

1. Risk Analysis

Initial Velocity Estimate: 0.5

While developing our project, we know that there will be small common interruptions that will affect our productivity, such as needing to talk to team members during work periods, fatigue from being stuck on bugs, etc. Our velocity is conservative such that we may set a baseline for a more accurate velocity estimation after our first iteration.

Iteration: 7 days

Risk: Not Enough Meetings

Description: Due to schedule conflicts, we don't have any good mutually available hours except for weekends, so we may not end up meeting enough.

Severity: Medium

Resolution: We meet over Discord (voice and video calls) to work on the project, use WhenToMeet to help find windows where we can get together.

Status: In Progress

Risk: Familiarity with Java

Description: Not all members are familiar with Java syntax and the specific definitions or behaviors

Severity: Low

Resolution: Team members that know Java will mentor/assist those that are not as familiar with the programming language

Status: Resolved

Risk: Familiarity with Git (version control)

Description: Not everyone is familiar with the proper usage of version control, and avoiding version conflicts.

Severity: Medium

Resolution: Review and understand lab tutorial on Git/GitHub, search for commands if needed, communicate with teammates if unsure of usage. Communicate with teammates and avoid duplicate work or multiple people working on the same files to keep merge conflicts to a minimum. Resolve conflicts when we get to it, practice good Git etiquette, and use safeguards like tests, reviewers, preventing unreasonable force pushes, etc.

Status: Resolved

Risk: Android Studio Experience

Description: Our team is not very familiar with Android Studio, and have had issues with the physical device/emulators.

Severity: Medium

Resolution: Go over old labs, try hard in new labs.

Status: In progress

Risk: Balancing Available time

Description: All of us are students with many other commitments of varying and inconsistent intensity. Finding the time to consistently complete project milestones may be difficult.

Severity: High

Resolution: Communicate when we can't pull 100% of our weight so the others can pick up the slack, keep reminders via programs like Google Calendar to dedicate time to project

Status: Resolved

Risk: Ensuring progress

Description: Ensuring that everyone is making some form of progress, and putting in the hours

Severity: High

Resolution: daily check-ins & properly planning work divisions amongst sub-groups

Status: In progress

Risk: Division of work

Description: Dividing up work between team members

Severity: low

Resolution: Pair programming

Status: In progress

Risk: (Relatively) Large Group Environment Unfamiliarity

Description: None of us are very experienced with large groups, thus we may struggle to coordinate our efforts, delegate tasks, and communicate effectively.

Severity: Medium

Resolution: Mutual patience and understanding and willingness to accept input from every group member.

Status: In progress

Risk: Open-Ended Nature of Project

Description: The project does not have a clear "solution" to a problem; we'll have to make decisions on design implementation, team meeting logistics, etc.

Severity: High

Resolution: Q/A customer meeting, and customer emails (piazza posts)

Status: Resolved

2. User Stories

1) User Story: Searching For Animals - (Priority: HIGH)

As a: Zoo Visitor

I want: to search for animals within the zoo (app)

so that: I can find animals I want to see

Scenario 1: Nothing typed in search bar

Given: I open up the app

And: I click on the search bar

When: I don't type anything in the search bar

Then: No results should appear below

Scenario 2: Multiple species of bears

Given: There are multiple types of bears at the zoo (grizzly bear, black bear, brown bear, panda bear)

When: I type in "bear" in the search bar

Then: A list of bears species with the word "bear" appears, listing the types of bears at the zoo (grizzly bear, black bear, brown bear, panda bear)

And: I have the option of continuing to type in the search bar to narrow down my search

Scenario 3: The animal does not exist at the zoo

Given: Blue whales don't exist at the zoo

When: I type in "Blue Whale"

Then: Nothing should appear in the search results

2) User Story: Selecting Animals - (Priority: HIGH)

(Dependent on: 1)

As a: Zoo Visitor

I want: to select animals I searched for

So that: I have a list of the animals I want to see

Scenario 1: Selecting "Grizzly Bear"

Given: I searched for Grizzly Bear

And: I see "Grizzly Bear" in the drop-down list

When: I select "Grizzly Bear"

Then: "Grizzly Bear" should appear in the list below the search bar

And: the number on top of the list should increment by one

Then: The search query should end

And: The search bar should be cleared

Scenario 2: Selecting nothing

Given: I have typed "bear" into the search bar

When: I don't make a selection

And: Don't click on any animal
Then: nothing should happen
And: the current options should persist

