******************* README FILE *******************

Document Name	**ReadMe.docx**	Course	**RESTful CRUD Web Services**
Written by:	Varun Gaur	Date	09th April 2016

CONTENTS

DELIVERABLES

Following files are delivered for this assignment:

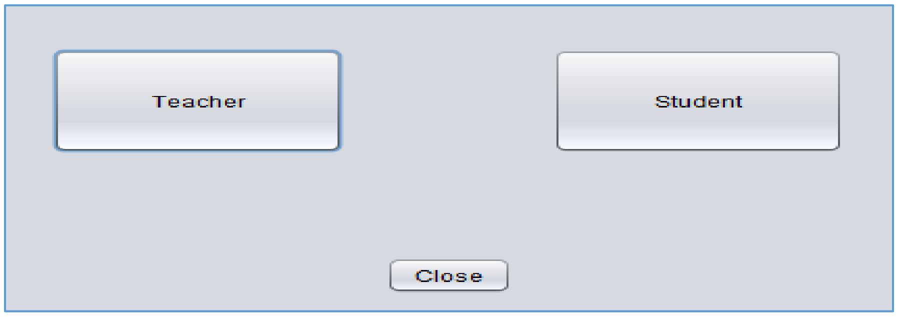
- *CRUD-GradeBook-vgaur1-NetBean.zip*
- *ReadMe.docx*

PROCESS FOLLOWED FOR IMPLEMENTING CRUD

Two separate programs has been delivered for this assignment. One for REST Client and another for REST Server. Please find below the clarification for each:

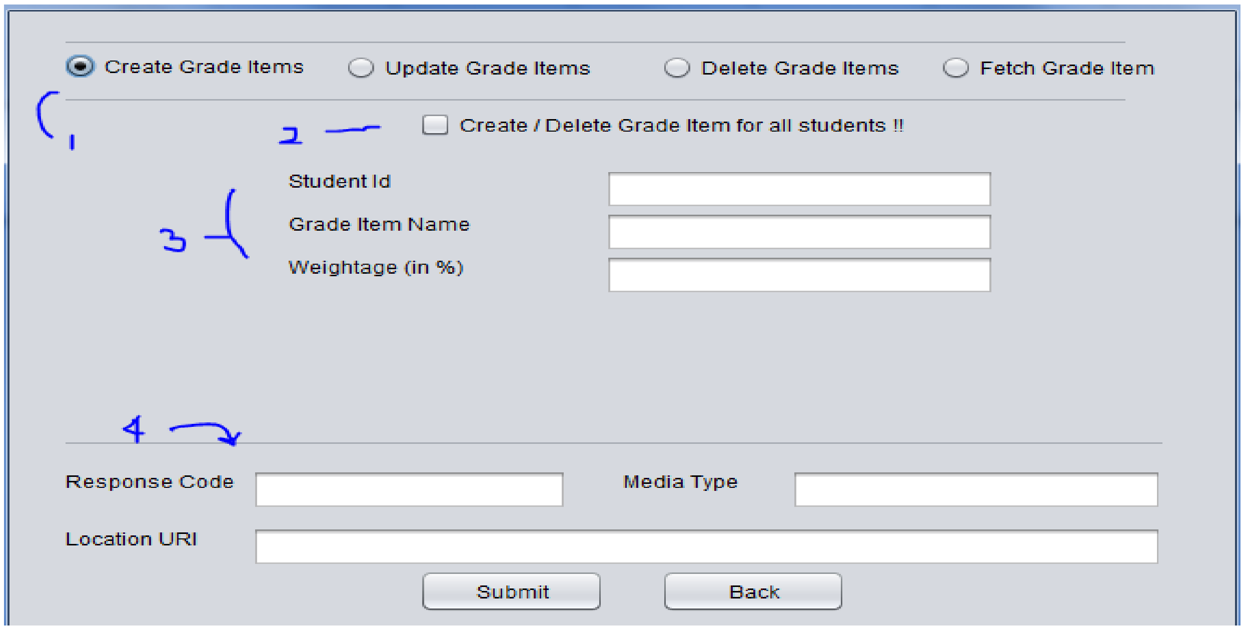
RESTful Client:

