**Software Architecture Model**

**Automatic Grading and Feedback Tool for Java**

We designed 3-tier software architecture model for our project which consists mainly of three modules naming:

1. User Interface / Presentation Layer
2. Business / Logic Layer
3. Database Layer
   1. Data Access Layer
   2. Data Source Layer

We followed the process similar to the OSI (Open Systems Interconnection) model.

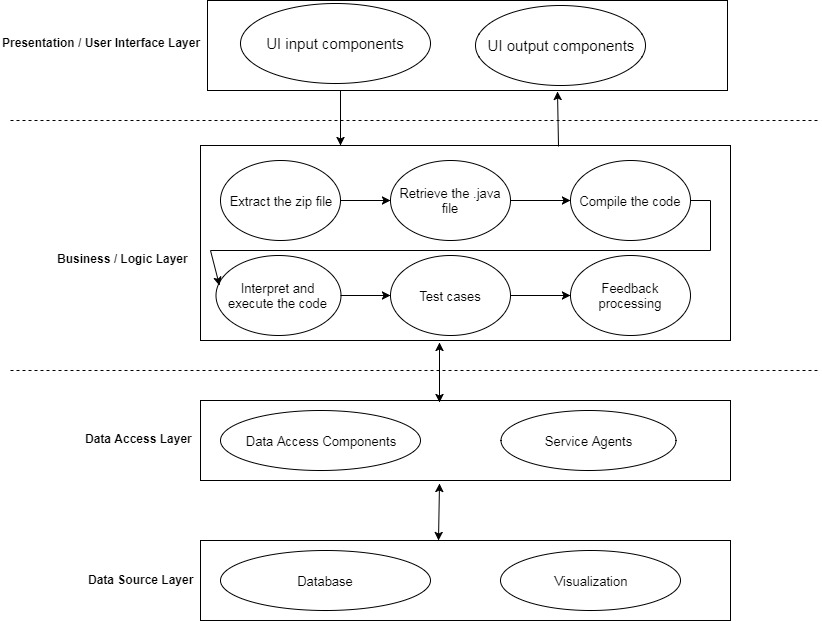


Fig: Software Architecture

Note: Test case and validation of the code should be parallel.

In this 3-tier architecture, as we mentioned that we followed OSI model, each model can communicate with the module next to it i.e. the module below or above to the current module.

**Module 1: Presentation / User Interface Layer:**

In this user interface layer, there are two components for our project.

1. UI (User Interface) input components.
2. UI output components.
3. UI input components:

In this component, it consists of some of the operations like:

* Student file submission.
* Test cases submission.
* Validation document submission.
* Grade criteria submission.
* Login.

1. UI output components:

* Grades.
* Feedback.
* Submission time.
* Information for the professor (Nice to have feature).

**Module 2: Business / Logic Layer:**

In this logic layer, it consists of **main engine** or **brain** of the software application. This layer should be clear, correct and efficient one. This layer tells the future of the application. For our project, the majority of the work will be done here. Main components or operations involved in our project are:

1. Extracting the zip file.
2. Retrieving the .java file.
3. Compilation of the code
4. Interpretation and execution of the code.
5. Test cases.
6. Feedback processing.
7. Extracting the zip file:

In this component, the student submitted assignment zip file will be extracted using scripts. Once extracted, the files will be stored in the hard disk of the computer in which the evaluation is taking place.

1. Retrieving the .java file:

In this component, the main operation to get the .java file in the group of files which are submitted by the student as a zip file.

1. Compilation of the code:

The .java file is then compiled for converting the high level language instructions to the intermediate bytecode which is understood by the virtual machine.

1. Interpretation and execution of the code:

The intermediate bytecode is then converted into machine language instructions specific to the computer.

1. Test cases:

Now the file output is compared to the test case outputs and validation is performed is performed according to the code structure to evaluate whether the assignment programming met good programming requirements.

1. Feedback:

In this phase, useful and correct feedback for the particular student is processed and written into a general text file and made available for the student to download and review it.

This module is very important for any application and also it should want to be very efficient in terms of time and space.

**Module 3: Database Layer:**

This module is important for operations on the data. Here we use a database, flat files, etc. according to our project requirements based on operating costs, handling capacity, capabilities, advantages, disadvantages, etc.

In this module, it consists of two categories:

1. Data access layer.
2. Data source layer.
3. Data access layer:

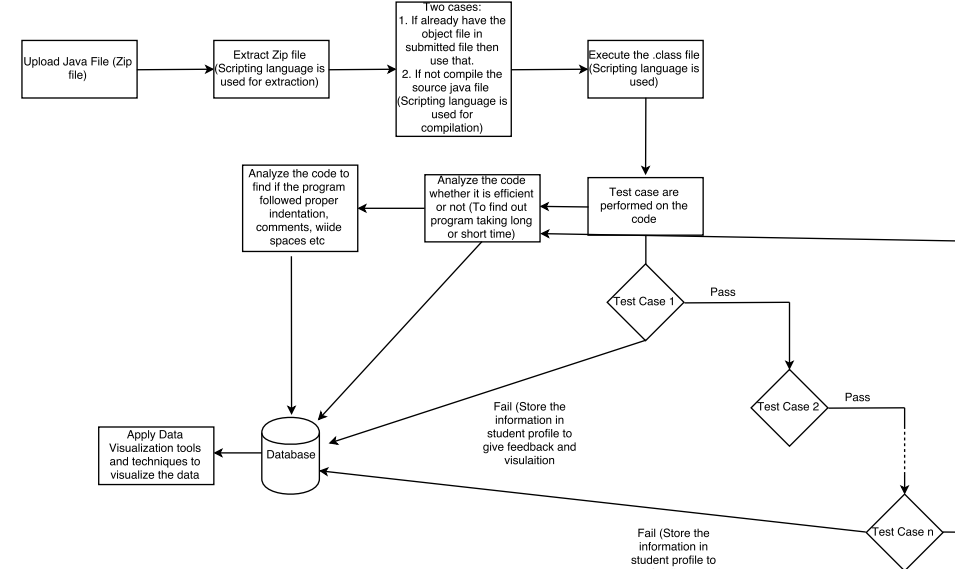
In this layer, we use the script to access the data and information from the database or flat files.

In the service agent component, we use API’s to interact with visualization tools to send information for the graphical display.

1. Data source layer:

Here come the actual storage devices. We are planning to use either flat files which are less cost to maintain but outdated. Currently, we don’t have many latest technologies on flat file operations. On the other hand, we have databases where the world concentrated on. We have latest developments on this area and very easy to transfer information from the database to visualization tools.

**Work flow Diagram:**

****

Initially, a student uploads his/her assignment file into the application tool which then will be extracted and .java file will be retrieved for the compilation process and then interpreted. The final output will be executed and compared with different test cases and compared to its expected result for each test case. If the assignment code compiled successfully and if all cases are executed then the full grade will be assigned to the student. If not, there will be a deduction of marks for incorrect syntaxes or unsuccessful compilation and for any test case failure. All the results will be stored in the database where visualization tools and techniques will be applied to the database for graphical representation of the data.