DESIGN DOCUMENT

Woody

Container Farm Project

Noah Groleau
Diana Karpeeva
Katchenin Cindy Coulibaly

Application Development III & Connected Object
Computer Science, John Abbott College
Youmna Badawy & Michael Haaf

April 8th, 2024

Function Overview

Woody is an app that turns basic container greenhouses into smart farms. With Woody, fleet owners can keep an eye on all their containers and do not need to worry about the security of their fleet. Meanwhile, farm technicians can easily manage the plants from outside the container, thanks to the various sensors monitoring and controlling the environmental conditions of the plants.

Design Overview

Screen Analysis

User Authentication

First page that the user is greeted with when entering the application. Allows the user to create an account or log into their already existing one. Once authentication is completed, they are redirected to the home page.

Settings

Allows the user to view and edit user profile information, as well as notification settings.

Home Page

Displays list of containers and their status. Each container item, when clicked, redirects the user to the container-specific dashboard page (where sensor information and controls are located). As well, displays a quick overview of upcoming tasks.

Task

Displays list of upcoming tasks and their information (deadline, category..) Users can filter tasks by three time conditions: due today, in the next week or in the next month. Users can also create new tasks from this page.

Dashboard

Contains a navigation bar at the top that allows the user to switch between overview and container-specific pages.

Overview

Displays a list of container cards, which contains their overall environmental conditions (temperature, humidity and soil moisture) and their lock status. If the user is a fleet owner, they can create new containers and edit already existing ones from this page.

Container Specific

Displays more in-depth information about the environmental and security conditions of the container. The security conditions are displayed using a chart that compares the latest readings of the security sensors (past twelve hours)

At the bottom, there is a card where users can remotely control the fan, lights, lock and more of the container.

Map

This page is only available to the fleet owner. Displays a list of container cards that when expanded, show a map with the container location and information, such as coordinates and container angles (pitch, yaw and roll).

Analytics

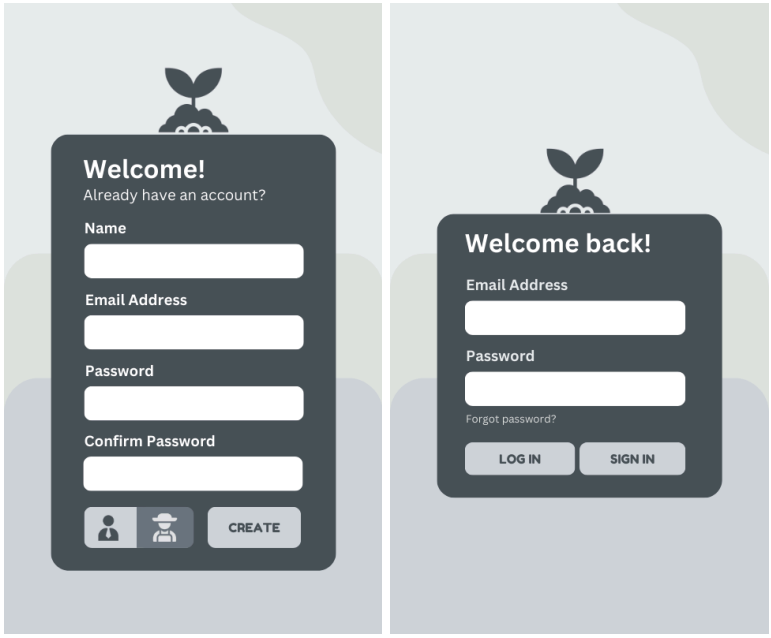
This page is only available to the fleet owner. Displays a series of graphs comparing readings of the containers' environmental conditions versus set thresholds. As well, displays a list of past security breaches.

About Us

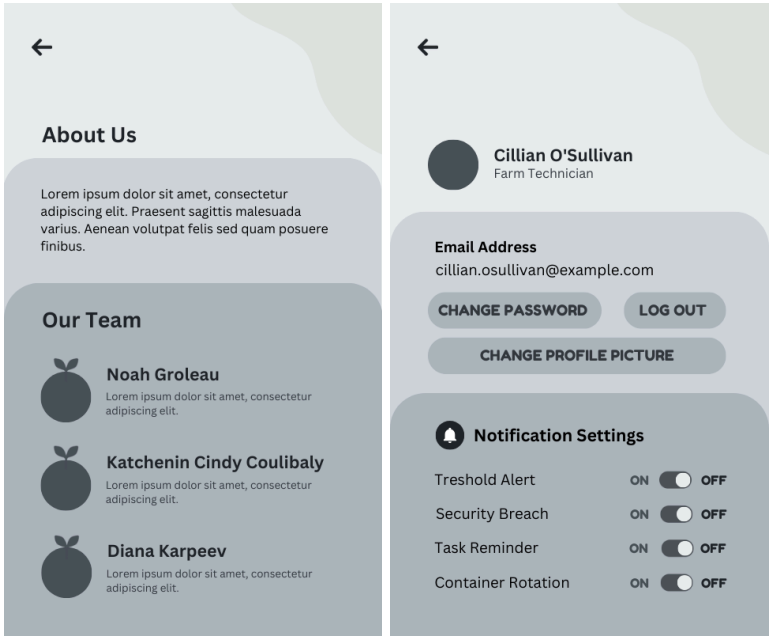
Displays short descriptions of our goals and team members.

