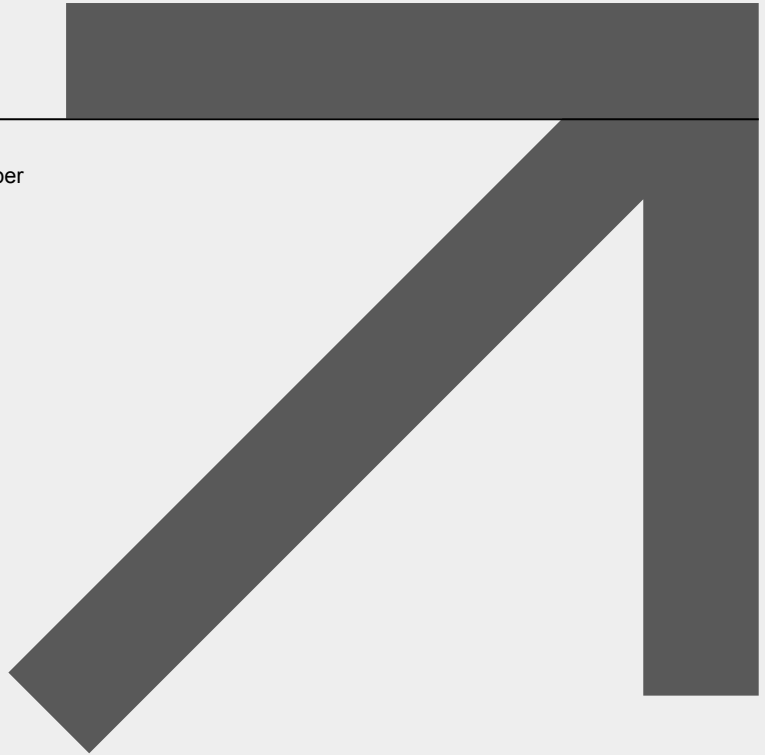
# RMD

Dixon Snider Phil Akagu-Jones The Vi Phung Vu Hai Nam Nguyen Naut Seper



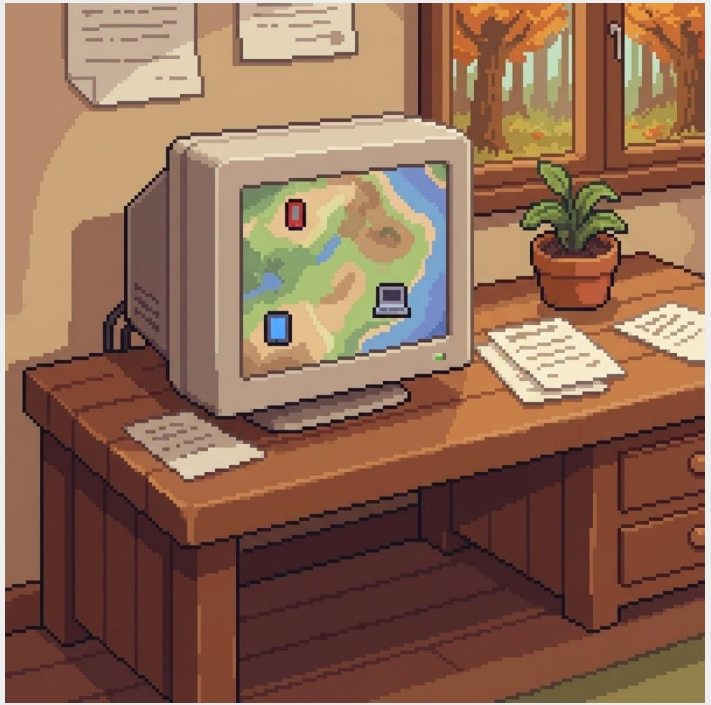
# The Idea

Ring My Device is an Android app that listens for authenticated commands over SMS & Meshtastic

It can:

- Send back location
- Ring loudly
- Take and upload photos
- Lock or wipe the device
- Toggle location services

You can also self-host a lightweight server to see all your gear on a map



# Why Pursue RMD?

1. Current Tools Fail When You Need Them Most
    - Most recovery systems stop working the moment a phone loses internet or disconnects from a Google account
    - In real situations, devices are often offline, out of data, on limited SIM plans, or intentionally disconnected
    - By using channels that almost always remain available, SMS and local mesh radio, we remove the weakest link in traditional solutions
2. More Privacy, More Ownership
    - Many users want phone recovery without giving Google continuous location access or relying on cloud services
    - A self-hosted option provides control, transparency, and long term independence from external platforms
    - It also allows users to manage not just phones but all their gear on their own map server
3. A Valuable Technical Challenge
    - The project brings together secure command handling, Android permissions, offline communication, mapping, logging, and optional server hosting
    - It’s a practical, real world engineering problem with high impact
    - By enabling recovery anywhere, with or without the internet, the system becomes more resilient, private, and useful than existing “find my device” tools

