System Analysis

**Record-Keeping System for a Further Education College**

Prepared by: Vladislavs Marisevs

For the attention of: Gary Cox

Candidate No: 1479135H

Class: Network and Software Systems

Table of Contents

[Introduction 3](#_Toc355741641)

[Present System 3](#_Toc355741642)

[New System requirements 4](#_Toc355741643)

[Project development phase 5](#_Toc355741644)

[Gantt chart diagram 5](#_Toc355741645)

[Use Cases 8](#_Toc355741646)

[Edit Personal Data 8](#_Toc355741647)

[Add or Edit Tutor Personal Information 8](#_Toc355741648)

[Add or Edit Student information 9](#_Toc355741649)

[Add or delete student class groups 9](#_Toc355741650)

[Add or edit class groups 10](#_Toc355741651)

[Add or delete teacher’s group modules 11](#_Toc355741652)

[Edit Course 11](#_Toc355741653)

[Add or Edit Module 12](#_Toc355741654)

[Add or Edit exams or assignments 12](#_Toc355741655)

[Add or edit student results 13](#_Toc355741656)

[Initial Class Diagram 13](#_Toc355741657)

[Classes 13](#_Toc355741658)

[Entity Relationship Diagram 17](#_Toc355741659)

[Table Teachers 17](#_Toc355741660)

[Table Students 18](#_Toc355741661)

[Table Course 19](#_Toc355741662)

[Table Groups 20](#_Toc355741663)

[Table Subjects 20](#_Toc355741664)

[Table Subject Exam 21](#_Toc355741665)

[Table GroupTeachersSubjects 22](#_Toc355741666)

[Table GroupStudents 22](#_Toc355741667)

[Table Student Results 23](#_Toc355741668)

[Database Tables and Relationships 24](#_Toc355741669)

[Teachers Table 24](#_Toc355741670)

[Course Table 24](#_Toc355741671)

[Group Table 25](#_Toc355741672)

[Group Students 25](#_Toc355741673)

[Group Teachers Subjects Table 26](#_Toc355741674)

[Student Results 27](#_Toc355741675)

[Student Table 28](#_Toc355741676)

[Subject Exam 29](#_Toc355741677)

[Subjects Table 30](#_Toc355741678)

[ER Diagram 31](#_Toc355741679)

[Queries and forms 32](#_Toc355741680)

[State Diagram 35](#_Toc355741681)

[Interface design and user’s manual 37](#_Toc355741682)

[Personal details form 39](#_Toc355741683)

[Add edit tutors form 40](#_Toc355741684)

[Add edit student form 41](#_Toc355741685)

[Add edit student to group 42](#_Toc355741686)

[Add edit course 43](#_Toc355741687)

[Add edit group 44](#_Toc355741688)

[Add edit modules 45](#_Toc355741689)

[Add edit exams 46](#_Toc355741690)

[Add edit teachers group modules 47](#_Toc355741691)

[Add edit student results 48](#_Toc355741692)

[Program extras 50](#_Toc355741693)

[Test statements 52](#_Toc355741694)

[Test statement 1 52](#_Toc355741695)

[Test statement 2 54](#_Toc355741696)

[Test statement 3 57](#_Toc355741697)

[Test statement 4 58](#_Toc355741698)

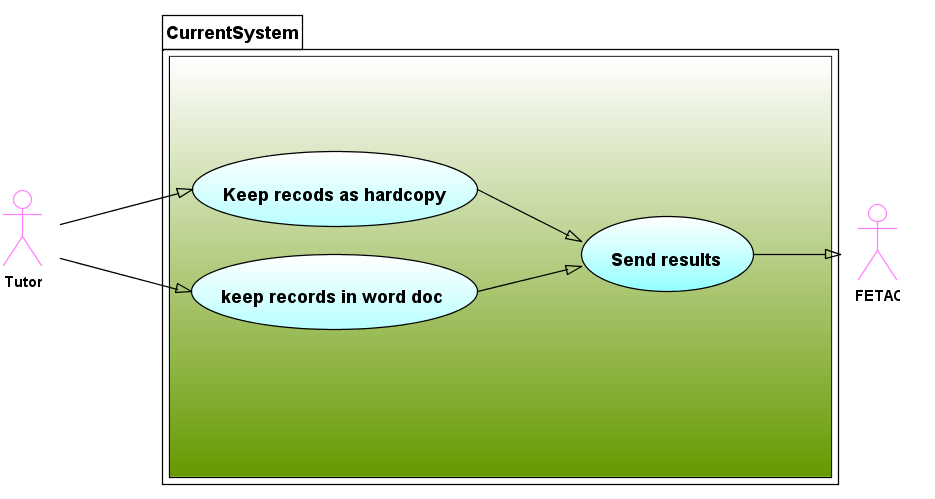
[Test statement 5 60](#_Toc355741699)

# Introduction

Further Education College (FEC), offering FETAC awards, has increased its student intake steadily over the past ten years. Record- keeping and administration time overheads have increased enormously and record-keeping and administration work are carried out on a more or less individual teacher basis and in an individual style. My assignment is dedicated to standardize this record keeping system for college.

## Present System

Currently there is no such standardized record keeping system, that’s why tutors keep their student records in individual format. Some teachers keep records in paper of hardcopy format, others in electronic form. The location of these records usually knows only particular tutor. Volume of records is approximately 500 per year. The Further Education Colleges are required to provide data to the Department of Education in the form of a database detailing all class groups, student names and numbers. This database is developed using the Facility Administration application and a copy of this database is available.

Use Case of current system:

This use case diagram provides an overview of the current system. There are 2 possible scenarios. In first scenario tutor stores student results in a hard copy base and later sends those records to FETAC. In second scenario tutor stores student results in a softcopy document and then later sends those to FETAC.

Problems with current system are:

* Records kept in different formats and locations. This method makes them less secured and makes more problems for students to overview the results.
* Tutors do not store any backups of results. Softcopy documents can get corrupted through hardware failure. Hard copy documents can get lost or damaged without a chance to restore.
* This system makes difficult for tutor to share information and collaborate on the management if student data.
* Current system makes impossible to pass information about student results if tutor needs to be changed during the year.

## New System requirements

Application that allows tutors enter exam or assignment results for a single student or for a class group. The system must have user friendly interface with the minimum amount of input required.

* System must me secured with login and password.
* Only tutor must have access to update and save student records.
* All records should be stored in one location.

# Project development phase

## Gantt chart diagram

Before starting the system development I should do the plan. I have listed all the tasks and set them estimated time.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tasks** | **Start Date** | **Duration (days)** | **End Date** |
| Read brief | 16 January 2013 | 1 | 16 January 2013 |
| Analyze system requirements from brief | 18 January 2013 | 5 | 22 January 2013 |
| Gather facts about similar applications | 23 January 2013 | 5 | 27 January 2013 |
| Do a plan about functionality of system | 28 January 2013 | 5 | 01 February 2013 |
| Imagine a system design | 02 February 2013 | 1 | 02 February 2013 |
| Go back and check all requirements from brief with plan | 03 February 2013 | 5 | 07 February 2013 |
| Do a database diagram | 08 February 2013 | 2 | 09 February 2013 |
| Holydays | 10 February 2013 | 7 | 16 February 2013 |
| Creating database tables | 17 February 2013 | 1 | 17 February 2013 |
| Creating usefull queries | 18 February 2013 | 3 | 20 February 2013 |
| Populating database with some information | 21 February 2013 | 1 | 21 February 2013 |
| Compare the database with system functionality | 22 February 2013 | 1 | 22 February 2013 |
| Design the forms and their methods | 23 February 2013 | 2 | 24 February 2013 |
| Creating classes that will be usefull | 25 February 2013 | 3 | 27 February 2013 |
| Creating administrator forms | 28 February 2013 | 3 | 02 March 2013 |
| Setting the Student and Teachers specifications | 03 March 2013 | 1 | 03 March 2013 |
| Testing Graphical User Interface | 04 March 2013 | 2 | 05 March 2013 |
| Checking output data from GUI with database input | 06 March 2013 | 1 | 06 March 2013 |
| Creating connection between database and graphical user interface | 07 March 2013 | 3 | 09 March 2013 |
| Testing operability | 11 March 2013 | 6 | 16 March 2013 |
| Fixing bugs in the whole system | 18 March 2013 | 6 | 23 March 2013 |
| Tidying the code and decrypting comments | 25 March 2013 | 3 | 27 March 2013 |
| Fixing forms desing | 28 March 2013 | 2 | 29 March 2013 |
| Holydays | 30 March 2013 | 9 | 07 April 2013 |
| Writing reports | 08 April 2013 | 11 | 18 April 2013 |
| Submitting the assignment | 19 April 2013 | 1 | 19 April 2013 |

To make my tasks easier to read I made Gantt chart. I was using Microsoft Excel to create this diagram. I got sick during my planned holidays. And my project was delayed for couple of days.

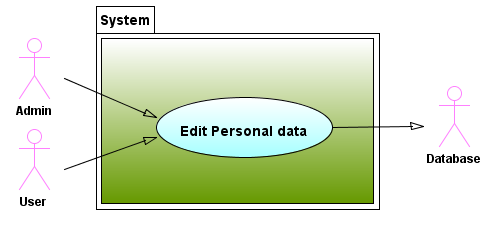
Gantt chart version on 30 of January

Gantt chart version on 1 of May

Technically project was going well during all time, but unexpectedly I got sick. That was the reason to change my deadlines for last two tasks.

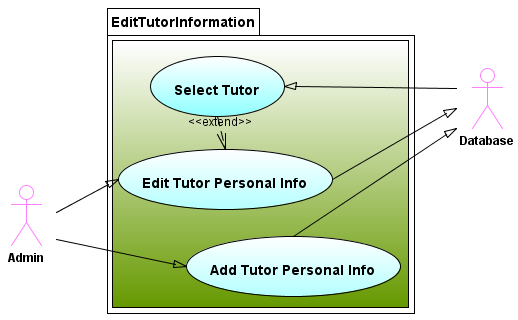
## Use Cases

### Edit Personal Data



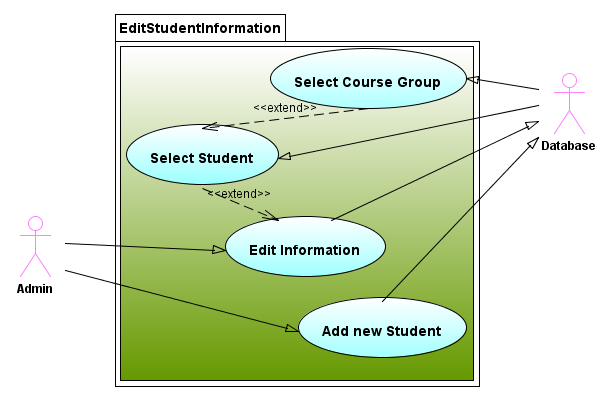
On this diagram we can see that administrator and user can edit their personal information and it stores in database.

### Add or Edit Tutor Personal Information



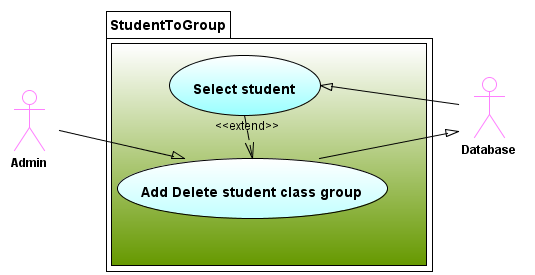
On this diagram we can see that, only administrator have rights to add new tutor to database, or to edit existing one. Simply select him from a list.

### Add or Edit Student information



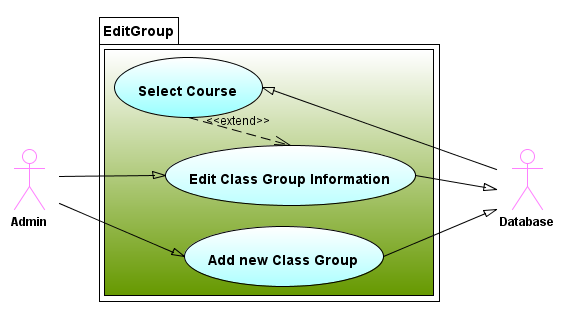
On this diagram we can see, that only administrator have rights to add new student or edit existing one. First of all he should select a course group and then he can select a student from this course group.

### Add or delete student class groups



On this diagram we can see, that only administrator can add or delete student class group. First of all admin should select a student using his pps number and then he is available to change information about student’s class groups.

### Add or edit class groups



On this diagram we can see, that only administrator can add or edit course groups. To edit course group, admin must select a course from list and then select a year of a course. To add new, admin must select a course set year of this group.

### Add or delete teacher’s group modules

### C:\Users\Evilguy\Desktop\assignment\Doc\0007.png

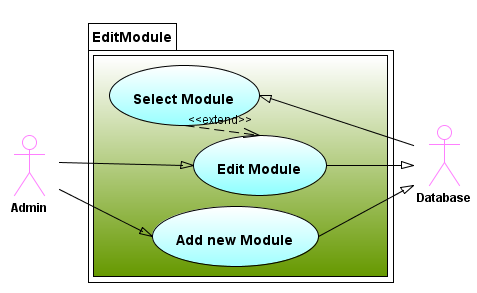
On this diagram we can see, that only administrator can add or remove teacher’s group modules. To add a relationship between tutor, class group and module, admin should select a teacher from list, select a class group from list and select a module. To remove a relationship, administrator must select a tutor and select class group with module.

