

Sprint Report #4

0.1 Team Members:

Marcus Berger

Dicheng Wu

Sponsor:

Jeff McGough

0.2 Prototype Progress

During sprint 3.5 the team organized the functionality and tied them together to create the working prototype. Other updates were made to fix known bugs and add a class search. While the team worked on putting the project together smaller bugs or needed pages that were implied by user stories were tackled as they came up. This resulted in too much time being used, so the student interface was not tackled. The student interface has since been dropped from the client's project requirements due to time.

Sprint 4 was dedicated to payroll and billing. First the client rewrote the user stories to provide more clarity to the team. After this the user stories were tackled by the group to produce the first draft of the payroll and billing interface. This included logging hours, payments, fees, rates, credits, and discounts. After this was done the team demoed the project for the client and got notes on the project as a whole. These notes will be tackled as part of the backlog in the next sprint.

0.3 Project deliverables of Sprint 3.5 and 4:

Sprint 3.5 had no defined deliverables, and was mostly clean up.

Sprint 4 deliverable was the first draft of the payroll and billing interfaces and functions, laid out in the user stories below.

0.3.1 Sprint 3.5 Backlog

1. Fix Assign Teacher to Class Dialog Bug

2. Add Class Search
3. Complete Role Sheet
4. Tie Functions Together
5. Fix Smaller Bugs

0.3.2 Sprint 4 Backlog:

1. View Teaching History
2. Look at The Current Amount Someone Owes
3. Generate an Invoice For The Amount Due
4. Look at Billing History
5. Apply Credits to a Students Account
6. Enter the Tuition and Fees Rate
7. Give Early Registration Discounts
8. Enter Staff Pay Rates
9. Enter a Full Payment for One Student
10. Enter a Full Payment for Several Students
11. Enter Payments From Multiple Sources For One or More Students
12. Compute Teacher Wages
13. Enter Staff Hours
14. Give Prorated Refunds

0.4 Sprint 3.5

During sprint 3.5 Class Search was added this page follows the same format as other search pages created in previous sprints. Also completed fixes to the assign teacher to class bug, role sheet completeness, and a few other minors bugs. At the end of the sprint the team tied what we had together so the product could run as a unit starting from the log in screen.

0.5 Prorated Refunds

The prorated refunds code will keep track of how much a class costed a student. If the student drops a class it will refund the remaining worth of the class to the students school credit field. This file is still in progress.

0.6 Enter Staff Hours

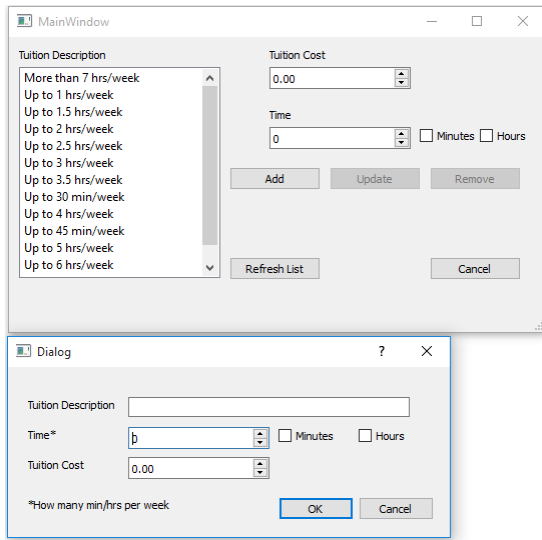
This page needs to undergo changes after meeting with client. The new version will allow a teacher to log their in class hours, office hours, trip hours, etc. These hours will then be paid out at their different rates respectfully. The hours will be logged as single numbers not by days or weeks.

0.7 Enter Teacher Wages

This was handled in the database in previous sprints assuming one pay rate. However in talks with the client the academy pays different pay rates based on what part of academy work they are doing. So the page will need to be updated once the list of possible pay rates have been provided by the client.

