

Free Gradekeeper

Ease and Functionality at Everyone's favorite
price.

Patrick Sibley
Carl Scogland
Jerod Brackin

CSC 4330-1
Fall '08

Our **Project**:

Our project is a program to help maintain grades for a school. It will let teachers teaching one or more classes assign and maintain grades for students in each of their classes or all of their subjects while averaging those grades automatically. It will also let students view their grades for each of their classes to track their progress.

The **Problem** we are addressing:

Maintaining grades for a class using paper and pencil is tedious and time consuming. It takes up too much space to try to keep up with piles of graded papers. Many other, mainstream systems made to keep up with such information are expensive or complicated. Schools that are not well off financially may not be able to afford such programs. Regardless, grade keeping is an essential aspect of schooling for both teachers and students.

The **Impact** our project will have:

Our project will give schools a free way to maintain their grades. If the schools are starting out or have limited funds, they won't be able to spend a great deal of money for a grading program for all their teachers. Instead of making the teachers use their valuable time to average and maintain students' grades by hand, the program will do this for teachers automatically, thus giving them more time to focus on their children's education and build a better future for them.

Major Goals of the project:

- To provide flexibility so that users can do what they need to do whenever they need to do it, however they want to do it, without going through too many tedious menus.
- To provide a logical, user friendly interface with no confusing menus so that any teacher or student can use it.
- To provide a time saving averaging program for teachers
- ~~— To let teachers of all grades from elementary to high school easily modify the interface to fit their needs.~~ Due to time constraints and setbacks, this goal had to be abandoned. The focus changed to high school settings only. The reasoning behind this was that a university would likely have enough funds to buy a commercial product while students in elementary settings might not know how to use computers as quite as proficiently as others

Users

The Free Gradekeeper has 3 groups of users:

Teachers: users who teach one or more classes at a school and assign grades based on students' performance,

Students: users who are enrolled in classes in some school setting and take tests to evaluate their knowledge.

Administrator: This one figure maintains school wide settings that neither a teacher nor a student should perform

The following lists the activities they will be able to perform.

Teachers' Activities

- Optionally enter descriptions and point values for all grades they plan to have throughout the semester (to enable students to plan what grades they need to make to get a certain grade)
- Input a set of grades for a test - one grade to each student in the class
- View and edit a specific student's grades
- See the average for a specific student's grades
- See the class's average for the total grade or for a test
- Set whether students can see class averages

