LMSlite

Project Report

Contributors:

Paul Davis, Benjamin Benson, Kaleab Teka

1. Project Description

a. **Overview:** LMSlite is a learning management system that is kept as simple as possible to run swiftly on lower-end hardware, with a graphical user interface.

b. **System Overview:** The data that will be accessed or managed by the system include:

   i. **Student:** id, degree, admittance date, name, address, phone, email, and password

   ii. **Staff:** id, department, hiring date, name, address, phone, role, email, password, and office number

   iii. **Course:** department, name, description, credit hours

   iv. **Section:** associated course, conducting instructor, room #, time

   v. **Assignment:** associated section, name, description, due date, and points possible

   vi. **Submission:** associated assignment, submitting student, date of submission, points earned, file submitted.

c. **Constraints:** Local database is to be realized using SQLite, and GUI is to be developed using Qt Creator.

d. **Version Control:** https://github.com/Pawls/LMS

2. Software Development Life-Cycle

a. **Agile Processes:** Why choose agile processes?

   i. **Knowledge is not lost if one programmer leaves:** Your project will continue at a steady pace.

   ii. **Group ownership of code:** We are all responsible for the high quality code that we deliver.

   iii. **Reliable code testing:** Programmers code is tested by their peers, to ensure lack of bias.

3. Requirements Phase

a. **Functional Requirements:**

   i. Must be two accessing modes: student and administrator.

   ii. Students and administrators will login through a shared window.

   iii. Software features for students and administrators should be in separate windows.

   iv. The administrator must be able to insert, update, and monitor all processes in the software.

   v. Administrators should be able to view student details.

   vi. Information available in the system will include student's name, student's ID, registered course, assignment grades, and GPA calculation

