Risk Analysis

Risk #1: Scheduling Conflicts

Description: We all have different schedules that don't line up perfectly, so it's hard to find time that all 6 of us are available.

Severity: High

Resolution: We have set aside a time that works for everyone to do weekly meetings: Fridays 3-4pm. We will also have quick stand-ups outside the lecture hall before each class at 12:50pm.

Status: Resolved

Risk #2: Android Speech-to-text

Description: We haven't worked with the Android speech-to-text features before, so we aren't sure how difficult they will be to implement.

Severity: Medium

Resolution: We will add a task for looking into the Android speech-to-text API and getting a minimal proof-of-concept feature. We will set the time estimate on this task on the high end since we aren't sure exactly how long it will take.

Status: In progress

Risk #3: Unfamiliarity With App Architecture

Description: Implementing an Android app with the architecture from lab 3 is something that's new to all of us.

Severity: Medium

Resolution: Like with risk #2, we will add a task for familiarizing ourselves with Android and architecture to factor in the time we will take to learn the necessary concepts.

Status: In progress

Risk #4: Android platform issues

Description: Most of us are pretty new to Android, so there are aspects of the platform that could be difficult to figure out. There are also likely to be weird bugs with certain aspects of using Android, releasing the app, etc.

Severity: Medium

Resolution: Budget in extra time to resolve any issues that arise, both in individual tasks/stories and in the iteration as a whole.

Status: In progress

Estimated initial velocity: 0.6

Justification: We learned in class that 0.7 is a good starting velocity, but based on 1) we are likely to have issues being available at the same time to work on the project, and 2) there may be tasks that take extra time or are more difficult than we thought, we are choosing 0.6 instead of 0.7.

Iteration Length: 1 week

Planning Poker

S#	Name	Hand	False Assumptions Uncovered
1	Enter and view goals	8 8 8 4 4 16	16: Lots of functions to write, testing could take a long time. Would have to setup project and architecture 8: Forgot about setting up unfamiliar architecture (layers), not too difficult to build UI and write a Goals class 4: Not taking into account testing time
1	Enter and view goals	8 8 8 8 8 8	(None)
2	Cross off/uncross goals	4 4 4 4 2 2	4: Feels pretty easy, but there could be tasks we're not thinking of. Also need to research strikethrough on Android 2: Setup already there, strikethrough is a small task
2	Cross off/uncross goals	4 4 4 4 4 4	(None)
3	Speech-to-text	16 8 8 2 8 8	16: One of the hardest stories because it's so unfamiliar, and we haven't worked at all with this before 8: Using Android API would be relatively straightforward. Speech-to-text is only a single feature. 2: There should be examples online we can use
3	Speech-to-text	8 8 8 8 8 8	(None)
4	View current date	2 4 2 4 2 4	4: Unfamiliarity with coding with tracking current date, unsure how difficult it was & wanted to add a cushion 2: There is an existing library for this, similar to datetime. Java method to get current time
4	View current	2 2 2 2 2 2	(None)

	date		
5	Hide crossed-off goals from previous days	4 2 8 8 4 2	8: Being conservative, task list seemed to have lots of items for this story 4: A lot of tasks, but a lot of them are really small 2: Seemed straightforward. Forgot about implementation for date that task was crossed off
5	Hide crossed-off goals from previous days	4 4 4 4 4 4	(None)
6	Show today's crossed-off goals at bottom of goals list	2 4 2 8 4 4	2: Sounds much simpler than other tasks, straightforward testing. Built in functions that can help 4: Didn't consider built-in methods for sorting 8: Changed mind based on discussion
6	Show today's crossed-off goals at bottom of goals list	2 2 2 4 2 2	(None)
7	Show goals in correct order	4 2 4 4 4 4	2: since we're already figuring out how to pull time from other user stories, this will be quite simple. 4: simple, but more complicated than a story like View Current Date.
8	Persist Goals	8 8 16 8 8 16	8: Android provides a way to do this and it's not something we need to do from scratch 16: Unfamiliarity, need to figure out the logic for when we're loading/storing data, edge cases
8	Persist Goals	16 16 16 16 16 8	(None)

User Stories

User Story #1: Enter and view goals

Description: As a user, I want to be able to add goals to a list and view the list of my goals so that I can track what I need to do for the day.

Estimate: 11.5 hours

Priority: High

BDD Scenarios:

BDD Scenario 1: No tasks
Given that I have not entered any tasks,
When I launch the app,
Then I should not see any tasks
And I should see text saying “No goals for the Day. Click the + at the upper right to enter your Most Important Thing."

