CMPE 331 TERM PROJECT

WRSCHOLAE

STUDENT INFORMATION SYSTEM

Computer Engineering

CMPE 331 - Fall 2018

Course Teacher: Pınar Hacıbeyoğlu

Faculty of Engineering and Natural Sciences

The Bilgi University

Project Members

- Ekin Silahlioglu (116200078)
- Selin Yesilselve (115200041)
- Simge Erek (115200023)

Project Description

The application that we will design is used in primary school for student assessment which is used by students, school teachers and parents.

Users will have restrictions during the accessing the information such as; teachers will be able to enter grades, enter homework and exam dates also students able to see only their grades which are given by the teacher, parents can only observe his or her kid review and data and not the others'.

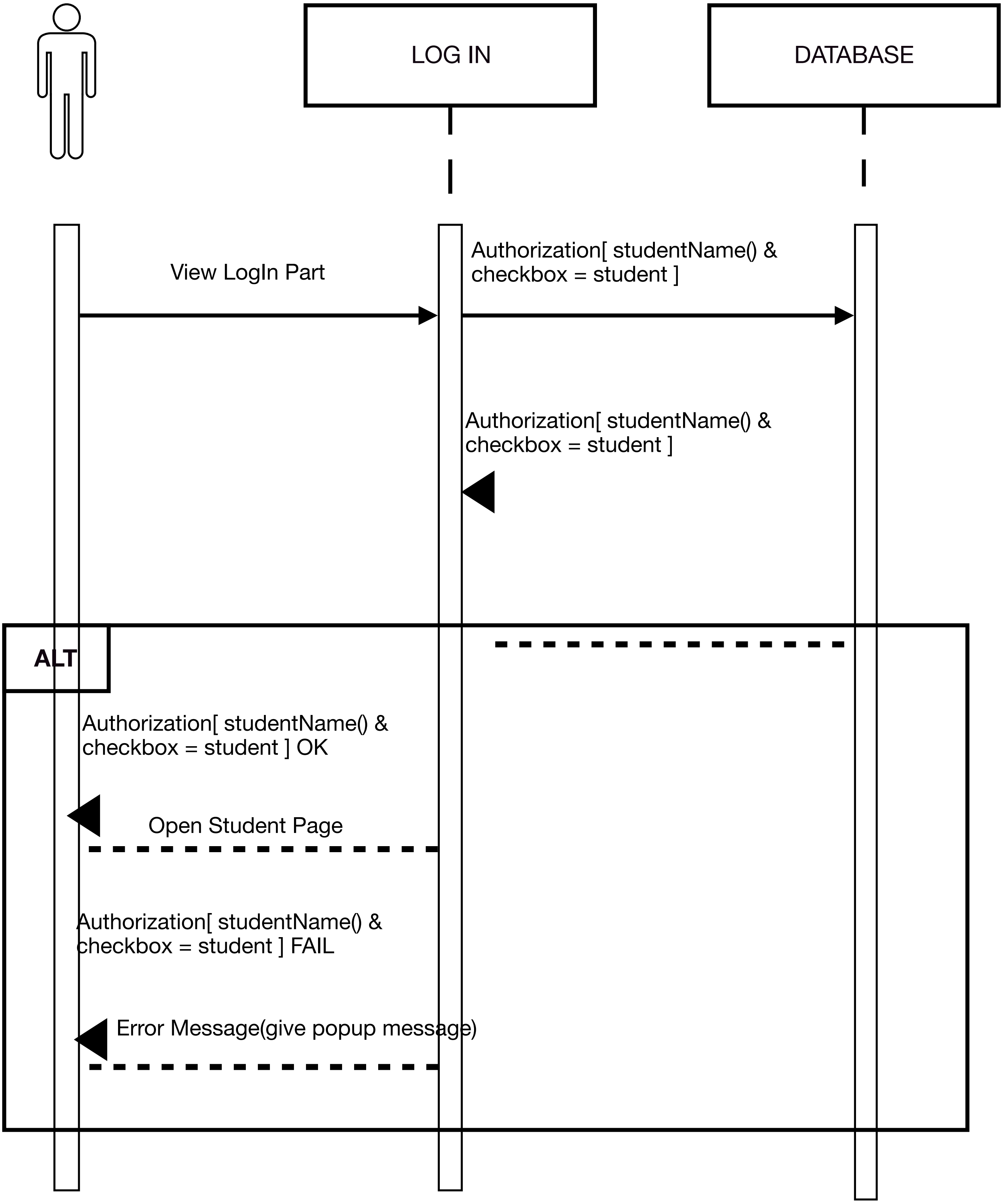
Addition to these, there will exist a page referring the basic academic properties for the kids, some links, pictures for students and families.

Project Purpose

This project's purpose is to make a platform that is suitable for primary school kids to access their course material and result to see their grades also for parents to learn their kids' process such as seeing their kid's grades, homework and finally for teachers to see or change the students' grade and see their change over the time also they can add homework and announcements for students and parents to see.

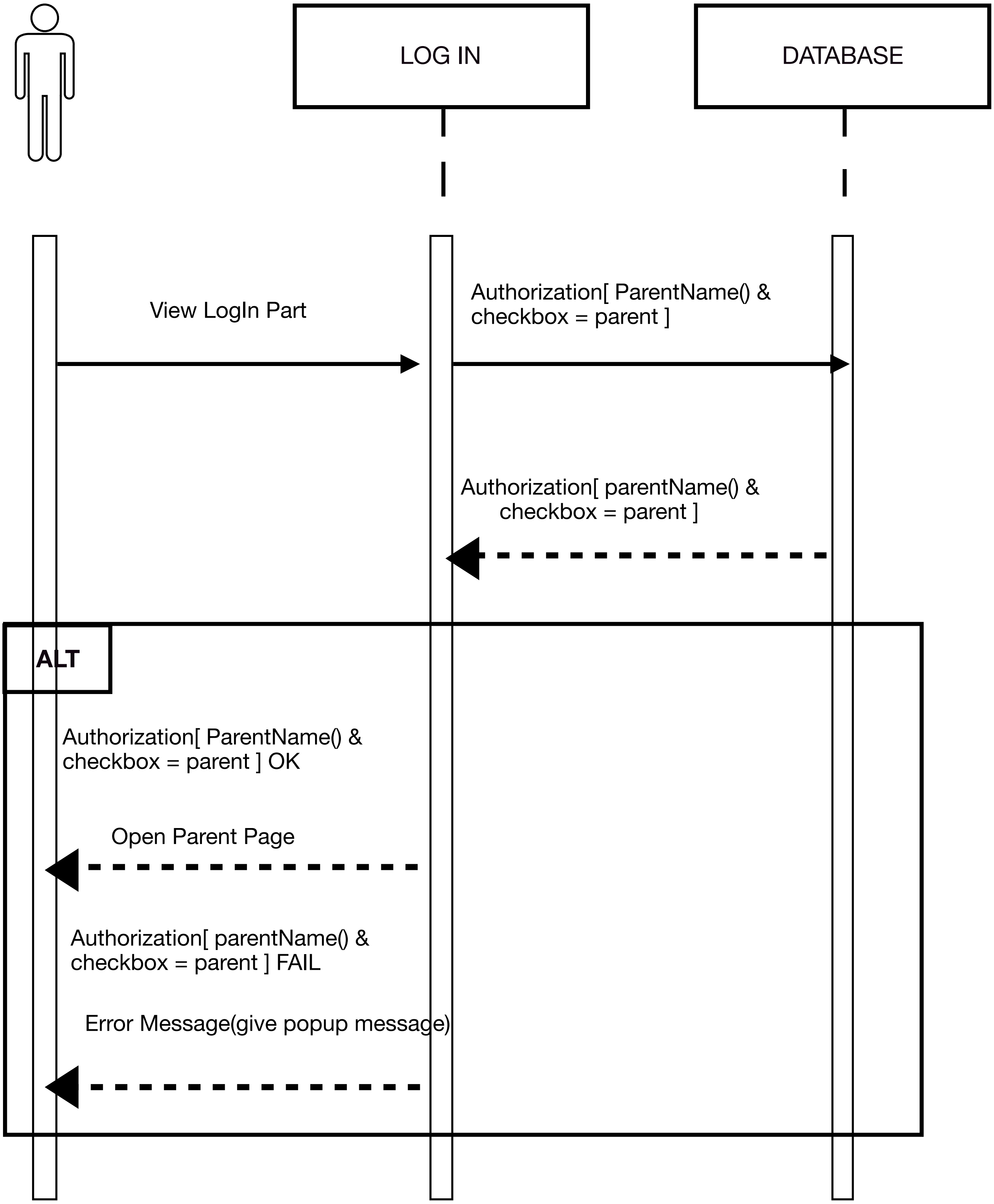
Use Case Diagrams

1)Student :

