Review

I. Java

- A. Classes
  - 1. Constructors
  - 2. Fields (public, private, protected)
  - 3. Getters/Setters
  - 4. toString
  - 5. equals
  - 6. hashCode
  - 7. Comparable interface
- B. Collections (Sets, Lists, Maps of all types)
- C. Deep Copy vs Shallow Copy
- D. Exceptions: declaring, throwing, and catching
- E. Inheritance
- F. Inner classes including anonymous inner classes
- G. Serialization
- H. I/O
  - a. Input: InputStream, Reader, InputStreamReader, BufferedInputStream, BufferedReader, Scanner, FileInputStream, FileReader
  - b. Output: OutputStream, Writer, OutputStreamWriter, BufferedOutputStream, BufferedWriter, PrintWriter, FileOutputStream, FileWriter
- J. StringBuilder

II. Databases

1. Create table statements
   a. Design tables for a given scenario
2. Insert / Update / Delete statements
3. Query: SELECT columns FROM tables WHERE condition
   a. Like you did in your project
   b. Be able to do joins
4. Transactions (What are they? How do they help?)

III. XML/JSON: especially JSON parsing as used in the project

1. Parser types (tree, token stream, object serialization)

IV. Server Project

1. Understand the purpose / responsibility of each component in the server architecture: Model, Data Access, Services, HTTP Handlers, Server Proxy
2. HTTP protocol: URLs, Contents of HTTP requests, Contents of HTTP responses, Difference between GET and POST requests

V. Software Design: Abstraction, Decomposition, Single Responsibility Principle/Cohesion, Abstracting all the way (avoid primitive obsession), Minimize dependencies, Separation of interface and implementation, Information hiding, Avoid code duplication

VI. High quality code: Comments, Good names, Indenting, Whitespace, Complex expressions, Curly braces, Statement per line, Parameters, Deep nesting, Wrapping long lines, Pseudo-code

VII. Unit testing including how you tested your project

VIII. Defensive Programming

1. Assertions, parameter checking (assertions or exceptions)

IX. Layouts and Widgets (all kinds of widgets used in project)

1. As they appear in the XML file
    a. All of the layout managers (Linear, Relative, Frame).
    b. Identifiers(@+)
2. Getting pointers to widgets
3. Attaching listeners to widgets
4. Display toast
5. RecyclerView and Adapters (how they work, what they're for)

X. Activities

1. Lifecycle
2. Code to start an activity
2. Calling another activity
3. Returning from an activity
    a. returning information from the activity about to die

4. What are bundles and intents?
    a. Adding Extras

XI. Fragments (what are they for?)

XII. Web Access

1. Server Proxy and Java's HttpURLConnection class

XIII. The toolbar

1. Options menu
2. Up button

XIV. Testing

1. Blackbox
    a. Equivalence partitioning
    b. Boundary value analysis
2. Whitebox
    a. Coverage: Line, branch, complete condition coverage, partial condition coverage