Tutorial Agency Management System

...matching every child to the *best* tutor...

A 2007 Master's Final Project

Tutorial Agency Management System (TAMS©) is a web based application which addresses the complex scheduling and matching problem associated with selecting the best tutors for individual children given mutual constraints

School of Information
University of California, Berkeley
102 South Hall
Berkeley, CA 94720-4600
Phone: (510) 642-1464
Fax: (510) 642-5814

Lois Wei & Minakshi Mukherjee
Advised & Guided by Prof. Nancy Van House

EXECUTIVE SUMMARY

Multivariate scheduling under multiple resource constraints has been a challenging problem under various disciplines – operations research, airline schedule, mathematics, sciences and recently under the domain of information management and systems. In the same context, management of student-tutor scheduling for a tutoring agency is a critical activity wherein it is of paramount importance that students are allocated to tutors in a manner that fulfills several and sometimes, stringent requirements.

For a tutorial agency such scheduling represents a task which is not only complex but also demands a large amount of manual time and effort on part of the tutoring agency staff. Further, just like any other complex manual system, it is error prone. Hence in many ways the lack of a proper scheduling system results in loss of revenue, higher costs and more importantly loss of goodwill on part of the tutoring agency.

This report describes the conception, design and development process of the Tutorial Agency Management System (TAMS©), a web based application which has been developed to facilitate the task of student tutor matching and scheduling for tutoring agencies, thereby attempting to alleviate the pains of manual student-tutor matching. TAMS aims to address the scheduling issues and provide an optimal service experience for managing tutor and student information and assist in finding the most suitable tutor for each student. Users will need to enter the constraints and requirements and then just sit back and let TAMS do the matching and suggest the ideal match.

TAMS works within a 3x3 constraint matrix – *subject*, *time* & *day* - for each student and tutor. Broadly, TAMS provides the following features:

- Provides a comprehensive view of all the students' schedule for a week
- Provides a comprehensive view of all tutors' schedule for a week
- Allows administrators to look up student's personal information and tutoring schedule details
- Allows administrators to look up tutor's personal information and tutoring schedule details
- Facilitates administrators to schedule and reschedule tutoring sessions and suggests newly assigned tutors based on the changes
- Provides trend analysis graphs

The design and development of TAMS was done for a real life tutoring agency and followed the User Centered Design process. The primary goal was to design a matching tool for the tutorial agency which can automate the scheduling tasks and which can ultimately be integrated with the current system and database of the agency. Even though TAMS has been developed centering a particular tutoring agency, it can be adjusted to be used by any other tutoring agency that is facing similar problems.

This report describes the process and results for the needs assessment of TAMS, the user interface design process, the database design, the logic of the matching algorithm and the final prototype. In the process of understanding the scheduling system, we became aware of the large number of constraints that can potentially play a part in making a scheduling system extremely complex and time consuming. We have developed a working prototype of the schedule matching tool. This version of TAMS focuses on a limited number of constraints and on one-to-one tutoring scheduling. Given the inherent complex nature of scheduling systems we understand that the TAMS application has many potential areas for improvement.

TABLE OF CONTENTS

Introduction 7
Introduction > About Our Client 9
Project Overview 10
Project Overview > Problem Statement 10
Project Overview > Scope 11
Project Overview > Objectives 11
Overview Of Development Methodology 12
User Needs Assessment 13
User Needs Assessment > Description Of The Current System 13
User Needs Assessment > Semi-Structued Interviews 14
User Needs Assessment > Document Analysis 16
User Needs Assessment > Competitive Analysis 18
User Needs Assessment > Personas 20
User Needs Assessment > Task Analysis 22
User Needs Assessment > Scenarios 22
TAMS Requirements Specifications 24
Tams Requirement Specifications > Feature List 24
Tams Requirement Specifications > System Requirements 25
Economic Feasibility Analysis 25
System Architecture 30
User Interface Design 31
User Interface Design > Initial Sketches & Low-Fi Prototype 31
User Interface Design > First Interactive Prototype 34
User Interface Design > Second Interactive Prototype 37
User Interface Design > Final Prototype 40
Database Development 42
Database Development > E-R Diagram 43
Database Development > Description & Data Dictionary 45
Database Development > Tables Created 45
Database Development > Tables Modified 46
Database Development > Tables Used 47
Database Development > Transactions & Sample Queries 48
Student-Tutor Matching Algorithm 50
Trend Analysis Tools 52
Interaction Flow Diagram 53
Final Application Description 53
Final Application Description > Overview 53
Final Application Description > Students Full Schedule Page 55
Final Application Description > Student Information Page 56
Final Application Description > Student Details Page 57
Final Application Description > Edit Student Schedule Page 58
Final Application Description > Tutors Full Schedule Page 59
Final Application Description > Tutor Information Page 60
Final Application Description > Tutor Details Page 61
Final Application Description > Reports 62
Implementation Technologies & Tools 63
Conclusions 64
Conclusions > Lessons Learnt 64
Conclusions > Scope For Future Work 65

APPENDIX

[A] Low Fi Prototype Interview Questions

[B] Details on Competitive Analysis

[C] User Needs Assessment Questionnaire

[D] Initial Sketches & Low fi Prototype

[E] First Interactive Prototype

[F] Second Interactive Prototype

LIST OF TABLES

- Table 1: Task analysis table 22
- Table 2: Available_Student_Timeslot Table 45
- Table 3: Student_Timeslot Table 46
- Table 4: Tutor_Student_Event Table 47
- Table 5: All student availability query result 49
- Table 6: Student availability query result 49
- Table 7: Requested subjects query result 50
- Table 8: Student-tutor assignment query result 50

LIST OF FIGURES

- Figure 1: Application Form used at the Tutoring Agency 16
- Figure 2: Meeting & discussions with agency staff 16
- Figure 3: A sample Post-It with details of two students 16
- Figure 4: Based on the Post It information, the schedules are shifted around 16
- Figure 5: A comprehensive view of manual scheduling 18
- Figure 6: Calculations for the 'As-Is' option. 27
- Figure 7: Calculations for the Commercial Off The Shelf Application 28
- Figure 8: Calculations for TAMS application 29
- Figure 9: Systems Architecture for TAMS 30
- Figure 10: Tutor/Student Scheduling Matching tool 32
- Figure 11: Display student Info :Subjects to be tutored 33
- Figure 12: Student schedule depicted in the first interactive prototype 35
- Figure 13: Schedule & Auto Matching (First Interactive Prototype) 36
- Figure 14: Revised Sketches 37
- Figure 15: Second Interactive Prototype – Full Schedule Page 38
- Figure 16: Second Interactive Prototype – Tutor Match Page 38
- Figure 17: Final interactive Prototype – Students Full Schedule Page 40
- Figure 18: Final Interactive Prototype – Student Details Page 41
- Figure 19: ER Diagram for TAMS Database 43
- Figure 20: TAMS Database with Relation to Overall Database 44
- Figure 21: Interaction Flow Diagram 53
- Figure 22: TAMS Icons 54
- Figure 23: Students Full Schedule Page 55
- Figure 24: Students Information Page 56
- Figure 25: Students Details Page 57
- Figure 26: Edit Student Schedule Page 58
- Figure 27: Tutor Full Schedule Page 59
- Figure 28: Tutor Information Page 60
- Figure 29: Tutor Details Page 61
- Figure 30: Reports Page 62

ACKNOWLEDGEMENTS

It is difficult to mention specific names, since so many people in the iSchool community has contributed to our project. The effort of all of our professors, and the peer learning from our classmates, have trained, inspired, and allowed us to acquire the skills and the knowledge that made the producing of our final project possible.

We would especially like to thank Professor Nancy Van House for thoughtfully supervising our project step by step, and for cheerfully encouraging us. Professor Ray Larson, Prof. Michael Buckland, and Professor Clifford Lynch for supervising the early research and work done on the project; for providing very valuable feedback on designing a multi-variant constraint system; for inspiring us to think in a greater scope (such as doing comparative analysis on the airline scheduling system). Thank you for your advice and for offering practical solutions on how to addressing the schedule matching problem not just through an information management perspective, but also through a general organizational & service design perspective.

We would also like to thank our classmates from the Professional Skills course, and definitely our instructor Michael Schaffer, for the valuable consulting hours that you have lend us to give us honest feedback on project presentation strategy. We really appreciate Kevin Heard and Gary Lum's effort to patiently provide system administration work, keeping our project prototype up and running on the server, and for restoring our files at desperate times.