Scenario 3: Trying to add animals already selected and in the list

Given: I have added "Grizzly Bear" in my list
And: I search for "Grizzly Bear" in the search bar
When: I select "Grizzly Bear" again
And: I click the "plan" button
Then: "Grizzly Bear" should not appear in my plan of animals

3) User Story: Search Improvements - (Priority: Low) (Dependent on: 2)

As a: Zoo Visitor

I want: to see the app offer suggestions of the animal names and search using the microphone
So that: I can speed up my queries and spend more time seeing the animals

Scenario 1: "Grizzly Bear" auto-completion

Given: The search bar is selected
And: Grizzly Bear isn't in my list
And: Grizzly Bears exist at the zoo
When: I type "Griz"
Then: I see "Grizzly Bear" in the drop-down results list
And: I can select it to add Grizzly Bear to my list

Scenario 2: No available auto-completion

Given: The search bar is selected
When: I type nonsense that could never be a substring/prefix of any zoo animal ever
Then: No auto-complete suggestions pop up
And: The drop-down list search results contain nothing
And: Instead a message is displayed "No such zoo animals were found under this search..."

Scenario 3: Using microphone and saying nothing

Given: I choose the microphone on the keyboard
When: I don't say anything
Then: Nothing will be typed into the search bar
And: No animals should pop up

Scenario 4: Using microphone and saying "bears"

Given: Microphone button is selected
When: I say "bears" into the mic
Then: a list of a variety of bears that are at the zoo pops up
And: I can select which ones I want to plan

4) User Story: Pressing Plan - (Priority: Medium)

(Dependent on: 3)

As a: Zoo Visitor

I want: plan a route from my selected animals

So that: I can visit the animals I want to see in-order of shortest distance from the starting location (entrance of zoo) to exhibit.

Scenario 1: Planning with no animals selected

Given: I have not selected any animals

When: I hit the plan button

Then: I am taken to a search bar where I can begin searching for zoo animals

Scenario 2: Planning my route

Given: I have selected all the exhibits I wish to see

When: I hit the "plan" button

Then: A screen should appear containing a plan of all the exhibits labeled by the animal's name, starting at the front gate, ordered by distance along the route

And: I should see the street/trail name and distance below the animal name

And: The listings in the plan should be sorted by distance, with each distance being a little larger than the next

Scenario 3: Planning route to see "Baboons" and "Hippos"

Given: I have selected "Baboons" and "Hippos" as exhibits I want to see

And: Baboons are 200 ft away from the entrance

And: Hippos are 300 ft away from the entrance

When: I press the "plan" button

Then: I should be shown a listing of "Baboons" followed by "Hippos", each with the street/trail name of their exhibits and the distances

5) User Story: Pressing Directions - (Priority: Medium)

(Dependent on: 4)

As a: Zoo Visitor

I want: to get directions to the first exhibit of my planned route

So that: I can begin my tour (I can get to the first exhibit)

Scenario 1: I have at least one exhibit planned

Given: That Baboons are the closest exhibit from the list of exhibits planned

When: I tap Directions

Then: There should be directions displayed on the screen to the Baboons

And: Should say "Proceed from Front Street down Treetops Way, 200 ft on your left"

And: I should not be able to return to the "Plan" screen

6) User Story: Pressing Next - (Priority: Low)

(Dependent on: 5)

As a: Zoo Visitor

I want: to see the street/trail name that I need to take to get to the next exhibit in my plan, as well as the distance from the current exhibit location to the next exhibit.

So that: I can see how to get to the next animal in my plan without being overwhelmed by a large, hard to read map.