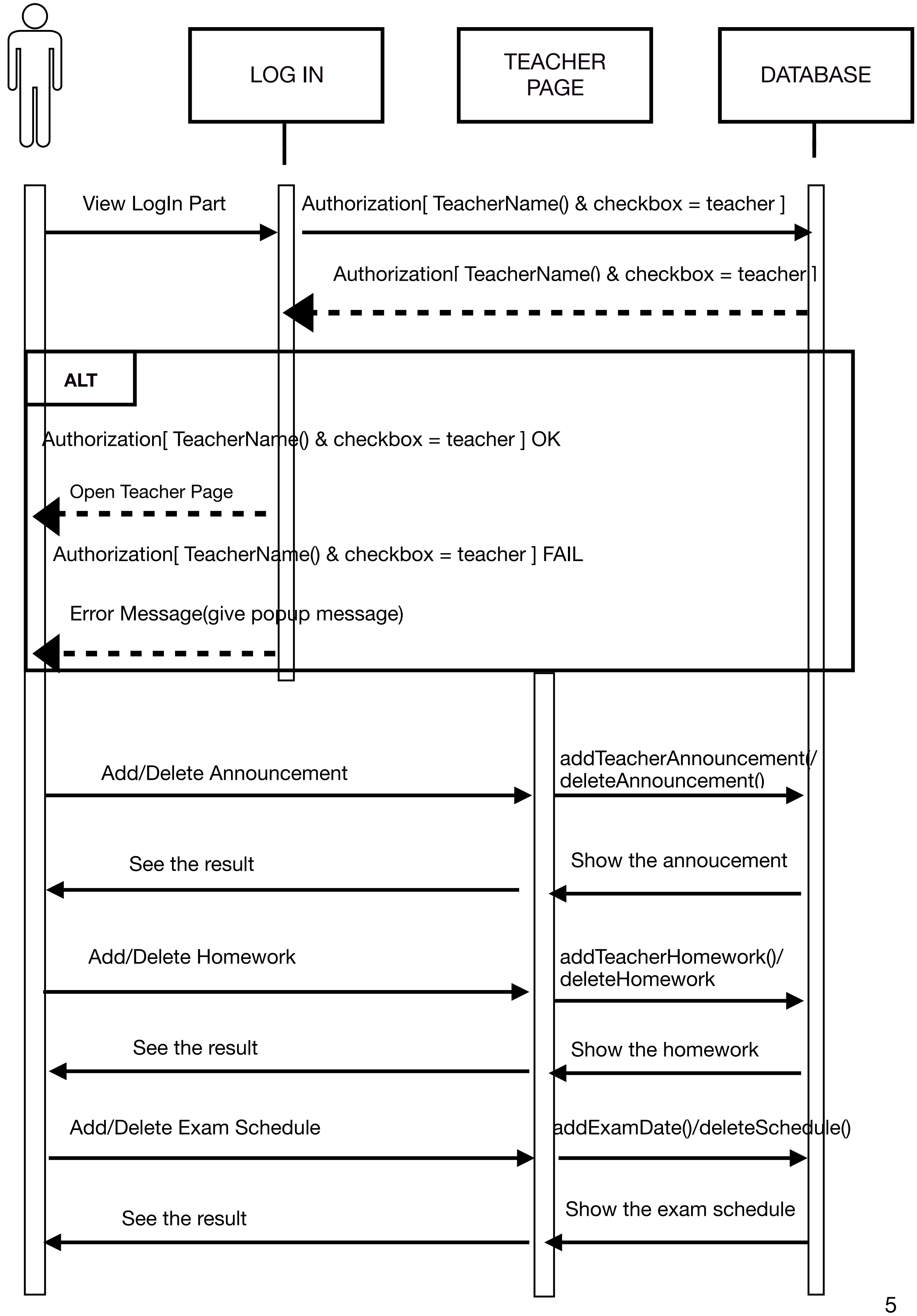


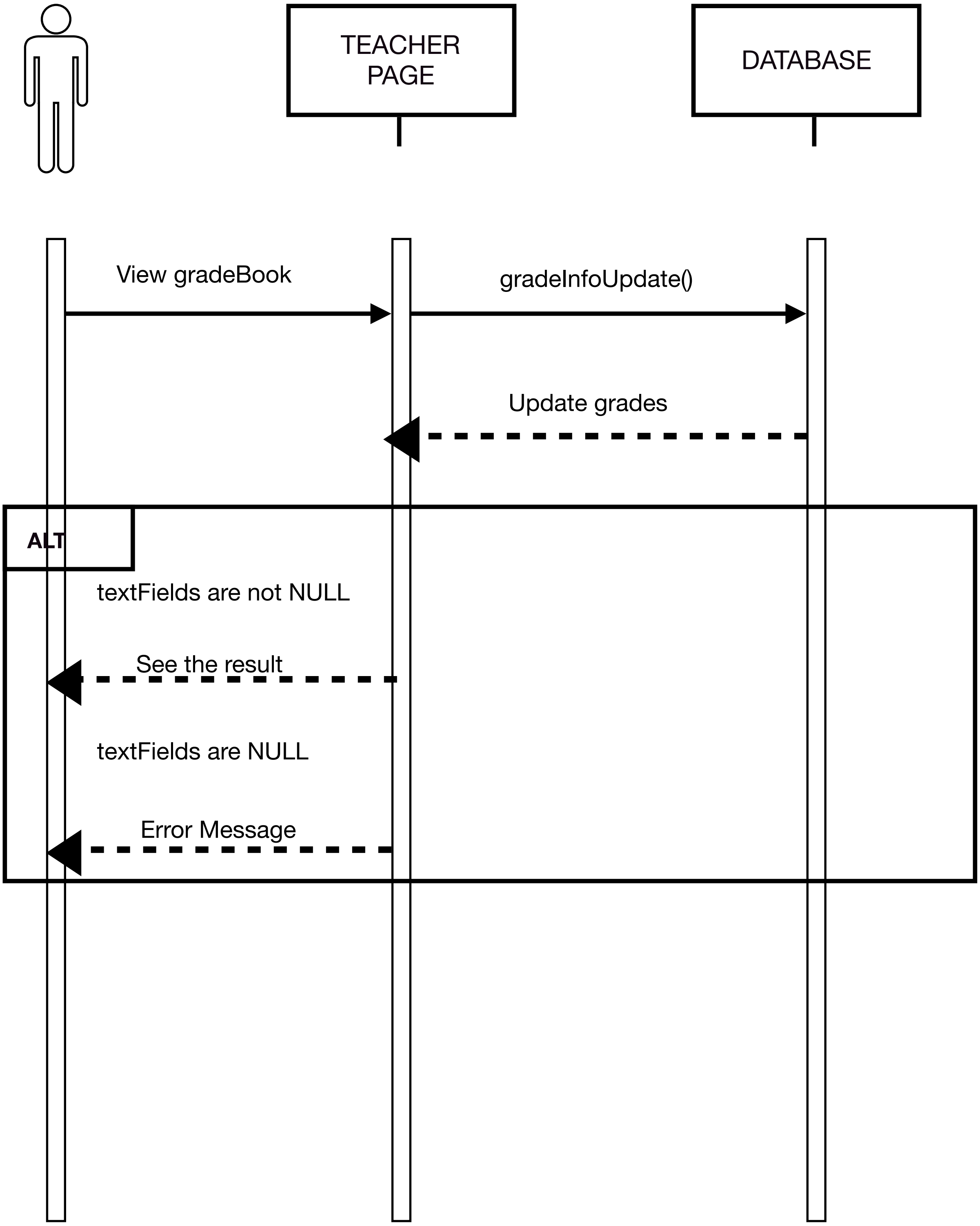
LOG IN
DATABASE
View LogIn Part
Authorization[studentName() & checkbox = student]
Authorization[studentName() & checkbox = student]
ALT
Authorization[studentName() & checkbox = student] OK
Open Student Page
Authorization[studentName() & checkbox = student] FAIL
Error Message(give popup message)

