

SE 339 Software Architecture and Design

Final Exam, Fall 2017

Instructor: Lotfi ben Othmane
Date: 8 December 2017
Deadline: 13 December 2017,
at 12:00 pm

Student name:
Hosam
Abdeltawab
University ID:
838973172

Directions

- This is a take home exam. Each student must work independently—this is not a group work.
- This exam is open-book, open-note and open-laptop.
- Your answers need to be precise and clear.
- Type your answers in the word document and submit your document as pdf file.
- The exam is worth 100 points

Section 1 (30 pts)

Question 1.1 (10 pts): Give five benefits of using microservices architecture style.

- Large software projects are broken down into smaller, more independent modules, some types of application become easier to build and maintain when they are broken down into smaller, composable pieces which work together.
- Better fault isolation; if one microservice fails, the others will continue to work.
- Scalability and reusability, as well as efficiency. Easy to scale and integrate with third-party services.
- Microservices simplify security monitoring because the various parts of an app are isolated. A security problem could happen in one section without affecting other areas of the project.
- Increase the autonomy of individual development teams within an organization, as ideas can be implemented and deployed without having to coordinate with a wider delivery function.

Question 1.2 (10 pts): Architecture erosion (or gap) is the difference between the prescriptive architecture and the actual (as-implemented architecture). Give three reasons for architecture erosion. Give an example for each of the reasons that you provide.

- When an implementation decision either does not fully achieve the architecture-as-planned or otherwise violate constraints or principles of that architecture. Like a strictly layered system, where each layer can only use services provided by the layer immediately below it
- When the design costs are too high to implement. Like when in a class (e.g. SE 491) the student's project's budget exceeds the budget that the school is providing.
- When a new service with its own data storage isn't simple to create. Like creating a new security website (e.g. banking website), and its database to store the hashed user's passwords and personal information

Question 1.3 (10 pts): Assume you need to connect a light bulb to a smart home framework. You found out that the library that you need to use to communicate with your device is written in Python while the framework uses Java. What tactics do you suggest to use to address the problem of this scenario and why do you believe it will work.

- I would do data manipulation between Python and Java since data are generic. I would use pipelining in Python on Linux to be able to transfer data back and forth since Linux can interact with both Python and Java.
- My solution would work because the issue there is to address is an issue of communication between the two platforms, and since you found a common platform or OS that can communicate with both Java and Python; then you can start communicating.

Section 2 (35 pts) - Read Section 5 of the paper “Software platforms for smart cities: Concepts, Requirements, Challenges, and Unified Reference Architecture” by Santana et al. provided earlier in Blackboard (Resources folder) and answer the questions below.

Question 2.1 (10 pts): The essence of smart cities platform is to support (1) connecting devices, (2) collect and process data, and offer (3) services using the data and connected devices. Do all the three reference architectures provide the three capabilities?

- Yes, because firstly, they describe and analyze the architecture of the provided platforms. Secondly, it derived a reference architecture, and finally, it compared all the architectures together.

Question 2.2 (15 pts): The proposed Unified Reference Architecture provides a Software Development Kit (SDK) to develop new services. Offering an SDK has security, performance, scalability, and/or interoperability implications on the architecture of the software. Give three quality attributes that the architecture shall satisfy.

- **Maintainability** – It shall be responsible for the management and communication of the city network nodes which will identify all the devices connected to the platform, including servers, sensors, actuators, and user devices.
- **Security** – It shall ensure user privacy which means that the data must be properly protected, and permission to store it must be acquired from the user.
- **Scalability** – It shall be used to retrieve data from city conditions which can be an efficient communication channel between the platform and city government with the citizens; moreover many tools can be used to accomplish this goal, such as Spark Streaming, which reads data streams of Twitter, and Spring Social, which is a Java-based framework to facilitate the connection with social networks such as Twitter, Facebook, and LinkedIn.

Question 2.3 (10 pts): The architecture of the OpenIOT reference architecture applies the *layers pattern* while the other two architectures do not use the pattern. Give three advantages that the pattern provides for the reference architecture.

- It provides a directory service that dynamically discovers the sensors deployed in the smart city.
- It allows the mash-up of the services defined in the platform and automatically creates a visual interface for end-users.
- It offers developers tools to implement applications directly on the platform.