BDD Scenario 2: One task
Given that I have not entered any tasks,
When I launch the app,
Then I should see a plus button
When I press that button,
Then I should see a text input field, checkmark button, and cancel button,
When I type “Example goal” into that field, and press the checkmark button,
Then I should see the list of goals with “Example goal” being the only goal in the list

BDD Scenario 3: Cancel entering a task
Given that I not entered any tasks,
When I launch the app,
Then I should see a plus button
When I press that button,
Then I should see a text input field, checkmark button, and cancel button,
When I press the cancel button,
Then I should see the list of goals,
And the list of goals should be empty and still show “No goals for the Day. Click the + at the upper right to enter your Most Important Thing.”

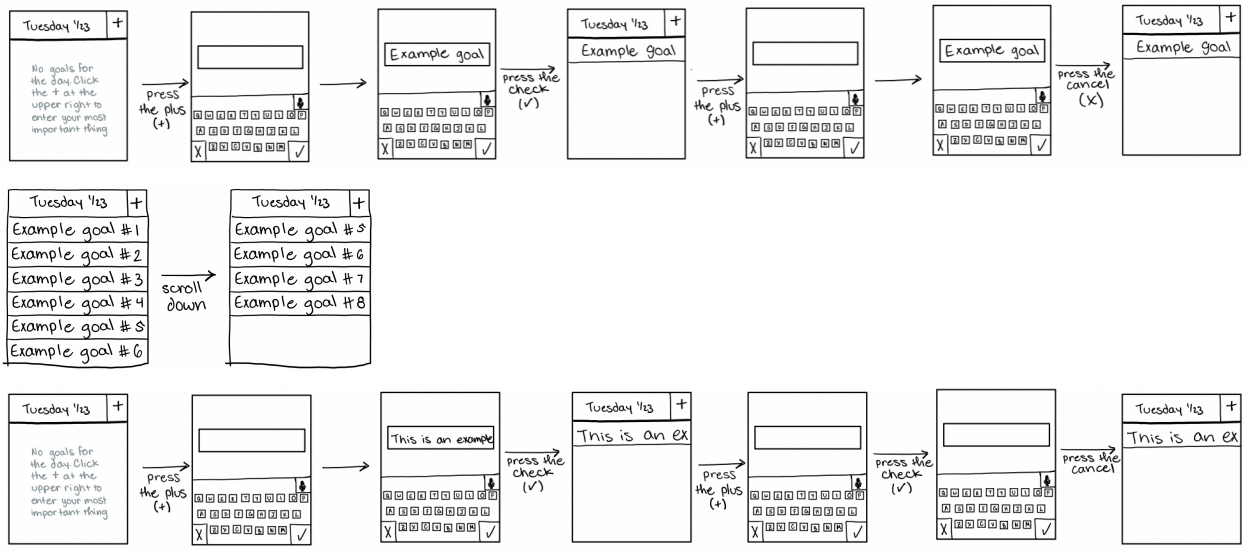
Scenario 4: Type and then cancel entering a task
Given that I not entered any tasks,
When I launch the app,
Then I should see a plus button
When I press that button,
Then I should see a text input field, checkmark button, and cancel button,
When I type “Goal 1” into the text input field,
And I press the cancel button,
Then I should see the list of goals,
And the list of goals should be empty and still show “No goals for the Day. Click the + at the upper right to enter your Most Important Thing.”

BDD Scenario 5: Tap check button without typing anything
Given that I not entered any tasks,
When I launch the app,
Then I should see a plus button
When I press that button,
Then I should see a text input field, checkmark button, and cancel button,
When I press the check button,
Then I should still see the text input field, checkmark button, and cancel button,
When I press the cancel button,
Then I should see the list of goals,
And the list of goals should be empty and still show “No goals for the Day. Click the + at the upper right to enter your Most Important Thing.”

BDD Scenario 6: More tasks that can fit on page
Given that I have more tasks that can fit on one page
When I launch the app
Then I should see the first few tasks
And when I scroll down
Then I should see the rest of my tasks

BDD Scenario 7: Long task
Given that I have a long task that can’t fit on the screen in one line
When I launch the app
Then I should see the task, truncated so it can fit on one line

Wire-framed UI Screen



User Story #2: Cross off/uncross goals

Description: As a user, I want to be able to cross off a goal so that I can track which goals I still need to complete.

