Review

I. Java

- A. Classes
  1. Constructors
  2. Fields (public, private, protected)
  3. Getters/Setters
  4. toString
  5. equals
  6. hashCode
- 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. Comparable interface with associated method
- I. I/O
  a. Input: Input stream, input readers, buffered input streams, buffered readers, scanners, fileReader, fileOutputStream, readers
  b. Output: output stream, outputStreamWriter, writer, buffered writer, fileWriter, bufferedOutputStream, printStream
- J. StringBuilder

II. Databases

1. Create statements
2. Insert
3. Delete
4. Query: SELECT columns FROM tables WHERE condition
   a. Like you did in your project
   b. Be able to do joins

III. High quality code: Abstraction, Cohesion, Abstracting all the way (avoid primitive obsession), minimize dependencies, separation of interface and implementation, information hiding, avoid code duplication, comments, good

names, indenting, whitespace, conditions, curly braces, statement per line, parameters, deep nesting, wrapping long lines, pseudo-code

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

V. Unit testing including how you tested your project

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

1. As they appear in the XML file
    a. All of the layout managers
    b. Identifiers(@+)
2. Attaching listeners to widgets
    a. Adding toasts to listeners debugging
    b. Adapters and recycler view

VII. 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

L. Fragments

M. Web Access (especially how you did it in your project – the map)

N. The toolbar and adding things to the tool bar

O. Defensive Programming

Assertions, parameter checking (assertions or exceptions)

P. Testing

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