### Edit Course

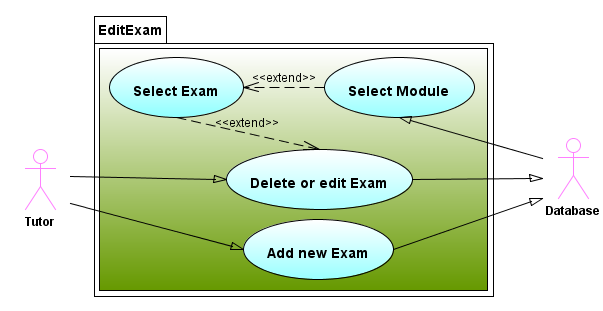
### C:\Users\Evilguy\Desktop\assignment\Doc\0008.png

On this diagram we can see, that only administrator can add new or edit course.

### Add or Edit Module

 On this diagram we can see, that only administrator can add new or edit module.

### Add or Edit exams or assignments



On this diagram we can see, that only administrator can add new, edit or remove exam from selected module.

### Add or edit student results

### C:\Users\Evilguy\Desktop\assignment\Doc\0011.png

On this diagram we can see, that tutor and administrator can add or edit student results. First of all user should select a class group, then module, exam. Before adding result user must select a student.

## Initial Class Diagram

### Classes

In this part of a document I will outline the technical details that will use to create the application. Class diagram identify the main objects that I will use in my software. To identify classes I listed all the nouns that describe the system:

* Course
* Class Group
* Exam
* Exam Result
* Student
* Module
* Tutor

Course – a course is needed in the system, because college does only specific set of courses. And to make this application flexible we need course class.

Class Group – has course specifications, but sometimes colleges don’t do the course, if they don’t have enough students. But after few years college can get back to this course, and database would have all information about it.

Exam – FETAC has a standard of 2 assignments and 1 exam, but what if this standard will be changed in next few years? To make this software more efficient for next 5-7 years, I added few extras.

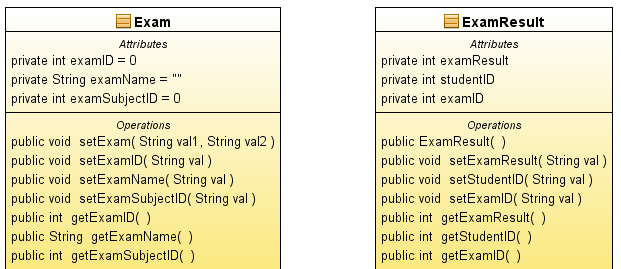
Exam Result – should keep information about student and his exam result. This is a main requirement of this system.

Module (subject) – is needed in the system to define student result module.

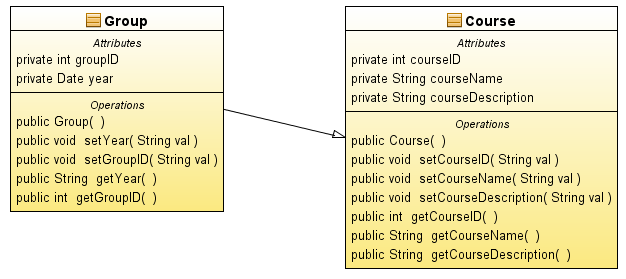
Student – is a person, who studies in this college. So college must have the main information about student. And all that information should be stored in database.

Tutor- is a person, who could have an access to database. There could be 2 types of tutors, administrators and users. Users could operate only with their student group results, and their private information. But admin will have rights to configure whole workability of this database.

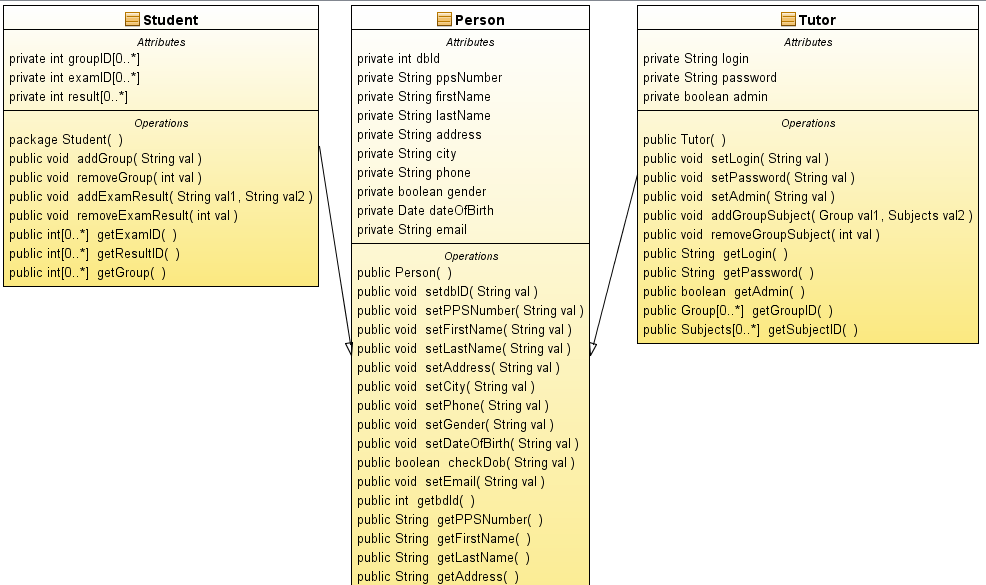
Exam and Exam result Classes



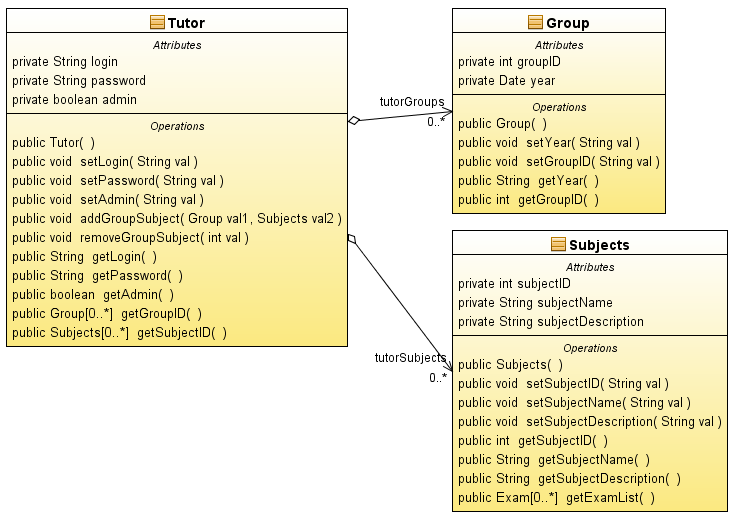
Group and Course Classes with relationship, that one group can have only one course.



Person class is a parent class to Student and Tutor class. This means that Tutor class and Student class inherit all methods from Person class.

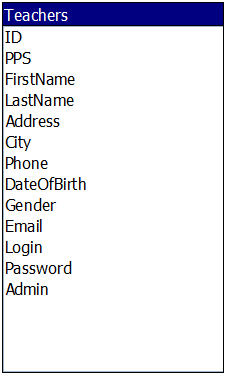


On this class diagram we can see that Tutor can have a set of groups and a set of modules.

Entity Relationship Diagram

This diagram is used by system analyst to identify the entities in the system and describe the relationships between entities. Entities are objects that are important to the system, have at least one attribute, occur more than once and each occurrence must be uniquely identifiable. Because each entity must be uniquely identifiable it is important that every entity has a code or primary key.

### Table Teachers



ID – in order to create stable working database, I was using a primary key in id field. Data type for this field is Auto Number, this means that it is generated automatically. This fixes the bug with unique field errors.

PPS – all citizens of Ireland must have Personal Public Service Number. It should be unique for each person. So it is better to do this field as indexed. Personal Public Service Number contains 7 digits and 2 letters.

FirstName – each tutor have name that stored in teachers table. (as text format)

LastName – each tutor have surname that stored in teachers table (as text format)

Address- each tutor have address that stored in teachers table. (as text format)

City – each tutor have city that stored in teachers table. (as text format)

Phone – each tutor have phone number that stored in teachers table. (as text format)

DateOfBirth – each teacher have date of birth that stored in teachers table (as date time format)

Gender – teachers gender is saved in Boolean format (true = male, false = female)

Email – each tutor have email that stored in teachers table. (as text format)

Login – each tutor have login that is used to enter the system. (as text format)

Password – to enter the system tutor must set a password. I was using text format, but to do a save application it is better to use MD5 format.

Admin – in the system there is two types of users, administrator and user. If user have administrator rights this = true. (Boolean format)

### Table Students

### C:\Users\Evilguy\Desktop\assignment\Doc\ER\008.png

ID – in order to create stable working database, I was using a primary key in id field. Data type for this field is Auto Number, this means that it is generated automatically. This fixes the bug with unique field errors.

PPS – all citizens of Ireland must have Personal Public Service Number. It should be unique for each person. So it is better to do this field as indexed. Personal Public Service Number contains 7 digits and 2 letters.

FirstName – each student have name that stored in student table. (as text format)

LastName – each student have surname that stored in student table (as text format)

Address- each student have address that stored in student table. (as text format)

City – each student have city that stored in student table. (as text format)

Phone – each student have phone number that stored in student table. (as text format)

DateOfBirth – each student have date of birth that stored in student table (as date time format)

Gender – student gender is saved in Boolean format (true = male, false = female)

Email – each student have email that stored in student table. (as text format)

### Table Course

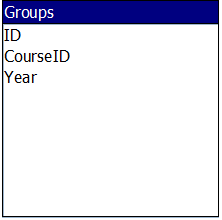
### C:\Users\Evilguy\Desktop\assignment\Doc\ER\001.png

ID – in order to create stable working database, I was using a primary key in id field. Data type for this field is Auto Number, this means that it is generated automatically. This fixes the bug with unique field errors.

CourseName – each course will have a name that will be stored on the course table.

Description – each course will have a short description that will be stored on the course table.

### Table Groups

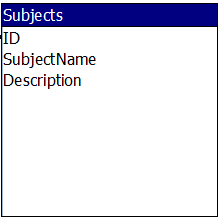


ID – in order to create stable working database, I was using a primary key in id field. Data type for this field is Auto Number, this means that it is generated automatically. This fixes the bug with unique field errors.

CourseID – each class group will have named course, and this field keeps course id.

Year – each class group will have a year that will be stored on the group table.

### Table Subjects

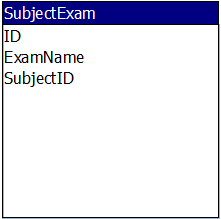


ID – in order to create stable working database, I was using a primary key in id field. Data type for this field is Auto Number, this means that it is generated automatically. This fixes the bug with unique field errors.

SubjectName – each module will have a name that will be stored on the subject table.

Description – each module will have a short description that will be stored on the subject table.

### Table Subject Exam

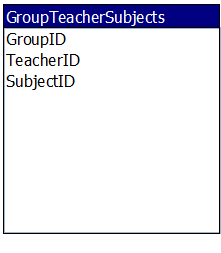


ID – in order to create stable working database, I was using a primary key in id field. Data type for this field is Auto Number, this means that it is generated automatically. This fixes the bug with unique field errors.

ExamName – each module will have an exam that will be stored on the subjectexam table.

SubjectID – this field will keep modules id, because sometimes exams must been relevant to module.

### Table GroupTeachersSubjects



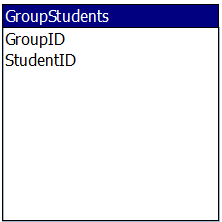
GroupID – this field is used to join Group Table.

TeacherID – this field is used to join Teachers Table.

SubjectID – this field is used to join Subject Table.

All fields in this table are made as combined primary key.

### Table GroupStudents

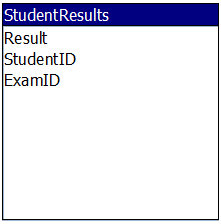


GroupID – this field is used to join GroupID.

StudentID – this field is used to join StudentID.

All fields in this table are made as combined primary key.

### Table Student Results



Result – this field store the exam result for selected student and exam.

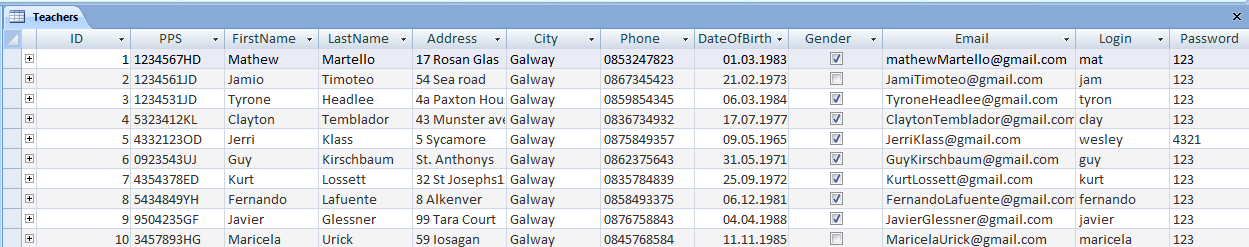
StudentID – this field is used to join StudentID.

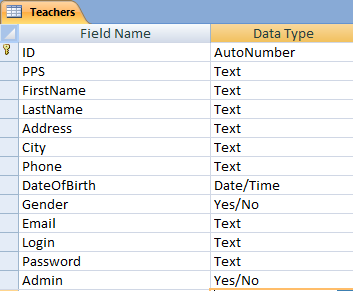
ExamID - this field is used to join ExamID.