2)<u>Parent</u> :

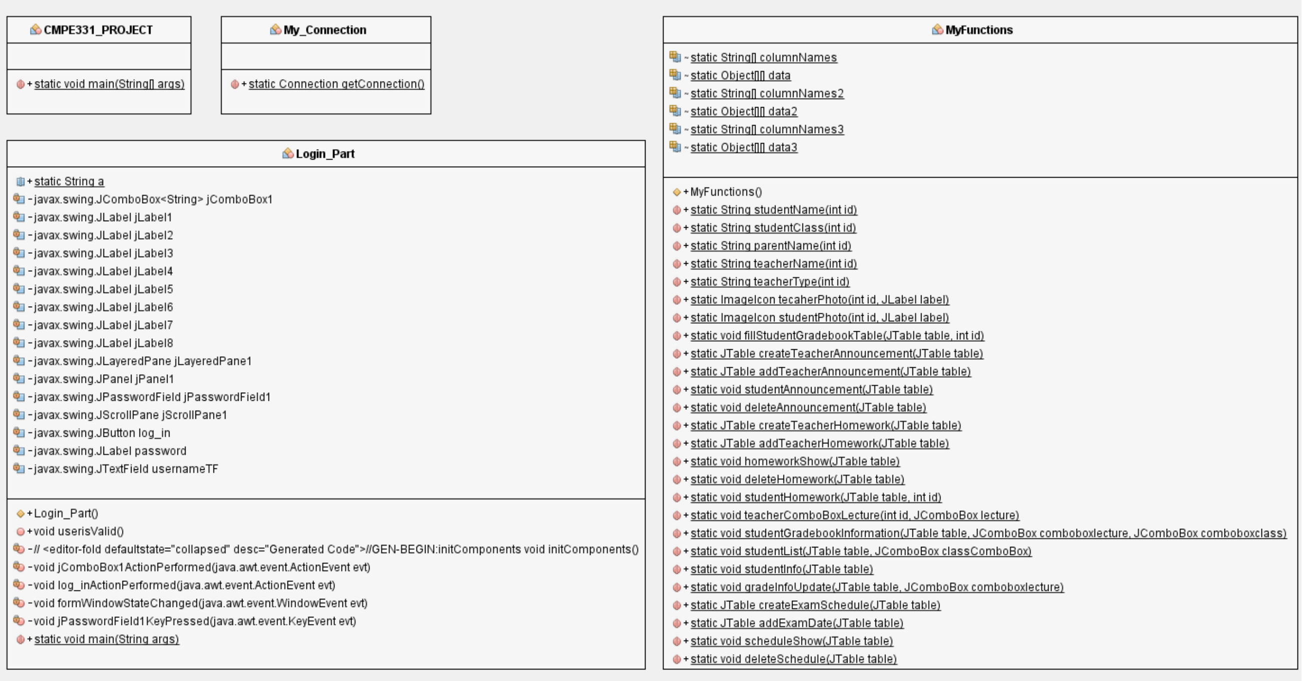

LOG IN
DATABASE
View LogIn Part
Authorization[ParentName() & checkbox = parent]
Authorization[parentName() & checkbox = parent]
ALT
Authorization[ParentName() & checkbox = parent] OK
Open Parent Page
Authorization[parentName() & checkbox = parent] FAIL
Error Message(give popup message)

3)<u>Teacher</u> :


LOG IN
TEACHER PAGE
DATABASE
View LogIn Part
Authorization[TeacherName() & checkbox = teacher]
Authorization[TeacherName() & checkbox = teacher]
ALT
Authorization[TeacherName() & checkbox = teacher] OK
Open Teacher Page
Authorization[TeacherName() & checkbox = teacher] FAIL
Error Message(give popup message)
Add/Delete Announcement
addTeacherAnnouncement()/
deleteAnnouncement()
See the result
Show the annoucement
Add/Delete Homework
addTeacherHomework()/
deleteHomework
See the result
Show the homework
Add/Delete Exam Schedule
addExamDate()/deleteSchedule()
See the result
Show the exam schedule


TEACHER PAGE
DATABASE
View gradeBook
gradeInfoUpdate()
Update grades
ALT
textFields are not NULL
See the result
textFields are NULL
Error Message

