

# Milestone 1: Planning

Eric Wang, Tristan Cooper, Sammyo Roychowdhury, Ryan Chen, Georolyn Ngo, Rahul Jones

## Risk Analysis [10%]

Risk: *[name]*

Description: *[narrative]*

Severity: [Low | Medium | High]

Resolution: *[plan]*

Status: [Resolved | In progress]

**Risk:** Not meeting enough

**Description:** We have only a few hours a week in common where we can all meet

**Severity:** High

**Resolution:** Have at least 3 team members at each meeting, and take meeting notes to keep everyone up to speed. Filled out a SyncCircle to find out everyone's common availabilities for meeting times. Daily stand-up meetings every MWF after lecture, TuTh online.

**Status:** Resolved

**Risk:** Unfamiliarity with Android development

**Description:** Unfamiliarity with Android development could slow down our development progress and lead to many bugs that will require additional debugging.

**Severity:** High

**Resolution:** Maintain realistic expectations for work estimates. Try to pair-program whenever possible. Text the slack whenever any issues pop up. Use generative AI to help understand Android development concepts.

**Status:** Resolved

**Risk:** Poor division of labor

**Description:** Inequitable split of tasks could lead to slowdowns, or features not being completed on time.

**Severity:** Low

**Resolution:** The daily standups should help prevent too many issues as people will be able to relay any issues they are having, which will let us redistribute tasks as needed to ensure timely completion.

**Status:** Unresolved

**Estimation of initial Velocity with justification:** 0.6

The recommended initial Velocity, according to HFSD, is 0.7, but due to our collective unfamiliarity with Android development, we are going with a slower velocity.

## Iteration Length:

# User Stories [40%]

(in [BDD style](#), named, estimated, prioritized, with [wire-framed UI screens](#), as necessary)

BDD format:

Title (one line describing the story)

Narrative:

As a [role]

I want [feature]

So that [benefit]

Acceptance Criteria: (presented as Scenarios)

Scenario 1: Title

Given [context]

And [some more context]...

When [event]

Then [outcome]

And [another outcome]...

### 1) Title: Display date

**Description:** *As a user*

*I want to see what day it is at the top of the screen  
so that I know what day it is.*

**Estimate:** 2

**Priority:** Low

**Acceptance Criteria:**

**Scenario 1:** Date change while the user is on the app

Given the date is January 31st

And the app is open

And it is midnight

When the day changes to February 1st

Then the date on the top of the app should change to the correct date, February 1st

**Scenario 2:** Date change while the user doesn't have the app open

Given the date is January 31st

And the app is closed

And it is the next day

When the user opens the app the next day

Then the date on the top of the app should change to the correct date, February 1st

## 2) Title: Add Goals

**Description:** *As a user*

*I want* to be able to add goals

*So that* I can see what I need to complete for the day.

**Estimate:** 16

**Priority:** High

**Acceptance Criteria:**

**Scenario 1:** No existing goals

Given that there are no goals in the list

And the screen says "No goals for the Day. Click the + at the upper right to enter your Most Important Thing"

When the user clicks the plus button, and types out/taps the mic button and writes/says "Get Lettuce"

And then clicks the check mark on the bottom right

Then "Get Lettuce" should be added to the task list and displayed on the screen.

And the keyboard should disappear

**Scenario 2:** Goals are present in the list

Given that "Get lettuce" is the only task we have on the list

When the user clicks the plus button

And types out/taps the mic button

And writes/says "Get tomato"

Then the new goal, "Get tomato" should be added to the bottom of the list, under "Get lettuce"

And the keyboard should disappear

## 3) Title: Mark off completed goals (depends on 2)

**Description:** *As a user*

*I want* the app to update on my completed tasks

*So that* I know what I have completed for the day.

**Estimate:** 8

**Priority:** Medium

**Acceptance Criteria:**

**Scenario 1:** User crosses off a completed goal

Given "Get lettuce" is in the list

And "Get lettuce" isn't completed

And "Get tomato" is under "Get lettuce"

**When** the user taps on "Get lettuce" to mark it as completed

**Then** "Get lettuce should be striked through and moved to the bottom of the list, under "Get tomato"

#### 4) Title: Unfinished task rollover and task erasure (depends on 3, 2, 1)

**Description:** *As a user*

*I want* the app to automatically move unfinished tasks to the next day and delete tasks that are completed once I go to the next day

*So that* I can finish tasks that are late and don't worry about tasks I have already finished the next day.

