Berkeley Academics Information Redesign

Submitted by:

Lulu Guo

Andrew Iskandar

May 4, 2005

With thanks to:

Advisor: Professor Ray Larson

Berkeley Academics Staff:
Jeannie Lee
Suzanne Suh
Sue Yi

Table of Contents

Appendices

Executive Summary

This report documents the findings of an information redesign project for *Berkeley Academics*, a local non-profit tutoring company. The goal of the project was to design and develop a web-based application that would replace many of the paper-based administrative tasks of our client and facilitate more efficient communication and collaboration. This became an area of need due to the expansion and growth of the company and the increased workload for the administrative staff and tutors. The design of this project would be accomplished through requirements gathering interviews, usability interviews and iterative design testing.

The most important findings of our project and the challenges we faced include:

1. **A web-based information system is needed to replace the current paper-based methods.** Many of the administrative tasks at *Berkeley Academics* were paper-based and this was manageable when the company was small. But with the growth of the company, a better system was needed in order to complete these tasks more efficiently, increase communication, and facilitate collaboration. A web-based information system was the best platform to meet this need.
2. **Application was seen as functional and easy to use; clients' expectations were met.** The client had a very favorable response to the application that we developed, expressing how useful it would be and the immediate need to implement it within their workflow.
3. **Designing and integrating a database schema was a challenge that influenced the development of the project.** *Berkeley Academics* had an existing web sign-up application with a database that needed to communicate with our application. It was decided early on to integrate our database schema with theirs in order to facilitate integration. This early database design choice influenced the development and implementation of the application.
4. **Presentation and organization of information is important to prevent overload and confusion to the user.** This application attempted to integrate many of the common administrative tasks for the *Berkeley Academics* staff. Thus, presenting information in a clear, concise, and understandable manner was very important in order to prevent confusion and information overload to the user. We also learned that other techniques, besides text (like color, placement, spacing, etc.) can be used to convey information as well.
5. **Gathering business and user requirements is very important.** We realized very early on that getting complete and thorough business and user requirements was very important. Since many of the design decisions for our proposed application centered around the business processes of *Berkeley Academics,* this early time investment would help us avoid difficulties and changes later on during development of the application.

There were areas for further development for the project, which include:

1. **During user tests, users suggested additional features**. Late in the development process of our application, users suggested additional feature requests. Because of the timing of these suggestions, implementing them would not have been feasible for the timeline of our project. Some of these suggestions include: payment reminders,

automatic re-application of students when their sessions expired, and drag and drop interaction.

2. **Multiple locations**. Our system was designed for tutor-student scheduling in one location. Since *Berkeley Academics* is planning to expand to multiple locations, being able to handle scheduling across these various locations could help centralize the administrative workload.
3. **Searching and querying.** Currently our system handles information requests through drop-down menus that show all of the options available (this is especially evident in student progress tracking.) A search and query system into the database could be useful when the dataset grows larger.
4. **Analysis of trends in the data.** The staff at *Berkeley Academics* realized that with all this information stored in the database, analyzing it to notice trends and patterns could be useful for marketing and curriculum planning.

Background

Berkeley Academics (www.berkeleyacademics.com) is a non-profit tutoring company based in Alameda, CA. Established in 2004, *Berkeley Academics* aims to establish an educational foundation in students, focusing their efforts through small-group and one-on-one tutoring and homework assistance. *Berkeley Academics* employs college students as tutors and reaches out to underprivileged students, through scholarship programs, in the surrounding community.

Currently, *Berkeley Academics* employs three full-time administrative staff who handle curriculum development, tutor-student scheduling, marketing, finance, and student progress tracking. In addition, they hire approximately 8-15 tutors. During a given period of time, they regularly see 20-30 students a week.

Berkeley Academics has an online sign-up page and some basic IT systems to handle their finances and word processing, but most of its administrative work is still paper-based and processed manually by the administrative staff. [1] Tutor scheduling is done using spreadsheets and printed out daily. Tutors record student progress by pulling their files from a filing cabinet and jotting down notes. And the administrative staff must process multiple files to handle payments and rescheduling.

Berkeley Academics has begun to expand to other cities, such as San Ramon, CA and conduct large-scale (over 200 students) academic summer camps. The administrative staff is excited about this future potential, but is concerned about the increased workload and burden this growth and expansion will place upon themselves. This is the problem area that this project hopes to address.

[1] See Appendix B for examples.

Project Timeline

Date	Phase	Tasks
11/05 - 12/06	User Analysis	Initial client interviews gathering information and documents about current work processes.
12/06 -01/06	Task Analysis; Requirements Gathering	Analyzed documents and interview notes to formulate high priority tasks. Used personas and scenarios to assist in this process. Gathered business requirement information for application through on-site client interviews and observations.
01/06 - 02/06	Initial Sketches and Lo-Fidelity Prototyping; Database Design	Designed paper-based and simple web-based mockups of user interface for application for users to test and evaluate initial design; Designed database augmentation and expansion to accommodate new application.
02/06 - 04/06	Hi-Fidelity and Final Prototype Development	Developed hi-fidelity and final prototype for final testing and deployment.
04/06 - 05/06	User Test; Final Report	Conducted final user test and analysis on client satisfaction; Wrote final report on process.

Needs Assessment

Problem Statement

Currently, much of the administrative work and record keeping at *Berkeley Academics* is paper-based. Not only does this consume precious resources and time, but it also causes a lot of difficulties in terms of recording keeping, communication and information coordination. Based upon our interviews with the administrative staff at *Berkeley Academics* we concluded that the following issues might be contributing to this:

1. The payment system is manual-based. The administrative staff needs to manually look up paper records in order to find out the payment status of a customer. Moreover, there is neither a notification for the staff when a customer makes a payment online (through *Paypal*) nor a reminder system when a payment is due.

2. The same kind of problems exists for history tracking of student enrollment and course progress. Since almost everything is paper-based, it is difficult for the administrative staff to answer questions pertaining to a student's enrollment status such as when was the student enrolled? How long has the student been enrolled? Which session(s) did the student attend? What is the student's current progress with his or her tutor? Student progress and attendance tracking is poorly supported by the current system. Without adequate record keeping, information retrieval tasks surrounding student records and progress cannot be effectively carried out.

3. Flexibility in tutor/student scheduling represents another problem. When a tutor or student cannot show up for a scheduled session, the staff need to check the schedules of the other tutors and select the one that is both available and capable of tutoring that student. This matching process usually involves a significant amount of time and needs to be made more efficient, especially given the fact that the business is growing and the number of tutors and students is expected to increase in the near future.

4. Online application process needs improvement. The administrative staff does not get notified when a new application comes in. The current version of *Berkeley Academic'* online application is a one-time process. It does not allow the user to modify their information or extend their enrollment from one session to another. It is not linked to their payment process. Therefore, after receiving an application online, the staff needs to rely on paper files to manually update all the information of a customer. Because of this inflexibility of the current online application system, the staff promotes a paper-based application process.

Personas

User 1: Scheduler/Administrator - Amy

Amy is a thirty-three years old female, college educated, married, and has two kids. She is familiar with computers, but mainly uses it for e-mail, word processing, and spreadsheet creation. She is one of the co-founders of *Berkeley Academics* and is part of the administrative staff. Her primary responsibilities are to process student applications, schedule their tutoring sessions, and to handle payment scheduling.

These tasks are all inter-related as a student's application determines their schedule and their schedule determines their payments. She needs to adjust payments and scheduling when a student is unable to attend a session. She records all this information in the paper records for each student and in spreadsheets. She needs to remember all of this information and keep track of information as it comes in. Most of it comes through the phone or direct interaction with the students' parents.

Amy's Goals:

- Amy hopes to expand *Berkeley Academics* to cover more regions in the northern Bay Area besides Alameda and San Ramon.
- She hopes her daily administrative work at *Berkeley Academics* can be done more effectively so that she can spend more time with her family.
- She is interested in enhancing her computer skills and knowledge in web-related technologies. She is thinking about signing up for one or two computer courses at a local community college.

User 2: Administrator/Tutor – Jane

Jane is thirty years old, college educated, and just married. She considers her computer indispensable and often uses it for her daily life as well as her profession. Although she has no formal programming knowledge, she does some simple web programming. She is one of the co-founders of *Berkeley Academics* and is also a part of the administrative staff.

Her primary responsibilities at *Berkeley Academics* are to develop the curriculum and to keep track of student progress. Her main interaction is with the students, tutoring and teaching them. She also helps arrange the scheduling of students with suitable tutors.

Jane's Goals:

- As another founder of *Berkeley Academics*, Jane also wants to grow the company, not only in terms of geographical coverage, but also in the comprehensiveness of the services provided.
- In the long term, Jane plans to go back to graduate school for a higher degree in Education. After that, she would like to become a researcher.

User 3: Tutor – Stanley

Stanley is currently a junior at U.C. Berkeley and is studying Cognitive Science. He is a paid tutor at *Berkeley Academics* helping students with their math and science homework. He is very familiar with computers using them for e-mailing, web-surfing, music listening, and movie watching.

His primary responsibility at *Berkeley Academics* is to tutor the students. He has given his schedule of availability to the administrative staff at *Berkeley Academics* and is told when to come in to teach the students. He has a regular schedule and generally teaches the same students each week. He needs to cater each session to the progress and needs of each student and he keeps track of this using paper records. Sometimes he needs to cover for other tutors when they are unavailable or to teach an unfamiliar student because of a scheduling change.

Stanley's Goals:

- One of the short-term goals for Stanley is to graduate with good grades and find himself a computer graphic design job at an IT company.
- He hopes to move to San Francisco after graduation so that he can better enjoy the city life there.

Task Analysis

User 1: Scheduler/Administrator - Amy

Applications
- Process paper applications and enter them into the system
- Create files for new students
- Note special needs and desires of students

Payments
- Process payments
- Process refunds
- Keep track of payment schedule
- Keep track of overdue payments

Scheduling
- Create tutor/student schedule from student applications
- Keep track of student attendance
- Process rescheduling of student/tutor
- Keep track of tutor schedules and skills

Informational
- Lookup records for student history
- Lookup student progress for parents

User 2: Administrator/Tutor - Jane

Applications
- Get notified of new online applications
- Log into the application system to check out new applications.

Scheduling
- Look up a particular tutor's schedule
- Query about availability of tutors
- Record rescheduling information (one time/week)

Payments
- Get payment report/summary of a customer
- Get reminded if a payment is due

Record keeping
- Ensuring information from online application goes into database
- View and update the student information through the system
- Query student information

User 3: Tutor - Stanley

Tutoring
- Record student progress
- Look up student progress to formulate lesson plan
- Add special notes

	Amy	Jane	Stanley
APPLICATIONS			
Process paper applications	H	M	N/A
Note special client needs	H	M	N/A
PAYMENTS			
Process payments	H	M	N/A
Process refunds	H	M	N/A
Keep track of payment schedule	M	M	N/A
Keep track of overdue payments	H	M	N/A
Reminder of payment schedule	M	M	N/A
SCHEDULING			
Create tutor/student schedule	M	M	N/A
Keep track of student attendance	M	M	N/A
Process rescheduling of student/tutor	H	H	N/A
Keep track of tutor schedules and skills	L	L	N/A
Query about tutor availability	L	M	N/A
RECORD KEEPING			
Lookup records for student info/history	H	H	N/A
Lookup student progress for parents	L	L	N/A
TUTORING			
Record progress of students	N/A	N/A	H
Look up student progress to formulate lesson plan	N/A	N/A	H
Add special notes	N/A	N/A	M

H: High; M: Medium; L: Low (indication of task importance)

Scenarios

Scenario 1: Parent submits a paper application for a new student.
Karen comes to the *Berkeley Academics* office with a paper application for her son, Paul, and gives it to one of the administrative staff. Amy takes the application and checks to see if the student or parent is already in the system. Since Karen is a new customer and Paul a new student, Amy enters their information for the first time. She first enters Karen's personal information, then Paul's personal information, and then the tutoring schedule for Paul. She sees that Karen wants Paul to be tutored MWF from 3:30-4:30 pm in Math and Chemistry and notes that in the special notes section. Amy then informs Karen of the cost of these sessions.

Scenario 2: Student reschedules a tutoring session.
Billy calls the *Berkeley Academics* office and talks to Jane, one of the staff, and tells her that he cannot make his regular Wednesday 3:30 pm tutoring session because of an orthodontist appointment. Billy asks if he can reschedule the session to Thursday at 4:30 pm. Jane checks the schedule and finds out that Thursday is still available. She then uses the system to reschedule Bill to Thursday at 4:30 pm. The system cancels the Wednesday session and adds a Thursday one for Bill, and the changes are instantly reflected on the updated schedule page.

Scenario 3: Parent comes in to see if they owe money.
Michael comes into the office to drop off his daughter Sarah for her tutoring session. He realizes that he has called the office many times to reschedule, cancel, and add sessions that he doesn't remember if he is up to date on his payments. Michael asks Amy, one of the staff, to check for him. Amy looks up Michael's payment details and sees that Sarah's payments are up to date, but that his son's, Joseph's, payments are not. Michael pays his outstanding payments and Amy enters the payment into the system.

Scenario 4: Admin checks status of payments.
It is close to the end of the second week after a session has begun. Amy realizes that it is time to check the status of payments to see if everything is going well. Amy logs into the system and navigates to the payment section. From there, she chooses to display the payment overview of all the customers for the current term. She browses through all the records and notices that two customers haven’t made their payments, that is, they still have an outstanding balance. She clicks on a customer’s name to view their contact information and then she picks up the phone to politely remind them of their payment schedule.

Scenario 5: Tutor prepares to meet a student and finishes session.
It is Tuesday evening, and Stanley is preparing for his math session tomorrow. The first thing he does is to log into the system and look up the student’s progress to formulate a lesson plan. He clicks on the date of the last lesson to check the progress notes left by the tutor and uses that information for his lesson.

After the session on Wednesday, Stanley logs into the system again and enters his progress notes for the tutor who co-teaches the same session on Friday.

Comparative Analysis

1) WebCalendar

http://www.k5n.us/webcalendar.php

WebCalendar is a PHP-based calendar application. It can work with MySQL, PostgreSQL, Oracle, DB2, or ODBC. *WebCalendar* can be configured to accommodate the following usage scenarios:

- A calendar system for a single user.
- A calendar system used by multiple users, managed by one or more administrators.
- An event scheduler that visitors can view and post new events.
- A calendar server viewable with iCal-compliant calendar applications (e.g. Mozilla Sunbird, Apple iCal) or RSS-enabled applications such as Firefox and Thunderbird.

Pros:

- The overall design is intuitive and easy to use.
- It is very easy to switch between monthly, weekly, and daily view with a single click.
- It has convenient mouse-over description of event.
- Users are able to set an event to be repeated daily, weekly, monthly, or yearly.

Cons:

- There is no legend on the interface to explain what different icons and colors mean.
- On the event detail page, information is too crowed, and the “Edit”, “ Delete”, and “Copy” links are not obvious.
- It doesn’t have a yearly view.

Figure 1 *WebCalendar* **[See Appendix C for additional screenshots]**

2) My Databook

http://www.theadminshop.com/mdb2/

My Databook is an online personal organizer and planner that incorporates a calendar, a planner, an address book, a notepad, a task recorder, a reminder system, and a journal. It is based on PHP 5.0 and MySQL 4.0. *My Databook* uses a horizontal menu bar that can be toggled to display each of the functions mentioned above. The overall layout is clean and tidy, which makes it easy to learn and use.

Pros:

- On the Home page, there are friendly highlights of upcoming appointments, tasks, and last added notes.
- Clear navigation that allows easy access to any section with a single click.
- Appointments and notes on the calendar are color-coded to make them easy to distinguish.

Cons:

- There is no weekly or yearly view. Events have to viewed either in monthly or daily view.
- When adding an appointment, there is only starting time but no ending time.
- Some inconsistent wording used in the links and the linked pages.
- Reminders are not set in the Planner but stand-alone in a different section, causing some inconvenience.
- It allows weekly, monthly, and yearly recurring of an event, but not daily.

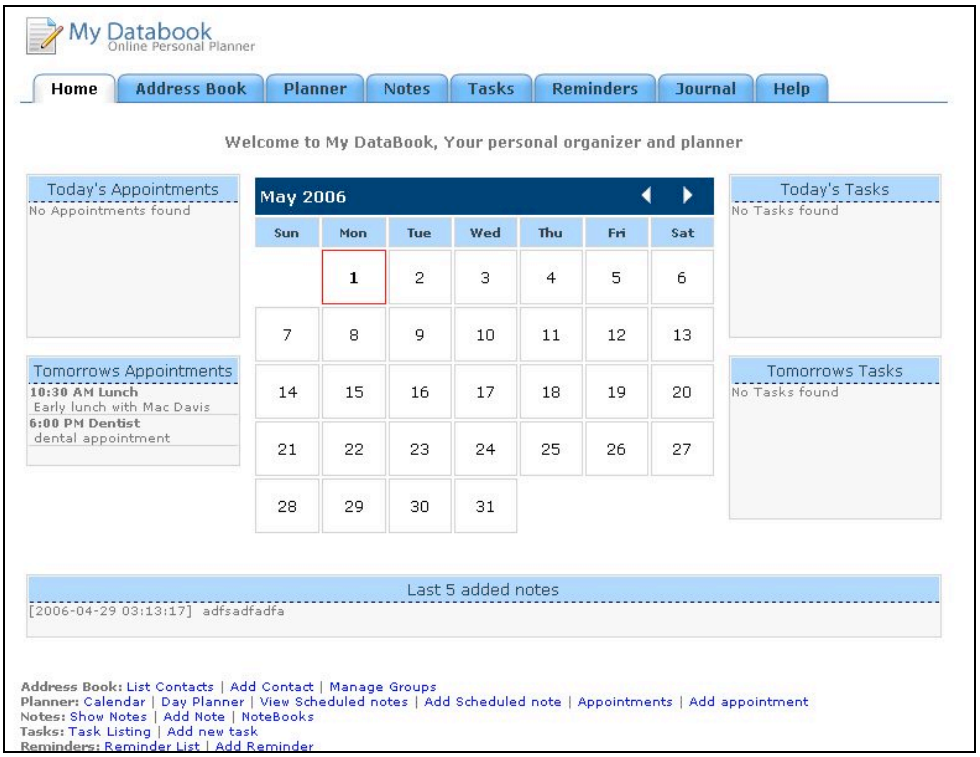


Figure 2 *My Databook* **[See Appendix C for additional screenshots]**

3) Comdev Event Calendar

http://www.comdevweb.com/calendemo.php

Comdev is a paid event calendar that is comprised of a front-end calendar view and a back-end administrative control panel. The commercial software includes comprehensive functionalities, which are not available in freeware. However, it does not support multi-user administration.

Pros:

- It is able to set beginning hour for Daily view.
- It is easy to set year, month, meek, or day view to be the default view.
- It provides a summary of events for each calendar cell.
- It is able to notify a friend about an event and set a reminder for a user.
- It enables HTML editor for entering event description.
- The calendar offers friendly highlights of current date and hour.

Cons:

- The system lacks of navigation links to go backward from the current page.
- Hour interval in day view is one hour, which is not very flexible for scheduling events starting from, say, 1:15 pm.
- In the back-end administration site, there is no page title for each section.
- It does not allow search on multiple criteria (e.g. Date, Title, Type) for an event.

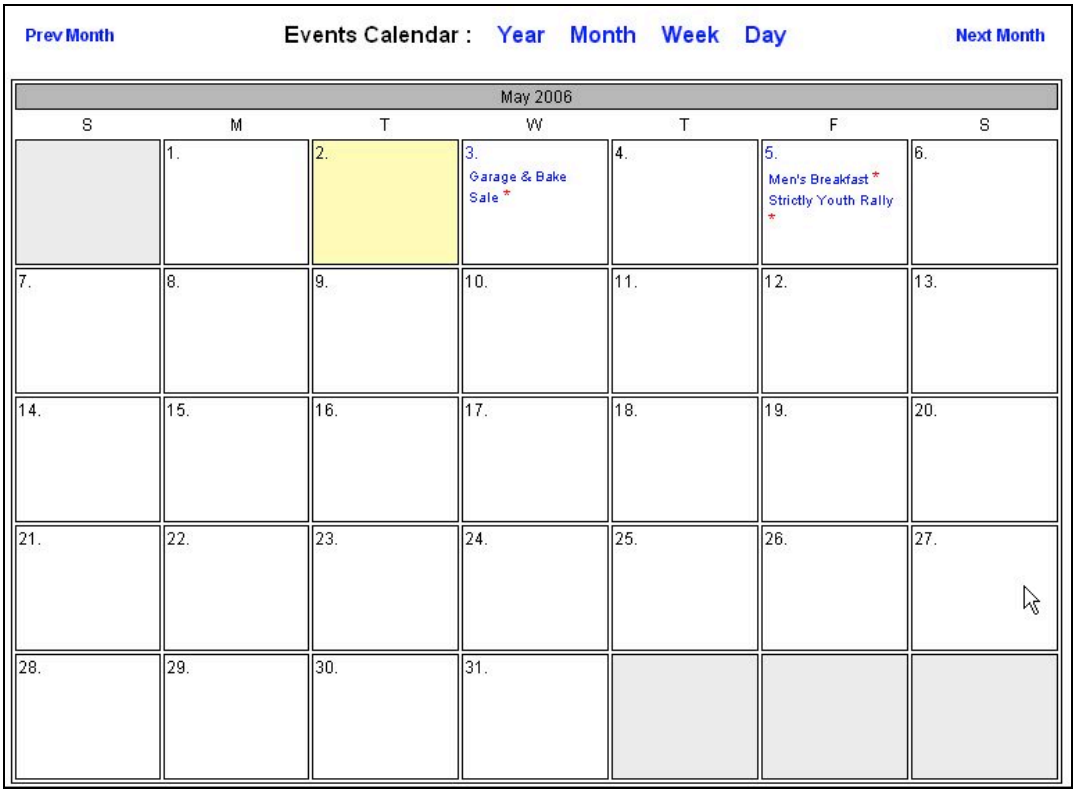


Figure 3 *Comdev Event Calendar* **[See Appendix C for additional screenshots]**

Solution Overview

The goal of this project was to develop a web-based application that would address the problem areas of *Berkeley Academics'* information system. A web-based information system would be more robust, communicative, collaborative, and efficient than a paper-based one and would improve the efficiency of *Berkeley Academics'* administrative work. This system would centralize the information and make it readily available and accessible to all of the interested parties. Tutor and student scheduling information, student academic progress tracking, and payment information would all be incorporated into this web-based application.

In order to achieve this goal of the project, many tools and technologies were used during development and testing:

PHP/MySQL – We decided to use PHP web scripting language, which talked to a MySQL database, to form the foundation of this project. This was partly due to the fact that *Berkeley Academics* already had systems using this architecture and it would be easier, for development, integration, and future support, to continue on this platform. Additional benefits of using PHP and MySQL are that they are free (open source), relatively fast, and easy to develop for (many resources available on the web).

Eclipse/CVS/PHPEclipse Plugin – The Eclipse[2] IDE and a PHPEclipse[3] plugin were used for development of this project. Eclipse is an open source IDE that facilitated versioning control through CVS. When combined with another open source plugin, PHPEclipse, an easy to use development environment was created for the project.

CSS/Javascript – CSS style sheets were used to maintain the "look-and-feel" of the project. Javascript was also used to increase the interactivity of the user interface and add some dynamic elements to the web pages.

PHP Objects – PHP objects were used throughout the project to represents entities in our database and to reduce the amount of duplicative code in our project. PHP objects introduced a layer of abstraction between the database and user interface that reduced the complexity of our PHP pages and made development more efficient and easy to do.

[2] http://www.eclipse.org
[3] http://www.phpeclipse.de/tiki-view_articles.php

Database Design/Integration

An important decision concerning the database design portion of the project was made early on during the project timeline. There was a choice of creating a completely new schema for our project or to expand an existing one *Berkeley Academics* was already using. While both options had their merits, it was decided that the database schema for our project would build upon the existing database and it was with this foundation that many of our application design decisions were made.

The reason why we decided to expand the existing schema, as opposed to creating a brand new one, was because *Berkeley Academics* has existing information systems that needed to be maintained as well as to be integrated with our project. *Berkeley Academics* has a web-based sign-up system which allows students to enroll in the variety of classes and services offered. This system holds all of the enrollment information from previous sign-ups and thus it was important to keep it accessible and usable. In addition, information from this sign-up system needed to be fed into our project as sign-ups were processed. Designing a completely new database would have required a complete overhaul of the existing sign-up page and a data migration. This would have been well beyond the scope of this project and placed an undue amount of burden on the resources allocated for this project. It was deemed appropriate to expand the existing *Berkeley Academics* database scheme to facilitate the integration of the two systems.

The design of the schema expansion was focused on functionality and ease of use. The additional tables in our schema needed to accommodate the core feature sets of our project. We added many new entities and join tables to facilitate this goal. In addition, we wanted our database to be easy to use and to limit the number of joins necessary for the application. Thus the tables in the database were not normalized in any stringent manner. We leveraged other means (i.e. PHP objects) in order to maintain consistency in our database and to add a layer of abstraction between our application and the database.

Schema Description

Original Schema

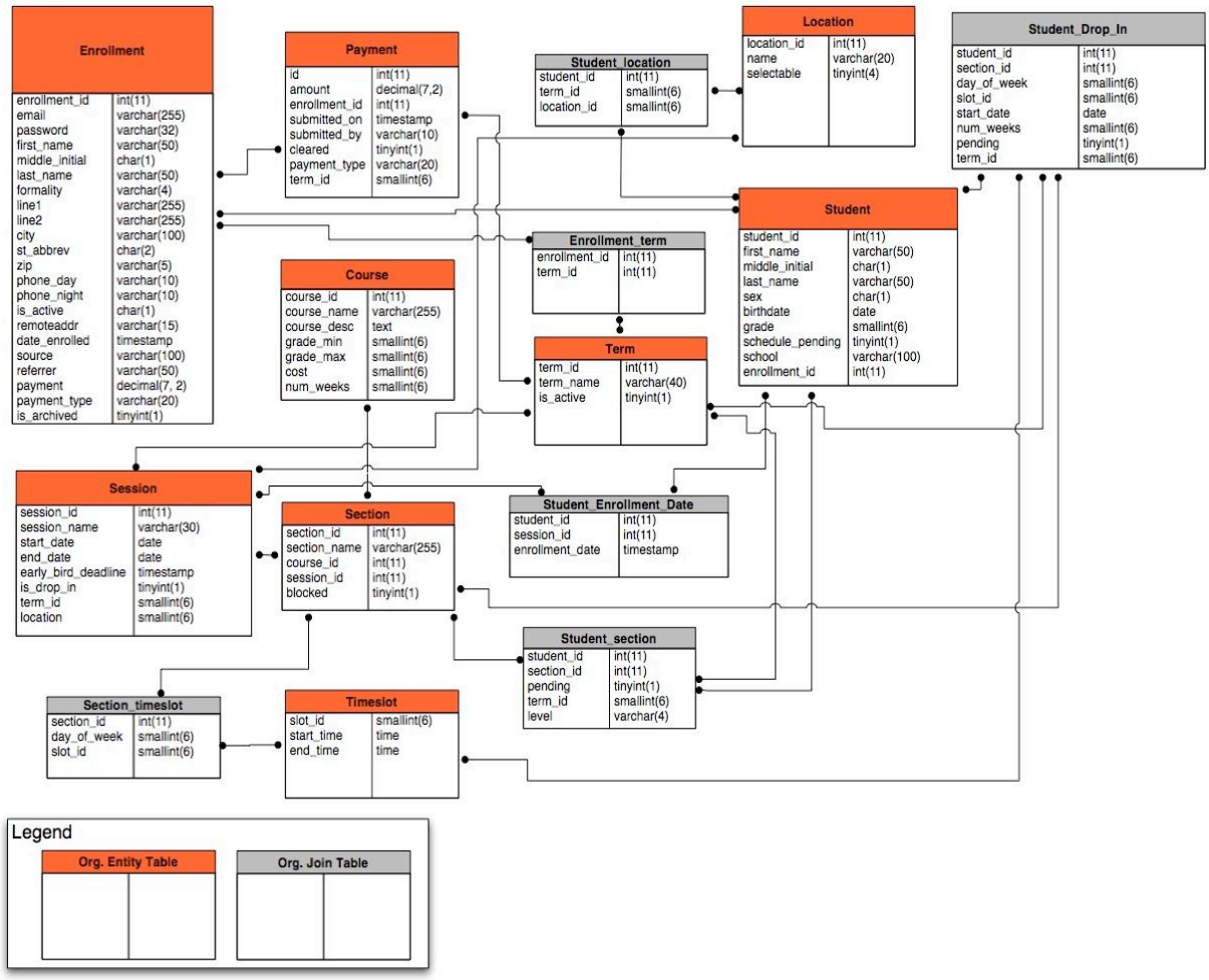


Figure 4 *Berkeley Academics'* original database schema.

Entity Descriptions:

- **Enrollment** – An enrollment represents parent information: login, password, contact information, etc. Each enrollment (parent) can have multiple children.
- **Student** – A student represents student information: name, age, grade, etc. Every student has a parent (enrollment).
- **Payment** – Contains payment information: when and who paid what amount.
- **Term** – A term represents a period of time that *Berkeley Academics* uses to structure its class offerings. (i.e. Fall term, Spring term, and Summer term). A term can have multiple sessions.
- **Session** – Each term is split up into separate sessions for billing and organizing purposes. A session can have multiple sections. A session can have sections with regular schedules or "drop-in" schedules.
- **Course** – Courses contain information and descriptions of the classes that *Berkeley Academics* offer.
- **Section** – A section is a specific instance of a course offered during a session during a particular timeslot. Each section can only have one course. A section can have a regular schedule or a "drop-in" schedule.
- **Timeslot** – The valid time periods for class offerings
- **Location** – Represents the various locations *Berkeley Academics* offers its classes. A student can only be in one location for a particular term

Join Table Descriptions:

- **Student_Location** – Associates students who have signed -p for classes to one of *Berkeley Academics'* locations.
- **Student_Drop_In** – Associates students to a "drop-in" course (i.e. a course that does not have a regular schedule) they have signed-up for. They sign up starting a certain date for a specific amount of weeks, choosing their own timeslots and day of week they would like to come in.
- **Enrollment_Term**– Associates parents (enrollments) to the terms that they have signed-up their children (students) for.
- **Student_Enrollment_Date** – Associates students to the sessions they have been signed-up for and when they enrolled. (The sign-up date is used for billing purposes and calculating discounts)
- **Student_Section** – Associates students to the particular course sections they have signed-up for.
- **Section_Timeslot** – Associates course sections to the timeslots that they are scheduled for.