[1]UML Diagrams

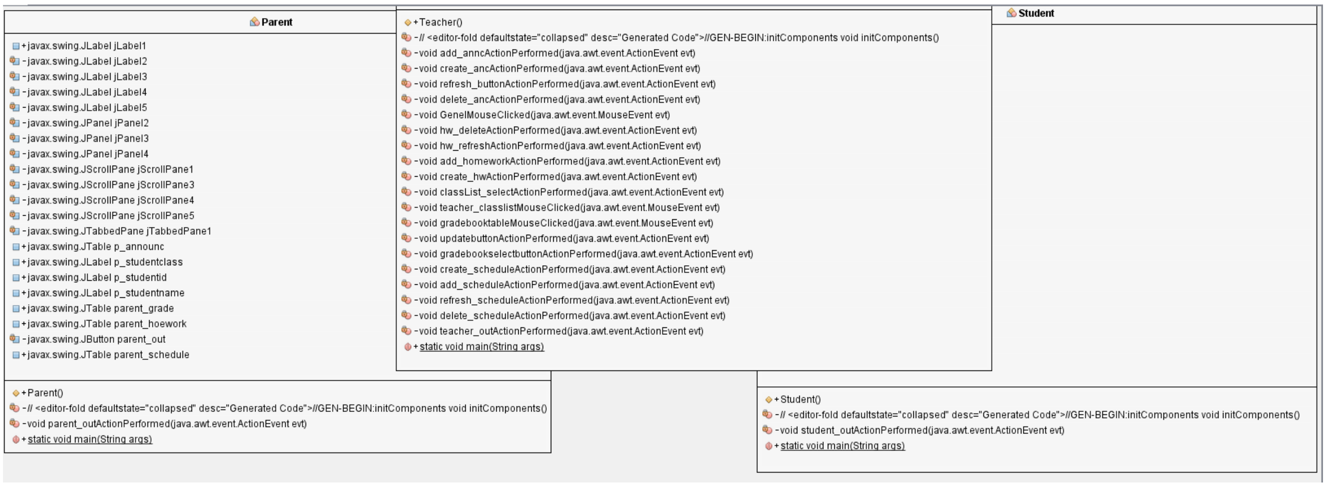


```
CMPE331_PROJECT

+ static void main(String[] args)
```

```
My_Connection

+ static Connection getConnection()
```

```
Login_Part

+ static String a
- javax.swing.JComboBox<String> jComboBox1
- javax.swing.JLabel jLabel1
- javax.swing.JLabel jLabel2
- javax.swing.JLabel jLabel3
- javax.swing.JLabel jLabel4
- javax.swing.JLabel jLabel5
- javax.swing.JLabel jLabel6
- javax.swing.JLabel jLabel7
- javax.swing.JLabel jLabel8
- javax.swing.JLayeredPane jLayeredPane1
- javax.swing.JPanel jPanel1
- javax.swing.JPasswordField jPasswordField1
- javax.swing.JScrollPane jScrollPane1
- javax.swing.JButton log_in
- javax.swing.JLabel password
- javax.swing.JTextField usernameTF

+ Login_Part()
+ void userisValid()
- // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents void initComponents()
- void jComboBox1ActionPerformed(java.awt.event.ActionEvent evt)
- void log_inActionPerformed(java.awt.event.ActionEvent evt)
- void formWindowStateChanged(java.awt.event.WindowEvent evt)
- void jPasswordField1KeyPressed(java.awt.event.KeyEvent evt)
+ static void main(String args)
```

```
MyFunctions

~ static String[] columnNames
~ static Object[][] data
~ static String[] columnNames2
~ static Object[][] data2
~ static String[] columnNames3
~ static Object[][] data3

+ MyFunctions()
+ static String studentName(int id)
+ static String studentClass(int id)
+ static String parentName(int id)
+ static String teacherName(int id)
+ static String teacherType(int id)
+ static ImageIcon tecaherPhoto(int id, JLabel label)
+ static ImageIcon studentPhoto(int id, JLabel label)
+ static void fillStudentGradebookTable(JTable table, int id)
+ static JTable createTeacherAnnouncement(JTable table)
+ static JTable addTeacherAnnouncement(JTable table)
+ static void studentAnnouncement(JTable table)
+ static void deleteAnnouncement(JTable table)
+ static JTable createTeacherHomework(JTable table)
+ static JTable addTeacherHomework(JTable table)
+ static void homeworkShow(JTable table)
+ static void deleteHomework(JTable table)
+ static void studentHomework(JTable table, int id)
+ static void teacherComboBoxLecture(int id, JComboBox lecture)
+ static void studentGradebookInformation(JTable table, JComboBox comboboxlecture, JComboBox comboboxclass)
+ static void studentList(JTable table, JComboBox classComboBox)
+ static void studentInfo(JTable table)
+ static void gradeInfoUpdate(JTable table, JComboBox comboboxlecture)
+ static JTable createExamSchedule(JTable table)
+ static JTable addExamDate(JTable table)
+ static void scheduleShow(JTable table)
+ static void deleteSchedule(JTable table)
```



```
Parent

+ javax.swing.JLabel jLabel1
- javax.swing.JLabel jLabel2
- javax.swing.JLabel jLabel3
- javax.swing.JLabel jLabel4
- javax.swing.JLabel jLabel5
- javax.swing.JPanel jPanel2
- javax.swing.JPanel jPanel3
- javax.swing.JPanel jPanel4
- javax.swing.JScrollPane jScrollPane1
- javax.swing.JScrollPane jScrollPane3
- javax.swing.JScrollPane jScrollPane4
- javax.swing.JScrollPane jScrollPane5
- javax.swing.JTabbedPane jTabbedPane1
+ javax.swing.JTable p_announc
+ javax.swing.JLabel p_studentclass
+ javax.swing.JLabel p_studentid
+ javax.swing.JLabel p_studentname
+ javax.swing.JTable parent_grade
+ javax.swing.JTable parent_hoework
- javax.swing.JButton parent_out
+ javax.swing.JTable parent_schedule

+ Parent()
- // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents void initComponents()
- void parent_outActionPerformed(java.awt.event.ActionEvent evt)
+ static void main(String args)
```

```
+ Teacher()
- // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents void initComponents()
- void add_anncActionPerformed(java.awt.event.ActionEvent evt)
- void create_ancActionPerformed(java.awt.event.ActionEvent evt)
- void refresh_buttonActionPerformed(java.awt.event.ActionEvent evt)
- void delete_ancActionPerformed(java.awt.event.ActionEvent evt)
- void GenelMouseClicked(java.awt.event.MouseEvent evt)
- void hw_deleteActionPerformed(java.awt.event.ActionEvent evt)
- void hw_refreshActionPerformed(java.awt.event.ActionEvent evt)
- void add_homeworkActionPerformed(java.awt.event.ActionEvent evt)
- void create_hwActionPerformed(java.awt.event.ActionEvent evt)
- void classList_selectActionPerformed(java.awt.event.ActionEvent evt)
- void teacher_classlistMouseClicked(java.awt.event.MouseEvent evt)
- void gradebooktableMouseClicked(java.awt.event.MouseEvent evt)
- void updatebuttonActionPerformed(java.awt.event.ActionEvent evt)
- void gradebookselectbuttonActionPerformed(java.awt.event.ActionEvent evt)
- void create_scheduleActionPerformed(java.awt.event.ActionEvent evt)
- void add_scheduleActionPerformed(java.awt.event.ActionEvent evt)
- void refresh_scheduleActionPerformed(java.awt.event.ActionEvent evt)
- void delete_scheduleActionPerformed(java.awt.event.ActionEvent evt)
- void teacher_outActionPerformed(java.awt.event.ActionEvent evt)
+ static void main(String args)
```

```
Student

+ Student()
- // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents void initComponents()
- void student_outActionPerformed(java.awt.event.ActionEvent evt)
+ static void main(String args)
```

[1] There is no relation between classes because we managed with the methods and LogIn.java class

Functions

public static String studentName(int id): It takes student name and last name from student database using "SELECT * FROM student" query according to their id which comes from login part.

public static String studentClass(int id) : It takes student's class from student database using "SELECT * FROM student" query according to their id which comes from login part.

public static String parentName(int id) : It takes parent's child name and last name from parent database using "SELECT * FROM parent" query according to their id which comes from login part.

public static String teacherType(int id) : It takes teacher's major such as class teacher, english teacher,… from teacher database using "SELECT * FROM teacher" query according to their id which comes from login part.

public static ImageIcon teacherPhoto(int id, JLabel label) : It takes teacher's photo from teacher database using "SELECT * FROM teacher" query according to their id which comes from login part and the label which comes from the UI part.

public static String teacherName(int id): It takes teacher's name from teacher database using "SELECT * FROM teacher" query according to their id which comes from login part.

public static ImageIcon studentPhoto(int id , JLabel label) : It takes student's photo from student database using "SELECT * FROM student" query according to their id which comes from login part and the label which comes from the UI part.

public static void fillStudentGradebookTable(JTable table,int id): It takes student's courses from courses own database and fill the table with the student's grade using "SELECT * FROM mathematics WHERE student_id =" + id+" UNION SELECT * *FROM science WHERE student_id =" +id+" UNION SELECT* * FROM literature WHERE student_id =" +id+" UNION SELECT **FROM english WHERE student_id =" +id+" UNION SELECT* * FROM physical_education WHERE student_id =" + id query according to their id which comes from login part.

public static JTable createTeacherAnnouncement(JTable table) : It creates announcement table for the "*addTeacherAnnouncement*" method.

public static JTable addTeacherAnnouncement(JTable table) : It creates announcements after take input from user and put them into the table and also add to the announcement database.

public static void studentAnnouncement(JTable table) : It takes the announcements from announcement database and put it into the table.

public static void deleteAnnouncement(JTable table) : It deletes the announcements from announcement database using "DELETE from announcement WHERE announcement = '" + value + "' AND teacher_name = '" + value2 + “'" query.

public static JTable createTeacherHomework(JTable table) : It creates homework table for the “*addTeacherHomework*” method.

public static JTable addTeacherHomework(JTable table) : It creates homework after take inputs from user and put them into the table and also add to the homework database.

public static void homeworkShow(JTable table) : It takes the homework and put them into a table to show it to the parents, teachers and students.

public static void deleteHomework(JTable table) : It deletes the homework from homework database using "DELETE from homework WHERE homework = '" + value0 + "' AND deadline = '" + value1 + "' AND lecture_name = '" + value2 + "' AND class = '" + value3 + "'"; query.

public static void teacherComboBoxLecture(int id, JComboBox lecture): According to the teacher’s type inside of the combobox the lectures will change.

public static void studentGradebookInformation(JTable table, JComboBox comboboxlecture ,JComboBox comboboxclass) : According to the combobox lecture and combobox class choices, the student list with their grades will be shown in the table.

public static void studentList(JTable table,JComboBox classComboBox) : According to the class type , student list will be shown inside the table.

public static void studentInfo(JTable table) : When the teacher clicked a row from the student list table, a pop up message will be appeared that contains information about that specific student.

public static JTable createExamSchedule(JTable table) : It creates exams table for the "*addExamDate*" method.

public static JTable addExamDate(JTable table) : It creates exams after take input from user and put them into the table and also add to the exam_schedule database.

public static void scheduleShow(JTable table) : It takes the exams and put them into a table to show it..

public static void deleteSchedule(JTable table) : It deletes the exams from exam_schedule database using "DELETE from exam_schedule WHERE lecture_name = '" + value0 +"' AND exam_date = '" + value1 +"'" query.

public static JTable studentSchedule(JTable table) : It takes the exams and put them into a table to show it for students.

public static void gradeInfoUpdate(JTable table, JComboBox comboboxlecture): Teacher can change the students' grade and the method updates the each course database according to the points.

