Q1)

1. **THREE TIER-CLIENT SERVER ARCHITECTURE**

A typical client server architecture consists of the following tier.

* Presentation tier
* Logical tier
* Data tier

The c**lient layer** consists of:

* The browser to display the user interface and results to the user. The user interacts with only the client layer.
* In case of a thick client architecture the client layer could additionally have components to validate the user input etc.

The **application layer** receives requests from the client layer and sends back the result to the client layer that is got from the data layer.

The **database server** is connected to the database and it sends and receives data from the database using SQL queries. The architecture is different from the MVC architecture in the aspect that in client server architecture all the communication between the different components must go through a middle tier layer and this model is linear one while in a MVC model the view communicates directly with the controller and the controller directly with the model and the model communicates either with the controller of view directly and the MVC model is a triangular one.

**Objective:**

Objective is separation of concerns. As all modules are independent and each runs on a separate platform, it is easier to replace or modify any of the modules without significant impact on others. Example of three-tier client server in web applications: browser represents one layer, and server represents another and the data storage represents another.

**Advantages:**

1. The added modularity makes it easier to modify or replace one tier without affecting the other tiers.
2. Separating the application functions from the database functions makes it easier to implement [load balancing](http://www.webopedia.com/TERM/L/load_balancing.html)
3. **SERVICE ORIENTED ARCHITECTURE**

Applications deployed in the web communicate with each other. The SOA architecture is responsible for getting the computation done using communication protocols and returns the result to the user. In SOA the applications are loosely coupled. For example if the payment for a commodity is done by credit card we have two different servers one for card authentication and the other for deducting the money.

The **SOA architecture consists** of:

1. All the web services are wrapped in a WSDL
2. Communication between the web services takes place through the SOAP protocol.
3. WSDL: Web Service Description Language- It provides a common method for the web services to interact with one another. It uses XML platform that is independent of the way of representing data. It provides a uniform interface.
4. SOAP: Service Oriented Application Protocol- SOAP provides a common communication protocol.

Service Oriented architecture uses the services which are controlled by different entities. That is, SOA utilizes distributed services which are defined and control by different domain or we can say it uses external already existing services and organizes them for the required application. Thus it is a combination of services and each component in this architecture is a service and they all are independent of each other. Thus in this, mostly everything is decided on runtime such as, when a service and which service is to be called.

**Objective:**

One thing common between client server architecture and SOA is separation of concerns. But unlike client server, there are no technology constraints, as services can be different and can use different technologies but yet they are independent and communicate in SOA.

**Advantage:**

The main advantage of SOA is that the internal functioning is hidden by calling WSDL as it calls the program inside. SOA also uses dynamic binding as the exact logic of which web service is called when and in what sequence is not known until the actual application is run.

1. **CLOUD COMPUTING ARCHITECTURE:**

Cloud computing model enables convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. The Cloud Computing Architecture of a cloud solution is the structure of the system, which comprises on-premise and cloud resources, services, middleware, and software components, geo-location, the externally visible properties of those, and the relationships between them. The term also refers to documentation of a system's cloud computing architecture. Documenting facilitates communication between stakeholders, documents early decisions about high-level design, and allows reuse of design components and patterns between projects.

A cloud consists of following layer.

* Software as a service (Saas)
* Infrastructure as a service (Iaas)
* Platform as a service (Paas)

**Objective:**

The cloud computing architecture mainly focuses at sharing of resources depending on the demand for the resources.

**Advantage:**

The main advantage of cloud computing is that a user who requires a particular resource like storage or other computing tools or resources can use the one provided by the cloud depending on with or without a payment (depends on the provider) rather than go to the complexities of developing one himself.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. SOA testing is more challenging than client-server web application because of:

1. Testing of interfaces and interdependencies amongst underlying applications

2. Validation of accuracy of results from interactions between individual and integrated

Services.

3. Complex test scenarios due to heterogeneity of applications

4. Gray box testing for low-level implementation of architecture

5. Increased end-to-end testing for regression testing/maintenance phase

6. Simulated roles of a consumer of service rather than end user

7. Verification of business requirements

8. Involvement of multiple stakeholders

9. Identification of right tools to facilitate testing

10. Migration testing (in case of existing applications prior to SOA implementation)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. SoapUI is the open source functional testing tool for testing of web services. For writing test cases in SoapUI, it is not mandatory to have the previous knowledge of functional testing. Thus it is pretty much simple to create the test cases and test suites. Multiple protocols such as SOAP, REST, JDBC, HTTP are suppprted by SoapUI. It performs test quickly and run automated functional tests.

