Answers to Guide 9

* Web Service
  + What is a web service and how will it be useful in your team project?
    - A service offered by an electronic device to another electronic device, communicating with each other via the World Wide Web, or a specific web service implemented in the particular technology or brand, W3C Web Services.
    - It will be useful in our team project as Web technologies, including HTTP, can be utilized for machine-to-machine communication, specifically for transferring machine-readable file formats such as XML, JSON, etc.
  + Why is it a good idea to “hide” a database “behind” a web service?
    - Implementation details of the database should not be leaked to the client, which would occur with a database-driven approach which is dependent on the underlying implementation details.
* RESTful Web Services
  + Compare and contrast SOAP and REST
    - SOAP: a standardized protocol that sends messages using other protocols such as HTTP and SMTP.
      * Specifications are official web standards, maintained and developed by the World Wide Web Consortium (W3C).
      * Has strict rules and advanced security features.
      * Higher complexity requiring more bandwidth and resources leading to slower page load times.
      * Only allows XML.
    - REST: not a protocol but an architectural style.
      * Provides a set of guidelines to follow to provide a RESTful web service; i.e. the 4 basic design principles of RESTful.
      * More flexible in order to address problems with SOAP.
      * More lightweight and has better performance.
      * Supports different messaging formats including XML, JSON, HTML, etc.
    - Both allow you to create your own API.
    - They are two API styles that approach the question of data transmission from different point of view.
  + What are the most basic design principles on which RESTful web services are based?
    - Use HTTP methods explicitly – use as defined by protocol in RFC 2616.
    - Be stateless – send complete, independent requests containing all necessary data so that intermediary servers can forward, route, and load-balance without any state being held locally in between requests.
    - Expose directory structure-like URI’s – self-documenting interface that require little explanation as to their purpose.
    - Transfer XML, JavaScript Object Notation (JSON), or both – resource representation reflects the current state of a resource and its attributes at the time a client application requests it.
  + Give a one-sentence explanation of why each of the four basic principles matters?
    - They are necessary in order to design Web services with less dependence on proprietary middleware.
    - They create a flexible way to provide different kinds of applications with data formatted in a standard way.
  + What is idempotency and why does it matter in RESTful architectures?
    - Idempotence is an operation that is free of side effects. It is the property of certain operations in mathematics and computer science whereby they can be applied multiple times without changing the result beyond the initial application.
    - In RESTful architectures, clients should be explicit about each operation to be invoked and only use them for their intended function as defined by their protocols.
* Create, read, update, and delete
  + What are the basic operations required for persistent storage (CRUD)?
    - Create:
    - Read:
    - Update:
    - Delete:
    - Refers to all of the major functions that are implemented in relational database applications.
  + In a REST architecture, how do these operations map onto the standard HTTTP operations (GET, PUT, POST, and DELETE)?
    - CREATE = PUT / POST
    - READ (retrieve) = GET
    - Update (modify) = PUT / POST / PATCH
    - Delete (destroy) = DELETE
* To build web/data services in the lab, you’ll need to be prepared to use the following tools.
  + Intellij IDEA – IDEA Ultimate is a web services IDE with the same core functionality as Android Studio
  + Google Cloud Platform (GCP) – is a cloud application hosting service that requires a Google account.
  + Google App Engine (GAE) use to implement a RESTful web data service
* Google App Engine
  + Compare and contrast GCP vs. GAE
    - App Engine app is created under your Google Cloud Platform project when you create an application resource.
    - App Engine application is a top-level container that includes the service, version, and instance resources that make up your app.
  + Explain the purpose of services for: CloudSQL and CloudEndpoints
    - User services in App Engine to factor your large apps into logical components that can securely share App Engine features and communicate with one another.
      * Services behave like micro-services
    - Can run whole app in a single service or design and deploy multiple services to run as a set of micro-services.
    - CloudSQL is a fully-managed database service that makes it easy to set up, maintain, manage, and administer your relational database on Google Cloud Platform
      * Can be sued with either MySQL or PostgreSQL
    - CloudEndpoints is used to develop, deploy, and manage API’s on any Google Cloud backend.
* Note: These are not free services, we’ll be providing grant-funded billing account numbers. Limit online costs by deleting old application versions and deleting the application deployment entirely after grading.