Estimate: 5 hours

Priority: High

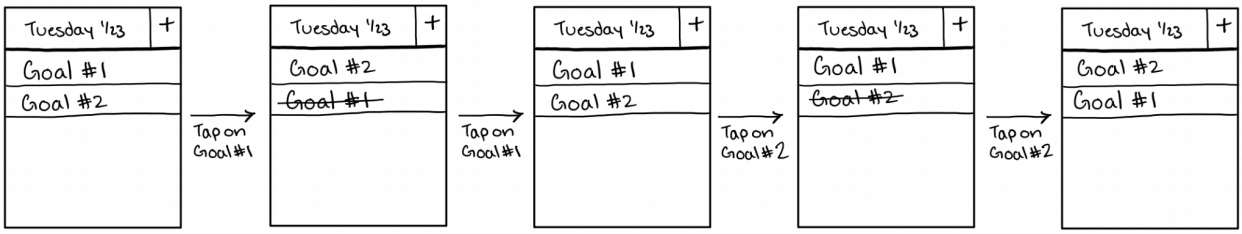
BDD Scenarios:

BDD Scenario 1: Cross off and uncross a task
Given that I have entered two goals: “Goal 1” and “Goal 2”
And that I have not yet crossed off either goal,
When I tap on “Goal 1”,
Then “Goal 1” should be crossed off
And “Goal 2” should not be crossed off
When I tap on “Goal 1” again
Then “Goal 1” and “Goal 2” should not be crossed off

BDD Scenario 2: Cross off multiple tasks
Given that I already have a goal (Goal 1) crossed off
And that I still have other goals (Goal 2 and Goal 3) that’s not already crossed off
When I click on Goal 2
Then Goal 2 should be crossed off

And Goal 1 should be crossed off
And Goal 3 should not be crossed off

Wire-framed UI Screen



User Story #3: Speech-to-text

Description: As a user, I want to be able to add new goals by speaking instead of typing so that users can save time and effort on a busy day.

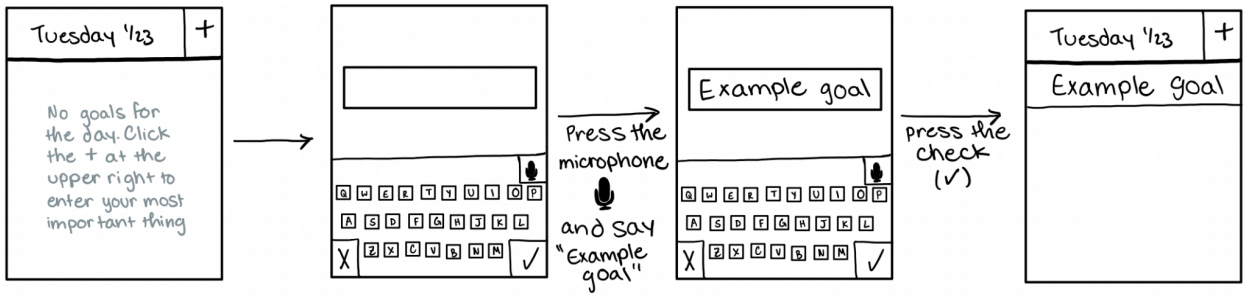
Estimate: 11.5 hours

Priority: Low

BDD Scenarios:

BDD Scenario 1: Enter a task using speech-to-text
Given that I haven't yet entered any tasks,
And that I am on the screen to enter a task,
Then I should see a microphone button,
When I press the microphone button,
And say "Example Goal" aloud,
And press the checkmark button,
Then I should be taken to the goals list screen,
And there should be one goal in the list: "Example Goal"

Wire-framed UI Screen



User Story #4: View current date

Description: As a user, I want to be able to see the current date so that I know what day my goals are for.

Estimate: 2.5 hours

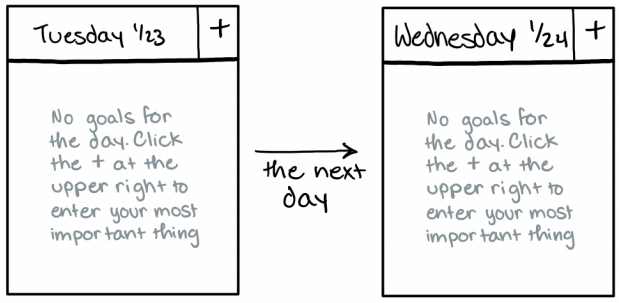
Priority: Low

BDD Scenarios:

BDD Scenario 1: View today
Given that I am using the app on a certain day
When I open the app
Then I should see today's date in the format DD MM/YY at the top

BDD Scenario 2: View another day
Given that I use the app on the day after the day in Scenario 1
When I open the app after 2:00am
Then I should see the current day (i.e. the day after the day in Scenario 1) displayed in the app

Wire-framed UI Screen



User Story #5: Hide crossed-off goals from previous days

Description: As a user, I want tasks that I crossed off on previous days to be hidden so that I can focus on today's tasks and not get distracted by past tasks.

