Macro Challenge – Plantara: Plant with Us

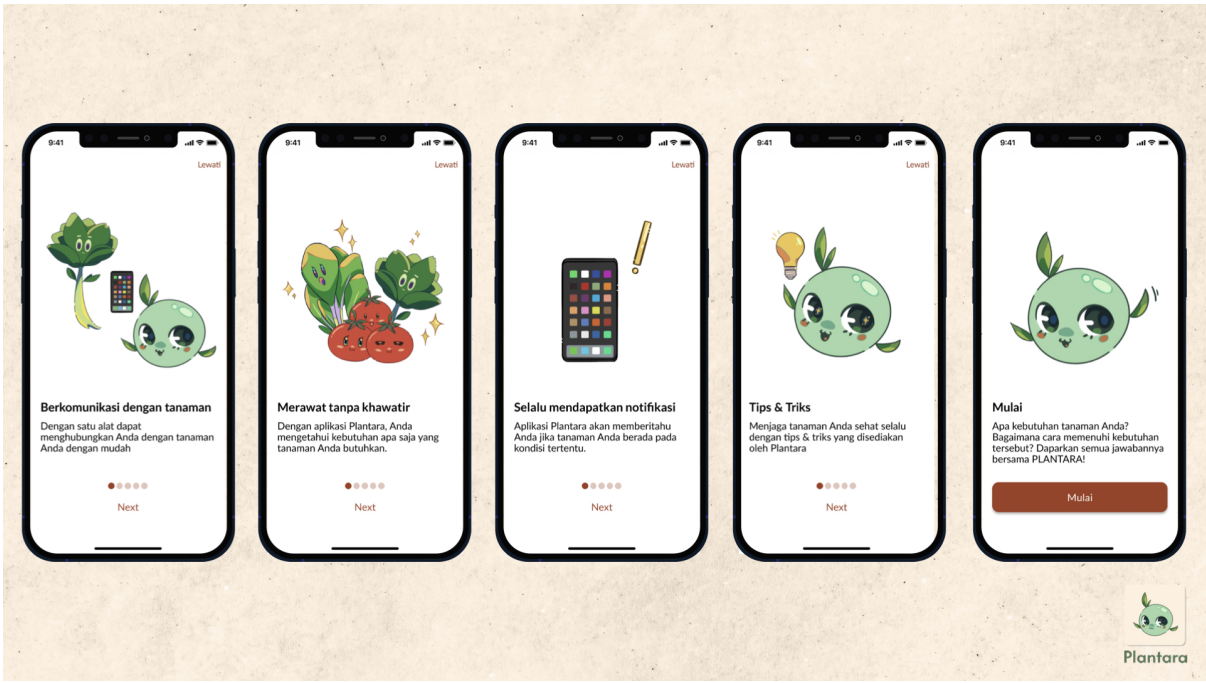
Technology	3D Modelling, Core Data, IOT, MVVM, Swift, SwiftUI, UserNotifications
Design	3D Modelling, Design System, Hi-Fi, Illustration, Lo-Fi, Mockup, Style Guide, Wireframe
Platform	iOS, iPadOS
Project Brief	Plantara is an IoT device that monitor essential plant need and displayed on an iOS Application
App Icon	
Challenge	Macro Challenge
App Category 1	Agriculture
App Category 2	Environment
Members	
Related to C4 Student's Challenge Database (Project Documentation 1)	