We would like to thank the Tutoring agency staff (whose names we could not mention for the sake of privacy) for giving us this opportunity to work on the tutoring matching tool, for sharing their personal experience and the frustration in manually conducting schedule matching, and for offering us many valuable feedback on our design.

ABOUT THE TEAM

Minakshi Mukherjee and Lois Wei, two final year 2007 graduate students of the Information School, University of California worked collaboratively on the Tutorial Agency Management System (TAMS) Application project as part of their final project.

Prof Nancy Van House, Professor, at the School of Information , University of California, Berkeley was the project advisor . Prof. Nancy Van House helped the team with valuable advice and guided the team in their efforts through the entire duration of the project.

LOIS WEI - Lois Wei has a BA degree from UC Berkeley in Computer Science. She worked for a few years on campus as a programmer for the UCWISE research project, a CITRIS funded project that focuses on designing online curriculum for science and engineering courses. She loves reading, cooking, and running. She volunteers for ImpACT, a non profit organization that serves the inner city Oakland kids.

MINAKSHI MUKHERJEE - Minakshi has an undergraduate degree in Computer Science and a Masters degree in Information Science from Birla Institute Of Technology, India. In the Information School, University of California, Berkeley she focused on HCI, Usability, User Interface Design and took a wide number of courses in this area. The TAMS project served as an interesting learning ground for her where she could use her technical as well as her usability and HCI knowledge and further on her skills in the user centered user interface design skills.

Tutorial Agency Management System

A Web Application to Schedule Students and Tutors Optimally

INTRODUCTION

Rebecca, a tutoring center administrator, puts the phone down, both happy and worried at the same time. The tutoring agency was doing far better in student enrollment than what she had hope for and projected as part of the plan. More and more parents liked and appreciated the differentiators her agency was presenting and wanted their sons and daughters to enroll. However, she had not been able to address the vexing problem of scheduling all these incoming students with the tutors she has on staff.

The recent phone call was from a father who wanted his daughter, Katie, to enroll in Math and Science sessions but could only bring her on Thursday and Fridays at 5:00pm. Similarly another new student Chris was waiting to hear on his enrollment request for Reading class on Tuesday and Jessica was waiting to schedule extensive tutoring (more than two sessions a week) in Math.

Every new student who came into the agency meant that Rebecca had to spend considerable hours sitting with all the records trying to work out who should be assigned to whom. It was just not enough to assign any tutor, given that every tutor had a subject constraint and time constraint. On top of all this, the existing students were also free to change their schedules or even subjects based on how they were performing in school and as they progressively moved to higher grades. Someone who was coming in on Tuesdays would place a request to move to Wednesday if they were enrolling in baseball practice sessions plus might need to start classes for higher grade Science and so on. Tutors themselves needed to reschedule based on holidays, vacations and other reasons.

Today Rebecca had to plan for Katie, Chris and Jessica while ensuring that none of the other 35 or so students are impacted and that she is able to pair these children with the tutors who can support them best. On top of this she has to sort out the schedule change request from Ron and Martha – her best Reading and Math tutors.

How does she schedule the three children and two teachers in the most optimum manner?

Management of student-tutor scheduling for a tutoring agency is always a critical activity wherein it is utmost necessary that the students are allocated to tutors in a manner that fulfills several and, sometimes, stringent requirements. Some students need extensive tutoring in a single subject and the agency needs to ensure that the same tutor is able to work across the students' availability. Some students need tutoring across several subjects and need to be scheduled with different tutors across many days. Some tutors are strong in some subjects and may not be as strong in others. Some students have hard constraints on days and timings while others are more flexible. New students enroll and have an impact on classes. Old students have new requirements. New tutors add to the scheduling complexity while old tutors retire or cut down on the number of hours. While this may seem like a manageable problem if we have 5 students and 5 tutors, in reality there are close to 50 students and around 15 tutors for an average tutoring agency.

For a tutorial agency, such scheduling represents a task which is not only complex, but also demands a large amount of manual time and effort on part of the tutoring agency staff. A simple 3 student – 2 tutor scheduling may take anywhere between 15 minutes to half an hour. Imagine the complexity if one constraint changes or worse still, if the tutoring agency needs to work on a 50 student – 15 tutor scheduling scenario which is the case every so often. The success and reputation of a tutoring agency is directly linked to how well they have been able to pair a student with a tutor and thereby created a high degree of satisfaction and increased the success chances for the student.

Multivariate scheduling under multiple resource constraints has been a challenging problem under various disciplines – operations research, airline schedule, mathematics, sciences and recently under the domain of information management and systems. Collecting constraint information, storing it, analyzing it and finally presenting it effectively are all related to the discipline of information management. A scheduling problem can perhaps still be managed if it is a one time occurrence within limited scenarios, but imagine if the constraints and the variables keep changing –this can easily become a half day job everyday to keep reconciling and updating the schedules. A six man-month effort for an average size tutoring agency every year!

TAMS allowed us to work on something with real life utility in the immediate time frame. We are privileged to work with real clients and have a chance to make a difference. Developing TAMS has given us an opportunity to get involved in every stage of the software application design, rather than just developing a front end or a database. Thus, this has provided us a chance to make practical usage of our two

years of learning in iSchool – something with the potential to touch as many disciplines as possible and not just one small set of courses.

The TAMS application is designed to alleviate the pains of manual student-tutor matching. Such a manual system not only takes substantial amount of time and effort, but is also error prone resulting in loss of tutoring opportunities. TAMS aims to address these issues and provide an optimal service experience for managing tutor and student information and assist in finding the most suitable tutor for each student. Users will need to enter the constraints and requirements and then just sit back and let TAMS do the matching and suggest the ideal match. The application will also provide trend analysis namely for visualizing student enrollment distribution, and tutoring subject distribution.

The design and development of TAMS was done for a real life tutoring agency. For various reasons, the tutoring agency did not want their name to be publicly used as part of this project and we respect their wishes. All the work presented within this report and in developing TAMS was carried out in collaboration with their staff. However, we must add, that even though TAMS has been developed centering on a particular tutoring agency, it can be adjusted to be used by any other tutoring agency that is facing similar problems.

INTRODUCTION > ABOUT OUR CLIENT

Our client is an after-school tutoring agency based out of Alameda, California. They conduct programs that offer continued educational assistance and enrichment to students through after school programs during the regular school year and through summer academic camps. In their after school programs, the tutoring agency provides homework tutoring and customized classes for kids to improve their verbal, reading, writing and math skills.

The agency projects two primary differentiators as compared to other such agencies in the area. The tutors at this agency are mostly current or previous students from UC Berkeley with majors in English, Education, Computer Science, Sociology, Mathematics, Psychology, etc. Therefore the quality of teaching is superior as well as they provide personal insights which help students prepare for entrance into premier national universities. The second differentiator is the promise to select the best tutor for the student and the availability of one-on-one tutoring. While this ensures that the parents can rest assured on the quality and quantity of education, it of course brings in the added complexity of scheduling such study sessions.

In keeping with our client's wishes, we shall not be using their name for this project, but refer to them simply as the Tutoring Agency or TA.

PROJECT OVERVIEW

Having introduced the raison d'être for this project, we will lay out the project outline in this section. At the onset we were fairly clear that with the given amount of time and effort which we had, we would not be in a position to design an elaborate system which might rival an ERP or CRM package for the tutorial agency. We wanted to take care of one problem area - that of scheduling. However, our application needed to be designed to integrate with the existing system. We were also building on top of an application which had been designed in 2006 by two former iSchool graduates for the same tutoring agency. It was an information redesign project aimed at designing a web-based application to enable the tutoring agency staff to easily add, delete, modify, and access the information related to students, courses, schedules, and payment. While it efficiently automates the tasks of information storage, retrieval and updating associated with book keeping, key value add modules like student-tutor scheduling was not implemented and was being executed manually.

PROJECT OVERVIEW > PROBLEM STATEMENT

The Tutoring Agency depends on a manual system of managing its information management and scheduling needs. Needless to say the manual system of information storage and retrieval is highly complex and error prone. Information needs to be inputted and updated on a daily basis in several different documents.

Matching the students to suitable available tutors is a time consuming and painful process in which the staff has to look up the schedules of possibly all the tutors to find which tutor is available at a certain day and time and also able to teach the required subject. This process of looking up schedules has to be repeated each time a new student is registered; a student needs rescheduling or a tutor adjusting their availability.

There are primarily 3 criterions used to decide on a student-tutor pairing – (a) subject, (b) day of the week, and (c) time of the day. In the case of students, the subject refers to the subject they need to be tutored in, and for tutors, the subject refers to the subject they teach.