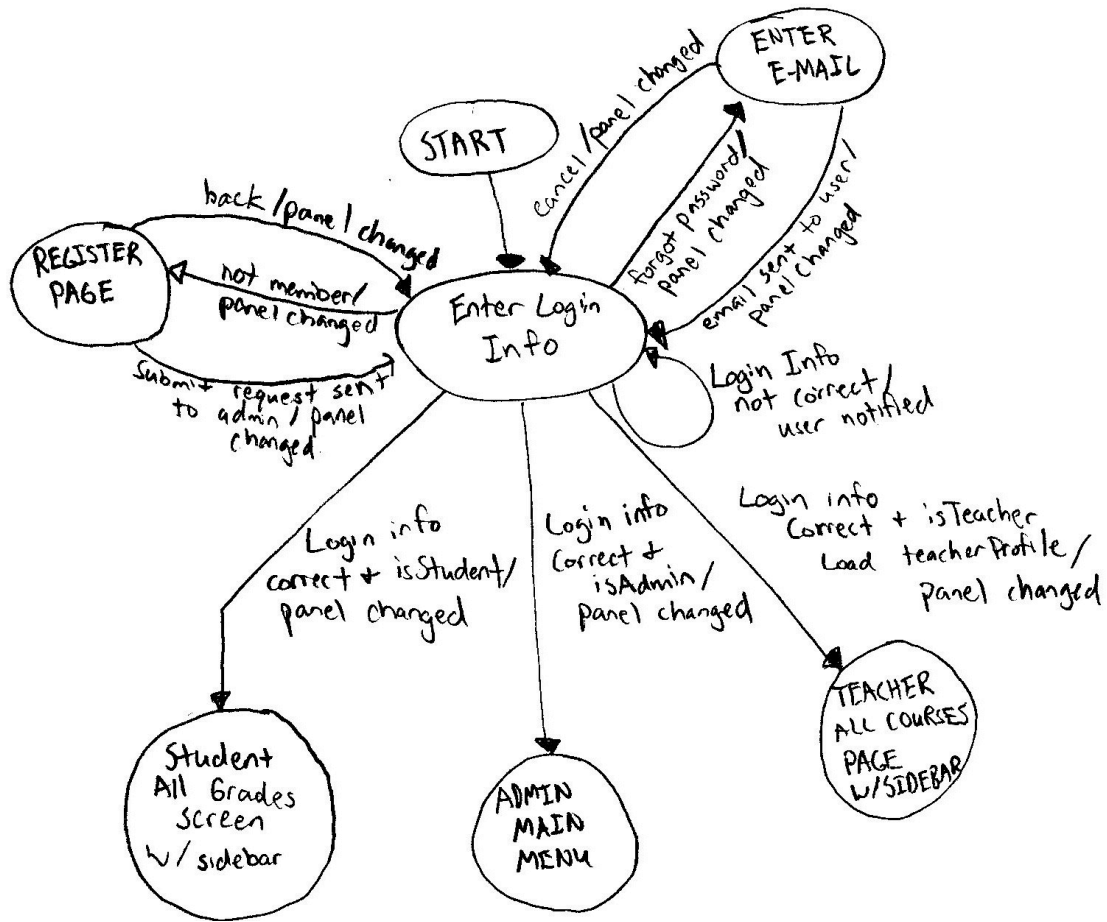
Students' Activities

- See all current grades at the same time
- Change password
- For each class
 - See who the teacher is
 - See past grades on past tests
 - For each of those, see the class average (if enabled)
 - See the total grade they have for the class
 - Input guesses for future grades to see what their average would be