[2]Structure of the Database

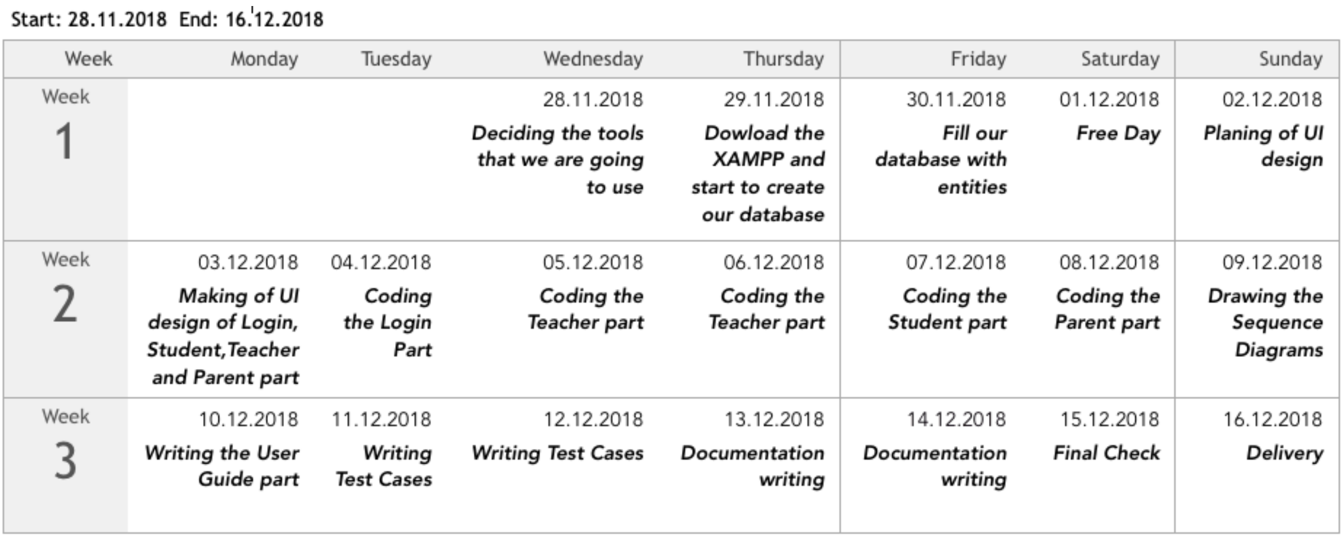
Our database name is “studentinformationsystem” which includes all necessary information that we are going to use for the application.

To begin with our actors, which are student, parent and teacher , that all of them have individual tables. For students’ grade information we have created each courses tables separately to manage course information easily. Addition to these tables, we have a table for announcement that will be shown in every students’ and parents’ page, there is exam schedule and homework tables which will have restricted visual according to classes.

Class teachers can manage mathematics, science and literature courses and reach that courses informations. The rest of the teachers will reach only their own classes’ information which they give courses.

[2] Database Screenshots are shown in the appendix.

Project Timeline



Start: 28.11.2018 End: 16.12.2018

Week	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Week 1			28.11.2018 *Deciding the tools that we are going to use*	29.11.2018 *Dowload the XAMPP and start to create our database*	30.11.2018 *Fill our database with entities*	01.12.2018 *Free Day*	02.12.2018 *Planing of UI design*
Week 2	03.12.2018 *Making of UI design of Login, Student,Teacher and Parent part*	04.12.2018 *Coding the Login Part*	05.12.2018 *Coding the Teacher part*	06.12.2018 *Coding the Teacher part*	07.12.2018 *Coding the Student part*	08.12.2018 *Coding the Parent part*	09.12.2018 *Drawing the Sequence Diagrams*
Week 3	10.12.2018 *Writing the User Guide part*	11.12.2018 *Writing Test Cases*	12.12.2018 *Writing Test Cases*	13.12.2018 *Documentation writing*	14.12.2018 *Documentation writing*	15.12.2018 *Final Check*	16.12.2018 *Delivery*

Structure of System & Software Development Tools

We are using Waterfall Model for our project. Because this is a wide range application and we need to decide every step our project in advance in order to delivery the best quality.

We use XAMPP to connect our PhpMyAdmin database to our NetBeans(IDE).

Group Members' Task

- Ekin Silahlioglu : Design, Coding, Testing & Documentation
- Selin Yesilselve : Design, Coding, Testing & Documentation
- Simge Erek : Design, Coding, Testing & Documentation

User Guide

First thing that you will do is enter your UserID and password after that choose your identity from the combo box.
First, consider that you are a student. You will be directed to your frame that will include name, last name, class , id. There is a SignOut button that will be guide you to the login part. There is tabbed pane which includes; GradeBook,Exam Schedule and Homework. In the GradeBook tab; you are able to see grade of the courses that you are taking.In the Exam Schedule tab; there is a table which shows lecture name and the given lecture's exam date.In the Homework tab; you can see information about homework which are homework, deadline and lecture name of the given homework. On the right of your screen there is an announcement table which includes teacher name that gives the announcement and announcement. We shouldn't be forget that all data except student's personal informations added by teachers to the system.

If you log in as a parent instructions will be exactly the same.

Finally; the last condition is being teacher. There is teachers name , id , and profess″on informations. There is also signs out button that will close the current page and turns you to login part. There is a lot of actions that teacher will able to do.To add announcement first things to do is press create announcement button which is becoming add announcement button after your pressed.There will be pop message that will guide you to add announcement step by step. When you press the refresh button The announcement that in the database will show you table and if you select a row from this table you are able to press delete button, which will delete selected index, otherwise it gives pop up message.
In the tabbed pane first there is ClassList message. You are able to see list of the students which in the class that you choose from combo box.When you select a row there will be pop up information message includes selected student's information. In the second tab; there is a homework table which you will create homework button which will transform add homework button after pressed same as announcement part. Again pop messages will guide you to add homework refresh and delete button functions are the same. In the third tab same as announcement and homework part teacher is able to add date of a exam and exam lecture's name guided by pop up messages. In the last part of the tabs there is Grade Info. After you select class and lecture; when you pressed select button; in the down table graded information of students which takes that lecture from chosen class informations will listed. To update a grade of the selected student you need to change the text field and then press the update button . If you don't give any information to the text fields you will get an error.

Test Cases

TEST CASE	EXPECTED OUTCOME	OUTCOME	EXPECTED RESULT
All mandatory fields should be validated and indicated by an asterisk (*) symbol.	It will be shown in UI.	When you enter your username or password, it appears.	SUCCESS
Validation error messages should be displayed properly.	If the user enter wrong username or password, a popup message will be appeared.	When the user enter wrong username or password, a popup message appears.	SUCCESS
Validation error messages should be displayed properly.	If the user does not exist in any database , a popup message will be appeared.	When the user does not exist, a popup message appears.	SUCCESS
Validation error messages should be displayed properly.	If the user enters the username and does not enter the password, a popup message will be appeared.	When the user enters the username and does not enter the password a popup message appears.	SUCCESS
Validation error messages should be displayed properly.	If the user enters the password and does not enter the username, a popup message will be appeared.	When the user enters the password and does not enter the username, a popup message appears.	SUCCESS
ID Validation and Page Opening of Student Page	The opening will give the information about that person according to the id, the user enters in the logIn part	The opening page of student gives the information about that user such as name,lastname,id…	SUCCESS
ID Validation and Page Opening of Parent Page	The opening will give the information about that person according to the id, the user enters in the logIn part	The opening page of parent gives the information about that user such as name,lastname,id…	SUCCESS