1. **MAIN MENU** (which gives user an option to select if he/she would like to proceed as a Teacher or a Student)

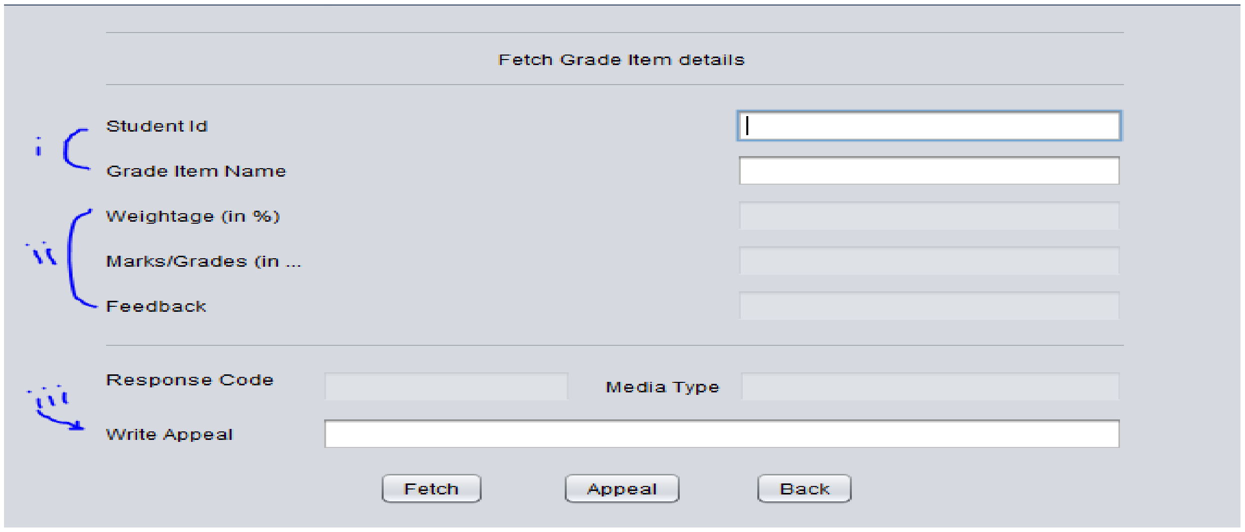


2. **TEACHER SCREEN:** It has following classifications:

i. **CRUD options:** It let user select which operation he/she want to perform viz. Create, Read (Fetch), Update or Delete Grade Items**.**
ii. **BULK processing:** For Create and Delete an additional option of creating/deleting a Grade Item name in bulk for each student in a single click has been provided.
iii. **User Inputs:** Here user will provide inputs required for individual operation like for READ (fetch), Student id and Grade Item Name are mandatorily provided by user.
iv. **Output:** Here details like Response code, Location URI etc. will be displayed.



3. **STUDENT Screen:** This screen is almost similar to Fetch (READ) option present on Teacher's Screen with just one additional feature of Grade Appeal. Students are provided with an option of making a Grade Appeal, for which they need to provide appeal in "Write Appeal" textbox and then click Appeal Button (iii).



