length: 7 days

User Stories:

#	Title	As a…	I want…	So that…	Acceptance criteria (BDD scenarios)
1	Search Animal Exhibits	Visitor	… to be able to search for the animals exhibits using names (or other characteristics)	…I can see a refined list of exhibits	**Scenario 1: Search for an exhibit** **Given** that the user typed "tiger" into the search bar **When** the user presses "enter" **Then** the list should update to display exhibits that 'match' exhibits related to "tiger" (the list should simply display every exhibit if the search term is empty) **Scenario 2: Cancel the search** **Given** that the user typed "dog" into the search bar **When** the user clicks on the cancel button **Then** the virtual keyboard and the word "dog" in the search bar disappears **Scenario 3: No exhibit is found for the search term** **Given** that the user typed in a search term that has no related exhibit **When** the user presses "enter" **Then** the results screen should not show any exhibits and instead inform the user that there are no associated exhibits
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	**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. **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 relevance. **And** clicking on the term autofills the search bar with “capybara” **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
3	Searching with microphone	Visitor	… to be able to search for animals by using my voice	…I can be faster in my searching	**Scenario 1: Search exhibits via voice** **Given** that the user has clicked on the mic button in the search bar **When** the user says “lion” **Then** the search bar should be filled with the text “lion” **And** the drop-down box should show related search terms like “mountain lion,” sorted by relevance. **Scenario 2: Cancel the search** **Given** that the user has selected the mic button **When** the user selects anywhere on the screen **Then** voice detection stops
4	Add/Remove exhibitions to list	Visitor	…to be able to add exhibits I want to visit to a list	…I can keep track of all the animals and exhibits that I want to visit.	**Scenario 1: Show exhibit detail** **Given** that the user is on the search page **When** the user selects a particular exhibit shown on the list **Then** the user is redirected to a page showing more details about the exhibit and the option to view directions and add the exhibit to the plan **Scenario 2: Add exhibit to list without redirection** **Given** that the user is on the search page

					When the user double taps on a particular exhibit shown on the list **Then** the exhibit is immediately added in the plan and the user is not redirected to the exhibit page **Scenario 3: Remove exhibit to list** **Given** that the user is on the planner page **When** the user double taps on the dotted button next to the exhibit that the user wants to delete from then list **Then** the exhibit is immediately removed in the plan and the user is not redirected to the exhibit page
5	Display number of exhibits on the list	Visitor	… to be able to keep track of how many exhibits are on my list	…I know if there are too many or too few exhibits on the list.	**Scenario 1: Check the number of exhibits in the list** **Given** that the user is on the planner page **When** the user scrolls up all the way to the top **Then** the user will see the number of exhibits on the list
6	Plan Creator	Visitor	…to be able to create a plan from the list of animals I created previously	…I can efficiently visit all the exhibits on my list	**Scenario 1: Planning visiting order** **Given** that the user has selected a list of 5 exhibits **When** clicking on the plan creator button **Then** the user will see the visiting order of exhibits **Scenario 2**: **No exhibits in the list** **Given** that the user has not added any exhibit to the list **When** clicking on the plan creator button **Then** the user will see the message requiring adding exhibits to the list
7	Get Exhibit Directions	Visitor	…to get directions for the closest selected exhibit	…I know what path I need to take to get there	**Scenario 1: Get directions to exhibit that is on the list** **Given** that the user added the elephant exhibit **When** the user clicks on the elephant exhibit item on the list **Then** a new page with direction will appear **Scenario 2: Get directions to an**

					exhibit at the end of the list **Given** that the user added the zebra exhibit, which is the last exhibit on the list **When** the user arrives at the destination **Then** there should be a notice saying that no more exhibits are on the list with directions to the exit
8	Next Button	Visitor	…to move on to the next exhibit on the plan with directions to it.	…I can get new directions for the next exhibit.	**Scenario 1: Clicking on the next button** **Given** the user is viewing the direction for one exhibit on its dedicated page **When** the user clicks on the next button **Then** the current page switches to the next exhibition page on the planner **Scenario 2: The last exhibit in the list** **Given** that the user just finished their last exhibit in the list **When** the user clicks on the next button **Then** a message pops up saying that there is no planned exhibit in the list, and brings it back to the search bar. **Scenario 3: Starting the plan** **Given** that the plan has some exhibits and has not started yet **When** the user clicks the Next button **Then** the plan will begin to direct the user and can no longer be modified

