Dependability and Failure Analysis of the Najm App

Course: CPIT-455

Student Name: Abdullah Mubarak Aljadani

ID: 2342239

Selected System: Najm Application (Traffic Accident Management System)

1 .System Context

Classification: Socio-Technical System.

The Najm application is a critical platform connecting vehicle owners, liability investigators, and insurance companies. It facilitates the reporting and management of minor traffic accidents in real-time. The system relies heavily on the interaction between technology (the app/GPS), processes (insurance/legal regulations), and people (drivers under stress).

2 .Critical Dependability Attributes

To ensure the system functions correctly within its environment, the following attributes are prioritized:

Availability: The system must be accessible at any moment to report an incident. Unavailability leads to traffic congestion and delays in legal procedures.

Reliability: The system must accurately capture and transmit GPS coordinates and accident photos. Reliability is crucial for evidence preservation; corrupted data could lead to incorrect liability assignment.

Security:The application processes highly sensitive Personal Identifiable Information (National IDs, license plates, insurance policies). Protecting this data is a strict legal and ethical requirement.

3 .Analysis of Failure Sources

The potential risks to the Najm system are classified into three categories:

Operational Failure (Primary Risk):

Context: Human error is the most significant threat. Drivers involved in accidents are often panicked or stressed.

Scenario: A user might input incorrect vehicle data or fail to follow the photography guidelines due to stress, leading to a "process failure" even if the software works perfectly.

Environmental / Hardware Failure:

Context: The physical environment where accidents happen.

Scenario: Poor network connectivity (4G/5G) at the accident site or hardware limitations (e.g., a user's phone with a low-quality camera or dying battery) preventing data upload.

Software Failure:

Context: Logic or coding errors.

Scenario: The application crashing during high-load periods, such as during heavy rainstorms when accident rates spike significantly

.

4 .Proposed Dependability Strategy

To mitigate the identified risks, the following strategies are recommended:

Fault Tolerance (Addressing Environmental Risks):

Implementation: Develop a robust "Offline Mode." If the network signal is lost, the app should locally cache the photos and incident report. Once connectivity is restored, the system automatically synchronizes the data. This ensures the service continues despite environmental failure.

Fault Prevention (Addressing Operational Risks):

Implementation: Design a "Stress-Resistant" User Interface (UI). The interface should be simplified with large buttons and automated data entry (e.g., scanning the license plate rather than typing) to minimize human error during stressful situations.

Verification & Validation (Addressing Software Risks):

Implementation: Conduct Stress and Scalability Testing. The system should be rigorously tested under simulated "surge" conditions (e.g., 10x normal traffic) to ensure the server infrastructure can handle peak loads without crashing.

References

Course Materials: Class 1 Slides - "Domain Spine: Dependable Systems", HIMMA Curriculum.

AI Assistance: System analysis, attribute prioritization, and brainstorming assisted by Gemini AI.