## Database Tables and Relationships

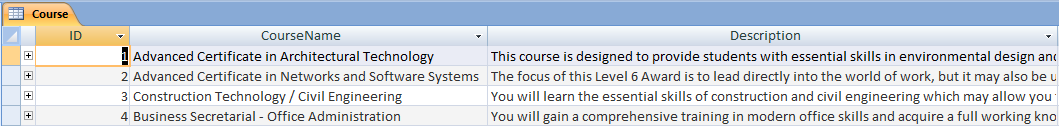
In this section I will provide an overview of database that I developed in Microsoft Access.

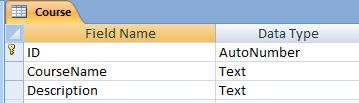
### Teachers Table



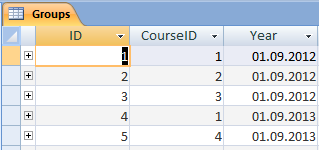


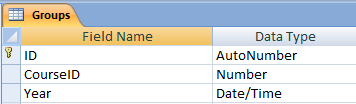
### Course Table





Group Table





### Group Students

### C:\Users\Evilguy\Desktop\assignment\tables\0031.png

### C:\Users\Evilguy\Desktop\assignment\tables\0032.png

### Group Teachers Subjects Table

### C:\Users\Evilguy\Desktop\assignment\tables\0041.png

### C:\Users\Evilguy\Desktop\assignment\tables\0042.png

### Student Results

### C:\Users\Evilguy\Desktop\assignment\tables\0051.png

### C:\Users\Evilguy\Desktop\assignment\tables\0052.png

### Student Table

### C:\Users\Evilguy\Desktop\assignment\tables\0061.png

### C:\Users\Evilguy\Desktop\assignment\tables\0062.png

### Subject Exam

### C:\Users\Evilguy\Desktop\assignment\tables\0071.png

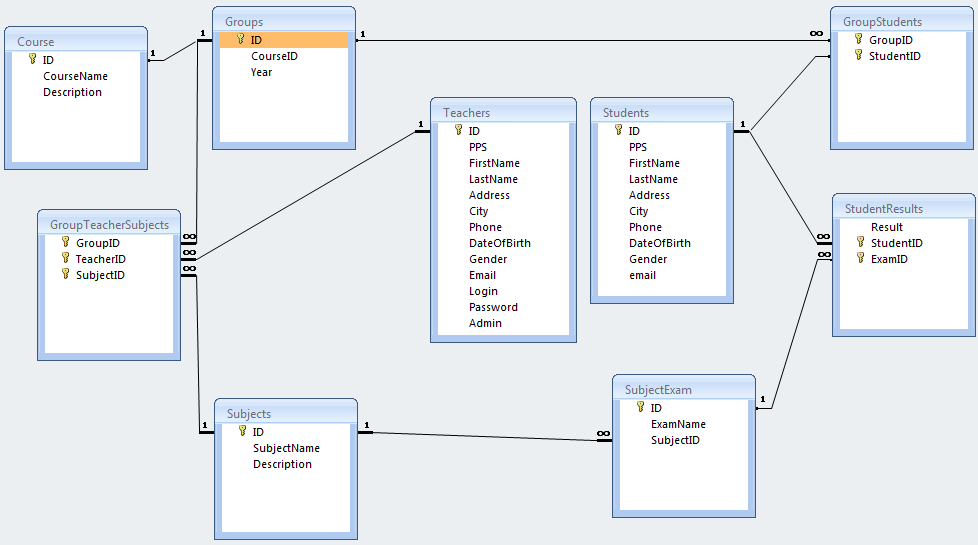
### C:\Users\Evilguy\Desktop\assignment\tables\0072.png

### Subjects Table

### C:\Users\Evilguy\Desktop\assignment\tables\0081.png

### C:\Users\Evilguy\Desktop\assignment\tables\0082.png

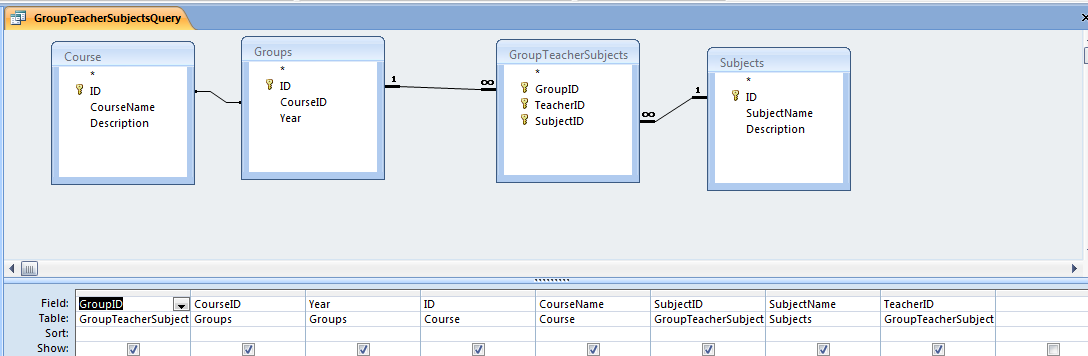
### ER Diagram



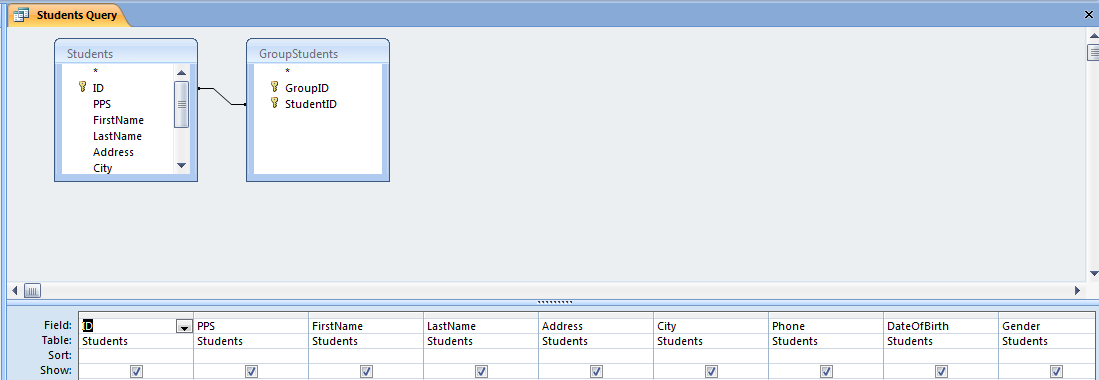
## Queries and forms

Queries and forms that I created in Ms Access Database, I was using to test my developed database workability. And to compare the results with java programmed application.

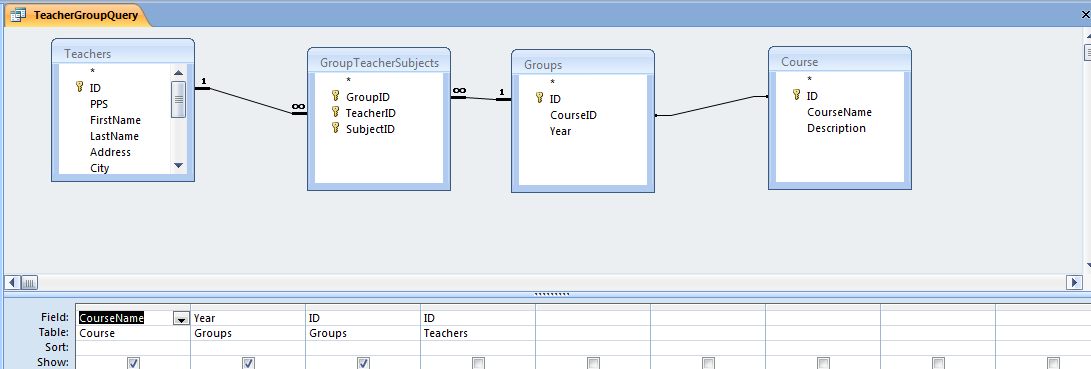
Group Teachers Subject Query that joins 3 tables together



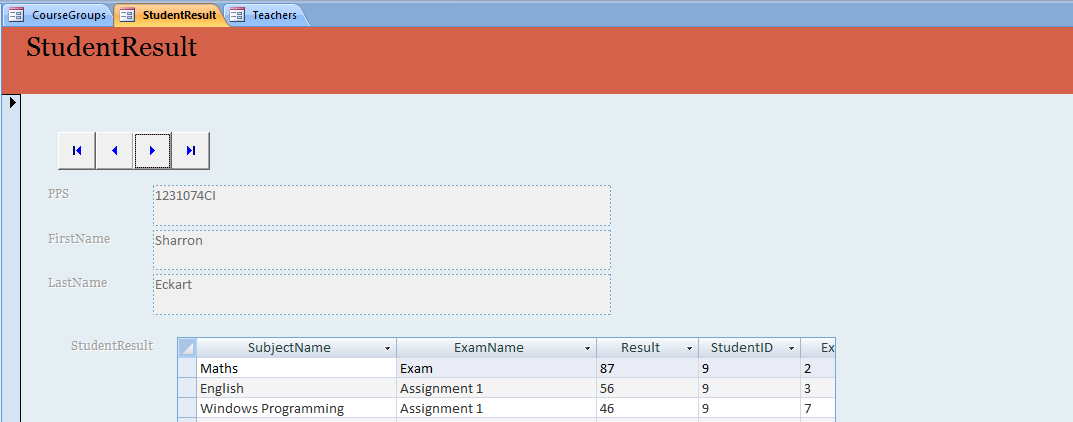
Group Students Query joins 2 tables together.



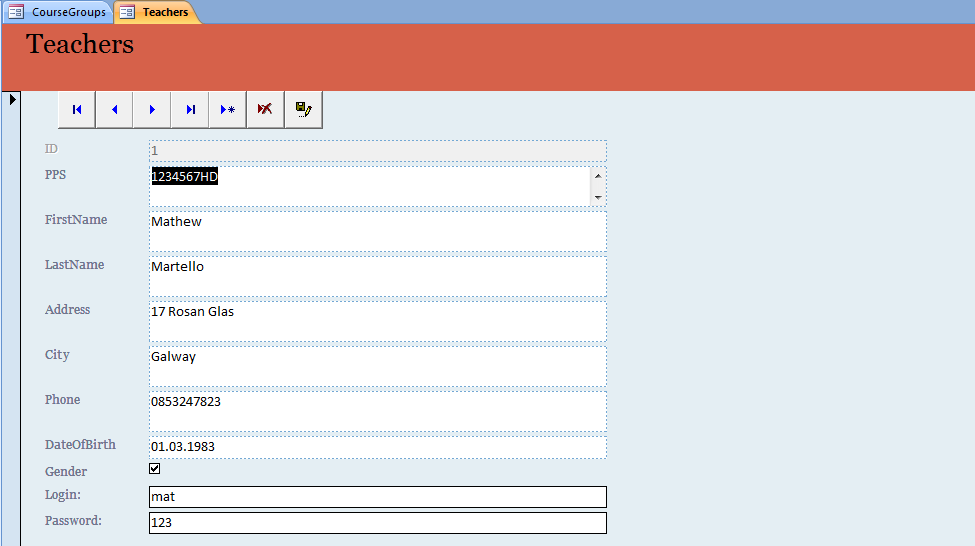
Teachers Group Table returns information about tutors groups



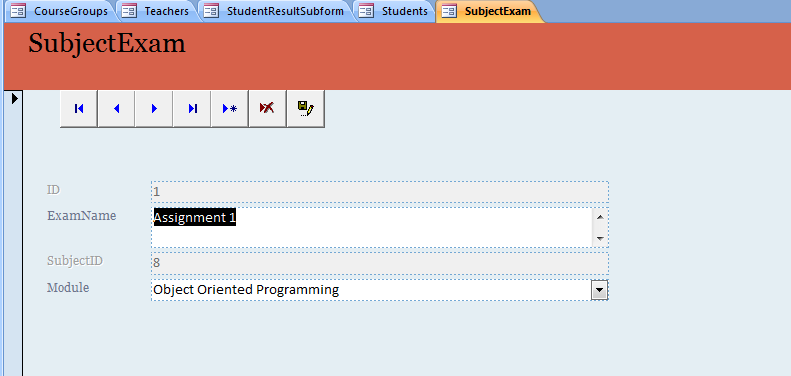
Student Result Form allows us to fill in the information about student results



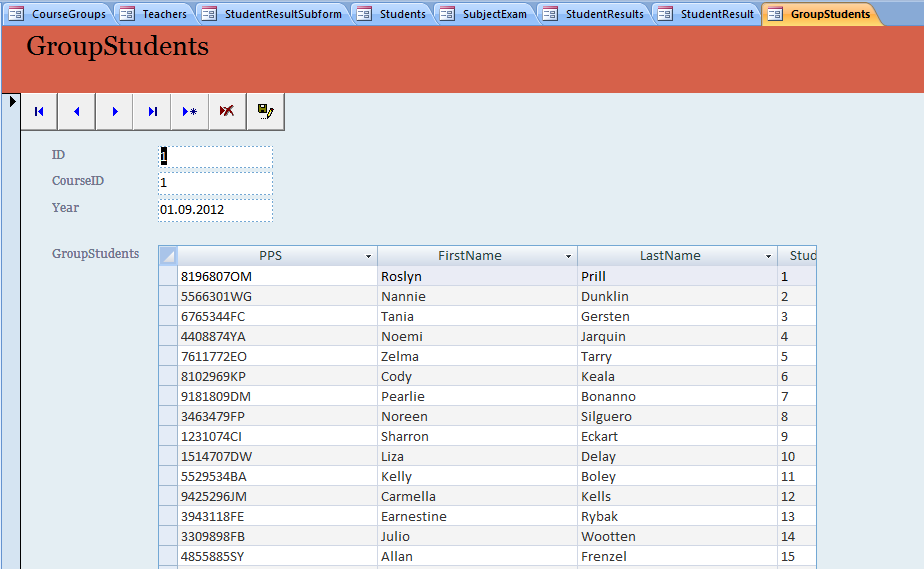
Teachers form allows us to edit tutor information