Estimate: 4 hours

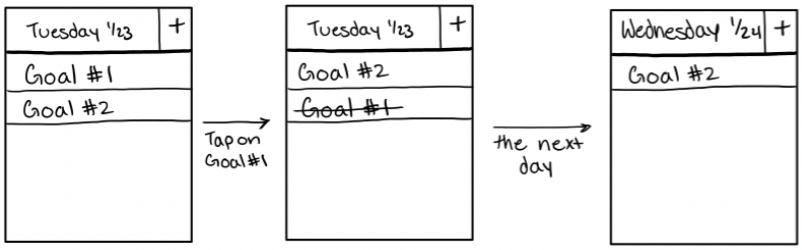
Priority: Medium

BDD Scenarios:

BDD Scenario 1: Cross off a goal and wait a day

Given that I see two goals in the goals list, "Goal 1" and "Goal 2",
When I cross off "Goal 1",
Then I should still see "Goal 1" and "Goal 2" in the list,
When I open the app again the next day after 2am,
Then I should see "Goal 2" in the list,
And I should not see "Goal 1" in the list

Wire-framed UI Screen



User Story #6: Show today's crossed-off goals at bottom of goals list

Description: As a user, I want goals that I crossed off today to show up at the bottom of my goals list, so that I can focus on tasks I haven't completed yet

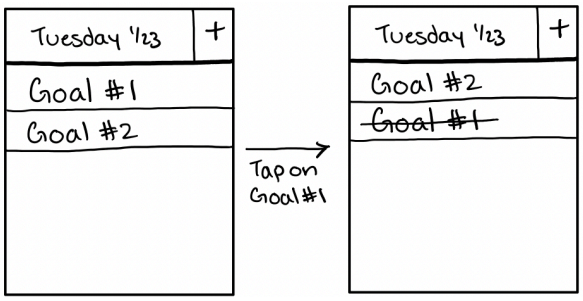
Estimate: 2.5 hours

Priority: Medium

BDD Scenarios:

BDD Scenario 1: Cross off a goal
Given that I see two goals in the goals list, “Goal 1” and “Goal 2”,
When I cross off “Goal 1”,
Then “Goal 1” should move to the bottom of the goals list, below “Goal 2”

Wire-framed UI Screen



User Story #7: Show goals in correct order

Description: As a user, I want goals to be displayed in the following order:

1. Active goals that have been crossed and then uncrossed, from most recently uncrossed to least recently uncrossed
2. Active goals that have NOT been crossed and then uncrossed, from oldest to newest entry date
3. Crossed-off goals, from most recently crossed-off to least recently crossed-off.

So that they are easy to locate and organize.

Estimate: 7 hours

Priority: Low

BDD Scenarios:

BDD Scenario 1: Multiple goals entered
Given that I have not yet entered any goals,
When I enter a goal named “Goal 1”,
Then I should see “Goal 1” in the goals list,
When I enter another goal named “Goal 2”,
Then I should see “Goal 1” in the list, and “Goal 2” in the list BELOW “Goal 1”

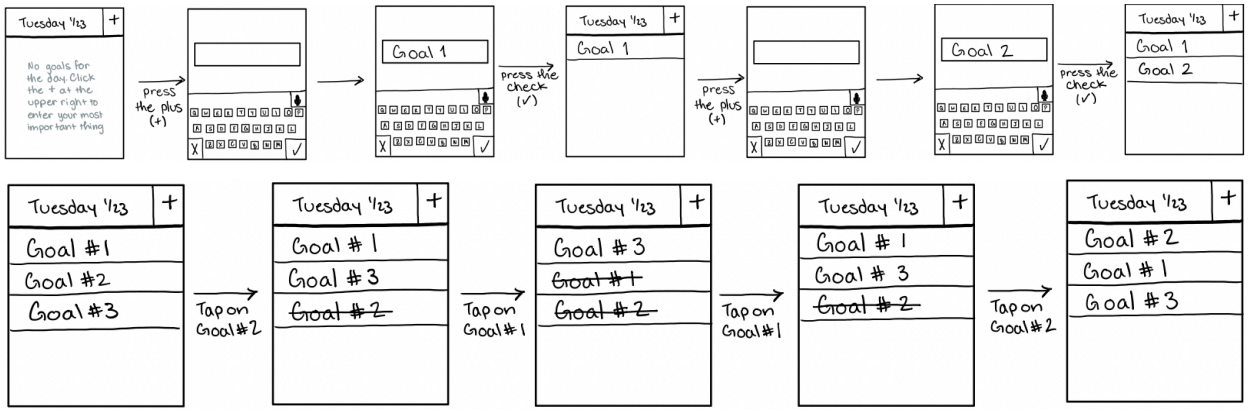
BDD Scenario 2: Multiple goals entered, crossed, and uncrossed
Given that I have entered “Goal 1”, then “Goal 2”, then “Goal 3”,