Section 3 (35 pts)

You are assigned to develop a system to manage the city busses. The goal of the software is to provide the fleet managers with synthesized reports that they can use to manage their fleet. The main functional requirements of the system are:

1. Maintain a record of all the busses. The record should include the vehicle identifications, the license plate numbers, and the date of start services, the dates of acquisition, and the acquisition amounts.
2. Maintain a record of the drivers, including driver license, date of start service, name, phone number, and address of each driver.
3. Maintain a record of the city lines, including their schedules, and their routes in the city.
4. Maintain a record of assignment of drivers to busses.
5. Maintain a record of repairs and servicing of the busses, including the dates and costs.
6. Maintain a record of accidents of the busses, including the dates, related drivers, and accident causes (human error, weather, etc.).
7. Maintain a history of usage of the busses, including the drivers, and the lines.
8. Generate reports about the fleet, including history of accidents, performance of drivers, and book value of the fleet.

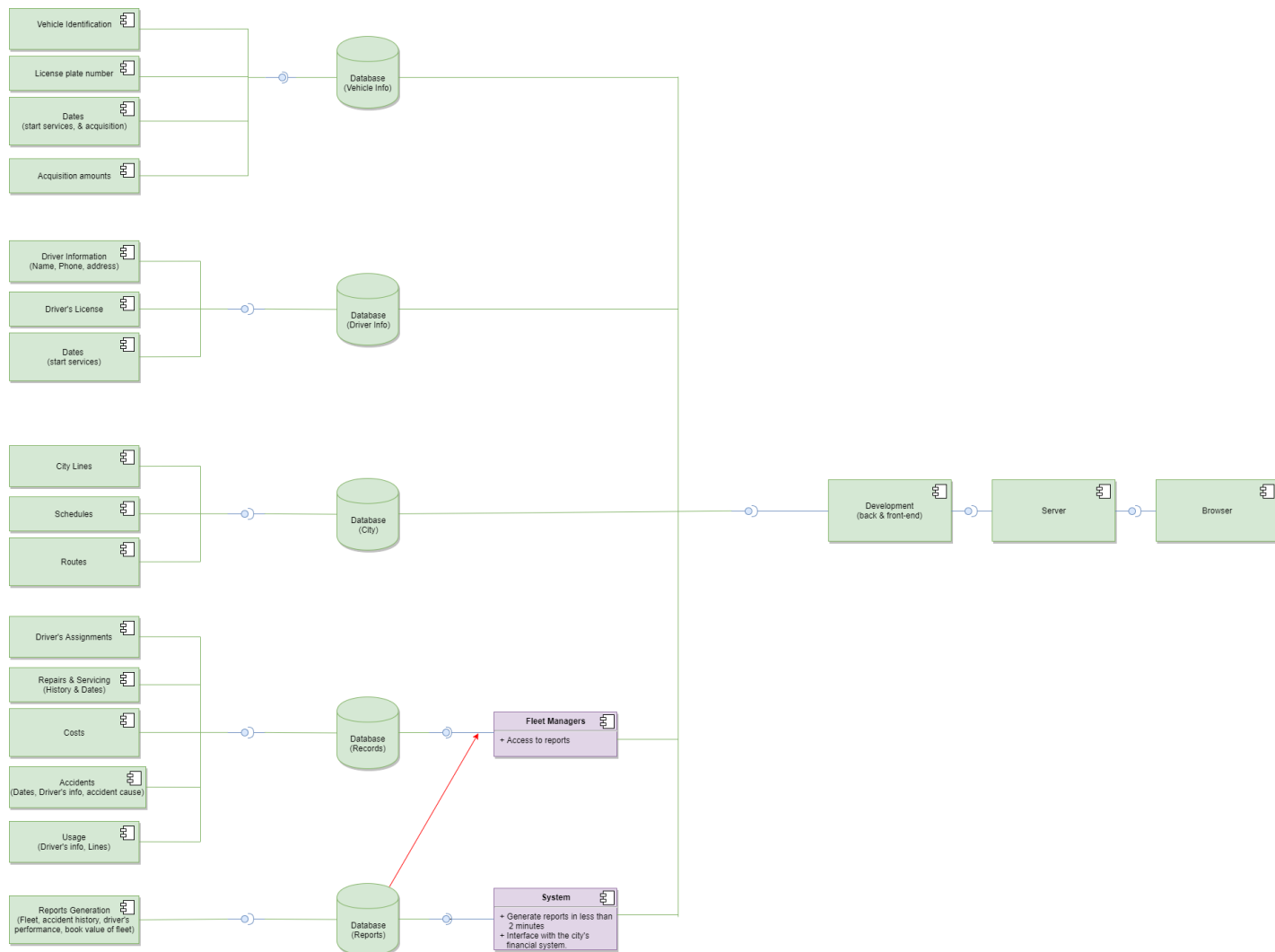
The quality attributes requirements are:

1. Only the fleet managers can access the reports.
2. The system shall interface with the financial system of the city.
3. The reports shall be generated in less than 2 min.