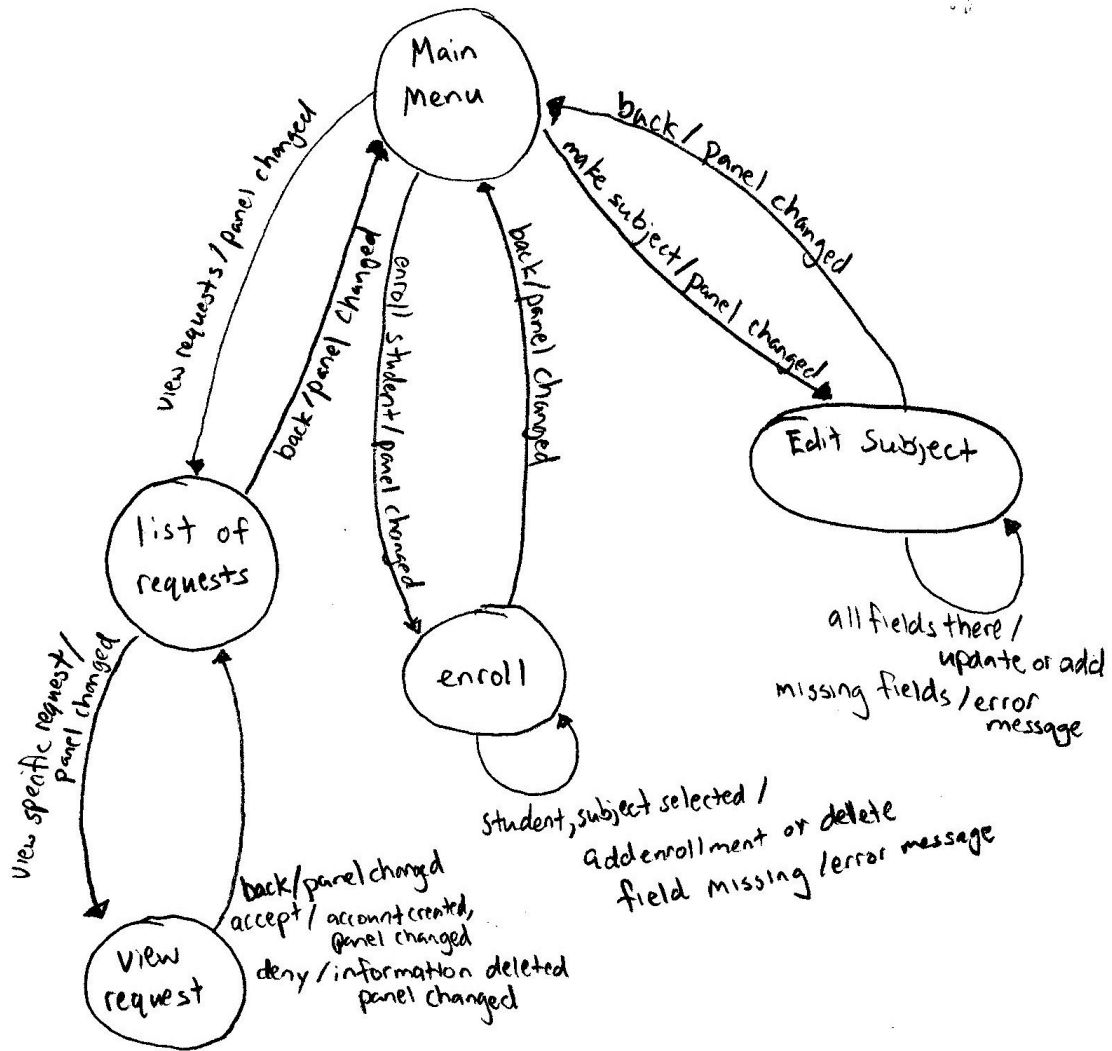
Administrator Activities

- Edit, accept, or deny user accounts from being created
- Enroll students in classes
- Edit, add, or delete courses in the school

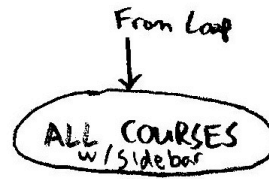
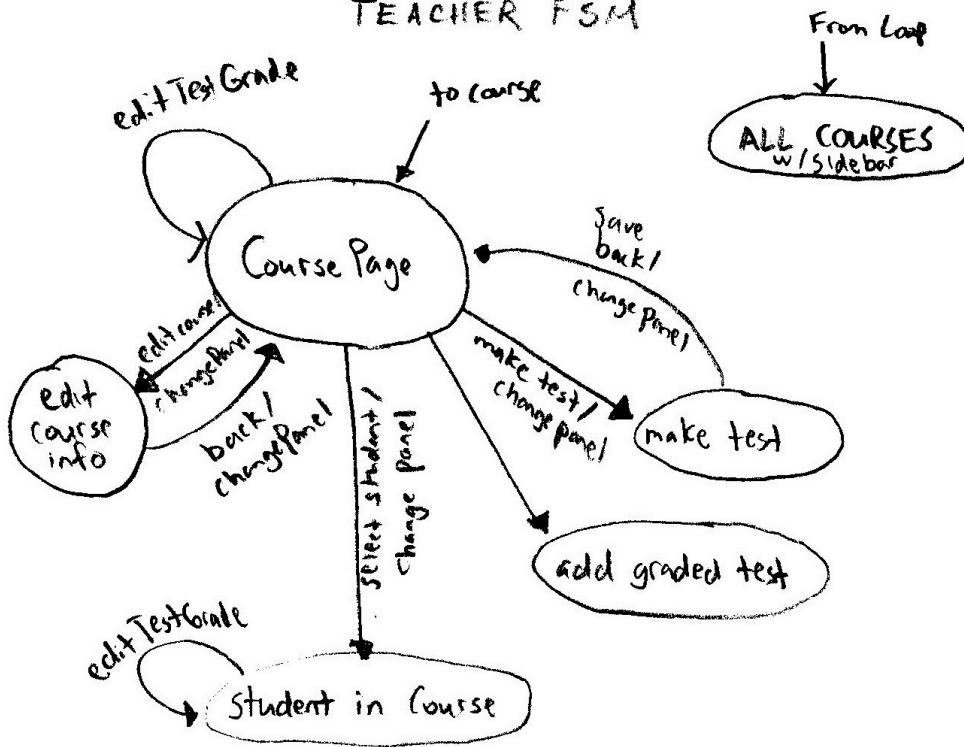
FSMs