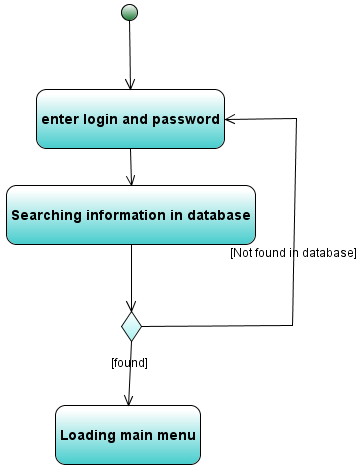
Module Exam form allows us to change exam information



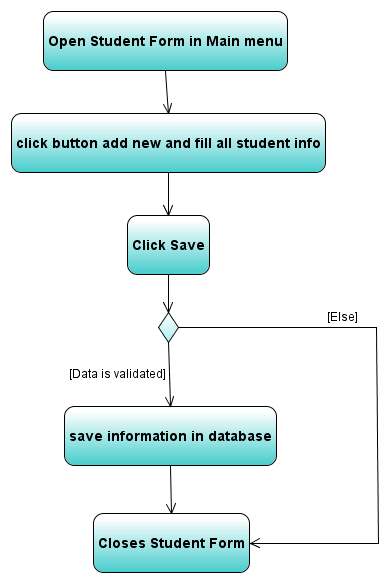
Student Group Form allows us to edit students in groups



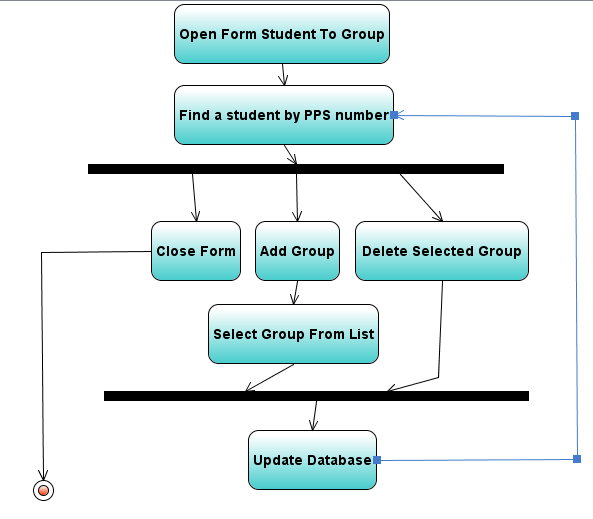
## State Diagram

****

On this diagram we can see how system is doing login method.



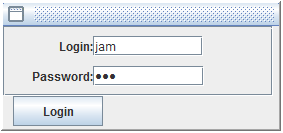
On this diagram user tries to update student information.

In this diagram we can see how works Student To Group Form.

# Interface design and user’s manual

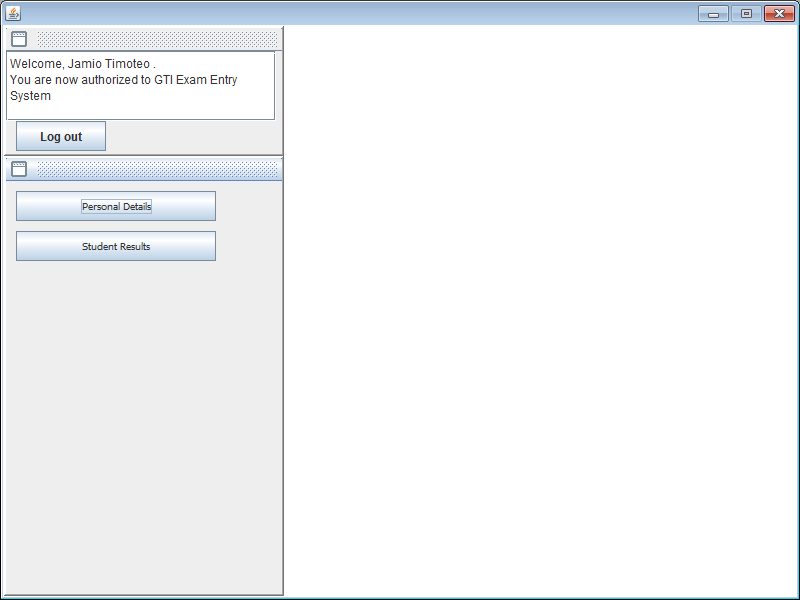
My project is dedicated to Galway Technical Institute record keeping system. I tried to do a user friendly design for this application. The resolution of this application is 800x600 px, so tutors can use this program on old computers as well.

To get into the system tutor must enter the valid password and login.

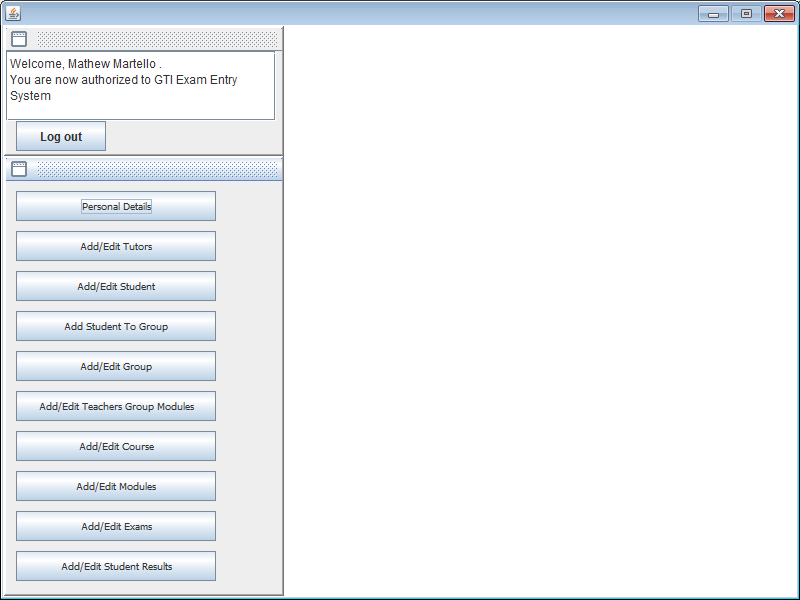


There are two types of users: Administrator and User. To login into the system you can use administrator **login: mat password: 123** and user **login: jam password: 123**

First type is tutor, he could do just simple things add or edit assignment results and edit his own information.

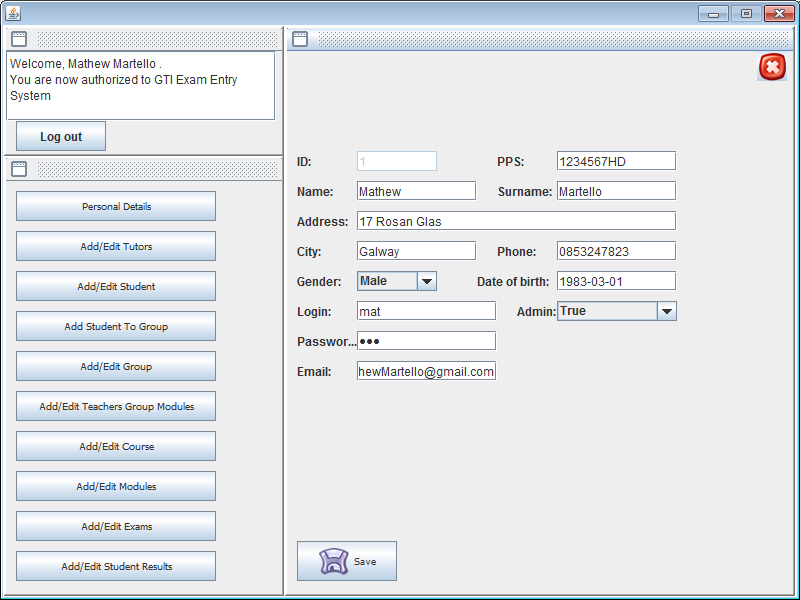


Second is administrator, he can do more than tutor. This user can change his information or add / edit other tutors information. Add or edit student information and set his class group or add one more class group for particular student. Add or edit college courses and add or edit class group. For example “Advanced Certificate in Networks and Software Systems” is a course, but class group needs year of studding. Add or edit modules. And make a relationship between tutor his group and his module (or subject) that he does for particular class group. The administrator can add/remove/ edit exams or assignments for modules. These users also can add/edit/view exam results for any class group and student.



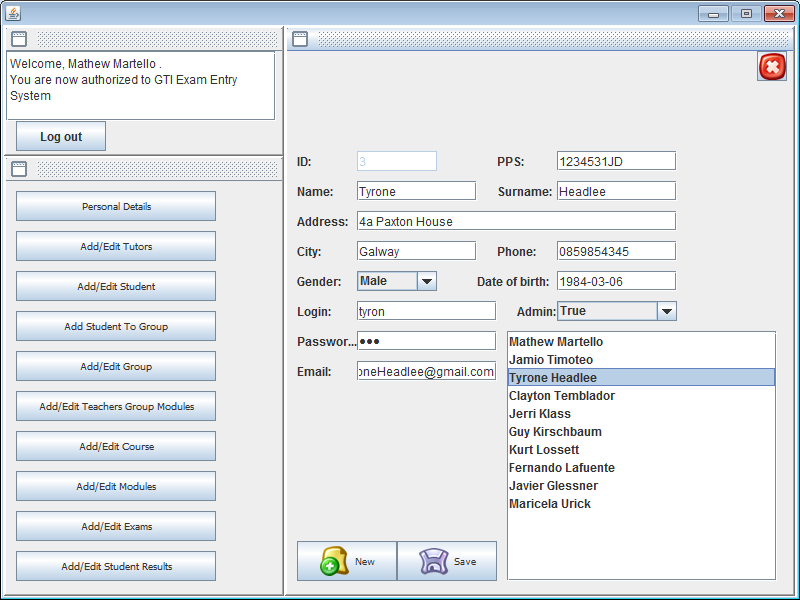
## Personal details form

This form shows the information about user that is logged in. He can change all information except database id, because it is generated automatically. To update information user should click save button and information will be saved in database.



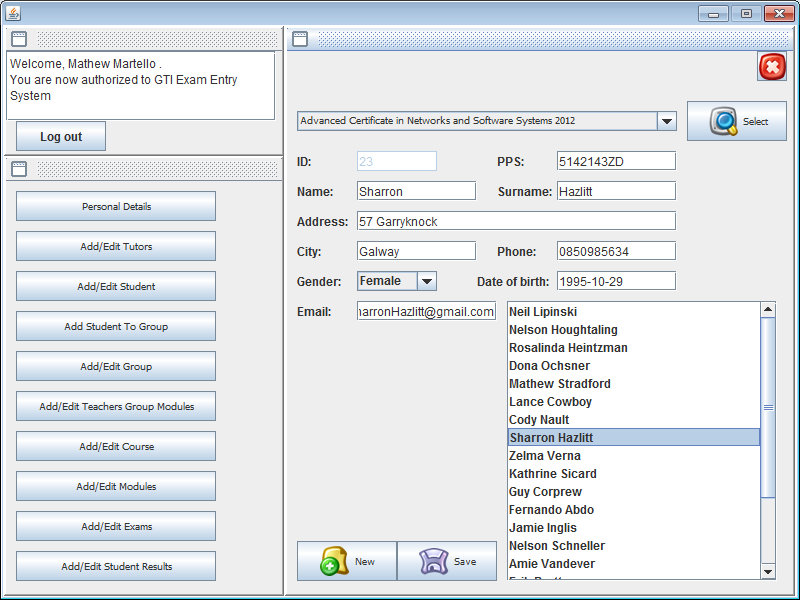
## Add edit tutors form

This form shows the information about current tutor, and user list that can be edited and saved into database. Also, we can add new tutor, to save new information, all text boxes should be filled in.



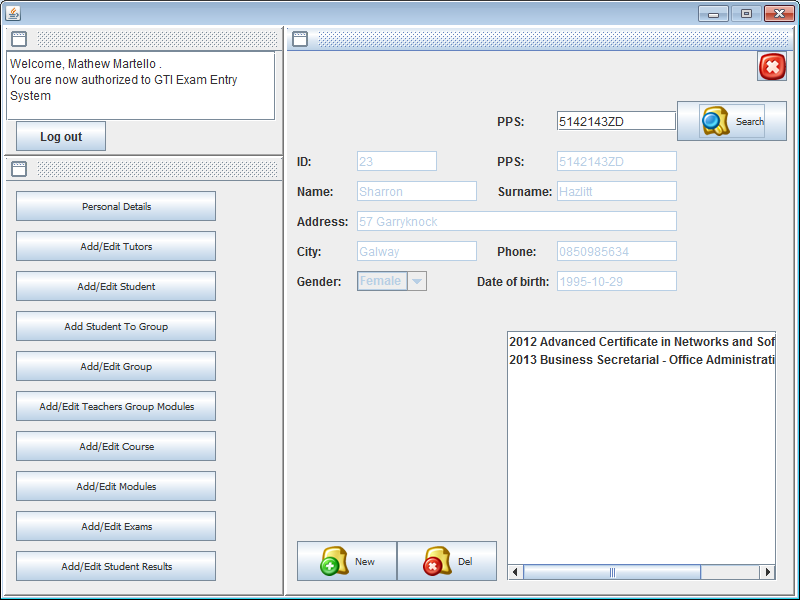
## Add edit student form

To get student information about existing student, administrator should select class group in combo box and click select button. Then student list will be filled in. And choosing the particular student the information about him will be updated on the screen. If administrator wants to add new student, he must select a class group first and then he can add new student to database.



