IDS517 2015 FALL PROJECT PROGRAMMER'S DOCUMENTATION

GROUP F15g109

Dinesh Kumaar Rajan

Narendar Tankasala

Premnath Ramanathan

Contents

1. Introduction:

An online system to facilitate quiz and exam administration and grading for quantitative statistics and math courses. This is done using JSF.

The programmer's documentation contains:

- Class Diagram
- Database Schema
- UML diagram
- Flow of program

2. Class Diagram:

LoginBean:

LoginBean
- username: String - password : String - message: String - loginURL: String - loginCheck: Boolean - ec : ExternalContext - session: Httpsession
+ login():String + logout(): String + clearLists() : String

The Login Bean is used to get username and password from the user and compare it with the already stored database username and passwords. Then it navigates the user to student or professor or teaching assistant or tutor homepage based on it.

Database:

Database
- username: String - password: String - url: String
+ connectDB() : Connection + validateTableName(String, Connection) : Boolean + fetchTableNames(ArrayList<String>, String, Connection): void + disconnect(conn: Connection) : void

This Bean is used to connect to the database. It gets the username and password and verifies it and connects to the database. It also has a method to disconnect from the database once the required action is over.

DatabaseConnect:

DatabaseConnect
- MYSQL_DATABASE_DRIVER : string - DB2_DATABASE_DRIVER : String - ORACLE_DATABASE_DRIVER: String - HOST_NAME: String - userName: String - password: String - host: String - rDbms: String - schema: String
+ fetchDBdata(): String + connect(): Connection

In this bean, we get the database information using various accessor methods. It is used to fetch data from Database and display it in the webpage.

Professor Bean:

ProfessorBean
-firstName: String -lastName: String -UIN: String -userName: String -password: String -department: String - courses: String
+ viewAsStudent(): String + uploadRoster() : String + viewCourseAssessment() : String + uploadCourseAssessment() : String + dloadCourseAssessment() : String + dloadStudRoster() : String + analyzeScores() : String + courseStats() : String + uploadQuiz() : String + uploadExam() : String + genReports(): String

This bean is used to store all information about the Professor and also the actions the Professor can do. The information of the Professor Includes his first and last names, UIN, username, password, department and the courses the Professor handles. These are obtained using accessor methods. The Professor can view the webpage as a student, upload/download assessments, upload/download quizzes, upload/ download rosters, score the students and generate reports. These are accomplished using various methods as shown in the above.

ActionBean:

ActionBeanFile
- uploadedFile: Uploadedfile - serialVersionUID : long - fileLabel : String - uploadedFilecontents: String - fileName: String - fileSize: long - fileContentType: String
+ processFileUpload(): String

This bean is used to upload any file into the server. It gets the file Label, filename, filesize using the various accessor methods.

Student Bean:

StudentBean
-firstName: String -lastName: String -UIN: String -userName: String -password: String -department: String
+ dloadCourseAssessmentAnswers() : String + uploadCourseAssessmentAnswers() : String + uploadQuizAnswers():String + dloadQuizAnswers():String + viewMarks() : String + onlineExam(): String