***Note**: (a). Student ID, Weightage and Grades are numeric (Integer) and corresponding validations has been placed in Front end. Also basic mandatory validation is also in place for individual operations. (b). Weightage can be updated for Grade Item either from Create Screen or Update Screen. But if Weightage is left black on Update screen, it will not be made blank as happening for feedback. This is done to ensure that Weightage doesn't get cleared on update.*

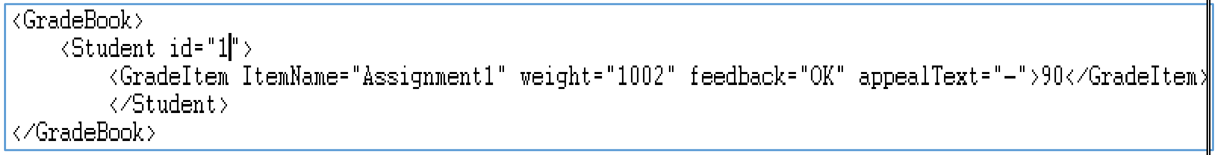
RESTful Server:

It support all the features of CRUD as follows.

i. **CREATE**: (Supported using POST Method):

a. Creation of Grade Item for an individual Student:

- URI: http://localhost:8080/CRUD-GradeBook-vgaur1-NetBeans-Server/BlackBoard/GradeBook/
- Payload: (XML in following format)

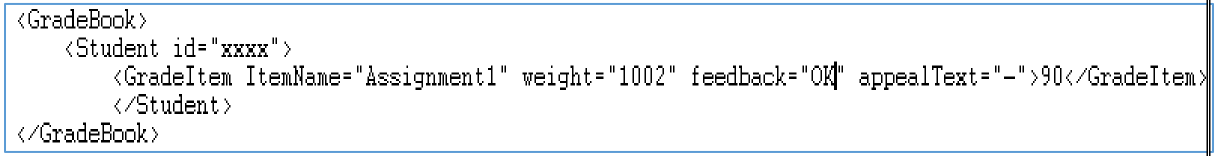


```
<GradeBook>
    <Student id="1">
        <GradeItem ItemName="Assignment1" weight="1002" feedback="OK" appealText="-">90</GradeItem>
        </Student>
</GradeBook>
```

- Return *Location* URI- http://localhost:8080/CRUD-GradeBook-vgaur1-NetBeans-Server/BlackBoard/GradeBook/{Student_Id}/{Grade_Item_Name}

b. Creation of Grade Item for all Students:

- URI: http://localhost:8080/CRUD-GradeBook-vgaur1-NetBeans-Server/BlackBoard/GradeBook/
- Payload: (XML in following format)



```
<GradeBook>
    <Student id="xxxx">
        <GradeItem ItemName="Assignment1" weight="1002" feedback="OK" appealText="-">90</GradeItem>
        </Student>
</GradeBook>
```

- Return Location URI- http://localhost:8080/CRUD-GradeBook-vgaur1-NetBeans-Server/BlackBoard/GradeBook/

Response codes: *CREATED, BAD_REQUEST, CONFLICT, INTERNAL SERVER_ERR* (201, 400, 409, 5xx)

ii. READ (Supported using GET Method):

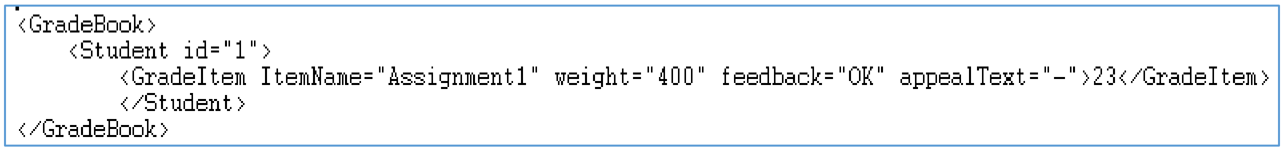
- Accessible URI's:
  - ✓ http://localhost:8080/CRUD-GradeBook-vgaur1-NetBeans-Server/BlackBoard/GradeBook/{Student_Id}/{Grade_Item_Name}
  - ✓ http://localhost:8080/CRUD-GradeBook-vgaur1-NetBeans-Server/BlackBoard/GradeBook/{Student_Id}
  - ✓ http://localhost:8080/CRUD-GradeBook-vgaur1-NetBeans-Server/BlackBoard/GradeBook/