Expanded Schema

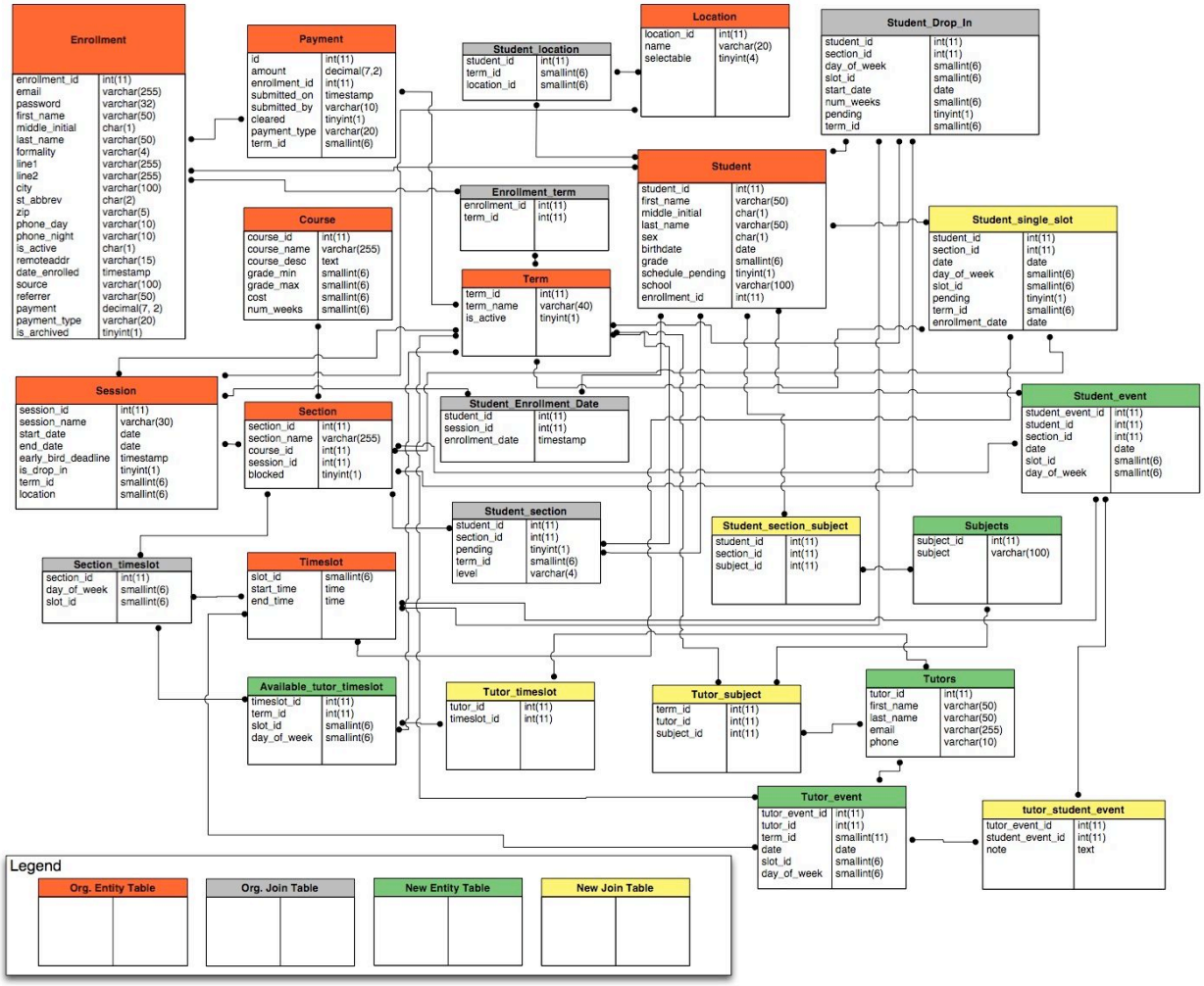


Figure 5 Expanded database schema for project.

New Entity Descriptions

- **Subjects** – The subjects that *Berkeley Academics* offers tutoring services for. Students can choose subjects to be tutored in and tutors can have specialty subjects that they can tutor for.
- **Student Event** – A student event represents the particular timeslot on a particular date for a section that a student has signed up for.
- **Tutors** – Contains information about the tutors (i.e. contact information)
- **Tutor Event** – A tutor event represents the particular timeslot on a particular date that a tutor is available to teach.
- **Available Tutor Timeslots** – This represents all the available timeslots a tutor can teach during a particular term.

New Join Table Descriptions

- **Student_Single_Slot -** Associates students who have signed up for one-on-one tutoring session to its section representation.
- **Student_Section_Subject** – Associates a student to the subjects they would like to be taught during a particular section. Students can be tutored in different subjects during different sections.
- **Tutor_Timeslot** – Associates a tutor with the timeslots they are available to tutor.
- **Tutor_Subject** – Associates tutors to the subjects they can teach.
- **Tutor_Student_Event** – Associates a student event to a tutor event. This association is made when a student is assigned to a tutor. This table also contains notes about the session that a tutor had with a student to track a student progress. Each tutor and student event association can have a single note associated with it.

UI Design Evolution

Our user interface evolved through five major stages: initial sketches, paper prototype, low-fidelity prototype, high-fidelity interactive prototype, and final prototype, each was designed based on tests and interviews of actual or potential system users at *Berkeley Academics*.

1) Initial Sketches

The initial sketches were developed based on the first round of interviews with the staff at *Berkeley Academics*. The purpose of these sketches was to capture all of the important functionalities found during the user requirement interviews and to make sure they were reflected in the design. The sketches were general enough so that any detailed design could be modified or added in later phases. We focused on the following pages and functions:

- Side navigation menu that shows all the different sections in the system and a list of all unassigned student names.
- *Schedule Overview* page that displays tutors and assigned students on a calendar, either weekly schedule or daily schedule.
- *Assign Student* page that schedules a student to a tutor on a particular date and timeslot.
- *Track A Student* page that adds a note to a student for a particular tutoring session.
- *Payment Summary* page that displays payment summary and allows the user to click for detailed payment information.
- *Tutor Schedule* page that shows weekly schedule of all tutors.

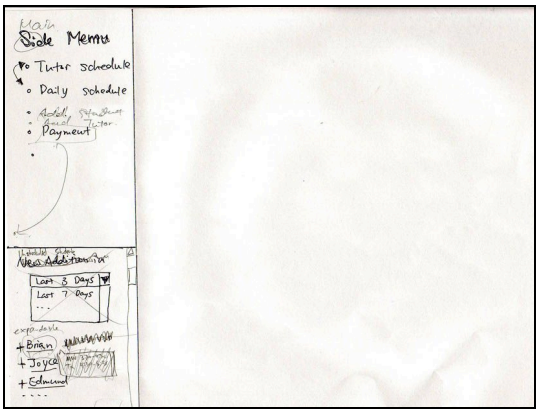


Figure 6 *Left Menu*

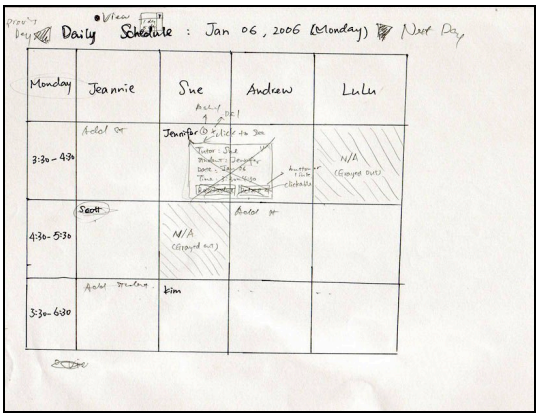
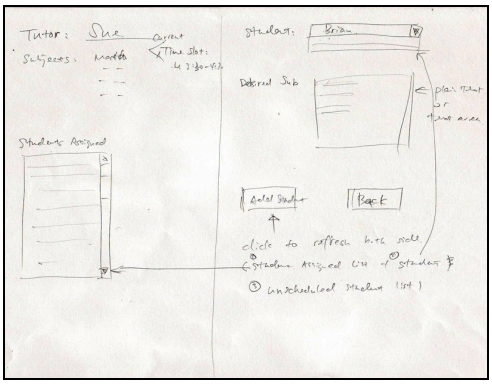


Figure 7 *Calendar Overview*: Daily View



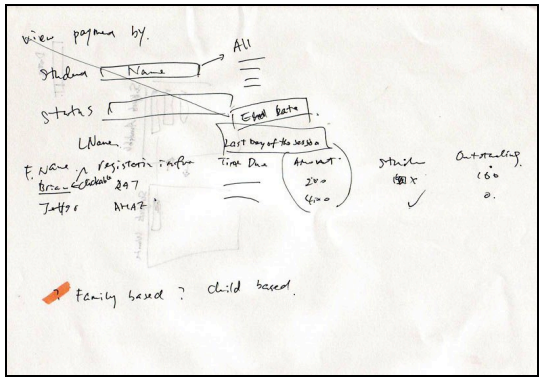


Figure 8 *Add A Student* **Figure 9** *Payment Summary*

[See Appendix D for additional screenshots]

2) Paper Prototype

After laying out the initial sketches, we conducted comparative studies of the websites or online systems that had similar functionalities and feature sets such as scheduling, payment management, and personnel record tracking. After reviewing and evaluating several systems, we came up with a paper prototype that was more detailed than the initial sketches and could allow a certain level of user interaction by utilizing removable tags and Post-it notes to simulate the system's reaction upon a task performed by a user. Compared with the initial sketches, the major changes we made included:

- In *Schedule Overview* page, we added Reserve-a-Tutor (RAT) and Assisted Homework/Academic Enrichment (AHAE) filters to let users choose what to display on the calendar. (RAT and AHAE are the two main "courses" offered by *Berkeley Academics)* We also added Reschedule Student, Add Note, and Delete Schedule links beside each student name, allowing users to perform these tasks.
- In the *Track A Student* page, we used two drop down lists to ask users to choose which student's notes to display and how many of them to display.
- In *Reschedule A Student* page, we decided to use drop-down lists to display all available tutors for a given time lot and the tutor subjects would be generated automatically according to which tutor is selected. We also added a Once/Permanent change selector to indicate if the rescheduling is a one-time change or a weekly one.

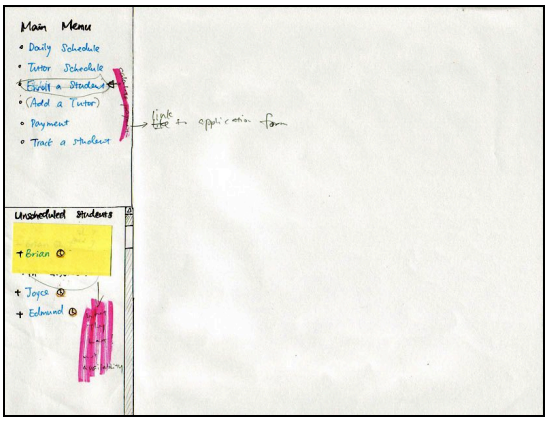


Figure 10 *Left Menu*

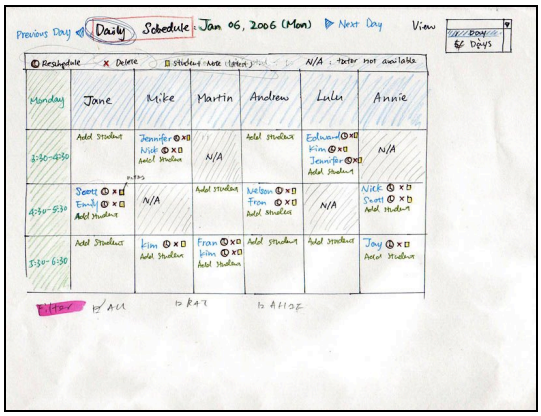


Figure 11 *Calendar Overview*: Daily View

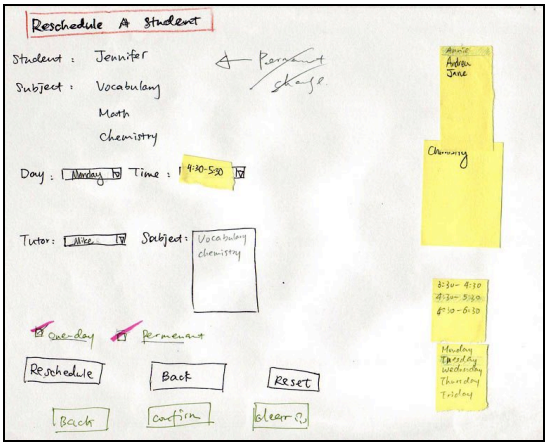


Figure 12 *Reschedule A Student*

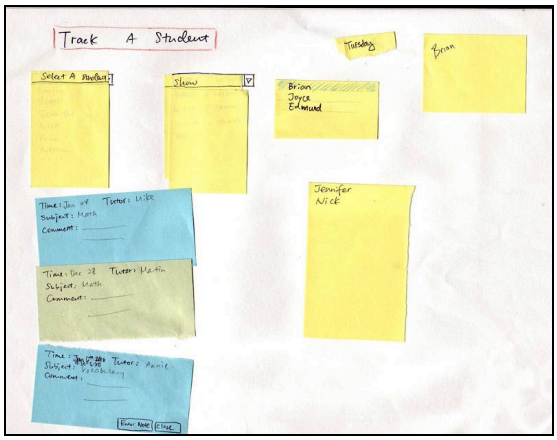


Figure 13: *Track A Student*

[See Appendix D for additional screenshots]

3) Low-fidelity Prototype

After we conducted the first round of user testing, we found that users were generally satisfied with the features the paper prototype incorporated. However, some detailed design changes would be considered and incorporated into our low-fidelity prototype. Moreover, some of the interactions that could not be simulated in the paper prototype would be added to our web-based low-fidelity design. Hence, in the low-fidelity prototype, we improved our design in the following areas:

- In *Track A Student* page, we added a selector for Term so that users can limit the students that could be selected to a particular term.
- We added an expandable list of unscheduled students. When clicked on, detailed student schedule availability was expanded under the student name.

- We re-structured the *Payment Summary* page, which now only displayed “Enrollment Name”, “Student Name”, “Total Charges”, “Total Payments”, and “Balance” fields.
- We added more pages that were not available in the paper prototype such as *Edit Student Note*, *Edit Payment*, *Add Payment*, etc.

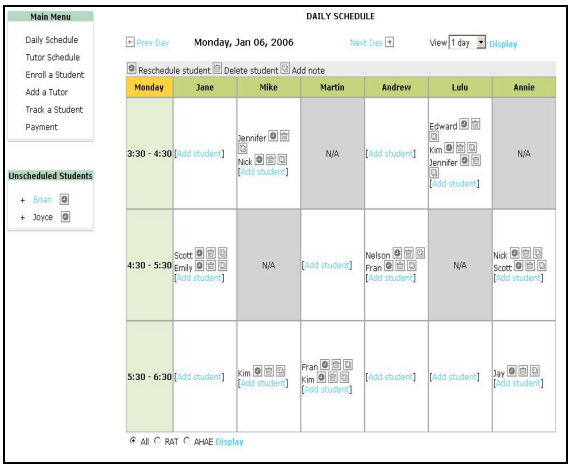


Figure 14 *Calendar Overview*: Daily View

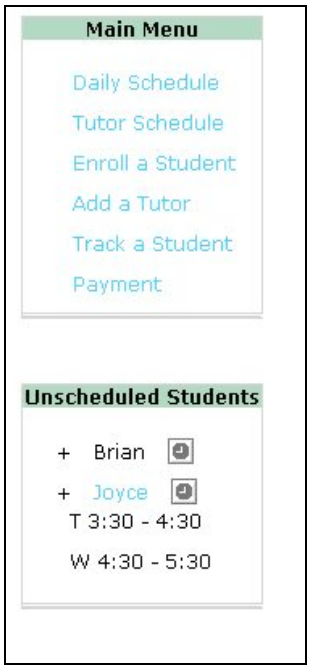


Figure 15 *Left Menu* expanded

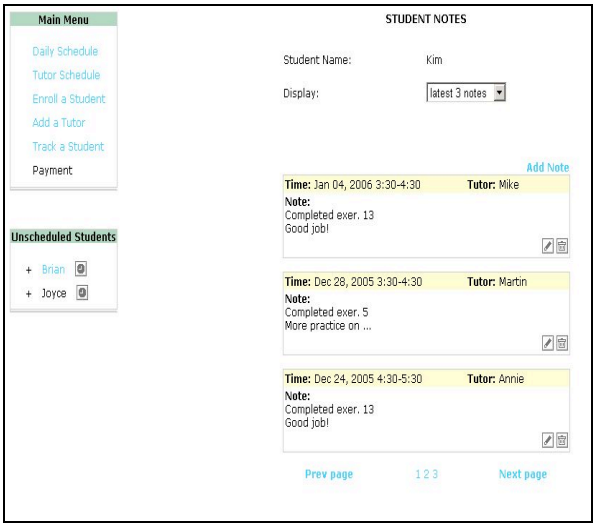


Figure 16 *Track A Student*

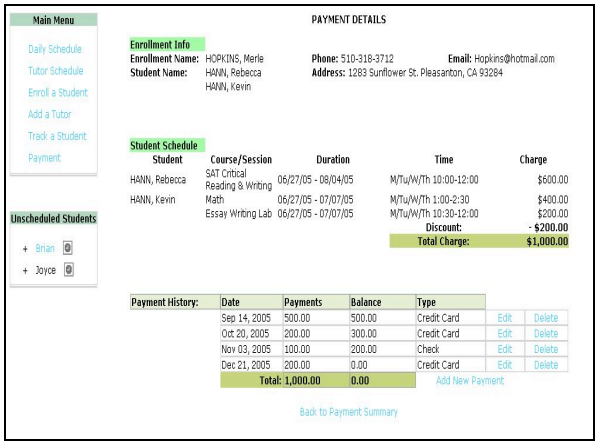


Figure 17 *Payment Detail*

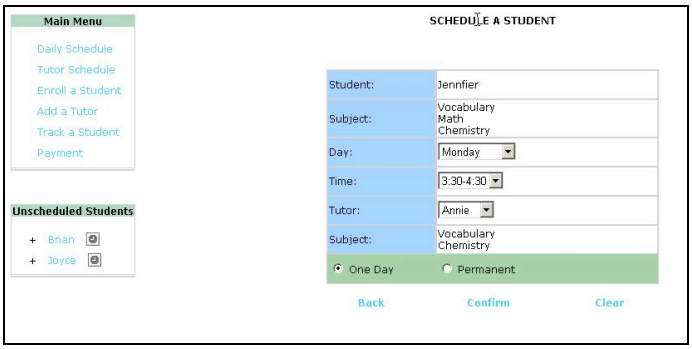


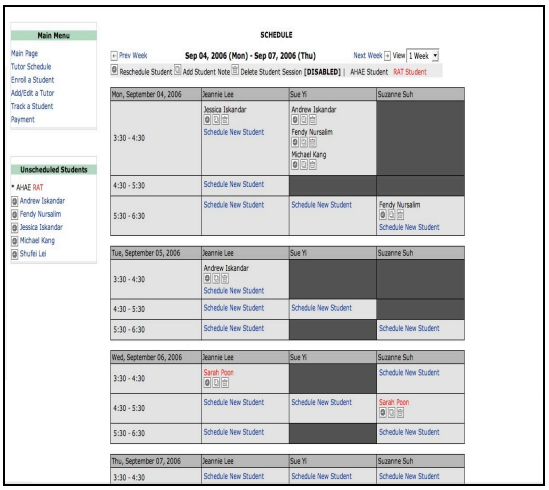
Figure 18 *Reschedule A Student*

[See Appendix D for additional screenshots]

4) High-fidelity Prototype

From the feedback of the second round user testing, we found users were a little bit confused when performing the task to add a student, mainly because they were not sure what the information meant in the Unscheduled Student list and what they could do with it. They also expressed some concern over the inconsistent wording across different sections. From this set of user tests, we collected more information on how users actually used the system and what their expectations of it were. We hence modified our design in the following ways:

- In *Schedule Overview* page, we made weekly view the default page because users indicated that weekly view was more useful to them.
- We color-coded RAT and AHAE time slots in the Unscheduled Student list in the left panel, *Schedule Overview* page, and *Add A Student* page to help users distinguish between the two.
- We chose more consistent wording across the whole system.
- A term selector was added in *Payment Summary* page to filter out unnecessary information.
- Colors were used in Balance field of the *Payment Summary* page to indicate if there were any outstanding balances.
- The *Payment Detail* page was re-structured to show four types of information: contact information, payment quick view, payment history, and student schedule information.
- We "hid" the textboxes using style sheets in the *Add A Student* page so that users wouldn't be misled to think they could input something into them.
- Confirmation pages for deletion were replaced by popup alert messages to lessen the deletion steps users had to experience.



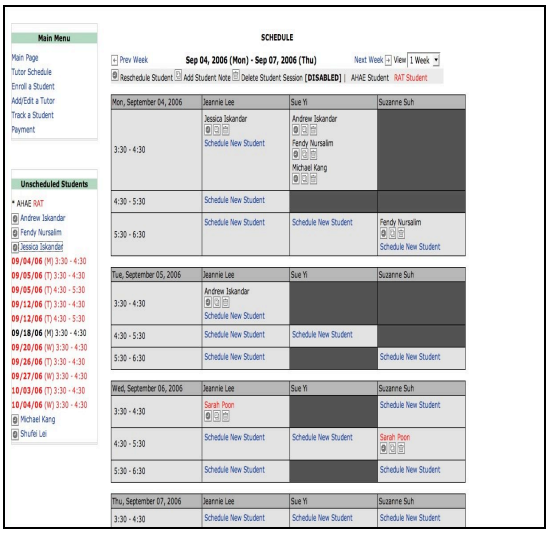
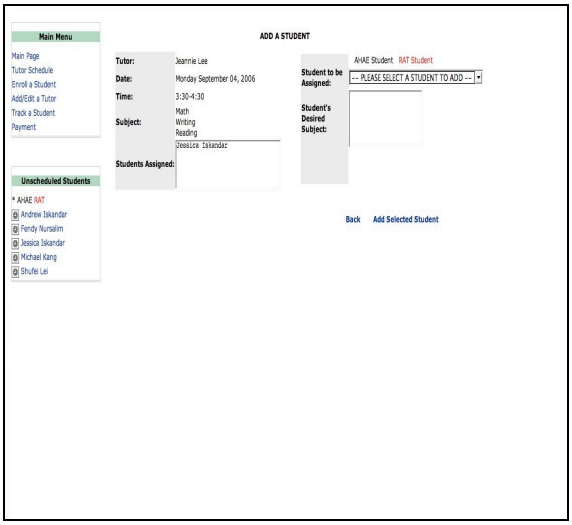


Figure 19 *Calendar Overview*: Weekly View **Figure 20** Left expandable menu color-coded.



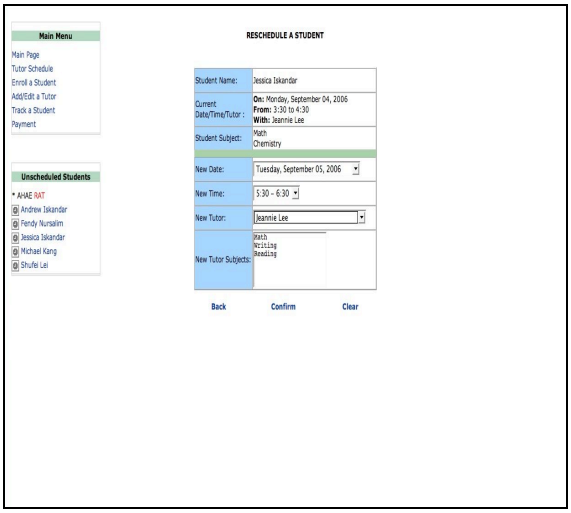
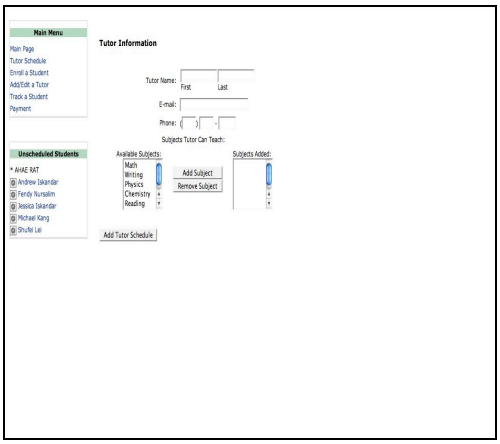


Figure 21 *Add A Student* **Figure 22** *Reschedule A Student*



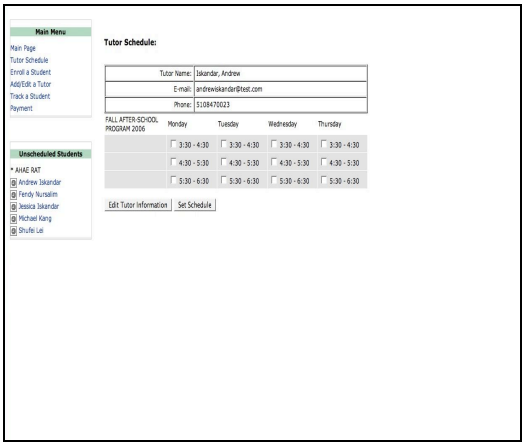


Figure 23 *Add A Tutor* **Figure 24** *Set Tutor Schedule*

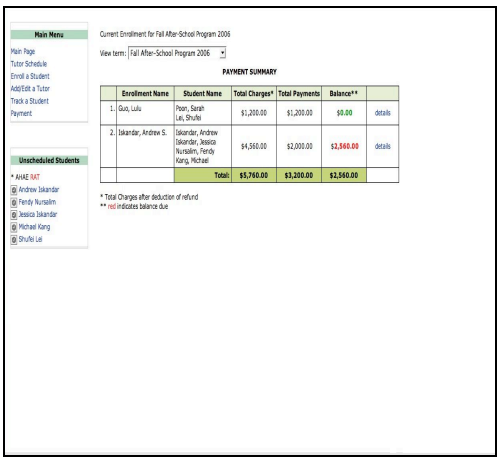


Figure 25 *Payment Summary*

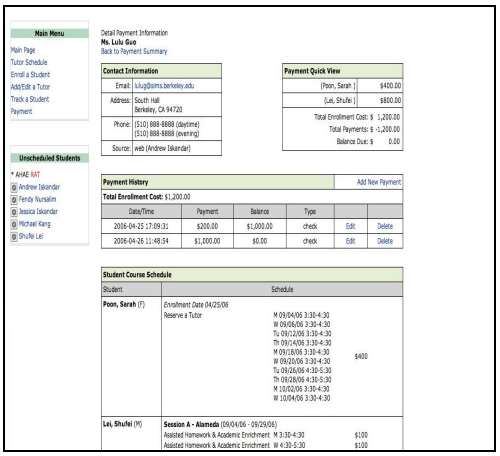


Figure 26 *Payment Detail*

[See Appendix D for additional screenshots]

5) Final Prototype

In the last round of user testing, we asked the users to give us feedback that was as detailed as possible because they were going to be implemented in our final application. The users were happy about the functionalities and appearance of the system overall, but also provided us some valuable task-specific information and detailed look-and-feel suggestions. The suggestions for improvement were reflected in our final application as follows:

- We re-grouped the navigation menu items and put dividers between different groups to make the structure clearer.
- We changed some wording to make them less ambiguous to users.
- In *Add/Edit Tutor* page, we made a separate drop-down list for editing a tutor and single-click interface to add a new tutor rather than combining this functionality.
- We unified page and table styles across the whole system.
- Add a New Student link changed to Assign a Student
- In *Edit Payment* page, the payment type list was modified so that type was automatically selected. User didn't need to go back to previous page to check what the type was.
- The expandable Unassigned Student list was modified to remember its state so that the next time the page was loaded, it "remembered" which part of it was expanded so that users didn't need to click on it again.
- We removed the Delete Student icon from *Schedule Overview* for users found it confusing and would prefer to only have Reschedule Student and Add Student Note on the page.

Implementation (Screenshots)

Schedule Overview

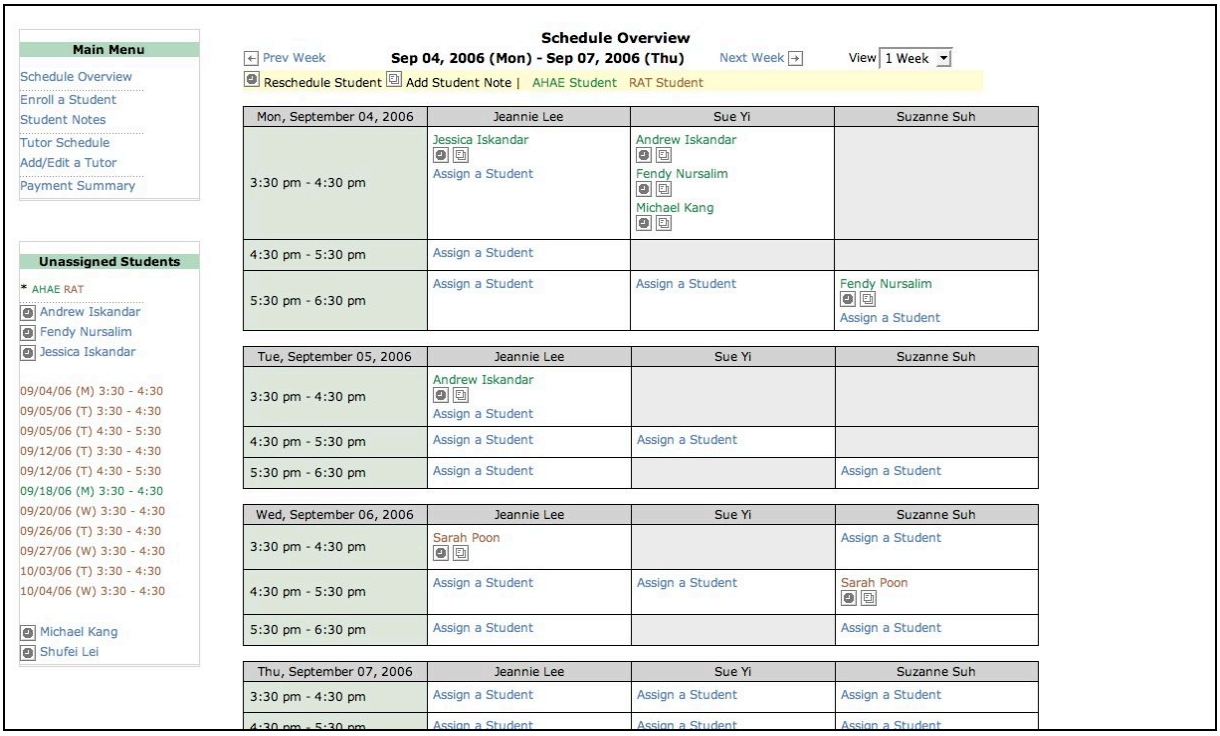


Figure 27 *Schedule Overview* (default weekly view)

Description: This is the opening page of the application. It shows the current week's (*Berkeley Academics'* weeks are from Monday – Thursday from 3:30 pm – 4:30 pm) schedule. This page gives a quick overview of which students are coming for which tutor. The user can always come to this page by clicking on the Schedule Overview link in the left-hand navigation menu.