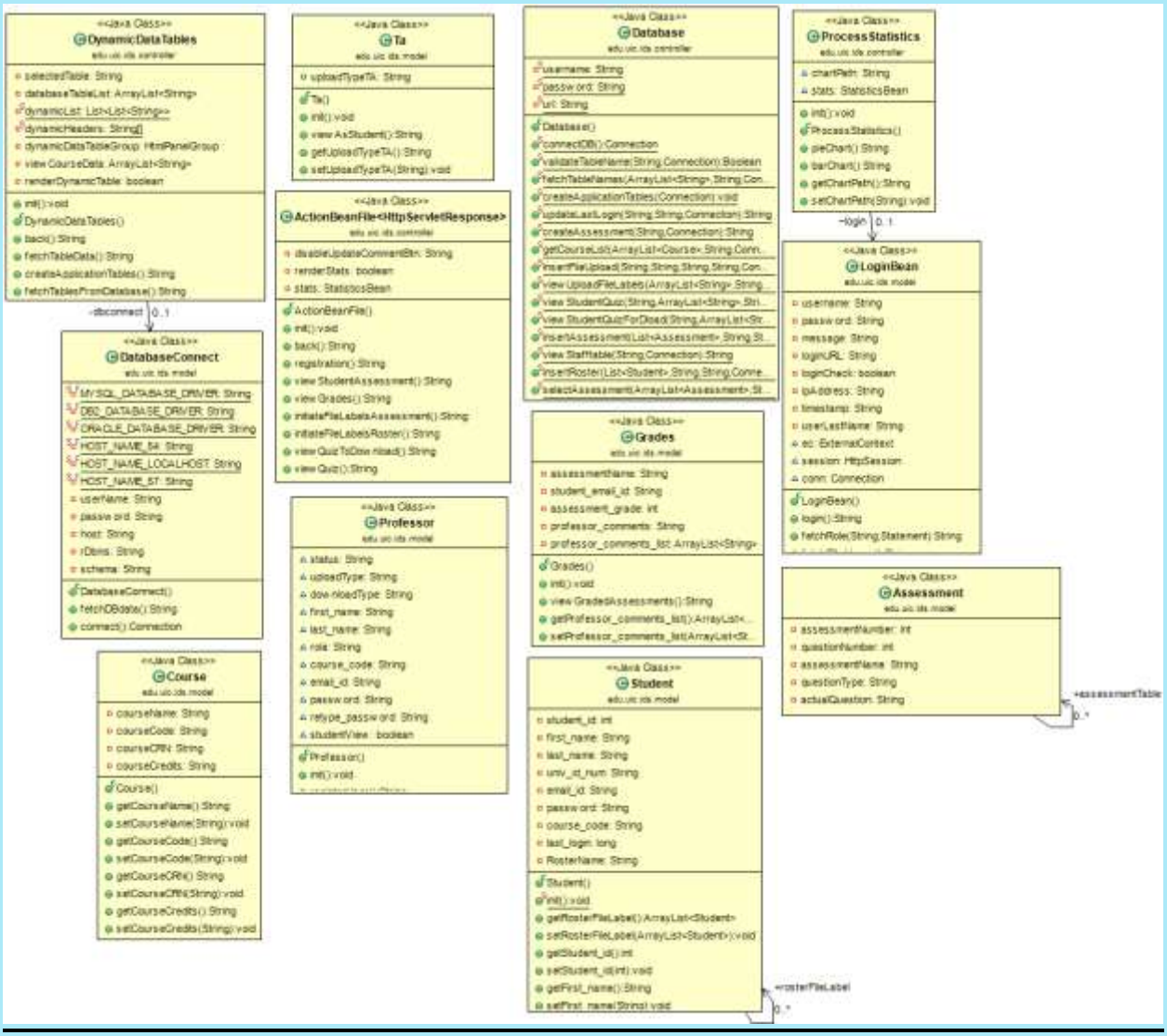
The student Bean has the information of the student like first and last names, UIN, username and password, department and the courses the student has taken. The information are collected using various accessor methods. Also the student can upload/download assessment answers, download assessments, view his marks and also write Exam online. These are done using the methods mentioned above.

TeachingAssistantBean:

TeachingAssistantBean
-firstName: String -lastName: String -UIN: String -userName: String -password: String -department: String - course:String
+ uploadRoster() : String + viewCourseAssessment() : String + uploadCourseAssessment() : String + dloadCourseAssessment() : String + dloadStudRoster() : String + analyzeScores() : String + courseStats() : String + uploadQuiz() : String + genReports(): String