# Challenges + Scope

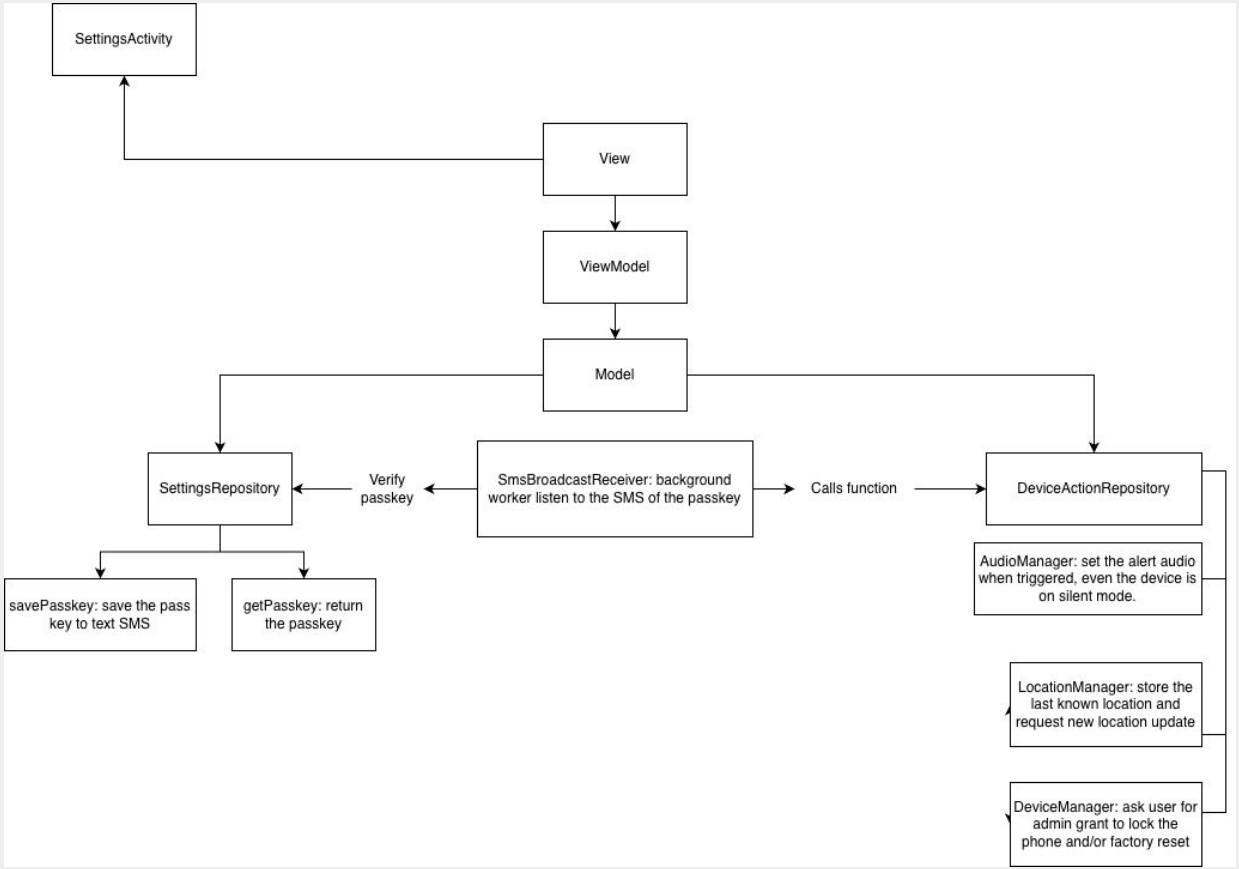
## Technical Challenges

- Building a system that stays reliable across different channels
- Designing a server-client flow that works with many limitations
- Implementing device admin features without exposing the device to abuse
- Navigating Android’s permission model and background execution limits

