**Question 1:**

**Describe the Software Selection Maturity Scale? What is its relationship to technology adoption in the Enterprise?**

The software selection maturity scale is a five level scale which measures the maturity of a given enterprise for its technology of evaluation and acquisition process.

The first scale is the initial, this consists of chaos processes.

The second level is basic. This is the basic process in place.

The third level is the proactive and defined level. Format process is defined followed and managed.

The fourth stage is is verified adjusted. Functionality claims are verified. Scope adjusted to match available software.

The last and fifth stage is the optimised tested and approved level. Contracts optimised for risk, implementation test and post evaluation.

Technology adoption, this is how software technologies get adopted into the enterprise, it is generally slow to adopt new technologies until the industry as a whole, has seen to adopt them. If the solutions to technology enterprises are insourced or outsourced, it is often the case that similar or competitor enterprises have already adopted created or done the same solution. Enterprises are often late majority adopters in the context of the information technology adoption lifecycle.

**Question 2**

**Compare and contrast the monolithic and SOA models of enterprise application software. Describe are the benefits of a SOA composition approach to application construction.**

The monolithic model was applications that provided many diverse functions on a wide variety of data sources. This approach had many problems, the first problem was the cost of maintaining such a large complex system and the lack of flexibility to be be able to provide new or improve solutions. The monolithic application approach has been decomposed into smaller purpose services that users can use to their needs in their business.

The service orientated architecture (SOA) is a style of software design which the construction of software solutions from a set of technology independent components which can can be composed together over a network using some network defined protocol.

The advantages of SOA; s are that it is more flexible and quickly to solve business needs. Services are easier to construct, maintain and test and offer cost and time advantages. Services are often built from open source and standards software building blocks to which many enterprises contribute resources and intellectual property rights.

**Question 3**

**Why have enterprises moved towards web technologies for service software construction? What are the principal benefits?**

Web technologies are becoming more and more popular. Web technologies such as HTML have a set of industry standards with a great amount of community support. The web has great serialisation formats and the most popular formats would be XML and JSON. The web stack offers cost saving solutions to common service construction concerns.

**Question 4.**

**What are the advantages and disadvantages of HTTP statelessness as the basis of a service application protocol tier?**

HTTP is a simple stateless protocol which has become the de facto standard application layer protocol of choice for remote APIs in the enterprise

It is ubiquitous with standards support and a strong industry and community contributed set of libraries and tools to work with

HTTP is a suitable solution for both the enterprise intranet and the public Internet and allows various encryption and authentication schemes to be used along side

**Question 5**

**Describe the architectural constraints of the REST architectural pattern**

Client server - Separation of concerns between the service provider and the service consumer. The provider offers one or more capabilities and listens for requests for those capabilities.

Layered - A fixed amount of nodes are placed between the ultimate service and service consumer. The nodes must be fully transparent so that they can be added and removed at will. This allows for distribution and scalability of RESTFUL service solution in practice.

The last architectural constraints is uniform contract. This constraint states that all services and service consumers must share a single, overall technical interface. Th9s is the primary constraint that really distinguishes REST from other architectures.

**Question 6**

**Explain the relationship between resources, models and views? What is meant by view aggregation?**

Resources are exposed as RESTFUL endpoints. these are the urls.

The model is the database of the project. In the models folder,theres is files ,these files are classes or tables for the database.

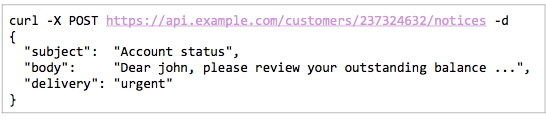
The views are the way we access the fields in the model. The views can retrieve information but also save the information tot he database.

**Question 7**

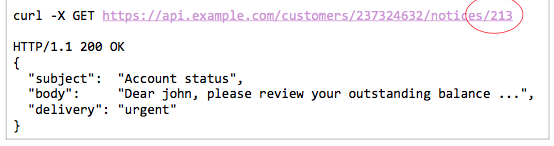
**Describe the five RESTful operations, giving examples using HTTP. What is meant by impotence? Mention which of the REST operations are idempotent and why.**

The five RESTful operation are POST, GET, PUT, PATCH, DELETE.