PROJECT OVERVIEW > SCOPE

On a larger scope, the tutoring agency wants to replace its current manual information management system with an application that automates the entire management of the tutorial agency. Although the earlier information redesign project automated the paper-based administrative tasks to a large extent, many of the tasks still needed to be done manually. Student-tutor matching was one of those cumbersome tasks.

The scope of the project broadly included the following:

- Design and develop the TAMS application focusing on automating the tutoring scheduling
- The design should be accomplished in a manner that future integration with the existing system is possible
- TAMS will work with a 3x3 constraint table which will include the subject, day and time for every student and tutor. We realize that in a scheduling system there may be a large number of constraints to consider which in a way increases the complexity of the system. However, our objective is to design and implement a system that can be put to use by our client. So we limited our constraints to only those that are meaningful in the current system of the tutoring agency

The out-of-scope activities included:

a) Integrating TAMS with the existing system as a result of the earlier information redesign project

b) Linking TAMS to an active calendar

c) TAMS also assumes that the registration information of new students and tutors and the maintenance of this data is being done by the current tutoring agency system

d) Tutoring Agency business trend report & graph dynamic generation

PROJECT OVERVIEW > OBJECTIVES

The overall objective of this project is to design and implement an ideal front end application that provides an optimal service experience for managing tutor and student information and assists in finding the most suitable tutor for each student. Users will need to enter the

constraints and requirements and then just sit back and let TAMS do the matching and suggest the ideal match. The application will also provide trend analysis namely for visualizing student enrollment distribution and tutoring subject distribution. The key application objectives can be mentioned as:

a) TAMS should be *user friendly*. It should have a simple and intuitive interface

b) TAMS should be *scalable* with respect to the number of students and tutors. Since the time when we started working on this project, our client have scaled down the after school classes temporarily to focus more on summer camp activities. However they plan to ramp up the numbers as the summer draws to a close and would need TAMS to support the higher enrollments

OVERVIEW OF DEVELOPMENT METHODOLOGY

User centered design methodology was followed in the research, design and implementation of TAMS. Although in many situations we had to customize the different guidelines to suit the requirements of the specific project, the overall project development can be broadly divided into three phases:

a) Needs assessment and Requirements gathering

b) User Interface Prototype development

c) Database design and implementation using PHP and MySQL

We followed an iterative approach for the prototype development by conducting user testing with our clients after each version and making changes to the next version according to the results and feedback from the testing.

The following sections describe each of the phases and our decision making process in greater detail.

USER NEEDS ASSESSMENT

User needs assessment is a crucial part of the entire project. Understanding the end users needs, the current process, the problems and pain areas with the current system and the expectations of our client from the automated system is an integral part of the design and development of TAMS.

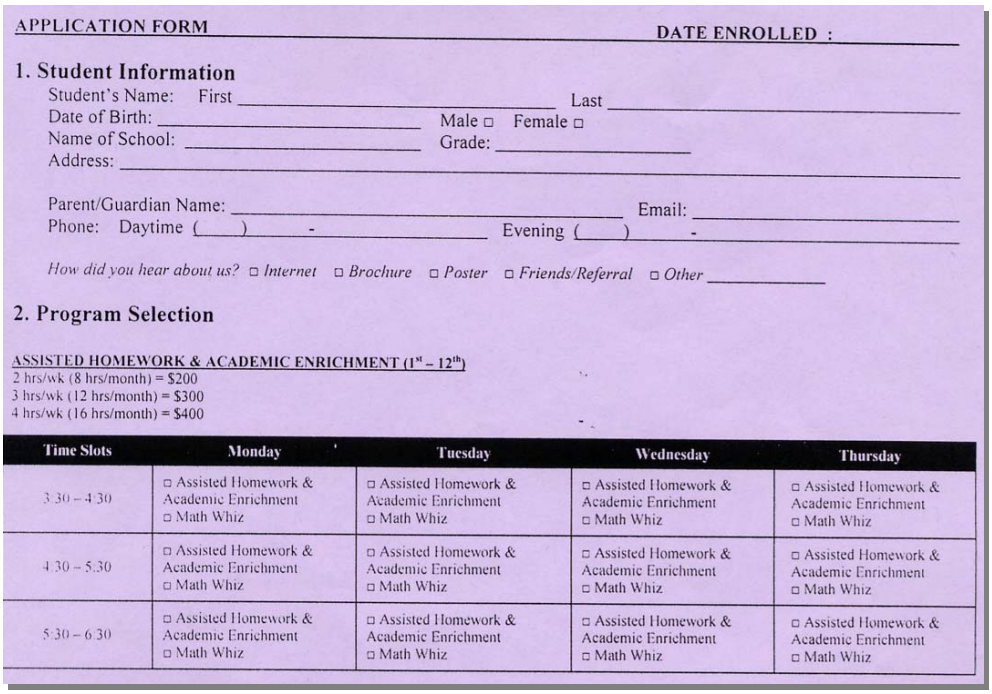
In our quest to understand the needs of the client and the current working of the tutorial agency, we met with our client and also traveled to the tutorial agency on many occasions and used a number of methods for user needs assessment which include:

a) Semi-structured interviews

b) Analysis of related documents and current system

c) Competitive analysis

USER NEEDS ASSESSMENT > DESCRIPTION OF THE CURRENT SYSTEM



APPLICATION FORM DATE ENROLLED :

1. Student Information

Student's Name: First _____ Last _____
Date of Birth: _____ Male ☐ Female ☐
Name of School: _____ Grade: _____
Address: _____

Parent/Guardian Name: _____ Email: _____
Phone: Daytime (____) _____ - _____ Evening (____) _____ - _____

How did you hear about us? ☐ *Internet* ☐ *Brochure* ☐ *Poster* ☐ *Friends/Referral* ☐ *Other* _____

2. Program Selection

ASSISTED HOMEWORK & ACADEMIC ENRICHMENT (1st – 12th)
2 hrs/wk (8 hrs/month) = $200
3 hrs/wk (12 hrs/month) = $300
4 hrs/wk (16 hrs/month) = $400

Time Slots	Monday	Tuesday	Wednesday	Thursday
3:30 – 4:30	☐ Assisted Homework & Academic Enrichment ☐ Math Whiz	☐ Assisted Homework & Academic Enrichment ☐ Math Whiz	☐ Assisted Homework & Academic Enrichment ☐ Math Whiz	☐ Assisted Homework & Academic Enrichment ☐ Math Whiz
4:30 – 5:30	☐ Assisted Homework & Academic Enrichment ☐ Math Whiz	☐ Assisted Homework & Academic Enrichment ☐ Math Whiz	☐ Assisted Homework & Academic Enrichment ☐ Math Whiz	☐ Assisted Homework & Academic Enrichment ☐ Math Whiz
5:30 – 6:30	☐ Assisted Homework & Academic Enrichment ☐ Math Whiz	☐ Assisted Homework & Academic Enrichment ☐ Math Whiz	☐ Assisted Homework & Academic Enrichment ☐ Math Whiz	☐ Assisted Homework & Academic Enrichment ☐ Math Whiz

• Figure 1: Application Form used at the Tutoring Agency

We went onsite to the tutoring agency a couple of times to understand the current scheduling system. During these visits we had detailed discussions and conducted semi-structured interviews with the administrative staff. During these discussions they explained in great detail about the entire scheduling process and talked about the pain points. We recorded the conversations with the permission of the participants, took notes and photographs. Some of the photographs have been attached in the Appendix section. The main points that stood out after the analysis of the current scheduling system are:

- Almost the entire scheduling task was done manually.

- Capturing the rapidly changing information on paper and conveying it to the rest of the tutors effectively was a highly complex task in conjunction with the administrative tasks.

- Last moment cancellations by students were a common occurrence. In many cases the students would be rescheduled to a new time which somehow failed to be registered on paper and the tutor was not notified. The student turns up in the new scheduled time and the tutor is not present. The tutoring agency administrative staffs have to improvise and try their best in those situations to accommodate the student for that day.

- A wide assortment of Microsoft Excel spreadsheets on individual laptops, paper based schedules pasted on walls, changes noted on the papers / sticky notes and pasted on walls or on the schedules so that the next administrative staff taking up the schedule is updated about the previous scheduling changes.

- At times there are so many scribbled notes, highlights, sticky notes, arrows all done on the same schedule put on the wall that it became difficult to make out which information was deleted and what was changed.

An automated system of scheduling match, schedule updating and a way to convey the new changes in the schedule to the tutors and the administrative staff would be tremendously appreciated.