Scenario 1: Proceeding from the Grizzlies

Given: I just finished visiting the "Grizzly Bear" exhibit

And: the next exhibit is on the planned route

When: I hit the "Next" button

Then: I should see the directions to the next exhibit along my planned route

Scenario 2: On last exhibit

Given: I have just visited my last exhibit within the planned route

And: I'm finished with my tour

When: I hit the Next button

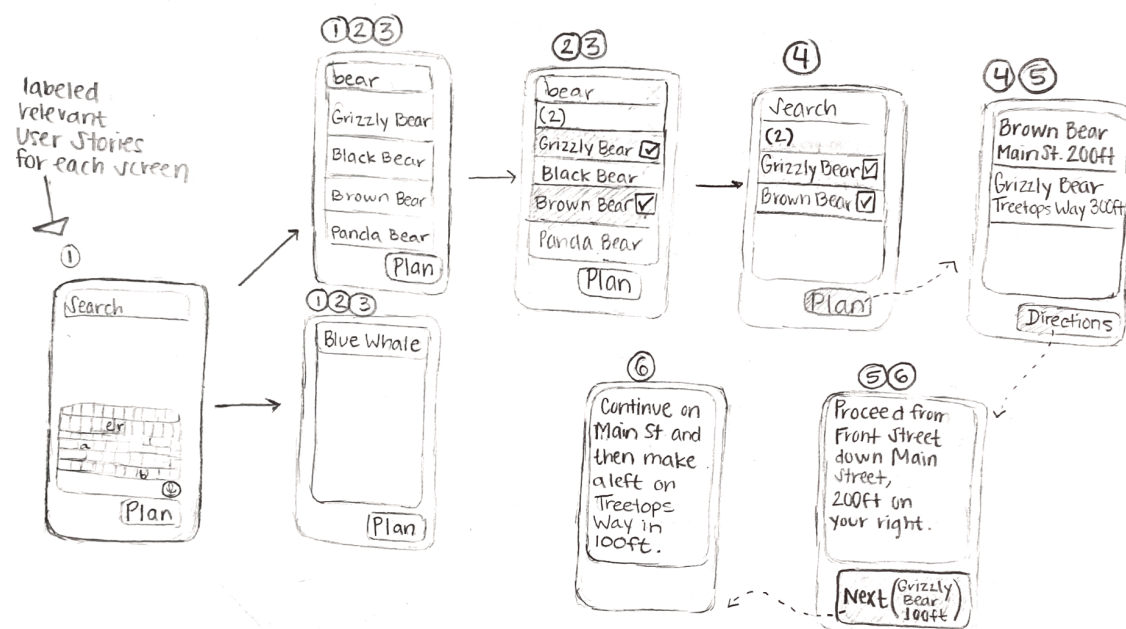
Then: The "next" button should be grayed

And: There should be no directions left to new exhibits

And: The live route should end

And: The screen will remain

Wired-Frame UI



3. Planning Poker

User Story	Voting Round	Hand						False Assumptions Uncovered
Searching Animals	1	5	5	8	2	3	5	Building the UI from scratch probably is a task under this substory, in addition to backend structure
	2	7	7	7	8	7	5	Getting familiar with Android environment
	3	10	10	9	10	10	10	
Selecting Animals	1	3	2	5	3	3	2	Testing the Lists might take a long time
	2	4	6	5	4	4	4	Time required to design UI for list
	3	5	5	5	5	5	5	
Search Improvements	1	5	8	4	4	4	3	Is autocomplete/microphone function a feature/library that already exists in Android
	2	4	6	4	5	3	4	Learning to use Android libraries and integrating it into our app
	3	5	5	5	5	5	5	Testing these functions might be time consuming
	4	8	7	8	8	8	8	
Plan	1	8	4	8	6	8	4	Mostly using UI from previous stories (ie. "selecting animals") UI, backend function Testing for ordering might be complicated Distance is pre-calculated from the entrance of zoo and ordered along the route Learning how to implement algorithm might take time
	2	6	8	6	6	8	6	Distances are not all from entrance; each exhibit shows distance ALONG route, which means route is planned beforehand
	3	8	8	8	8	8	8	
Pressing	1	3	4	4	5	4	5	How are we getting directions? What's

Directions								the format of the directions? Especially user-centered directions like "Go left"
	2	4	5	5	6	5	5	Implementing these directions might take some time
	3	5	5	5	5	5	5	
Next Button	1	5	5	6	8	2	2	Are we just updating the list? Are directions calculated here or in Plan?
	2	3	6	5	4	3	3	Does the current position need to be tracked? No, the system assumes the user is at the previous exhibit.
	3	4	4	4	4	4	4	

The screenshot shows a Discord server named "CS110 Project" with a live video call in progress. The video call participants are Don, Lucas, Brian, kitkat, alar, and Sharkmeat. The main window displays a Jira board for "CSE 110 Team 19". The board has a backlog of issues, including "Scenario Based System Test", "Participation Survey", and "Android Studio Expert". The "Participation Survey" issue is selected, showing its details, including a description, labels, and dependencies. The issue is assigned to "LucasKasman" and has a due date of "2 days ago".

