Title: Security and Reliability Enhancements High Level Design.

Author : Mohamed Akthar K

2.1. Introduction

This report describes a professional High Level Design (HLD) to make an existing cycle-parking system more secure and more reliable. The improvements are in terms of three main areas:

Cycle insurance: optional insurance cover and claim payment on cycling cycles that are parked.

Theft tracking system: Tracking, reporting and providing collaboration with local law enforcement.

Identity verification: vigorous identity proving and verification of parkers to minimize fraud as well as facilitate investigations.

Audience: product managers, solution architecture, front end+ backend engineers, QA, and operations.

Assumptions: There is also a central cyclic parking system with user accounts, parking spaces, payment and alerts.

It can add integrations (third party KYC, police APIs, insurer APIs).

2.2. High Level System Overview

Components:

- Mobile/Web App (User): identity capture, purchase of insurance UI, incident reporting UI.
- Operator Portal: tracks incidents, grants claims, coordinates with the law enforcement.

- Parking Manager Service: basic ticket bookings, slot states, locking hardware control.
- Identity Service: KYC onboarding, OTP, biometric (option), document.

2.3. UML Diagrams

- Sequence Diagram (Report Theft): User -Mobile App-> Theft Tracking - Incident Manager- Law Enforcement Gateway- Police.

- Class Diagram: User and ParkingSession classes, Event and Incident classes, InsurancePolicy and EvidenceBundle classes.

2.4. Technical Architecture

- Front-End: React/Angular (not needed to demonstrate).
- Backend: Node.js (Express / NestJS) or Java (Spring Boot) — an example code is provided below usage of Node.js.
- DB: julius for relational information, MongoDB missing versatile documents, Kafka missing events.
- Identity/KYC: 3rd party KYC (Onfido, IDfy, Socure, etc.).
- Authentication: Keycloak / Auth0 / bespoke OIDC.
- Monitoring: Prometheus + Grafana; ELK stack of logs.

2.5. APIs and Integration Points

List of essential APIs (brief):

- POST/api/v1/identity/verify- submit IDdocs and selfie- returns verificationId.
- GET /v1/identity/id/status - retrieve status of verification.
- POST /api/v1/events -ingest sensor/CCTV event.

- POST/api/v1/incidents a create incident (typically after correlation, internal).
- GET /api/v1/incidents/id/ — retrieve incident information.
- POST /api/v1/ purchase/insure — purchase insurance.
- POST /api/v1/law-enforcement/report — push police report (signed bundle).

2.6. References

- **Draw.io for diagrams URL Link:-**

  (https://viewer.diagrams.net/?tags=%7B%7DClightbox=1Chighlight=0000ffCedit=_blankClayers=1Cnav=1Cdark=auto#G1xgUNrvk2L_7uXfzsGDCUzE-AZNm5alv7).