Story Priority:

Story 1: High
Story 4: High
Story 6: High
Story 2: Medium
Story 7: Medium
Story 8: Medium
Story 3: Low
Story 5: Low

Story Prerequisites

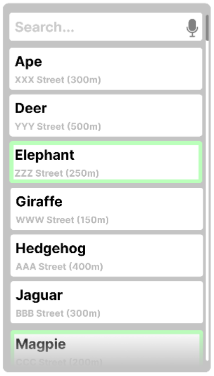
- Stories 2 and 3 require Story 1
- Stories 1, 2, and 3 are all related
- Stories 4 and 5 are related
- Story 6 requires Story 4
- Stories 6 and 7 are related

Wireframes

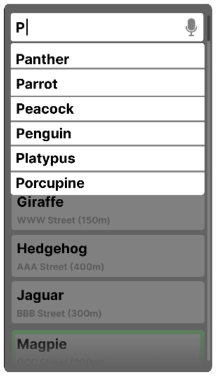
User stories (at least one screen per story):

1. Search Animal Exhibit
2. Search Bar Autocomplete
3. Searching with microphone
4. Add exhibitions to list
5. Display number of exhibits in the list
6. Plan Creator
7. Get Exhibit Directions

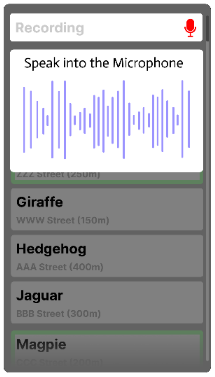
1 Search Animal Exhibit



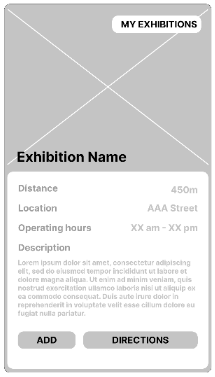
2 Search Bar Autocomplete



3 Searching with microphone

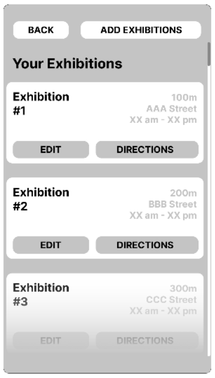


4 Add exhibitions to list

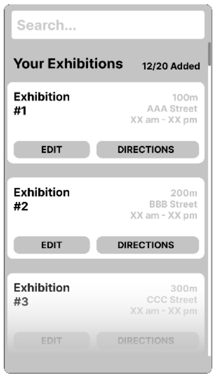


Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

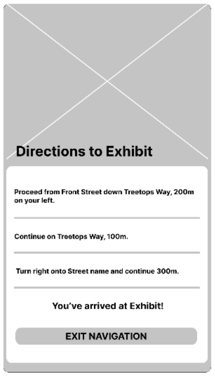
5 Plan Creator



6 Display number of exhibits in the list



7 Get Exhibit Directions



Proceed from Front Street down Treetops Way, 200m on your left.

Continue on Treetops Way, 100m.

Turn right onto Street name and continue 300m.

You've arrived at Exhibit!

Action bar (work in progress)

Planning Poker

Story	Task	Hand	Assumption	Final
1 **Search Animal Exhibit** **(Total: 21)**	1 - Creating a database to store all the data about the animals	3/5/5/8/8/10	Assumed the database should be written using Mysql which requires more time to accomplish Assumed there would be too many exhibits to use a smaller data structure (when realistically there would not)	6
	2 - Creating a search bar	2/2/2/2/2/5	Assumed some functions for search bar autocomplete will be done here	3
	3 - Implement the functionality to look up data using the input from the search bar	1/5/8/8/8/10	Assumed look up functionality would be a method implemented in database	3
	4 - Support partial input search	1/3/5/5/8/10	Assumed look up functionality would be a method implemented in database Assumed that the data we have is categorized	5
	5 - Implement a UI to display the search results	2/2/3/5/8/8	Assumed developers' difference in experience designing Android UI	4
2 **Search Bar Autocomplete** **(Total: 8)**	N/A	2/2/5/8/13/13	Assumed partial input search is implemented correctly	8
3 **Searching with microphone** **(Total: 5)**	N/A	2/5/5/8/8/13	Assumed that we can use voice recognition API	5