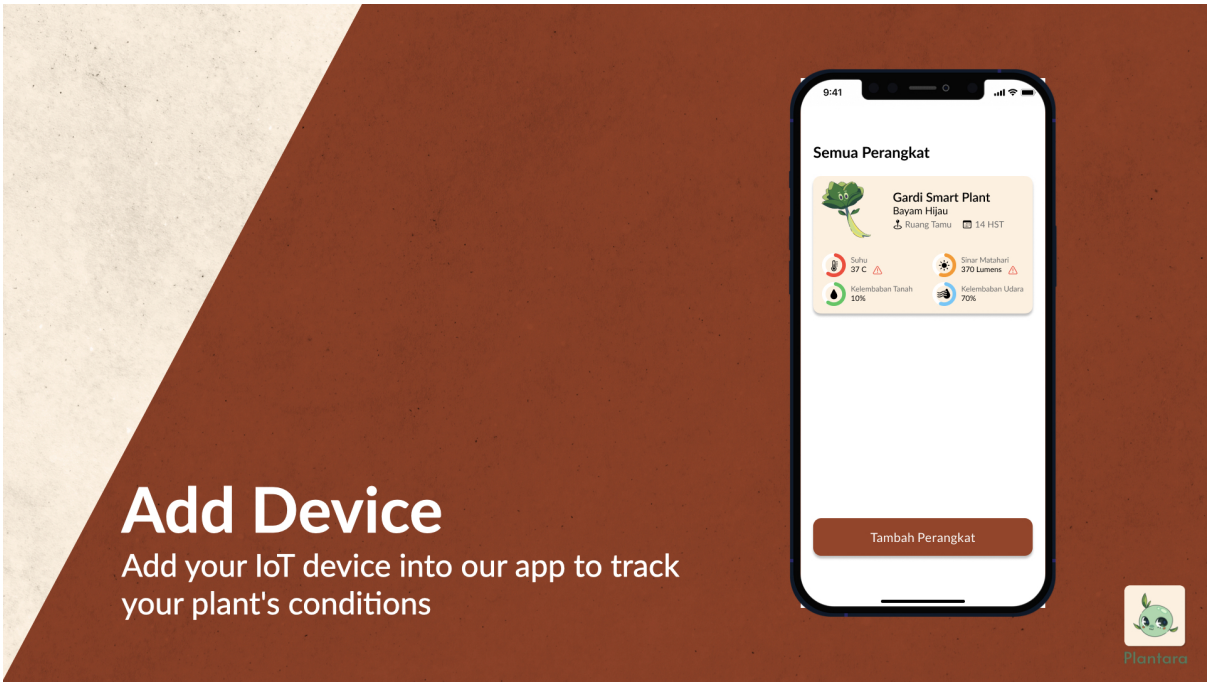
App Store Link	
Test Flight Link	https://testflight.apple.com/join/XL2oFU9j
Link Github	https://github.com/OidneR/Macro-Team4
↗ Related to C4 Academy Team-Role Experience (Project Documentation)	

About Plantara

Plantara is an iOS application that can help plants to communicate with user via IoT device. The IoT device will detect air humidity, soil moisture, light intensity, and temperature of the plant. With the sensor on the IoT, the system can provide notifications to the user when the plant is in need of something.

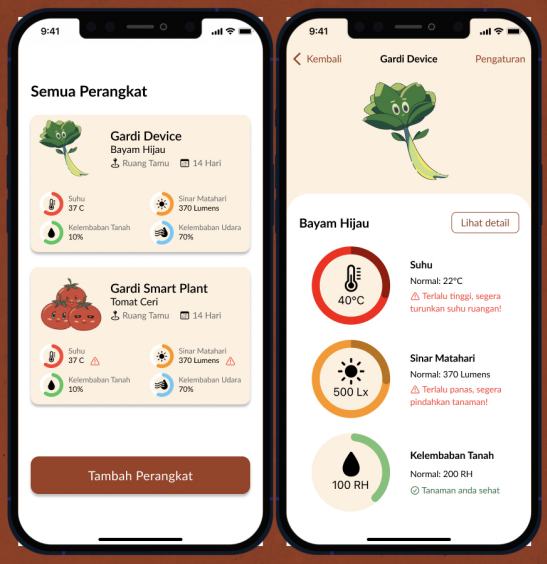
Overview


Lewati
Berkomunikasi dengan tanaman
Dengan satu alat dapat menghubungkan Anda dengan tanaman Anda dengan mudah
Next
Lewati
Merawat tanpa khawatir
Dengan aplikasi Plantara, Anda mengetahui kebutuhan apa saja yang tanaman Anda butuhkan.
Next
Lewati
Selalu mendapatkan notifikasi
Aplikasi Plantara akan memberitahu Anda jika tanaman Anda berada pada kondisi tertentu.
Next
Lewati
Tips & Triks
Menjaga tanaman Anda sehat selalu dengan tips & triks yang disediakan oleh Plantara
Next
Lewati
Mulai
Apa kebutuhan tanaman Anda? Bagaimana cara memenuhi kebutuhan tersebut? Daparkan semua jawabannya bersama PLANTARA!
Mulai
Plantara