Features/Workflow: From this page, a user can *assign a student* to a tutor on a particular date. A user can also *reschedule a student* to a new date. A user can also *add a note* for a particular tutor-student pair on a given date. A user can view the schedule by **week** or by **day** and traverse the calendar by week or day as well.

Berkeley Academics has two main "course" divisions: Assisted Homework/Academic Enrichment (AHAE) and Reserve-a-Tutor (RAT). There are different business requirements for these two types of courses and students who have signed up for these courses are color-coded accordingly.

The left hand navigation menu (visible on all pages) allows the user to view all **unassigned students** and toggle the visibility of their unassigned dates and timeslots. A user can also enroll a student, view student notes, view tutor schedule, add/edit tutors, and view payment summary.

Assign a Student

Figure 28 *Assign a Student*

Description: After clicking on one of the "Assign a Student" links on the *Schedule Overview* page for a given date/timeslot, the user will be directed to this page. This is where a user can assign any unassigned students to this tutor at this particular date and time. The application uses business rules to determine which students can be assigned to this particular tutor at this date and time.

Features/Workflow: The left hand column gives information about the current tutor being assigned to, the time and date information, the subjects a tutor can teach, as well as the current students already assigned to this tutor during this date and time.

The right hand column is where the user interacts with the system and chooses a student to be assigned. Only students that are "valid," according to *Berkeley Academics'* business requirements to be assigned to this tutor can be chosen through the drop down. If a student is selected, their subjects appear in interface and the user can visually match if this student/tutor pair is reasonable.

The user then clicks on the Assign Selected Student link, and she will be presented with a confirmation alert screen, and is brought back to the *Schedule Overview* page. If the user clicks on the Back link, she will also be brought to the *Schedule Overview* page.

Reschedule a Student

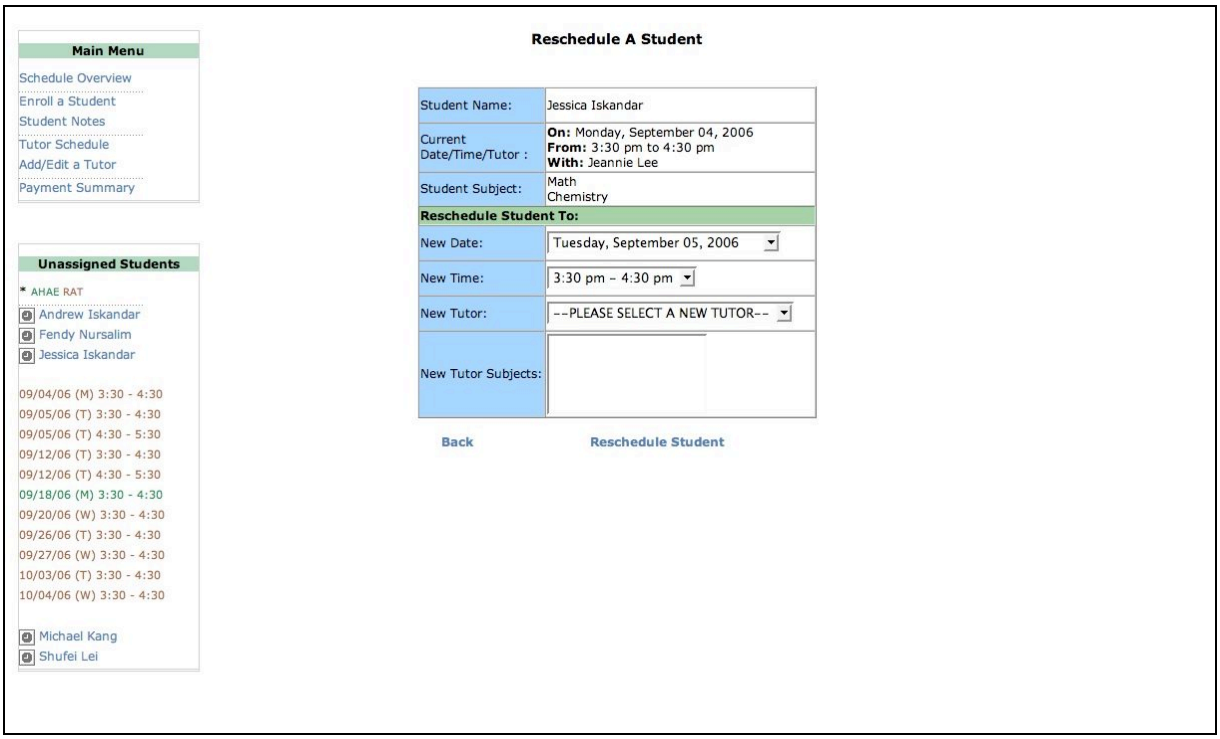


Figure 29 *Reschedule a Student*

Description: After clicking on the Reschedule Student icon link on the *Schedule Overview* page, the user is brought to this page. This is where the user can reschedule a student to a new date, time and/or tutor. It shows the current student, time, date, and tutor information of the student that the user is trying to reschedule.

Features/Workflow: As the user selects new dates and times, the application populates the tutor drop down with tutors that the student can validly be assigned to according to *Berkeley Academics'* business requirements.

When the user selects a tutor, the subjects that the tutor can teach will appear so that the user can appropriately match the student.

The user then clicks on the Reschedule Student link and a confirmation alert will appear to the user. Upon confirming the rescheduling of the student, the user will be brought back to the *Schedule Overview* page and the student will be rescheduled. If you user clicks on the Back link, she will also be brought back to the *Schedule Overview* page but nothing will have changed.

Add Student Note

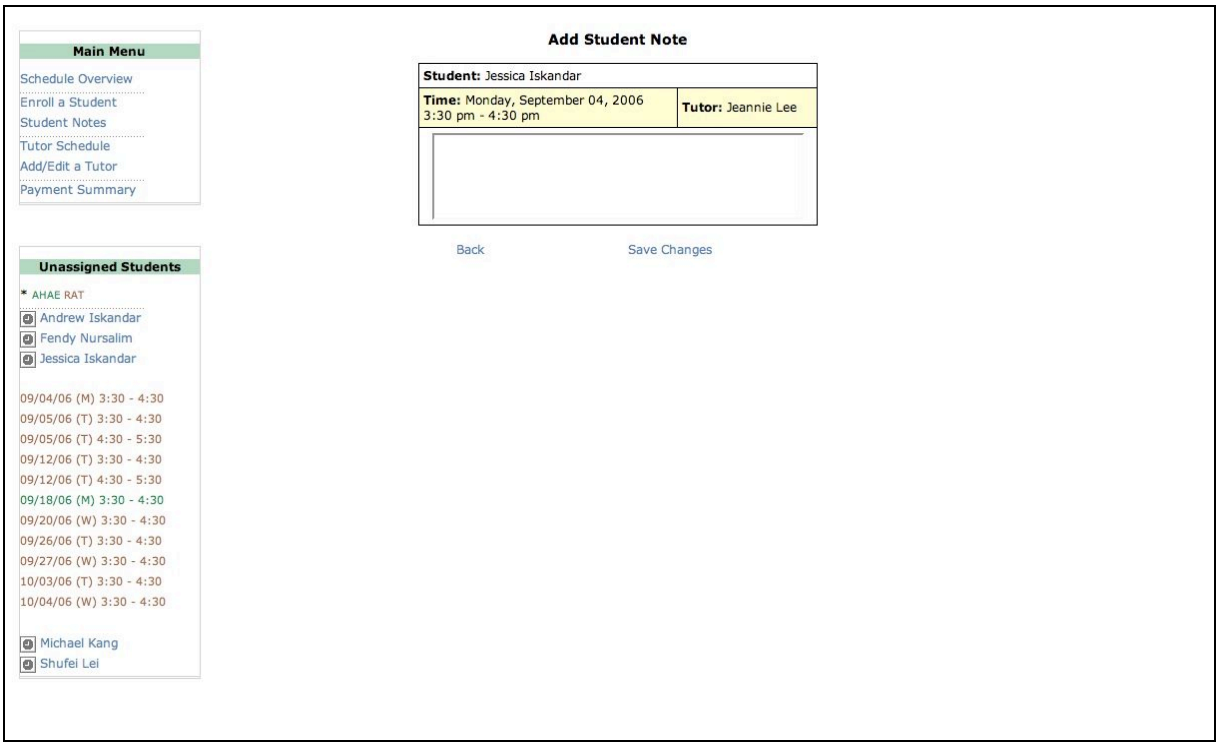


Figure 30 *Add Student Note*

Description: After clicking on the Add Student Note image link, the user will be brought to this page. This is where the user will write notes about a particular tutor-student tutoring session for a given date and time. Typically, the tutor would access this page to track a students' progress, achievements, problem areas, etc.

Features/Workflow: The user can type in a note in the text field and then hit the Save Changes link to add a note to the system. If the user clicks on the Back link, she will be brought back to the *Schedule Overview* page.

Student Notes (Selecting a Student)

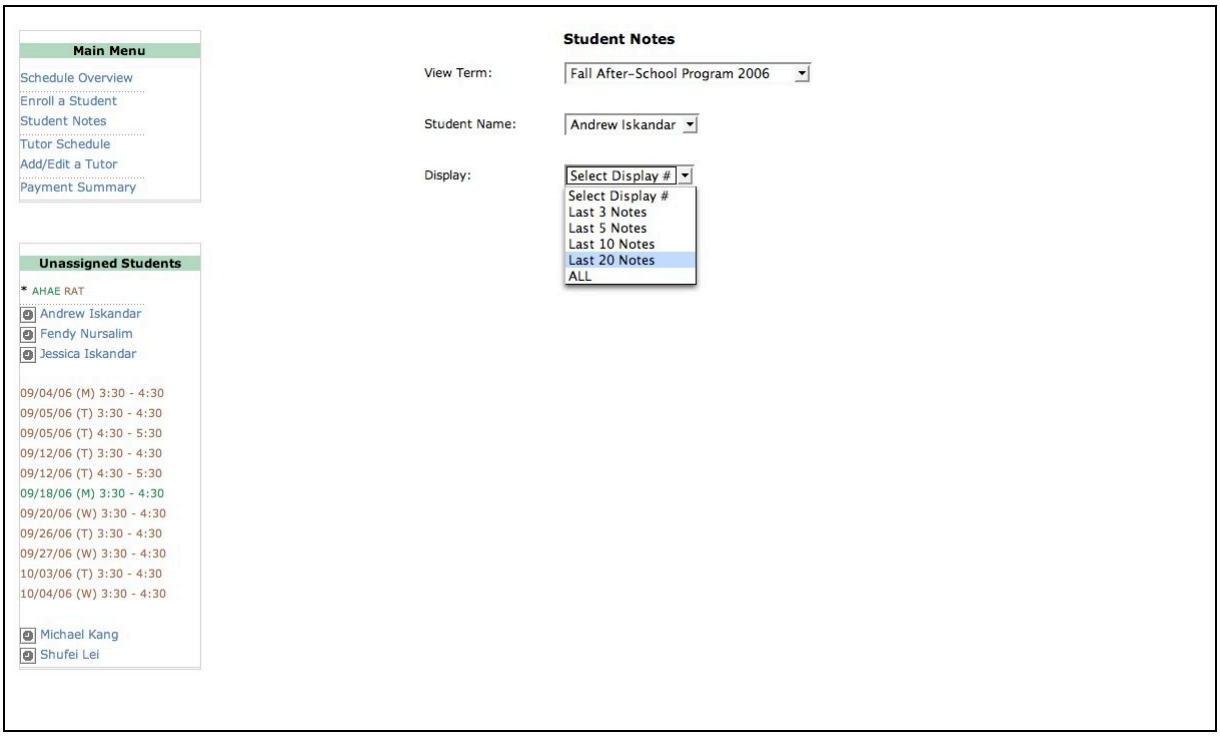


Figure 31 *Student Notes* (select a term, student, and display number)

Description: The user would be brought to this page upon clicking the Student Notes link on the left hand navigation menu. This is where a user can view all the notes for a particular student during a particular term.

Features/Workflow: The user first selects the term they would like to access. The application will then populate a drop down with students signed up for that term. After the user has selected the student she wants to see notes for and the number of notes she would like to see, the application then displays the notes in the *Student Notes (View Student Notes)* page.

Student Notes (View Student Notes)

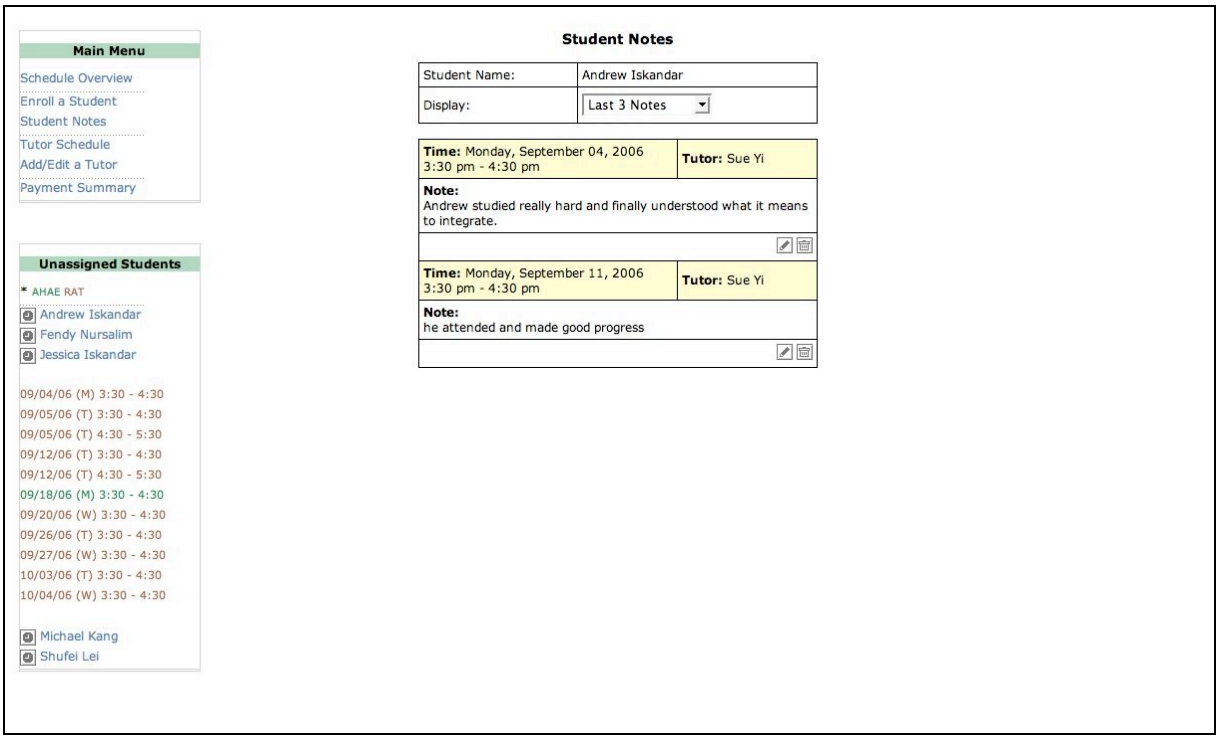


Figure 32 *Student Notes* (viewing notes for a student)

Description: The user is brought to this page upon selecting a term, student, and note display number from the *Student Notes (Selecting a Student)* Page. This page just shows all the notes for the student that the user selected to see.

Features/Workflow: The user can read all of the notes on this page. If she wants to, she can edit or delete a note by clicking on the appropriate icon links. If the user attempts to delete a note, a confirmation alert would appear to verify this decision.

Edit Student Note

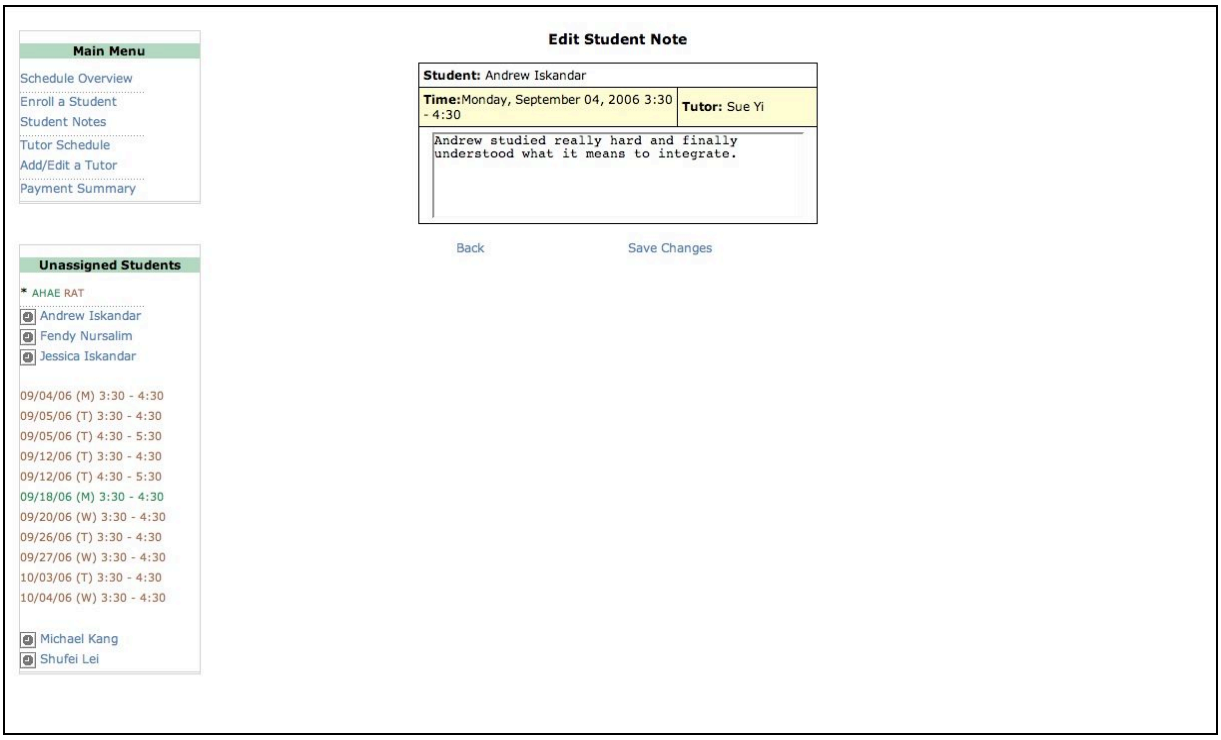


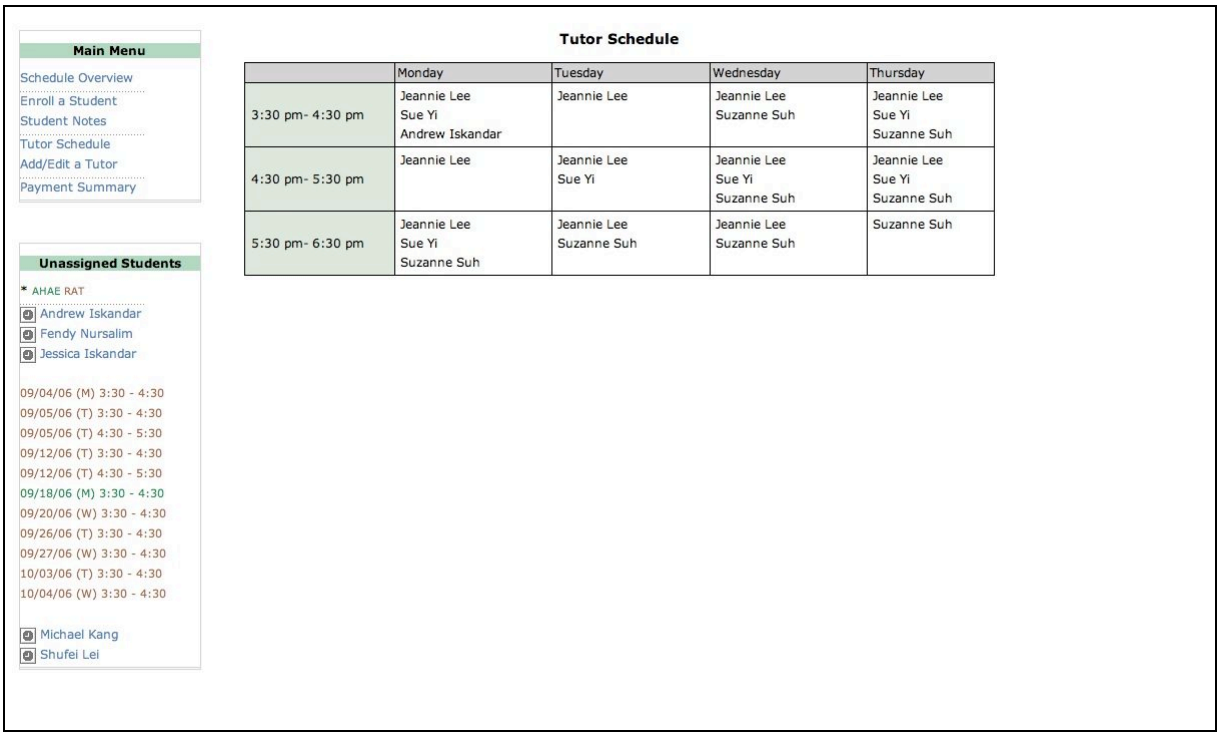
Figure 33 *Edit Student Note*

Description: The user is brought to this page upon clicking on the edit note image link. The user can edit a given note.

Features/Workflow: The user edits the note as they see fit. She can then hit the save changes link to save the edits she has made and she will be brought back to the *Student Notes (View Student Notes)* page.

She can also hit the back link and be brought back to the *Student Notes (View Student Notes)* page. Doing so will not save any of the edits she has made

Tutor Schedule



	Monday	Tuesday	Wednesday	Thursday
3:30 pm- 4:30 pm	Jeannie Lee Sue Yi Andrew Iskandar	Jeannie Lee	Jeannie Lee Suzanne Suh	Jeannie Lee Sue Yi Suzanne Suh
4:30 pm- 5:30 pm	Jeannie Lee	Jeannie Lee Sue Yi	Jeannie Lee Sue Yi Suzanne Suh	Jeannie Lee Sue Yi Suzanne Suh
5:30 pm- 6:30 pm	Jeannie Lee Sue Yi Suzanne Suh	Jeannie Lee Suzanne Suh	Jeannie Lee Suzanne Suh	Suzanne Suh

Figure 34: *Tutor Schedule*

Description: The user is brought to this page by clicking on the Tutor Schedule link in the left hand navigation menu. This page just shows the regular schedule of the tutors, which day and times they are coming (according to the schedule set when the tutor was added or edited.)

Features/Workflow: The user looks at this information to judge if they need more tutors for a particular day or timeslot.

Add/Edit Tutor

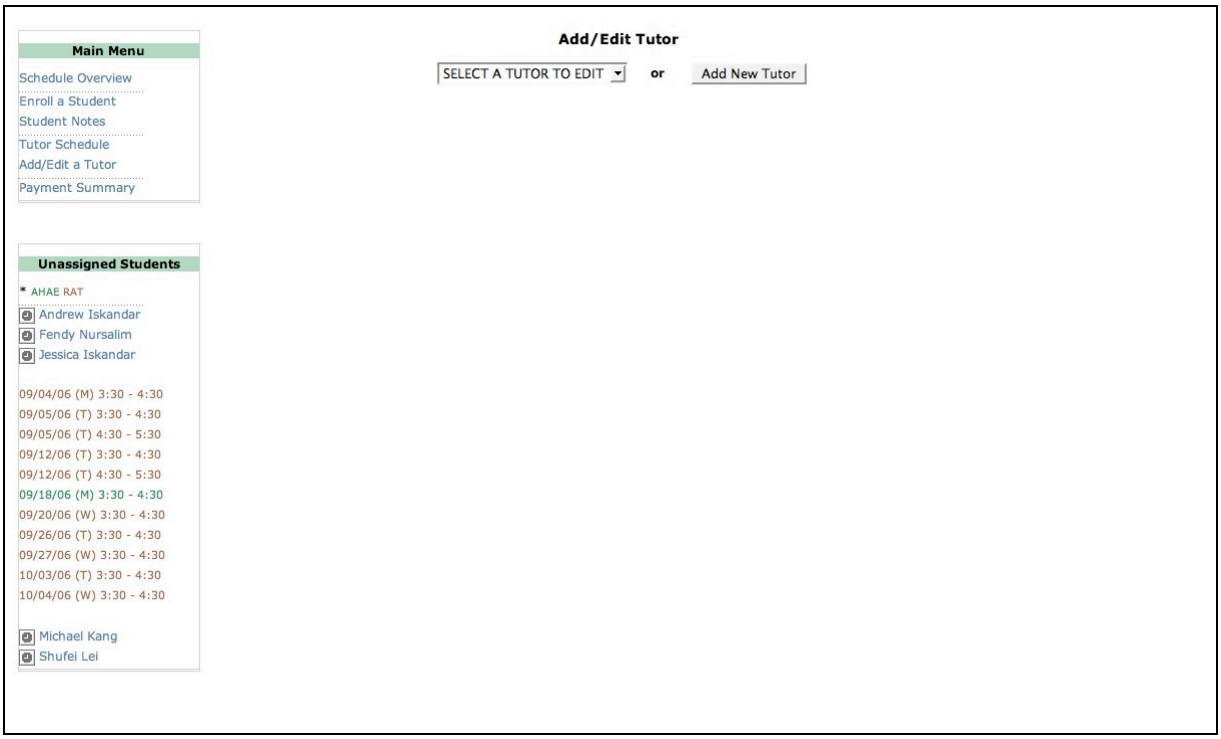


Figure 35 *Add/Edit Tutor*

Description: The user can choose to edit the existing information about a tutor and his/her schedule or add a new tutor to the system.

Features/Workflow: The drop-down menu shows all the tutors that currently exist in the system. Upon selecting a tutor from this menu, the user will be brought to the *Add/Edit (Tutor Information)* page automatically. This page will be populated with the selected tutor's information and begin the process of editing this tutor's information.

If the user hits the add new tutor button, the user will brought the *Add/Edit (Tutor Information)* page as well, but with all the fields blank.

Add/Edit Tutor (Tutor Information)

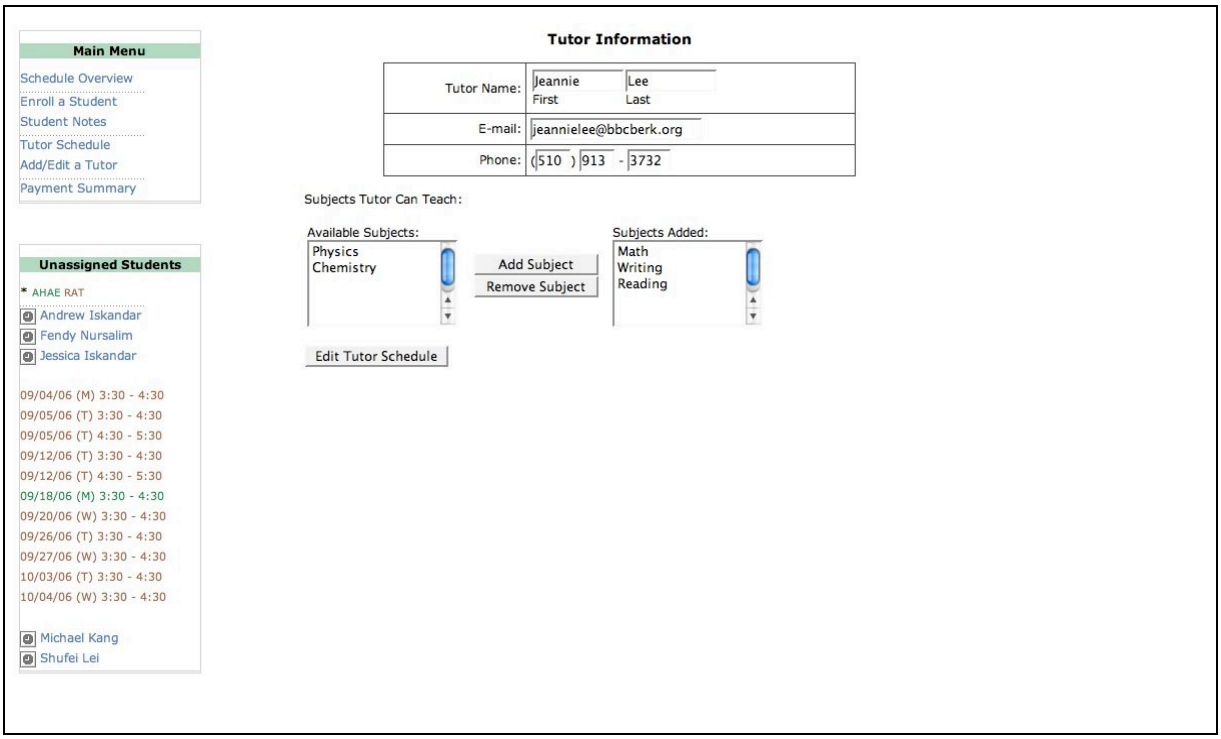


Figure 36 *Add/Edit Tutor* (editing tutor information)

Description: The user is brought to this page after they have either selected a tutor to edit or chosen to add a new tutor from the *Add/Edit Tutor* page. This is where the user enters into the application tutor contact information and subjects they can teach. If the user has chosen to edit a tutor, the tutor's information will be populated in the fields. If the user has chosen to add a new tutor, all the fields will be blank.

Features/Workflow: The user can select multiple subjects from the **available subjects** list to the **subjects added** list.

This page is error checked to make sure that the user inputs all the required information. The user is notified of the missing required fields.

After the user has edited the fields the user will hit the edit tutor schedule button to go to the *Add/Edit Tutor (Tutor Schedule)* page. This button will change to add tutor schedule button when the user is adding a new tutor, but will be brought to the same page.