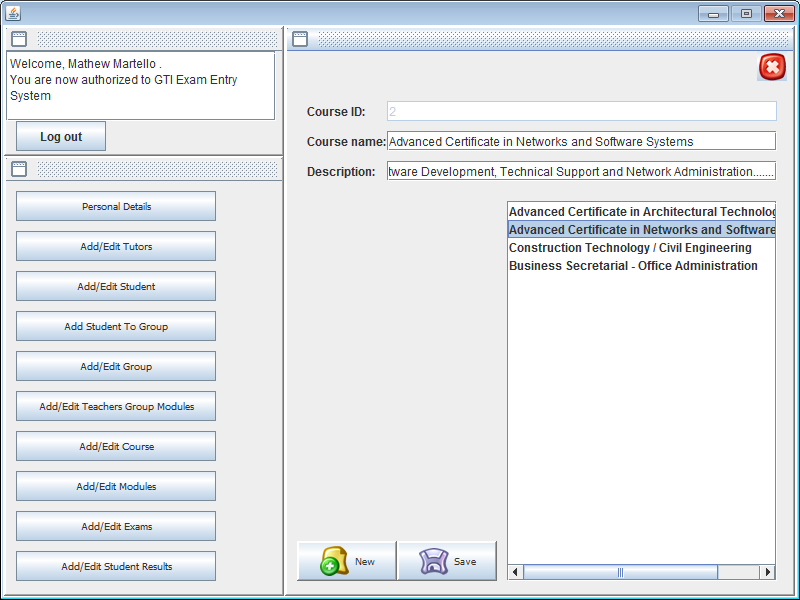
## Add edit student to group

Using this form administrator can add or remove student class group. But first he must choose a student, using his Personal Public Service Number. On this picture we can see that this student did 2 classes in this college.



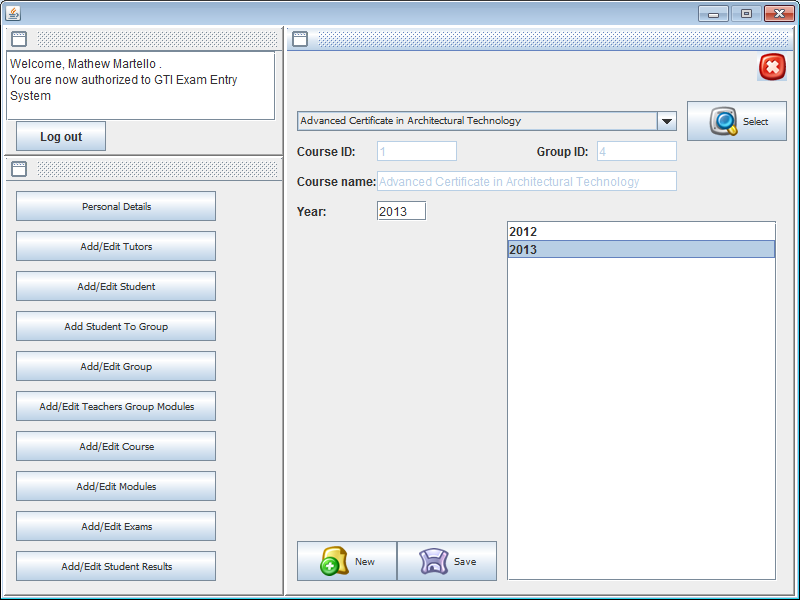
## Add edit course

Using this form administrator can add or edit college courses.



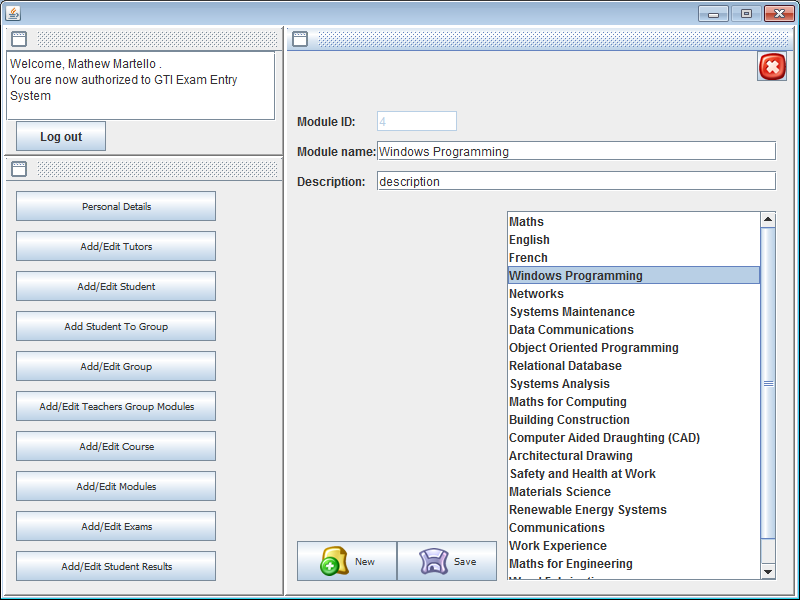
## Add edit group

Using this form administrator can add or edit class group. Class group have two variables, course identification and year of course. To view all class groups for particular course, user should choose it in combo box and press select button.



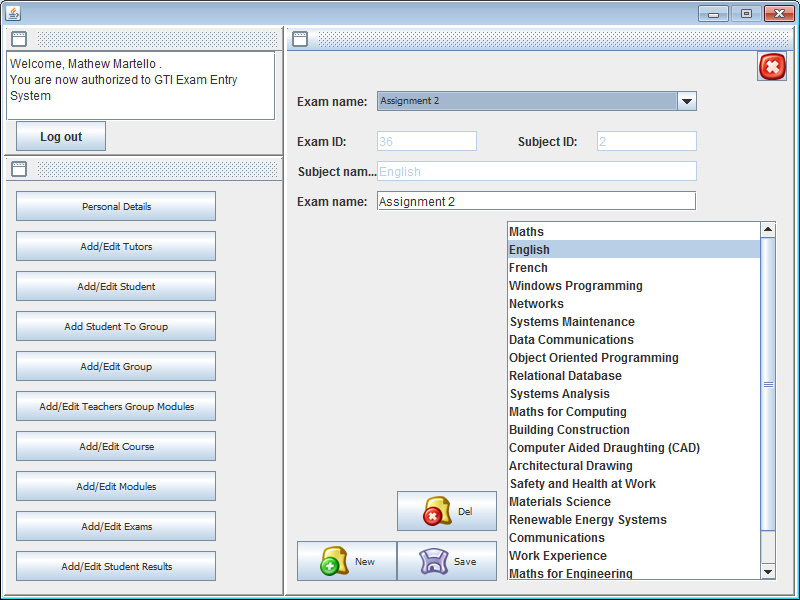
## Add edit modules

Using this form administrator can add or edit new module.



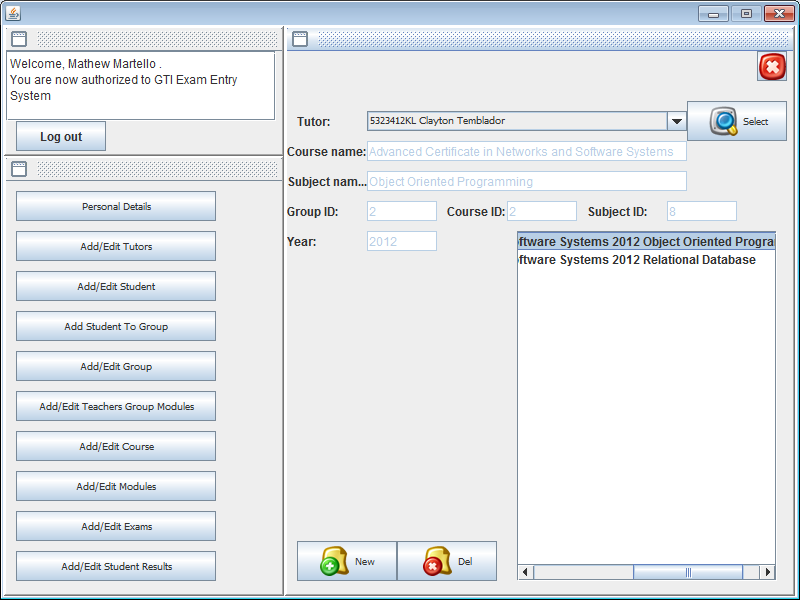
## Add edit exams

Using this form administrator can add, remove or edit exams for particular module. In the list user can see all the modules in college, he should choose one and combo box will be updated with this module exams or assignments.



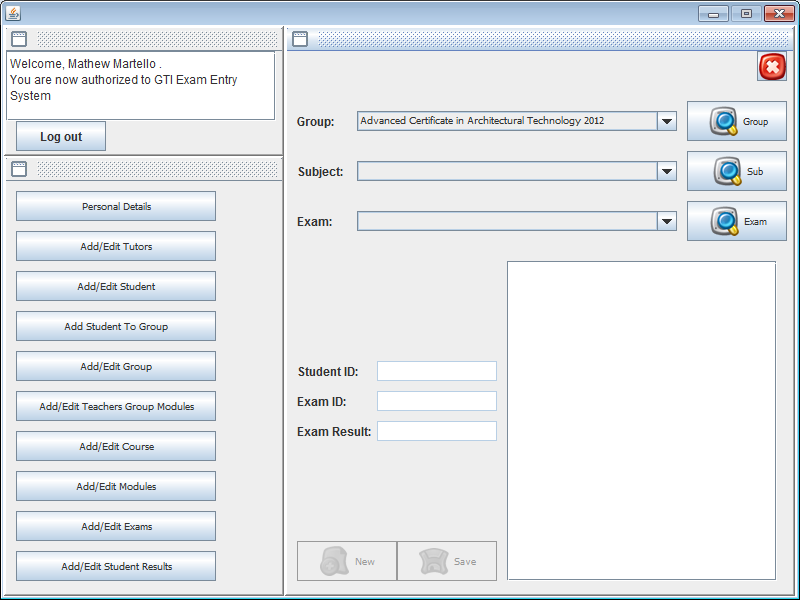
## Add edit teachers group modules

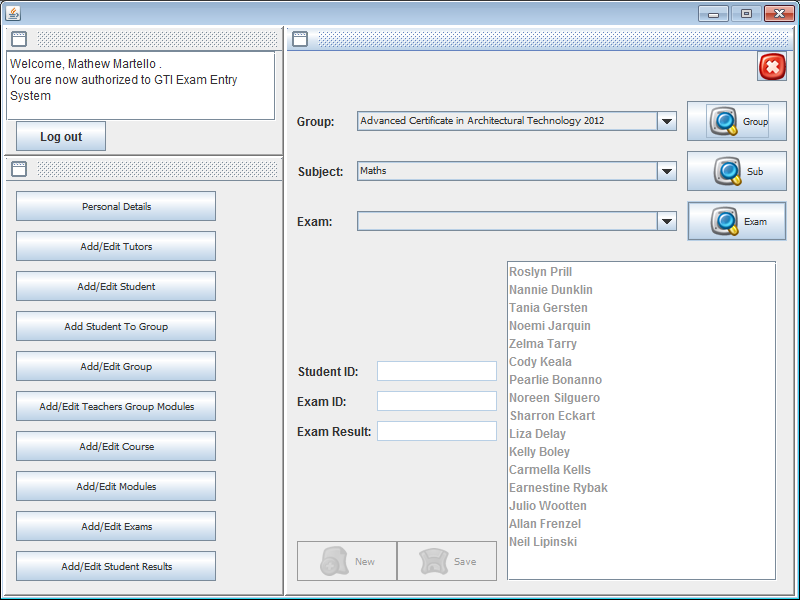
Using this form administrator can add or delete relationship between tutor class group and module. This means that tutor does a module for class group. So we can add to a tutor more groups or remove them. This gives an option to non administrator users add or edit student exam results for selected module.

