# Sprint 1 Planning Document

Team 20: Divay Gupta, Kartik Uppalapati, Maeve Tra, Meghna Rajagopalan, Oleg Aprelikov

## Sprint Overview

In this sprint, we aim to set up the basic components of the application. This includes user authentication (login and sign up), setting up the database for the closet, setting up the screens on the app to view the clothing items in the closet as well as the laundry basket, connecting the app to the weather API and the user’s Google Calendar in order to get information about the weather and events, and accessing the phone camera and photo gallery for pictures of clothes. We also hope to complete the basic recommendation algorithm once the database and the weather/event checking APIs are all set up. The recommendation algorithm will read all the information about the weather, the user’s events for the day, and which clothes are not in the laundry basket in order to determine which clothes to recommend to the user. For this sprint, clothes will be recommended in the order in which they are obtained from the database, but in a future sprint, we will enable users to mark a specific clothing item as a favorite so that it is recommended before non-favorite clothing items. Additionally, clothing items that are worn more often will be recommended before clothing items that are worn less often. In case of conflict, the favorite before non-favorite rule takes higher priority.

**Scrum Master:** Meghna Rajagopalan

**Meeting Plan:** Thursday @ 8:00 am, in-person

**Risks and Challenges:** Most of us are using Flutter for the first time, so that will be a challenge for us to learn how to use properly. Additionally, one thing we have learned about Flutter is that the front-end and back-end are developed simultaneously. This will make unit-testing more of a challenge.

## Current Sprint Detail

### User Story #1

As a user, I would like to be able to register an account/login using my Google account (gmail).

| # | Description | Estimated Time | Owner |
| --- | --- | --- | --- |
| 1 | Configure Google API | 2 hours | Meghna |
| 2 | Configure Google Sign-in | 1 hour | Meghna |
| 3 | Create a login button that starts the sign-in flow and a sign up button for registering | 1 hour | Meghna |
| 4 | Authenticate with a backend server | 2 hours | Meghna |
| 5 | Enabling Server-Side Access | 2 hours | Meghna |
| 6 | Debug and test with mock login | 1 hours | Meghna |

**Acceptance Criteria**

* Given that the login button is correctly set up to start the sign-in flow, when the user clicks on the login button, a Google account page should appear with the user’s list of Google accounts to choose from.
* Given that Google Sign-in is correctly configured, once the user signs in with their account, we will be able to retrieve their basic profile including Google ID, name, profile URL, and email address.
* Given that the backend server is set up for authentication, once the user signs in with their Google account, if the user has an account already, the server will establish an authenticated session for the user. If the user does not have an account, the server creates a new user record from the information in the ID token payload, and establishes a session for the user.

### User Story #2

As a user, I would like to be able to open my phone’s camera from the app and take pictures of my clothes to add them to the closet.

| # | Description | Estimated Time | Owner |
| --- | --- | --- | --- |
| 1 | Create and initialize a CameraController (from Flutter’s *camera* package) | 4 hours (2 hours to learn Flutter, 2 to configure) | Maeve |
| 2 | Create an option chooser for selecting the camera (once they choose the plus button on the nav bar) | 1.5 hours | Maeve |
| 3 | Displaying Camera Preview/camera’s feed | 2.5 hours | Maeve |
| 4 | Create a button to take pictures | 1 hour | Maeve |
| 5 | Display the just-taken picture | 2.5 hours | Maeve |
| 6 | Create a “Go back” button to go back to the camera’s feed | 1 hour | Maeve |
| 7 | Create a “Continue” button and connect to database to add pictures to database | 2 hours | Maeve |
| 8 | Debug and test with mock pictures | 2 hours | Maeve |

**Acceptance Criteria**

* Given that the camera is correctly set up, once the user selects the camera from the nav bar, the camera’s feed should appear with a button to take pictures.
* Given that the button triggers the camera, when the user taps on the button, a picture will be taken and displayed with a “Go back” button and a “Continue” button.
* Given that the “Continue” button is correctly set up, when the user chooses the “Continue” button, the picture will be saved to the database.