Few web service based testing provided by SoapUI:

* **Security Testing (SQL Injection and Boundary value analysis)**

SQL injection is a feature that is used to test as it is possible to get access to the database by testing the security of the database. SQL code is injected into the system to check if a user can harm a system. SQL injection is used to check the security of the database and to check if the database takes in invalid input data. Testing can be done by sending values at the boundary and check the behaviour of the system at such a situation. This makes the system more robust by analysing its behaviour at boundaries.

* **Data Driven Testing (functional testing)**

Creating data-driven tests in soapUI Pro is easy. Use the DataSource TestStep to read and loop test data from external sources, including Excel, XML, JDBC, and Files.

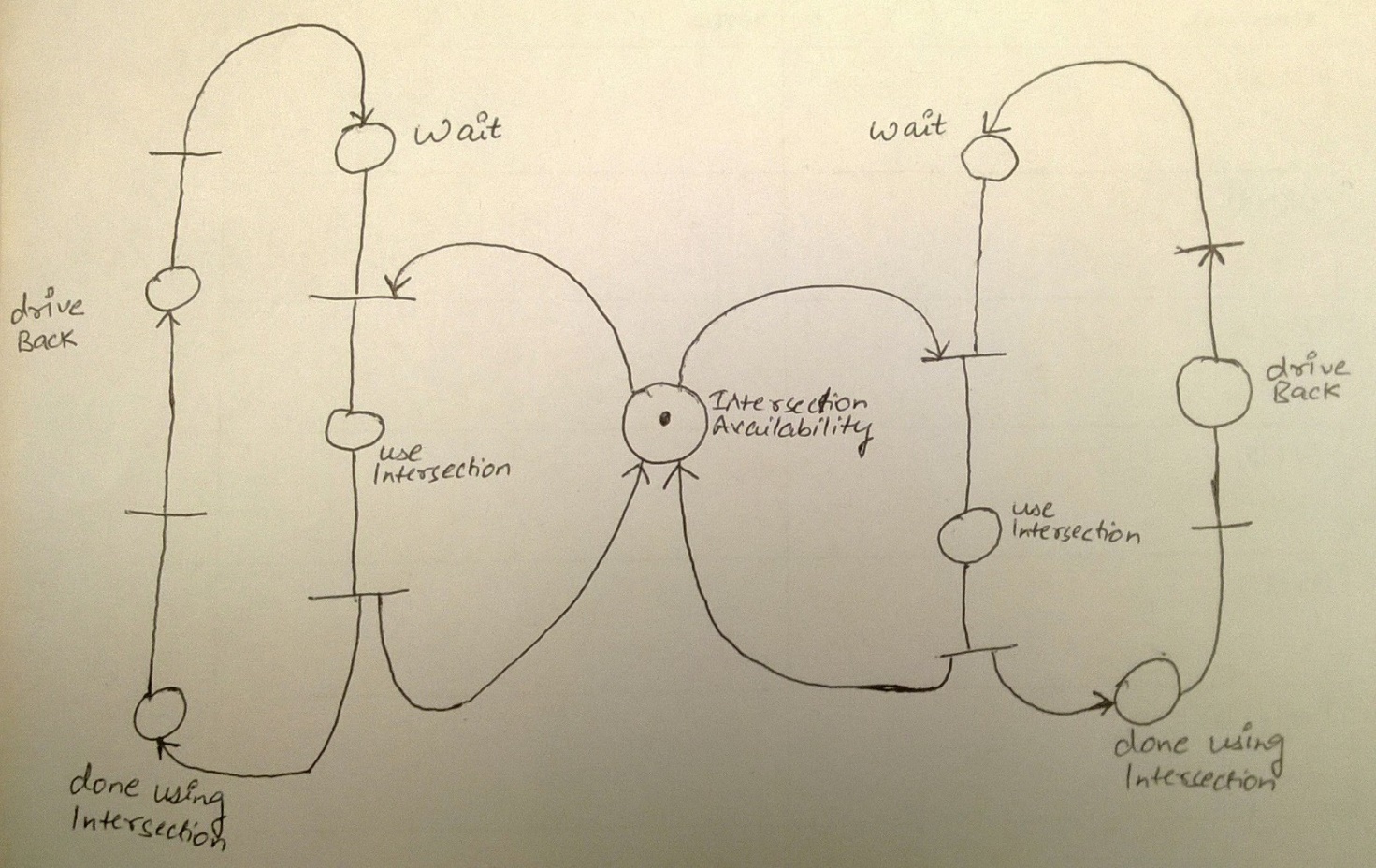
* **Auto Mock creation**

soapUI is loaded with enterprise-class features. It takes a WSDL from your desired location and automagically generates the MockService and its methods.

