Class Assistant Project Summary

Week Eight

Levy Devin

October 20 2013

**Table of Content**

Introduction………………………………………………………………………….3

Use Cases……………………………………………………………………………3

Database……………………………………………………………………………..4

Description by Layer………………………………………………………………...6

1.DAL………………………………………………………………………..6

2.Repository………………………………………………………………….8

3.Service……………………………………………………………………..9

4.Business……………………………………………………………………10

5.UI…………………………………………………………………………..12

Future Features………………………………………………………………………14

**Introduction**

In academia, the importance of instructor and student knowledge share is an imperative that cannot be ignored. Having the ability to share communicative standards such as grades or assignments helps both parties in their quests for efficiency and productivity.

Students will be able to better collaborate with each other in addition to their teachers. Teachers will have the ability to better communicate with students and avoid unnecessary conversations regarding grading and scheduling. Everything that deals with those issues will be posted and automated for students to see, further freeing the facilitators time – which is a commodity in short supply and that can be better used for purposes such as teaching rather than returning extraneous emails. Ultimately, this system will help save time of both the teachers and the students, and as a result time can more people will be happier.

**Use Cases**

The brief functionality of the application can be described in User Stores:

1. User is able to create an account for the application, which includes UserName/Password for logIn;
2. Depending on user status, upon login he will be redirected either to Student page or Professor page;
3. Student can: look up his/her grades and brief statis and see the posed comments;
4. Professor can: add and edit event, post grade for each student per exam, edit grade, see stats by exam.

The full list of implemented features will described in details in Description by Layer section.

**Database**

To store the necessary information for this application entity-relational database was created.



Database consists of the following tables:

1. People

|  |  |  |  |
| --- | --- | --- | --- |
| Data Field Name | Type | Description | Possible Values |
| id | int | Unique Id | 1+ |
| Fname | varchar | User First Name | Name |
| Lname | varchar | User Last Name | Last Name |
| Password | varchar | User Password | Any values |
| UserName | varchar | User UserName | String of Characters |
| LabId | int | Lab a person belongs to | 1-4 |

2.Lab

|  |  |  |  |
| --- | --- | --- | --- |
| Data Field Name | Type | Description | Possible Values |
| Id | int | Unique Id | 1+ |
| Date | varchar | Date of lab | Day / time |

3.Grades

|  |  |  |  |
| --- | --- | --- | --- |
| Data Field Name | Type | Description | Possible Values |
| Id | int | Unique Id | 1+ |
| value | int | Value of the grade | 0-120 |
| ActivityId | int | Id of exam | 1+ |
| PersonID | int | Id of a Person | 1+ |

4.Activities

|  |  |  |  |
| --- | --- | --- | --- |
| Data Field Name | Type | Description | Possible Values |
| id | int | Unique Id | 1+ |
| ActivityName | varchar | Name of Activity | Exam1-Exam4 |
| Schedule | Date/time | Schedule of exam | Yyyy/mm/dd hour/min |

5.Comments

|  |  |  |  |
| --- | --- | --- | --- |
| Data Field Name | Type | Description | Possible Values |
| id | int | Unique id | 1+ |
| text | varchar | Text of the comment | text |
| ActivityId | int | Id of the exam | 1+ |

Tables Pictures and File are not implemented in this version.

**Description by Layer**

1. **DAL**

A Data Access Layer (DAL) (figure 1) contains main classes from DataBase Layer to hold all application data. Those classes provide containers for interaction with data stored in entity-relational database (in this case Microsoft SQL Server) and hides complexity from external world.

The classes included to DAL are (figure 2):

* Person: this class have properties to hold an information about the registered student, such as first and last name, UserName and Password, and Lab.
* Activity: contains information about exams, including schedule.
* Lab: hold information about lab date.
* Grade: hold grades value for each exam for each registered student;
* Comment: allows to store a professor’s comment for each exam, which is displayed on user page;
* File (not implemented in this version): allows to upload and download study guides for exams;
* Pictures (not implemented in this version): attaches a picture to a student’s profile;
* TableContext: this class hold a specific set of properties from other classes to enable an interaction with several DAL classes at the time;

