Exam - Analysis and Verification of Software

1 Context

Thanks to your large experience managing large commercial software projects in the industry, and your knowledge of the latest techniques in software testing, you've been designated leader of the Aupark (automatic parking) system project. The Aupark system is already under development, yet, during the past six months, the project has not been able to meet the quality required to be made available to the public. Your duty is to make sure the project reaches production on time, and with as less failures as possible.

2 Scenario 1: The Aupark System

Aupark is a parking system that has as objective to be fully automated and has introduced a novel ecological feature: it requires no paper tickets. Each Aupark system installation controls a parking lot. A parking lot contains many parking spots. Each parking spot is controlled by sensors to know if it is occupied or not and contains a green/red light to indicate visually if a spot is free or taken respectively. Parking spots are organized in lanes. Lanes are organized in floors.

Each parking has outside its entrance big led panels that show: how many parking spots are there in total, how many are free, and what is the price per hour of parking. Entrances and exits to the parking are controlled by an automatic barrier. When a car appears at an entrance, Aupark takes a picture of its plate, recognizes the plate with an OCR system (online character recognition), opens the barrier, and records the plate and the time at which the car went in. Car drivers leave their car at a parking spot and leave through a pedestrian exit. When drivers exit the parking, they have two options:

Pay at a toll. A driver goes to a payment toll, introduces the car's plate into the machine, and proceeds to pay. Then the driver drives the car to the exit, Aupark takes another picture of the car's plate, and if recognized as *already paid* the barrier opens automatically without any interaction.

Pay at the exit. A driver drives his car to the exit, stops before the barrier, and pays at a toll next to the barrier without going out of their car. The barrier opens if the payment is successful.

2.1 Details - Responsibilities

Aupark relies, as specified above, on many different components:

- The OCR library uses state-of-the-art machine learning techniques and is developed by an internal team in Aupark under your supervision.
- All hardware (*i.e.,* led panels, led lights, barriers, high-definition cameras) is produced and guaranteed by a third-party company named Auhard. It is however Aupark's responsibility to properly install and connect all components.
- All payments are performed through direct interaction with an external payment system provided by a third-party company.

2.2 Your Task

You must produce a testing plan for Aupark. The testing plan will ensure that the system's quality is enforced and that few problems are found when the real system is deployed. The company's objective is to install 500 different parking lots using Aupark in the upcoming months and make it 5000 during the next year. The expected plan should include *at least*:

- A list of priorities including what are the most important components to test (*i.e.,* the ones that count the most for the project), and what are the most critical (*i.e.,* the ones that may present the most problems).
- A checklist of tests for those critical and important components.
- A strategy to test at least two of the following: the OCR system and its interactions, the payments, the hardware interactions, and the production environment on each parking installation.

3 Scenario 2: The Aupark Management Console

All Aupark parking lots are managed in Aupark headquarters, located in building M5 building of Cité Scientifique, Villeneuve d'Ascq. The Aupark management console centralizes information from all parking lots and provides *the MANAGER* with relevant information in form of dashboards and reports so she takes decisions. Originally, all data was stored in a hidden server room in building M5, but with the change in management, it has been decided to move all of it to the cloud.

The Aupark management console allows *the MANAGER* to see and control the following information:

See Statistics. Several statistics in the shape of plots and graphs. For example, average parking occupation at a national level, per parking lot, per city, per department, per region.

Receive Broken Hardware Alerts. Alerts indicate when something unexpected happens (*e.g.,* broken barriers) so a technician can be sent to make the corresponding repairs.

Change prices. Prices can be adjusted centrally and impact any parking lot.

All the above are shown using dashboards implemented by a team under your supervision. The dashboards include historical plots of data (bar charts, line charts) and live data coming directly from the different parking lots.

3.1 Details - Responsibilities

The existing system is currently deployed in a custom hidden server in building M5. This application is made of three main components:

- A web server implementing a REST API that receives and stores different notifications from all Aupark parkings. Notifications represent events such as broken hardware alerts, incoming and outgoing cars, and payments. All information is stored in a relational database.
- A Big Data batch process that processes all incoming information once a day and produces daily statistics in a separate database.
- A streaming service streams incoming data to the live dashboard UI. Streaming throttle (events streamed per second) is configurable to reduce resource consumption.

3.2 Your Task

You must produce a testing plan for Aupark management console migration to the cloud. The testing plan will ensure that the system's quality is at least *maintained* after the migration. The expected plan should include *at least*.

- A list of priorities including what are the most important components to test (*i.e.,* the ones that count the most for the project), and what are the most critical (*i.e.,* the ones that may present the most problems).
- A checklist of tests for those critical and important components.
- A strategy to test at least two of the following: the web server, the batch big data process, the streaming service.

4 Extra and Hints

Please include extra details with ideas that seemed reasonable, but that you discarded somehow. Explain why you discarded them.

Remember to think about failing cases (*e.g.,* a number divided by zero must fail) as well as working cases (*e.g.,* 2+2 is 4). Pay attention to potential infrastructure and hardware problems: networking, files, database connections.

You are allowed to look up any information you need on the internet and the course's material. Ask if you have questions about the architecture of the scenario. But remember: the focus is on ***the tests***, not the applications.