### User Story #3

As a user, I would also like the option to upload photos from my phone’s gallery.

| # | Description | Estimated Time | Owner |
| --- | --- | --- | --- |
| 1 | Configure Flutter’s Image Picker package for gallery access | 1 hour | Maeve |
| 2 | Create an option chooser for selecting the gallery (once they choose the plus button on the nav bar) | 1 hour | Maeve |
| 3 | Create and configure the Image view on screen | 2 hours | Maeve |
| 4 | Connect the Image view to the “Continue” button in the Camera part | 1.5 hours | Maeve |
| 5 | Debug and test with mock gallery | 2 hours | Maeve |

**Acceptance Criteria**

* Given that the gallery access is correctly set up, once the user selects the gallery from the nav bar, the user’s photo gallery should appear with the option to select pictures.
* Given that the Image view is configured, when the user taps on a picture in their gallery, a full-screen picture should appear with the “Continue” button.
* Given that the “Continue” button is correctly set up, when the user chooses the “Continue” button, the picture will be saved to the database.

### User Story #4

As a user, I would like to be able to add/edit details of all my clothing items (without the AI feature)

| # | Description | Estimated Time | Owner |
| --- | --- | --- | --- |
| 1 | Create a fillable/editable detail page | 2 hours | Divay |
| 2 | Make some fields required (e.g. category, subcategory) | 1.5 hours | Divay |
| 3 | Create a “Save” button | 1.5 hours | Divay |
| 4 | Add/Update item with the details to database once the user clicks “save” | 2 hours | Divay |
| 5 | Connect the database to a grid view | 2.5 hours | Divay |
| 6 | Debug and test with unit test | 3 hours | Divay |

**Acceptance Criteria**

* Given that the detail page is configured to be fillable and editable, when the user taps on one of the fields, the keyboard will appear and allow the user to type in the details.
* Given that some fields are required, when the user tries to save the details without filling out these fields, a warning should appear and the detail page will not be saved.
* Given that the “Save” button is set up to connect the detail page with the database, when the user chooses “Save”, the details of the item should be updated in the database.

### User Story #5

As a user, I would like to be able to view all my items in the closet, under a specific category in the closet (tops, bottoms, hats, shoes, etc).

| # | Description | Estimated Time | Owner |
| --- | --- | --- | --- |
| 1 | Create a grid-view page. This is the closet page. | 2 hours | Maeve |
| 2 | Request all the user’s clothes from the database | 2 hours | Maeve |
| 3 | Display 1 clothing item per grid square. Each grid square should have the picture of the clothing item and it’s name. | 2 hours | Maeve |
| 4 | Make tabs at the top of the screen to sort the clothes (tops, bottoms, hats, shoes, etc) | 3 hours | Oleg |
| 5 | Program the app to display only those clothes that are in the selected category (based on the selected tab) | 2 hours | Oleg |
| 6 | Program the app to display a message (as a label where the grid-view is) if any of the categories have no clothes in that category in the database. The message should say “No results found.” | 2 hours | Oleg |
| 7 | Make the grid squares clickable. Selecting one should lead to the uneditable details page of that specific clothing item. | 2 hours | Oleg |
| 8 | Debug and test | 3 hours | Oleg |

**Acceptance Criteria**

* Given that the user has entered clothes into the database, when they open their closet, Then the page should display a gridview of their clothes, with each square containing a picture of the clothing item and the name of the clothing item
* Given that the user has entered the correct category of clothing for each item in the database, When they select a tab on the closet page, Then the page should display a gridview of exactly those clothes that fall in that category
* Given that the user has not entered any clothes into the database for a specific category, When they open their closet and select that tab, Then the page should display a message saying that no results were found
* Given that the user has opened the closet page, When they select a specific square in the grid-view, Then the non-editable details page should open for that specific clothing item.

### User Story #6

As a user, I would like to be able to select a specific clothing item and see the details for that item on a details page

| # | Description | Estimated Time | Owner |
| --- | --- | --- | --- |
| 1 | Create a non-editable details page | 2 hours | Meghna |
| 2 | Request all the non-empty fields in the database | 2 hours | Meghna |
| 3 | Display the data from each non-empty field | 2 hours | Meghna |
| 4 | Add a button that leads to the editable details page so that the user can make edits | 1 hour | Meghna |
| 5 | Debug and test with unit test | 2 hours | Meghna |

**Acceptance Criteria**

* Given that the user has added some clothing items into the database, When the user clicks opens the non-editable details page, Then it should display all the details that the user entered accurately
* Given that the user has added some clothing items into the database, When the user opens the non-editable details page, Then it should display all the details that the user entered in a neat and aesthetically pleasing way
* Given that the user has the non-editable details page open, When they click the edit button, Then the app should take them to the editable details page.