b. **Non-functional Requirements:**

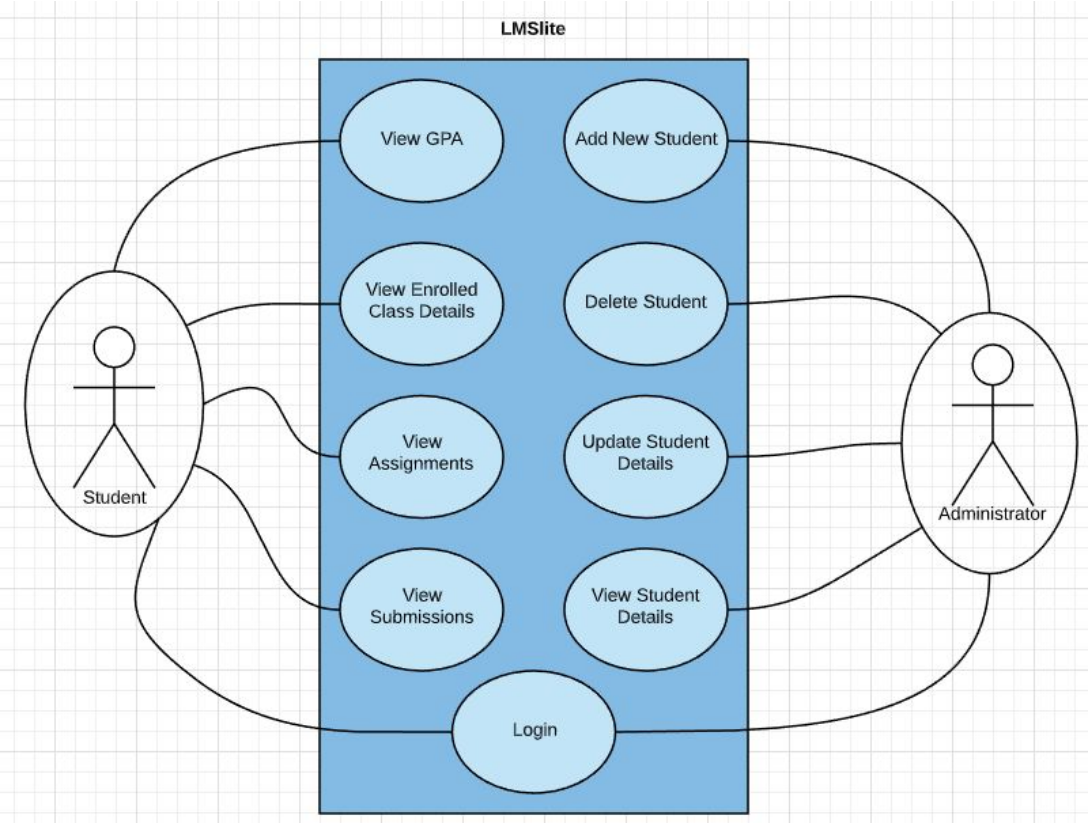
   i. Passwords must be kept private

   ii. No student should be able to access another student's data

   iii. The software must not be impaired by larger amounts of data. Speed must be maintained throughout the growth of the school.

c. **Product Domain**: The software will be used on a college campus by both students and administrative staff.

d. Use Case Diagram:



e. Detailed Use Cases:

i. Login: Student
   1. User enters login credentials
   2. System verifies identity and user-type
   3. System opens student window

ii. Login: Administrator
   1. User enters login credentials
   2. System verifies identity and user-type
   3. System opens administrator window

iii. View Enrollment Details: Student
   1. User enters login credentials
   2. System verifies identity and user-type
   3. System opens student window
   4. System displays enrollment details on the “Home” tab

iv. View Assignment Details: Student
   1. User enters login credentials
   2. System verifies identity and user-type
   3. System opens student window
   4. User clicks on the “Assignment” tab

5. System displays assignment details

v. View GPA: Student
    1. User enters login credentials
    2. System verifies identity and user-type
    3. System opens student window
    4. System calculates GPA and displays under the enrollment table

vi. View Submission Details: Student
    1. User enters login credentials
    2. System verifies identity and user-type
    3. System opens student window
    4. User clicks on the “Assignment” tab
    5. System displays submission details with corresponding assignment

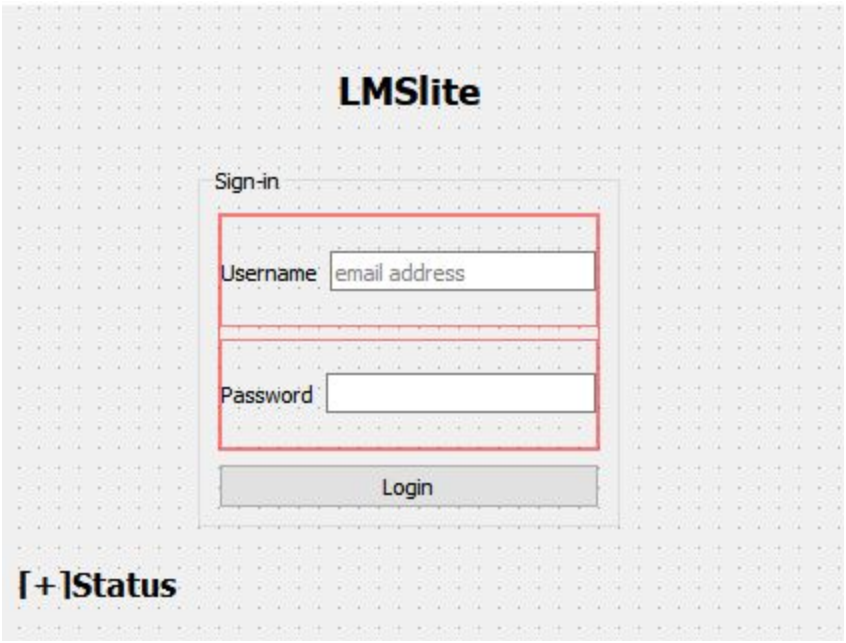
vii. Add New Student: Administrator
    1. User enters login credentials
    2. System verifies identity and user-type
    3. System opens administrator window
    4. User enters student details in the edit boxes
    5. User clicks “Add Record”
    6. System prompts for a temporary user password.
    7. User clicks “Ok” after the password is entered to complete.