**Estimate:** 8

**Priority:** Medium

**Acceptance Criteria:**

##### Scenario 1:

**Given** I have 2 checked off goals in my list, "Get Lettuce" and "Get Tomato"

**And** I have no goals to rollover to the next day

**When** the day changes to the next day

**Then** the checked off goals should be deleted

**And** list should say that there are no tasks

##### Scenario 2:

**Given** I have "Get Tomato" as a goal AND it is not finished AND underneath it is another goal "Get Bun" AND it is not finished

**And** another goal "Get Lettuce" is completed

**And** I have turned off my phone but the app is still open on my phone

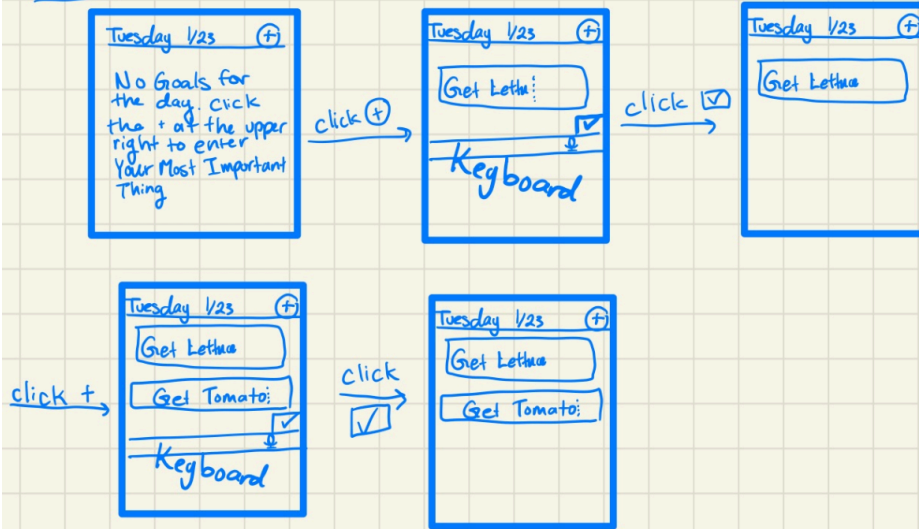
**When** the day changes to the next day and I open my phone

**Then** "Get Lettuce" should be deleted and disappear

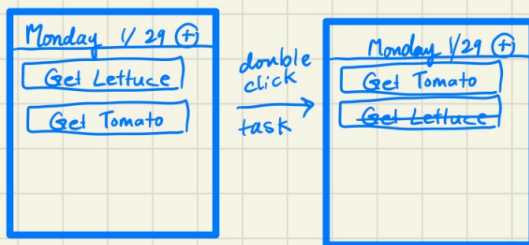
**And** "Get Tomato" should appear as the first goal on the list

**And** "Get Bun" should appear as the second goal on the list

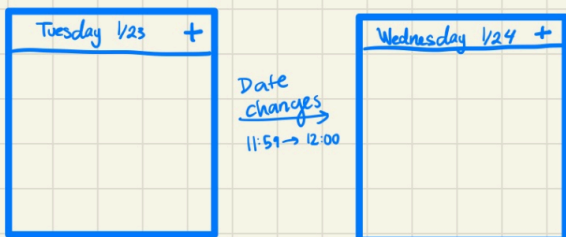
### Add Task



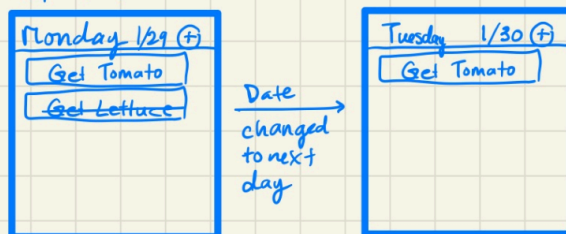
### Mark off completed



### Display Date



### Task Rollover



## Questions to Ask:

What's the expected behavior when the user taps on a striked-through task?

Is the entire goal a 'tap zone'? Or is there a specific area/button the user can tap on?