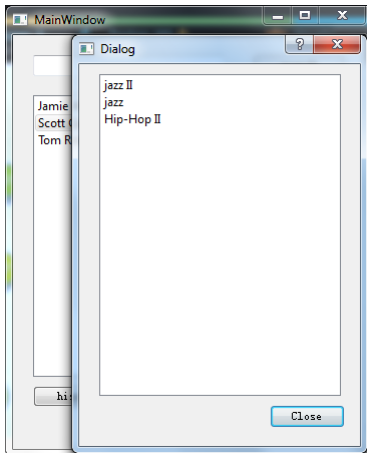
0.8 Enter Tuition Rates and Fees

These pages allow the user to look at the current tuition rates and fees. The user can then update existing rates, add new rates, or remove rates that are no longer needed. Tuition rates are logged in the database as flat minute rates. The user can enter new rates in the form of minutes or hours.



0.9 Teacher History

This page includes several components. Firstly, user can use this page to find a specific teacher by entering a full or partial name of teachers. Secondly, users can click the history button to open up another window which has a list of classes the teacher has taught.

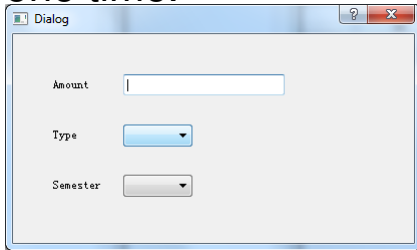


0.10 Set Semester

This page helps users to set the current semester from pool of semester and add a new semester to the system.

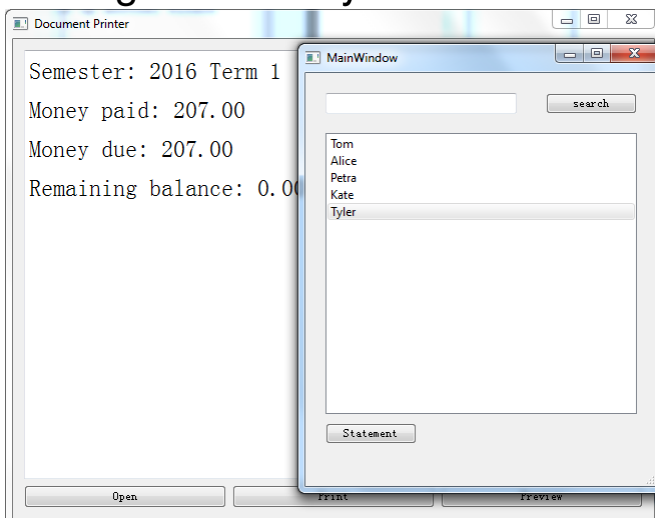
0.11 partial payment

This page includes several components. Firstly, user can use this page to find a specific student by entering a full or partial name of students. Secondly, users can click the pay button to open up another window which asks user to input amount of money paid, payment's method and semester paid. The user also can choose single or multiple students at one time.



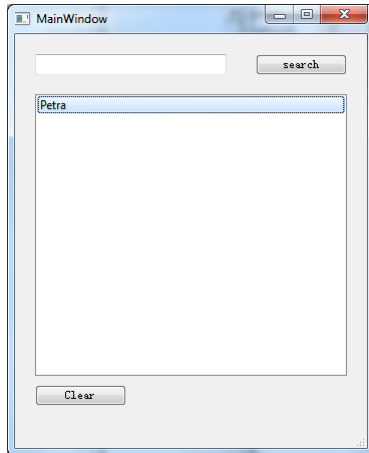
0.12 Student's owe

This page includes several components. Firstly, user can use this page to find a specific student by entering a full or partial name of students. Secondly, users can click the statement button to open up another window which shows the amount of students paid, the amount of students due and the balance of student's account. Users also can print the invoice generated by the window.