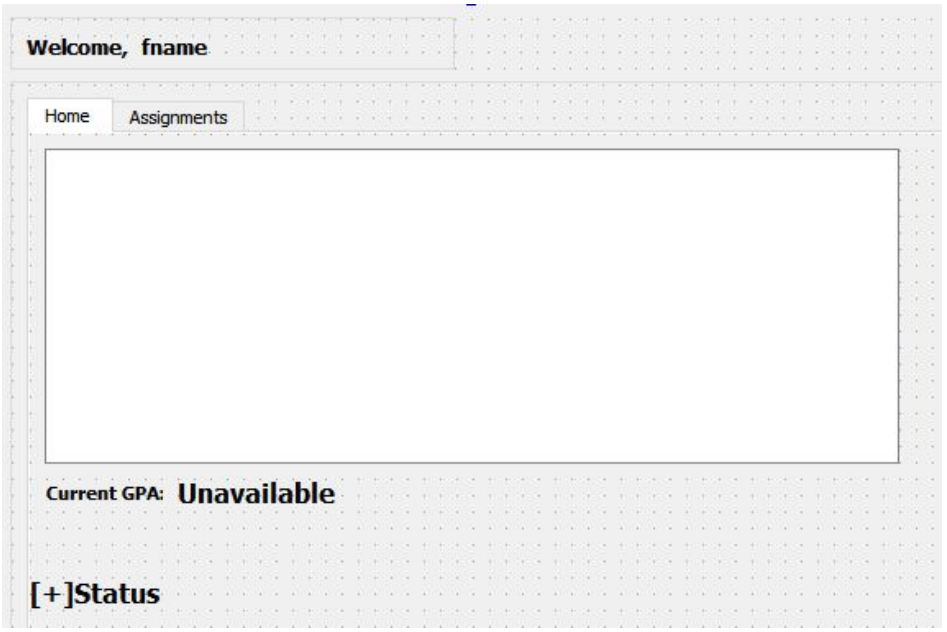
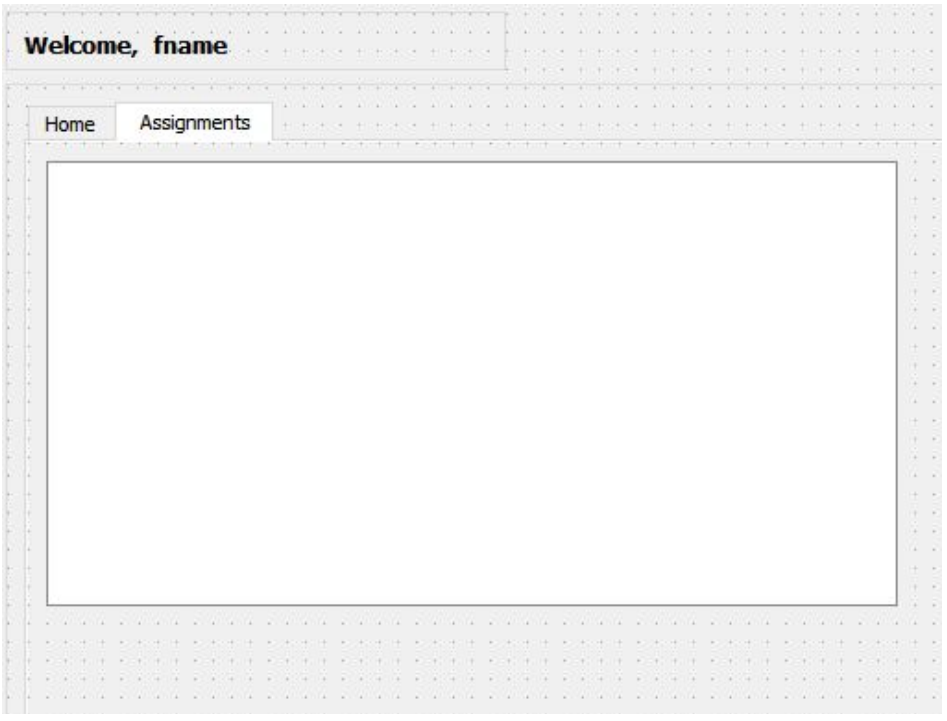
viii. Delete Student: Administrator
    1. User enters login credentials
    2. System verifies identity and user-type
    3. System opens administrator window
    4. User double-clicks student record to populate details in the edit fields
    5. User clicks “Delete Record”
    6. After giving confirmation, the system removes the record.