USER NEEDS ASSESSMENT > SEMI-STRUCTUED INTERVIEWS

• Figure 2: Meeting & discussions with agency staff

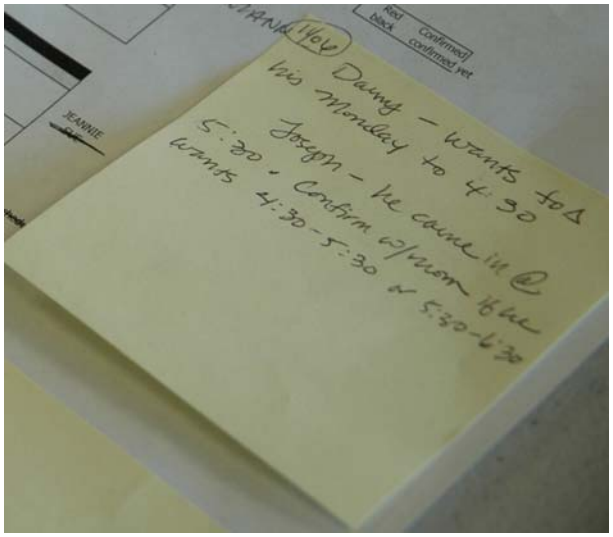
As part of the needs assessment and requirement gathering process, we set up a number of meetings with the tutoring agency management and administrative staff to understand the current working process and identify their pain areas. There are currently three staff members who conduct the management and administrative tasks in the agency. We met all the three of the members in two separate occasions. Of the two meetings for the needs assessment, the first meeting was more informal in which we tried to understand the extent to which the previous project had been used, their general size of students and tutors, their technical support structure and discussed in length about the current manual system of information management, business

processes, marketing methods and services provided. With the permission of the interviewees, we recorded the discussion for future use. This informal discussion gave us the basic idea about the size, needs of the agency, general managerial tasks done etc.

On our second meeting with the tutoring agency staff, we conducted a semi-structured interview on them. Our first informal meeting had given us a preliminary idea about the organization which helped us frame the questions for the interview. The questionnaire is provided in Appendix [C]. A few days before conducting the interviews we sent the list of questions that we were looking at covering in the interview to the three administrative staff. Having taken a look at the questions prior to the actual interview enabled our client staff to be aware of the areas that we were looking at focusing in the interview and enabled us to keep the interview focused. In addition, they were more prepared with the discussion points that they wanted to talk in the interview and also kept the related documents that they wanted to share with us ready before the interview.

The interviews were conducted in a group as that was preferred by the staff for a number of reasons:

- Group interviewing could bring out facts that could be overlooked and forgotten in one-to-one interviews.
- Facilitate more open discussions about different views and problem areas and identify the common major pain areas that we would focus on.
- Group interviews deemed more time and resource efficient.



- Figure 3: A sample Post-It with details of two students

- Figure 4: Based on the Post It information, the schedules are shifted around

The above two figures give an example of two documents which are used extensively for managing the schedules and all changes – the 'Post It' and the 'Weekly Planner'. For example, the Post It says -

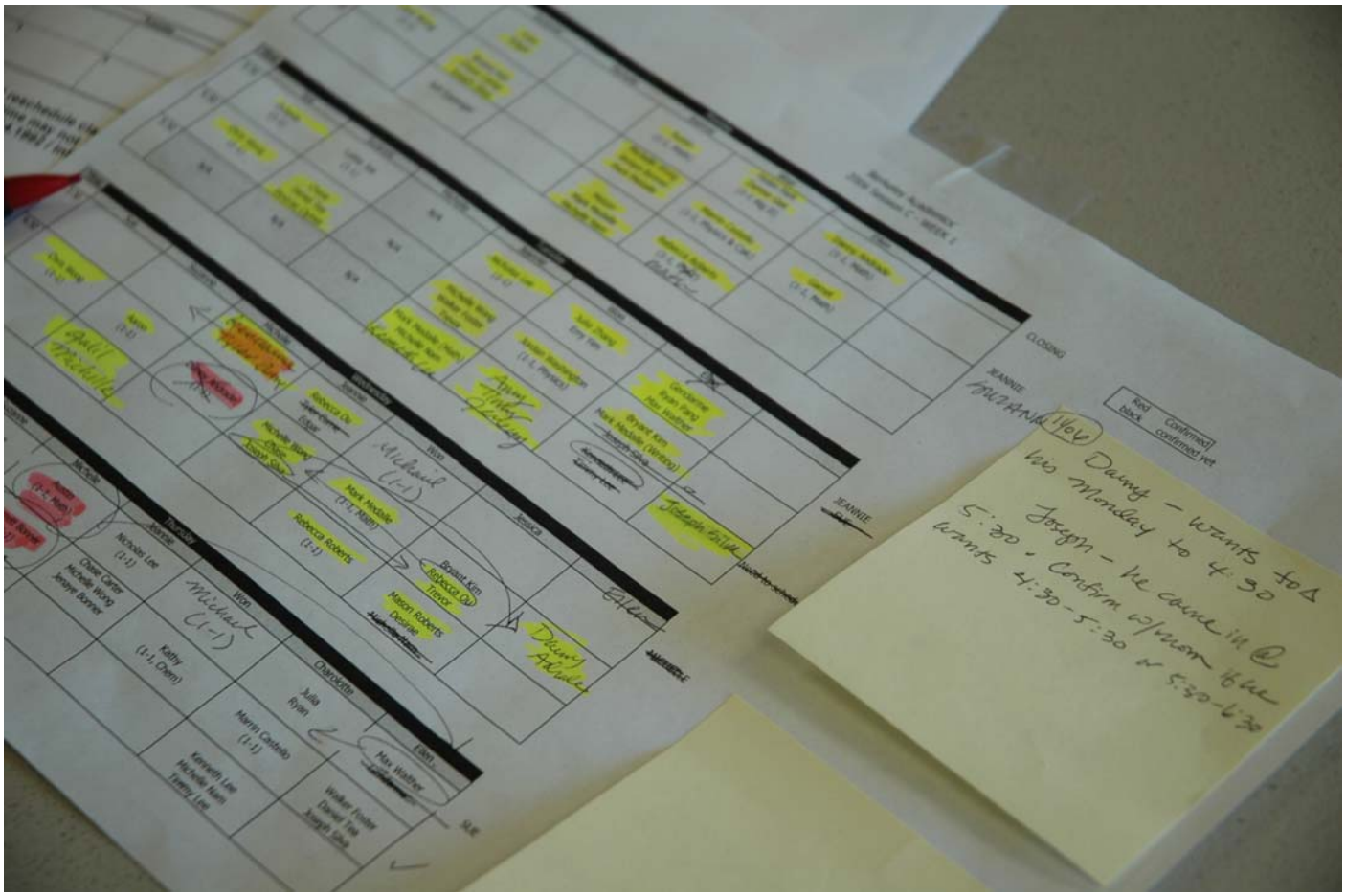
> Danny - wants to Δ his Monday to 4:30 // Joseph – he came in @ 5:30. Confirm w/mom if he wants 4:30 – 5:30 or 5:30 – 6:30"

Now based on this information, several changes take place for Wednesday including rescheduling Danny, bringing in substitute teacher, Ellen, and at least 3-4 further changes in schedule. All this is done manually using several Post It notes around the weekly planner.

To understand the current methods of information entry and updating, we analyzed the relevant documents. The tutoring agency uses a mix of Microsoft Excel spreadsheets on individual computers with documents that they have to update on a daily basis, such as payment information, and schedule changes. They use Post It notes a lot for any additional information that cannot be input in the documents as there are no defined categories for them.

As part of this process, we also aimed to understand the existing tutoring operations and the information redesign project, which is built based on the existing information management system of the tutoring agency. We believe by gaining a clear understanding in these areas we will establish a strong foundation for building an information management tool that best suit our clients' needs and also avoid duplicating efforts on the work already done.

The following is a photo taken during the interview to serve as a reminder of how the current tutoring operations are conducted at the tutoring agency:



• Figure 5: A comprehensive view of manual scheduling

We reviewed the information redesign project documents, and played with the final prototype. We also communicated with the project team to understand the backend database schema and the implementation details of how the prototype was built.

USER NEEDS ASSESSMENT > COMPETITIVE ANALYSIS

In User Centered design process, prototyping is an important part. The best way to get good prototype ideas and also understand the usability of various approaches is to analyze related products which are already available. This technique is customarily known as 'competitive analysis.' During our own research on the available alternatives, we focused on the general domain of scheduling applications and looked at various feature sets being offered as well as interface designs used. Given that we could not really find a direct competitor, we would term our research more of a 'comparative' as opposed to 'competitive' analysis.