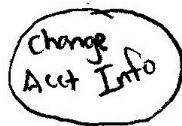
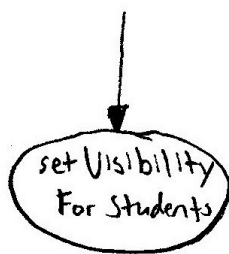
ADMIN FSM

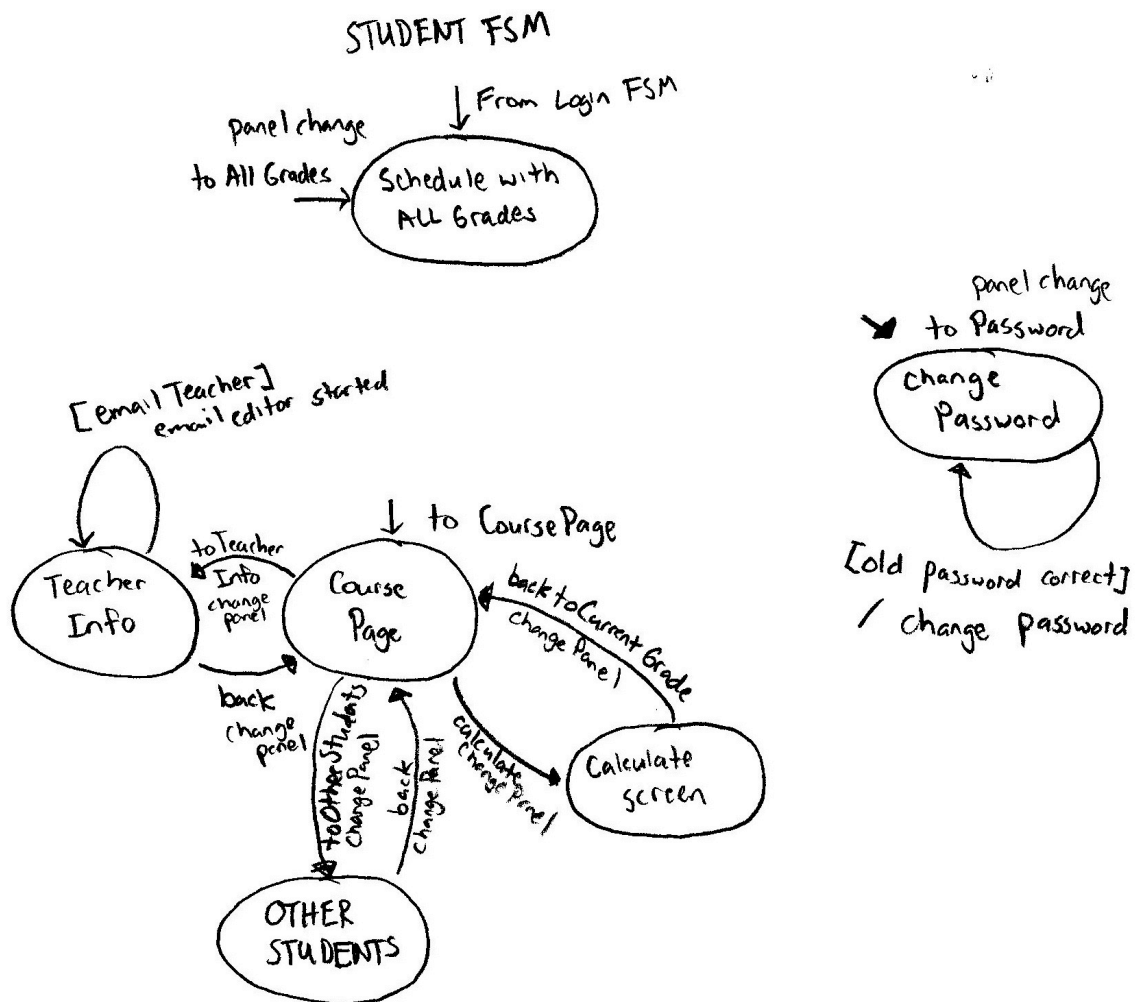


TEACHER FSM



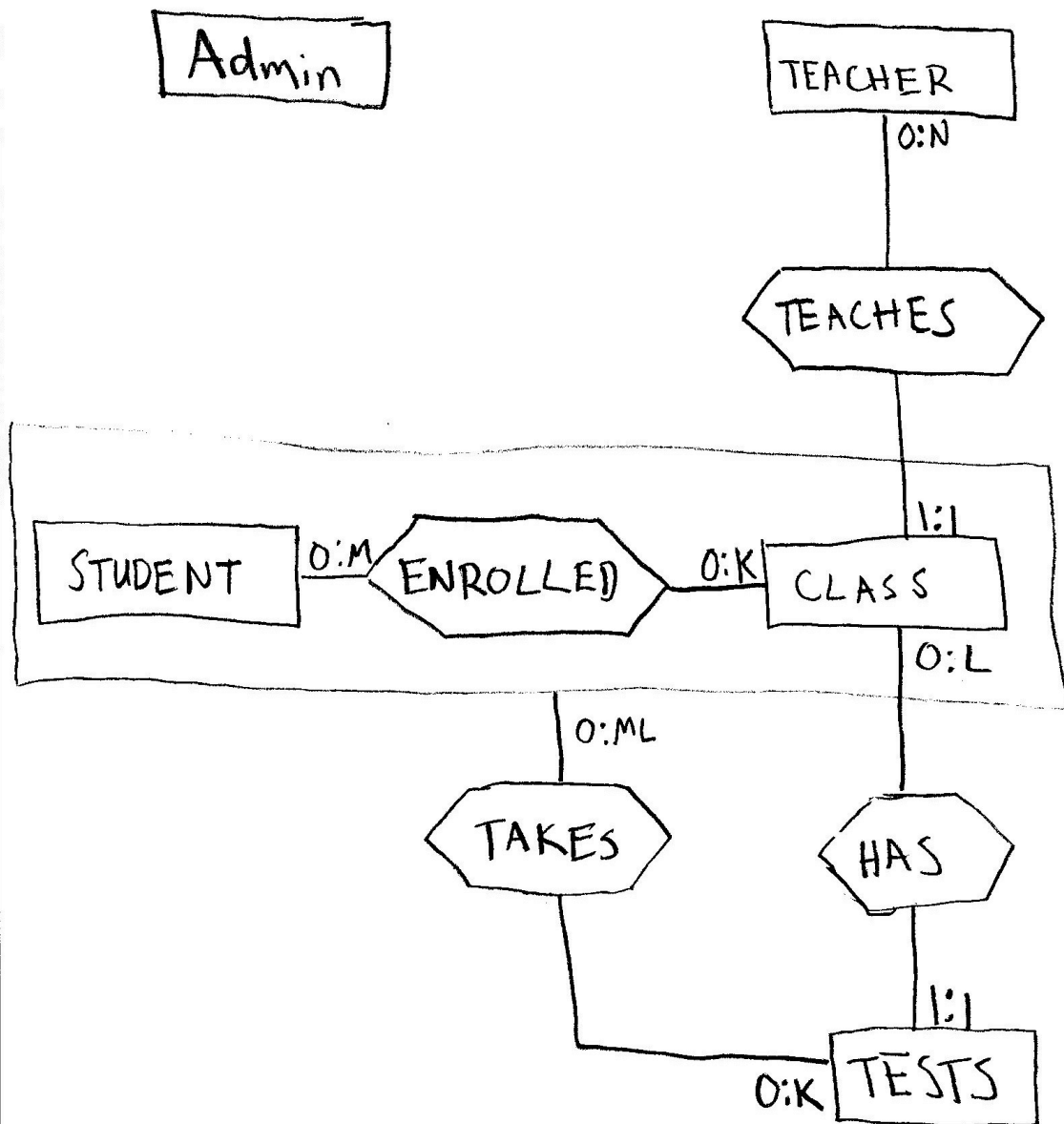
to settings





Note on FSMs: In order to achieve the flexibility goal listed, many paths and transitions need to be available in our FSMs. Because of the sidebar, the student can always get to the schedule screen, the change password screen, and the course page screen. Likewise for the teacher being able to get to any screen quickly. That being understood, there are understood paths in the FSMs that would be entirely too tedious and confusing to show.