Course Websites

110 Team 19

Download the ZenHub app

Backlog

CS10-110 Team 19 Board

CS10-110 Spring 2022/100seker-cse-110-team-19F15 - Pressing Directions

LucasKasman opened this issue 2 days ago · 2 comments

Open Edit Close issue

Pressing Directions #15

Add a description

Pressing Directions has no dependencies

Add dependency

Pressing Directions is an epic

Set start and end date

0 of 0 issues completed

0 of 5 Epic points completed

No issues yet

Create new issues here or Add issues to this Epic

Estimate Assignees

LucasKasman converted this to an epic 2 days ago

LucasKasman added label 2 days ago

#1526 commented 33 minutes ago

Hey team! Please add your planning poker estimate with ZenHub @#1526 @brianu9 @brianu1 @karima-kumar @LucasKasman

#152023 commented 33 minutes ago

Please add your planning poker estimate with ZenHub @#152023

LucasKasman set the estimate to 5 17 minutes ago

Pipelines

CS10-110 Team 19

User Stories

Labels

Click the pin icon to assign high priority status. Once pinned, a high priority badge is applied in the issue and it's placed at the top of the pipeline in the board.

Assignees

No one - assign yourself

Sprints

No sprint assigned

Final Estimate

5 epic points

Planning poker

Epics

Not inside an epic

Releases

Not inside a release

View this issue on GitHub

Future scenario, but seems like a cool feature)

CS10 Project

TEST CHANNELS

general

resources

post-meeting-notes

VOICE CHANNELS

General

Backlog

App WS

Download the ZenHub app

No issues yet

Create new issues here or Add issues to this Epic

Estimate Assignees Labels

Assigned No one - assign yourself

Final Estimate

5 epic points

Types

Not inside an epic

Releases

Not inside a release

View this issue on GitHub

LucasKasman converted this to an epic 2 days ago

LucasKasman added label 2 days ago

#1526 commented 33 minutes ago

Hey team! Please add your planning poker estimate with ZenHub @#1526 @brianu9 @brianu1 @karima-kumar @LucasKasman

#152023 commented 33 minutes ago

Please add your planning poker estimate with ZenHub @#152023

LucasKasman set the estimate to 5 17 minutes ago

League of Legends

Video Connected

Screen

Mr Mad Cow

Online Now · This Week

4. Tasks

Iteration 1: Getting Going

1) User Story: Searching For Animals - 10 hours - HIGH Priority

Project Creation: Learn how to do Android (make app launch, UI for homepage) 3

Animal Class: Make a class for animals 1

Animal Lists: Make and populate a data structure (list?) to hold the list of animals by name and class/common name (maybe hashtable/hashmap?) 2

Search Bar: Create user interface for the search bar 1

Search Method(s): Implement Search function with queries to the data structure (add to list class, get string from search bar and give it to list) 1

Search Results: Create user interface for displaying the search results of the animals 2

2) User Story: Selecting Animals - 5 hours - HIGH Priority

Select button: Implement a select button on the search UI (toggle?) to, use string to get an animal object from the list of animals in the zoo and store the selected animal. 2.5

Storing Selected Animals: Implement a data structure class to store the selected animals, which receives an animal object and stores it. - 0.5

Selected Animal Display: Display the list of selected animals on the user interface as they are selected - 1.5

Number of Animals Selected: Display the number of selected animals 0.5

Iteration 2: Going Further

3) User Story: Pressing Plan - 8 hours - MEDIUM Priority

Plan Button: Implement the "Plan" button response 1

Plan Entries/Page: Create the UI for the "Plan" page 3

Create Plan: Sort and display exhibits in order of distance 4

4) User Story: Pressing Directions - 5 hours - MEDIUM Priority

Directions Button: Implement "Directions" button within UI 1

First Exhibit: Get the first exhibit to visit and the corresponding directions 2

First Exhibit UI: Implement the UI for the first exhibit (same as task 2 of user story 6 but for the first case) 3

5) User Story: Pressing Next - 4 hours - LOW Priority

Next Button UI: Implement "Next" button within UI 1

Next Exhibit Page: Implement UI for each exhibit (Cards?) 3

BACKLOG:

6) User Story: Search Improvements - 9 hours - LOW Priority

Android Auto-Complete : Learning the Android auto-complete functionality 3

Auto-Complete: Implement the auto-complete function 2

Android Microphone Input : Learning the Android voice input functionality 2

Microphone Input : Implement the microphone input function 2

5. Scenario-Based System Tests

End of Iteration 1 (User stories: Searching For Animals, Selecting Animals)