When I cross off “Goal 2”,
Then I should see “Goal 1”, then “Goal 3”, then “Goal “2”,
When I cross off “Goal 1”,
Then I should see “Goal 3”, then “Goal 1”, then “Goal 2”,
When I uncross “Goal 1”,
Then I should see “Goal 1”, then “Goal 3”, then “Goal 2”,
When I uncross “Goal 2”,
“Then I should see “Goal 2”, then “Goal 1”, then “Goal 3”.

BDD Scenario 3: Multiple goals entered and crossed
Given that I have 3 goals entered as ordered: “Goal 1”, ‘Goal 2”, “Goal 3”,
When I tap “Goal 1”,
Then I should see “Goal 1” crossed off at the bottom of the list,
When I tap “Goal 2”,
Then I should see “Goal 2” crossed off below “Goal 3” and above “Goal 1”

BDD Scenario 4: A never-crossed goal, a crossed/uncrossed goal, and a crossed-off goal
Given that I have entered 3 goals, named “Goal 1,” “Goal 2,” and “Goal 3” (entered in that order),
When I cross off Goal 1,
Then the ordering should be Goal 2, Goal 3, Goal 1,
When I cross off Goal 3,
Then the ordering should be Goal 2, Goal 3, Goal 1,
When I uncross Goal 1,
Then the ordering should be Goal 1, Goal 2, Goal 3

Wire-framed UI Screen



User Story #8: Persist goals

Description: As a user, I want my goals to be saved and still show up after I relaunch the app, so that I can track my progress over extended periods of time.

Estimate: 17.5 hours

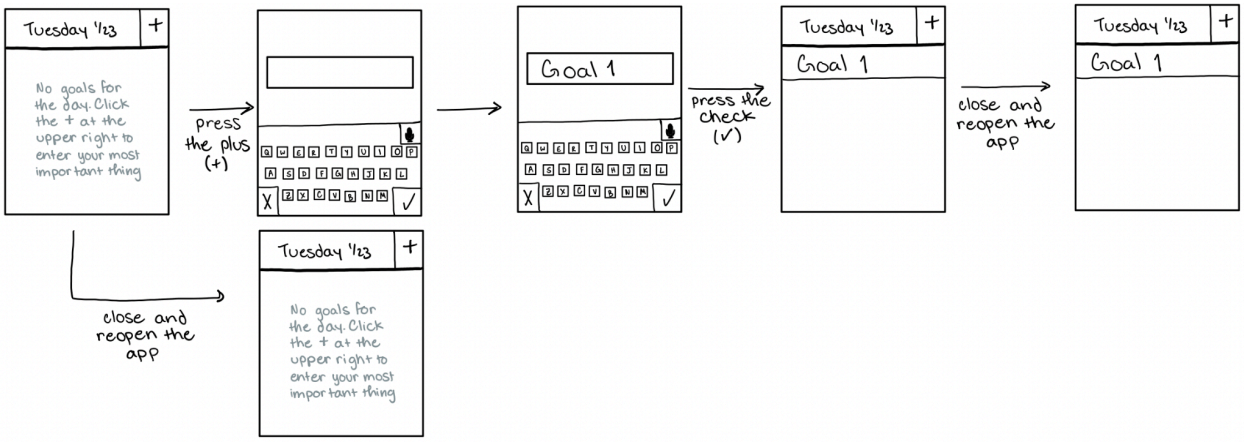
Priority: High

BDD Scenarios:

BDD Scenario 1: Multiple Goals
Given that I have two goals in my goal list, “Goal 1” followed by “Goal 2”,
When I close out of the app and reopen it,
Then I should still see “Goal 1” followed by “Goal 2” in my goals list

BDD Scenario 2: No Goals
Given that I have no goals in my goal list,
When I close out of the app and reopen it,
Then I should see no goals in my goal list

Wire-framed UI Screen



Tasks

Story #1: Enter and view goals

- UI to render list of goals (est. 2 hours)
- Plus button to enter a goal (est. 1 hour)
- UI to enter a goal (i.e. textbox) (est. 2 hours)
- UI to cancel entering goal (1 hour)
- Store list of goals (0.5 hour)
- Add a goal to list when user enters a goal (0.5 hours)
- Ensure the list of goals is scrollable (use ScrollView) (1.5 hours)
- Testing of above tasks (3 hours)