ERM DIAGRAM



Entities

Teacher

(userID, password, firstName, lastName, gender, age, streetAddress, state, zipCode, city, classCount, email, officeLocation, phone)

Class

(classID, teacherID, description, studentCount, roomNo, meetingTime, semester, year, testCount, AverageGrade, PointsPossible, gradingScale, capacity)

Student

(userID, password, firstName, lastName, gender, age, streetAddress, state, zipCode, city, classCount, email, parentName, phone)

Tests

(testID, Description, grouping, PointsPossible, AverageGrade, gradingScale, weight, dateGiven)

Admin

(userID, password, firstName, lastName, gender, age, streetAddress, state, zipCode, city, classCount, email, officeLocation, phone)

Relationships

Teaches

(teacherID, classID)

Has

(classID, testID)

Enrolled

(studentID, classID, ClassGrade, PointsMade, Percentage)

Takes

(studentID, classID, testID, PointsMade, DateTaken, Percentage)

Link Descriptions

- 0:n on Teacher to teaches link:

A teacher may be teaching anywhere from 0 to n classes. (0 mostly applies to new teachers without classes.

- 1:1 on class to teaches link:

A class will be taught by exactly one teacher.

- 1:m on students to enrolled link:

A student can be enrolled in at minimum 0 classes (during summer, for example), but at most m classes.

-0: k on class to enrolled link:

A class can have anywhere from 0 (when a class is initially created, for example) to k students in it.

-0:L on class to tests link:

A class can have anywhere from 0 students (usually only when it is initially created and students have not yet scheduled it) to L students.

-1:1 on tests to has link:

A test is part of exactly one class and not used by any other.

-0:m*L on (student enrolled in a class) to takes link:

A student enrolled in as many as m classes with L tests in each, can have up to m*L tests.

-0:k on tests to takes link:

If there are k students in a class, at most there need be k tests.

Menus

The most efficient way to achieve the level of flexibility that we want is to use a sidebar in our program. This sidebar would always be on the screen, always allowing whichever user to go to the most important or most frequently used places in the program they need to go.

The following menus are listings of the items we want in each menu. We will deal with aesthetics, colors, and fonts later after we are sure that the menus are as functional as they need to be.

Student Side Bar

~~<| Back Forward|> //link to the previous or next page~~ cut due to time
<Student Name> //not a link. Just showing the student's name
-All Grades //link to the all grades screen
-Schedule
 <List of Courses> (in alphabetical order) an example follows
 English //each of these is a link to the course page for that subject
 Math
 .
 .
 .
-Change Password //link to the change password screen
-Logout //logs user out of system

Account Screen

Enter Previous Password <text field>
Enter New Password <text field>
Reenter New Password <text field>
Submit //when clicked, this will change the password to whatever is in the new
Change email address <text field>
Submit
//password text field only if the old password is correct.

All Grades screen

<Student name>'s Grades

<List of Courses>

<Overall Course Grade>

<Up to last 5 grades for course tests>

English **B**
Test3 – 85 B Quiz3 – 75 C Test4 – 80 – B ...

Math **A**
Test1 – 93 A Test2 – 88 B Homework5 – 75 – C ...

.
.

/**Each subject title is a link to that course's page. These are the only links on this screen. The screen is set up this way because each subject may have different descriptions and numbers of tests. The purpose of this screen is not to overwhelm anyone with a bunch of numbers on a screen. It is to show the most recent grades in all the courses and the overall grade in each course. */

Student: ...<student-name>...

Course : ...<course-name>.... Semester: ...<fall-or-spr>... Year:

...<???>...