### User Story #7

As a user, I would like to be able to search for an item using the search bar.

| # | Description | Estimated Time | Owner |
| --- | --- | --- | --- |
| 1 | Add a text box to the closet page, above the grid-view. This is the search box. | 2 hours | Oleg |
| 2 | Add a search button next to the text box. | 1 hour | Oleg |
| 3 | Program the search button so that when it is pressed, if the search box is non-empty, it should make a request to the database for the user’s clothes and only display the results that include in their name or details the word(s) that the user typed in the search box | 4 hours | Oleg |
| 4 | Program the grid-view to reload with the new results whenever the user clicks the search button (and the search box is non-empty) | 1 hour | Oleg |
| 5 | Program the page so that if for the selected category (based on tab selected) none of the results match what the user typed in, then the grid-view should be invisible, and a label (uneditable text) should display no results found | 2 hours | Oleg |
| 6 | Debug and test with unit test | 3 hours | Oleg |

**Acceptance Criteria**

* Given that the user has entered text in the search box, When they click the search button, Then the grid view should reload to display exactly the clothing items that contain the text entered by the user in the name or details of the item
* Given that the user has not entered text in the search box, When they click the search button, Then the grid-view should not reload and instead remain unchanged
* Given that the user has entered text in the search box that is not included in any of the clothing items for the selected category (based on tab selected), then the grid-view should become invisible and the message “No results found” should be displayed.

### User Story #8

As a user, I would like to receive recommendations on what to wear based on the weather.

| # | Description | Estimated Time | Owner |
| --- | --- | --- | --- |
| 1 | Connect the app to the Weather API | 2 hours | Kartik |
| 2 | Add to the user registration process: request user to allow app to access their location | 1 hour | Kartik |
| 3 | Get the user’s location (only if the user has allowed the location to be accessed by the app) | 1 hour | Kartik |
| 4 | Program the app to request the weather from the API for the user’s location every hour | 2 hours | Kartik |
| 5 | Store the weather information in a variable that can be accessed by the recommendation algorithm | 1 hour | Kartik |
| 6 | Create a button so that the user can request the app to update its weather data. Pressing the button should cause the app to request the weather for the user’s location. | 1 hour | Kartik |
| 7 | Debug and test with unit test | 1 hour | Kartik |

**Acceptance Criteria**

* Given that the user has clicked the sign-up button and entered their Google account details, When the Google Account has been verified, Then the app should request to access the user’s location
* Given that the user has allowed the app to access the user’s location, When a periodic hour has passed, Then the app should request the weather data from the API for the user’s location
* Given that the user has allowed the app to access the user’s location, When the user clicks on the Update Weather button, Then the app should request the weather data from the API for the user’s location
* Given that the user has allowed the app to access the user’s location, When the weather has been requested from the API, Then the obtained weather information should be accurate to the location and time of the user
* Given that the user has allowed the app to access the user’s location, When the weather has been requested from the API, Then the obtained weather information should be accessible from the necessary classes in the app

### User Story #9

As a user, I would like to receive recommendations on what to wear based on my Google Calendar.

| # | Description | Estimated Time | Owner |
| --- | --- | --- | --- |
| 1 | Add to the user registration process: request user to allow app to access their Google calendar | 1 hour | Kartik |
| 2 | Connect the app to the user’s Google calendar (only if user has allowed access) | 2 hours | Kartik |
| 3 | Program access to the user’s calendar for the next 7 days. This access should take place at midnight every day. | 2 hours | Kartik |
| 4 | Store this data in a variable that can be accessed by the recommendation algorithm as well as the laundry page | 1 hour | Kartik |
| 5 | Create a button so that the user can press it to trigger updating the calendar information to the latest information. | 1 hour | Kartik |
| 6 | Debug and test with unit test | 2 hour | Kartik |

