Module Name	Description	Purpose
User Interface (UI)	Develop the UI using BeeWare (Toga) to provide a native-looking app experience.	Allows users to interact with the app and manage cameras, view events, configure settings.
Database Integration	Use PostgreSQL as the central database for all application needs.	Store user data, classification results, training and testing datasets, and historical video data. Ensure data integrity and scalability in a multi-user environment.
Backend Development	Use Flask (or Django) to create RESTful APIs for app functionality.	Handle user authentication, data access, and classification requests.
Image & Video Processing	Integrate Pillow, OpenCV, and YOLO for image preprocessing, video handling, and object detection.	Detect and classify animals in video feeds and trigger alerts based on detections.
Camera & Device Integration	Set up Bluetooth/WiFi integration using PyBluez or PyWiFi for camera communication.	Enable communication with trail cameras even without network access.
Alerts and Notifications	Use Firebase Cloud Messaging (FCM) to implement real-time user notifications.	Notify users when animals are detected in video feeds by the model.
Training & Testing Dataset Management	Implement tables for managing training, testing datasets, and results.	Store labeled datasets for model training and evaluate model accuracy with test data.
Classification Results	Implement separate tables for training/testing results and real-world deployed app results.	Differentiate between experimental model evaluations and live, deployed results.
Training Metrics (Optional)	Store metrics such as loss, accuracy, and validation performance during training epochs.	Monitor model training progress and identify issues such as overfitting.