**Description:**

Our application takes search parameters of category, number of questions, difficulty and type of question from the user and uses it to retrieve questions from the Open Trivia Database API.

Here is how our application meets the task requirements

1. Implement a native Android application

The name of the native Android application project in Android Studio is: *nameHere*

1. Has at least three kinds of views in our Layout:

Our application uses TextView, EditText, Button and Spinner. See content\_main.xml for details of how they are incorporated.

Here is a screenshot of the layout (with default input values) prior to the questions being retrieved:

*Screenshot here*

1. Requires input from the user

Here is a screenshot of the modified inputs from the user:

*Screenshot here*

1. Makes an HTTP request (using an appropriate HTTP method) to our web service

Our application does HTTP GET requests from GetCategories.java and GetQuestions.java. The HTTP request for GetCategories.java is: *put the HTTP request here* which requires no parameters and returns all categories tracked by the Open Trivia Database in order to populate the dropdown menu for the user to select a category. The HTTP request for GetQuestions.java is: *put the HTTP request here* where the *parameters* are the user’s requested number of questions, category, difficulty and type of question.

1. Receives and parses an XML or JSON formatted reply from the web service

An example of the JSON reply is:

*Insert JSON reply here*

1. Displays new information to the user

Here is the screenshot after the questions have been returned to the user:

*Insert screenshot here*

1. Is repeatable (the user can repeatedly reuse the application without restarting it)

The user can change the inputs and hit “Get Questions” to immediately get new questions. Here is an example after changing input parameters:

*Insert screenshot here*

1. Implement a web application, deployed to Heroku

The URL of our web service deployed to Heroku is:

*Insert URL here*

The project directory name is *nameHere*

1. Using an HttpServlet to implement a simple (can be single path) API

In our web app project:

Model: *insert name of model file here*

View: *insert name of view file here*

Controller: *insert name of controller file here*

1. Receives an HTTP request from the native Android application

*Controller file name here* receives the HTTP GET request for categories and \_\_\_ receives the HTTP GET request with the arguments “amount, category, difficulty and type” to retrieve the questions. It passes these parameters on to the model.

1. Executes business logic appropriate to your application

*Model file name here* makes an HTTP request to:

*Insert the category api here*

\_\_\_\_\_ makes an HTTP GET request to:

*Insert full API string here*

*View file name here* formats the response to the mobile application in a simple JSON format:

*Insert the JSON that is sent to the mobile app here*

**To document the rest of the requirements:**

1. Handle error conditions – Does not need to be documented.
2. Log useful information – Itemize what information you log and why you chose it.
3. Store the log information in a database – Give your Atlas connection string with the three shards
4. Display operations analytics and full logs on a web-based dashboard – Provide a screenshot.