**Acceptance Criteria**

* Given that the user has clicked the sign-up button and entered their Google account details, once the Google Account has been verified, the app should request access to the user’s Google Calendar.
* Given that the user has allowed the app to access the user’s Google Calendar, When the time is 12:00am, Then the app should request the Calendar data from the user’s Google Calendar for the next 7 days.
* Given that the user has allowed the app to access the user’s Google Calendar, When the user clicks on the Update Calendar button, Then the app should request the Calendar data from the user’s Google Calendar for the next 7 days.
* Given that the user has allowed the app to access the user’s Google Calendar, When the user’s calendar info has been requested for the next 7 days, Then the obtained calendar info should be correct information for the next 7 days.
* Given that the user has allowed the app to access the user’s Google Calendar, When the user’s calendar info has been requested for the next 7 days, Then the obtained calendar info should be accessible from all the necessary classes in the app.

### User Story #10

As a user, I would like to be able to receive recommendations on what to wear

| # | Description | Estimated Time | Owner |
| --- | --- | --- | --- |
| 1 | Let user decide on how many outfit recommendations they want, (tbd), per day and at what time | 2 hour | Kartik |
| 2 | Pull in weather information and based weather filter recommended clothes set accordingly (if it’s raining don’t include shorts in clothes to recommend) | 3 hour | Kartik |
| 3 | Pull in daily event information from google calendar (if user allows it and has google calendar set up) and recommend clothes based on user’s events | 3 hour | Kartik |
| 4 | Find similar items between both recommendations and put them at priority. | 2 hour | Kartik |
| 5 | Display the recommendations | 1 hour | Kartik |
| 6 | Debug and test with unit test | 2 hour | Kartik |

**Acceptance Criteria**

* Given that the user allows the app to use the current location which is needed for the weather API, the user’s google account is verified, and their google calendar isn’t empty the recommendation algorithm should suggest an outfit for the user.
* Given that the user does not allow the app to know the user’s current location and the user does not have a google account or does not use google calendar, the recommendation algorithm should suggest an outfit for the user based on time of day.
* Given that the user allows the app to give notifications, a notification should be given every time an outfit recommendation has been made.
* Given that there are available clothes in the closet for an outfit to be made and the recommendation quota has not been satisfied yet, make an outfit recommendation at the user’s preset time.

### User Story #11

As a user, I would like to be able to reset my password

| # | Description | Estimated Time | Owner |
| --- | --- | --- | --- |
| 1 | Create a profile page that opens when the user clicks on the person outline from the navigation bar of the app | 2 hours | Meghna |
| 2 | If a user logged in with a gmail account, use google API to let user reset password using third party OAauth method | 2 hours | Meghna |
| 3 | Let users with non-gmail accounts change password by clicking on reset password option | 2 hours | Meghna |
| 4 | Let user password reset by typing in new strong password twice and then pressing confirm button | 2 hours | Meghna |
| 5 | Save new password in database | 2 hours | Meghna |
| 6 | Debug and test with unit test | 2 hours | Meghna |

**Acceptance Criteria**

* Given that the user is logged in with a gmail account, the app should use the google API third party OAuth method to reset their password through the app
* Given that the user is logged in with a non-gmail account, they can also reset their password through the app
* A password can be reset if the new password adheres to the strong password policy (1 or more uppercase letters, 1 or more lowercase letters, and 1 or more numbers) and then the user presses the confirm button to save the new password to the database
* Given that the user has reset their password, the app should detect a password change and log them out so the user can log back in with the new password

### User Story #12

As a user, I would like to be able to approve or decline the recommendations

| # | Description | Estimated Time | Owner |
| --- | --- | --- | --- |
| 1 | Split outfit recommendation into parts (hat, top, bottom, socks, shoes) | 1 hour | Oleg |
| 2 | Let user swipe left or right on each part of the outfit (left for reject and right for accept) | 1 hour | Oleg |
| 3 | Let user get new recommendation for outfit part if they swiped left on it | 1 hour | Oleg |
| 4 | If user swipes right, then increase the chances of that item being recommended for future outfits | 2 hours | Oleg |
| 5 | Debug and test with unit test | 2 hours | Oleg |