Story #2: Cross off/uncross goals

- Add attribute to goal representation to store whether it is crossed off (0.5 hours)
- Add code to mark a goal as crossed off (0.5 hours)
- Add click listener for when a user taps a task (1 hour)
- Modify UI for task list to display a task as strikethrough if it is crossed off (1 hour)
- Testing of above tasks (2 hours)

Story #3: Speech-to-text

- Determine minimum Android version for system keyboard Speech-to-text and if system keyboard is a viable way for our app to do speech-to-text (0.5 hour)
- Test whether system keyboard speech-to-text works as required for our app (0.5 hour)
- Look into whether requesting microphone permissions manually in our app is required (0.5 hour)
- Implement microphone permissions (if required) (2 hours)
- Research Android Speech-to-text API (1 hour)
- Get a proof-of-concept Speech-to-text feature implemented (have the user say something, get the text for what they said in code) (2 hours)
- Get text for what user said (1 hour)
- Put speech-to-text result in text box to enter a goal (0.5 hours)
- Display a microphone button on the “add a text” (0.5 hours)
- Run speech-to-text when the user taps the microphone button (1 hour)
- Testing of above tasks (2 hours)

Story #4: View current date

- Write code to get current date, with the caveat that if it’s before 2 AM on Feb 3, then it should show Feb 2 (0.5 hour)
- Write code to format a date in specified format (day of week, month, day) (0.5 hour)
- UI to display text for date (0.5 hour)
- Combine above tasks to display current date on UI (0.5 hour)
- Testing of above tasks (0.5 hour)

Story #5: Hide crossed-off goals from previous days

- Add an attribute to goal representation for the day it was crossed off, considering any time before 2 AM on Feb 3 to actually be Feb 2 (1 hour)
- Write code to check whether a goal was 1) crossed off and 2) entered on a previous day (1 hour)
- Write code to hide goals matching criteria above (0.5 hour)
- Testing of above tasks (1.5 hours)

Story #6: Show today's crossed-off goals at bottom of goals list

- Write code to sort list of tasks by whether they're crossed off (crossed off tasks at end) (0.5 hour)
- Apply this code to our rendering of goals list, in order to put crossed-off goals at the bottom (1 hour)
- Testing of above tasks (1 hour)

Story #7: Show goals in correct order

- Add attributes to goal representation for the time it was entered, time it was crossed, and time it was uncrossed (2 hours)
- Write code to sort list of tasks in order described in Story #7 (using above attribute) (2 hours)
- Apply this code when rendering list of goals (0.5 hour)
- Testing of above tasks (2.5 hours)

Story #8: Persist goals

- Research persisting data in storage in Android (2 hours)
- Get minimal proof-of-concept for data persistence (3 hours)
- Design persistent/serialized representation of tasks (2.5 hours)
- Write code to store tasks in storage (2 hours)
- Write code to load tasks from storage (read from file/DB) (2 hours)
- Integrate load/store in our code (1 hour)
- Handle edge cases (e.g. no files yet) (2 hours)
- Testing of above tasks (3 hours)

Iterations/Milestone

Initial velocity = 0.6. 6 team members. Each member will spend about 10 hours a week (each iteration is 1 week). This works out to 6 * 10 = 60 hours of dev work per week, times our velocity is 0.6 * 60 = 36 productive hours per iteration. So we can fit up to 36 hours of User Story work in each iteration.

Iteration 1

We are putting User Stories **1 (Enter and view goals), 2 (Cross off/uncross goals), and 8 (Persist goals)** in Iteration 1 because they are the high-priority User Stories. Story **1** should be done at the beginning. Their total time estimate is 11.5(story 1) + 5(story 2) + 17.5(story 8) = **34 hours**, which is just under our 36 hours to allocate per iteration.

Iteration 2

We are putting User Stories **5 (Hide crossed-off goals), 6 (Show today's crossed-off goals at bottom of goals list), 3 (Speech-to-text), 4 (View current date), and 7 (Show goals in correct order)**. We first chose stories 5 & 6 because they have medium priority, and we already put all the high priority stories in Iteration 1. We then chose stories 3, 4, and 7 because they are low priority, but they all fit in this iteration. The total time estimate, in hours, is 4 (story 5) + 2.5 (story 6) + 11.5 (story 3) + 2.5 (story 4) + 7 (story 7) = **27.5 hours**, which fits in our 36 hours to allocate per iteration.