TEST CASE	EXPECTED OUTCOME	OUTCOME	EXPECTED RESULT
ID Validation and Page Opening of Teacher Page	The opening will give the information about that person according to the id, the user enters in the logIn part	The opening page of teacher gives the information about that user such as name,lastname,id…	SUCCESS
GradeBook Visual for Student	If student enters his/her own page, He/she can only see his/her own grades	When student enters his/her own page, He/she sees his/her own grades	SUCCESS
GradeBook Visual for Parent	If parent enters the page he/she can only see his/her own child's grades.	When parent enters the page he/she can only see his/her own child's grades.	SUCCESS
Sign Out	It will close the application	It closes the application	SUCCESS
GradeBook Visual for Teacher	If teacher chooses class and course type from comboboxes, a student list will appear according to the choice.	When teacher chooses class and course type from comboboxes, a student list appears according to the choice.	SUCCESS
Creating Announcement	Each teacher can create annoucement according to their subject.	Each teacher creates annoucement according to their subject.	SUCCESS
Update Grades	If teacher updates one of the courses' grade in student list, it will update the database.	Teacher needs to update each courses grade at that moment , otherwise it gives an error message.	FAIL
Delete Information from Table	If teacher wants to delete homework, announcement or exam schedule, he/she will need to choose a row to delete. If teacher does not select the row before clicked the button, it will give pop up message.	When teacher select the row it deletes after teacher clicks the delete button. And teacher clicks button before he/she select the row, it gives pop up message	SUCCESS
Creating Homework	Each teacher can create homework according to their subject.	Each teacher creates homework according to their subject.	SUCCESS

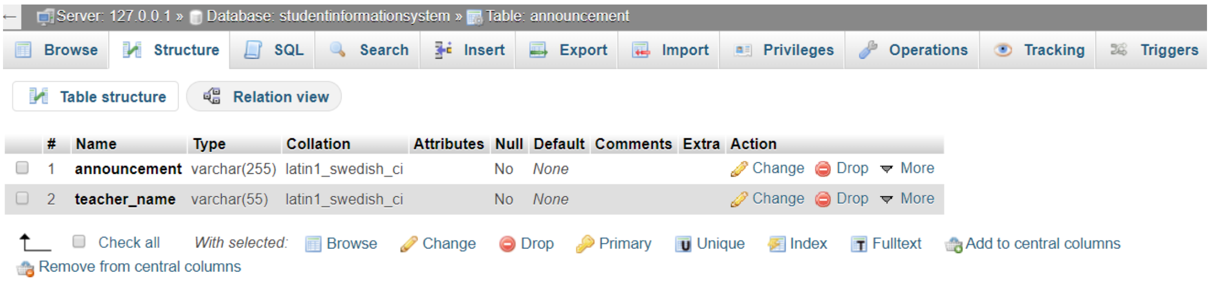
TEST CASE	EXPECTED OUTCOME	OUTCOME	EXPECTED RESULT
Creating Exam Schedule	Each teacher can create exam schedule according to their subject.	Each teacher creates exam schedule according to their subject.	SUCCESS
Class List Visual	Each teacher can see only his/her own classes' class list that he/she give teaching.	Each teacher sees only his/her own classes' class list that he/she give teaching.	SUCCESS
Class List - Pop Up Message	If teacher clicks the selected row from the student list it will give pop up message that includes student information.	When teacher clicks the selected row from the student list it gives pop up message that includes student information.	SUCCESS

Future Works

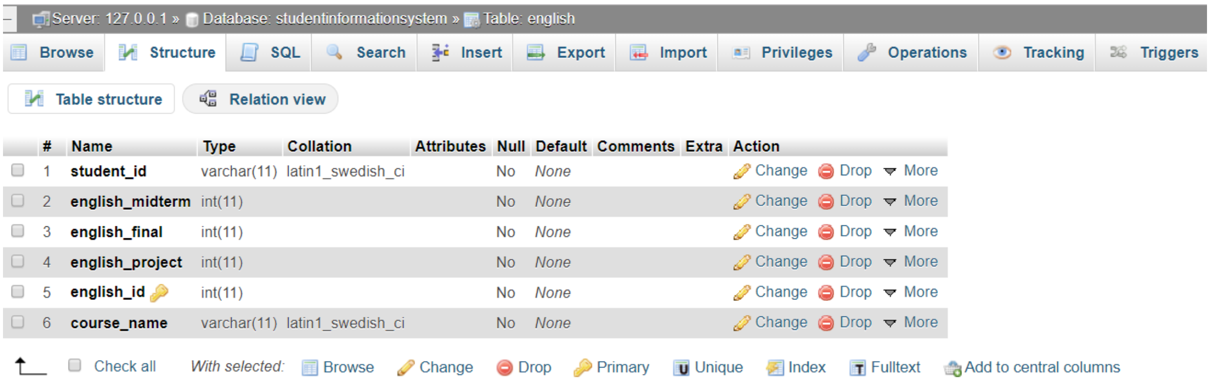
In future we can make improvements for our application. These are:

- Increase the number of teachers and and principal, all working stuff in the school.
- We can upgrade our application to include not only primary school students but also middle school and high school students.
- We can turn our computer based application into a web based application or mobile phone application.
- We can add attendance to our system.
- Teacher can add comments on students' grades individually that only parents can see.

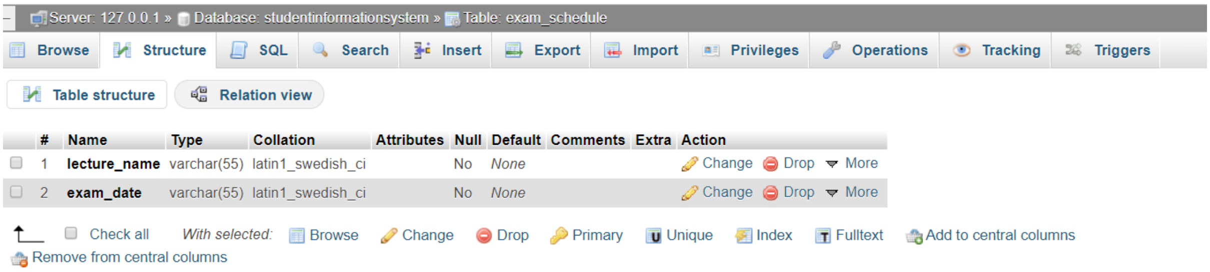
Appendix



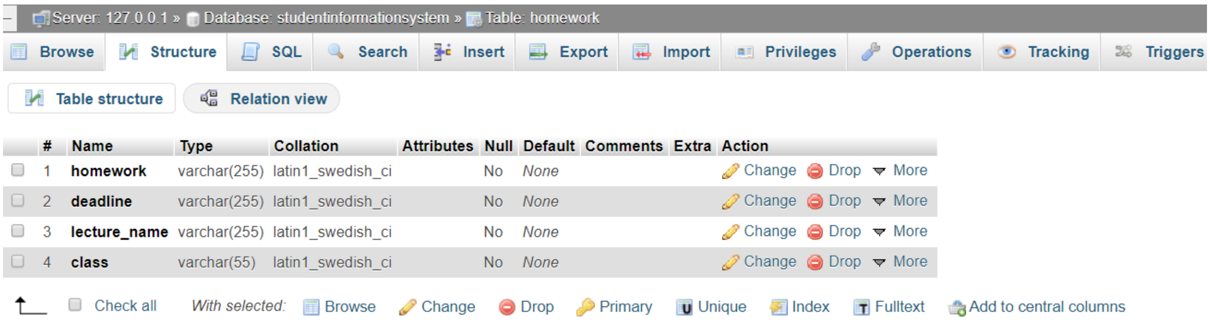
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	**announcement**	varchar(255)	latin1_swedish_ci		No	*None*			Change Drop More
□	2	**teacher_name**	varchar(55)	latin1_swedish_ci		No	*None*			Change Drop More