**Acceptance Criteria**

* Given that the user has set at least one outfit recommendation per day, the homepage should display an outfit in its individual parts at the user’s preset time of day
* Given that the user swipes left on a piece of clothing in the outfit recommendation, replace it with a new clothing recommendation. If all the clothes in that category are in the laundry then play a mini shake animation on the clothing piece to signify there is nothing else to recommend.
* Given that the user swipes right on a piece of clothing in the outfit recommendation, then increase that item’s recommendation probability by a multiplier of 1.15

### User Story #13

As a user, I would like to be able to select a clothing item in the closet and see if it is in the “Laundry basket” or not on the item’s details page

| # | Description | Estimated Time | Owner |
| --- | --- | --- | --- |
| 1 | Add a tag or note in the non-editable details page that says “In laundry basket” | 1 hour | Oleg |
| 2 | Debug and test with unit test | 1 hour | Oleg |

**Acceptance Criteria**

* Given that the user or app has moved an existing clothing item into the laundry basket, when they click on the clothing item, display the status of the clothing item accurately.
* Given that the user or app has moved an existing clothing item into the laundry basket, when they click on the clothing item, display the status text for “In Laundry Basket” in an aesthetically pleasing box or tag format.
* Given that the user or app has moved an existing clothing item into the laundry basket, when the non-editable page for that item is open, the app should take them to the editable details page.

### User Story #14

As a user, I would like to be able to remove items from the “Laundry basket” using a toggle on the details page of that specific clothing item

| # | Description | Estimated Time | Owner |
| --- | --- | --- | --- |
| 1 | Create toggle to change Laundry basket status of clothing item on the editable details page | 1.5 hours | Divay |
| 2 | Animate toggle when the user clicks on it | 1 hour | Divay |
| 2 | Update the information for the clothing item from database | 2 hours | Divay |
| 3 | Display the updated information for the clothing item from database | 1.5 hours | Divay |
| 4 | Debug and test with unit test | 1 hour | Oleg |

**Acceptance Criteria**

* Given that the user or app has clicked on a clothing item, when they click the edit button on the non-editable details page, display the item’s status in the database appropriately.
* Given that the user has clicked on a clothing item, when they click the edit button on the non-editable details page, display the laundry basket status toggle in an aesthetically pleasing way.
* Given that the user has clicked on a clothing item, when the toggle on the editable page for that item is clicked, animate the toggle appropriately.
* Given that the user has clicked on the toggle for a clothing item, when the save button on the editable page for that item is clicked, update the laundry status for the item in the database as needed.

### User Story #15

As a user, I would like to be able to select a clothing item in the “Laundry basket” and see its details on a details page.

| # | Description | Estimated Time | Owner |
| --- | --- | --- | --- |
| 1 | Create a fillable/editable detail page | 2 hours | Divay |
| 2 | Make some fields required (e.g. category, subcategory) | 1.5 hours | Divay |
| 3 | Create a “save” button | 1.5 hours | Divay |
| 4 | Add/Update item with the details to database once the user clicks “save” | 2 hours | Divay |
| 5 | Connect the database to a grid-view | 2.5 hours | Divay |
| 6 | Debug and test with unit test | 3 hours | Divay |

**Acceptance Criteria**

* Given that the user or app has moved an existing clothing item into the laundry basket, when they click on the clothing item, display the details of the clothing item accurately.
* Given that the user or app has moved an existing clothing item into the laundry basket, when they open the non-editable details page, display all the item details in an aesthetically pleasing way.
* Given that the user has the non-editable details page open, When they click the edit button, take them to the editable details page.
* Given that the user has the non-editable details page open, when the “save” button is clicked, update the item details in the database appropriately.

## Remaining Backlog

### Functional Requirements

**User account**

1. ~~As a user, I would like to be able to register an account using my Google account (gmail).~~
2. As a user, I would like to be able to login and manage my account.
3. ~~As a user, I would like to be able to reset my password.~~
4. ~~As a user, I would like to be able to sync my Google calendar with the app.~~