Scenario-Based System Tests

End-to-End Scenario 1: Jessica has a Successful Day

1. Launch the “Sucessorator” app on Tuesday, January 23. You should see “Tuesday 1/23” at the top, and a plus button to the right of it. You should see “No goals for the Day. Click the + at the upper right to enter your Most Important Thing.” in the middle of the screen, in gray text. **(User story 1, 4)**
2. Click the plus button. You should see a keyboard and a microphone. Tap on the microphone and say “Prepare for the midterm”. **(User story 1, 3)**
3. You should see a check mark on the bottom right. Tap on the check mark. The keyboard should disappear. You should see the “Tuesday 1/23” text again, with “Prepare for the midterm” just below it. **(User story 1, 3, 4)**
4. Click the plus button again. You should see a keyboard and a microphone. Tap on the microphone and say “Grocery shopping”. **(User story 1, 3)**
5. Tap on the check mark in the bottom right. The keyboard should disappear. You should see the “Tuesday 1/23” text again, with “Prepare for the midterm” just below it followed by “Grocery shopping”. **(User story 1, 4)**
6. Make sure that Successorator is the displayed app, and sleep your phone. Launch the app again. You should see the same 2 goals as before (“Prepare for the midterm” and “Grocery shopping”). **(User story 8)**
7. Tap on “Prepare for the midterm”. It should move down to the bottom of the list and be in strikethrough. **(User story 2, 6)**
8. Using the keyboard/microphone mechanism described previously, enter “Text Maria” and press the check mark. You should see “Text Maria” after “Grocery Shopping”, but above “Prepare for the midterm”. **(User Story 1, 6, 7)**
9. Tap “Grocery Shopping”. It should move down to below “Text Maria” but above “Prepare for the midterm”, and be in strikethrough. Turn off your phone and go to sleep (system-based testing is tiring!) **(User Story 1, 2, 6, 7)**
10. The next morning, after 2 AM, launch the app again. The date at the top should say “Wednesday 1/24”. There should only be one goal in the list: “Text Maria”. **(User Story 4, 5)**

End-to-End Scenario 2: Our end-to-end scenario

1. Launch the “Successorator” app after 2 AM. You should see the current day at the top, formatted as “[Day of Week] [Month]/[Day]”. You should also see a plus button. You should see text on the screen that says “No goals for the Day. Click the + at the upper right to enter your Most Important Thing.”. **(User story 1, 4)**
2. Tap the plus button. You should see a text input field, checkmark button, and cancel button. **(User story 1)**

3. Tap the cancel button. You should be taken back to the goal list view, and there shouldn't be any goals in the list. It should still say "No goals for the Day. Click the + at the upper right to enter your Most Important Thing." **(User story 1)**
4. Tap the plus button again; you should again see the text input field, checkmark button, and cancel button. Tap on the text input field. You should see a keyboard. Type "Goal 1" using the keyboard. Press the checkmark button. The keyboard should disappear. You should be taken to the goal list view, and it should have one entry: "Goal 1". **(User story 1)**
5. Tap the plus button again; you should again see the input, checkmark button, and cancel button. Tap on the text input field. You should see a keyboard. Type "Ephemeral goal" into the text input field, then tap the cancel button. You should be taken to the goal list view, and it should still only show one entry: "Goal 1". **(User story 1)**
6. Tap the plus button again; you should again see the text input field, checkmark button, and cancel button. Tap the checkmark button without typing anything into the text input. You should still be on the same screen (input with buttons). Tap the cancel button. You should be taken to the goal list view, and it should still only show one entry: "Goal 1". **(User story 1)**
7. Tap the plus button again; you should again see the input, checkmark button, and cancel button. Type "Goal 2222222222222222222222222222222222222222222222222222222222222" into the text input field, and press the checkmark button. You should be taken to the goal list view, and it should have two entries. "Goal 1" should be on top, and "Goal 222…" should be under it, and should be truncated to fit on one line. **(User story 1, 7)**
8. Tap on "Goal 1". You should see "Goal 1" become strikethroughed, and move below "Goal 222…". **(User story 2)**
9. Tap on "Goal 1" again. You should see "Goal 1" become not strikethroughed anymore, and move to be above "Goal 222…". **(User story 2, 6, 7)**
10. Tap on "Goal 1" again. You should see "Goal 1" become strikethroughed, and move to be under "Goal 222…". Tap on "Goal 222…". You should see "Goal 1" and "Goal 222…" both be strikethroughed, and see "Goal 222…" be above "Goal 1". **(User story 2, 6, 7)**
11. Tap the plus button again; you should again see the input field, checkmark button, and cancel button. Tap on the input field. You should see a keyboard, and a microphone on the keyboard. Tap the microphone, and say "IVI" aloud. You should see "IVI" in the text input field. Tap the checkmark button. You should be taken to the goal list view, and you should see "IVI" at the top of the list, with "Goal 1" and "Goal 222…" below, in that order, both strikethroughed. **(User stories 3, 6, & 7)**
12. Tap on "Goal 222…". It should become not strikethroughed and move to be at the task list, with the order being "Goal 222…", then "IVI", then "Goal 1", and "Goal 1" should be the only strikethroughed item. **(User stories 2 & 7)**