1. Juan launches the app by tapping the app icon from their mobile device. He sees a page with a search bar at the top and a grayed out Plan button.
2. He clicks the search bar and begins typing his query "bear"
3. A list begins developing below the search bar, with animals that have the word "bear" in its name.
4. He scrolls through the list and clicks on "black bear". He sees "black bear" appear in the list at the bottom of the screen. He also sees a number badge labeled "1" at the top of the list. He has just added one animal!
5. The search has ended and Juan taps the search bar to begin a new query. Since "black" is still in the search bar, Juan clicks the X on the search bar to clear it. This time he begins a search for Grizzly bears. He types the full name "grizzly bear".
6. He sees the singular entry pop up and clicks on "grizzly bear". He sees "grizzly bear" appear in the list at the bottom of the screen. He also sees a number badge labeled "2" at the top of the list. He has just added another animal!
7. Juan suddenly forgets if he had added grizzly bear to the list and clicks on the grizzly bear entry again. He sees the list stay the same and the number badge not change and still be "2".
8. Juan is super satisfied with our product and closes the app!

End of Milestone 1 / End of Iteration 2 (User stories: Searching For Animals, Selecting Animals, Pressing Plan, Pressing Directions, Pressing Next)

1. Juan launches the app from their mobile device. He sees a page with a search bar at the top.
2. He wants to look-up black bears, so he clicks the search bar and begins typing "black", but he decides that typing is too much of a hassle, so he uses android's mic function, he says "bear" into the mic. A result for "black bear" appears within a drop-down list under the search bar.
3. He selects the "black bear" and it gets added to a new list.
4. Next Juan wants to see some birds, so he clicks back onto the search bar, erases his previous query and begins typing "bi". Oh! An auto-complete query "birds" pops up for Juan. So he selects it. Now he sees a list of birds.
5. He scrolls through the list and selects "rainbow finch". Now he's done and wants to begin his adventure. He sees the "Plan" button at the bottom.
6. He clicks the "Plan" button and a list of his selected animals appear in cards, complete with their current distance from the zoo entrance and exhibit names. He now sees a directions button at the bottom

7. Juan begins his journey by pressing the "Directions" button and it directs him to his first exhibit, the "black bear", with directions "Go left of copper trail" and "Follow along Emerald lane for 300 ft". He reaches the exhibit.
8. Juan soon grows bored and notices the "Next" button at the bottom. He presses it and a new set of directions appear, 100 ft towards your left from the current exhibit.
9. Juan finishes his experience and is satisfied.

6. Iterations/Milestone

Work Hours Available per iteration:

$5\text{hr/person} * 6\text{ people} * 0.5\text{ velocity} = 15\text{ work hours available per iteration}$

Iteration Length:

- 1st iteration : 15 hours
- 2nd iteration : 17 hours

Iteration 1

1. Searching For Animals (US 1)
2. Selecting Animals (US 2)

Iteration 2

3. Pressing Plan (US 4)
4. Pressing Next (US 6)
5. Pressing Directions (US 5)

Backlog

6. Search Improvements (US 3)

Backlog justification:

We have User Story: Search Improvements in the Backlog because we believe with our current estimated velocity of 0.5, and the available people hours we have in the next 2 weeks before the first milestone (60 hours), we will not be able to finish this User Story during the first milestone before the higher priority user priorities.

Developer Story (Iteration 1): Bug free Integration

As a Developer, I want to be able to work in parallel with my partners on integrated systems without causing bugs.

Task 1: Parameters and Return Types:

Specify what data values our code is going to take as parameters and return before we write it, so that others in our team can work on other sections before we are done without causing bugs.

Task 2: Communicate unexpected issues

If one of us discovers that something we agreed upon is not going to work, they will immediately notify the rest of the team so that the team can adapt before too much work is wasted.

Task 3: Good Mock Design

Before a class is finished being written, we will make a mock of it so that the other team members can begin testing their code that relies on the class.

Developer Story (Iteration 2): Bug free Integration

As a Developer, I want to be able to work in parallel with my partners on integrated systems without causing bugs.

Task 1: Parameters and Return Types:

Specify what data values our code is going to take as parameters and return before we write it, so that others in our team can work on other sections before we are done without causing bugs.

Task 2: Communicate unexpected issues

If one of us discovers that something we agreed upon is not going to work, they will immediately notify the rest of the team so that the team can adapt before too much work is wasted.

Task 3: Good Mock Design

Before a class is finished being written, we will make a mock of it so that the other team members can begin testing their code that relies on the class.

Task 4: Good Algorithm Design for Plan

We will use a graph to represent the map in order to make the process of making the route simple.

Task 5: Coordinates

We will brainstorm a good way to determine which orientation (left, right) the exhibits will be along the path

7. Zenhub

Link to Team 19's Zenhub:

<https://app.zenhub.com/workspaces/cse-110-team-19-625e02ba59c8c4001894ec22/board>