- Return URI: None
- Payload: None

Response codes: *OK, NOT_FOUND, BAD_REQUEST, GONE, INTERNAL SERVER_ERR* (200, 404, 400, 410, 5xx.)

iii. **UPDATE** (Supported using PUT Method):

- URI: http://localhost:8080/CRUD-GradeBook-vgaur1-NetBeans-Server/BlackBoard/GradeBook/
- Return URI: None
- Payload:



```
<GradeBook>
    <Student id="1">
        <GradeItem ItemName="Assignment1" weight="400" feedback="OK" appealText="-">23</GradeItem>
        </Student>
</GradeBook>
```

Response codes: *OK, NOT_FOUND, BAD_REQUEST, CONFLICT, INTERNAL SERVER_ERR* (200, 404, 400, 409, 5xx.)

iv. **DELETE** (Supported using DELETE Method):

a. Deletion of Grade Item for an individual Student:

- URI: http://localhost:8080/CRUD-GradeBook-vgaur1-NetBeans-Server/BlackBoard/GradeBook/{Student_ID}/{Grade_Item_Name}
- Payload: None
- Return Location URI: None

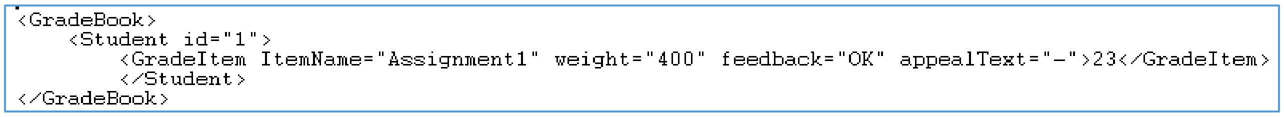
b. Deletion of Grade Item for all Students:

- URI: http://localhost:8080/CRUD-GradeBook-vgaur1-NetBeans-Server/BlackBoard/GradeBook/DelGradeItems/{Grade_Item_Name}
- Payload: None
- Return *Location* URI: None

Response codes: *OK, NOT_FOUND, BAD_REQUEST, GONE, INTERNAL SERVER_ERR* (200, 404, 400, 410, 5xx.)

v. **APPEAL** (Supported using PUT Method):

- URI: http://localhost:8080/CRUD-GradeBook-vgaur1-NetBeans-Server/BlackBoard/GradeBook/appeal
- Return URI: None
- Payload: XML format request



```
<GradeBook>
    <Student id="1">
        <GradeItem ItemName="Assignment1" weight="400" feedback="OK" appealText="-">23</GradeItem>
        </Student>
</GradeBook>
```

Response codes: *OK, NOT_FOUND, BAD_REQUEST, CONFLICT, INTERNAL SERVER_ERR* (200, 404, 400, 409, 5xx.)

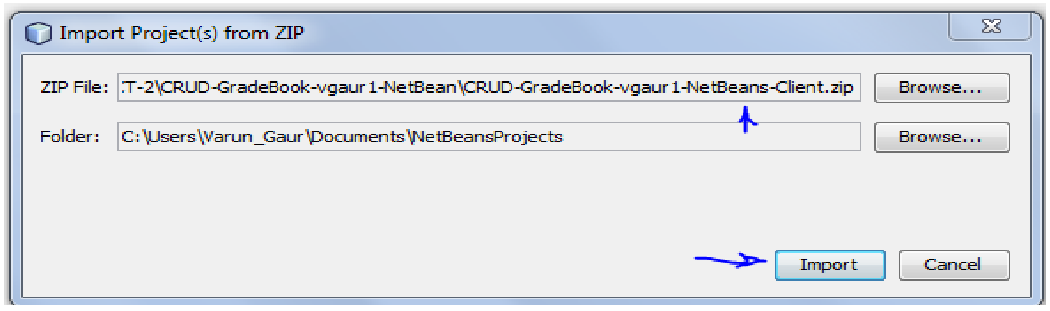
HOW TO EXECUTE (BUILD and RUN)