As part of the competitive analysis, we analyzed existing products that had some functions similar to our project goals or provided matching services in other capacities. We also read research papers related to our project to get further our knowledge about how to manage multivariate scheduling systems. Our intent were to view the existing ideas as something that we could learn from, improve upon and incorporate in our project rather than something to 'compete' against. We understand that good and useful ideas can emerge from systems that may seem very different than our project. So instead of narrowly looking at student/tutor scheduling applications, we attempted to look at scheduling and matching systems which catered to a wider audience (namely airline scheduling project, doctor scheduling applications, projects connecting students with specific activities or projects on campus etc) in addition to the tutor/student scheduling systems.

We divided the websites/ projects/ applications that we looked at into two broad sections which focused on different aspects:

- Existing Scheduling applications available in the market
- Projects or Papers that deal with similar multivariate scheduling systems

Each of these applications had a variety of features but we focused our analysis to only the features or pages that had similar functionalities to the TAMS application. We noted the interesting and useful features, features that were a bit confusing or cumbersome and also compared its functionality with TAMS.

Scheduling Software

a) LANTIV Timetabler 6 - *www.lantiv.com*

b) Appointment Plus Tutor Scheduling - *www.appointment-plus.com*

c) Tutor Scheduling System - *www.interstellarsoftware.com*

d) Physician Scheduler® 6.0 - *www.physicianscheduler.com*

Scheduling Projects

- Airline reservation
- Campus connect

Take Aways From The Competitive Analysis

There are many scheduling and timetabling applications and software currently available. These softwares have different features, capabilities and price tags. We also tried to find through our competitive analysis whether any of these currently available software could be a viable alternative to TAMS and can cater to the needs of our client with some customization. We found that although many of them had some really useful features and interfaces, we did not find any software that had all the features similar to the goals of TAMS. Our goal was to design TAMS such that it could be integrated with the rest of the tutoring agency components and would fulfill the immediate scheduling needs of the agency. The competitive analysis helped us in identifying various interface methods and pick out the features we found would best fit our needs keeping in mind the goals of TAMS and the user needs from the Needs analysis results.

Appendix [B] gives a detailed description of each of the applications, projects and papers analyzed as part of the competitive analysis.

USER NEEDS ASSESSMENT > PERSONAS

User 1: Rebecca - Administrator / Scheduler

Rebecca is a 35 years old working mother. She is married and has 2 kids. She graduated from UC Berkeley as a Political Economy of Industrial Societies major. She has worked 10 years for Deloitte & Touche in accounting and finance. Since she loves kids, and has a passion for teaching, she decided to change her career to start a tutoring agency with a few of her friends. She is currently the director of the tutoring agency. She is extremely enthusiastic about motivating students to reach their highest academic goals.

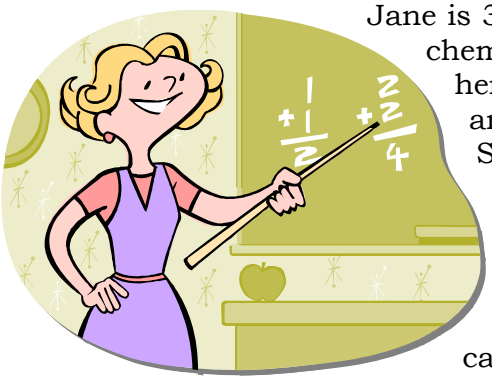
At the tutoring agency, Rebecca is responsible for keeping the tutoring agency running, planning future directions, keeping track of budget, and recruiting new students and tutors. Since the tutoring agency is small, she also works on tutoring scheduling on a regular basis.

Rebecca's goals for using TAMS:

- To improve the over all tutoring scheduling efficiency of the administrators at the tutoring agency.
- To reduce the headache and the stress that are frequently experienced by the administrators surrounding tutoring scheduling complexities currently done using Excel spreadsheet in a decentralized manner.

- Viewing the overall tutoring schedule of the agency for the current week.
- Viewing the business growth trend at the tutoring agency, such as the student enrollment patterns.

User 2: Jane - Scheduler / Tutor



Jane is 32 years old. She graduated from UC Berkeley with a major in chemistry. She just recently celebrated one year old birthday for her daughter. She wants to spend more time with her family, and to volunteer as a mentor for a non-profit agency (ImpACT). So she decided to quit her full time job as a chemist to work part time for the tutoring agency that her college friend Rebecca has started.

Jane's responsibilities at the tutoring agency are: day to day tutoring scheduling and re-scheduling; answering incoming calls; respond to email requests; taking care of student registration. She also works as a tutor in chemistry, physics, and math.

Jane's goals for using TAMS:

- To schedule tutoring sessions efficiently.
- To quickly detect any problems with tutoring scheduling (e.g. there are no physics tutors on Wednesday afternoon, yet this student wants to schedule a physics tutoring session at that time), so can negotiate tutoring availability with students, or recruit more tutors.
- To easily adjust student tutor match due to session rescheduling and cancellation requests.
- To keep track of and view current tutoring schedule.

USER NEEDS ASSESSMENT > TASK ANALYSIS

	Rebecca	Jane
Store Information Required for Automatic Matching of Students to Tutors		
Store tutoring competency of all tutors *	H	H
Store all students' schedule of availability *	H	H
Store subjects that needs tutoring for all students *	H	H
Store tutor schedule of availability *	H	H
Viewing Tutor / Student Information		
View tutor detail *	M	H
View a list of all students	H	H
View a list of all tutors	H	H
View student detail *	M	M
View Overall Schedule		
View tutoring schedule of the current week *	H	H
Search Tutor / Student		
Search for a tutor by keyword	M	M
Search for a student by keyword	M	M
Student / Tutor Matching		
Automatic matching of tutor to students *	H	H
Allow rematch of tutor-student tutoring session due to rescheduling or cancellation *	H	H
Viewing the Current Business Status and Growth Trend		
Student enrollment growth and pattern	H	L
Student by subject enrollment growth and pattern	H	L
Notations: **H** = High Importance; **M** = Medium Importance; **L** = Low Importance **Field with an *** = necessary features;		

• Table 1: Task analysis table

USER NEEDS ASSESSMENT > SCENARIOS

Scenario 1: Use the tutor student matching tool for rescheduling

Jane just got a phone call from Laura's mom. Laura is a middle school student who comes regularly to the tutoring agency on Monday afternoons to receive math and writing tutoring. Laura's mom told Jane that Laura has a dentist appointment this coming Monday afternoon to pull out her wisdom teeth. So she was wondering if Laura could reschedule her math tutoring to Wednesday afternoon instead.

Using TAMS, and within a minutes, Jane adjusted Laura's tutoring availability day and time information. Jane then reloaded Laura's personal tutoring schedule, and she noticed that TAMS has just re-matched Laura to the math and writing tutors on Wednesday. Jane then informs Laura's mom over the phone that it is okay for Laura to come in on Wednesdays, and assures her that the tutors on Wednesdays that has been assigned to Laura are two nice college students.

Scenario 2: Viewing Student Detail and Tutoring History

Kyle Li is a third grader. He has been receiving tutoring on reading and writing since April of last year when he immigrated with his parents from Hong Kong to the U.S.A. Recently, Kyle's mom called Jane, and excitedly told her that Kyle's school teacher has been impressed by Kyle's quick speed in learning English, and has recommended that Kyle be moved to a regular English class rather than to continue with the ESL program.

Kyle's mom would like to get the email address of all the tutors that have tutored Kyle on reading and writing, and write a personal thank you note to them, and invite them to Kyle's birthday party next weekend.

To obtain such a list of the email addresses, Jane found Kyle's personal profile from TAMS, and then located his tutoring history which indicates all the tutors that has tutored Kyle. Jane then proceeded with locating all the email addresses of the tutors from the tutor profiles.

Scenario 3: Viewing a Tutor's schedule

Rebecca was informed about this educational conference in Monterey for the next three days. To decide on whether to attend this conference, she wants to know what her current tutoring schedule is like for this week. Or more specifically, which students she is assigned to tutor. She has recently been tutoring two boys from Mexico. Since she speaks Spanish fluently, she doesn't want to miss out on tutoring them if they are still planning on coming in to receive tutoring this week.

She finds her name in the list of tutor's page, and selects her own personal profile page, she notices that on her schedule for this week, she is still assigned to the two Spanish speaking boys. But she is tutoring them in the morning, so she decided that she'll drive down to Monterey in the afternoon of that day.