Add Device
Add your IoT device into our app to track your plant's conditions
Semua Perangkat
Gardi Smart Plant
Bayam Hijau
Ruang Tamu
14 HST
Suhu
37 C
Sinar Matahari
370 Lumens
Kelembaban Tanah
10%
Kelembaban Udara
70%
Tambah Perangkat
Plantara

Plants' needs

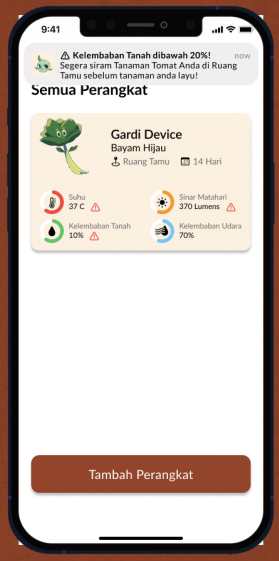
Track your plants' needs to keep it growing well and healthy



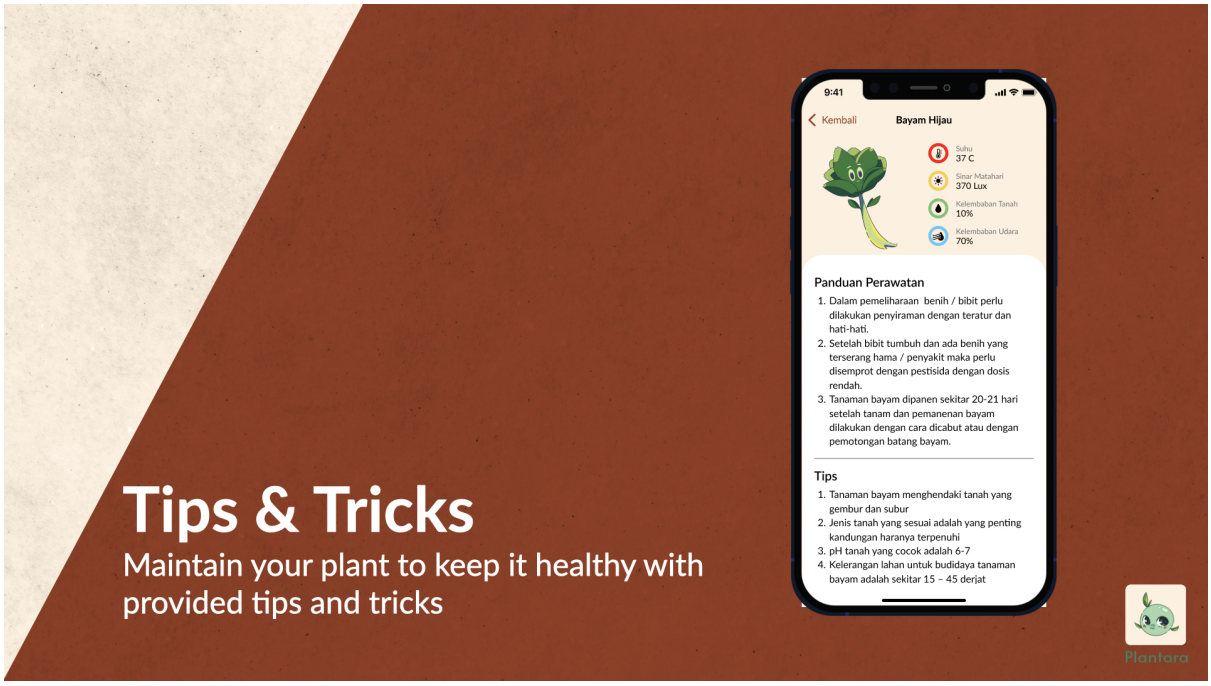
Plantara

Notifications

Always get notifications relating to your plant's conditions so you won't be missing out