Garding scale: A=85-100, B=75-84, C=65-75, D=55-64, F=0-54

Scores for test, homeworks (including max-marks and percentage of grade, if applicable):

Tests(90%) : #1(100, 20%) #2(50, 20%) #3(100, 25%) #4(100, 25%)

Student : 90 30 --- ---

ClassAverage : 87 40 --- ---

Homeworks(10%): #1(10) #2(15) #3(15) #4(10)

Student : 10 10 5 10

ClassAverage : 9 8 12 10

Current grade : Total Test Scores = 120(150), weighted-value = 72(90)

Total Homework Scores = 35(60), weighted-value = 6(10)

Total = 78(100), grade = B

Teacher Info

Find other students

Calculate Projected Final Grade

Projected Total	590	650	%	%
Teacher Info	Find other students	Back to Current Grade	Calculate	

```
<|Back Forward|> //links to the next and previous pages
<Teacher Name>
-All Courses
-Account
-Classes
    <List of Classes Currently Teaching(alphabetical)>
    //each a link to that course page
-Settings //link to the settings screen where teacher sets visibilities
-Logout //logs user out of the system
```

<Course Name>
 <List of Students in course (alphabetical)> <Descriptions>Total Grade
 .<Grade of Student1>.....<Total Grade of student>

	Test1	Test 2	...	Total Grade
Alan	90 A	82 B	...	81.5 B
Bobby	85 B	76 C	...	72.1 C
Carl	95 A	73 C

```
/** any individual grade can be clicked to turn into a text field which can be edited.
A "save changes" button will be at the bottom right hand corner of the screen. Any
attempt to change pages without saving will display an "are you sure" box*/
```

Student in a class grades screen

<Student Name>'s <Class Name> grades
<Description> <Grade made> <percentage> edit

Math-Carl's Grades

Test1 90/95 90% A edit

.

Save Changes All <Student Name>'s grades (only in elem)
/**edit changes the student's grade to an editable text box. Save changes saves all
changes when clicked. All <Student Name>'s grades goes to a page that shows all of
that student's grades. */

Enroll Student Screen

Select Class to add to <drop down list of classes>
Select Student <drop down list of students in class>
What to do? <drop down add, edit, delete>

will be prompted "are you sure"

*note, Select student will only be available for elementary because one student is
added to all classes.*

User Functions in a programming sense

Generic User

Function	Parameters	Other Inputs	Value Returned	Other Outputs
average	arrayOfGrades		average	
getTestGrade	testID, studentID, classID		testGrade	
Back	lastPage			Main frame changed
Forward	nextPage			Main frame changed
toPassword				Main frame changed
changePassword	newPassword, confirmPassword, oldPassword,			Password changed
startEmailEditor				
sendRequestForLogin	type, name			request sent

Student

Function	Parameters	Other Inputs	Value Returned	Other Outputs
toAllGrades	studentID			Main frame changed
toOtherStudents	courseID			Main frame changed
emailTeacher				Email Editor started
toTeacherInfo	courseID			Main frame

				changed
calculateProjectedGrade	arrayOfGrades, arrayOfExpectedGrades		projectedGrade	
backToCurrentGrade	courseID			Main frame changed
toCoursePage	courseID			Main frame changed

Teacher

Function	Parameters	Other Inputs	Value Returned	Other Outputs
editTestGrade	testID, studentID, classID, newGrade			testGrade changed

This program cannot currently be run on apple computers or computers without the latest version of java installed. Macs rely on their own service, software update, to handle any updates. For this reason, they do not have the latest version of java. This program, however, uses a newer class from the swing package in the building of the GUIs.

This program uses a database, and, as a result, needs a driver for this database. The driver used was com.mysql.jdbc.Driver. This was included in a jar so the program could run.

Appendix of classes

Main.java – includes the main method

Classes based on the Entities

User.java

Teacher.java

Admin.java

Student.java

Takes.java

Subject.java

Enrollment.java

Test.java

DBAccess.java –for accessing the database

JPanel subclasses for the GUI elements

Initial GUIs

Logon.java

Registration.java

For the Student

SideBar.java

AllGrades.java

Course.java

Account.java

CalculateProjected.java

For the Teacher

AllCourses.java

TCoursePage.java

TSettings.java

TSidePanel.java

TAccount.java

TStudentGrade.java

For the Admin

AdminMain.java

AdminApprove.java

AdminMakeSubj.java

Enroll.java

UserReqs.java