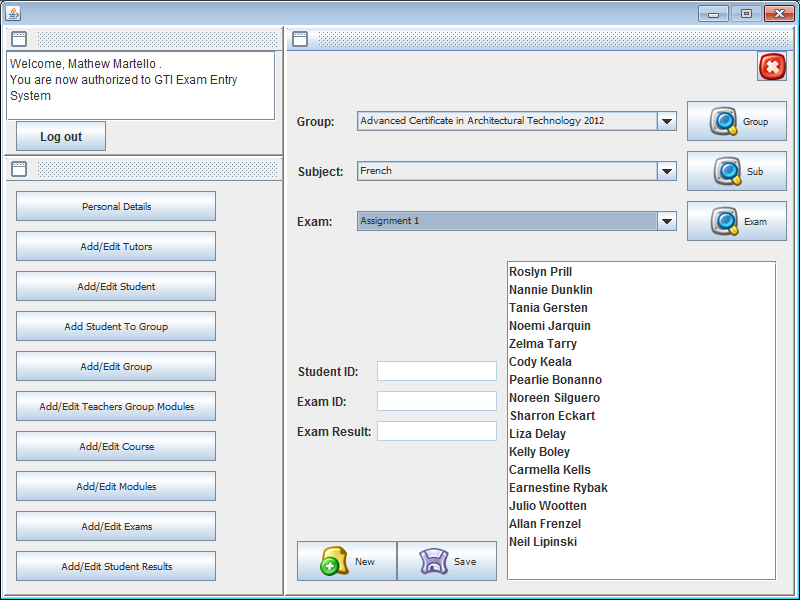


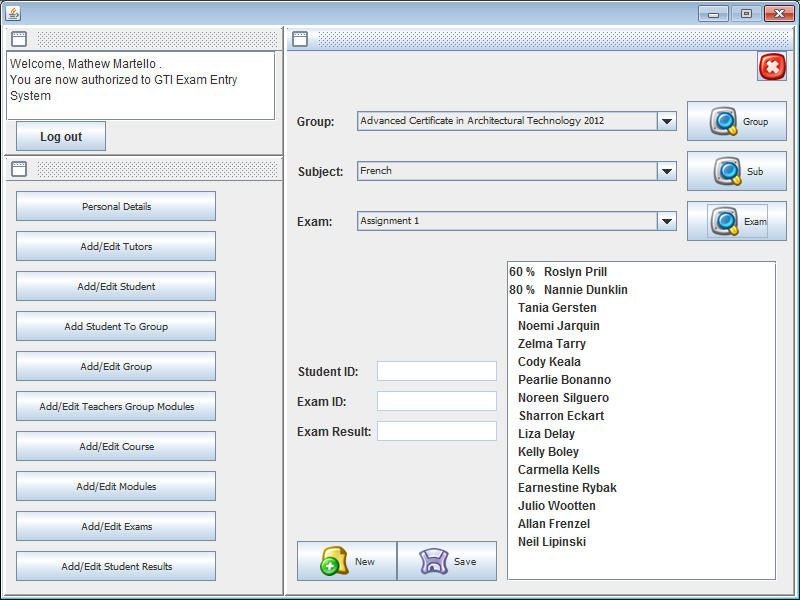
## Add edit student results

Using this form administrator or user can add or edit student results. First of all user should select a class group and click *Group* button, it confirms that the class group was selected. Then module should be selected and to confirm the selection user should click *Sub* button. And to view student results user should select an exam or assignment in combo box and click Exam button to confirm it. Now we can see list of students, if there is a result in percentage before student name and surname. That mean this student was rated. If there is no result before name, then there is no record with this student mark. This relationship makes result table keep only useful information about student.



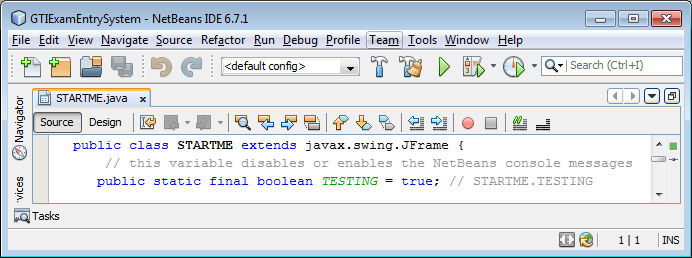






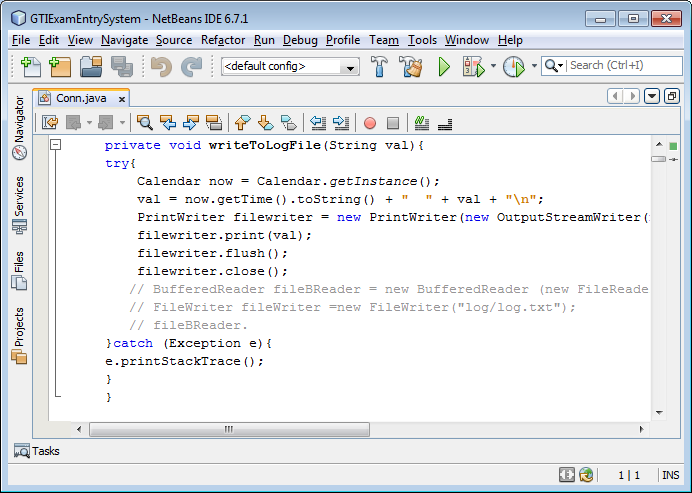
## Program extras

This is quite massive application and to debug an error sometimes took a lot of time. To reduce this time I set up a Boolean constant that enables (true) or disables (false) program messages that show the result set that it gets from database. This constant is in main form (STARTME.java) its name is TESTING.



To debug other errors I did a log file method. This method keeps a track of all login users, or login fails with current date and time. It saves information about wrong connections or connection errors.

Method:



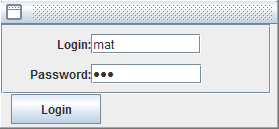
The log file is in \*\GTIExamEntrySystem.0.(version)\log\log.txt

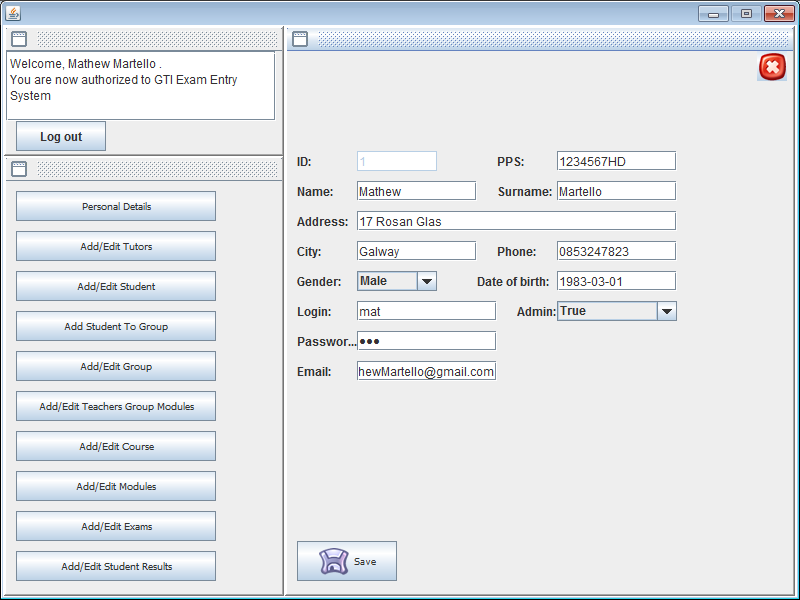
The database is in \*\GTIExamEntrySystem.0.213\db\ ExamEntrySystem.mdb

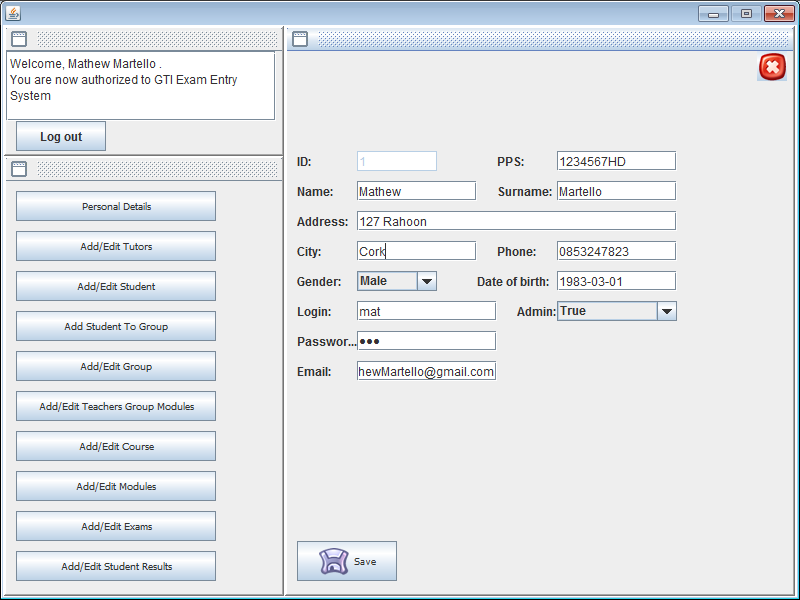
# Test statements

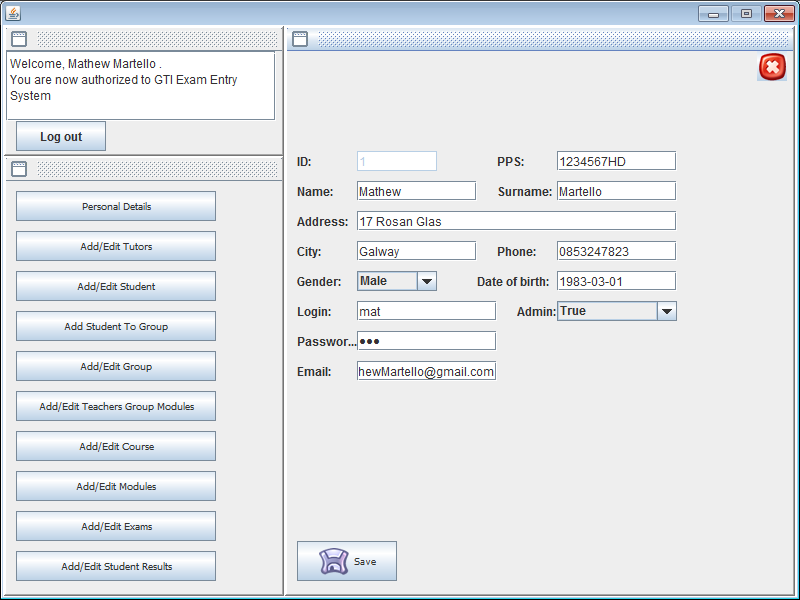
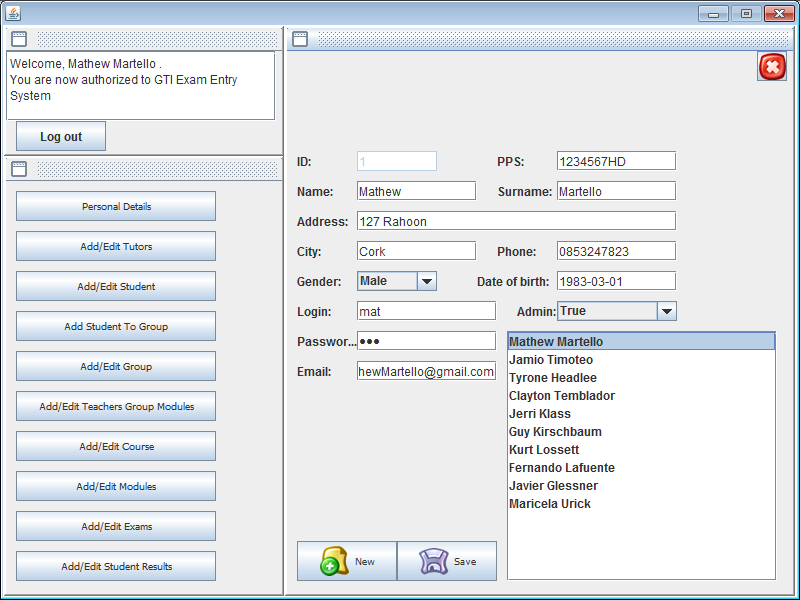
## Test statement 1

I will try to login into system as administrator. Change the address to new one in personal details form and change it back in tutor list form. (Login: mat Password: 123)