ix. Update Student: Administrator
    1. User enters login credentials
    2. System verifies identity and user-type
    3. System opens administrator window
    4. User double-clicks student record to populate details in the edit fields
    5. User enters student detail changes in the edit boxes
    6. User clicks “Update Record”
    7. System updates student details.

f. Diagrams: *See Next Page.*

g. User Interface:

Login Page	
Administrator Main Window	

Student Window Tab 1	
Student Window Tab 2	

4. Implementation phase

a. **Coding Language:** All software coding was done in C++ and

b. **Software:** Qt Creator was used to build the GUI and organize all code.

   i. Qt is a cross-platform framework for designing and coding user-interfaces.

   ii. Libraries used include QModelIndex, QInputDialog, QMessageBox, QApplication, and QtSql

c. **Database:** Database was built using DB Browser for SQLite to import test data that was generated using Mockaroo.com

   i. **Schemas:**

Schema

Assignment LMS
assn_id | section_id | name | description | date_due | points_possible

Course LMS
course_id | dpt_id | name | description | credit_hrs

Degree LMS
degree_id | dpt_id | name | description

Department LMS
dpt_id | admin_id | name | description

Enrolled LMS
enroll_id | section_id | student_id

Section LMS
section_id | course_id | instructor_id | room | time

Staff LMS
staff_id | dpt_id | date_hired | first_name | last_name | address | city | state | zip_code | phone | email | role | office | password

Student LMS
student_id | degree_id | date_admitted | first_name | last_name | address | city | state | zip_code | phone | email | password

Submission LMS
sub_id | __rand | assn_id | enroll_id | date_sub | points_earned | submission

d. **Software-Database interface:** Using the Qt API, we access the database using SQLite queries.

   i. **Student Login:**

```
SELECT * FROM Student
WHERE email=? AND password=?
```

   ii. **Administrator Login:**

```
SELECT * FROM Staff
WHERE email=? AND password=?
```

   iii. **Student Home Tab:**

```
SELECT
        Enrolled.section_id,
        Course.name AS 'class',
        Staff.last_name AS Instructor,
```

```
        credit_hrs,
        CASE
                WHEN (((sum(points_earned)*1.00)/sum(points_possible))*100) >= 90 THEN 'A'
                WHEN (((sum(points_earned)*1.00)/sum(points_possible))*100) >= 80 THEN 'B'
                WHEN (((sum(points_earned)*1.00)/sum(points_possible))*100) >= 70 THEN 'C'
                WHEN (((sum(points_earned)*1.00)/sum(points_possible))*100) >= 60 THEN 'D'
                ELSE 'F'
        END AS letter_grade
FROM Enrolled
JOIN Section ON Enrolled.section_id=Section.section_id
JOIN Course ON Section.course_id=Course.course_id
JOIN Staff ON Staff.staff_id=Section.instructor_id
LEFT JOIN Submission ON Enrolled.enroll_id=Submission.enroll_id
JOIN Assignment ON Enrolled.section_id = Assignment.section_id
WHERE student_id = ?
GROUP BY Enrolled.section_id;
```

iv. Student Assignment Tab:

```
SELECT
        Enrolled.section_id,
        Assignment.name as assignmentName,
        Assignment.description as assignmentDescription,
        Assignment.date_due,
        Assignment.points_possible,
        Submission.points_earned,
        Submission.submission
FROM Submission
JOIN Enrolled ON Submission.enroll_id=Enrolled.enroll_id
JOIN Assignment ON Submission.assn_id = Assignment.assn_id
WHERE Enrolled.student_id = ?
ORDER BY Assignment.section_id;
```

v. Admin Window Display:

```
SELECT
        student_id, degree_id, date_admitted, first_name, last_name, address, city, state,
        zip_code, phone, email
FROM Student
```

vi. Add Record:

```
INSERT INTO
Student(degree_id,date_admitted,first_name,last_name,address,city,state,zip_code,phone,ema
il,password)
VALUES
(:degree_id,:date_admitted,:first_name,:last_name,:address,:city,:state,:zip_code,:phone,:email
,:password)
```

vii. **Delete Record:**

```
DELETE FROM Student
WHERE student_id=?
```

viii. **Update Record:**

```
UPDATE Student SET
    degree_id=:degree_id,
    date_admitted=:date_admitted,
    first_name=:first_name,
    last_name=:last_name,
    address=:address,
    city=:city,
    state=:state,
    zip_code=:zip_code,
    phone=:phone,
    email=:email
WHERE student_id=?
```