TAMS REQUIREMENTS SPECIFICATIONS

As a summation of the user needs assessment process and in keeping with the project scope and objectives, we drew up a list of features which TAMS application should support as well as general solution and system requirements for the application to work effectively. These have been listed in this section.

TAMS REQUIREMENT SPECIFICATIONS > FEATURE LIST

1. TAMS access would be controlled
    a. Users would need a login and password to access the application
    b. A 'remember me' option should be available
    c. 'Log out' should be available at all points within the application
2. There will be 3 sections
    a. Students
    b. Tutors
    c. Reports
3. Automatically pair up students with tutors based on:
    a. Tutor/student times of availability
    b. Tutoring subject
    c. Keeping the same tutor with the same student as much as possible
4. Can perform "re-match" or update the tutoring schedule based on:
    a. Tutoring cancellation, rescheduling
    b. New student drop in
5. Can let the tutoring agency staff adjust the schedule manually

6. Centralized and web-based
7. The following information should be displayed:
    a. Students
        i. Subjects students need tutoring on
        ii. Availability of students for tutoring
        iii. Total number of sessions so far
        iv. Who has tutored this student
        v. Student progress report based on tutors' notes after each session
    b. Tutors
        i. Tutors' subject of specialty
        ii. Tutors' hours of availability

TAMS REQUIREMENT SPECIFICATIONS > SYSTEM REQUIREMENTS

1. The application should be supported on Microsoft Internet Explorer 6.0 or higher, and Mozilla Firefox.
2. The minimum supported screen resolution (for a horizontal fit) is 1024 pixel.

ECONOMIC FEASIBILITY ANALYSIS

A key component of any development project is to figure out in advance if the project and the developed software make sense from an economic point of view. Therefore if it takes ten thousand dollars to build the software when an equally useful version is available in the market at one thousand, then it does not make sense to build it. Similarly if the cost of developing far outweighs the cost of doing without it, then also undertaking such a project does not make sense. This process is called the Economic Feasibility Analysis or EFA.

Since our main interest was in seeing the TAMS application deployed at the tutoring agency, it was critical that we understand if such an application made economic sense for the tutoring agency.

We understood that the tutoring agency had a real pain point with respect to scheduling students and tutors effectively. There were three ways in which this pain could be addressed:

A. By throwing man power at the problem. This means that keep using the same paper based system, but ensure that enough man hours are dedicated into getting this done.

B. Buy a scheduling application from the market. During our competitive research we did not find any software which fit the problem statement 100% but there were scheduling solutions for schools, classrooms, resources and so on. We believe some of these applications can also be modified to address this problem, but then that would mean extra effort and customization on part of the company selling the application

C. The third approach is developing TAMS. One key assumption we have made is that the effort going into developing TAMS by the students is complementary since it is part of the curriculum. In technical sense though this effort has a monetary value and should probably be included, however in real life scenario, the tutoring agency is paying nothing to us.

The EFA is based on the following points:

1) Two types of money outflow is considered – immediate (associated with developing or acquiring the solution) and over the next 3 years (associated with maintaining the application and operational cost)

2) The benefits of having the application are two folds – it frees up tutor time (tutors in the agency are doubling up as administrative staff as well) and reduces the error associated with the manual system. (The time and difficulty of manual scheduling task can increase manifold as the number of tutors and students increases).

3) The initial cost comprises of buying or developing the software, training, effort spent in choosing the software and paying for customization and the effort spent by the tutors in working with the project team for usability studies, solution discussion and so on.

4) The operation costs consist of things like labor, consumables, hardware (PC), application maintenance, hosting service and so on.

5) The operational costs have been considered for the next 3 years and the NPV has been calculated at 10% IRR.

6) Further assumptions have also been made which are presented along with the calculations.

Option A: Allow the current system to continue

Budget - Option A

As Is - Allow the current manual system to continue

Benefits		NPV
There are no monetary benefits to using the manual system apart from getting the job done. The benefits would be comparable and tangible in other options when we compare them to the manual system	$	-
Total Benefit	**$**	**-**

Development/Solution Acquisition Costs		
Since there is no development or purchase of packaged software involved hence there is no such costs	$	-
Total Development/Solution Acquisition Cost	**$**	**-**

Operational Costs			Year 0	Year 1	Year 2
Labour cost ($26.81ph x 4hrs pd)	$	79,895.79	$ 27,880.00	$ 29,274.00	$ 30,737.70
PC (1 part time for this purpose 10%) depreciation on purchase	$	84.35	$30.83	$30.83	$30.83
Stationary & consumables	$	286.57	$ 100.00	$ 105.00	$ 110.25
Office space (30% allocated for this task)	$	1,728.00	$ 576.00	$ 633.60	$ 696.96
Error due to manual system (1/2 hr per day - 12.5% error rate)	$	9,986.97	$ 3,485.00	$ 3,659.25	$ 3,842.21
Total operational costs	**$**	**91,981.67**	$ 28,586.83	$ 30,043.43	$ 31,575.74

Assumptions			
Teacher's Salary	$ 26.81	ph	*Salary.com
IRR	10%		
Cost of PC	$ 1,000.00		
Salvage value of PC	$ 75.00		
Office space	64	sf	
Rate	$ 2.50	psf-pm	

Net Gain/(Loss) **(91,981.67)**

Cost of Error can be treated in different ways - in this case it has been treated simplistically as the cost of rework. Opportunity cost or loss of goodwill has not been factored in, though they do account for high intangible costs

• Figure 6: Calculations for the 'As-Is' option.

Based on the calculations, if the current system is allowed to continue, the net loss for the tutoring agency is approx. $92,000 as the present value for the next 3 years. Most of this money is due to the effort spent by the tutors in working on the scheduling problem on a daily basis.

Option B: Allow the current system to continue

Budget - Option B

COTS - Purchase a "Commercial Off The Shelf" Software Package

Benefits	NPV	Year 0	Year 1	Year 2	Assumptions			
Revenue through utilization of Teaching Resource (for 1.5 hrs extra per day)	$ 50,293.08	$ 17,550.00	$ 18,427.50	$ 19,348.88	Min ph rate per resource	$ 45.00	ph	*Tutoring Agency
Revenue through utilization of Teaching Resource (reduction in error to 8%)	$6,035.17	$ 2,106.00	$ 2,211.30	$ 2,321.87				
Total Benefit	**$ 56,328.25**	$ 19,656.00	$ 20,638.80	$ 21,670.74				
Development/Solution Acquisition Costs								
Training (1 day, 4 people)	$ 857.85	$ 857.85	$ -	$ -				
Effort in solution identification & purchase process (2 weeks, 2 ppl, 20%)	$ 857.85	$ 857.85	$ -	$ -				
Customization of the software (2 days, 1 consultant)	$ 1,200.00	$ 1,200.00	$ -	$ -	Consultant	$ 75.00	ph	
Total Development/Solution Acquisition Cost	**$ 2,915.69**	$ 1,715.69	$ -	$ -				
Operational Costs					**Assumptions**			
On Demand Service from NetSimplicity	$ 8,562.72	$ 2,988.00	$ 3,137.40	$ 3,294.27	Teacher's Salary	$ 26.81	ph	*Salary.com
Labour cost ($26.81ph x 2.5hr pd)	$ 49,934.87	$ 17,425.00	$ 18,296.25	$ 19,211.06	IRR	10%		
PC (1 part time for this purpose 20%) depreciation on purchase	$ 168.69	$61.67	$61.67	$61.67	Cost of PC	$ 1,000.00		
Stationary & consumables	$ 214.93	$ 75.00	$ 78.75	$ 82.69	Salvage value of PC	$ 75.00		
Office space (30% allocated for this task)	$ 1,728.00	$ 576.00	$ 633.60	$ 696.96	Office space	64	sf	
Total operational costs	**$ 60,609.20**	$ 21,125.67	$ 22,207.67	$ 23,346.65	Rate	$ 2.50	psf-pm	
					NetSimplicity	$ 249.00	pm	
Net Gain/(Loss)	**(7,196.65)**							

Based on the assessment of alternative solutions so far, a solution comparable on all features have not yet been found. There are various solutions which offer some functionalities but not all. An example of NetSimplicity is taken which offers the scheduling function to some degree

- Figure 7: Calculations for the Commercial Off The Shelf Application

Based on the calculations, this option fares far better than keeping a status quo. By buying a software application, the tutoring agency almost reaches a break-even, though it still looses about $7,200 over a 3 year period. Most of the savings accrue due to the realize of tutor bandwidth to take more classes and generate more revenues. However, since none of the existing applications can do exactly as the agency wants, there would be some amount of time the tutors would have to spend on residual tasks. If there was no third option then the recommendation would have been to buy a scheduling application.