Plantara



Design Journey

> Our design journey started from **Customer Journey Map → User Story → User Flow → Crazy 8 → Wireframe → Prototype → Usability Testing → Refine Prototype → Hand Off**

Style Guide

Colors

Brand Colors

Primary / #FFF0DF

Secondary / #9D3F22

State Colors

Sunlight	Air Humidity	Soil Moist	Temperature
#FF9500	#5AC8FA	#34C759	#FF3B30

State Colors

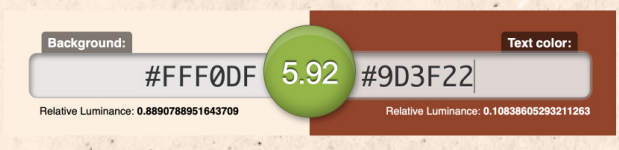
Sunlight	Air Humidity	Soil Moist	Temperature
#BF7000	#0573A5	#1A642D	#980800

Text Colors

Primary Text	Text Safe	Text Warning
#000000	#327047	#FF3B30

Text Colors

Text Setting deafult	Text plant detail	Text Empty Stage
#3C3C43	#757575	#8E8E93



Typography

Aa

Heading

Line height and paragraph spacing for heading is 1.1 x font size

Lato

Google Fonts

Name	Font size	Line Height
Heading 1	24 px	26.4 px
Heading 2 Bold	20 px	22 px
Heading 2 Semibold	20 px	22 px
Heading 2	20 px	22 px

Aa

Body

Line height and paragraph spacing for body text is 1.4 x font size

Lato

Google Fonts

Name	Font size	Line Height
Body Semibold Body	16 px	22.4 px
Small Text Bold Caption	14 px	19.6 px
Small Text Bold Caption Small	12 px	18 px