What should the order of rolled over tasks be?

What should adding a goal look like?

What happens if the added goal would go "offscreen" if there are a lot of goals

What kinds of information needs to be collected for a goal?

Is there a UI design vision, as far as colors and animations go?

## Tasks [20%]

(for the Stories in your *first* Iteration (assigned to Stories as appropriate, and estimated)

**Task:** Set up code base

**Estimate:** 2 hours

**Description:** Set up the project's structure before beginning development

**Task:** Implement the structure of the main UI

**Estimate:** 3 hours

**Description:** Construct components of the main user interface where tasks will be displayed. This is the main view of the app.

**User story 2:** Add goals

**Task:** Implement the add goal button

**Estimate:** 4 hours

**Description:** Once the user taps on the add goal button, the keyboard should be brought up and there should be a text field for the user to type in their goal.

**Task:** Implement the add goals UI

**Estimate:** 5 hours

**Description:** Design and construct the UI for adding a new goal to the list.

**Task:** Implement the confirm button

**Estimate:** 5 hours

**Description:** Once the user has entered their goal in the text field provided, the goal should be added to the bottom of the list of goals when the user taps on the confirm button in the bottom right corner.

**Task:** Testing

**Estimate:** 2 hours

**Description:** Test the implementation of the add goals functionality and make sure it meets the two acceptance criteria of the user story. The test should include black- and white-box testing.

**User story 3:** Mark off completed goals

**Task:** Implement tap goal button

**Estimate:** 2 hours

**Description:** Once the user taps on one of the goals, it should be marked as complete (backend)

**Task:** Move goals from incomplete to complete

**Estimate:** 4 hours

**Description:** After a goal is marked as completed, it should be moved to the bottom of the completed goals list, and be striked through.

**Task:** Testing

**Estimate:** 2 hours

**Description:** Check if the feature works. Test cases where all goals have been checked off, the goal is clicked again, etc.

**User story 4:** Unfinished task rollover and task erasure

**Task:** Implement Unfinished Task Rollover Logic

**Estimate:** 2 hours

**Description:** Develop the functionality that identifies unfinished tasks at the end of the calendar day and adds them to the next day.

**Task:** Implement Completed Task Erasure Logic

**Estimate:** 2 hours

**Description:** Develop logic to remove completed tasks from the task list at the end/change of the day.

**Task:** Update change-of-day UI

**Estimate:** 3 hours

**Description:** Update the application UI to reflect the changes made by the business logic above. Includes removing completed tasks from the view on day-change and adding unfinished tasks to the top of the next day's list.

**Task:** Testing

**Estimate:** 1 hour

**Description:** Ensure some goals are marked as completed, and some are left unfinished by the end of the day. On the new day, verify that completed tasks are erased and unfinished tasks are rolled over and displayed as the day's goals.

## Two Iterations [10%]

(Named, with all User Stories and "loose" Tasks (if any) assigned to Iterations)

### Iteration 1:

**Total Hours:** 23

**Task:** Set up code base

**Estimate:** 2 hours

**Description:** Set up the project's structure before beginning development

**Task:** Implement the structure of the main UI

**Estimate:** 3 hours

**Description:** Construct components of the main user interface where tasks will be displayed. This is the main view of the app.

### User story 2: Add goals

**Task:** Implement the add goal button

**Estimate:** 4 hours

**Description:** Once the user taps on the add goal button, the keyboard should be brought up, and there should be a text field for the user to type in their goal.

**Task:** Implement the add goals UI

**Estimate:** 5 hours

**Description:** Design and construct the UI for adding a new goal to the list.

**Task:** Implement the confirm button

**Estimate:** 5 hours

**Description:** Once the user has entered their goal in the text field provided, the goal should be added to the bottom of the list of goals when the user taps on the confirm button in the bottom right corner.

**Task:** Testing

**Estimate:** 2 hours

**Description:** Test the implementation of the add goals functionality and make sure it meets the two acceptance criteria of the user story. The test should include black- and white-box testing.



**User story 1: Display Date**

**Task:** Implement Android Studio digital clock

**Estimate:** 1.5 hours

**Description:** Once the user opens the app, a digital clock with the correct time should be visible on the top left of the main UI view.