5. Testing and Debugging Phase

a. Test Case Summary: Attempt to insert student as administrator

i. **Test Procedure:**

1. Login as administrator.
2. Enter student details in edit fields.
3. Click “Add Record”
4. Enter temporary password and click “Ok”

ii. **Test Data**

1. Login Username: ifairhead0@yandex.ru
2. Login Password: J7mzuZ
3. Student details
   degree_id: 1
   date_admitted: 8/1/2017
   first_name: Paul
   last_name: Davis
   phone: 555-555-5555
   email: paulwall@blank.com
   address: 123 Nunya Ln
   city: Houston
   state: Texas
   zip_code: 77777
4. Temporary password: 8675309

iii. **Expected Result**

1. Popup message indicating success
2. New student should populate at the bottom of the list

iv. **Actual Result**
    1. If successful, the result is as expected.
    2. If failed, there is a popup indicating failure.
v. **Status:** Success

b. Test Case Summary: Attempt to delete student as administrator

i. **Test Procedure:**
    1. Login as administrator.
    2. Double-click student to delete
    3. Click “Delete Record”
    4. Confirm deletion.
ii. **Test Data**
    1. Login Username: ifairhead0@yandex.ru
    2. Login Password: J7mzuZ
    3. Student ID: 202
iii. **Expected Result**
    1. Popup message indicating success
    2. Student should disappear from the list.
iv. **Actual Result**
    1. If student is selected, the result is as expected.
    2. If no student selected, there is a popup indicating failure.
v. **Status:** Success

c. Test Case Summary: Attempt to update student as administrator

i. **Test Procedure:**
    1. Login as administrator.
    2. Double-click student to update
    3. Change student details
    4. Click “Update Record”
    5. Confirm update.
ii. **Test Data**
    1. Login Username: ifairhead0@yandex.ru
    2. Login Password: J7mzuZ
    3. Student details
       degree_id: 2
       date_admitted: 9/2/2018
       first_name: Pablo
       last_name: Davis-Marquez
       phone: 555-123-4567
       email: pablowall@blank.com
       address: 999 Nunya Ln
       city: Chicago
       state: Illinois
       zip_code: 44444

iii. **Expected Result**
    1. Popup message indicating success
    2. Student should update in list
iv. **Actual Result**
    1. If student exists, the result is as expected.
    2. If student does not exist, there is a popup suggesting “Add Record” instead.
v. **Status:** Success

d. Test Case Summary: Attempt to view student assignments as student

i. **Test Procedure:**
    1. Login as student.
    2. Select Assignment tab
ii. **Test Data**
    1. Login Username: paulwall@blank.com
    2. Login Password: 8675309
iii. **Expected Result**
    1. System should reflect the following results that were obtained from SQL query:

section_id	assignmentName	assignmentDescription	date_due	points_possible	points_earned	submission
1	Lotlux	magna ac consequat metus sapien ut nunc vestib...	8/10/2020	25	25	sub1-1.txt
1	Alpha	eget nunc donec quis orci eget orci vehicula cond...	7/20/2020	25	20	sub1-2.txt
2	Cookley	tincidunt nulla mollis molestie lorem quisque ut e...	8/9/2020	100	90	sub2-1.txt
2	Subin	ultrices posuere cubilia curae nulla dapibus dolor...	8/28/2020	50	50	sub2-2.txt
3	Fix San	a nibh in quis justo maecenas rhoncus aliquam l...	6/1/2020	100	100	sub3-1.txt
3	Andalax	at turpis a pede posuere nonummy integer non v...	5/31/2020	100	70	sub3-2.txt
4	Bamity	aliquam sit amet diam in magna bibendum imper...	6/4/2020	100	80	sub4-1.txt
4	Treeflex	quis justo maecenas rhoncus aliquam lacus mor...	5/8/2020	10	0	sub4-2.txt

iv. **Actual Result**
    1. Results were as expected.
v. **Status:** Success

e. Test Case Summary: Attempt to view enrolled class details as student

i. **Test Procedure:**
    1. Login as student.
    2. Select Home tab
ii. **Test Data**
    1. Login Username: paulwall@blank.com
    2. Login Password: 8675309
iii. **Expected Result**