Constraints

1. The system shall be accessible from the browsers.

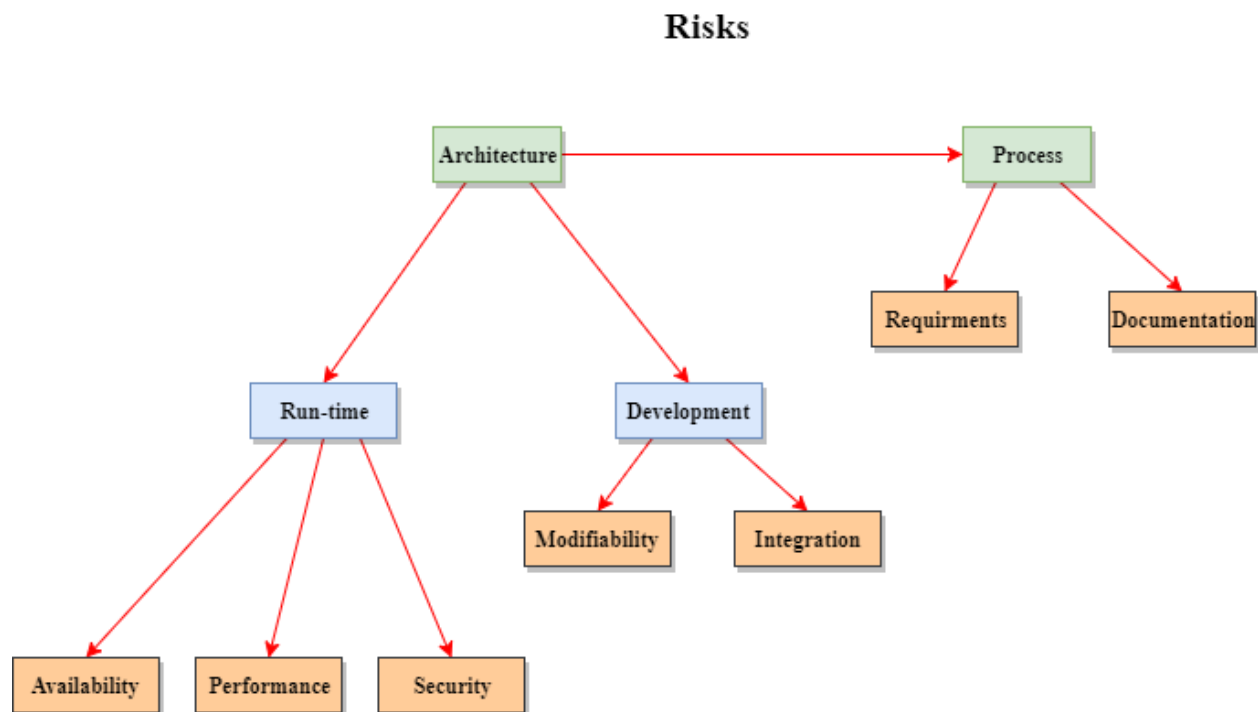
Question 3.1 (20 pts): Develop a UML component diagram of the software that you propose to develop. (Use a tool to draw your diagram.) Your architecture shall specify the technologies and libraries that you propose to use and justify the choices. It shall also describe the components that you propose (including the responsibilities of each of them) and the communication between these components.



Question 3.2 (5 pts) what architecture style (Web services, microservices, client/server, etc.) did you use? Justify your choice.

- I used web services as well as client/server. The web services are for the databases and cloud management; moreover, the client/server were for the final phase of the development of the project where back-ends and front-ends need to link together in order to communicate correctly with the interfaces and databases to view the required information on the web browser's screen.

Question 3.3 (10 pts) Use the risk themes to evaluate the architecture that you propose to develop.



- **Availability:** Using infrastructure that does not support availability mechanisms.
- **Performance:** Unfamiliarity with infrastructure choices, not knowing performance requirements, or not performing any performance modeling or prototyping.
- **Security:** Not using known mechanisms to support security goals.
- **Modifiability:** Unknown requirements or supporting the addition and deletion of multiple different reports.
- **Integration:** Lack of uniformity throughout the system.
- **Requirements:** New emerging requirements. Disagreement among the stakeholders as to the use of the system.
- **Documentation:** Defects in existing architecture diagrams, inconsistency among different views.