Figire1: Layer Diagram

DAL

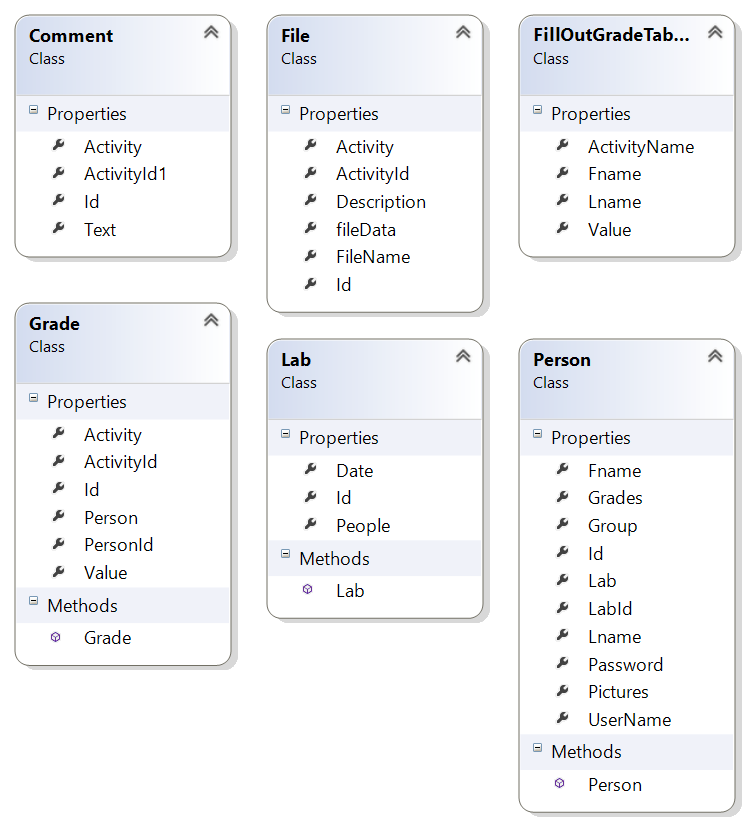
Service/

Repo

Business

UI

Figure 2: DAL Class Diagram



1. **Repository Layer.**

This layer (RepoPattern, figure 4) allows isolation of DAL from Business Layer. Using Generics allows to interact with DAL in a more efficient manner and would ensure that application can be still functional if DAL is replaced. It provides basic functionality such as Create, Update, Insert, Delete. This layer contains a class that holds:

* Generic Class;
* Generic Interface;
* Interface;

Figure 3: Repository Layer (RepoPattern, figure 3)

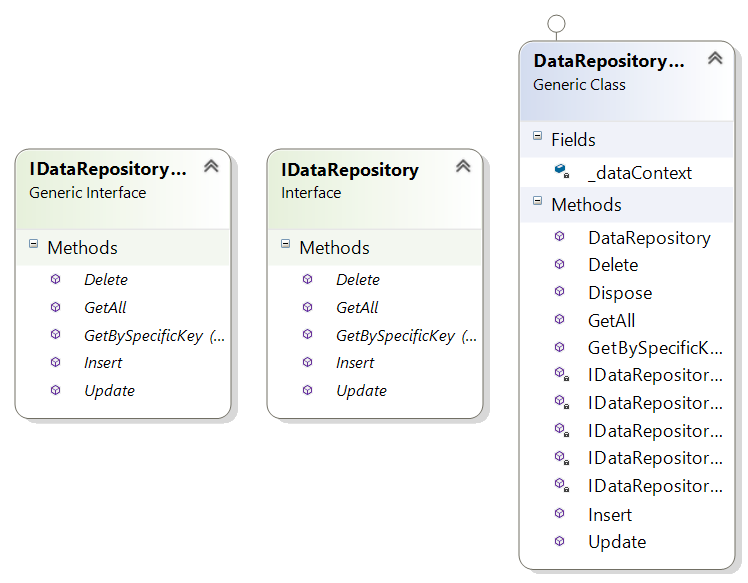


Figure 4: Layer Diagram

DAL

Service/

Repo

Business

UI

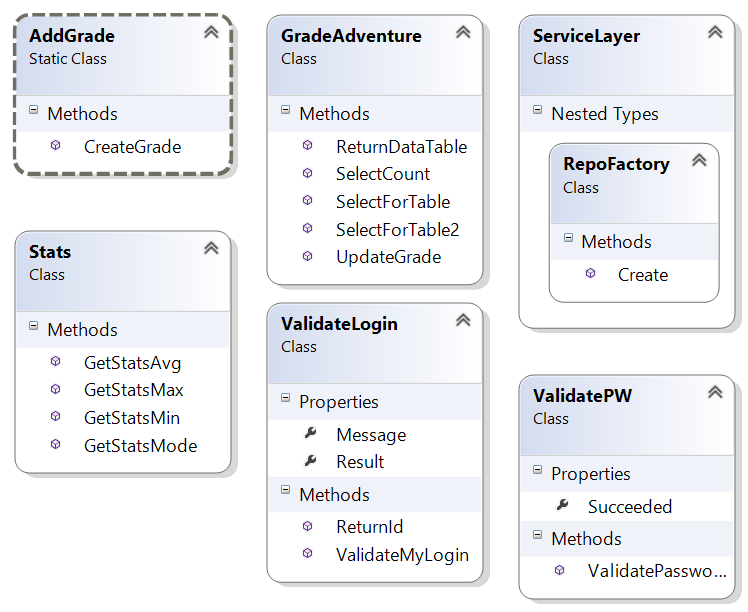
1. **Service Layer**

This layer allows to fetch the data from DAL in the way that is more specific for this particular application and pass in to Business Layer (figure 4):