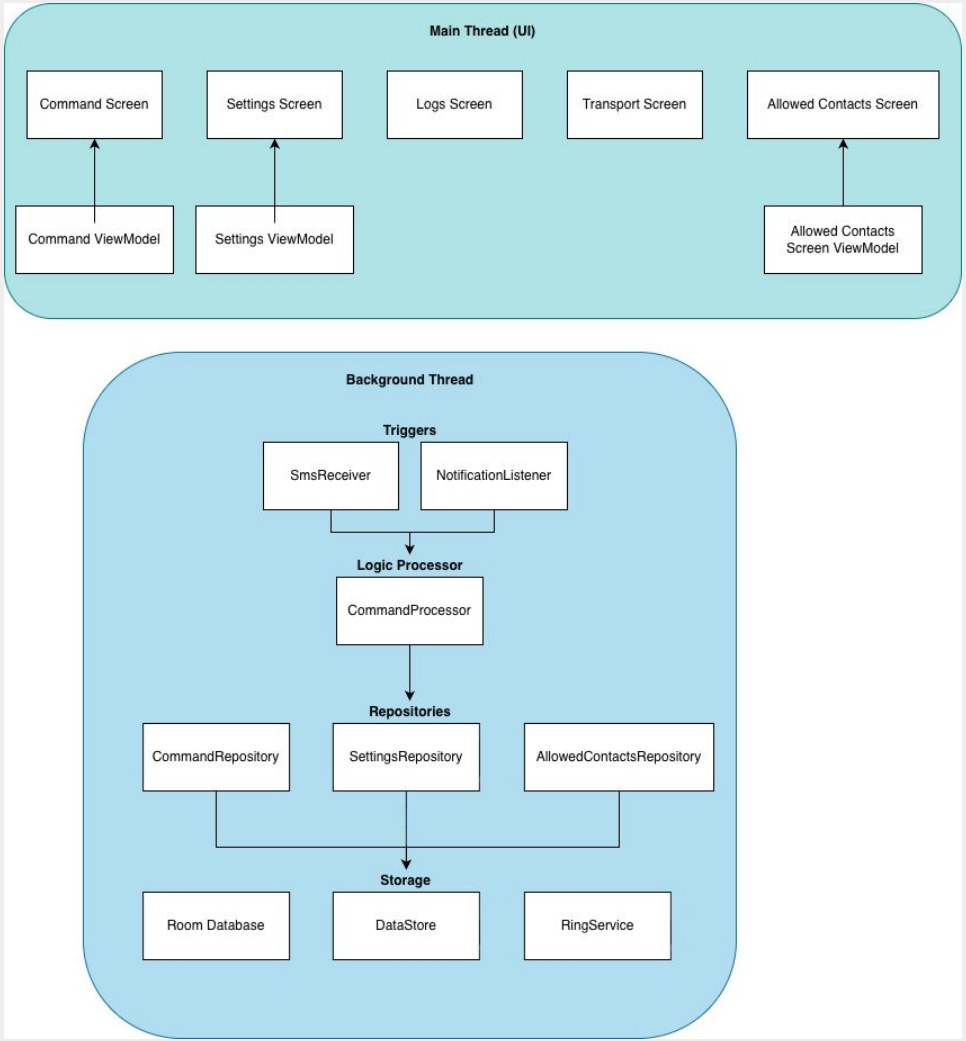
## Project Scope

- The project breaks naturally into clear components
- The system was large enough to be challenging but still manageable for our team size
- Each member was able to contribute according to their strengths while learning new skills

# MVVM Diagram



# Threads Diagram



# The Demo

# Project Contributions

## Naut

- Led core development across both the server and Android app
- Built the self hosted RMD server in Go
- Implemented full Android command coverage
- Expanded the portal to trigger commands from the browser

## Dixon Snider

- Built the early Android app foundation and initial settings
- Implemented OpenStreetMap functionality, demo controls, and full map integration
- Refactored MainActivity, reorganized project structure, and managed early merges

## Nam Nguyen

- Implemented persisted settings storage for stable configuration across sessions
- Built the test ring feature for fast internal validation and improved reliability

## Phil

- Created early UI pages: General Settings, Find My Device Settings, and Appearance Settings
- Implemented alternate versions of the settings import/export feature and the camera capture/upload

## Vi

- Designed early Figma demos that shaped app’s layout
- Implemented the secure remote device wipe feature (factory reset)

# Lessons Learned

- Building a self-hosted remote device management system gives users full control, avoids dependence on commercial services, and creates a flexible base for future features
- We gained hands on experience with backend and frontend integration, mobile security, and real communication protocols used in production systems
- Coordinating features early and documenting decisions clearly is essential
- Complex systems can behave unpredictably when many modules interact
- Testing across different device states and communication channels is crucial for reliability
- Overall, the project strengthened our teamwork, problem-solving, and understanding of what it takes to deliver a true end-to-end system