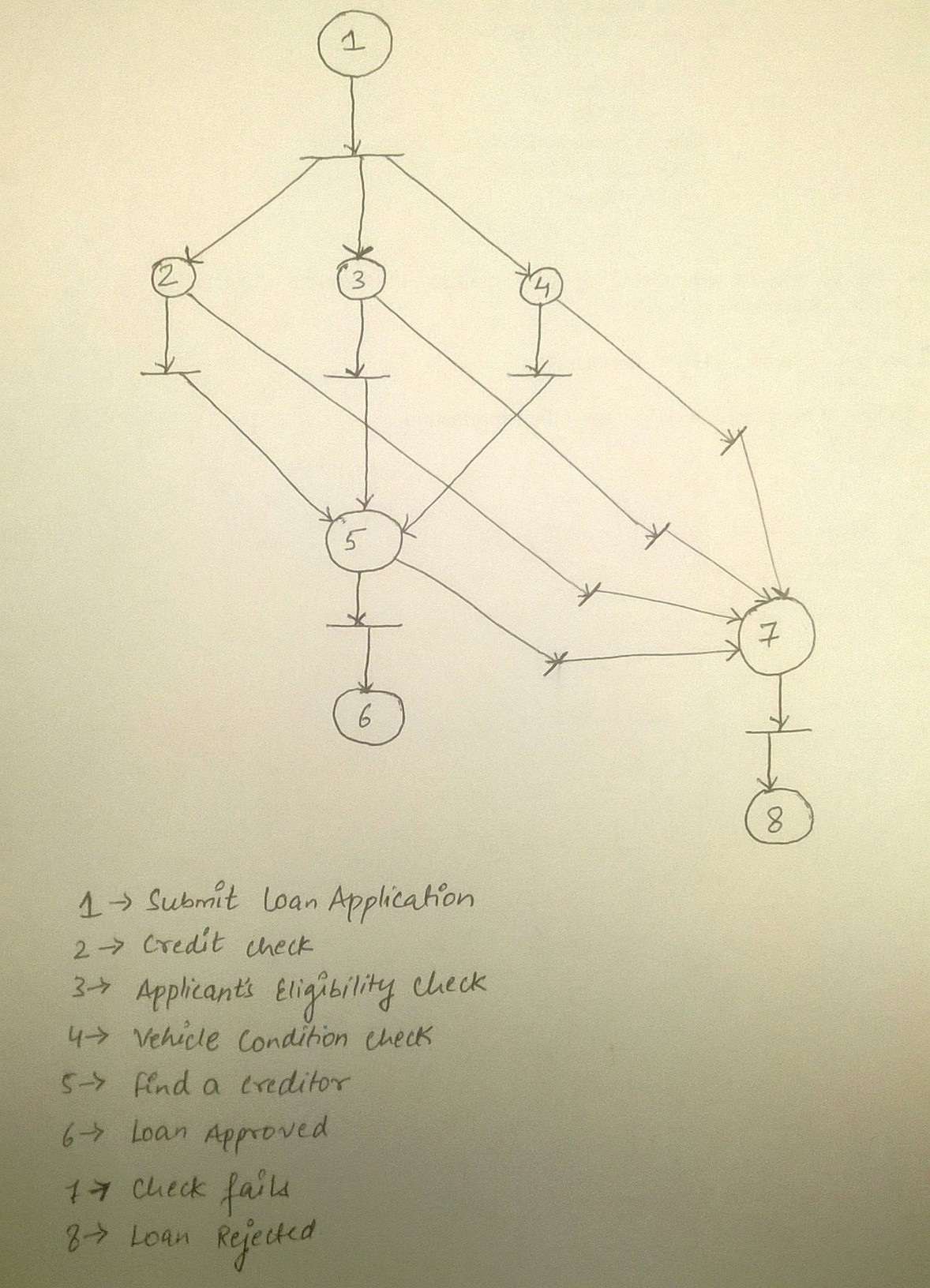
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Q2)

1. **Petri-Net for 4-Way traffic junction**



1. **Petri-net of loan approval application.**



Test cases can be generated for each step pass or fail. Say for e.g., when all the check passes but creditor is not found then it should fail. Similarly even if one check fails, loan should get rejected. Only when all check passes and creditor is also found the loan should get approved. Also once the application is submitted it should process al the checks concurrently.

**TEST CASES:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case** | **Credit Card Check** | **Eligibility Check** | **Vehicle Condition Check** | **Creditor Found** | **Output** |
| **1** | Fail | Fail | Fail | Fail | Reject |
| **2** | Pass | Fail | Fail | Fail | Reject |
| **3** | Pass | Pass | Fail | Fail | Reject |
| **4** | Pass | Pass | Pass | Fail | Reject |
| **5** | Pass | Pass | Pass | Pass | Loan Approved |