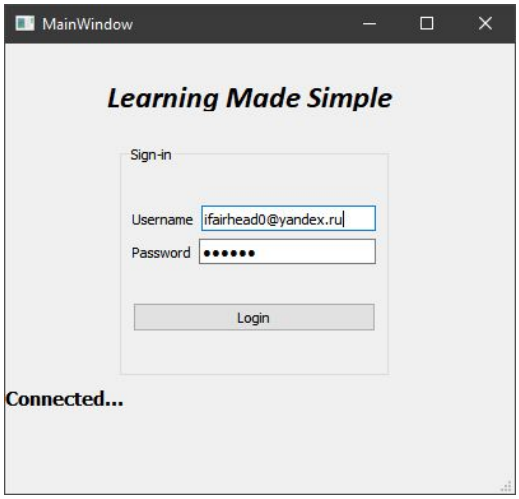
1. System should reflect the following results that were obtained from SQL query:

section_id	class	Instructor	credit_hrs	letter_grade
1	PSY2433	Raatz	2	A
2	HIS3200	Levet	1	A
3	MAT1824	Carver	3	B
4	MAT1829	Gowlett	3	C

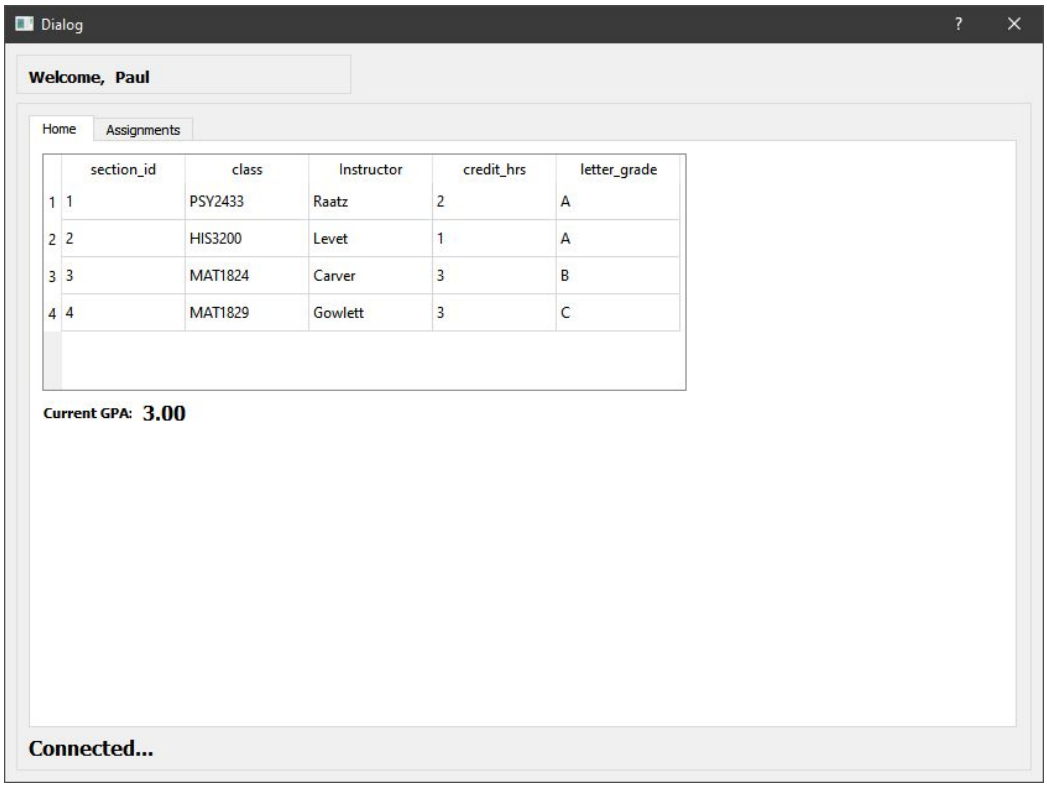
iv. **Actual Result**
    1. Results were as expected.
v. **Status:** Success