13. Tap on “Goal 1”. It should become unstrikethroughed and move to the top of the list. The list should now be “Goal 1”, “IVI”, “Goal 222…”, and none of them should be strikethroughed. **(User stories 2 & 7)**
14. Using the plus button and either text or microphone inputs, as previously described, enter 20 more goals, named “NewGoal 1”, “NewGoal 2”, etc. through “NewGoal 20”. You should now be able to scroll up and down the goal list in order to see all of the 23 goals entered so far. The “NewGoal X” goals should be under the 3 previous goals, in order from 1 to 20. **(User story 1)**
15. Tap on “NewGoal 1”. It should move to the very bottom of the list, and be strikethroughed. **(User stories 2 & 6)**
16. Close the app. Reopen it. You should see the same goals in the same order as before you closed it, and the same date at the top. **(User stories 4 & 8)**
17. Close the app again. Open it again the next day BEFORE 2 AM. You should still see the same goals in the same order, and the same date at the top. **(User stories 4 & 5)**
18. Close the app again. Open it again on that day AFTER 2 AM. You should now see the date at the top be 1 day later than previously. You should see the same goals in the list in the same order, EXCEPT that “NewGoal 1” (the goal you crossed off yesterday) should no longer be in the list. **(User stories 4 & 5)**

Github Project

Link: https://github.com/orgs/CSE-110-Winter-2024/projects/173

Clarifications from Piazza

6. Is there a button to press to finish a task? Or a simple tap on the task gets rid of it?

A simple tap. However, it is not "rid": it is moved below the unfinished items and shown in strike-through, like "~~Prepare for the midterm~~". Also, I should add (whoops), that if you tap a finished item, it returns to being unfinished, at the top of the unfinished items.

1. When an item is tapped on to mark it as finished (and strike-through the item), where does it move to in the finished items section? Is it always the bottom of the list, always the top of the list, or something else?

Top of the finished section.

2. What format should the current date be in? (May 5th, 2025) (MM/DD/YYYY) (Monday)

Exactly as shown in Jessica's Scenario.

1. When a user closes the app's session, should they be able to see this day's tasks upon reopening the app?

Yes. I'm not sure what a "session" is, but it doesn't matter. The data doesn't vanish.

3. Do we want the task list to always update at 12:00 AM of the new day, even if we have our app open?

I'm glad you caught this. You might know that Daylight Saving Time starts at 2AM. So if we weren't going to reinvent the wheel, perhaps we should do the same thing. Let's update at 2AM of the next day, even if the app is open. By the way, it doesn't have to update at any particular moment. It should just seem that way from the user's perspective.

Are we using an android built in microphone (voice to text)?

Yes. Again, see @228, for a discussion of the Android Way.

Does the "add" button take the user to a different view, such as a pop up or new window, or does it stay on the current view?

UX design is up to you. Whatever is easy, whatever works, or whatever you like. Could just be a pop-up with a one-line text-entry box.

Is there a character limit for the item descriptions?

A todo item must take no more than one line. You can simply truncate (or let the UI truncate) whatever is typed in. You could also limit it in the text entry box, but it doesn't really matter.

Is the list scrollable?

My advice to you is to do the easiest thing, because MVC doesn't care (right now?). So it's up to you - you can either:
add a scrollbar
not add a scrollbar, but todo items at the end won't be visible
prohibit long lists (no more todo items can be entered after the list is full)

3. Is there a notification / reminder system?

No. That's why Jessica always has the app in the foreground when she sleeps her phone.

5. If all tasks are complete, does the screen show "no more goals"?

No. It just shows them all in strike-through.

It goes back to displaying that initial message in gray text.

1. if we download / first start the app when it is between midnight and 2 AM of some day (e.g. January 20, 2024) should the app display January 19 or January 20?

January 19th, or more precisely, "Friday 1/19". :)

Will the user be able to back out of creating a goal when they have clicked the plus sign?

Good catch, yes, a cancel button or whatever would be good. You determine the UX, though.

3) What should the order of rolled over tasks be?

Good question, thanks for catching that! Same as the previous day (unchanged).