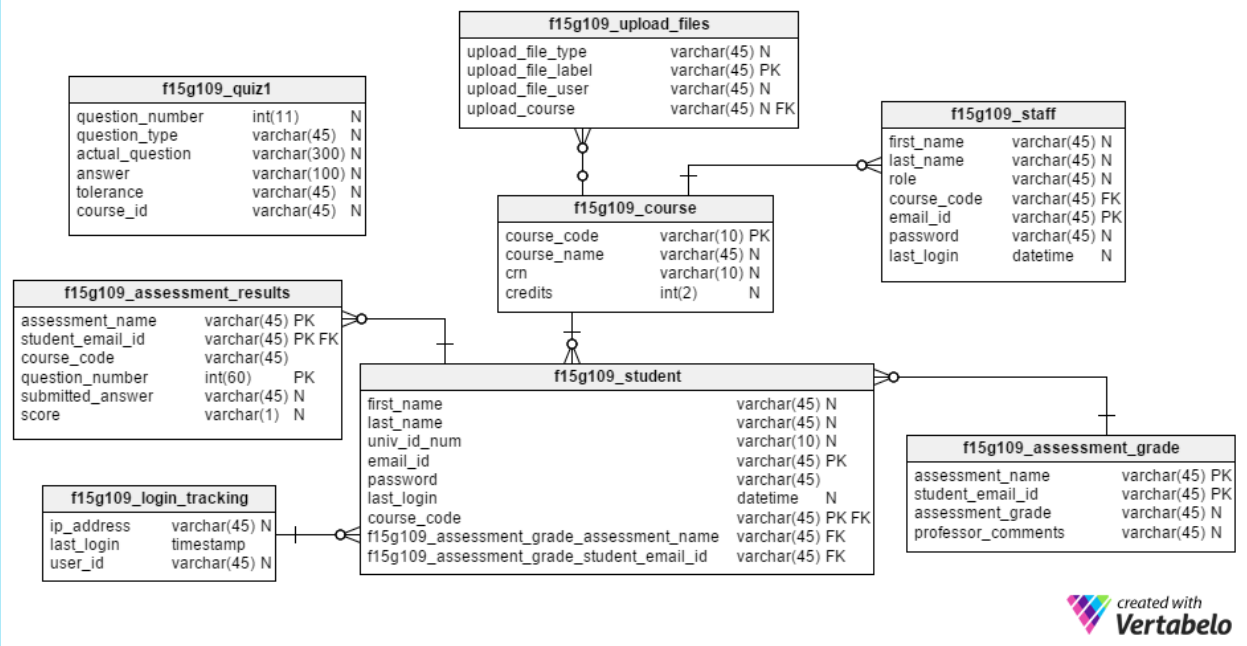
This bean is used to store all information about the Professor and also the actions the Teaching assistant can do. The information of the Teaching assistant includes his first and last names, UIN, username, password, department and the courses the Teaching assistant handles. These are obtained using accessor methods. The Teaching assistant can view the webpage as a student, upload/download assessments, upload/download quizzes, upload/ download rosters, score the students and generate reports. These are accomplished using various methods as shown in the above.

3. UML Diagram:



4. Database Schema

DB schema covers lists the tables and the relationships between the entities which are a part of the database schema.



Student table

This table contains the details of all the student users who are using the application. The student table contains all the necessary information of the student including his demographic information, his login credentials and information on the last login.

Columns

- *first_name*
- *last_name*
- *univ_id_num*
- *email_id*
- *password*
- *last_login*
- *course code*
- *f15g109_assessment_grade_assessment_name*
- *f15g109_assessment_grade_student_email_id*

Course table

This table contains the details of all the courses available for that semester. This table contains all the necessary information of the course.

Columns

- *course_name*
- *course_code*
- *credits*
- *crn*

Staff table

This table contains the list of all the staff handling course for that semester. This is a comprehensive table containing the details of all the course instructors, TAs and tutors.

Columns

- *role*
- *first_name*
- *last_name*
- *course_code*
- *email_id*
- *password*
- *last_login*

Assessment grade table

This is a dynamic table created when an assessment file is uploaded. This table contains all the necessary information on the assessment which is uploaded.

Columns

- *assessment_name*
- *student_email_id*
- *assessment_grade*
- *professor_comments*

Assessments Results table

This table contains the following fields.

Columns

- *Assessment_name*
- *Student_email_id*
- *Course_code*
- *Question_number*
- *Submitted_answer*
- *Question_score*

Upload files table

The upload files table contains the upload file details.

Columns

- *Upload_file_type*
- *Upload_file label*
- *Upload_file_user*
- *Upload_course*