**Task:** Test clock when a new day starts

**Estimate:** 0.5 hours

**Description:** Test the implementation of the clock when a new day starts. The test should include white-box testing.

**Iteration 2:**

**Total hours:** 24 hours

**User story 3: Mark off completed goals**

**Task:** Implement tap goal button

**Estimate:** 2 hours

**Description:** Once the user taps on one of the goals, an example function should be called

**Task:** Move goals function

**Estimate:** 4 hours

**Description:** The function should take the selected goal and move it to a list under the regular goals that will have letters with a line going through them.

**Task:** Testing

**Estimate:** 2 hours

**Description:** Check if the feature works. Test cases where all goals have been checked off, the goal is clicked again, etc.

**User story 4: Unfinished task rollover and task erasure**

**Task:** Implement Unfinished Task Rollover Logic

**Estimate:** 2 hours

**Description:** Develop the functionality that identifies unfinished tasks at the end of the calendar day and adds them to the next day.

**Task:** Implement Completed Task Erasure Logic

**Estimate:** 2 hours

**Description:** Develop logic to remove completed tasks from the task list at the end/change of the day.

**Task:** Update change-of-day UI

**Estimate:** 3 hours

**Description:** Update the application UI to reflect the changes made by the business logic above. Includes removing completed tasks from the view on day-change and adding unfinished tasks to the top of the next day's list.

**Task:** Testing

**Estimate:** 1 hour

**Description:** Ensure some goals are marked as completed, and some are left unfinished by the end of the day. On the new day, verify that completed tasks are erased and unfinished tasks are rolled over and displayed as the day's goals.

## Planning Poker documentation [5%]

**All team members' estimates** (remember, for each user story, play until modestly converged):

<https://docs.google.com/spreadsheets/d/1550sDo4FZsdpro03Q4Kkd9l6rsJVOadfDmM9BZkgvdE/edit?usp=sharing>

S#	Name	Hand	False assumptions uncovered
1	Display Date	2 2 1 2 4 4	None
2	Add goals	4 16 4 16 8 4	Saving app data will be easy to implement / a built in function
2	Add goals	8 8 8 16 8 16	None
3	Mark off completed goals	4 8 2 8 4 4	Can cards be unselected and have their position reversed (feature creep?)
4	Unfinished task rollover and task erasure	4 16 8 16 8 8	None

## Voting history

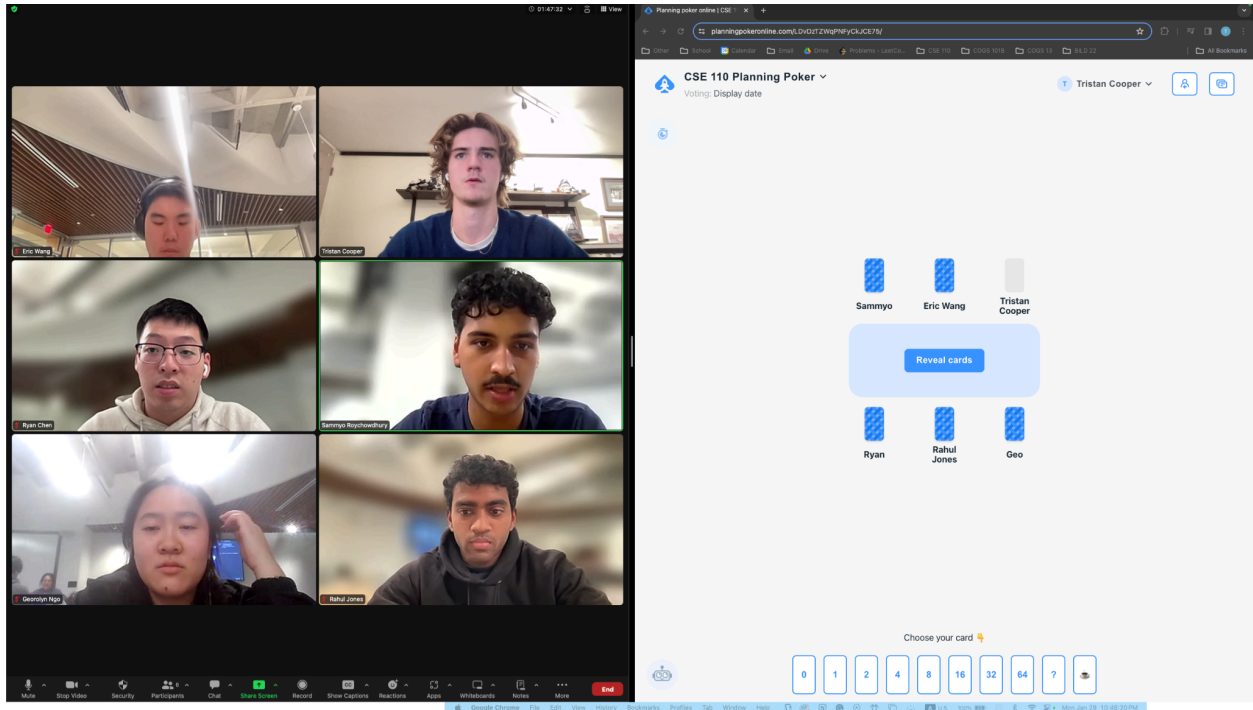
Meeting Mon Jan 29 2024 (1...