This layer consist of following classes (figure 5):

* AddGrade: this class allows for the grade to be added and stored in database;
* GradeAdventure: this class provides methods for updating already created grades;
* Stats: this class provides data for statistics from the grade table in database;
* ValidateLoging : contains method for checking if the user is registered or not;
* ValidatePassword (not implemented in this version): this class allows to check user input password to be the certain length;
* ServiceLayer: This layer return a specific class to be called against Repository Layer.

Figure 5. Service Layer.



1. **Business Layer**

This layer provides object with functionality, such as Create, Delete, Insert, Update, etc for each class of DAL. This prevents application from using DAL directly from UI layer (figure 6).

The classes in Business layer are:

* BusinessActivities: this class has methods to return to UI layer information to manipulate activities (exams in this case);
* BusinessComments: this class has methods to return to UI layer information to manipulate comments (post and display);
* BusinessGrade: provides methods to extract/save information about grades from/to Database;
* BusinessFileContet (not implemented in this version):allows the manipulation of File DAL class;
* BusinessLab: contains method for interaction with Lab class;
* BusinessPerson: this class has methods to return to UI layer information to manipulate registered users;
* BusinessPicture (not implemented in this version): provides methods for uploading/displaying pictures;

Figutre 6: Layer Diagram:

DAL

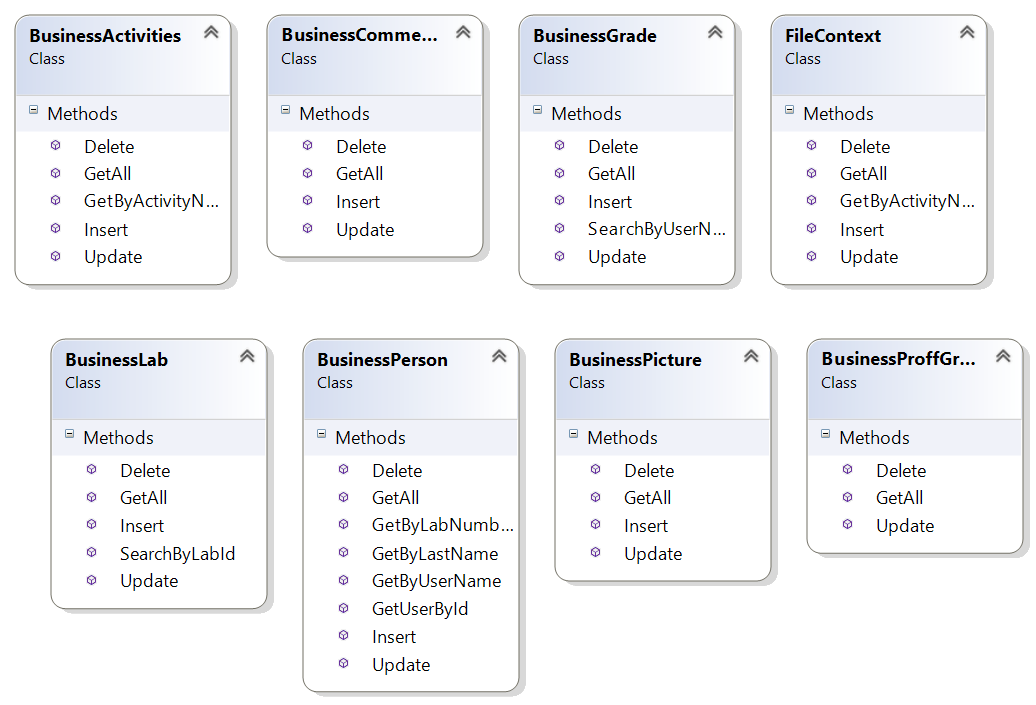
Service/

Repo

Business

UI

Figure 7: Business Layer Classes:



**5. UI Layer.**

This layer provides User Interface for interaction with the application. There are two levels of users: Professor and Student. A Professor user has following options:

* ProfAddGrade: allows the professor to post a grade for each student for each exam;
* ProfEditGrade: allows the professor to edit a previously inserted grade for each existing student for each exam;
* ProfAddEvent: allows the professor to create an exam and enter it date/time and display it on the screen;
* ProfStats: allows the professor to see the statistical information about each exam, such as min, max avg grade, and mode.
* ProfMainPage: this page allows the professor to navigate between other pages and has two external links for school web site and for American Psychiatric Association site;
* MemberPage: displays information for each individual member, including grades for exam, percent total and letter grade. Also displays a comment from professor;
* Login: allows user to login to the application. Based on UserName/Password, the user will be redirected either to professor page or to Student page;
* RegisterUser: allows users to create an account for this application, create his/her username and password;