Login Tracking table

This table is created in order to save the users' login time. It contains

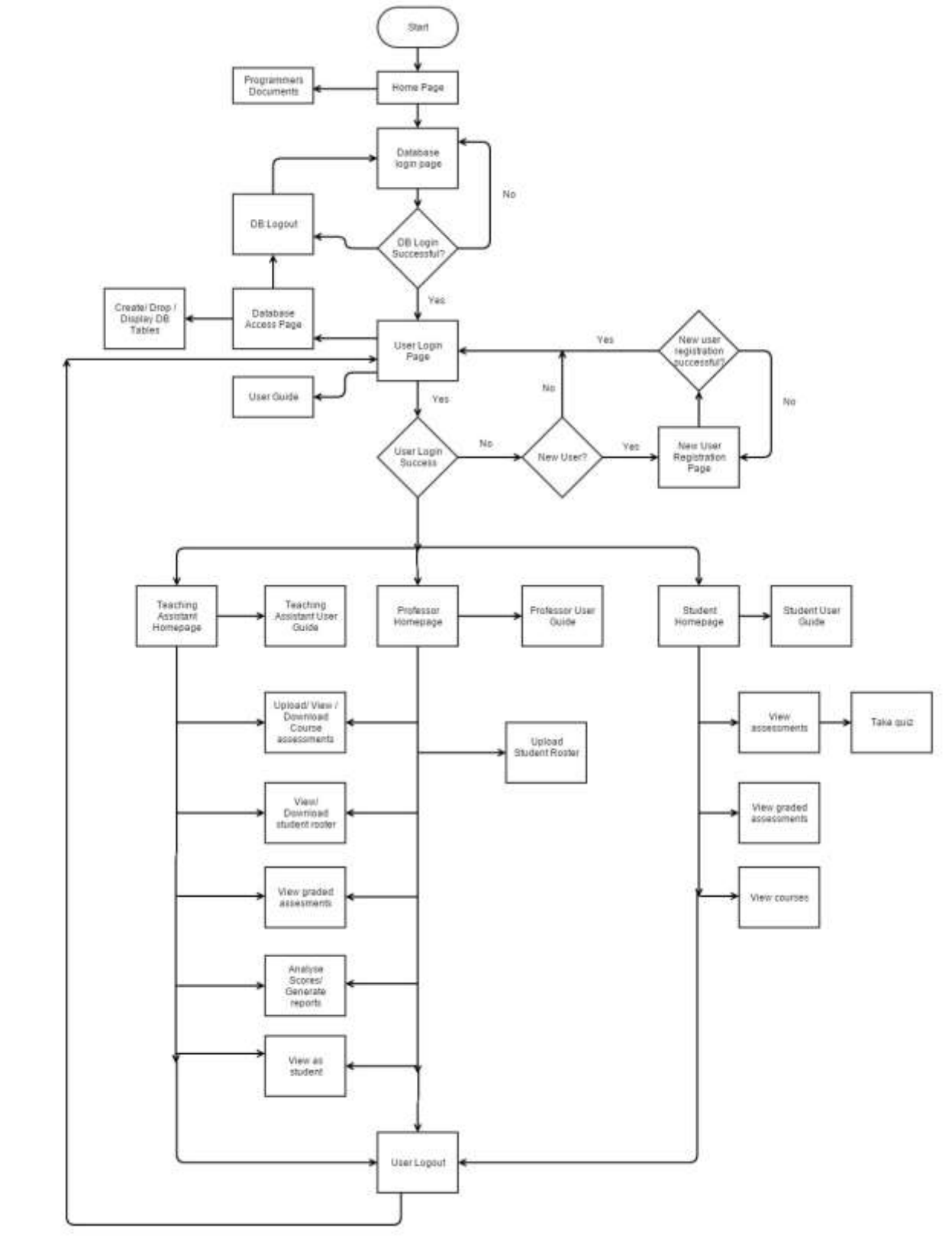
- *Ip_address*
- *Last_login*
- *User_id*

Quiz table

This table stores information about the assessments uploaded by the professors/ TA. This table contains

- *Question_number*
- *Question_Type*
- *Actual_question*
- *Answer*
- *Tolerance*
- *Course_id*

5. Flow of project:



5.1 Introduction:

The web application starts from the group home page which has the programmer's documents. It also has the database login link. Clicking on it leads to the DB login page. The credentials are pre-populated. Successfully logging in leads to the user login page.

5.2 Database access page:

This page contains access to the DB. In the Database, we can create application tables required for the web page. We can also view the tables using the display table option. Application tables must be dropped using drop application option. Other tables created can be dropped one by one using the drop selected table option.

5.3 User login:

This page has a user guide to help users to login. Previously registered users can login using their username and password. Username is the user's email – id. New users need to register to login.

5.3.1 New User Registration:

New users need to enter their details to register as a new user. After registration, they will be directed to the user login page where they can login.

5.4 Professor/TA/Student login:

This is the Users Login page. Here the user has a user guide which mentions in detail the actions that can be carried out by the user.

5.5 Logout:

Logging out leads to the user login page. There will be an option to log out from Database in the user login page.