Iconography

Icon sets

SF Symbols

Buttons

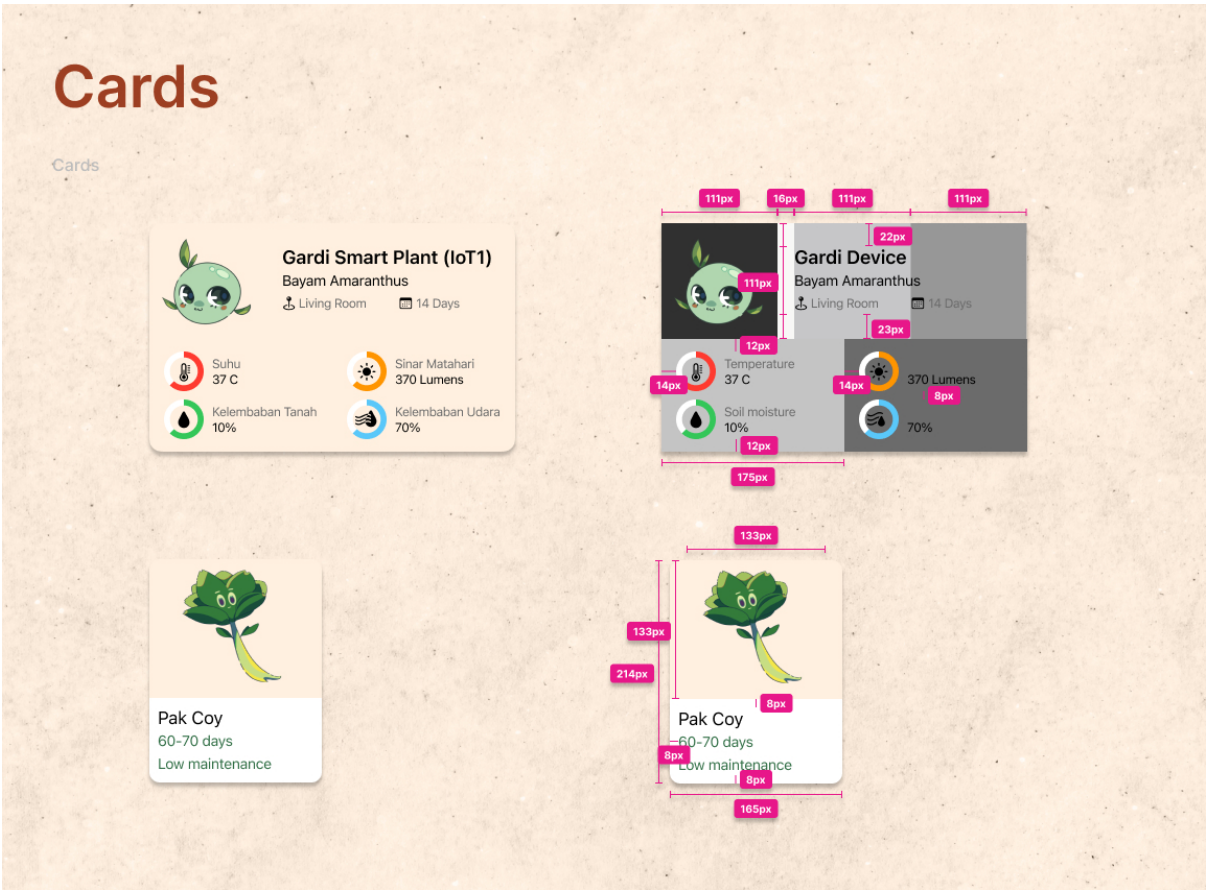
Color

Default

Button

Primary

Tambah Perangkat



Lo-Fi Prototype

https://www.figma.com/embed?embed_host=notion&url=https%3A%2F%2Fwww.figma.com%2Ffile%2FgoBoyQpdWgIhpepFr6FrXQ%2FDesign%3Fnode-id%3D1166%253A21142

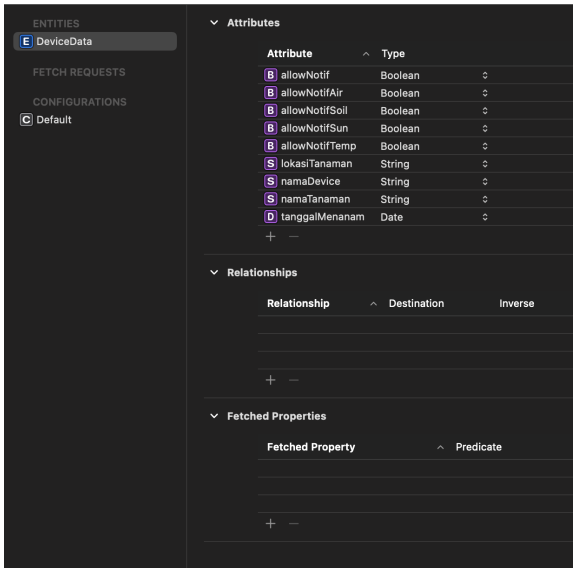
Hi-Fi Prototype

https://www.figma.com/embed?embed_host=notion&url=https%3A%2F%2Fwww.figma.com%2Ffile%2FgoBoyQpdWgIhpepFr6FrXQ%2FDesign%3Fnode-id%3D1166%253A21955

Hi-Fi IpadOs

https://www.figma.com/embed?embed_host=notion&url=https%3A%2F%2Fwww.figma.com%2Ffile%2FgoBoyQpdWgIhpepFr6FrXQ%2FDesign%3Fnode-id%3D1166%253A24842

