Review

I. Java
   a. Classes
      i. Constructors
      ii. Fields (public, private, protected)
      iii. Getters/Setters
      iv. toString
      v. equals
      vi. hashCode
      vii. Comparable interface
   b. Collections (Sets, Lists, Maps of all types)
   c. Exceptions: declaring, throwing, and catching
   d. Inheritance
   e. Inner classes including anonymous inner classes
   f. I/O
      i. Input: InputStream, Reader, InputStreamReader, BufferedInputStream, BufferedReader, Scanner, FileInputStream, FileReader
      ii. Output: OutputStream, Writer, OutputStreamWriter, BufferedOutputStream, BufferedWriter, PrintWriter, FileOutputStream, FileWriter
      iii. StringBuilder

II. Databases
   a. Create table statements
   b. Design tables for a given scenario
   c. Insert / Update / Delete statements
   d. Query: SELECT columns FROM tables WHERE condition
      i. Like you did in your project
      ii. Be able to do joins

III. XML/JSON: especially JSON creation and parsing as used in the project
   a. Serializing objects as XML/JSON
      i. Use Gson to serialize a Java object to JSON

b. Parsing XML/JSON data
  i. Parser types (tree, token stream, object serializer)
  ii. Use Gson to deserialize a Java object from JSON
IV. Server Project
  a. Understand the purpose / responsibility of each component in the server architecture: Model, Data Access, Services, HTTP Handlers, Server Proxy
  b. 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
  a. Assertions, parameter checking (assertions or exceptions)
IX. Layouts and Widgets (all kinds of widgets used in project)
  a. As they appear in the XML file
    i. All of the layout managers (Linear, Relative, Frame).
    ii. Identifiers(@+)
  b. Getting pointers to widgets
  c. Attaching listeners to widgets
  d. Display toast
  e. RecyclerView and Adapters (how they work, what they're for)
X. Activities
  a. Lifecycle
  b. Code to start an activity
  c. Returning from an activity
    i. returning information from an activity to its caller
  d. What are bundles and intents?

-
  -
    - i. How to pass arguments into activities and fragments
- XI. Fragments (what are they for?)
- XII. Web Access
  - a. Server Proxy and Java’s HttpURLConnection class
- XIII. The toolbar
  - a. Options menu
  - b. Up button
- XIV. Testing
  - a. Blackbox
    - i. Equivalence partitioning
    - ii. Boundary value analysis
  - b. Whitebox
    - i. Coverage: Line, branch, complete condition coverage, partial condition coverage