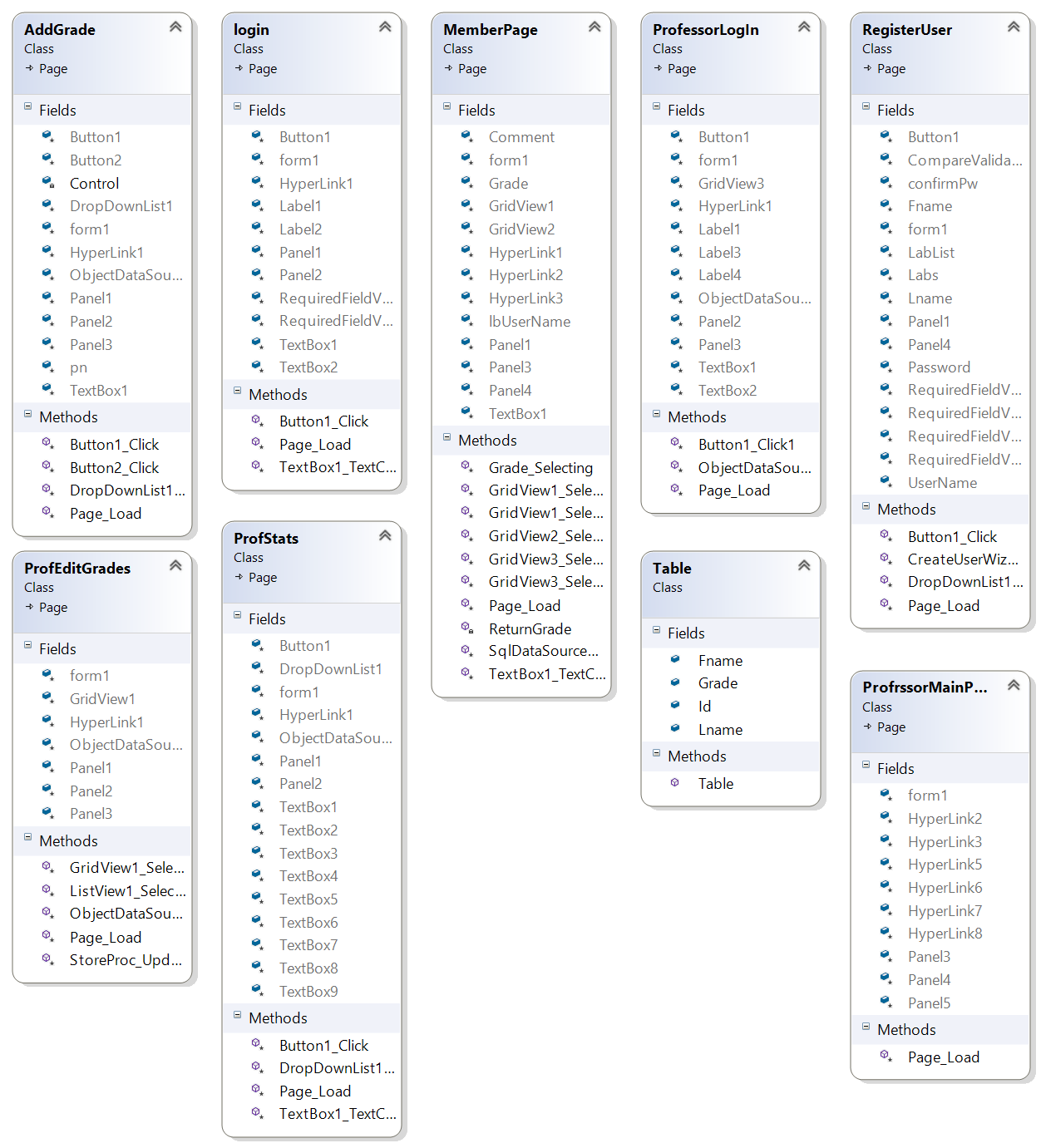
UI

DAL

Service/

Repo

Business



**Future Features**

In the next version of this application the following features/improvements are expected:

1. Implementation of Picture class to allow students to post and view their pictures and to help the professor to remember the students faster;
2. Implementation of the file exchange: this will let the professor to upload a study guide for each exam and ensure that every registered student has access to it.
3. Creation of the new logIn group for Lab instructors, which will be based on professor’s functionality only in regards to each individual lab.
4. Improvement of basic user input and enforcement of input rules.
5. Improvement of error handling and fixing error messages when needed.