Add/Edit Tutor (Tutor Schedule)

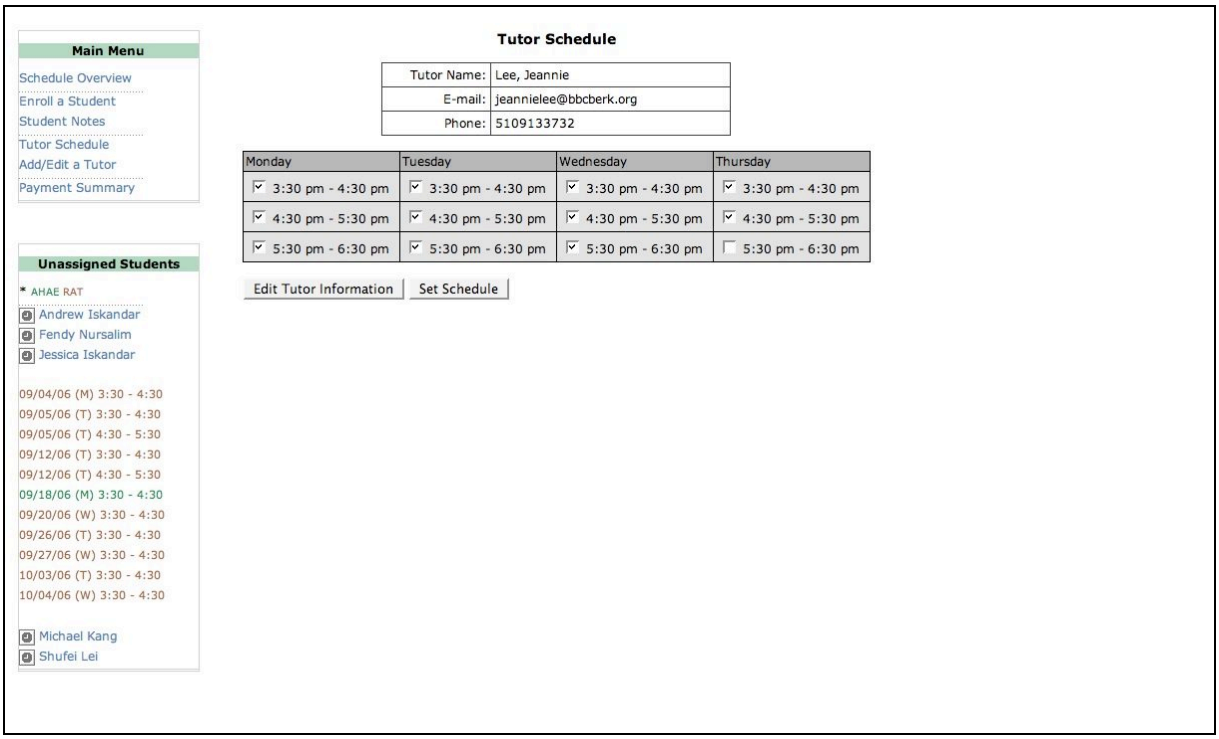


Figure 37 *Add/Edit Tutor* (setting or editing tutor schedule)

Description: The user is brought to this page after entering or editing a tutor's information. This is where the user can edit or add a tutor's schedule. This page shows all the available timeslots a tutor can teach. Tutors are expected to have a regular schedule which is why only one week is shown (*Berkeley Academics* working hours are from 3:30 pm – 6:30 pm Monday to Thursday)

Features/Workflow: This page shows the tutor information that the user has inputted in the previous page.

After selecting/deselecting the timeslots that a tutor is available, the user then hits the set schedule button and is brought to the *Confirm Tutor Schedule* page. If the user hits the edit tutor information button, she will be brought back to the *Add/Edit Tutor (Tutor Information)* page.

Add/Edit Tutor (Confirm Tutor Schedule)

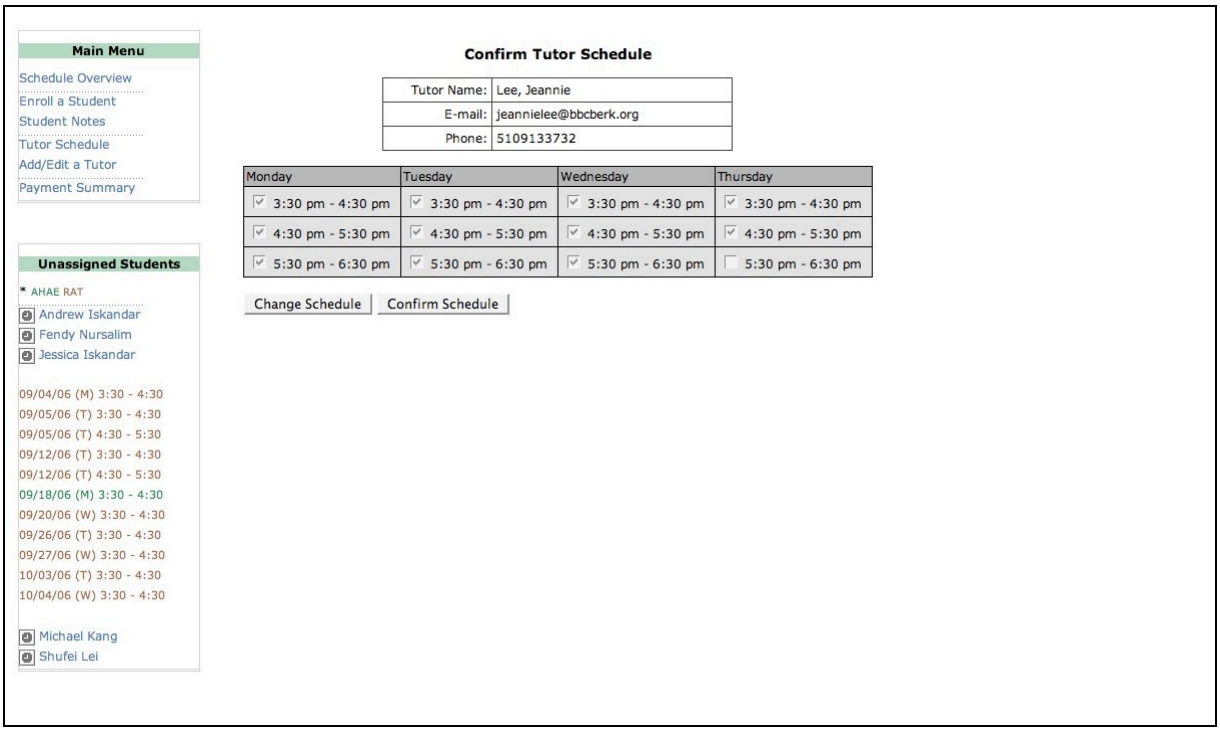


Figure 38 *Add/Edit Tutor* (confirmation screen)

Description: The user is brought to this page after entering a tutor's schedule. This is a confirmation page that the application presents to the user to make sure that the information they've entered for a tutor is correct.

Features/Workflow: The user can hit the change schedule button to go back to the *Add/Edit Tutor (Tutor Schedule)* page.

If the user is okay with the tutor information and schedule, they can hit the confirm schedule button which will bring them back to the *Add/Edit tutor* page.

Payment Summary

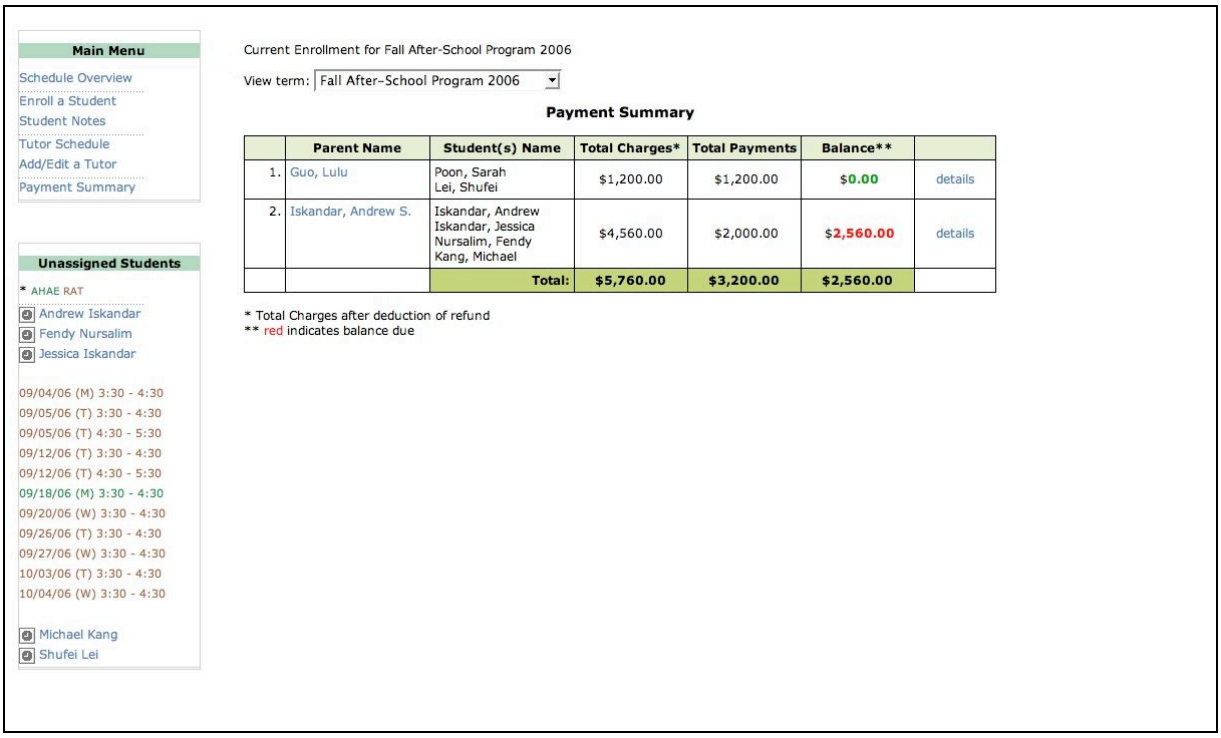


Figure 39 *Payment Summary*

Description: This page shows the financial summary of a given term. It shows all the parents who have enrolled for a given term, their students, their total charges, their total payments, as well as their outstanding balance. It also gives grand totals of changes, payments, and outstanding balances so that the user can get a quick summary of the financial situation. The user is brought to the page when they click on the payment summary link in the left hand navigation menu.

Features/Workflow: The user can view the financial summary of old terms (by selecting an older term from the drop down menu) but by default it goes to the current term.

The balances are color coded to highlight outstanding balances and give the user an indication from whom they need to collect from.

The user can hit the details link in each parent summary row to get more information about that parent and their children. This will bring them to the *Detailed Payment Information* page.

Detailed Payment Information

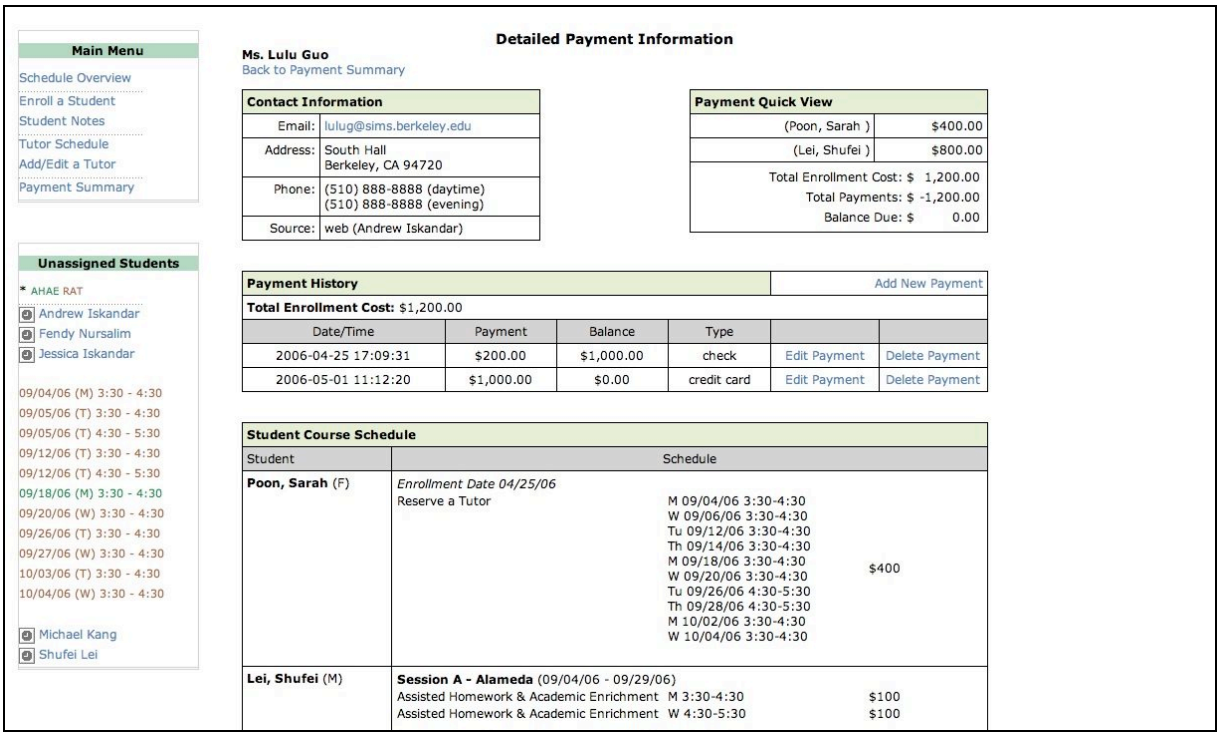


Figure 40 *Detailed Payment Information*

Description: This page shows the user detailed payment, parent, and student information. It shows contact information for the parent, summary information for his/her students (what courses they've signed up for, how much those courses cost, etc.) and also their payment history.

Features/Workflow: This page has a **payment quick view** area to give a quick summary to the user how much the parent owes for each child and their outstanding balances and total payments.

It also shows the user the schedule of the students and the total cost of the courses they've signed up for.

The user can add a new payment by clicking on the add new payment link. This will bring them to the *Add New Payment* page. The user can also edit a payment by clicking on the edit payment link for each transaction row in the payment history. The user can also delete a payment by clicking on the delete payment link. The user will be asked to confirm a deletion.

Edit Payment History

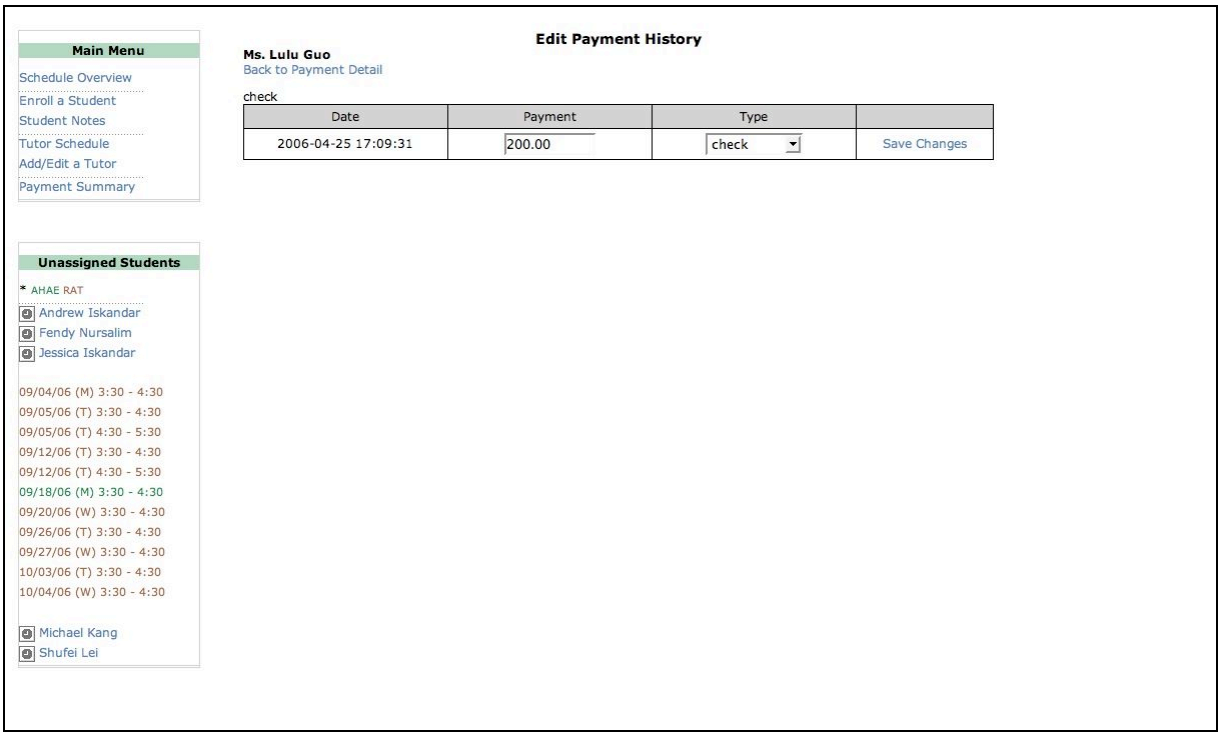


Figure 41 *Edit Payment History*

Description: The user will be brought to this page if she clicks on the edit payment link on the *Detailed Payment Summary* page. The user can edit a given payment, changing its amount or type.

Features/Workflow: After the user has made the edits for a payment, they will click on the save changes link and be brought back to the *Detailed Payment Information* page.

Add New Payment

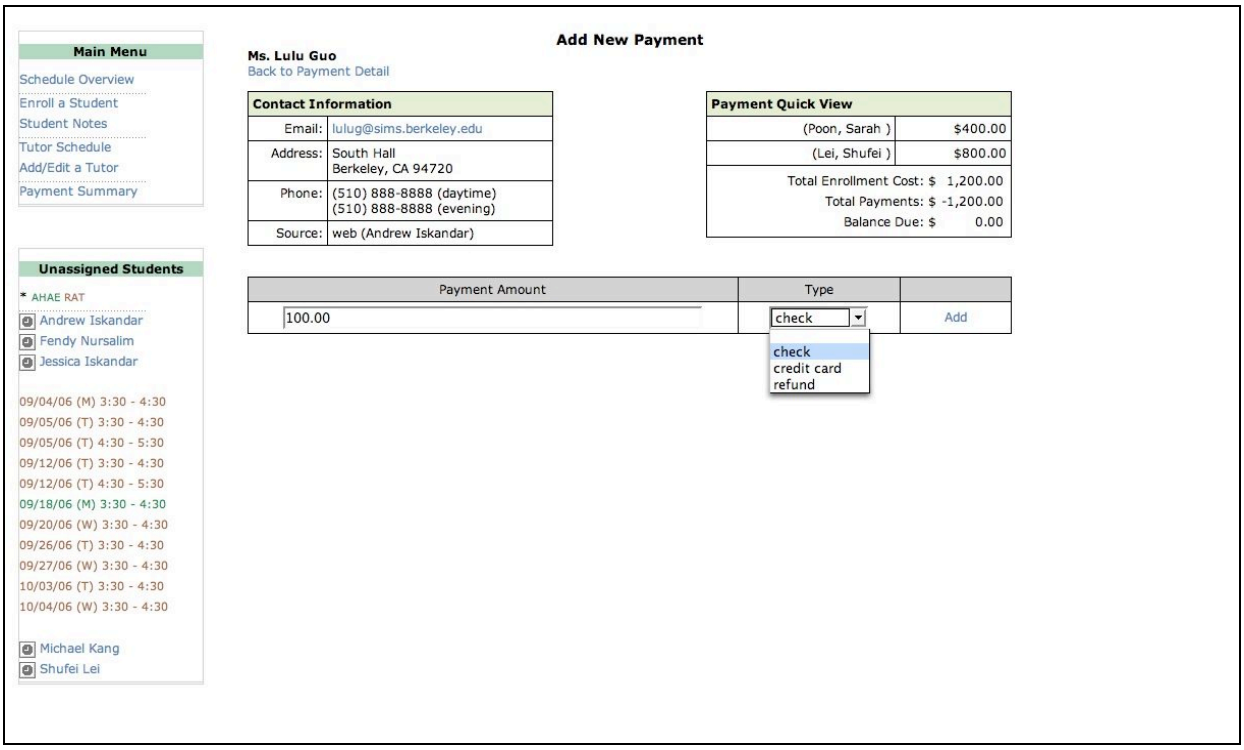


Figure 42: *Add New Payment*

Description: The user is brought to this page when she clicks on the add new payment link on the *Detailed Payment Summary* page. The user can add a new payment to this page.

Features/Workflow: After the user has inputted the payment information, she will click on the add link and a payment will be added to the system. The user will then be brought back to the *Detailed Payment Information* page.

System Interaction Flow Diagram

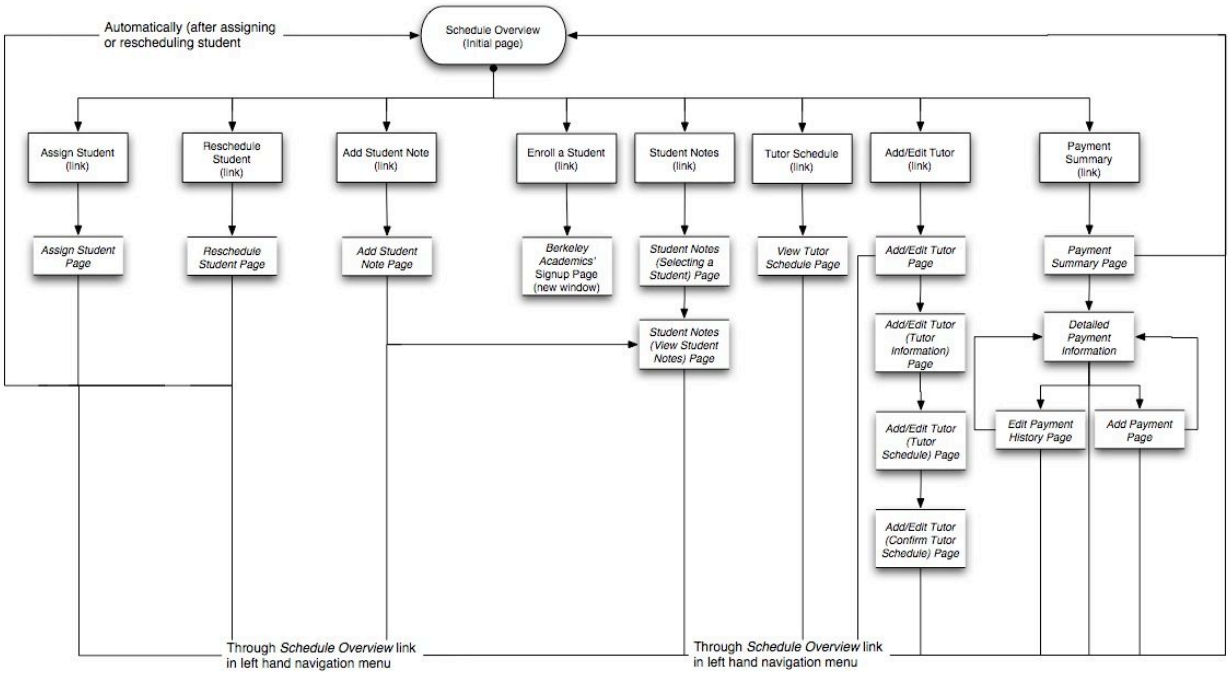


Figure 43 System interaction flow diagram of the application.

Future Development

During the testing of the application, our users and clients suggested additional features that could be incorporated into our application. Many of these features build upon the data collected through this system and would be a natural expansion for the project. Many of these suggestions came late during the development cycle and could not be incorporated into the system. Our clients only realized these feature requests after interacting with our hi-fidelity or final prototype.

- **Dynamic payment reminders** – While our system is capable of showing which users are overdue on their payments, our *Schedule Overview* page lends itself to a more up to date and dynamic reminder system.
- **Automatic/One-Click re-enrollment for students** – Often students try out *Berkeley Academics* for a period of time to see if they like it. A feature that would allow easy and fast re-enrollment based upon previous schedules would be well received.
- **Report generation** – Student progress is being collected in our application and this information can be used to automatically generate progress reports. Financial graphs and reports can also be created from the payment information being collected.
- **Data and trend analysis** – Since a lot more information is being gathered into a database through our system, analysis of this data can be used to show trends and patterns. These trends and patterns can then be used for marketing and curriculum development.
- **Searching and querying** – Our system generally uses drop-down lists that allow the user to visually search for information. If the company grows large enough, this method may become cumbersome and unwieldy. This issue can be addressed by introducing a searching and querying system.
- **Increased dynamic interaction** – While our system is functional and usable, some of the elements lend themselves to increased interactivity. Some of our interfaces, for example, can be transformed to use a "drag-and-drop" mode of interaction.

Conclusions

During the project, we learned important lessons as we faced challenges and various difficulties. We learned how important it is to gather business and user requirements early on in order to avoid changes during development. We also learned the importance of designing a database schema as it influences the development of the application itself. Through the various rounds of user testing, we also learned the need to design our user interface so that the potential user would not be confused or overloaded with information, especially since our project goal was to aggregate many tasks.

The *Berkeley Academics* staff received our proposed application very well. They were pleased with the functionality of the system and how it met their business and user requirements. They expressed their interest to quickly incorporate it into their workflow and how it would increase their efficiency and avoid previous problems.

APPENDIX A: Project Proposal

Final Project Proposal
The Berkeley Academics Information Redesign

Description

Berkeley Academics (www.berkeleyacademics.com) aims to provide students with the skills necessary to secure entry into the nation's elite institutions of higher learning. To help students build competence in verbal and math skills, Berkeley Academics cater to each student's level through their multi-tiered and personal curriculum.

Currently, much of the administrative work and record keeping at *Berkeley Academics* is paper-based. Not only does this consume processing resources and time but also causes a lot of difficulties in terms of record keeping and information updating. Based on our interviews with the staff at Berkeley Academics we concluded that the following issues might be contributing to this:

- The payment system is manual-based. The program directors need to manually look up in their paper records in order to find out the payment status of a customer. Moreover, there is neither a notification for the directors when a customer makes a payment online (through Paypal) nor a reminder-system when a payment is due.

- The similar kind of problem exists for history tracking of student enrollment and course progress. Since almost everything is paper-based, it is difficult for the directors to answer the questions pertaining to a student's enrollment status. Student progress and attendance tracking is poorly supported by the current system. Without adequate record-keeping, information retrieval tasks surround student information and progress can't be effectively carried out.

- Flexibility in tutor/student scheduling represents another problem. When a tutor or student can't show up for a scheduled session, the directors need to check the schedules of the other tutors one by one from their paper files and select the one that is both available and capable of tutoring that student. This matching process usually involves a significant amount of time and needs to be made more efficient, especially given the fact that the business is growing and the number of tutors and students is expected to increase in the near future.

The goal of our project is to move the current paper-based system to a web-based accessed database. The digital version of the system should be capable of storing information, allow querying, and increase the flexibility of scheduling and payment. The newly designed user interface and its backend database should enable users to easily add, delete, modify, and query the information related to students, courses, schedules, and payment. It should also support functions of flexible rescheduling of the tutors and students. It is hoped that this system will

make the administrative processes of staff more efficient and free up resources for other critical business endeavors.

Rationale

The *Berkeley Academics Information Redesign* project is a suitable capstone for the SIMS Master's degree because it integrates various emphases of the program. First and foremost this project will redesign and implement an information system for a real-world client. This will involve analyzing the current information system, designing an improved system, and finally implementing it. The proposed redesigned system will improve the storage, retrieval, and analysis of data for the client. In order to analyze the current information system, the project team will need to observe and interview the client. Because the client is currently using a paper-based information system in order to store, record, and analyze data, the project team will need to determine the salient processes to incorporate into the digital information system. Feature requests and requirements for the new information system will be determined during the observation and interviews of the client. This process will involve a basic needs assessment analysis.

The process of porting the current paper-based information system to a digital one will require several design processes. A functional and adoptable user interface to the system will be designed using user studies and an iterative design process. Based upon the analysis of the current information system and the client needs, personas, tasks, scenarios, lo-fi prototypes, and hi-fi prototypes will created to facilitate the design of a user interface. The goal will be to design a UI that is suitable to the characteristics of our client and their information tasks. The project team will also need to present the information in an efficient and clear manner to enhance the usability of our system. This will also involve iterations of the user-testing process. Finally, a technical system will be designed in order to implement the information system. This segment will include designing and integrating a database and a web-based interface. The design of the technical aspects of the system will emphasize functionality as well as expandability.

The implementation process will realize all the elements of the analysis and design phases of the project. Because this is a collaborative development, project management skills will be concretized during this process. The project team will learn to work together and according to an agreed upon timeline to complete the project.

The project team expects to utilize many of the skills they have learned during their careers at SIMS in this real-world project. User interface design, information visualization, user needs analysis, information retrieval, database design, and project management skills will all be used throughout the project. This project also has the added bonus of having a real-world client that will directly and immediately benefit from the implementation and deployment of the redesigned information system.

Roles

The people involved in the project and their major responsibilities are described below:

- **Andrew Iskandar:** Andrew will be the project manager and will be in charge of the database design. He will keep in close contact with Berkeley Academics staff as a client liaison to facilitate communication between the project team and the clients. He will also be a developer and designer of the final system.

- **Lulu Guo:** Lulu's role includes user interface designer and web developer. She is also responsible for the documentation and presentation of the design materials and products. She will also be a developer and designer of the final system.

- **Ray Larson:** Professor Larson will be the academic advisor of the project.

- **Berkeley Academic Staff (Sue Yi, Jeannie Lee, Suzanne Suh):** These are the clients/potential users from Berkeley Academic. They will be invited to provide their design ideas/requirements as well as feedbacks and to test multiple versions of prototypes during the course of the project.

APPENDIX B: Typical Problem Examples

So, let me describe to you what happened today:

- sue was sick. So suzanne (who's not been in the office much this semester b/c of baby and san ramon), and I are in the office.

- we try to figure out who's going to take over for sue's three hours of tutoring. Esther wang is available so we schedule her in.

- jenny kim comes in for 4:30-6:30pm tutoring appointment, but last week, they had told the student that he could come in "if he wants to, either Monday or Wednesday," and did not confirm with him whether or not he would be in. he doesn't show. But thank goodness he didn't show, b/c we had one extra tutor

- b/c, sue had scheduled in an "extra make-up hour" for a student that she didn't note and didn't tell us about. So we have an extra kid

- another kid shows up, it's his first day, he's on the schedule, but we weren't sure if he was coming in or not, since this is the last week of this session and he's never showed up before.

- his sister shows up and asks if she could be tutored right then and there as well.

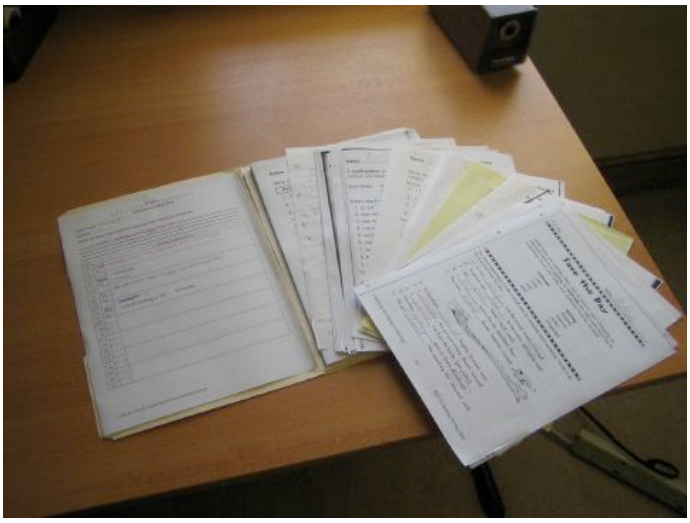
- another kid comes and says, sue told me I could come. We have no one to tutor her. she has BIO, which only jenny yen knows (who tutored her last week, but today was not available)

Needless to say... we need SOME sort of system...