Following steps need to be followed for building the project and executing it:

1. Unzip the shared zipped folder to any user location (directory).
2. Import both Client and server project in NetBeans:
    a. FILE->Import Projects->From Zip File

       (Zip File Name: CRUD-GradeBook-vgaur1-NetBeans-Client.zip)

       (Zip File Name: CRUD-GradeBook-vgaur1-NetBeans-Srvr.zip)



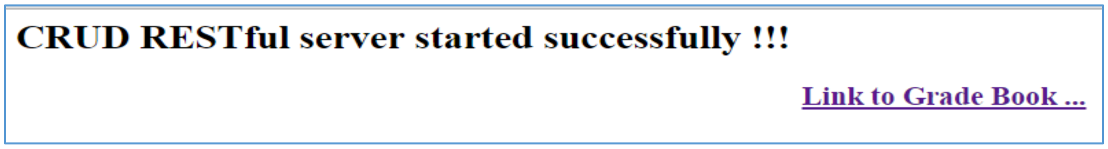
Import Project(s) from ZIP

ZIP File: :T-2\CRUD-GradeBook-vgaur1-NetBean\CRUD-GradeBook-vgaur1-NetBeans-Client.zip

Folder: C:\Users\Varun_Gaur\Documents\NetBeansProjects

** At this point, both Client and Server projects are imported in NetBeans.

3. START REST SERVER: Following steps should be followed:
    a. Right click on the Server (CRUD-GradeBook-vgaur1-NetBeans-Srvr) and Select RUN.
    b. Select SERVER as GlassFish Server 4.1.1 & Remember in Current IDE Session.
    c. Click OK. Following screen should open on system's Default browser:

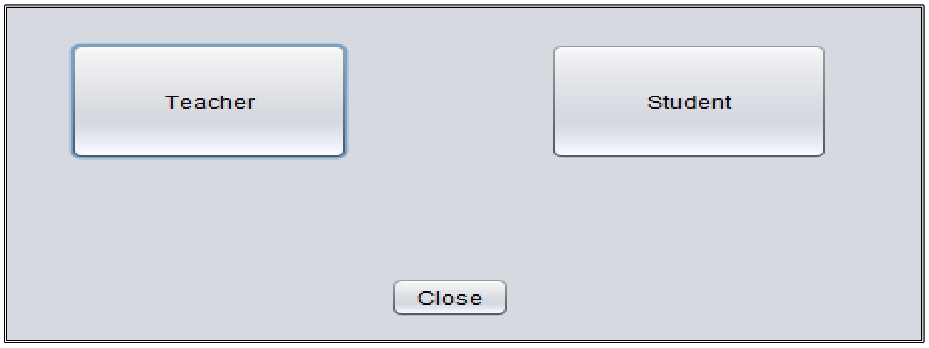


CRUD RESTful server started successfully !!!

Link to Grade Book ...

4. START CLIENT: Similar to server following steps should be followed:
    a. Right click on the Client (CRUD-GradeBook-vgaur1-NetBeans-Client) and Select RUN.
    b. If system prompt for selecting Main Class, Please select CRUDBlackBoard_UI

    Client screen should pop up on successful startup of Client:



***Note (For step-2):** I am sharing unzipped folders both Server and Client for your reference. In case shared Zipped files are corrupted for some reason (& doesn't get loaded in NetBeans). For loading folders directly instead of zip files, use following option: FILE → OPEN PROJECT → *{Select both Server/Client Folder}*

IMPORTANT POINT:

Web-Service is built to listen on localhost/8080 PORT. IF PORT NUMBER is changed in CLIENT/SERVER, program would not execute as expected and in that scenario, Port number need to be changed to 8080.

MINIMUM SYSTEM REQUIREMENTS [PART III]

- NetBeans IDE (8.1)
- GlassFish Server (4.1.1)
- Window 7 (32/64 Bit)

FEEDBACK & UPDATES

For any updates and feedback, please contact me.