Description Summary for CS440 Group 22 for Development Project MAPAW

Group Members: Yasmin Sawaf, Rajashree Dahal, Daniel Barajas, Fey Ogutuga

Project Overview: MAPAW is a community-driven wildlife tracking platform that allows users to share and discuss animal sightings through an interactive map. Users can upload images and pinpoint locations, creating a dynamic database of species distribution. The platform fosters collaboration among wildlife enthusiasts, researchers, and conservationists, enabling species identification, local sightings, and habitat insights. MAPAW enhances wildlife research and conservation awareness by making sightings shareable, supporting scientific research, and promoting public engagement in biodiversity monitoring.

Purpose of the Project: MAPAW serves animal enthusiasts, conservationists, and researchers, enabling them to track and analyze animal movements for conservation, research, and awareness. It centralizes data on animal locations, addressing gaps in wildlife tracking, identification, and awareness. The project aims to enhance animal tracking, conservation, and adoption by providing a centralized platform. Its goal is to improve conservation, raise public awareness, and promote informed human-animal interactions.

Scope of the Work: The scope of work includes several events such as creating an account, logging animal sightings, uploading media, filtering sightings, joining discussion forum, rate animal behavior, start navigation to sighting location.

Scope of the Product: MAPAW features interactive maps, a system for logging sightings, and a forum for user interaction. Creating a pin, searching for a pin, using the map view, participating in a forum, verifying wildlife data, and using wildlife data are some of the product scenarios.

Stakeholders: The stakeholders involved are researchers & conservationalists, wildlife organizations, users/contributors, developers and technical meam.

Requirements: The application should allow users to explore animal sightings in their area and share them with the public. The application also allows for special report cases, such as reporting a lost, rare, or dangerous animal. The application allows users to report rare animal sightings by submitting detailed information such as species, location, time, and, optionally, a photo. The sighting should be flagged as rare and sent to conservation authorities. The overall system must support at least 100 concurrent users while maintaining performance. The pinned sightings must be within ± 10 meters of the actual location. Likewise, species identification should be at least 90% accurate. In addition, the application should prevent duplicate sightings within 50 meters and 5 minutes to maintain the data integrity. The system should be able to maintain a 99.5% uptime. It should handle 500 concurrent users at launch, scaling to 50,000 users within a timeframe of three years without any performance degradation. The application must be adaptable across different devices, ensuring accessibility for all users. Likewise, encryption and compliance with GDPR and CCPA will protect sensitive user data.

Databases and background information should be mostly concealed from users. Users should only be able to enter files into the database using the in-app camera to prevent uploading malicious files and capture the exact location of the picture that was taken. The application should be designed for simplicity and ease of use to accommodate a wide range of users. The system should be integrated with mapping APIs and cloud storage, which is very important for real-time data exchange. The app should be accessible in both Google Play Store and Apple App

Description Summary for CS440 Group 22 for Development Project MAPAW

Group Members: Yasmin Sawaf, Rajashree Dahal, Daniel Barajas, Fey Ogutuga

Store. The major update should be scheduled every 4 months, and minor updates will be carried out every two months.

Design Goals: MAPAW is built with clean, well-documented code that's easy to maintain and expand without disrupting existing features. It is fully accessible, supporting users of all abilities through clear visuals, screen reader compatibility, and best practices. Security, privacy, and scalability are prioritized to protect user data and ensure the app grows smoothly as more users and features are added.

System Architecture: The frontend is developed using Flutter. The frontend is developed using Flutter. The users interact with the app through mobile or web interfaces that support core features like wildlife sightings, maps, and community forums. The frontend communicates directly with the backend and external APIs to provide real-time data and interactivity. The backend is powered by Firebase, offering services such as: authentication for user login and account management, firestone for real-time database operations, cloud storage for managing media, and cloud functions for backend automation. Google Maps API will provide location services and mapping functionality for sites and events. AI animal identification API supports image recognition features, helping users identify species based on uploaded photos.

Key Design Considerations: MAPAW is a mobile/web platform built with Flutter. It uses device features like the camera for photo submissions and GPS for geolocation tagging. All core data—including sightings, forum content, and user profiles—is stored in Firebase Firestore. Media files (images/videos) are stored separately in Firebase Storage. Offline access is supported, with changes automatically synced when the user reconnects to the internet. Firebase Authentication handles login using OAuth2 methods (e.g. Google, email). Data access is controlled through role-based Firebase Security Rules, ensuring that users only access what they can. All data is encrypted via HTTPS to protect user privacy. The UI is gesture-friendly, with features like an interactive map, animal sighting forms, species cards, and a community forum. Real-time updates are handled through Firebase listeners, and navigation is supported through tab and drawer menus. Composite design pattern is used. The Pin class composes multiple Reply, Tag, and AnimalCard objects — forming a "whole-part" hierarchy. Pin can be treated as a single object, but it contains many Reply objects and Tag objects. These nested objects (Replies, Tags) are part of the structure, and operations like displaying a pin will naturally include its replies and tags — this reflects the Composite pattern's intent. The app structurally allows us to treat both individual and grouped components (e.g., a Pin vs. a Pin + its Replies + Tags) the same way in logic or UI. Suppose the developers choose to implement a live-updating map view (i.e., the UI automatically updates when a new pin is added or a filter is applied). In that case, we are also using principles from the Observer pattern: The MapView class would act as an Observer of the Pin list (the Subject). When the list of Pins changes, MapView gets notified and re-renders.

In conclusion, MAPAW is inspired by social media and it helps build meaningful connections with other enthusiasts and aims for the share of knowledge in the community.