Option C: Tutorial Agency Management System (TAMS)

Budget - Option C

TAMS - Develop the TAMS (Tutoring Agency Management System) software

Benefits	NPV	Year 0	Year 1	Year 2
Revenue through utilization of Teaching Resource (for 3 hrs extra per day)	$ 96,459.55	$ 33,660.00	$ 35,343.00	$ 37,110.15
Revenue through utilization of Teaching Resource (reduction in error to 3%)	$12,740.91	$ 4,446.00	$ 4,668.30	$ 4,901.72
Total Benefit	**$ 109,200.46**	$ 38,106.00	$ 40,011.30	$ 42,011.87

Development/Solution Acquisition Costs				
Training (1 day, 4 people)	$ 857.85	$ 857.85	$ -	$ -
Effort in working with the project team (4h/wk ; 8 wks)	$ 857.85	$ 857.85	$ -	$ -
Development of the solution by UCB Team	$ -	$ -	$ -	$ -
Total Development/Solution Acquisition Cost	**$ 1,715.69**	$ 1,715.69	$ -	$ -

Operational Costs				
Software Maintenance (5h/m)	$ 6,408.38	$ -	$ 3,606.58	$ 3,786.91
Labour cost ($26.81ph x 1hr pd)	$ 19,973.95	$ 6,970.00	$ 7,318.50	$ 7,684.43
PC (1 part time for this purpose 30%) depreciation on purchase	$ 253.04	$92.50	$92.50	$92.50
Stationary & consumables	$ 143.29	$ 50.00	$ 52.50	$ 55.13
Office space (30% allocated for this task)	$ 1,728.00	$ 576.00	$ 633.60	$ 696.96
Total operational costs	**$ 28,506.65**	$ 7,688.50	$ 11,703.68	$ 12,315.92

Net Gain/(Loss) **78,978.12**

Assumptions			
Min ph rate per resource	$ 45.00	ph	*Tutoring Agency

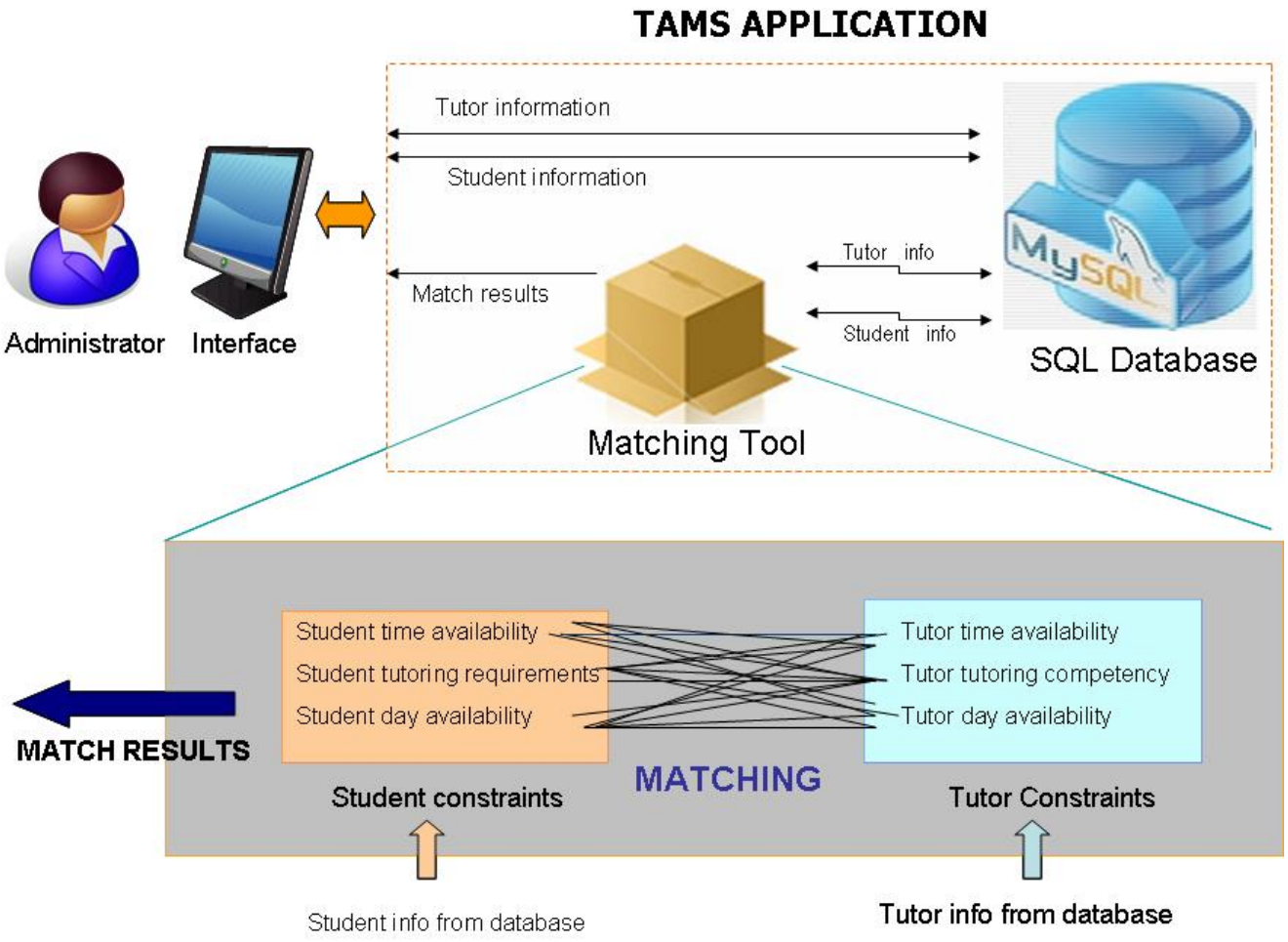
Assumptions			
Teacher's Salary	$ 26.81	ph	*Salary.com
IRR	10%		
Cost of PC	$ 1,000.00		
Salvage value of PC	$ 75.00		
Office space	64	sf	
Rate	$ 2.50	psf-pm	
NetSimplicity	$ 249.00	pm	
Software Engg Salary	$ 30.05	ph	*Salary.com

The solution is development by the UCB team as part of a class project. As an EFA analysis, the tutoring agency does not pay anything for the solution and therefore the cost has to be taken as zero, even though there is a notional cost of effort from the UCB team involved.

- Figure 8: Calculations for TAMS application

With an overall net gain of approx $79,000, TAMS presents the best option which the tutoring agency can go for to alleviate the scheduling problem. Since TAMS is a custom designed application with little cost of development, it addresses all the areas listed out and has been designed with the specific tutor profiles in mind. The significant upside is due to high utilization of tutoring resources as well as low cost of ownership of the application (the tutoring agency need not pay a yearly subscription and so on).

SYSTEM ARCHITECTURE



• Figure 9: Systems Architecture for TAMS

USER INTERFACE DESIGN

The User Interface development for TAMS was accomplished following the UCD (User Centered Design) process. This design philosophy focuses on the user of the system and their needs.

From the interviews done as part of the needs assessment our clients strongly expressed the need for a system that could simplify their current 'hairy' process of student tutor matching and scheduling.

USER INTERFACE DESIGN > INITIAL SKETCHES & LOW-FI PROTOTYPE

The initial sketches were aimed at capturing the most important functionalities that had surfaced based on results of user needs assessment, use scenarios and task analysis. Our aim was to keep the interface as simple as possible and separate the complexities of the back end from the front end design.

To test the usefulness of the initial design, the low-fi prototype usability testing was carried out. For this, a paper based prototype of the TAMS application was created. A paper based prototype is quick to set up, flexible, could be easily modified to respond to users' feedback and gives the users a feeling that it is not permanent and final and thus encourages them to come up with ideas and critiques about the interface. However, the challenge to address when coming up with the low-fi prototype is how to design the schedule matching interface that allows appointment cancellation, rescheduling, and adding a new tutoring session.