4 **Add/Remove exhibitions to list** **(Total: 9)**	6 - Implement an UI for the list	1/2/3/3/5/5	Assumed developers' difference in experience designing Android UI	3
	7 - Implement the functionality to select the displayed exhibitions	2/3/3/3/3/3	Assumed that we agreed on very similar functionality concepts	3
	8 - Implement the functionality to add/remove the selected exhibitions	2/2/2/3/5/5	Assumed that the database is implemented	3
5 **Display number of exhibits on the list** **(Total: 4)**	9 - Implement a button that allows the user to see the list	1/1/1/2/3/3	Assume that the list UI is completed	2
	10 - Somewhere near the button, a number with the amount of exhibits on the list	1/1/1/1/2/3	Assumed that the database is implemented	1
	11 - If the list is open, at the top the number of exhibits on the list should be created	1/1/1/1/2/2	Assumed that the database is implemented	1
6 **Plan Creator** **(Total: 18)**	12 - Implement a UI for the text plan with a direction button ready to be used later	3/3/3/3/8/8	Assumed developers' difference in experience designing Android UI	6
	13 - Implement the geographic functionality to look up data about how far each	5/5/5/10/10/13	Assumed that the algorithm is provided	8
	14 - Implement the functionality that the exhibits are ordered from closest to the farthest location	2/3/5/8/10/13	Assumed algorithm for search is provided	4

	from the user's location			
7 **Get Exhibit Directions** **(Total: 5)**	N/A	3/5/8/8/13/13	Assumed path is pre-generated upon the creation of the plan (Task 13) using the provided algorithm	5
8 **Next Button** **(Total: 2)**	N/A	1/2/2/2/5/10	Assume button simply modifies an index number Display information and other functions handled elsewhere	2
Total time needed: 72 hours				

Photo when we were playing planning poker:

Tasks for the Stories in your first Iteration:

Story #1: Search Animal Exhibit

Task 1: Creating a database to store all the data about the animals
Task 2: Creating a search bar
Task 3: Implement the functionality to look up data using the input from the search bar
Task 4: Support partial input search
Task 5: Implement a UI to display the search results
Story Test Task 1:

Story #4: Add/Remove exhibitions to list

Task 6: Implement a UI for the list
Task 7: Implement the functionality to select the displayed exhibitions
Task 8: Implement the functionality to add/remove the selected exhibitions

Story #5: Display number of exhibits on the list

Task 9: Implement a button that allows the user to see the list

Task 10: Somewhere near the button, a number with the amount of exhibits on the list should be displayed in a circle.

Task 11: If the list is open, at the top the number of exhibits on the list should be created

Story #6: Plan Creator

Task 12: Implement a UI for the text plan with a direction button ready to be used later

Task 13: Implement the geographic functionality to look up data about how far each exhibit is from the user's location

Task 14: Implement the functionality that the exhibits are ordered from closest to the farthest location from the user's location

Scenario-Based System Tests:

SBST 1:

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. (US#6,7)
8. There should be a next button below the directions that displays the next exhibit on the list with the distance that it is away. (US#8)
9. Tapping the next button should display the directions for the next exhibit on the list. (US#7,8)
10. If there are no more exhibits on the list, tapping the next button should bring it back to the search bar, (the screen that is shown when the app is first opened). (US#8,1)

SBST 2:

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. There should be a next button below the directions that displays the next exhibit on the list with the distance that it is away. (US#8)
12. Tapping the next button should display the directions for the next exhibit on the list. (US#7,8)
13. If there are no more exhibits on the list, tapping the next button should bring it back to the search bar, (the screen that is shown when the app is first opened). (US#8,1)

Two Iterations for Milestone:

1. Prioritized User Stories and Developer Stories assigned to named Iterations
    - Iteration1:
        - Story#1 Search Animal Exhibit
        - Story#4 Add exhibitions to list
        - Story#5 Display number of exhibits on the list
        - Story#6 Plan Creator
    - Iteration2:
        - Story#2 Search Bar Autocomplete
        - Story#3 Searching with microphone
        - Story#7 Get Exhibit Directions
        - Story#8 Next Button
2. 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 can not count it into the estimates of the velocity for the Milestone because they are not predictable at the beginning of our plan. And for our first Milestone planning, we do not have any backlog stories for now.

Population of ZenHub with the above:

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