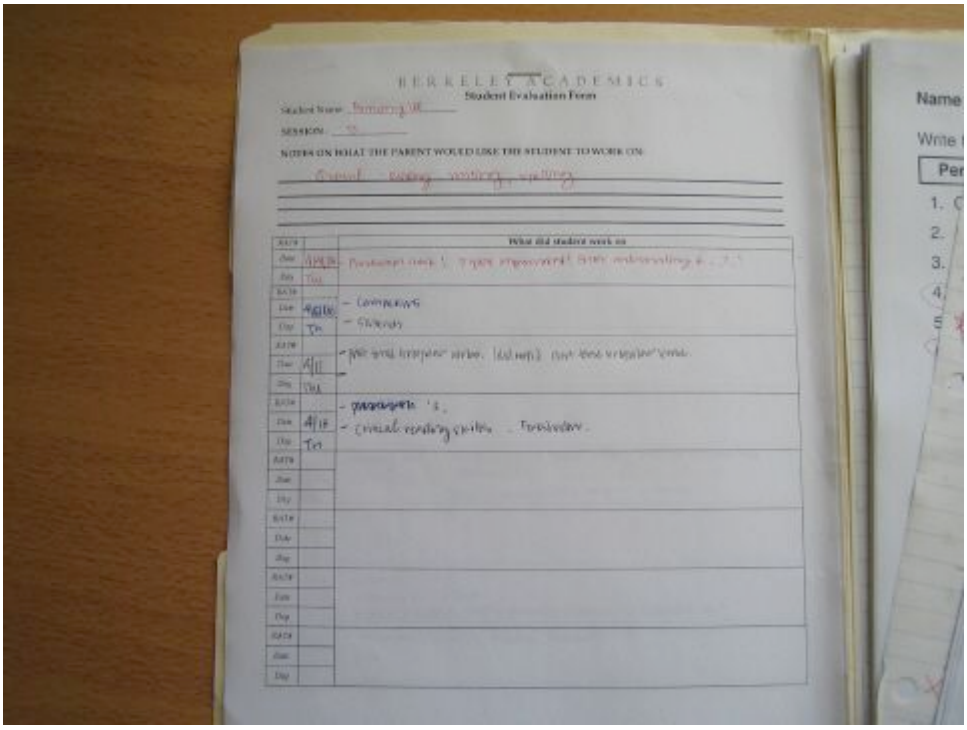
Email describing typical problems at Berkeley Academics

One of the many file cabinets holding student files and records

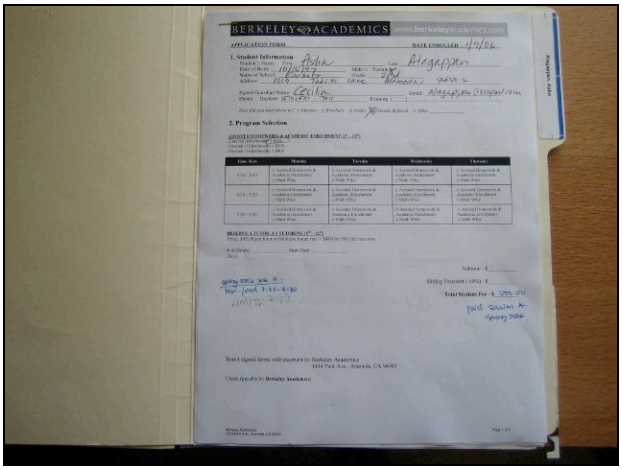
File cabinet



Files for a single student

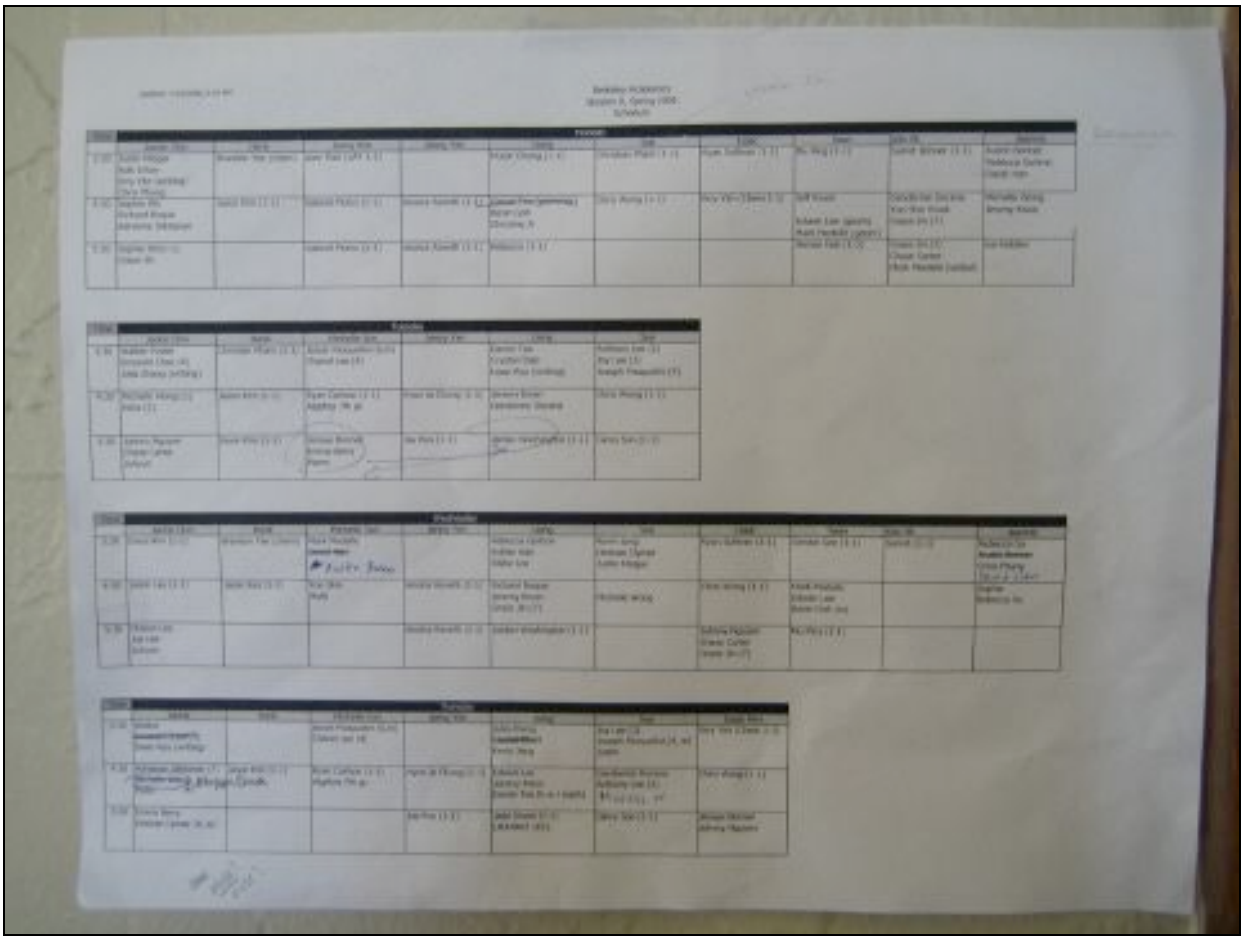


Student progress tracking



Student schedule information

Weekly schedule on a wall

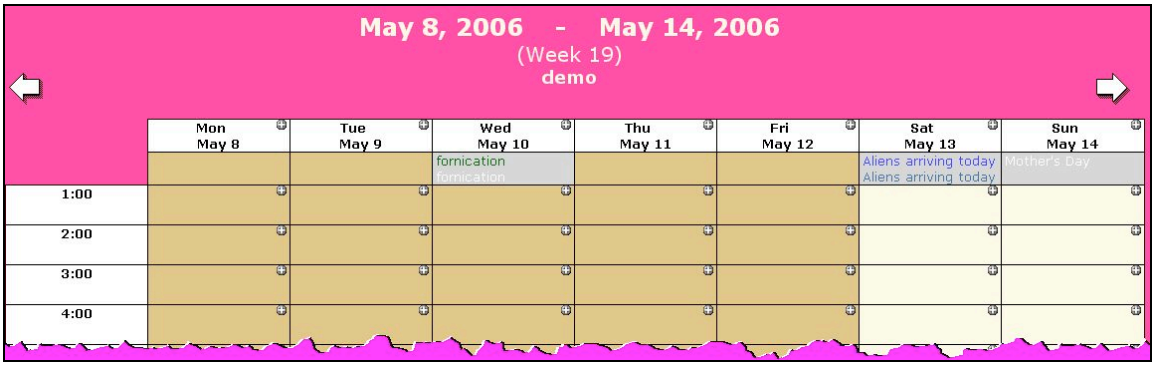


Weekly schedule made with Excel; changes are marked by hand

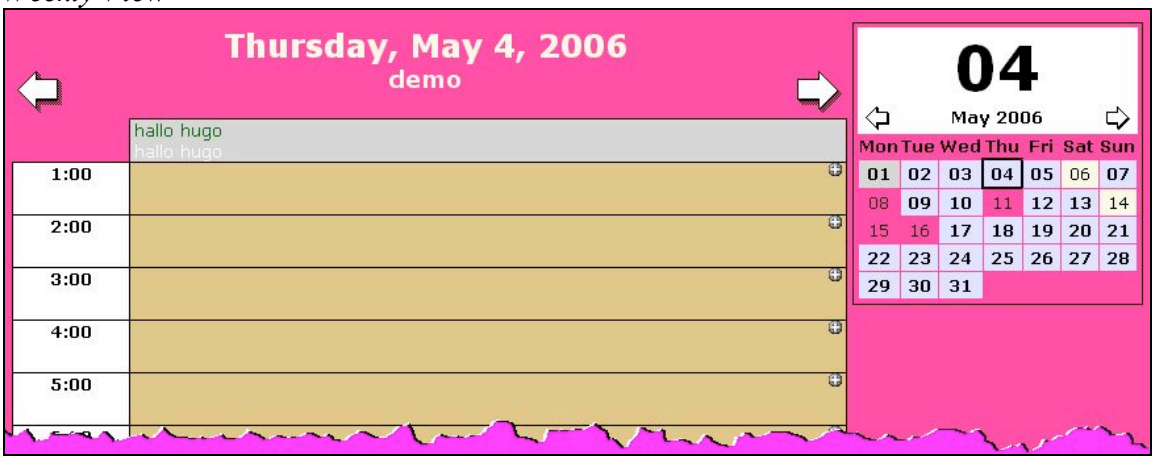
APPENDIX C: Comparative Analysis Screenshots

WebCalendar

Monthly View



Weekly View



Daily View

puppy class

Description:	free puppy class form members
Date:	Sunday, May 7, 2006
Time:	17:00
Priority:	Medium
Access:	Public
Created by:	demo
Updated:	Monday, May 1, 2006 9:08
Send Reminder:	Yes - 4 hours before event
Participants:	demo Public Access

Printer Friendly

Edit entry
Delete entry
Copy entry

Event View

Edit Entry

Details | Participants | Repeat

Brief Description: puppy class

Full Description: free puppy class form members

Access: Public

Priority: Medium

Date: 7 May 2006 Select...

Timed event

Time: 17 : 00

Duration: 0 : 00 (hours: minutes)

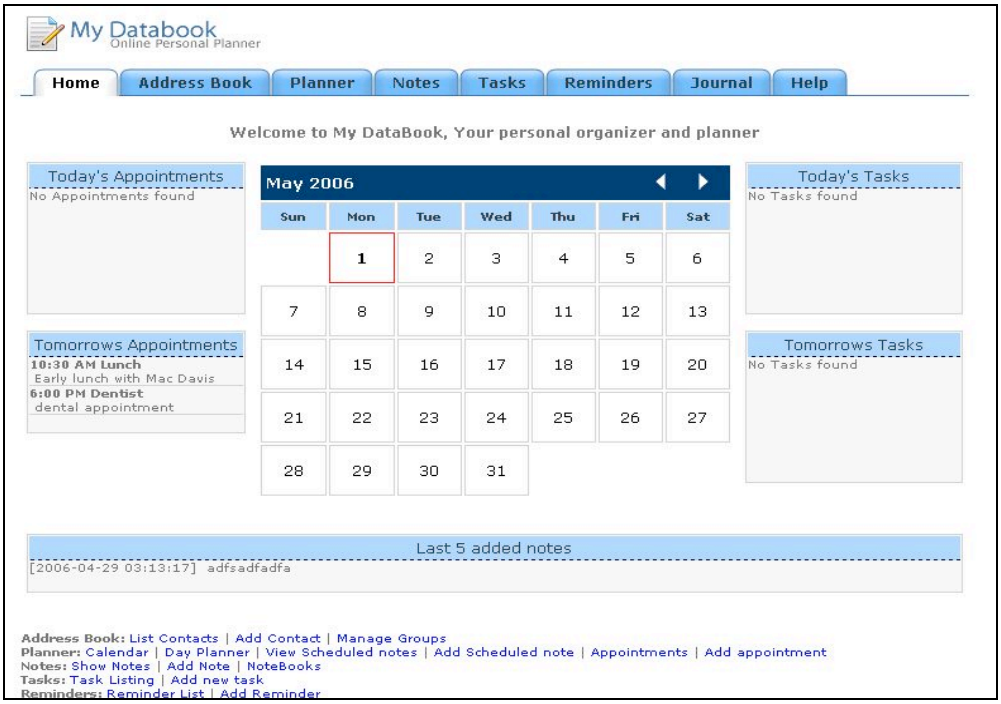
Send Reminder: Yes No 0 days 4 hours 0 minutes before event

Save

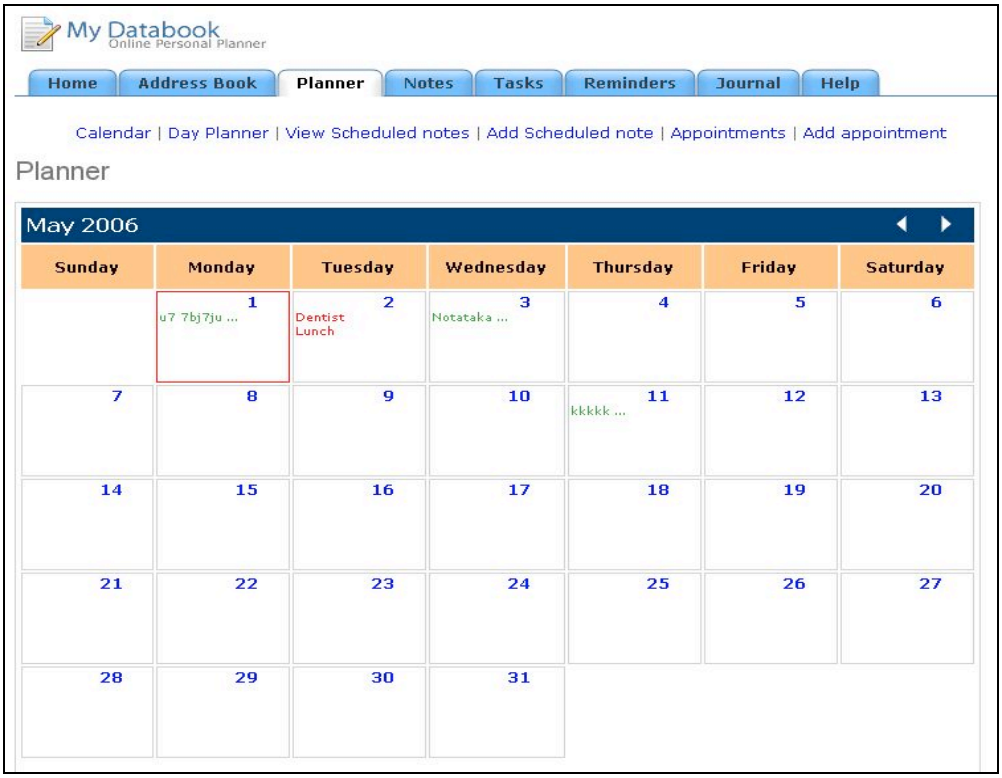
Delete entry

Edit Event

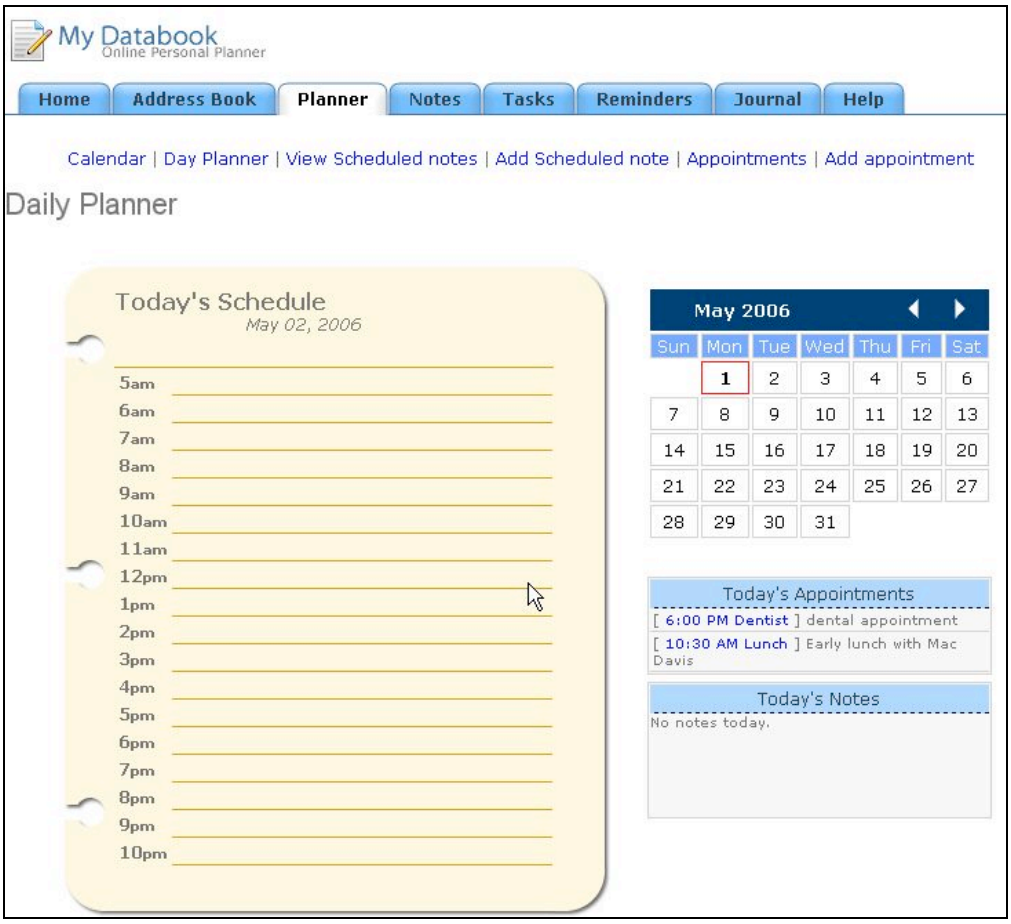
My Databook



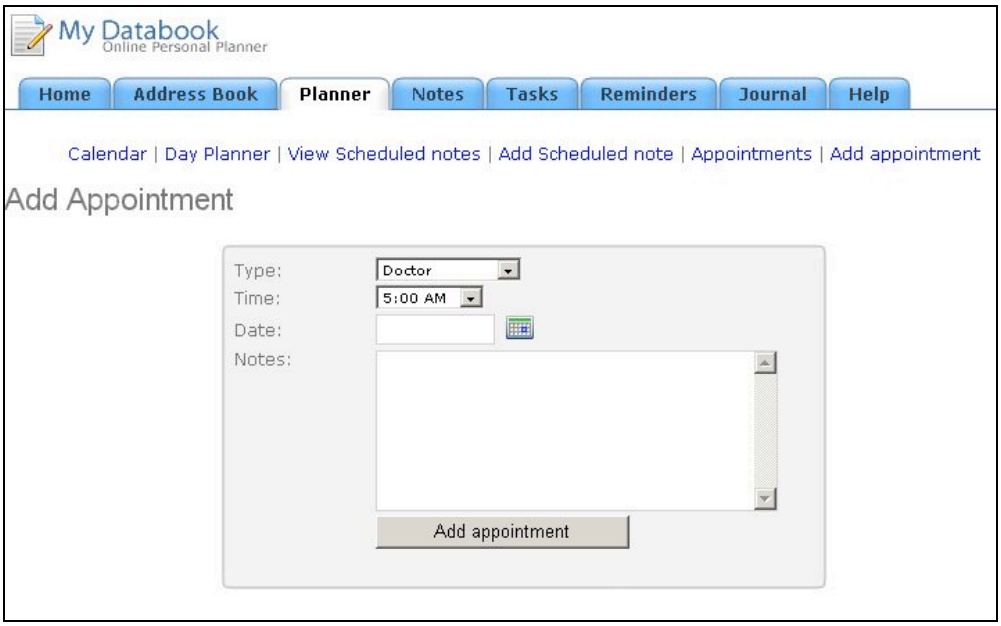
Home Page



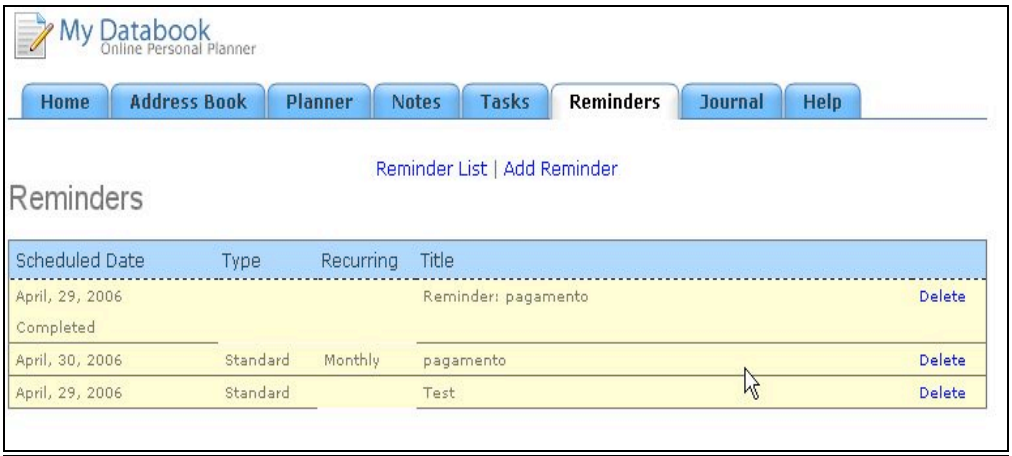
Planner



Daily Planner

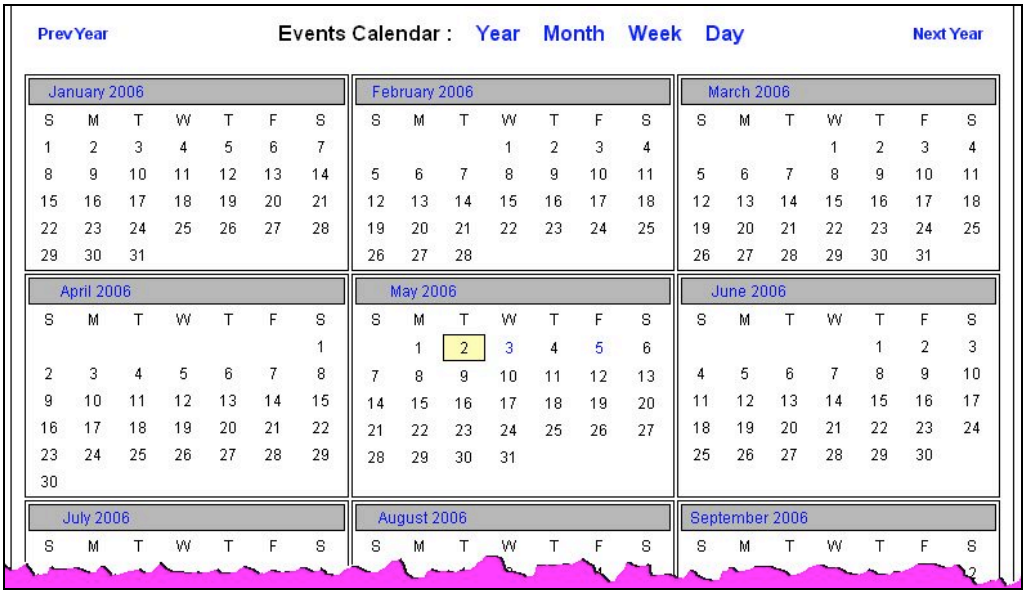


Add Appointment

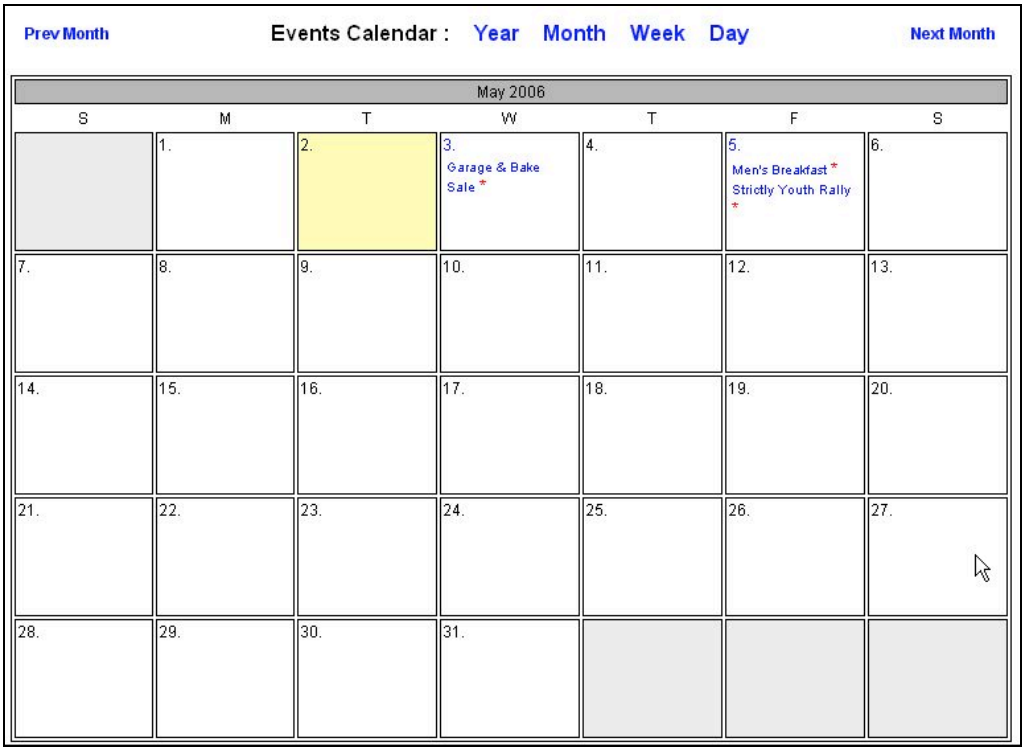

My Databook
Online Personal Planner
Home
Address Book
Planner
Notes
Tasks
Reminders
Journal
Help
Reminder List | Add Reminder
Reminders
Scheduled Date
Type
Recurring
Title
April, 29, 2006
Completed
Reminder: pagamento
Delete
April, 30, 2006
Standard
Monthly
pagamento
Delete
April, 29, 2006
Standard
Test
Delete

Reminders

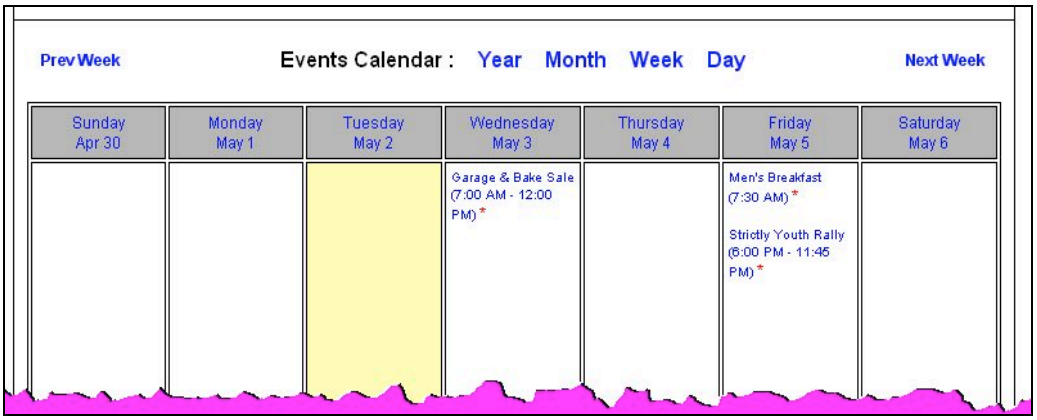
Comdev Event Calendar



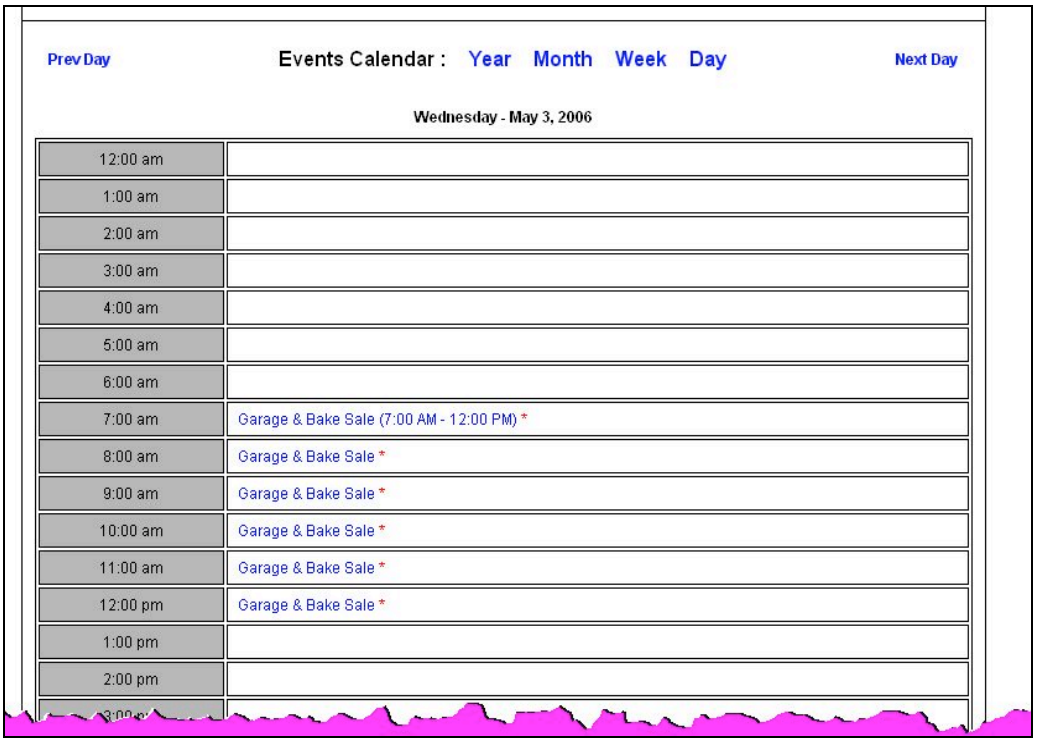
Yearly View



Monthly View



Weekly View



Daily View

Back

Events Calendar : Year Month Week Day

Event Details

Event : Garage & Bake Sale *
Venue : Christ Lutheran Church
Date & Time : 2006-05-03 (7:00 AM - 12:00 PM)
Type : Church
Description :
Location: Christ Lutheran Church
Church Address,
City, State, 10001
Other Events "Bag for a Dollar" time 11:00am to Noon

All proceeds will go towards building new playground equipment at the church.