Issue name	Result	Agreement	Duration	Date	Voted / total	Player results
January 29, 2024						
PP-1 - Display date	2			29 Jan, 22:48	6/6	Geo (1), Sammyo (2), Eric Wang (2), Rahul Jones (2), Ryan (4), Tristan Cooper (4)
PP-2 - Add goals	8		06:18	29 Jan, 22:54	6/6	Sammyo (4), Tristan Cooper (4), Geo (4), Ryan (8), Eric Wang (16), Rahul Jones (16)
PP-2 - Add goals	8		06:11	29 Jan, 23:01	6/6	Ryan (8), Sammyo (8), Eric Wang (8), Geo (8), Rahul Jones (16), Tristan Cooper (16)
PP-3 - Mark off completed goals	16		04:12	29 Jan, 23:05	1/6	Geo (16)
PP-3 - Mark off completed goals	4		01:07	29 Jan, 23:06	6/6	Geo (2), Ryan (4), Sammyo (4), Tristan Cooper (4), Eric Wang (8), Rahul Jones (8)
PP-4 - Unfinished task rollover and task erasure	8		07:03	29 Jan, 23:13	6/6	Sammyo (4), Ryan (8), Tristan Cooper (8), Geo (8), Eric Wang (16), Rahul Jones (16)

## Assumptions Uncovered:

**Photo of you playing planning poker** (there is an upload limit size; watch your photo size)



## Scenario-Based System Tests [8%]

These are based on the end-to-end scenarios given in the requirements (plus removed assumptions and details of UI interaction). Scenarios must be added for implied end-to-end Scenarios.

Annotated with User Stories covered (should cover all, of course)

### Jessica has a Successful Day

1. Start the app by tapping the app icon. You should see a mostly empty screen with "Sunday 2/3" at the top. [User Story #1]
2. Tap the + button and a keyboard should pop up.
3. Tap the microphone button, and say "prepare for the midterm" and tap the check-mark. The keyboard should disappear, and 'Prepare for the midterm' should appear near the top of the page, just below the "Tuesday" bar. [User Story #2]
4. Tap the + button and a keyboard should pop up.
5. Type in "Grocery shopping" and tap the check-mark. The keyboard should disappear, and 'Grocery Shopping' should appear right underneath 'Prepare for the midterm'. [User Story #2]

6. Tap the 'Prepare for the midterm' task to mark it as complete. It should move down the list after 'Grocery shopping' and appears in strike-through: '~~Prepare for the midterm~~'. [User Story #3]
7. Tap the + button and a keyboard should pop up.
8. Type in "text Maria" and tap the check-mark. The keyboard should disappear, and 'text Maria' should appear between 'Grocery Shopping' and '~~Prepare for the midterm~~'
9. Tap the 'Grocery Shopping' task to mark it as complete. It should move down the list after '~~Prepare for the midterm~~' and appear in strike-through: '~~Grocery Shopping~~'. [User Story #3]
10. Turn off your phone.
11. Wait till the next day. Turn on your phone. You should see the date displayed as "Monday 2/4" [User Story #2]. 'text Maria' should still be displayed at the top. The two completed items from the previous day are gone [User Story #4].