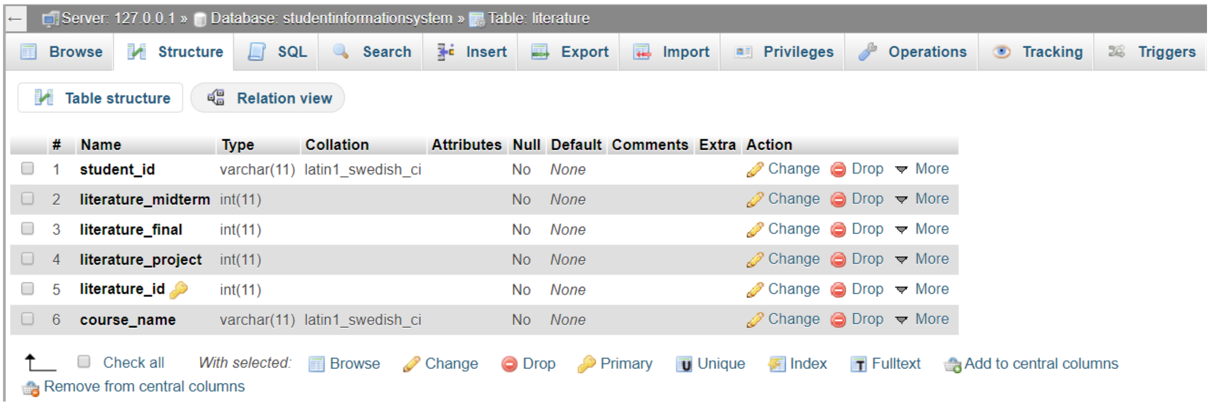
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	**student_id**	varchar(11)	latin1_swedish_ci		No	*None*			Change Drop More
□	2	**english_midterm**	int(11)			No	*None*			Change Drop More
□	3	**english_final**	int(11)			No	*None*			Change Drop More
□	4	**english_project**	int(11)			No	*None*			Change Drop More
□	5	**english_id**	int(11)			No	*None*			Change Drop More
□	6	**course_name**	varchar(11)	latin1_swedish_ci		No	*None*			Change Drop More



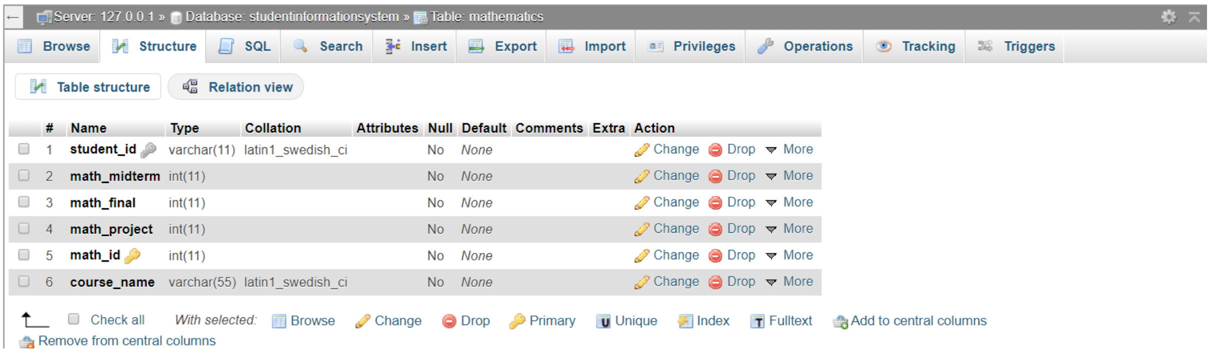
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	**lecture_name**	varchar(55)	latin1_swedish_ci		No	*None*			Change Drop More
□	2	**exam_date**	varchar(55)	latin1_swedish_ci		No	*None*			Change Drop More



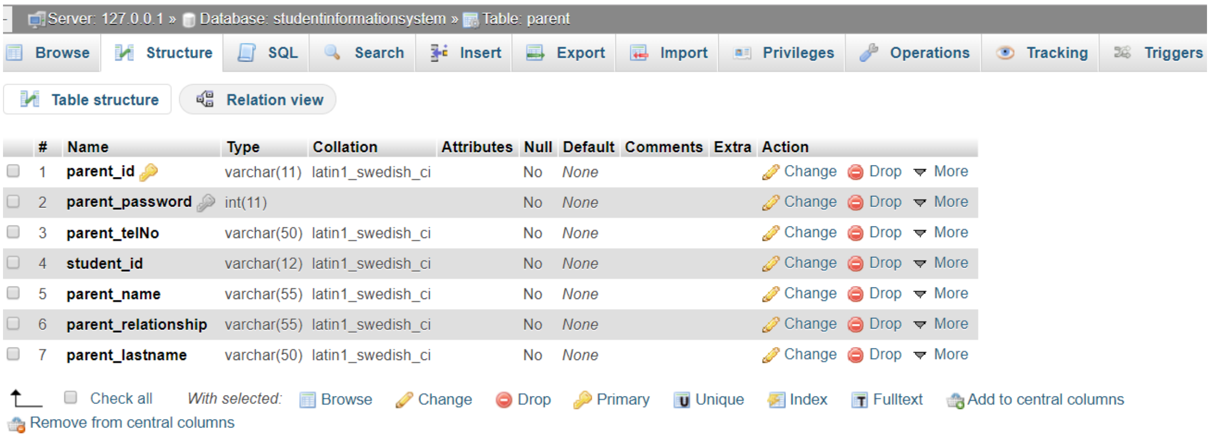
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	**homework**	varchar(255)	latin1_swedish_ci		No	*None*			Change Drop More
□	2	**deadline**	varchar(255)	latin1_swedish_ci		No	*None*			Change Drop More
□	3	**lecture_name**	varchar(255)	latin1_swedish_ci		No	*None*			Change Drop More
□	4	**class**	varchar(55)	latin1_swedish_ci		No	*None*			Change Drop More



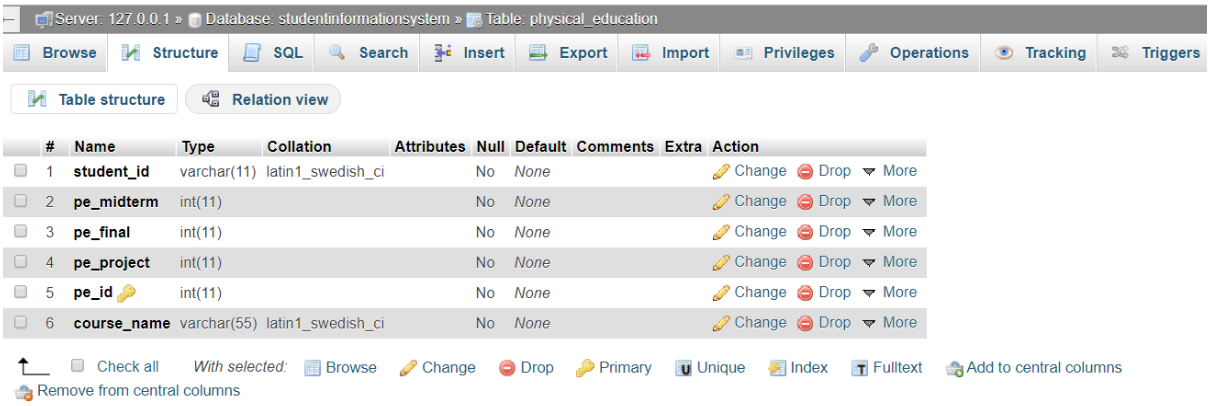
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	student_id	varchar(11)	latin1_swedish_ci		No	*None*			Change Drop More
2	literature_midterm	int(11)			No	*None*			Change Drop More
3	literature_final	int(11)			No	*None*			Change Drop More
4	literature_project	int(11)			No	*None*			Change Drop More
5	literature_id	int(11)			No	*None*			Change Drop More
6	course_name	varchar(11)	latin1_swedish_ci		No	*None*			Change Drop More