Contact Person : John Doe , johndoe@domain.com , 012 456 7899
Michael Allan , allan@domain.com , 606 123 4544

Tell A Friend About This Event

Your Name : * Your Email : *
Friend's Name : * Friend's Email : *
Your Message :

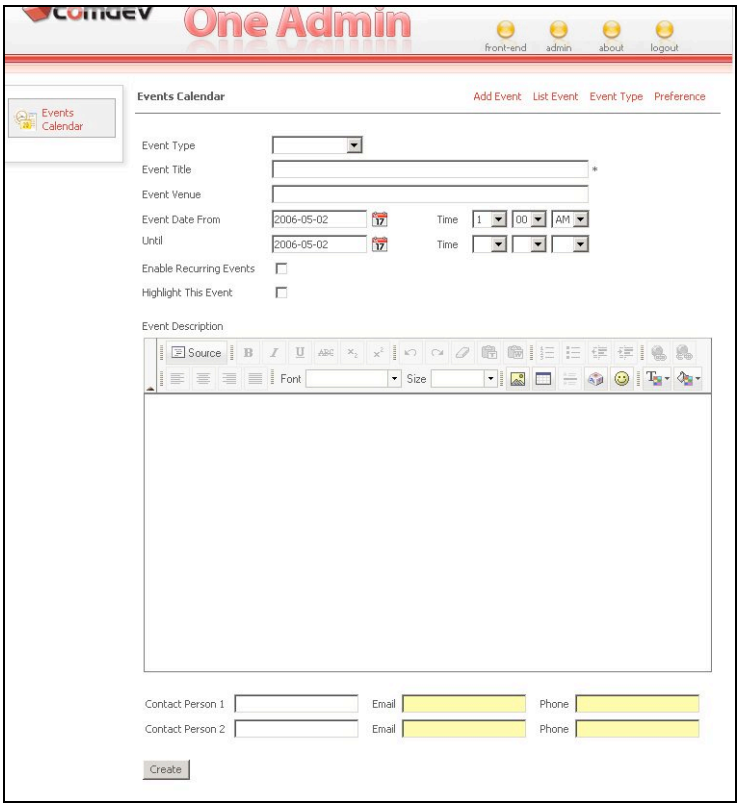
Send

Remind About This Event

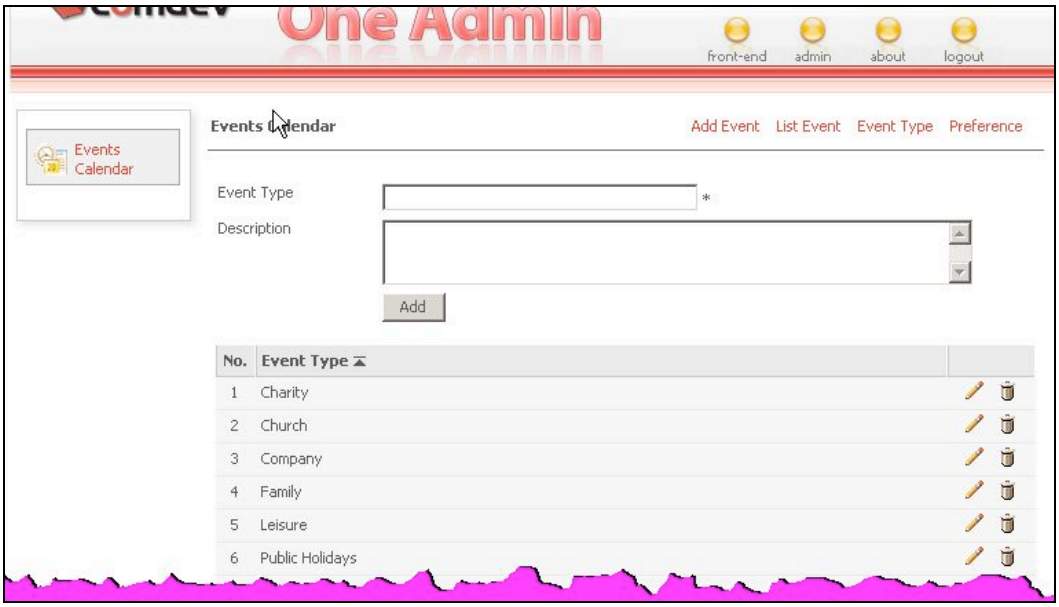
Your Name : * Your Email : *
Remind Me In : 3 Days

Remind Me

Event View

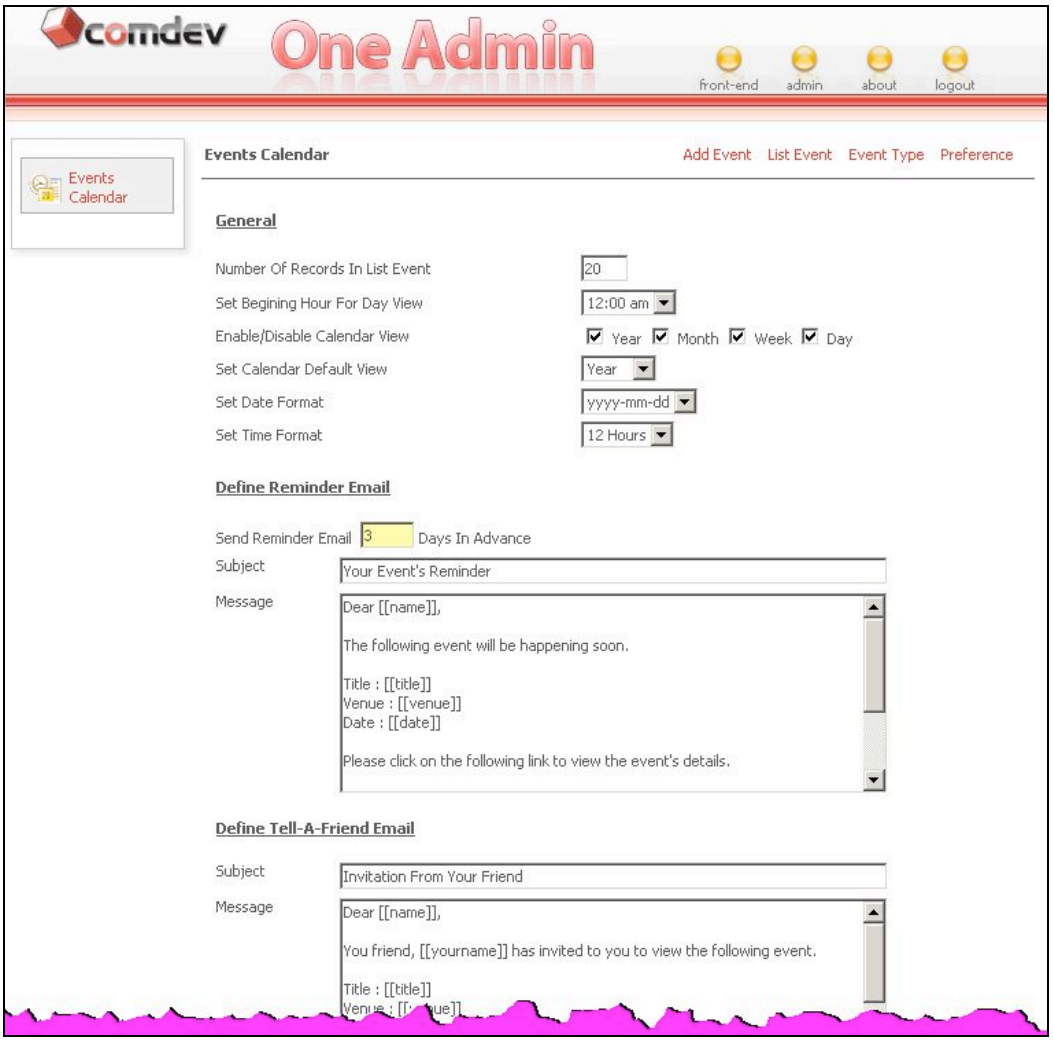


Add Event



Event Type

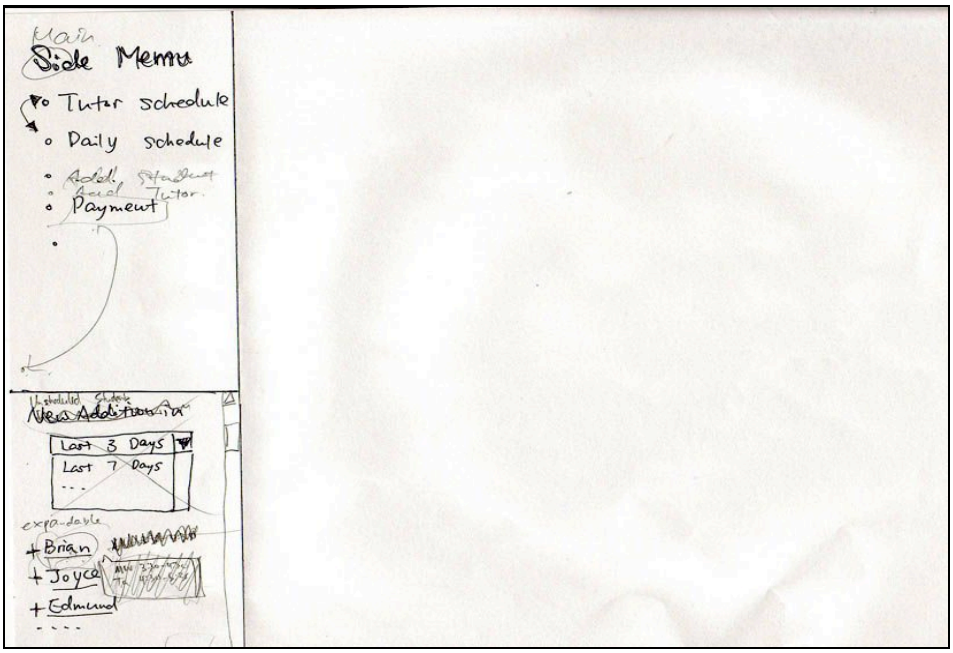
List Event


comdev
One Admin
front-end
admin
about
logout
Events Calendar
Events Calendar
Add Event
List Event
Event Type
Preference
General
Number Of Records In List Event
20
Set Begining Hour For Day View
12:00 am
Enable/Disable Calendar View
Year
Month
Week
Day
Set Calendar Default View
Year
Set Date Format
yyyy-mm-dd
Set Time Format
12 Hours
Define Reminder Email
Send Reminder Email
3
Days In Advance
Subject
Your Event's Reminder
Message
Dear [[name]],
The following event will be happening soon.
Title : [[title]]
Venue : [[venue]]
Date : [[date]]
Please click on the following link to view the event's details.
Define Tell-A-Friend Email
Subject
Invitation From Your Friend
Message
Dear [[name]],
You friend, [[yourname]] has invited to you to view the following event.
Title : [[title]]

