Summary

- **Process since M6**
  - Backend
    - Partially finished outfit endpoints (complex logic)
    - Improved image uploading
    - Completed the integration with Google Calendar API
    - Added unit tests in user and clothes modules
    - Set up Github Actions for continuous integration
  - Frontend
    - Enabled add clothes into closet with image and attributes
    - Allowed select clothes images from gallery
    - Enabled display selected image in the closet
    - Enabled get outfit suggestion
    - Completed logout functionality
    - Added login and register UI tests
    - Polished Github actions for android continuous integration

- **Plans until M8**
  - Backend
    - Complete all outfit endpoints
    - Improve outfits API to save user's feedback
    - Allow user to save his/her city in the profile
    - Improve weather endpoint's performance
    - Add unit tests for other major modules in backend
  - Frontend
    - Implement update and delete clothes items from the app
    - Implement upload image from camera
    - Complete like/dislike button functionality
    - Complete calendar functionality
    - Add more UI tests for other major use cases
    - Run Codacy for automated code review

- **Major decisions and changes**
  - None

- **Member contributions**
  - Steven Yang
    - Improved image uploading
    - Finished part of outfit coding

        - Set up Github Actions for continuous integration
    - John Li
        - Completed the integration with Google Calendar API
        - Added unit tests in user module
        - Designed the logic to get outfits
        - Finished part of outfit coding
    - Zhuoyi Li
        - Allowed user to select clothes attributes and send data to the server
        - Completed logout functionality
        - Added login and register UI tests
        - Polished Github actions for android continuous integration
    - Jingyang Sun
        - Allowed select clothes image from gallery
        - Enabled display selected image in the closet
        - Enabled upload clothes image to the server
        - Enabled get outfit suggestion

Testing - Backend

Unit tests

Interface	**Expected Behavior**
User Module	
POST /api/users/signup Expects input body: { name: String, email: String, password: String }	- Invalid inputs: 422 - Existing user sign up: 422 - Sign up successfully: 201
POST /api/users/login Expects input body: { email: String, password: String }	- User does not exist: 401 - Password is incorrect: 401 - Login successfully: 200
Clothes Module	
GET /api/clothes/:userId	Get all clothes: 200 Get clothes by category: 200 Get clothes by invalid category: 404
GET /api/clothes/:userId/:clothingId	Get one clothing: 200
POST /api/clothes/:userId Expects input body: { category: String, color: String, seasons: String[], occasions: String[] }	Post one clothing: 201
DELETE /api/clothes/:userId/:clothingId	Delete one clothing: 200 Delete clothes but not specify clothingId: 404 Delete clothing that has been deleted: 404
PUT /api/clothes/:userId/:clothingId Expects input body: { category: String, color: String, seasons: String[], occasions: String[] }	Update one clothing: 200

Testing - Frontend

Login Test Case	**Expected Result**
	On start, check all text boxes and the login button are present on the screen

Enter “test@test.com” in the login email field	
Press “Login” button	Present a toast message “Please enter all fields”
Enter “123wrong” in the login password field	Present a toast message “Invalid record, please try again”
Enter “123123” in the login password filed	
Press “Login” button	Present a toast message“Login successfully”; navigate to the main screen

Outfit Test Case	**Expected Result**
	On start, check the outfits’ views are not present
Press “give me some outfit ideas” button	Presents several outfits on the screen; below each outfit, a “like it!” button and a “dislike” button are present and enabled
Press the first outfit’s “like it!” button	Presents a toast message “Your preference has been recorded”
Press the second outfit’s “dislike” button	Presents a snackbar with a message “We will not suggest this outfit anymore” and a “undo” button

(Add) Clothes Test Case	**Expected Result**
Press “+”	Navigate to the add clothes screen
Press “Save Clothes”	Present a toast message “You haven’t selected an image”
Press “+” in the new screen	Navigate to the gallery; present the selected image when come back to the add clothes screen
Select “shirt” in the category selector	Present “shirt” in the category selector and a toast message “shirt”
Select “white” in the color selector	Present “white” in the color selector and a

	toast message “white”
Press “Save Clothes”	Present “Missing clothes values” message
Check the “Spring” checkbox	Present a check mark in the “Spring” checkbox and a toast message “spring”
Select “home” in the occasion selector	Present “home” in the occasion selector and a toast message “home”
Enter “T-shirt” in the Name field	
Press “Save Clothes”	Present “Successfully add clothes!” message; navigate back to the add clothes screen with the selected clothes image present

Non-functional requirements

- Users should wait at most 2 seconds when requesting for an outfit suggestion.
    - Test the “/api/outfits/one” endpoint locally and calculate the average response time. The requirement is met if the time is less than or equal to 2 seconds. (We tested about 10 times and the average response time is around **420** milliseconds.)

- Users do not need more than 5 steps to add a cloth item.
    - Press every enabled button, selector and checkbox on the screen before saving a clothes item and count the number of steps. The requirement is met if the number is less than or equal to 5 and a user has successfully added a cloth item.