6. Results

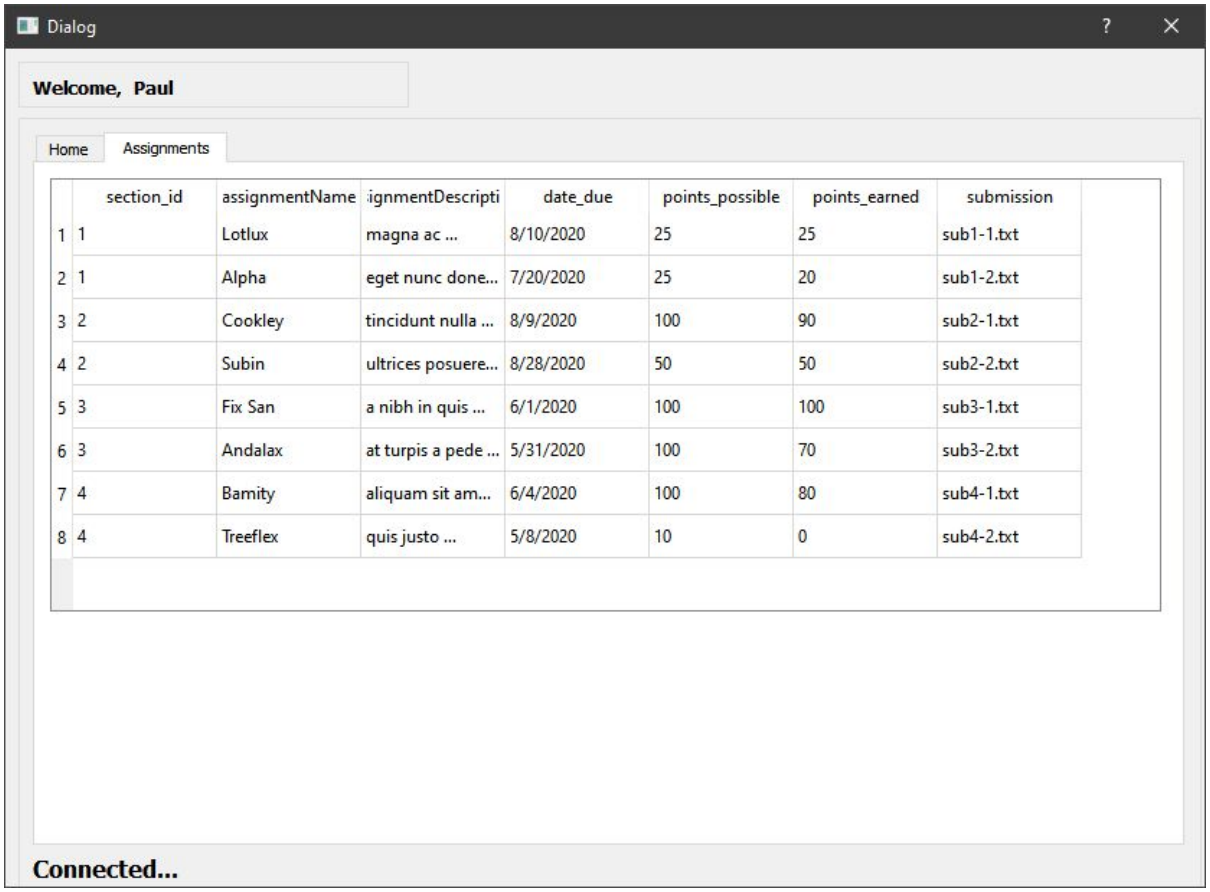
a. Login Page



b. Student Home tab

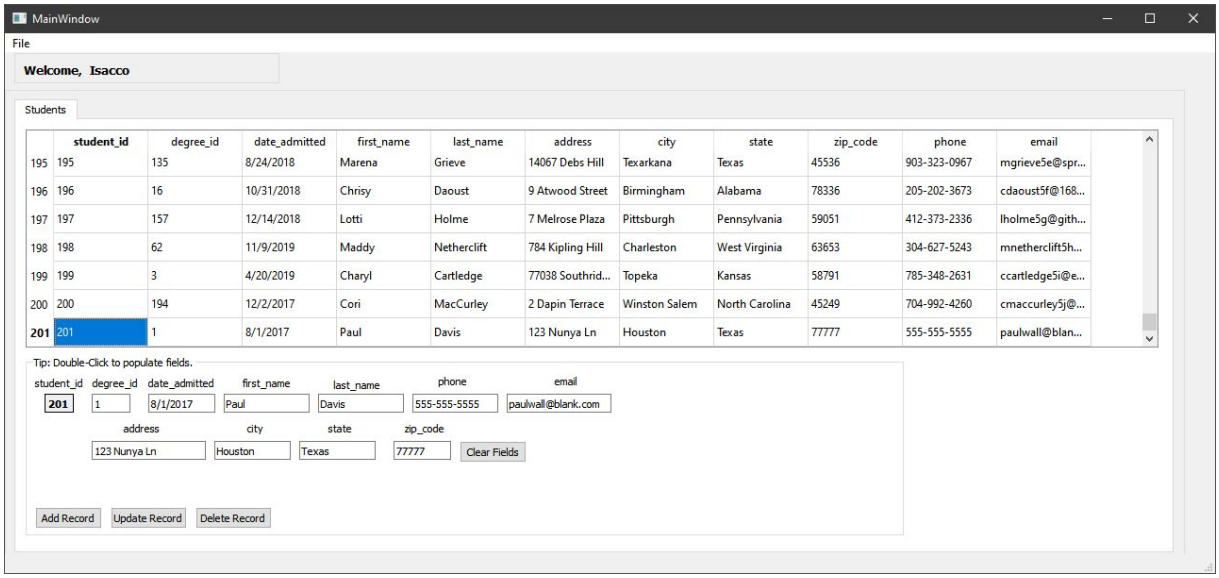


c. Student Assignment tab



	section_id	assignmentName	ignmentDescripti	date_due	points_possible	points_earned	submission
1	1	Lotlux	magna ac ...	8/10/2020	25	25	sub1-1.txt
2	1	Alpha	eget nunc done...	7/20/2020	25	20	sub1-2.txt
3	2	Cookley	tincidunt nulla ...	8/9/2020	100	90	sub2-1.txt
4	2	Subin	ultrices posuere...	8/28/2020	50	50	sub2-2.txt
5	3	Fix San	a nibh in quis ...	6/1/2020	100	100	sub3-1.txt
6	3	Andalax	at turpis a pede ...	5/31/2020	100	70	sub3-2.txt
7	4	Bamity	aliquam sit am...	6/4/2020	100	80	sub4-1.txt
8	4	Treeflex	quis justo ...	5/8/2020	10	0	sub4-2.txt

d. Administrator Window



	student_id	degree_id	date_admitted	first_name	last_name	address	city	state	zip_code	phone	email
195	195	135	8/24/2018	Marena	Grieve	14067 Debs Hill	Texarkana	Texas	45536	903-323-0967	mgrieve5e@spr...
196	196	16	10/31/2018	Chrisy	Daoust	9 Atwood Street	Birmingham	Alabama	78336	205-202-3673	cdaoust5f@168...
197	197	157	12/14/2018	Lotti	Holme	7 Melrose Plaza	Pittsburgh	Pennsylvania	59051	412-373-2336	lholme5g@gith...
198	198	62	11/9/2019	Maddy	Netherclift	784 Kipling Hill	Charleston	West Virginia	63653	304-627-5243	mnetherclift5h...
199	199	3	4/20/2019	Charyl	Cartledge	77038 Southrid...	Topeka	Kansas	58791	785-348-2631	ccartledge5i@e...
200	200	194	12/2/2017	Cori	MacCurley	2 Dapin Terrace	Winston Salem	North Carolina	45249	704-992-4260	cmaccurley5j@...
201	201	1	8/1/2017	Paul	Davis	123 Nunya Ln	Houston	Texas	77777	555-555-5555	paulwall@blan...