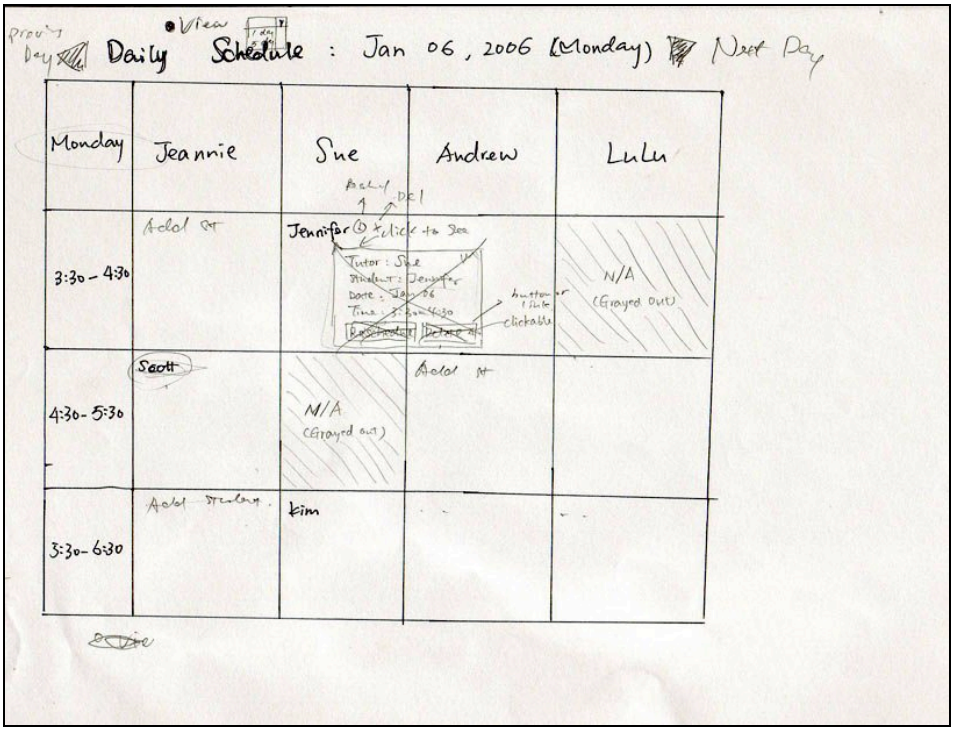
Preference

APPENDIX D: Design Evolution Screenshots

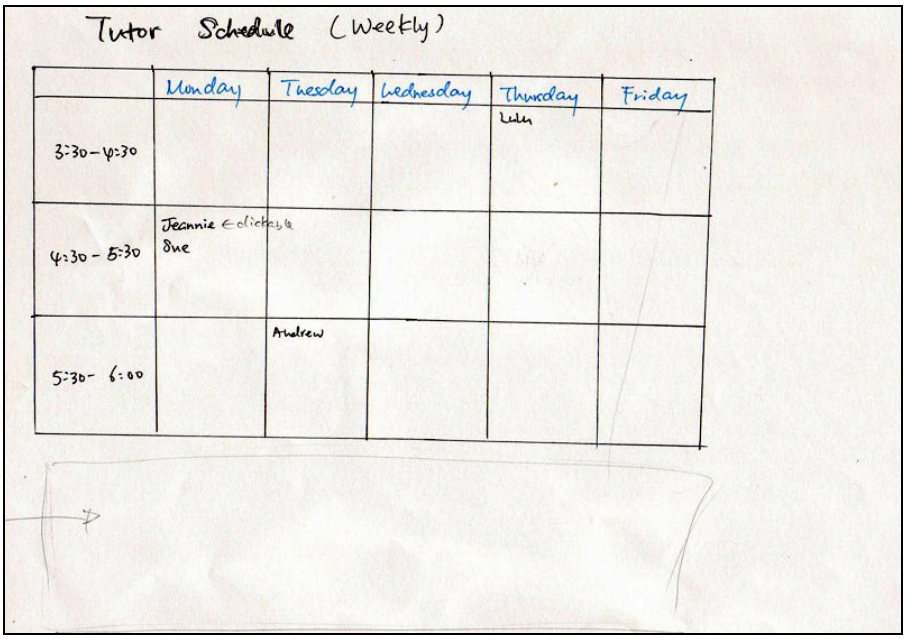
1) Initial Sketches



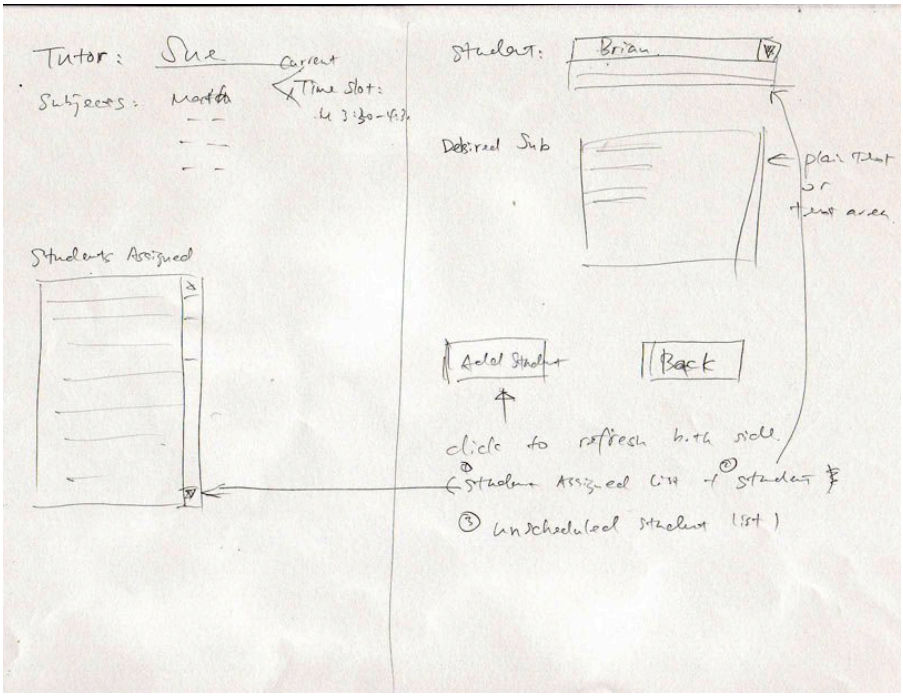
Left Menu



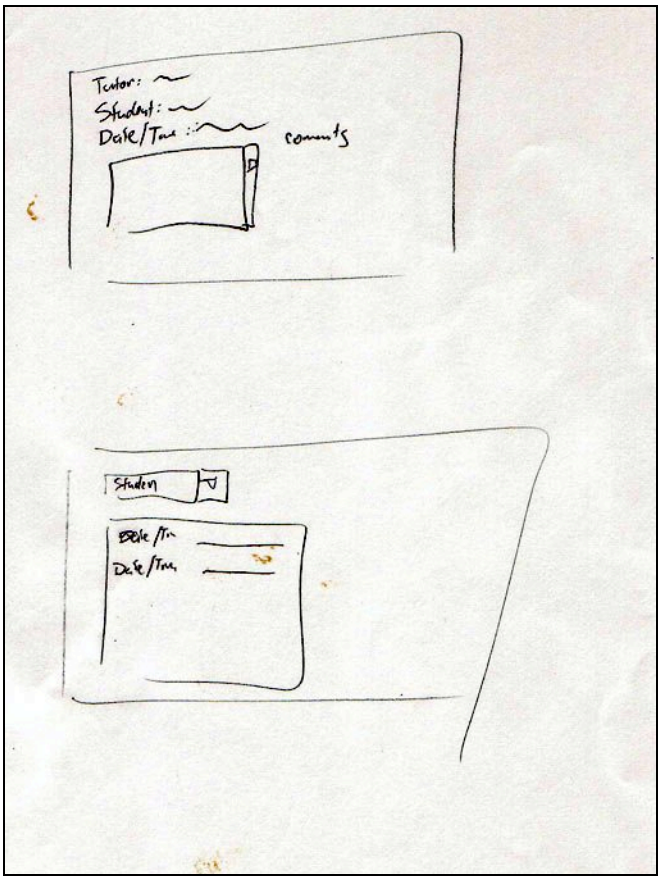
Schedule Overview: Daily View



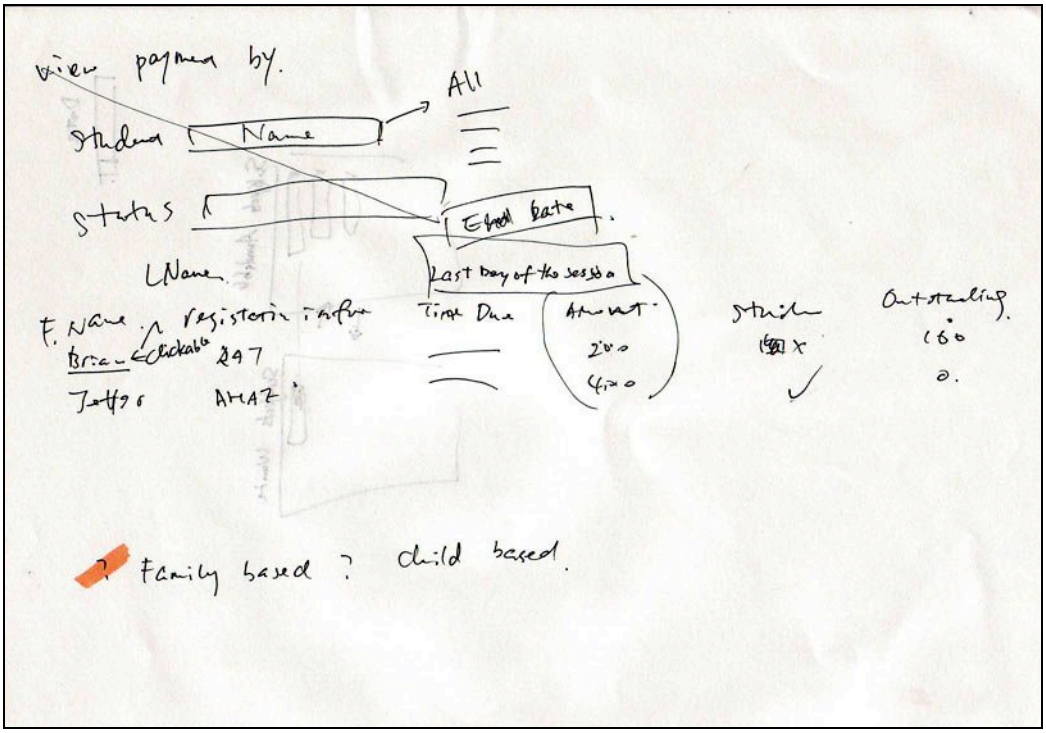
Tutor Schedule



Add A Student

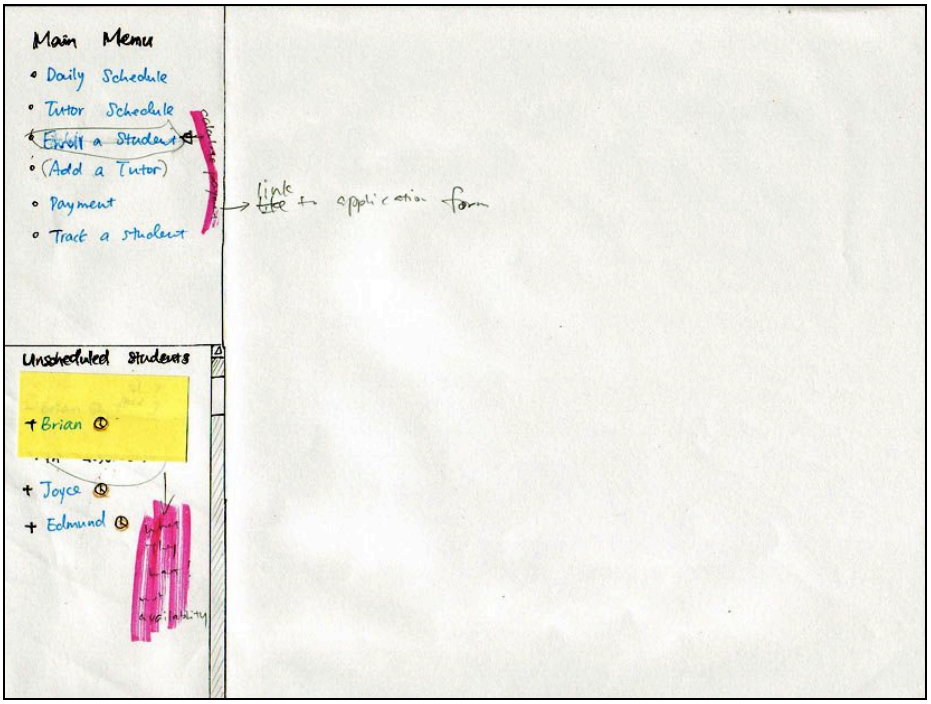


Track A Student

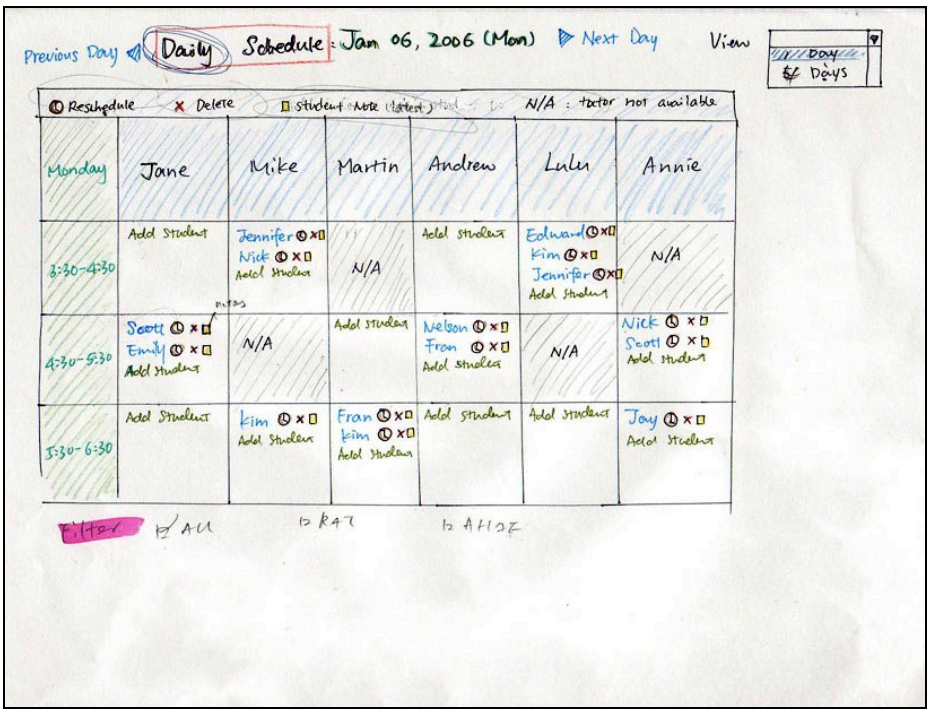


Payment Summary

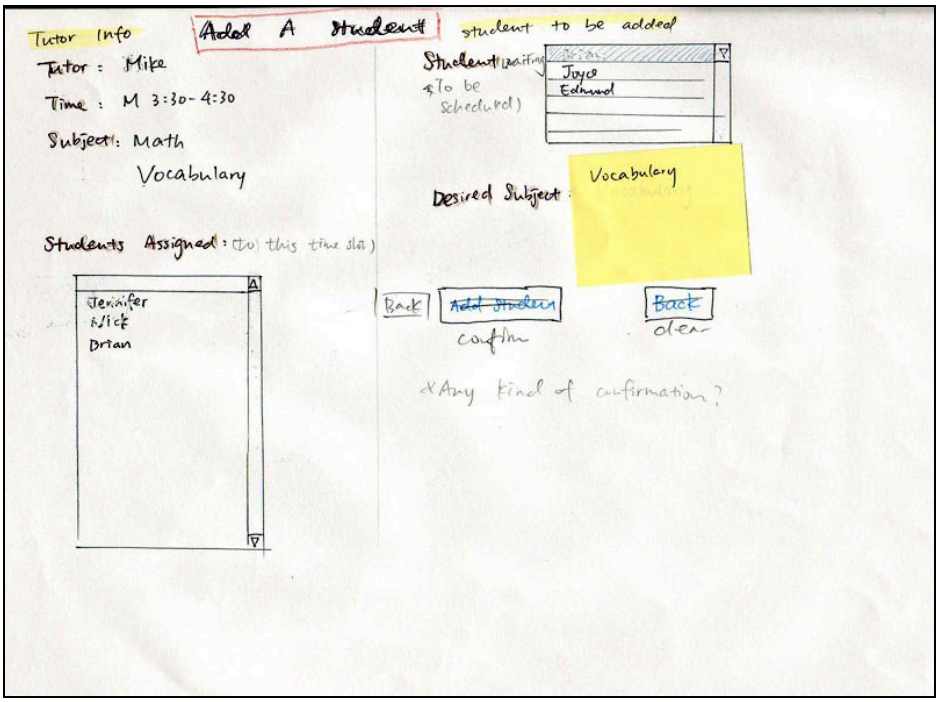
2) Paper Prototype



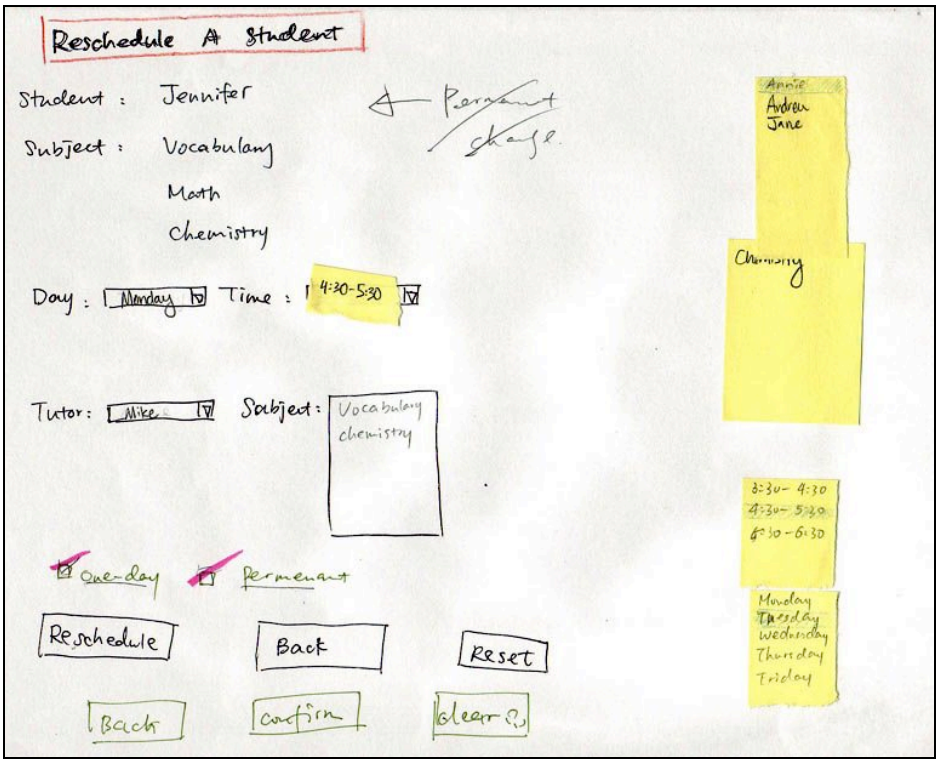
Left Menu



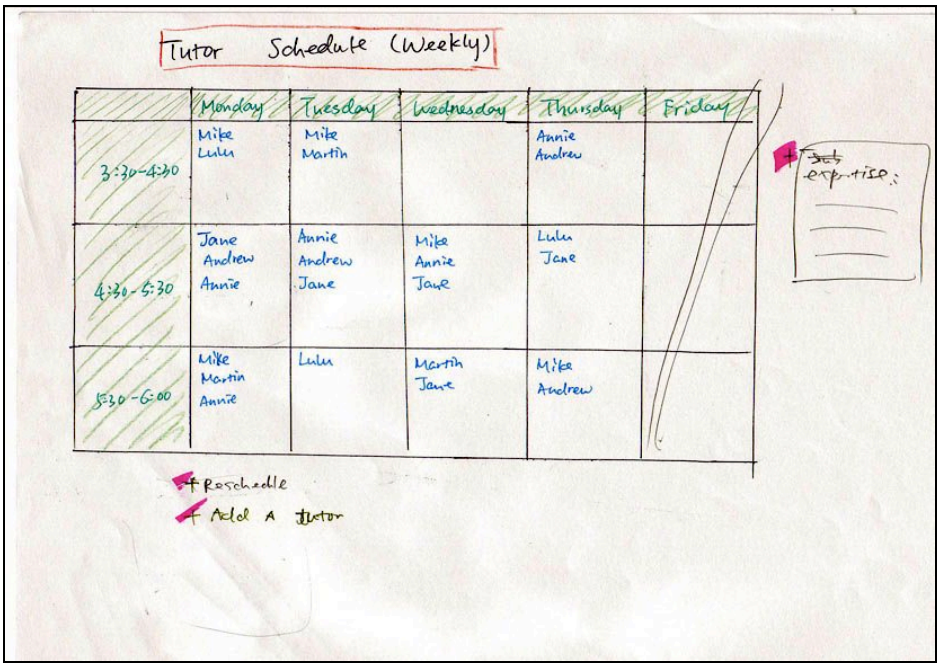
Schedule Overview: Daily View



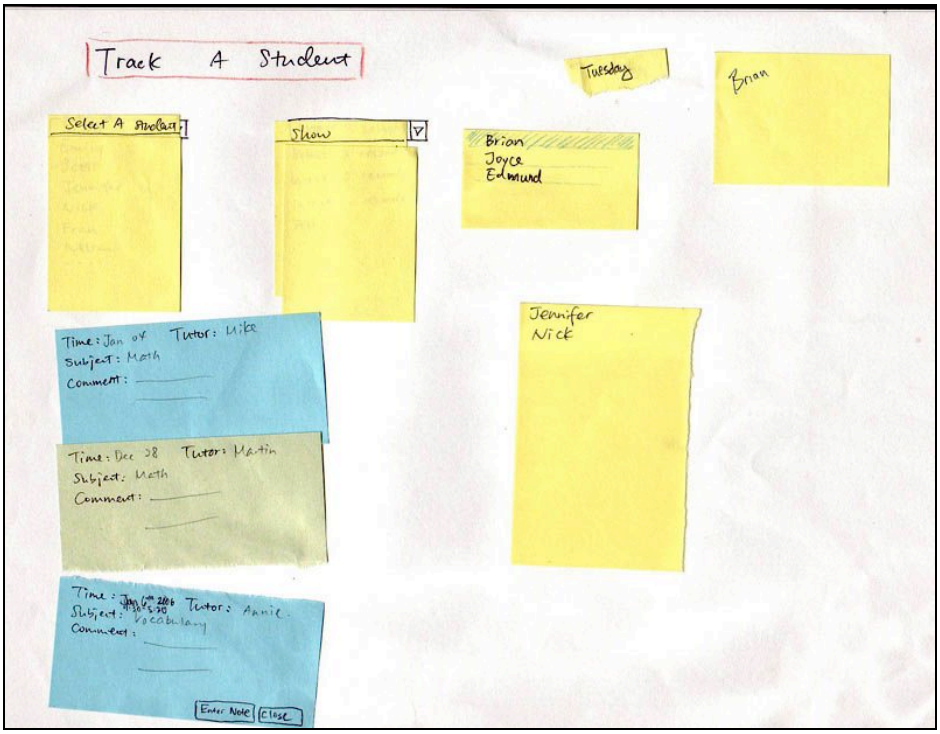
Add A Student



Reschedule A Student

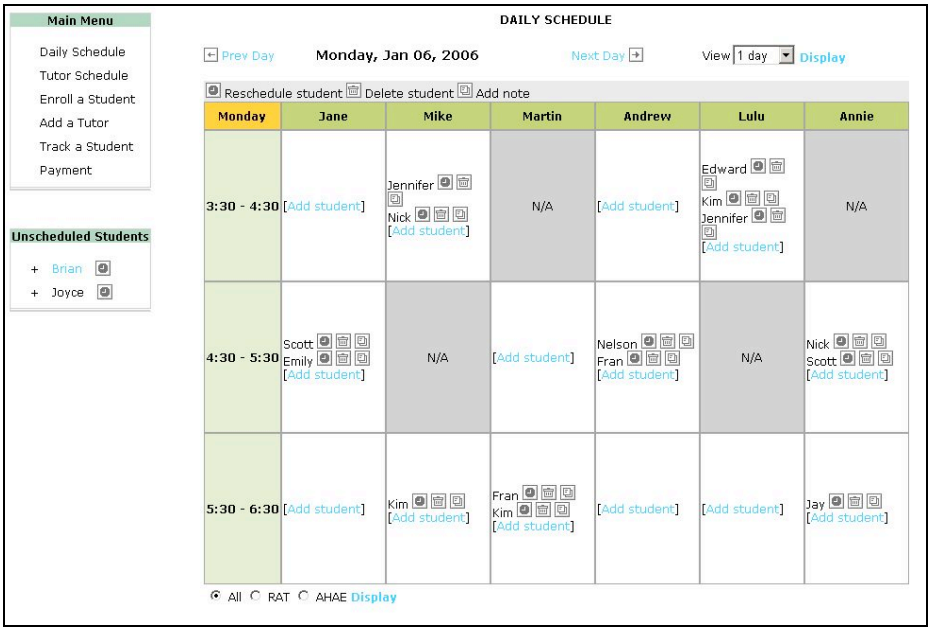


Tutor Schedule

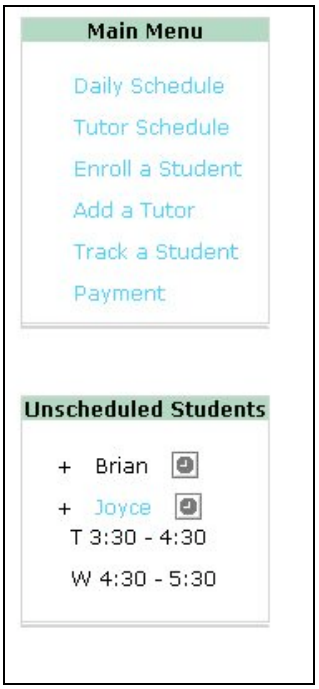


Track A Student

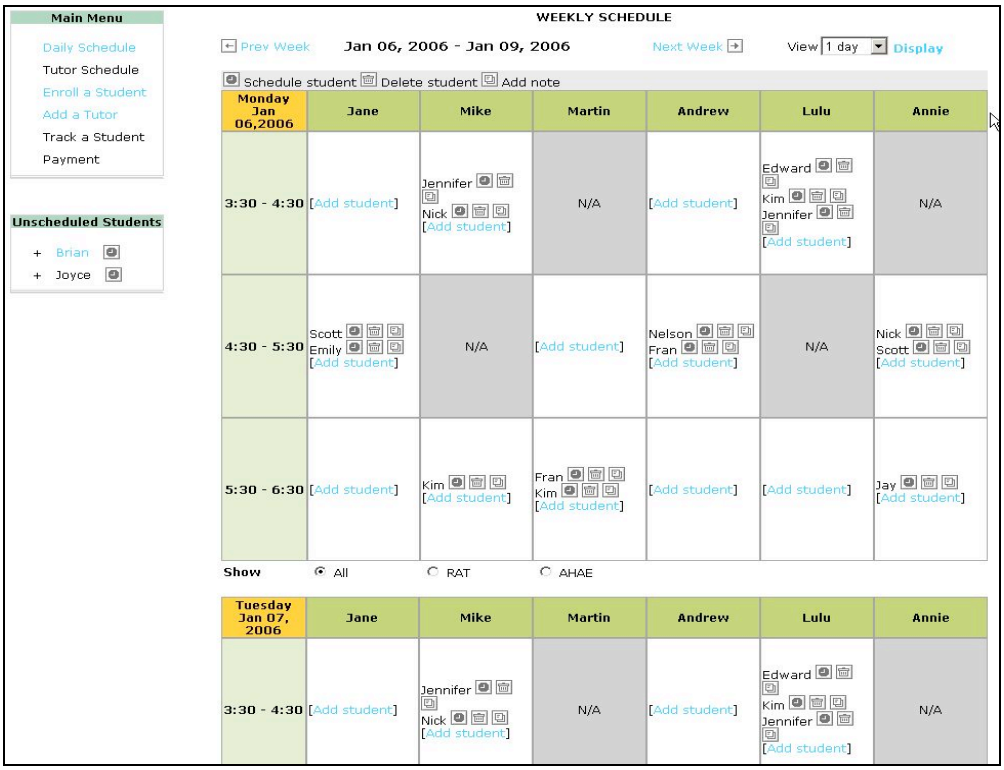
3) Low-fidelity Prototype



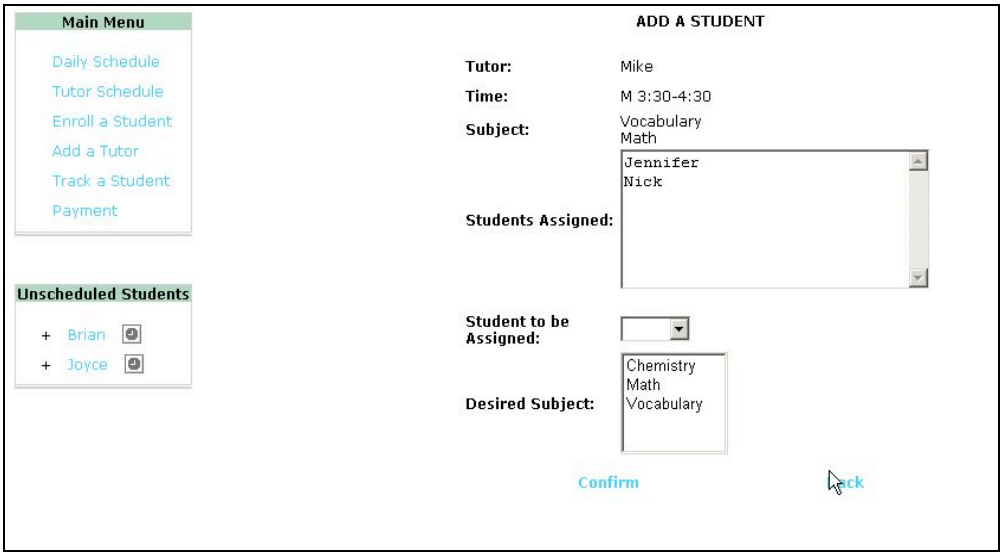
Schedule Overview: Daily View



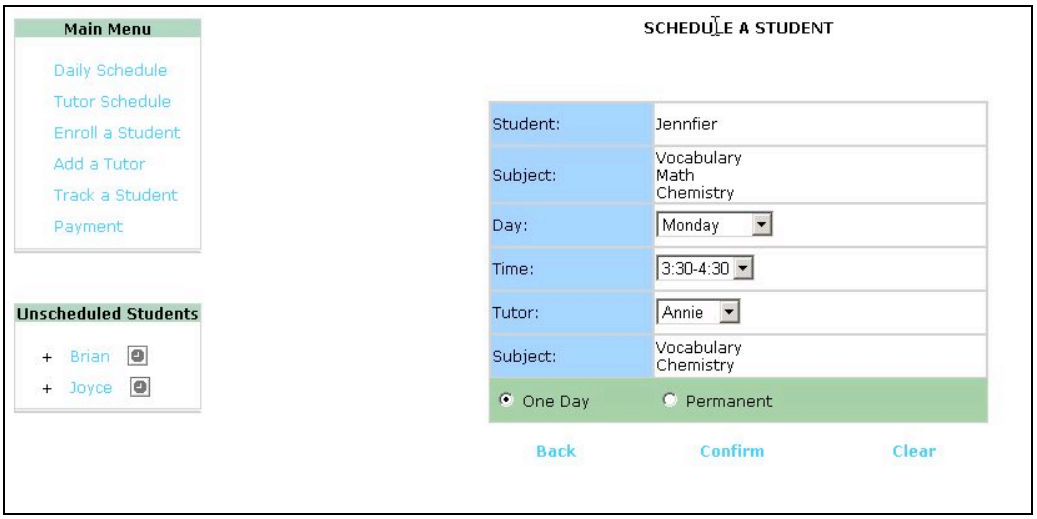
Left Menu expanded



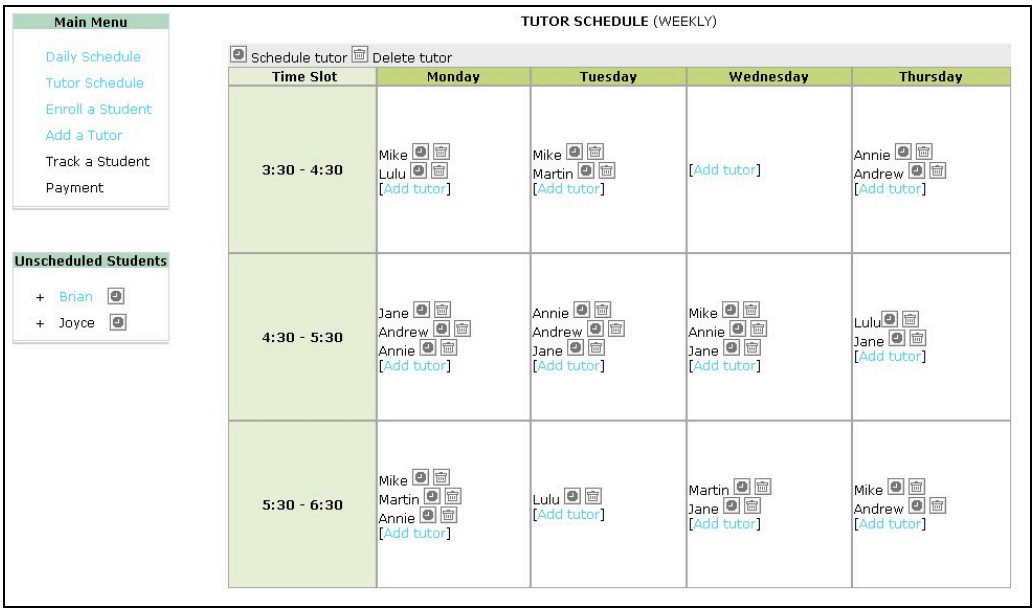
Schedule Overview: Weekly View



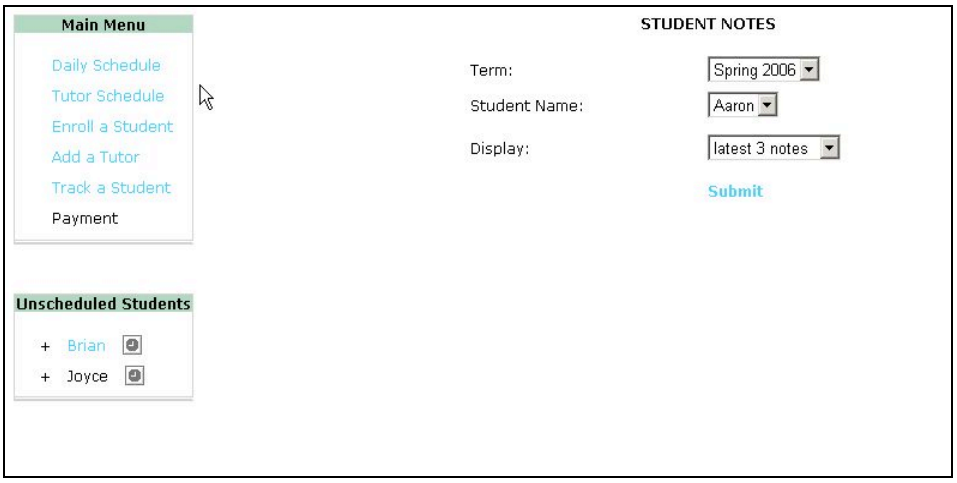
Add A Student



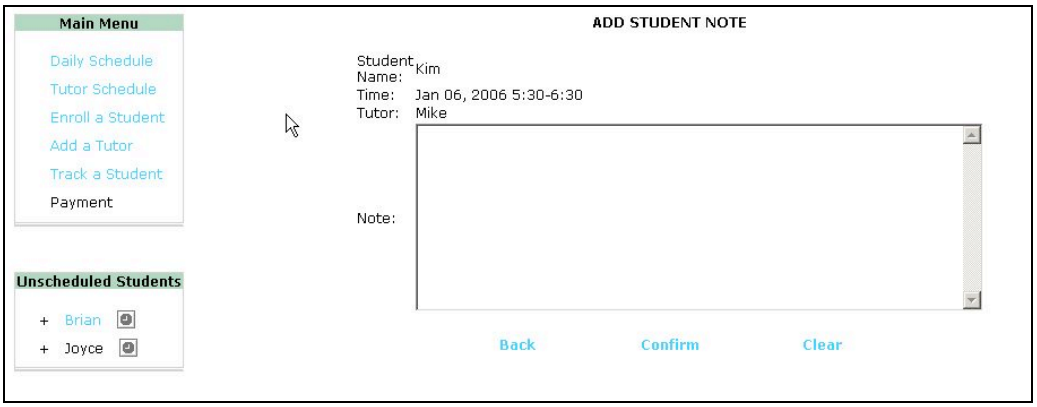
(Re)Schedule A Student



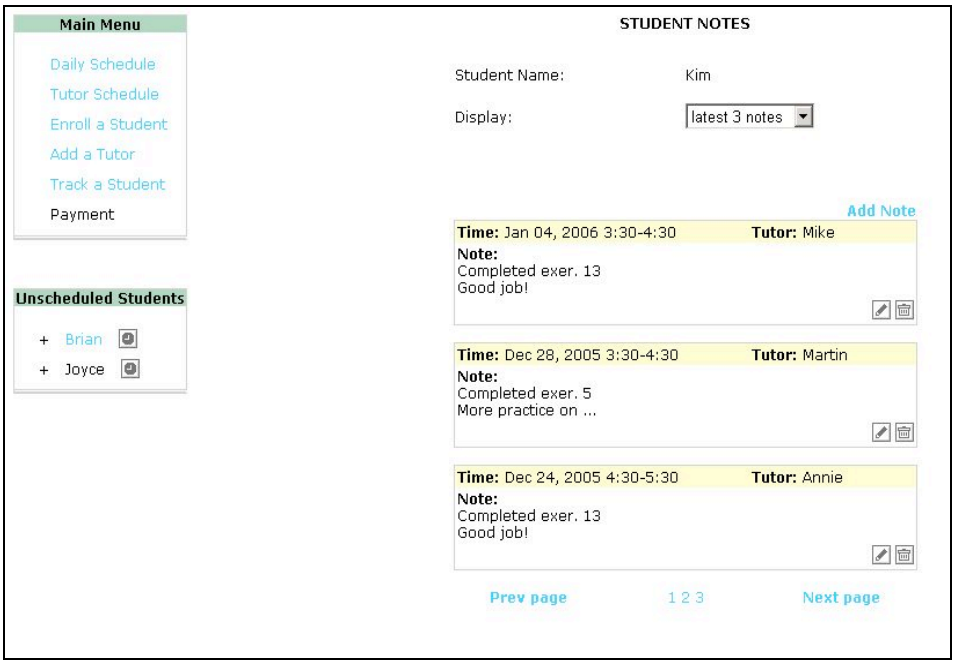
Tutor Schedule



Track A Student



Add Student Note



Student Notes

Main Menu

Daily Schedule
Tutor Schedule
Enroll a Student
Add a Tutor
Track a Student
Payment

Unscheduled Students

+ Brian
+ Joyce

EDIT STUDENT NOTE

Student Name: Kim
Time: Jan 04, 2006 3:30-4:30
Tutor: Mike
Subject: Math
Note:

```
Completed exer. 13
Good job!
```

Back Confirm Clear

Edit Student Note

Main Menu

Daily Schedule
Tutor Schedule
Enroll a Student
Add a Tutor
Track a Student
Payment

Unscheduled Students

+ Brian
+ Joyce

PAYMENT SUMMARY

Enrollment Name	Student Name	Total Charges	Total Payments	Balance
HOPKINS, Merle	HANN, Rebecca HANN, Kevin	1,000.00	1,000.00 Details	0.00
FLORES, Brenda	MILLS, Patricia	400.00	200.00 Details	200.00
	Total:	600.00	500.00	200.00

Payment Summary

Main Menu

Daily Schedule
Tutor Schedule
Enroll a Student
Add a Tutor
Track a Student
Payment

Unscheduled Students

+ Brian
+ Joyce

PAYMENT DETAILS

Enrollment Info

Enrollment Name: HOPKINS, Merle
Student Name: HANN, Rebecca
HANN, Kevin

Phone: 510-318-3712 **Email:** Hopkins@hotmail.com
Address: 1283 Sunflower St. Pleasanton, CA 93284

Student Schedule

Student	Course/Session	Duration	Time	Charge
HANN, Rebecca	SAT Critical Reading & Writing	06/27/05 - 08/04/05	M/Tu/W/Th 10:00-12:00	$600.00
HANN, Kevin	Math	06/27/05 - 07/07/05	M/Tu/W/Th 1:00-2:30	$400.00
	Essay Writing Lab	06/27/05 - 07/07/05	M/Tu/W/Th 10:30-12:00	$200.00
			Discount:	- $200.00
			Total Charge:	$1,000.00

Payment History:	Date	Payments	Balance	Type		
	Sep 14, 2005	500.00	500.00	Credit Card	Edit	Delete
	Oct 20, 2005	200.00	300.00	Credit Card	Edit	Delete
	Nov 03, 2005	100.00	200.00	Check	Edit	Delete
	Dec 21, 2005	200.00	0.00	Credit Card	Edit	Delete
	Total:	1,000.00	0.00			

Add New Payment

Back to Payment Summary

Payment Details

Main Menu

Daily Schedule
Tutor Schedule
Enroll a Student
Add a Tutor
Track a Student
Payment

Unscheduled Students

+ Brian
+ Joyce

ADD PAYMENT

Enrollment Info
Enrollment Name: HOPKINS, Merle
Student Name: HANN, Rebecca
HANN, Kevin

Phone: 510-318-3712
Address: 1283 Sunflower St. Pleasanton, CA 93284
Email: Hopkins@hotmail.com

Date	Payments	Balance	Type

Back Confirm

Add Payment

Main Menu

Daily Schedule
Tutor Schedule
Enroll a Student
Add a Tutor
Track a Student
Payment

Unscheduled Students

+ Brian
+ Joyce

EDIT PAYMENT DETAILS

Enrollment Info
Enrollment Name: HOPKINS, Merle
Student Name: HANN, Rebecca
HANN, Kevin

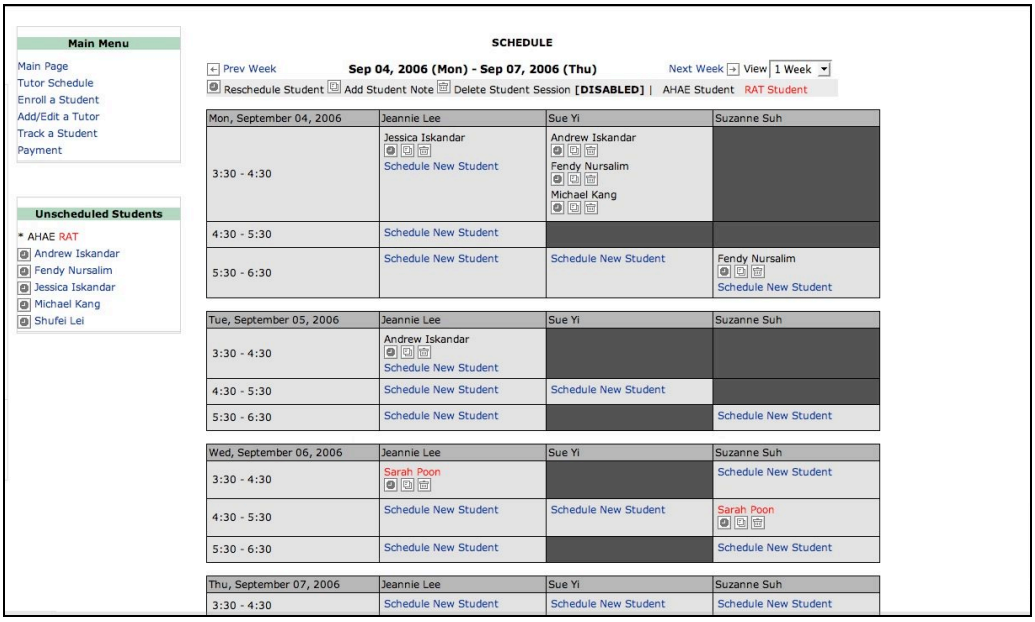
Phone: 510-318-3712
Address: 1283 Sunflower St. Pleasanton, CA 93284
Email: Hopkins@hotmail.com

Payment History:	Date	Payments	Balance	Type	
	Sep 14, 2005	500.00	500.00	Credit Card	Edit

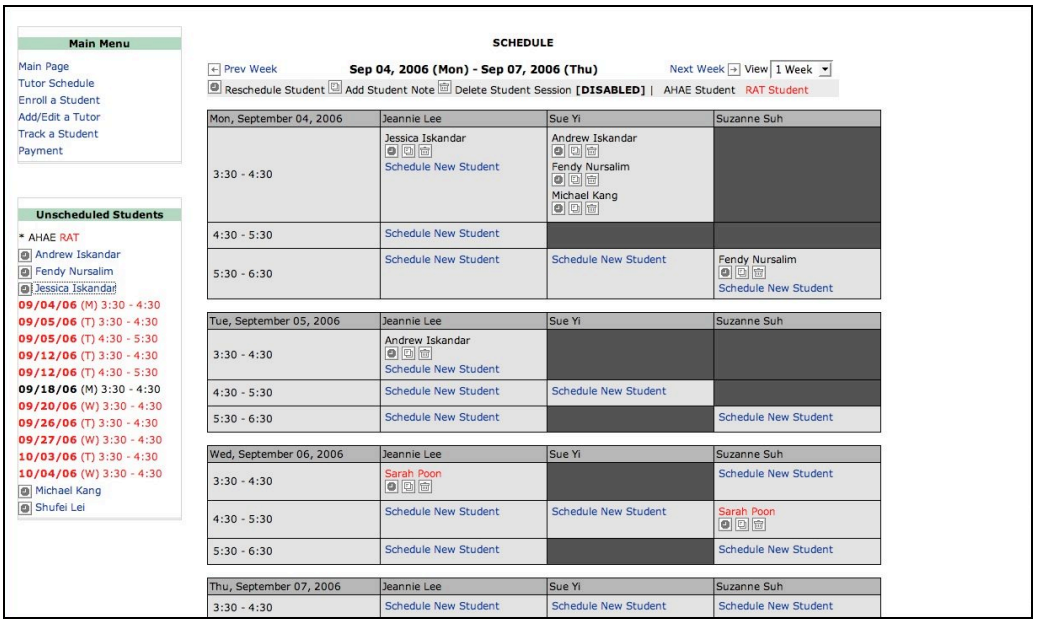
Back to Payment Details

Edit Payment

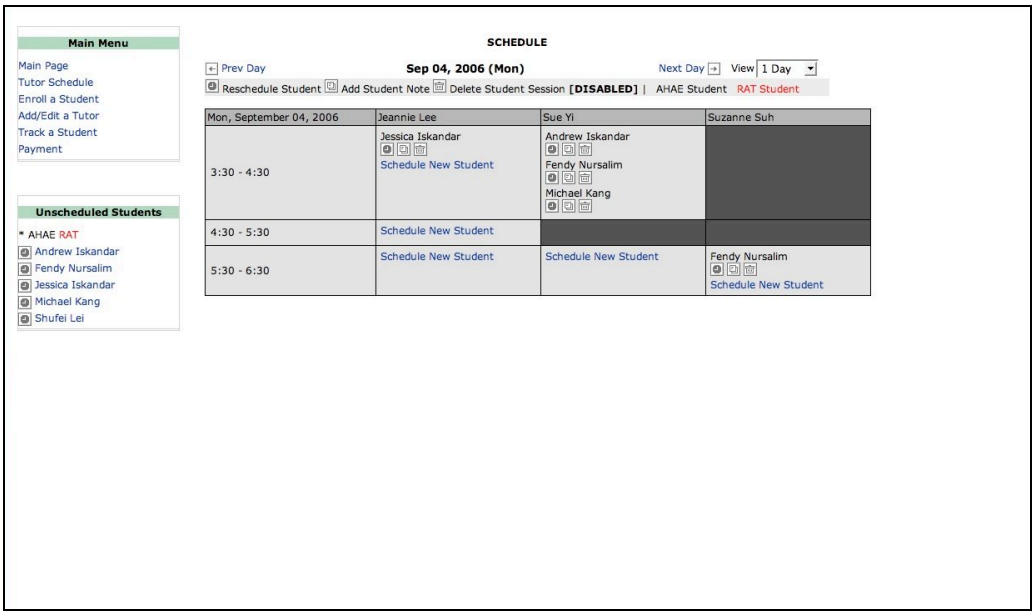
4) High-fidelity Prototype



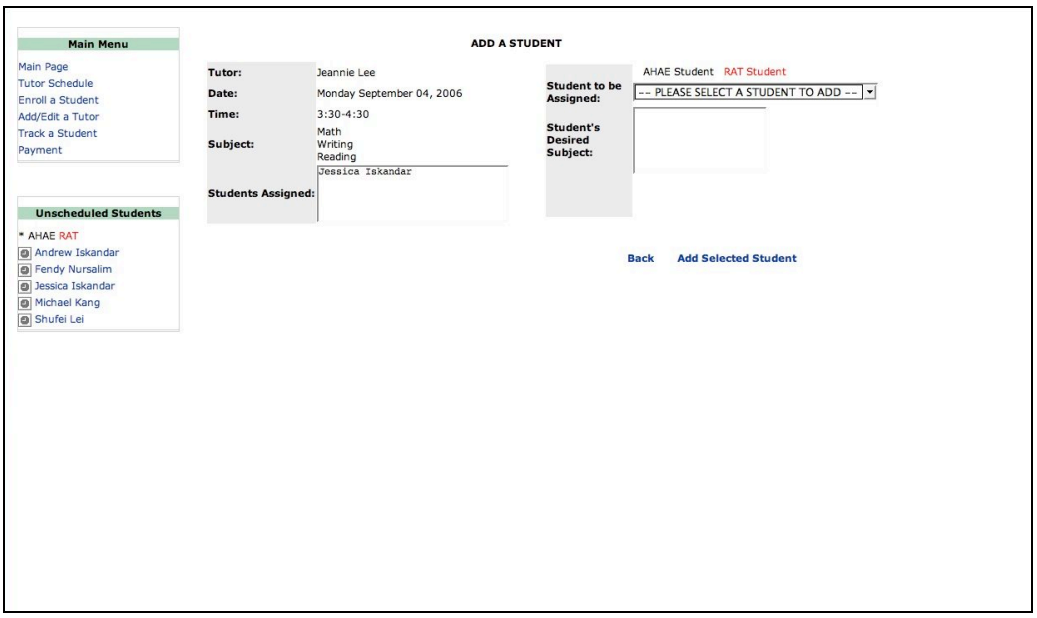
Schedule Overview: Weekly View (default)