Tech Corner

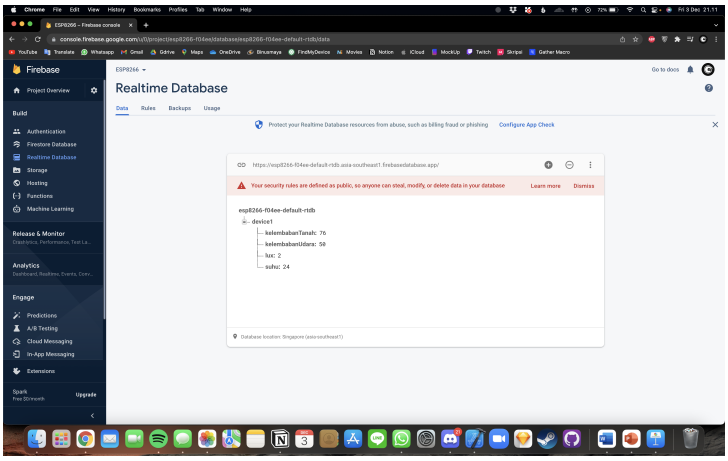


1. CoreData

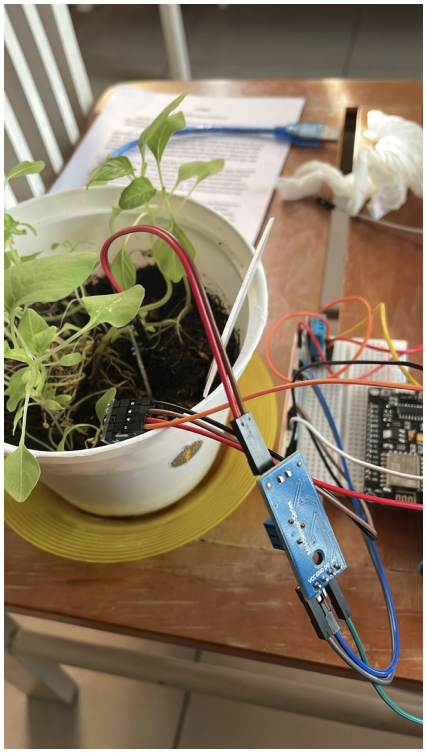
We use CoreData to store data from devices connected to the application. The data stored is the name of the device, the type of plant, the location of the device, and the data settings of the device.

2. Firebase

In the meantime, we are using Firebase as a third party to deliver realtime data from IoT into our application. We are still developing bluetooth to be able to receive realtime data from IoT.



3. Arduino



We use Arduino as the basis of our IoT and we connect Arduino with several sensors such as humidity, soil moisture, temperature, and lux (sunlight) sensors.

4. Notification

Because the application aims to help people find out what plants need at that time, we use notifications in order to make users can keep in touch with their plants so their plant's needs are met.

```
Toggle("Allow Notification", isOn: $allowAllNotif.animation(.easeInOut(duration: 2)))
    .onChange(of: allowAllNotif, perform: { newValue in
        notif()
        if allowAllNotif == true{
            if isNotDetermined == true{
                UNUserNotificationCenter.current().requestAuthorization(options: .alert) { settings, error in
                }
            }
        }else if allowAllNotif == true{
            if isNotDetermined == true{
                allowAllNotif = false
                isPresented = true
            }
        }

    })
    .alert(isPresented: $isDenied) {
        Alert(title: Text("Enable Notifications?"), message: Text("To use this feature you must enable notifications in
            settings"), primaryButton: .default(Text("Settings")){
            if let url = URL(string: UIApplication.openSettingsURLString) {
                if UIApplication.shared.canOpenURL(url) {
                    UIApplication.shared.open(url, options: [:], completionHandler: nil)
                }
            }
        }, secondaryButton: .cancel())
    }
```

5. SwiftUI

Because we want to try to develop an application using SwiftUI, so before development begins, we decided to use SwiftUI as the basis of our application interface. We find it easier to use SwiftUI because we can create components directly in the code and the interface is also displayed in realtime in the preview.