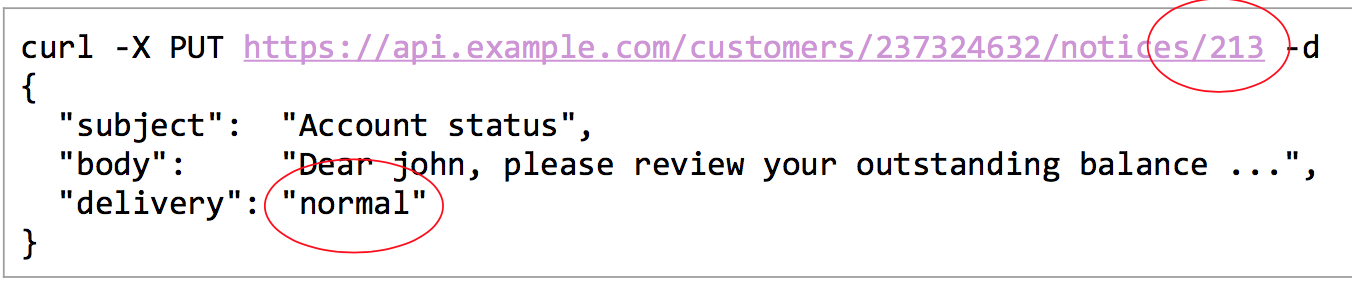
The POST is used to insert data into the database.



The GET is used to retrieve a resource by its unique identifier and its contents are returned.



The PUT operation is to update information in the database. The resource is addressed by its unique ID, and the contents relating to that resource are updated with the new data.



The PATCH operation is used to specify a resource by its unique ID and some of the contents are replaced by the newly specified attributes.



The DELTE operation is is used to remove data in the database. The id relating to data is passed and all the data relating to that unique id is removed from the database.

An example of this operation is as follows:



**Question 8**

**Explain the problem of failure propagation in SOA systems. What are the desirable characteristics of an API versioning system? What are the two kinds of API compatibility?**

The failure of propagation in SOA systems

The characteristics of an API versioning system should had the following API stability, major changes, minor changes, build identifier.

The API stability means how likely it is to change and be relied upon.

The major changes, this is the new features that have been added or existing ones changed.

The minor changes, the existing features have been updated.

The build identifier pinpoints the precise origins of the API version.

The two kinds of API compatibility are Backward compatibility and Forward compatibility.

The backward compatibility is, it changes the API to allow legacy API consumers to transparently interoperate with the new version as if it was the old version.

Forward compatibility the API is designed in such a way that it will transparently interoperate with a future version of itself allowing clients using a new version of the API to work with legacy services at least to the extent of the functionality of the functionality offered by the lagacy API.

**Question 9**

**Describe the major elements of the logical data model. Describe how it abstracts the details of database access in the application tier.**

The logical data layer abstracts the database logic and entities in to some language specific models. The logical data layer should be entirely independent of the database vendor implementation. The enterprise developers should be hidden from the details of how the database accepts queries them, return results and the specifies of the data is represented in the database.

**Question 10**

**Describe in detail the pathology of a SQL injection exploit. What should the application developer to avoid this kind avoid this kind of vulnerability.**

The pathology is when the attacker tries to exploit a query formation vulnerability by repeatedly sending queries to the service with the malformed input in the hope that a flaw in the system will be found Once a flaw is found, the attacker can potentially mount arbitrary attacks on the system to learn more about the schema, the data and extract or modify values.

To fix this vulnerability in the system, the developer should use pre parser function which checks the validity of the query before it is executed/ Using prepared statements or parameterised queries, using stored procedures with typed arguments and finally, isolating the execution with a tight security sandbox using database privileges.