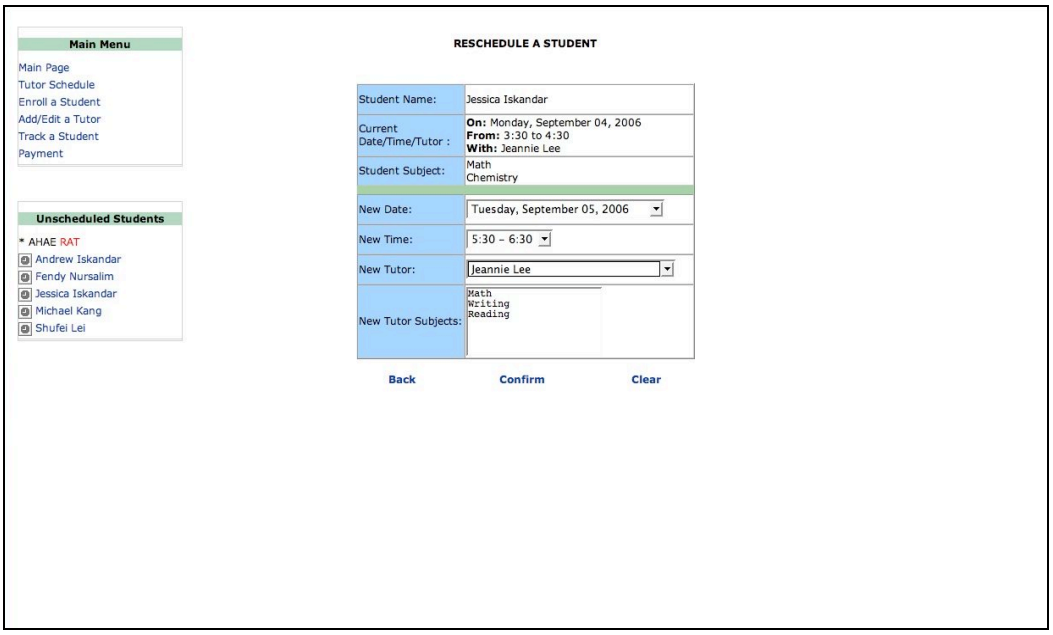
Left Menu expanded and color-coded



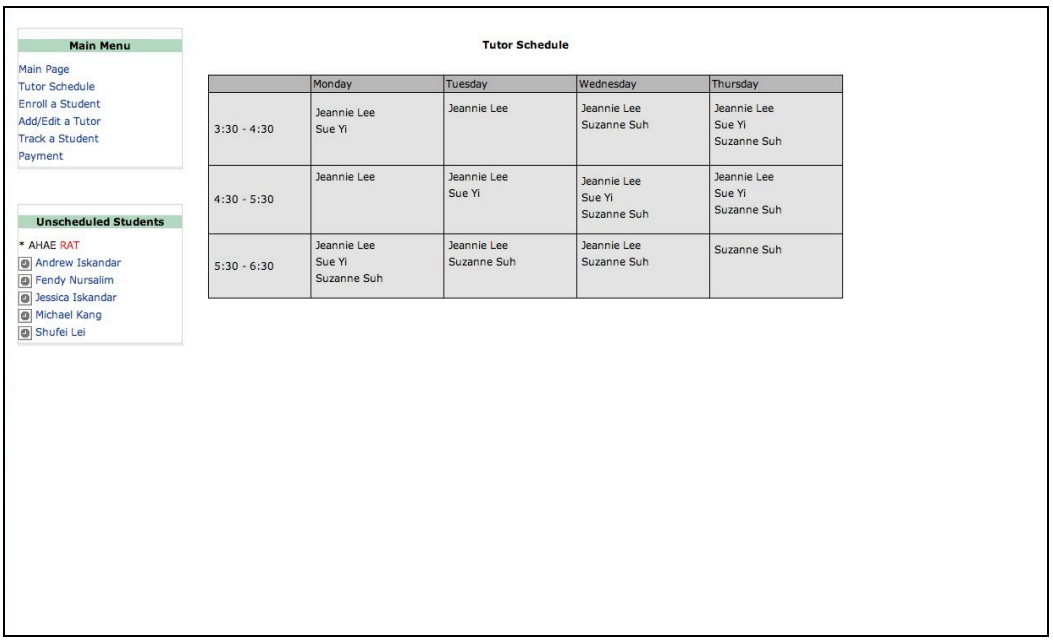
Schedule Overview: Daily View



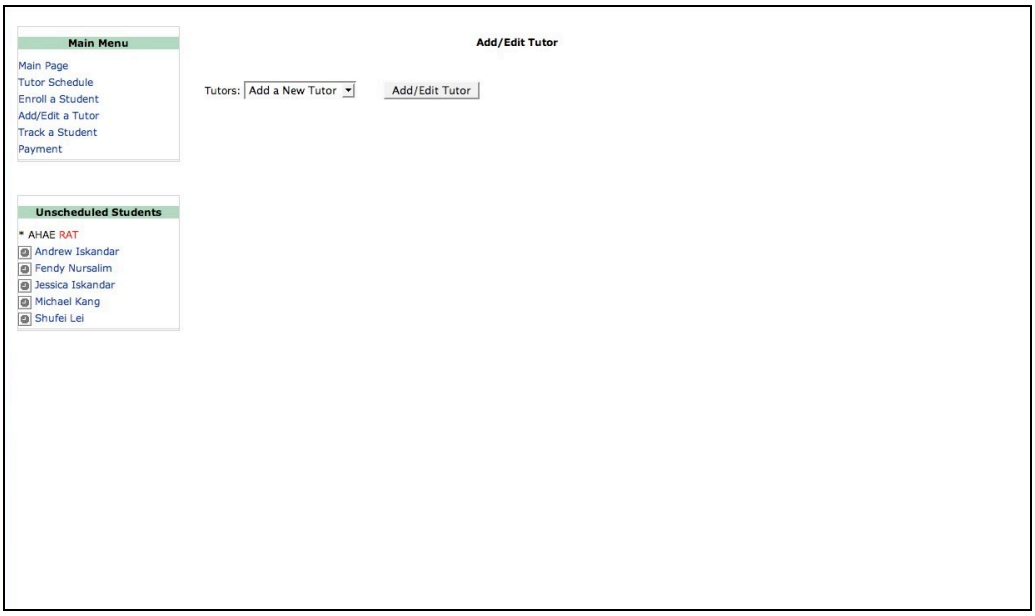
Add A Student



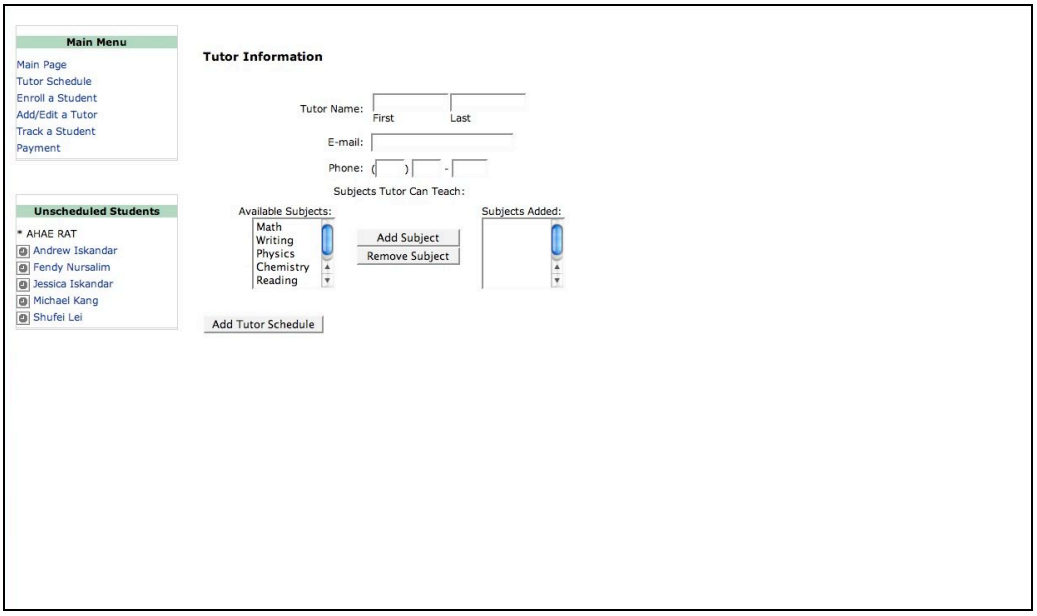
Reschedule A Student



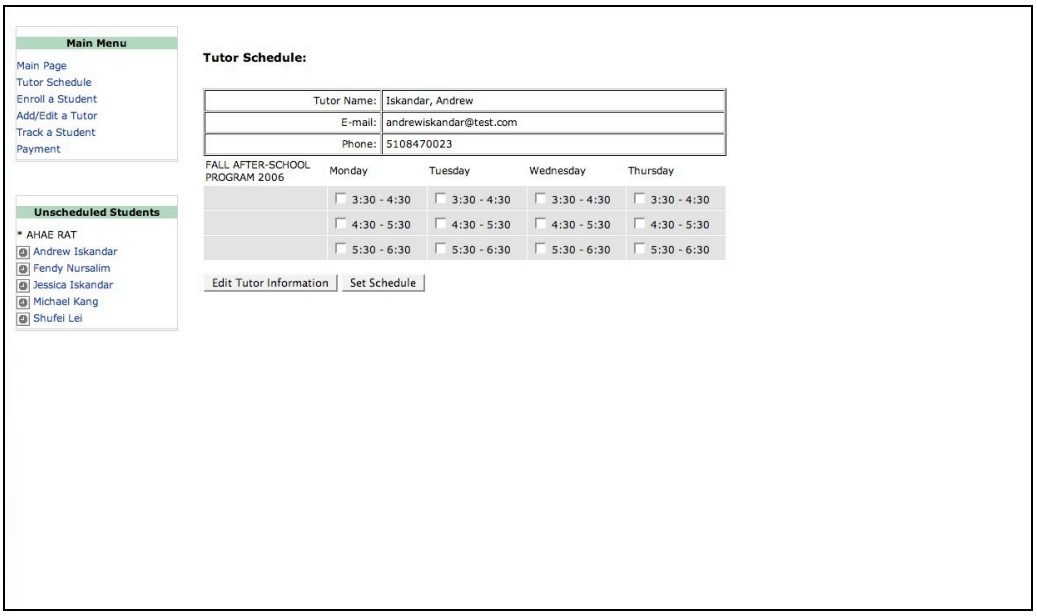
Tutor Schedule



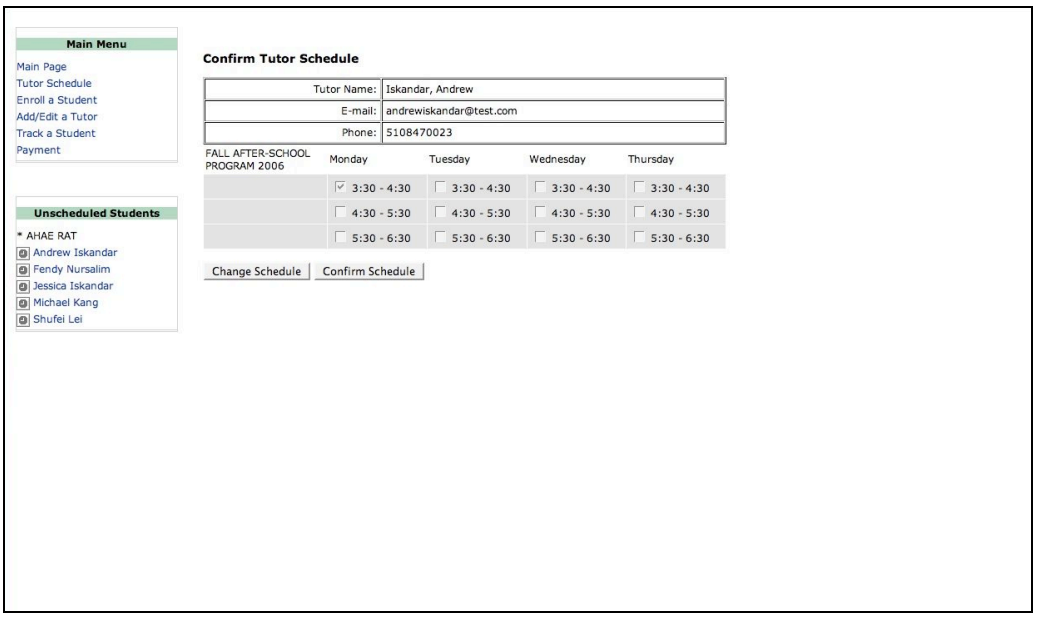
Add/Edit Tutor



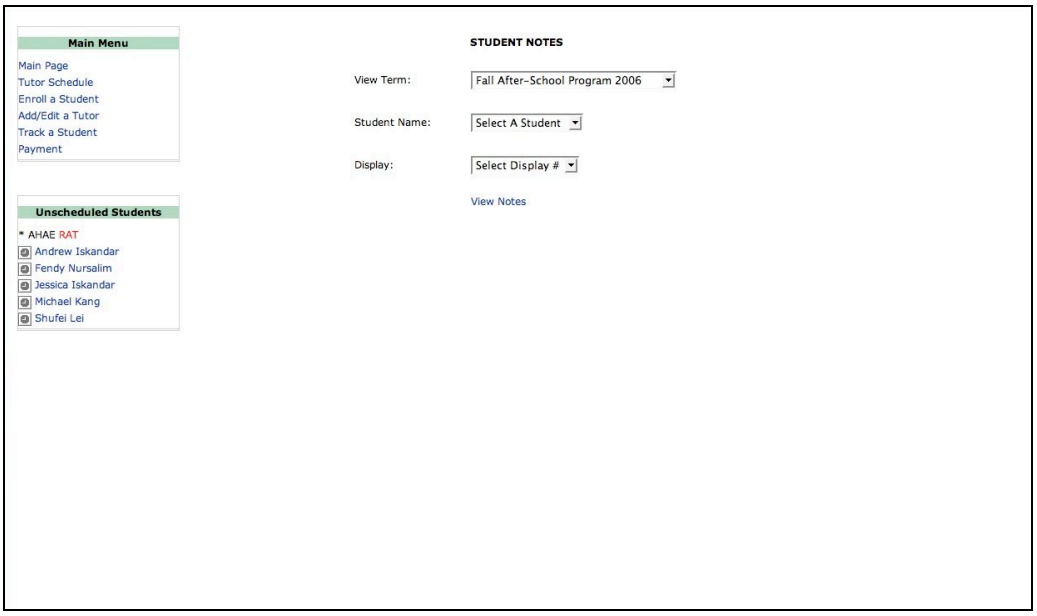
Tutor Information



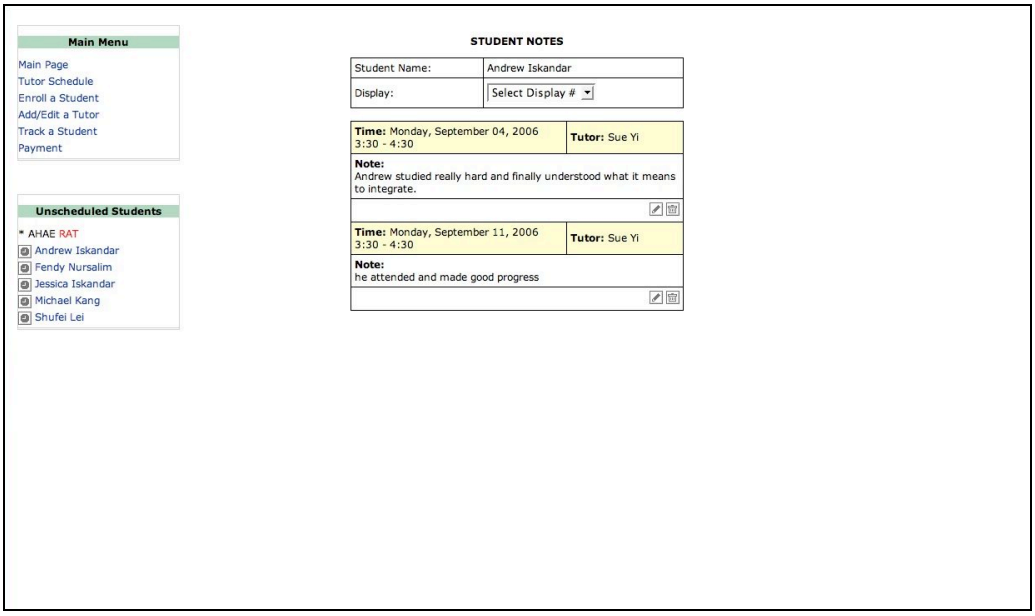
Tutor Schedule



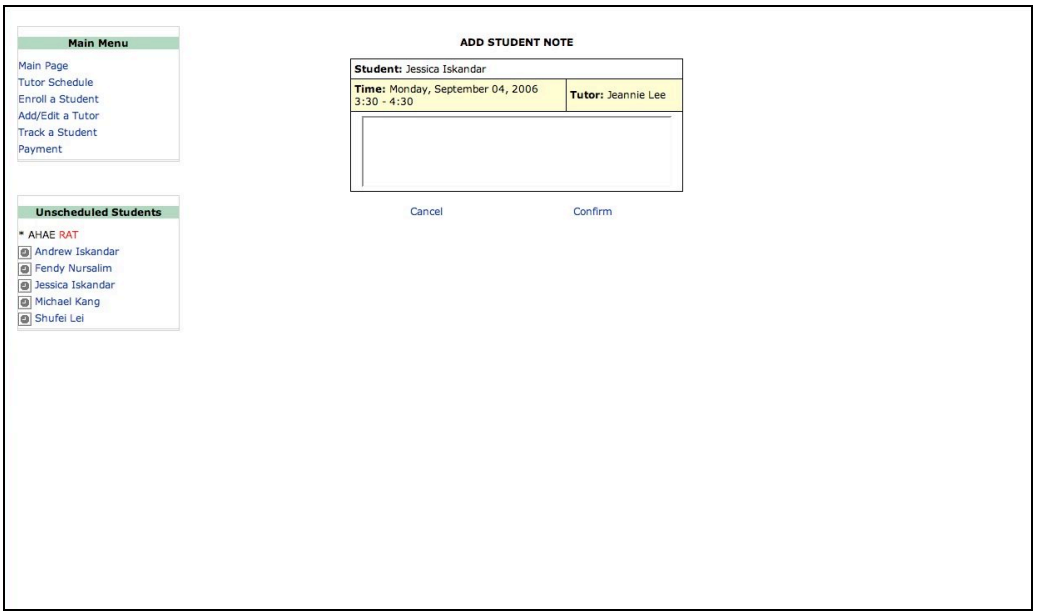
Confirm Tutor Schedule



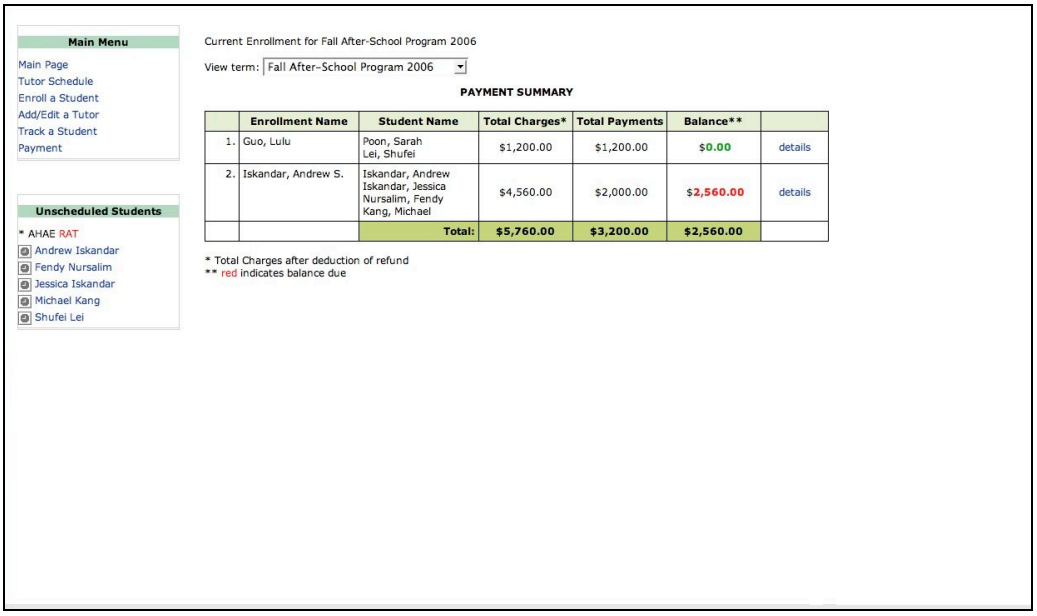
Track A Student



Student Notes



Add Student Note



Payment Summary

Main Menu

Main Page
Tutor Schedule
Enroll a Student
Add/Edit a Tutor
Track a Student
Payment

Unscheduled Students

* AHAE RAT
Andrew Iskandar
Fendy Nursalim
Jessica Iskandar
Michael Kang
Shufei Lei

Detail Payment Information
Ms. Lulu Guo
Back to Payment Summary

Contact Information	
Email:	lulug@sims.berkeley.edu
Address:	South Hall Berkeley, CA 94720
Phone:	(510) 888-8888 (daytime) (510) 888-8888 (evening)
Source:	web (Andrew Iskandar)

Payment Quick View	
(Poon, Sarah)	$400.00
(Lei, Shufei)	$800.00
Total Enrollment Cost: $	1,200.00
Total Payments: $	-1,200.00
Balance Due: $	0.00

Payment History — Add New Payment

Total Enrollment Cost: $1,200.00

Date/Time	Payment	Balance	Type		
2006-04-25 17:09:31	$200.00	$1,000.00	check	Edit	Delete
2006-04-26 11:48:54	$1,000.00	$0.00	check	Edit	Delete

Student Course Schedule

Student	Schedule		
Poon, Sarah (F)	*Enrollment Date 04/25/06* Reserve a Tutor	M 09/04/06 3:30-4:30 W 09/06/06 3:30-4:30 Tu 09/12/06 3:30-4:30 Th 09/14/06 3:30-4:30 M 09/18/06 3:30-4:30 W 09/20/06 3:30-4:30 Tu 09/26/06 4:30-5:30 Th 09/28/06 4:30-5:30 M 10/02/06 3:30-4:30 W 10/04/06 3:30-4:30	$400
Lei, Shufei (M)	**Session A - Alameda** (09/04/06 - 09/29/06) Assisted Homework & Academic Enrichment Assisted Homework & Academic Enrichment	M 3:30-4:30 W 4:30-5:30	$100 $100

Payment Detail

Main Menu

Main Page
Tutor Schedule
Enroll a Student
Add/Edit a Tutor
Track a Student
Payment

Unscheduled Students

* AHAE RAT
Andrew Iskandar
Fendy Nursalim
Jessica Iskandar
Michael Kang
Shufei Lei

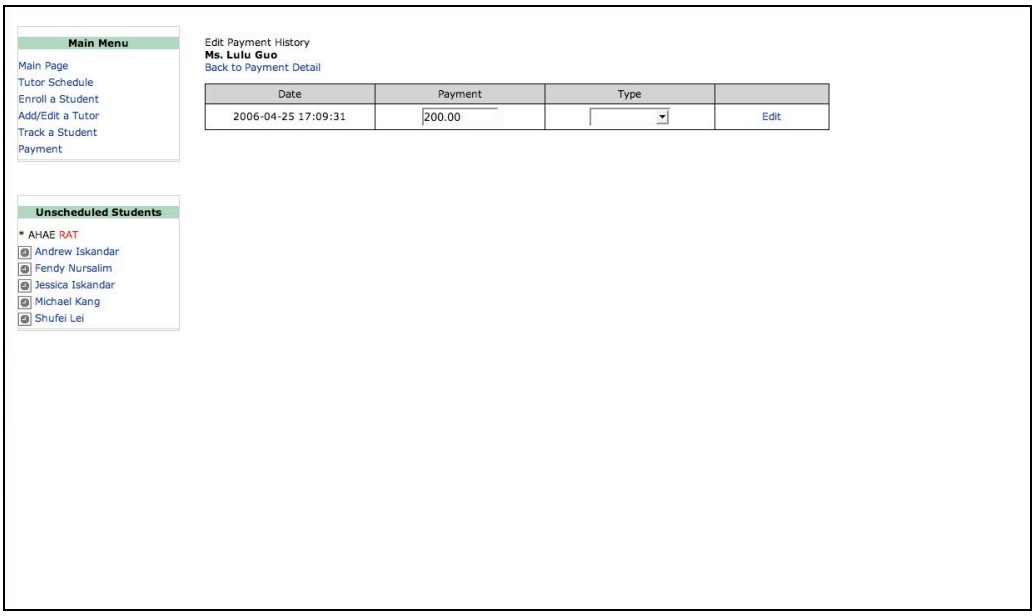
Add New Payment
Ms. Lulu Guo
Back to Payment Detail

Contact Information	
Email:	lulug@sims.berkeley.edu
Address:	South Hall Berkeley, CA 94720
Phone:	(510) 888-8888 (daytime) (510) 888-8888 (evening)
Source:	web (Andrew Iskandar)

Payment Quick View	
(Poon, Sarah)	$400.00
(Lei, Shufei)	$800.00
Total Enrollment Cost: $	1,200.00
Total Payments: $	-1,200.00
Balance Due: $	0.00

Payment Amount	Type	Cleared	
100	check credit card refund		Add

Add Payment



Edit Payment

APPENDIX E: User Interview / Test Notes

1) First Round of User Testing (February 2, 2005)

Task: add a student

- A little hesitant about what to do.
- Looking at the left and right panels back and forth.
- The right side calendar seems to contain too much information.
- First clicked on the Add a Student link in the left panel, then clicked on the one in the calendar.
- Confused by the expanded information in the lower-left panel. Is it time available or unavailable to that client? Is it availability or desired time? Is it one-time or weekly? How often does the student want to come in per week?
- Wants what student's desire schedule instead of availability information.

Task: reschedule a student

- Clicked on Track a Student --> Daily Schedule --> Legend --> Tutor Schedule --> Clock icon
- Suggested Reset and Reschedule are confusing words. Change to Clear and Confirm.
- The three buttons better be Back, Confirm, Clear.
- Add check boxes for One-day change and Permanent change

Task: Delete an class

- [x] icon is not very clear
- The user suggested there should be a confirmation message for deletion

Task: Add/Check a Note

- The user performed the task without difficulty
- The user liked the idea of the student tracking function

Others:

- Enroll a Student should lead to application form
- In the Daily Schedule page, there should be indication to show if it's a one-day schedule or a weekly schedule

2) Second Round of User Testing (March 15, 2006)

Task 1: Add student

- [Daily Page] The user directly looked at the calendar side, ignoring the Unscheduled Student list in the left panel.
- [Daily Page] It was mentioned that weekly view is more useful than daily view (so make weekly view default)
- [Add Student Page], "Subject" was confusing. Was it tutor expertise or the subject the students were taking? We need to differentiate tutor and student information on this page.
- [Add Student Page] The user wanted to assign a student to a particular subject (so subject information is important when scheduling a student)
- [Daily Page] Needed to know if that tutor's availability on the calendar. So color code or put a star to indicate if that tutor already booked a RAT

Take 2: track a student
Successful without much difficulty

Task 3: Payment

- [Payment Page] In payment detail page: change to "Course" and "Session/duration"
- [Add New Payment Page] Delete the Balance column

Others suggestions:

- Consistent wording: add a student/add a tutor in the left panel
- Current Tutor schedule page is not particularly useful. Can be modified to a table with Time slot as row and Subject as column.
- General appearance Okay
- [Payment Page] Have a term selector to filter payments?

3) Third Round of User Testing (April 27, 2006)

Task 1: Schedule a Student

- "How do you know what she is taking?" user was not sure what subjects/session she is signed up for
- "How do you know what to click on?"
- No feedback on adding multiple sessions.
- What does "cancel" mean on "add student note"?
- [Unscheduled Students] does not make sense to user.
- AHAE/RAT legend is not clear. Change colors
- Add [prev/next week] buttons at the bottom of calendar.
- Lists keep collapsing. Keep them open somehow.
- Tooltip of subjects tutor can teach on index.php page
- "Add Student" confirm box should say "Yes" instead of "Ok"
- Add PM to all times.

Task 2: Reschedule a Student

- Clear that clock meant reschedule. Tooltips?
- On "reschedule page" change "confirm" to save or save changes
- What does "reset" do?

Task 3: Add a student note.

- "Add note" page, need to change to "save changes"
- Tooltip should appear above "edit" and "trash" buttons

Task 4: Track student progress

- When changing "display no" should just submit automatically onchange.

Task 5: Find out outstanding balances

- Enrollment name should be changed to parent name
- "details" link should be on parent name.
- "Track a Student" link should be "Student Notes"
- When editing a payment, the payment type isn't automatically selected
- "Cleared" on payment page is no longer needed

Task 6: Add a tutor

- On "Add/Edit Tutor" page. Changing the menu should change the label of the button
- After adding/editing a tutor, go to tutor-schedule page.
- Tutor Information needs to be centered
- Add/Edit tutor needs to be centered.

Misc.

- RAT needs to be a different color. Red is to "alertish"
- Somehow show "payment" info on main page. When they need to pay. (RAT)
- Pro-rating.
- Separate out links on left.php to be more organized and related to each other
- "Add a Student" --> "Schedule a Student"
- "Enroll Student" --> "Add a Student"