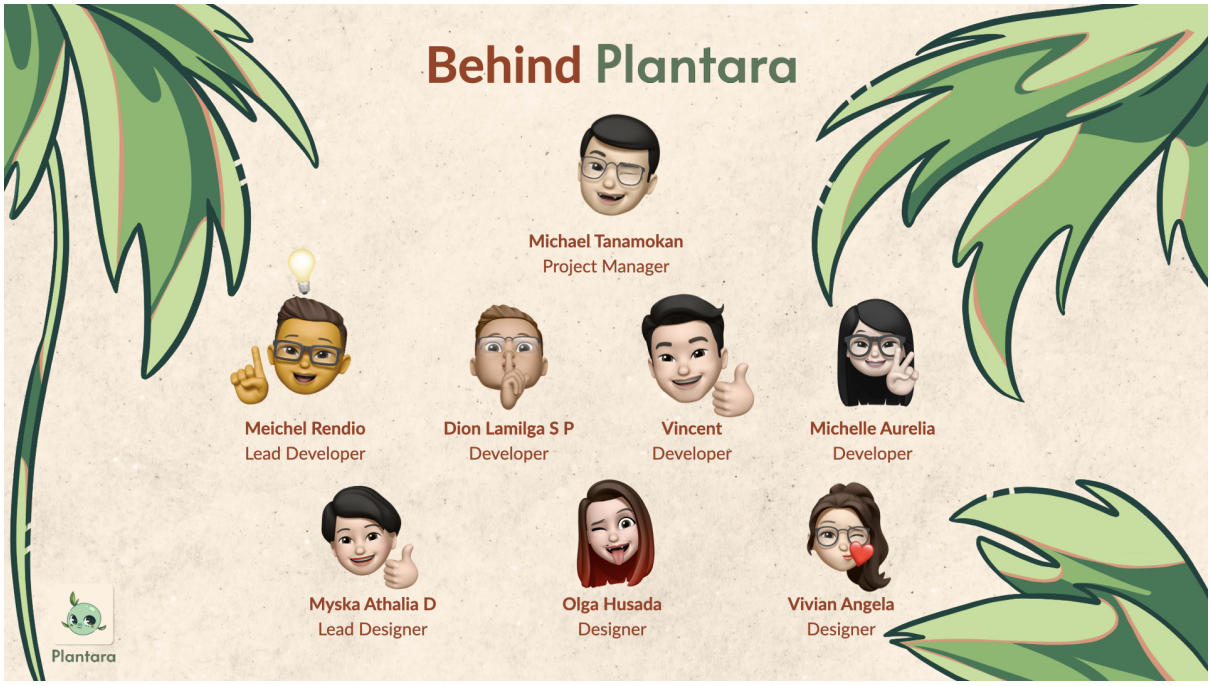
```swift
import Foundation
import SwiftUI

struct CardView: View{

    @Binding var viewModel: [Tanaman]
    @Binding var Grid : [Int]

    var body: some View{
        VStack{
            if !self.Grid.isEmpty{
                ScrollView(.vertical, showsIndicators: false) {
                    VStack(spacing : 25){
                        ForEach(self.Grid,id: \.self){i in

                            CardViewHStack(i: i, viewModel: viewModel, Grid: Grid)
                        }
                    }
                    .padding()
                }
            }
        }
    }
}

struct CardViewHStack: View{
    var i: Int
    var viewModel: [Tanaman]
    var Grid :[Int]
    var body: some View{
        HStack(spacing: 15){
            ForEach(i...i+1,id: \.self){j in
                VStack{
                    if j != viewModel.count{
                        CardViewReusable(dataTanaman: viewModel[j])
                    }
                }
            }
            if i == self.Grid.last && self.viewModel.count % 2 != 0 {
                Spacer(minLength: 0)
            }
        }
    }
}
```

☘ Project Documentation

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/a61d1629-24f1-49a4-99d1-91bb531d4bc8/Macro_-_Cross_Mentor_Feedback.pdf

☘ App Demo

☘ Team Behind Plantara


Behind Plantara
Michael Tanamokan
Project Manager
Meichel Rendio
Lead Developer
Dion Lamilga S P
Developer
Vincent
Developer
Michelle Aurelia
Developer
Myska Athalia D
Lead Designer
Olga Husada
Designer
Vivian Angela
Designer
Plantara