App Prototype

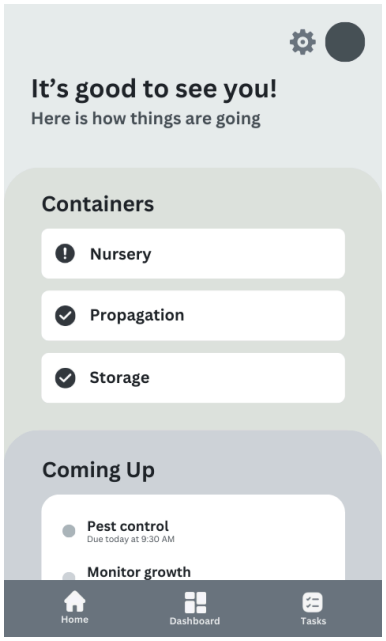
User Authentication Pages

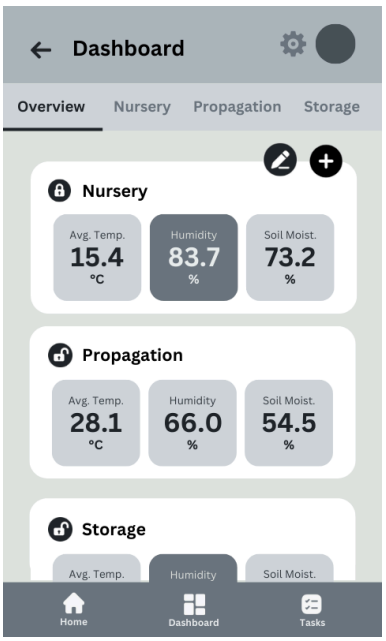


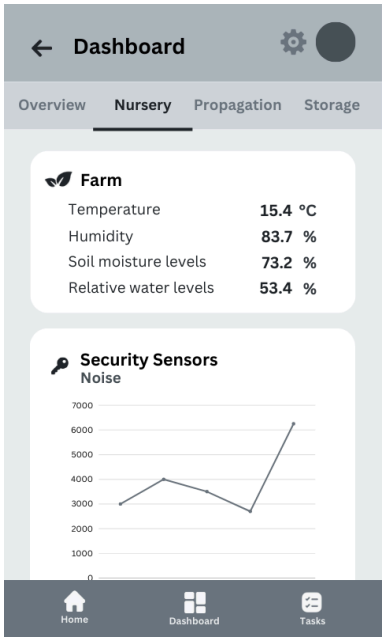
About Us and User Settings

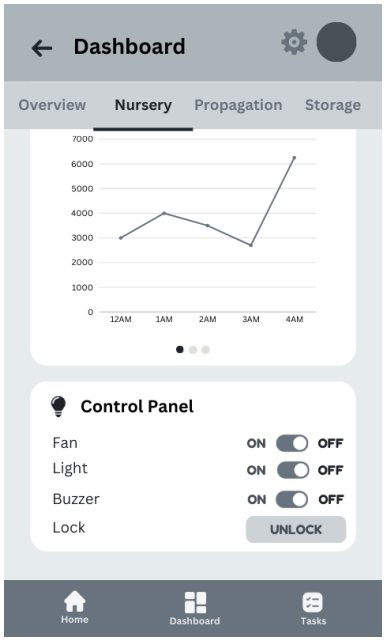


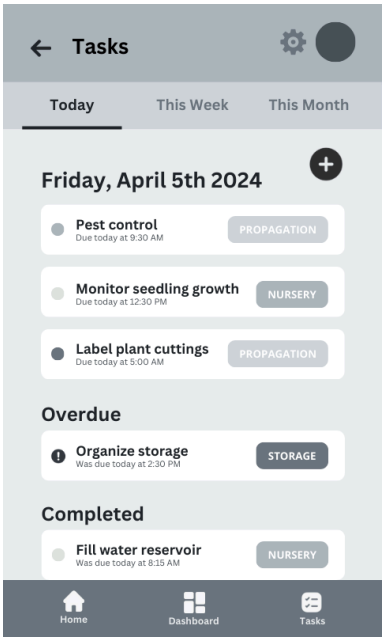
Farm Technician View











Fleet Owner View

App Features

High Priority

General

- User authentication
- Save user preferences
- Notifications on certain events (ie. thresholds breached, security alerts)
- Tab bar to easily navigate between pages
- Role-specific views and functionalities
- Display list of containers
- Ability to control container security conditions
- Display container security conditions
- Data representation by graphs:
    - View and track data changes over time about the different subsystems.
    - View and track data in an intuitive way
    - Compare data with set thresholds