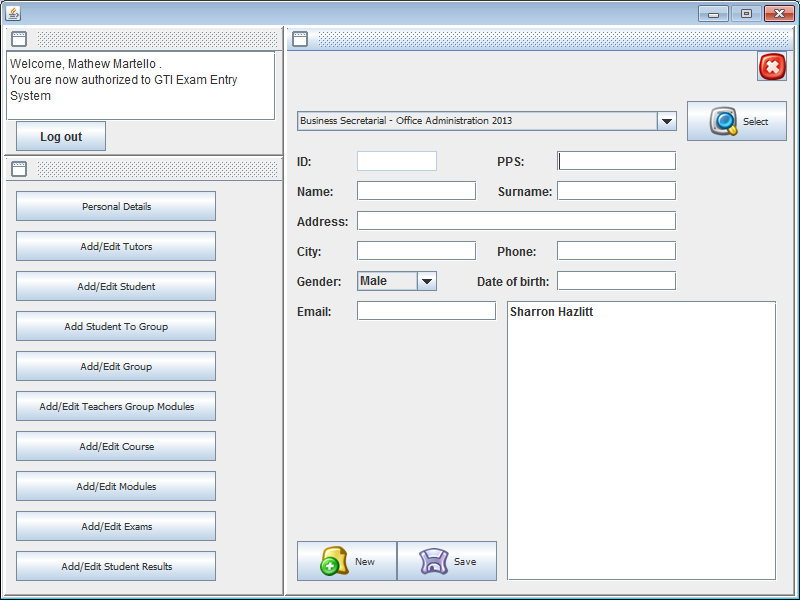


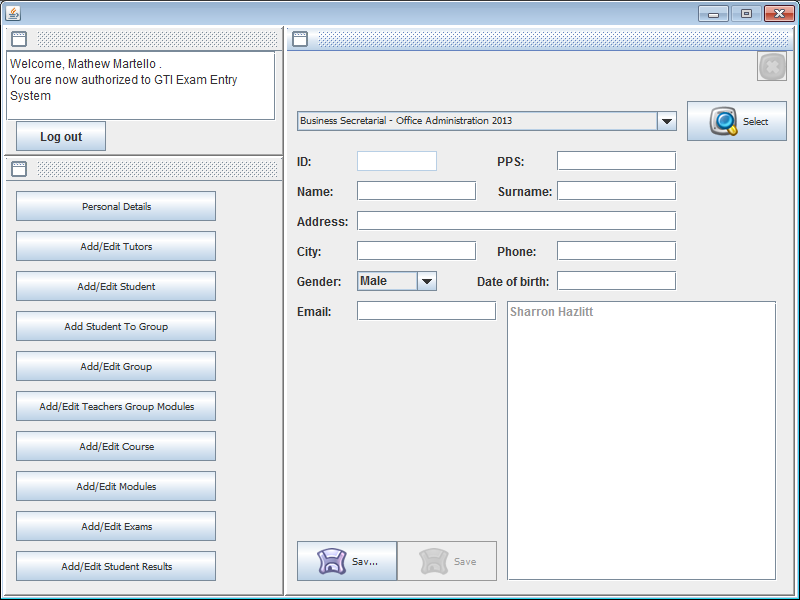


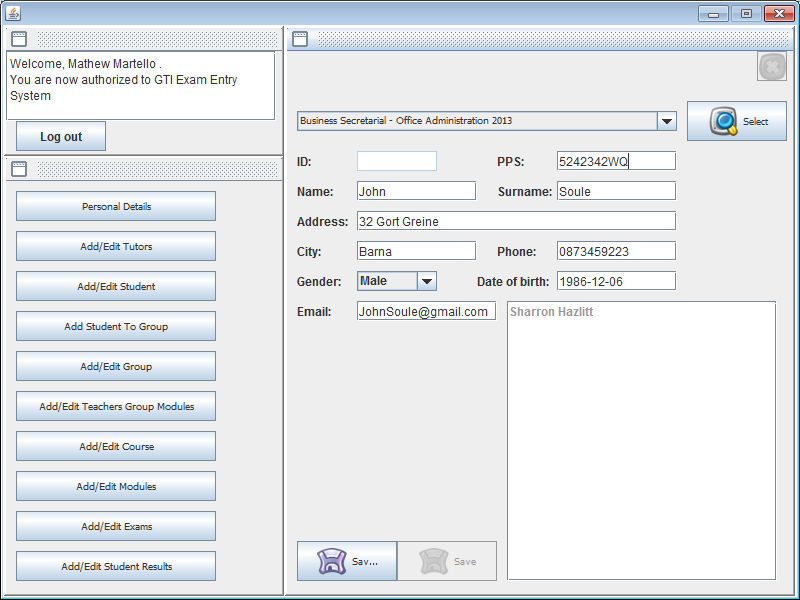




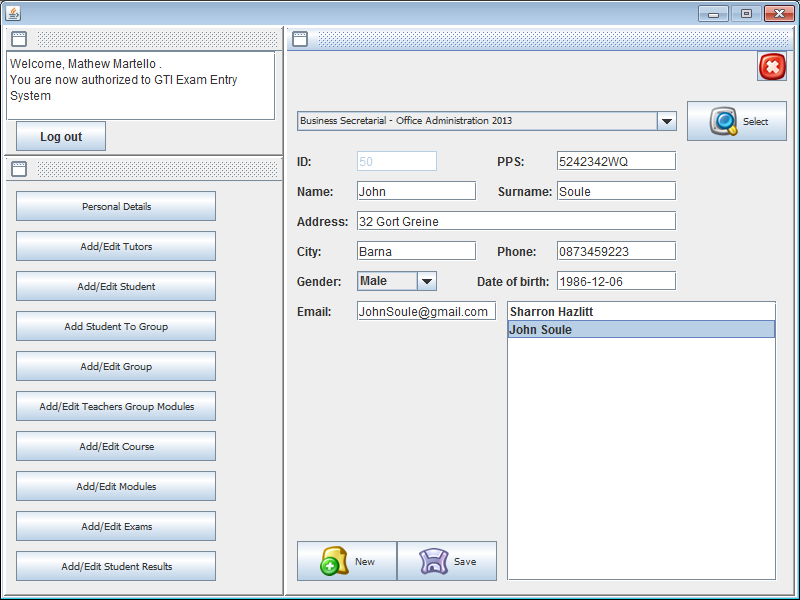
## Test statement 2

In this test statement I will try to create new student fill all information about him. 

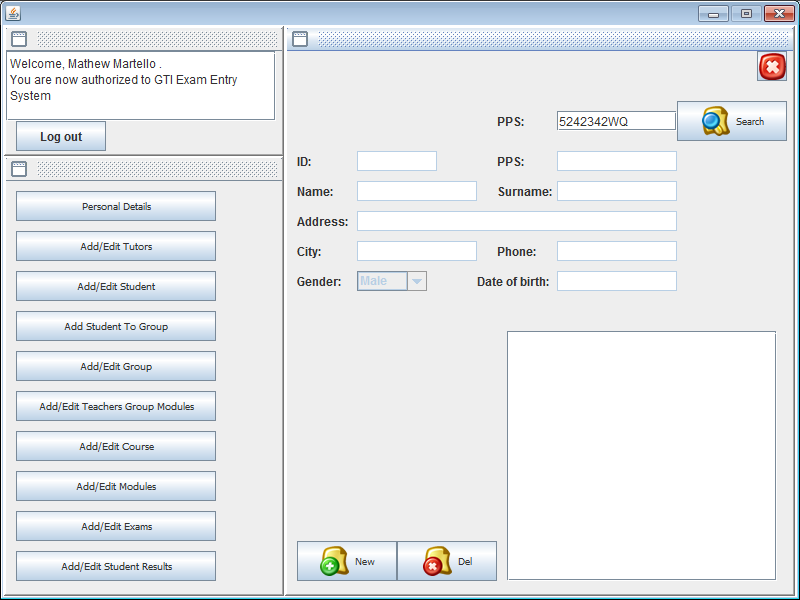




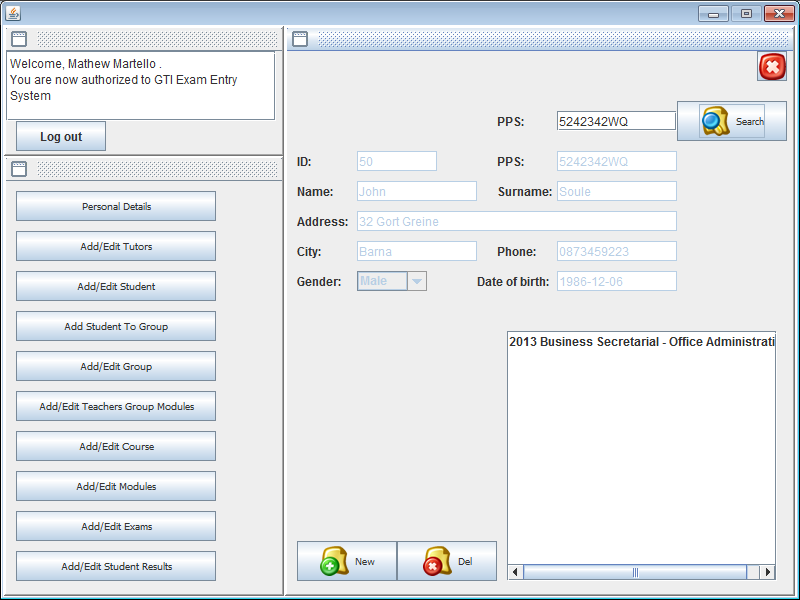
Check generated id from database.

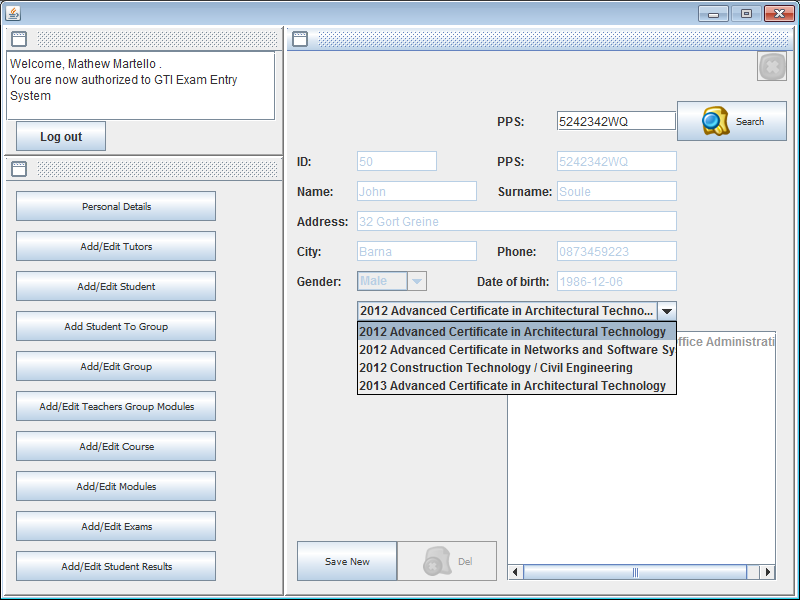


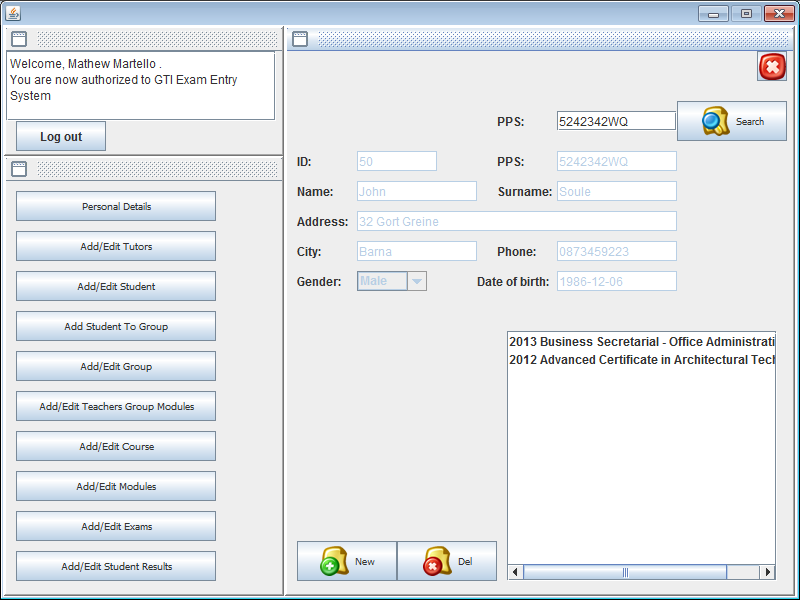
Open Student to group form and find new student using PPS number.



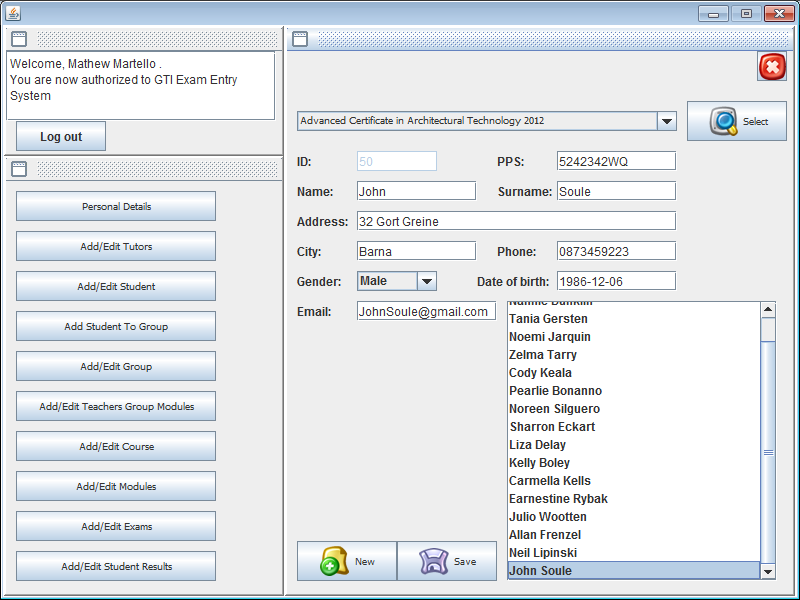
Add one more class group to a new student.