**Managing closet**

1. As a user, I would like to be able to add clothing items to my closet
2. As a user, I would like to be able to remove clothing items
3. ~~As a user, I would like to be able to select a specific clothing item and see the details for that item on a details page~~
4. ~~As a user, I would like to be able to edit details of all my clothing items.~~
5. ~~As a user, I would like to be able to open my phone’s camera from the app and take pictures of my clothes to add them to the closet.~~
6. ~~As a user, I would also like the option to upload photos from my phone’s gallery.~~
7. As a user, I would like to have the details of the items (color, sleeve length, etc) to be recognized by an Artificial intelligence and filled out automatically.
8. As a user, I would like to be able to edit any details that were recognized incorrectly by the Artificial Intelligence, and also report that the mistake was made.
9. ~~As a user, I would like to be able to view all my items in the closet, under a specific category in the closet (tops, bottoms, hats, shoes, etc).~~
10. ~~As a user, I would like to be able to search for an item using the search bar.~~
11. As a user, I would like to be able to filter the items based on their colors, materials, etc.
12. As a user, I would like to extract my item from its original background and put it on a custom background (if time allows).

**Clothing Recommendations**

1. ~~As a user, I would like to receive recommendations on what to wear based on the weather~~
2. ~~As a user, I would like to receive recommendations on what to wear based on my schedule in my Google calendar.~~
3. ~~As a user, I would like to receive recommendations for one type of clothing item (tops, bottoms, shoes, hats, etc) at a time.~~
4. As a user, I would also like to receive full outfit combination recommendations if I choose the option to allow it (if time allows)
5. ~~As a user, I would like to be able to approve or decline the recommendations.~~
6. As a user, I would like to be able to receive new recommendations if I decline the old one.
7. As a user, I would like the clothing item to be automatically put in the “Laundry basket” after I have accepted it a certain (user-specified) number of times
8. As a user, I would like to be able to say that an item is my favorite so that it is recommended more often.

**Outfit Combinations**

1. As a user, I would like to be able to mix and match items in my closet to create outfit combinations.
2. As a user, I would like to be able to save an outfit combination that I like.
3. As a user, I would like to be able to browse my saved outfit combinations.
4. As a user, I would like to be able to edit a saved outfit combination.
5. As a user, I would like to be able to delete a saved outfit combination.
6. As a user, I would like to be able to name a saved outfit combination.
7. As a user, I would like to be able to search for a saved combination using its name.
8. As a user, I would like to be able to upload a picture of myself and then see the clothing items modelled on the picture. (if time allows)
9. As a user, I would like to be able to assign an outfit combination for a specific date.
10. As a user, I would like to be able to set weather preferences for a specific outfit combination so that the app recommends that outfit on a day with that weather (if time allows)

**Laundry**

1. ~~As a user, I would like to be able to select a clothing item in the closet and see if it is in the “Laundry basket” or not on the item’s details page~~
2. As a user, I would like to be able to see what items are in the “Laundry basket” (looks similar to seeing what clothes are in the main closet)
3. ~~As a user, I would like to be able to select a clothing item in the “Laundry basket” and see its details on a details page.~~
4. As a user, I would like to be able to add items to the “Laundry basket” using a toggle on the details page of that specific clothing item
5. ~~As a user, I would like to be able to remove items from the “Laundry basket” using a toggle on the details page of that specific clothing item~~
6. As a user, I would like to be able to clear all items from the “Laundry basket” at once.
7. As a user, I would like to be able to tell if an item is in the “Laundry basket” from the main closet (“To Laundry” and “In Laundry” button).
8. As a user, I would like to be able to set how often I want to do my laundry.
9. As a user, I would like to be reminded to wash an item if it is still in the “Laundry basket” (the reminder is sent periodically, with the time period being the user-selected time for how often they want to do laundry; default is 1 week)
10. As a user, I would like to be reminded to wash an item if I need the item for an upcoming event, or if the outfit has been assigned to an upcoming date.
11. As a user, I would like to be able to set how far ahead I want to be reminded to do my laundry.
12. As a user, I would like to be reminded to do laundry if my last laundry load exceeds the duration set above (i.e. if I want to do laundry every week, remind me if I haven’t done it on the 8th day).
13. As a user, I would like to receive notifications from the app to do laundry.
14. As a user, I would like to see the last time a clothing item was washed on its details page (if time allows)
15. As a user, I would like to set reminders for washing a specific item for a specific time period that might be different from the normal laundry time period (example, for jackets that are not washed as often) (if time allows)

### 