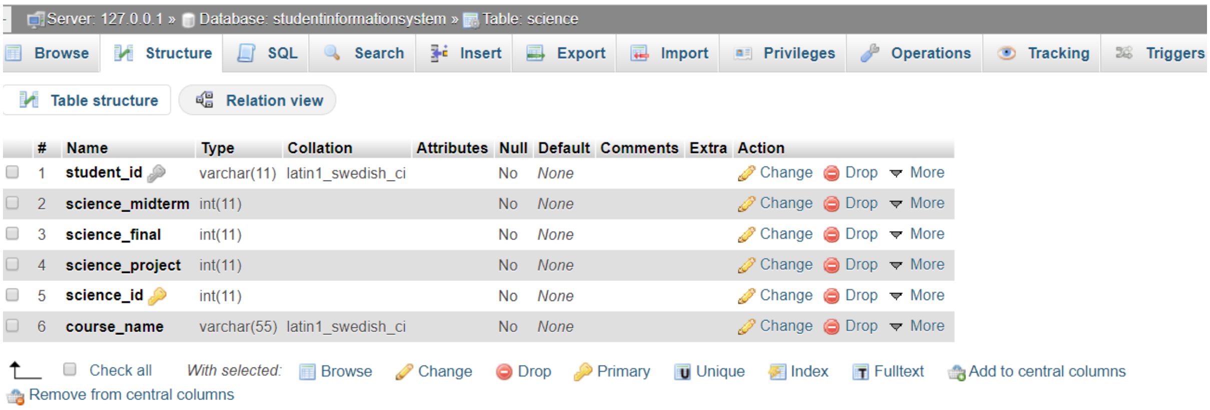
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	student_id	varchar(11)	latin1_swedish_ci		No	*None*			Change Drop More
2	math_midterm	int(11)			No	*None*			Change Drop More
3	math_final	int(11)			No	*None*			Change Drop More
4	math_project	int(11)			No	*None*			Change Drop More
5	math_id	int(11)			No	*None*			Change Drop More
6	course_name	varchar(55)	latin1_swedish_ci		No	*None*			Change Drop More



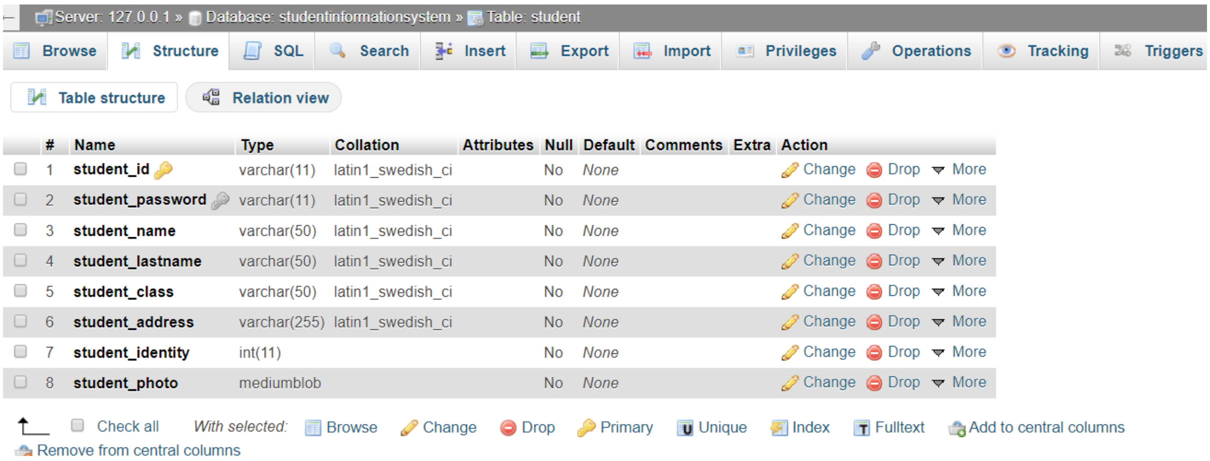
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	parent_id	varchar(11)	latin1_swedish_ci		No	*None*			Change Drop More
2	parent_password	int(11)			No	*None*			Change Drop More
3	parent_telNo	varchar(50)	latin1_swedish_ci		No	*None*			Change Drop More
4	student_id	varchar(12)	latin1_swedish_ci		No	*None*			Change Drop More
5	parent_name	varchar(55)	latin1_swedish_ci		No	*None*			Change Drop More
6	parent_relationship	varchar(55)	latin1_swedish_ci		No	*None*			Change Drop More
7	parent_lastname	varchar(50)	latin1_swedish_ci		No	*None*			Change Drop More



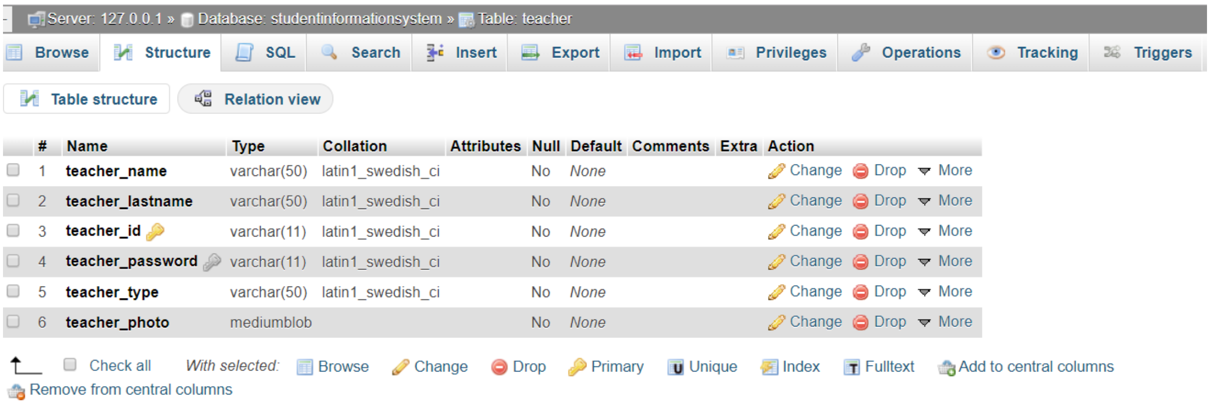
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	student_id	varchar(11)	latin1_swedish_ci		No	*None*			Change Drop More
2	pe_midterm	int(11)			No	*None*			Change Drop More
3	pe_final	int(11)			No	*None*			Change Drop More
4	pe_project	int(11)			No	*None*			Change Drop More
5	pe_id	int(11)			No	*None*			Change Drop More
6	course_name	varchar(55)	latin1_swedish_ci		No	*None*			Change Drop More


Server: 127.0.0.1 » Database: studentinformationsystem » Table: science

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	**student_id**	varchar(11)	latin1_swedish_ci		No	*None*			Change Drop More
☐	2	**science_midterm**	int(11)			No	*None*			Change Drop More
☐	3	**science_final**	int(11)			No	*None*			Change Drop More
☐	4	**science_project**	int(11)			No	*None*			Change Drop More
☐	5	**science_id**	int(11)			No	*None*			Change Drop More
☐	6	**course_name**	varchar(55)	latin1_swedish_ci		No	*None*			Change Drop More


Server: 127.0.0.1 » Database: studentinformationsystem » Table: student

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	**student_id**	varchar(11)	latin1_swedish_ci		No	*None*			Change Drop More
☐	2	**student_password**	varchar(11)	latin1_swedish_ci		No	*None*			Change Drop More
☐	3	**student_name**	varchar(50)	latin1_swedish_ci		No	*None*			Change Drop More
☐	4	**student_lastname**	varchar(50)	latin1_swedish_ci		No	*None*			Change Drop More
☐	5	**student_class**	varchar(50)	latin1_swedish_ci		No	*None*			Change Drop More
☐	6	**student_address**	varchar(255)	latin1_swedish_ci		No	*None*			Change Drop More
☐	7	**student_identity**	int(11)			No	*None*			Change Drop More
☐	8	**student_photo**	mediumblob			No	*None*			Change Drop More


Server: 127.0.0.1 » Database: studentinformationsystem » Table: teacher

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	**teacher_name**	varchar(50)	latin1_swedish_ci		No	*None*			Change Drop More
☐	2	**teacher_lastname**	varchar(50)	latin1_swedish_ci		No	*None*			Change Drop More
☐	3	**teacher_id**	varchar(11)	latin1_swedish_ci		No	*None*			Change Drop More
☐	4	**teacher_password**	varchar(11)	latin1_swedish_ci		No	*None*			Change Drop More
☐	5	**teacher_type**	varchar(50)	latin1_swedish_ci		No	*None*			Change Drop More
☐	6	**teacher_photo**	mediumblob			No	*None*			Change Drop More