Farm Technician Specific

- Ability to control container environmental conditions
- Display container environmental conditions
- Saves user defined thresholds for monitored data (ex: min/max temperatures)

Fleet Owner Specific

- Display container geolocation
- Ability to create or delete containers
- Ability to modify individual containers → only owner
- Provide a view on historical security highlights and changes over a given span of time.

Medium Priority

General

- Ability to modify user profile information
- Ability to customize thresholds
- Notifications are specific to the user's credentials
- Accessibility and responsiveness
- Networking: robust mechanisms in case of network disruptions

Low Priority

General

- Tasks notify user when approaching deadline
- About us page with help & support info
- Geolocation page displays a visualization of the container
- Ability to set tasks from a Tasks Page

Potential Showstoppers and Open ended questions

- Should the technician be able to control the color of the light?
- Should the technician be able to control the pulse duration of the light?

Cloud Infrastructure

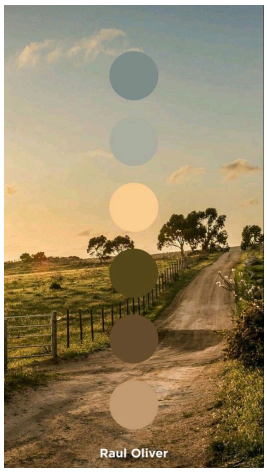
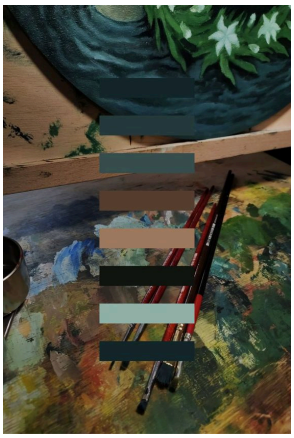
Possible steps:

To connect Raspberry Pi scripts to a mobile app using Microsoft Azure as the cloud infrastructure and IoT gateway, you will need to follow these steps:

1. Set Up Azure IoT Hub and Device Provisioning Service (DPS):
    - Create an Azure IoT Hub to manage device connections and data routing.

    - Set up an Azure IoT Hub DPS to simplify device provisioning and securely connect devices to the IoT Hub.
    - Link the DPS to the IoT Hub to enable devices to be automatically registered and assigned to the IoT Hub upon provisioning.
2. Configure the Raspberry Pi as an IoT Device:
    - Use the Azure IoT Hub SDK for Python on the Raspberry Pi to connect to the IoT Hub.
    - Implement the necessary scripts on the Raspberry Pi to send data to the IoT Hub.
3. Develop the Mobile App with .NET MAUI:
    - Use Visual Studio 2022 with the necessary workloads for ASP.NET, Azure, and .NET MAUI development.
    - Create a .NET MAUI app that will serve as the user interface for both Android and iOS devices.
    - Integrate the Azure Mobile Apps backend to connect the mobile app to the Azure infrastructure.
4. Implement Cloud-to-Device (C2D) Communication:
    - Utilize Azure IoT Hub's capabilities for C2D communication to send commands or data from the cloud to the Raspberry Pi.
    - Implement features such as Direct Methods, Twin Desired Properties, and File Upload as needed for your application.
5. Secure the Connection:
    - Ensure secure communication between the Raspberry Pi and the mobile app through the Azure IoT Hub.
    - Use Azure Key Vault for managing secrets and connection strings securely.
6. Testing and Deployment:
    - Test the integration of the Raspberry Pi with the mobile app using tools like Azure IoT Explorer.
    - Deploy the mobile app to app stores and ensure it can communicate with the Raspberry Pi through the Azure IoT Hub.
7. Monitoring and Management:

- Utilize Azure Monitor and Azure IoT Hub's monitoring features to keep track of device health and data flow.
- Implement device management features as needed, such as updating firmware or changing configurations.

8. Scalability and Extensibility:
    - Design your solution to be scalable, considering the use of Azure services like Azure Functions, Azure Stream Analytics, and Azure Cosmos DB for data processing and storage.
    - Plan for extensibility, allowing for the addition of new features or devices as your IoT solution grows.

Colour theme Ideas