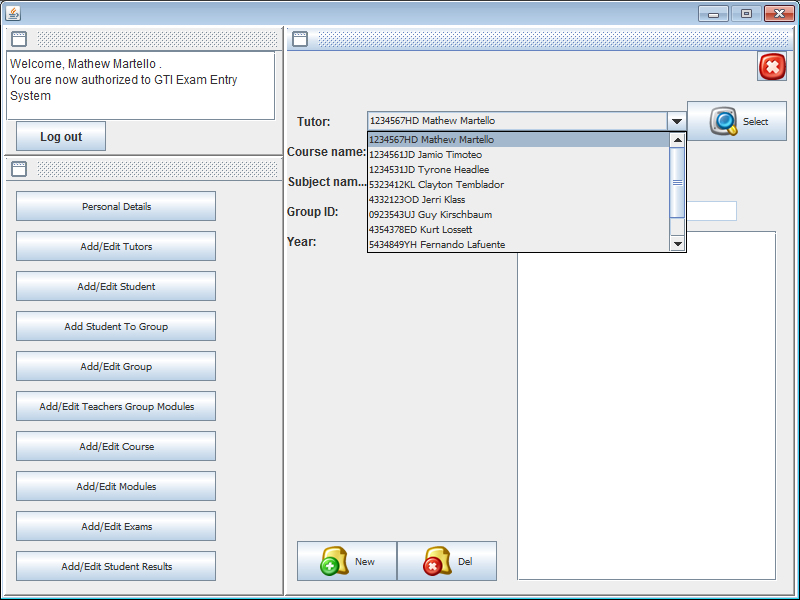


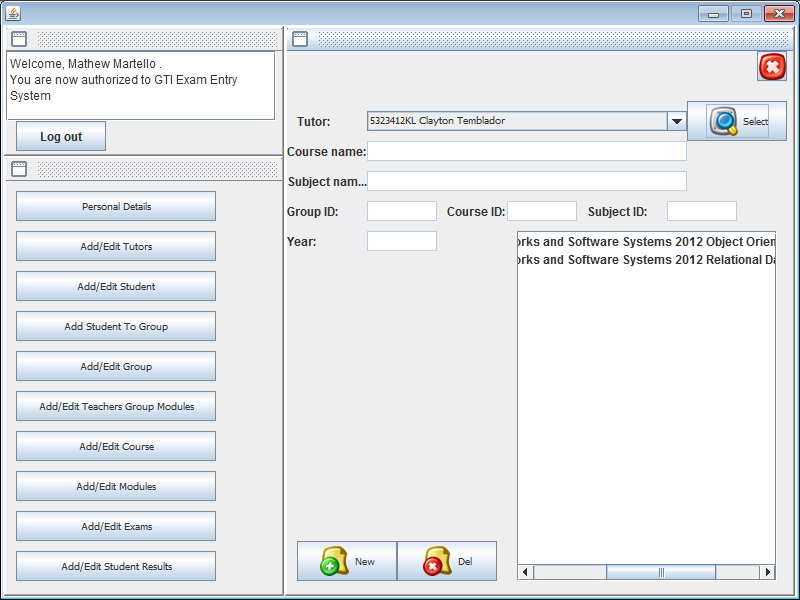
Find new student in recently added class group.

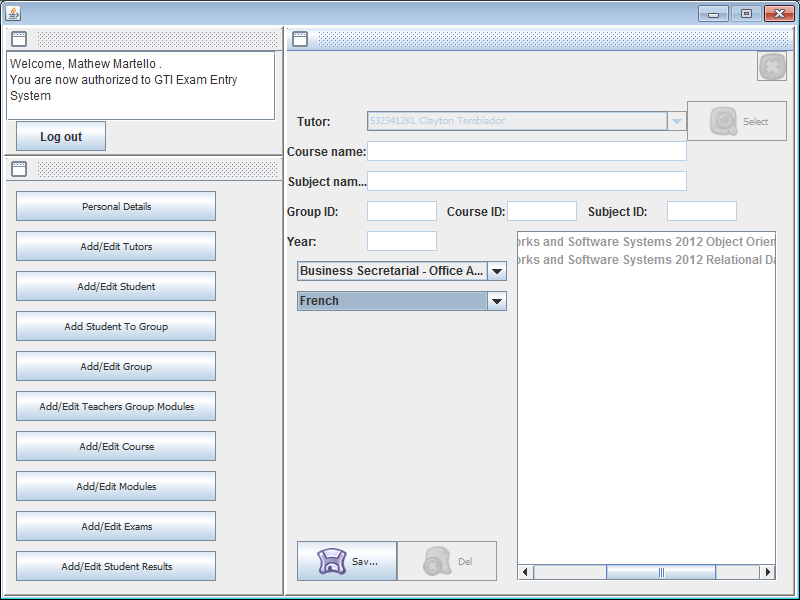


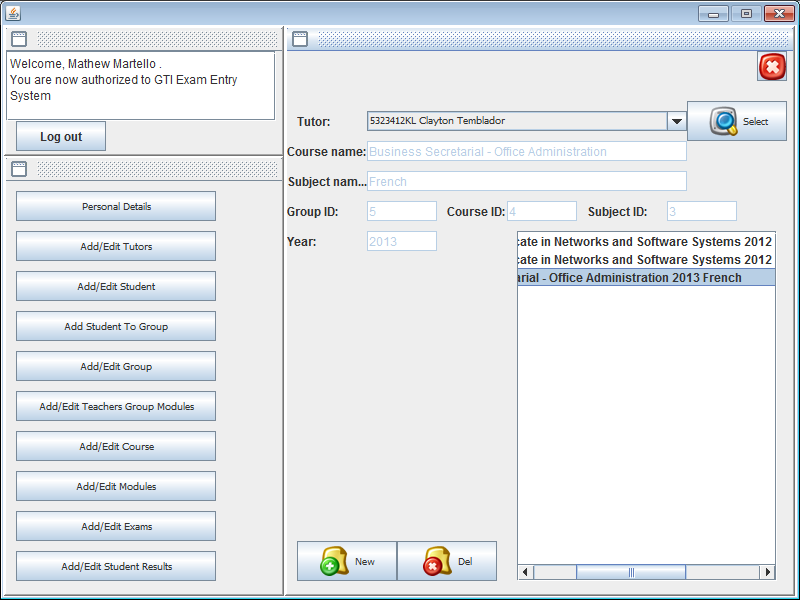
## Test statement 3

In this test statement I will try to add new class group module to selected tutor. Click new button in add/edit teachers group modules form.



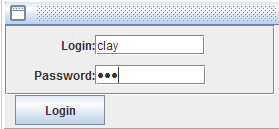


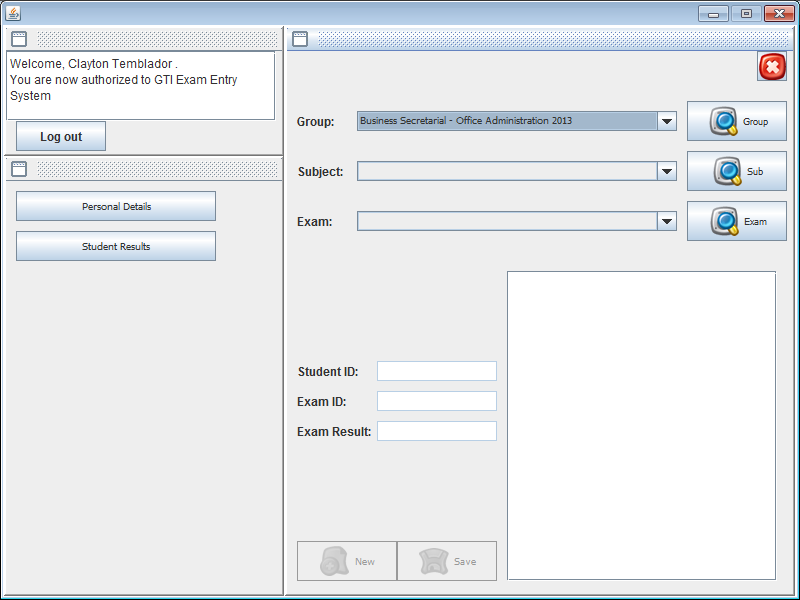


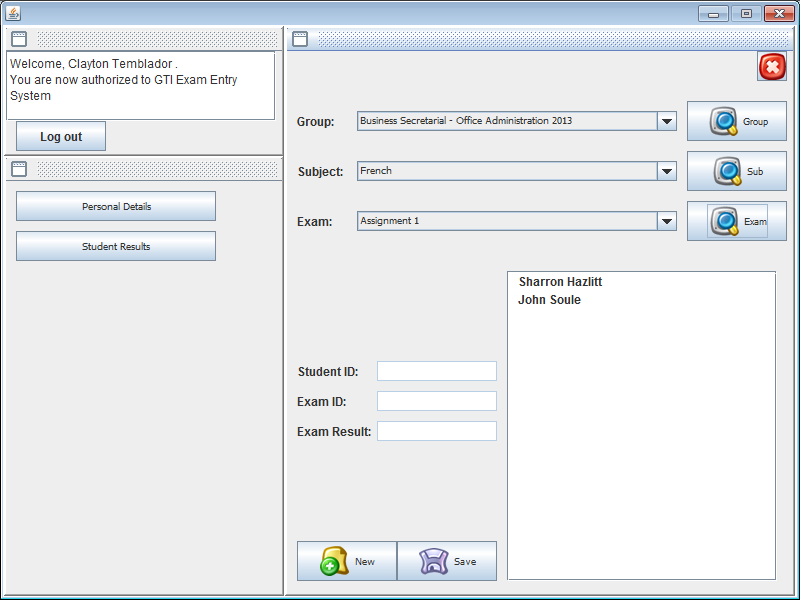


## Test statement 4

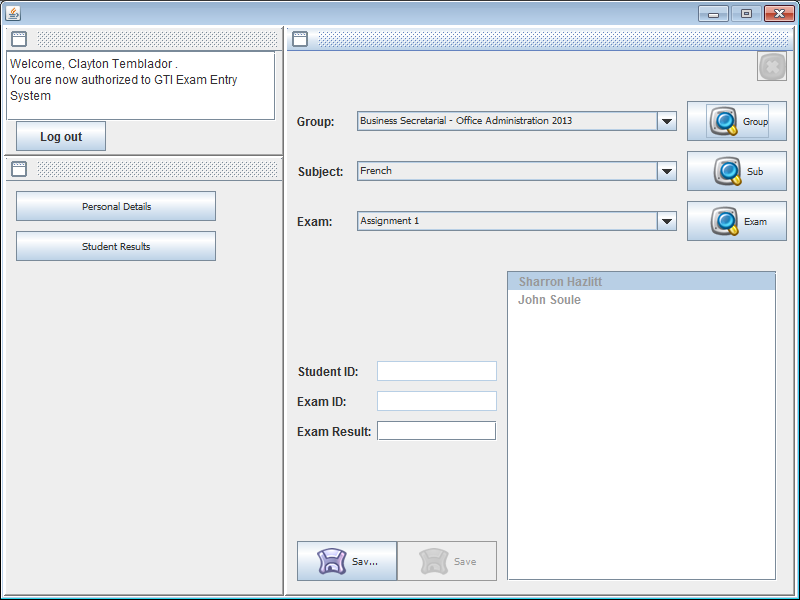
In this statement I will try to login into system as Login clay and password 123. And find the new class group module, which he is teaching.

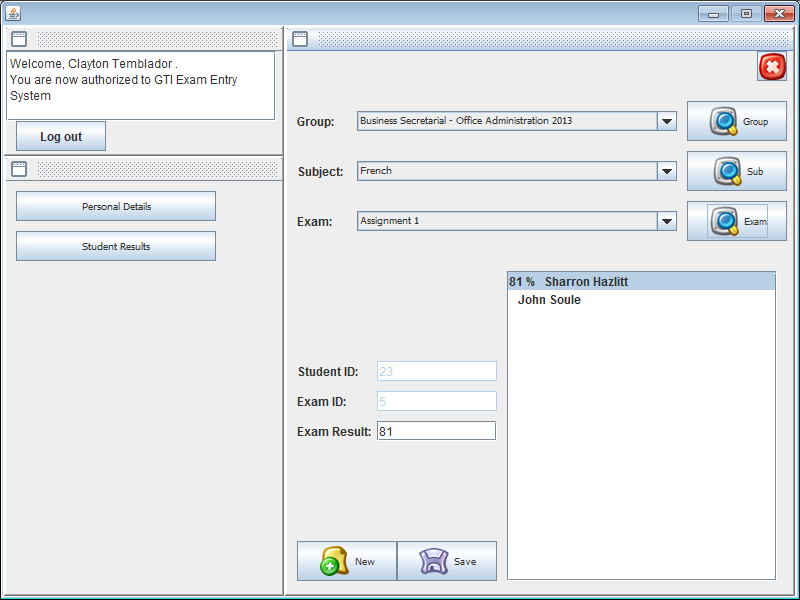






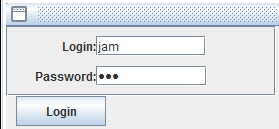
In next few steps I will add a result for one of the students. And to update the list I will click Exam after saving new result.



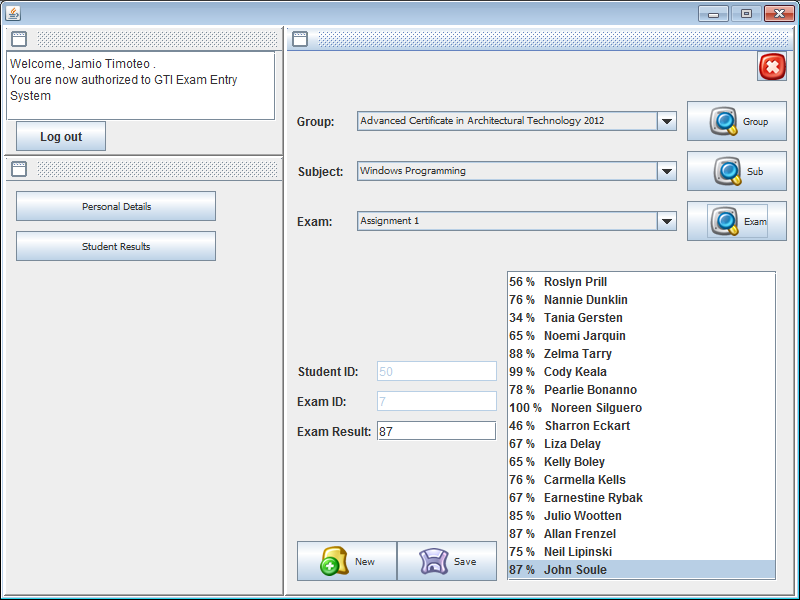


## Test statement 5

I will try to login as jam with password 123.



I can select a group and add new results for each student going one by another, and on the end I can click Exam button to view list students with results.



To double check that everything is working correctly I will login as administrator and view this class group results.