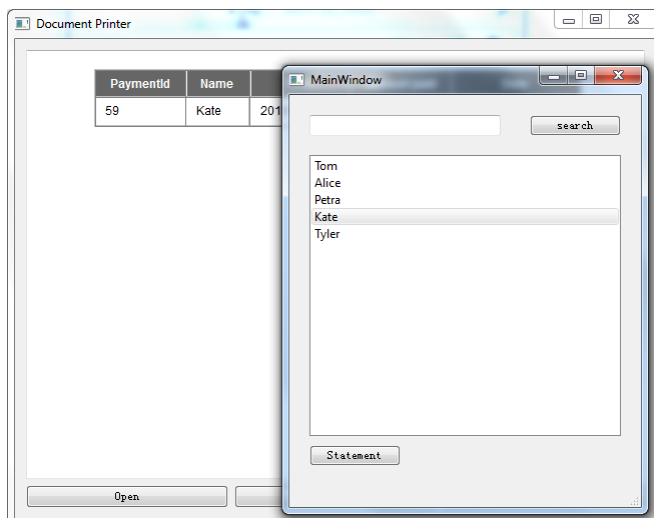
0.13 enter payment

This page includes several components. Firstly, user can use this page to find a specific student by entering a full or partial name of students. Secondly, users can click the clear button to clear the due of selected students. Users can select single or multiple students at one time.



0.14 bill history

This page includes several components. Firstly, user can use this page to find a specific student by entering a full or partial name of students. Secondly, users can click the statement button in order to get the list of payments. In addition, user can print the bill history.



0.15 Sprint 3.5 and 4 Issues

Some issues that were encountered during this sprint were:

1. During winter break the team wanted to tackle the student interface app. Building other required pages in the main local GUI made this a problem. Therefore while some implied GUI pages were fixed and created, the team was unable to get to the student interface. As a result the student interface was removed from the project requirements.
2. Spring semester ramp up - Coming back from a month break meant that the team needed to ramp back up and get used to the new classes and new commitments. This caused the team to move slower than it should have in this sprint.
3. Team communication - The meetings this sprint were less effective as team members would have other commitments, or in some cases lack the drive to do the work. This lessened as the sprint continued, but sometimes still came up.
4. Requirement Confusion - At times during this sprint the requirements were confusing. This resulted in a rewrite of the requirements for payroll and billing to provide more clarity and adjust for dropping the student interface. The rework helped the team greatly, however the user stories still sometimes left out core parts of how the academy does its work compared to other places. This resulted in spending more time than usually speaking with the client, and group friction.

0.16 Client Interactions

Client interactions came in the form of requirements reviews and a demo of the prototype at the end of the sprint. Topics included:

1. Progress reports on where we are in the project
2. Asking questions to refine functionality, understand academy needs, and update requirements.
3. Discuss the future of the project and ideas on how to execute them.
4. Prototype the current version of the prototype.

0.17 Group Meeting

The group has a standard meeting time of TTH 10:00 - 12:00 and once during the weekend and at home as needed.

Meeting can continue pass these times if needed, and other times during the weekend. However extra weekend times fluctuate and do not remain constant from week to week.

0.18 Work Distribution

Marcus:

1. 3.5 Fix Assign Teacher Bug
2. 3.5 Update GUI Functions, Minor Additions and Fixes
3. 3.5 Tie Various Functions Into a Single Prototype
4. Prorated Refund (Still in Progress)
5. Enter Teacher Hours (Still in Progress)
6. Teacher Pay Rates (Needs Updated With New Information)
7. Early Registration Discounts
8. Gives Credits
9. Enter and Update Tuition Rates
10. Enter and Update Fee Rates and Type (Needs Updated With New Information)
11. documentation
12. trello management

Dicheng:

1. 3.5 Tie function together
2. 3.5 Class Search Pages
3. 3.5 Updated Role sheet
4. 3.5 Research Linux Boxes

5. 3.5 Tie Functions Together Into a Single Prototype
6. Enter Payments From Multiple Sources For One or More Students
7. Look at The Billing History
8. Enter a Full Payment For One or More Students
9. Enter Payments From Multiple Sources For One or More Students
10. Generate an invoice for the amount due (Needs Updated)
11. Look at a The Current Amount a Student/Family Owes
12. View Teaching History
13. Auto and Manual discount giving
14. Set semester

Together:

1. Updated database construction
2. GUI page breakdown based on user stories and product backlog
3. Coded some functionality