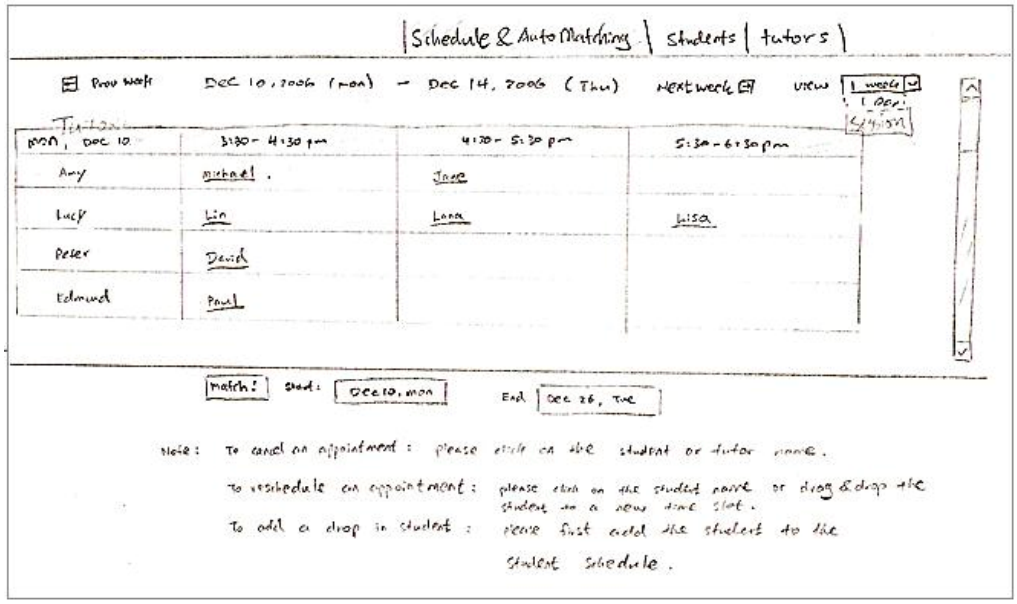
Prototype Description

To create the offline or paper-based prototypes, we used pencil and pen to sketch on 8.5 x 11 papers what each page of the schedule matching tool should look like. We focused on sketching out all the necessary and desirable features on each page without presenting the look and feel of the page. We tried to make sure each page layout and style is consistent with the overall layout and style of the schedule matching tool.

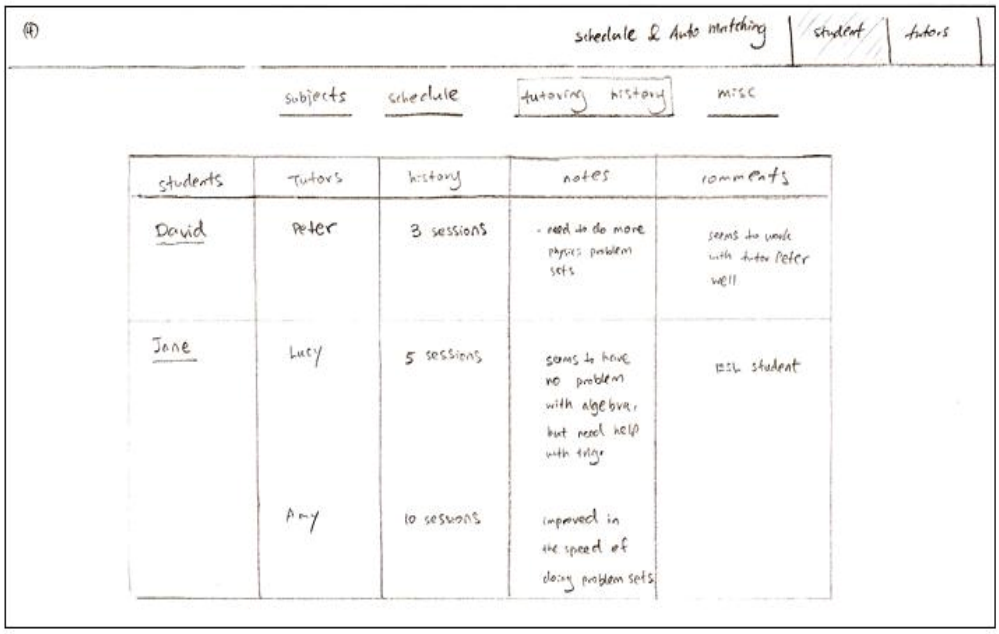
We sketched the following pages and functions:

- Tutor / Student Schedule Matching Tool
- Display Student Information: subjects to be tutored
- Display Student Information: student schedule of availability

- Display Student Information: tutoring history
- Display Student Information: tutoring history for an individual student
- Display Student Information: cancellation, rescheduling, and payment logging
- Display Tutor Information: subject of specialty
- Display Tutor Information: hours of availability



• Figure 10: Tutor/Student Scheduling Matching tool



- Figure 11: Display student Info :Subjects to be tutored

The complete set of screen shots from the paper based prototype can be found in Appendix [D].

Low-Fi Testing

We conducted the low-fi usability testing using the paper based prototype. As our primary target users were the administrative staff of the tutoring agency, we selected them as our participants. Two of the three administrative staff participated in the low-fi usability testing. We prepared a testing script which had a list of tasks that the participant was asked to carry out. Each of the participant were instructed to think aloud, tell if they clicked or browsed over a feature and to feel free to comment on features which they found were good, clear, obscure, difficult to understand etc. Although we had planned a particular sequence of tasks and questions to ask during the test, each test evolved in a very different way distinct from one another.

Design Feedback

Users suggested the following improvement for the Tutoring Matching Tool:

i. There may not be an exact match of student and tutor for one-on-one appointments. If there are more students than what can be scheduled, the matching tool needs to allow the tutoring

agency staff to match substitute tutors who can make the appointment time, and can teach the specific subject area.

ii. Availability of an *Undo* button

iii. Appointment Cancellation needs to have the option for selecting:

a) Sick – in which case there is no charge for the appointment

b) No Show – in which case there is a charge for the appointment

iv. The tutoring agency staff also pointed out the scheduling tool needs to be closely linked with the payment system. There needs to be an interface for showing the number of payment made on a given student given that payment is collected for every 10 hours of one-on-one tutoring.

Things That The Evaluation Could Not Tell Us

We could not really tell how quickly the users would be able to navigate and explore this interface, because the processing of the tool could not be measured with doing the paper walk-through. Also, the UI feedback could not be used to access the efficiency and the actual design of the schedule matching algorithm.

Additional Thoughts

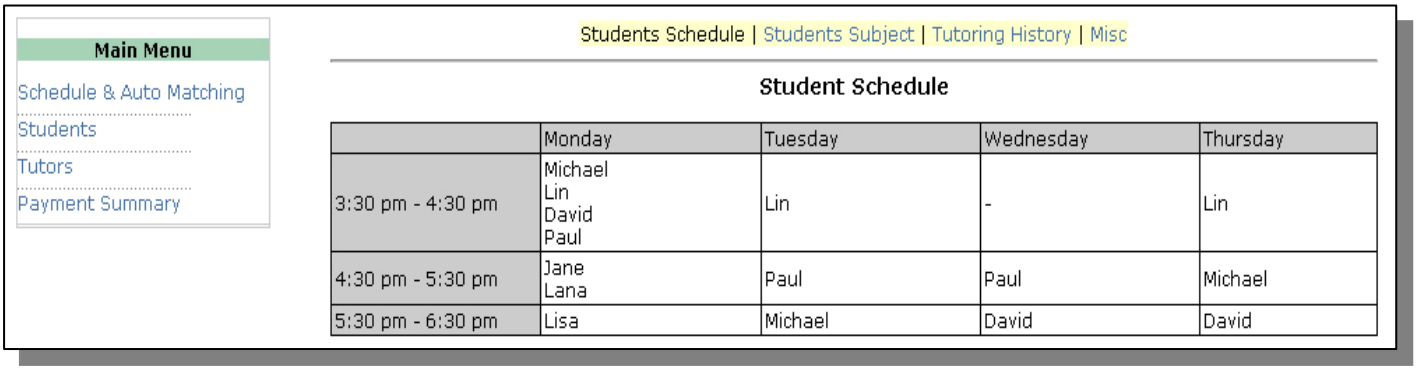
Although we got useful feedback in our low-fi testing, additional important feature requirements also surfaced namely pairing the tutor/student matching tool with the payment system. But realizing the complexity involved in the creation of a fully functional tutor/scheduling matching tool and given the limited time and resources, we had to limit the scope of this version of the project to just the one-to-one scheduling and put the integration of the payment system as a future endeavor.

USER INTERFACE DESIGN > FIRST INTERACTIVE PROTOTYPE

The low-fi user testing results reveals that our clients were happy with the features and the potential usefulness that the tool promised. However, due to lack of interactivity of the paper based prototype, the usefulness and the efficiency of the final scheduling tool could not be represented to our clients. To give a feeling of interactivity to the prototype, the first interactive prototype was created using PHP and HTML. This was integrated with the final prototype of the previous

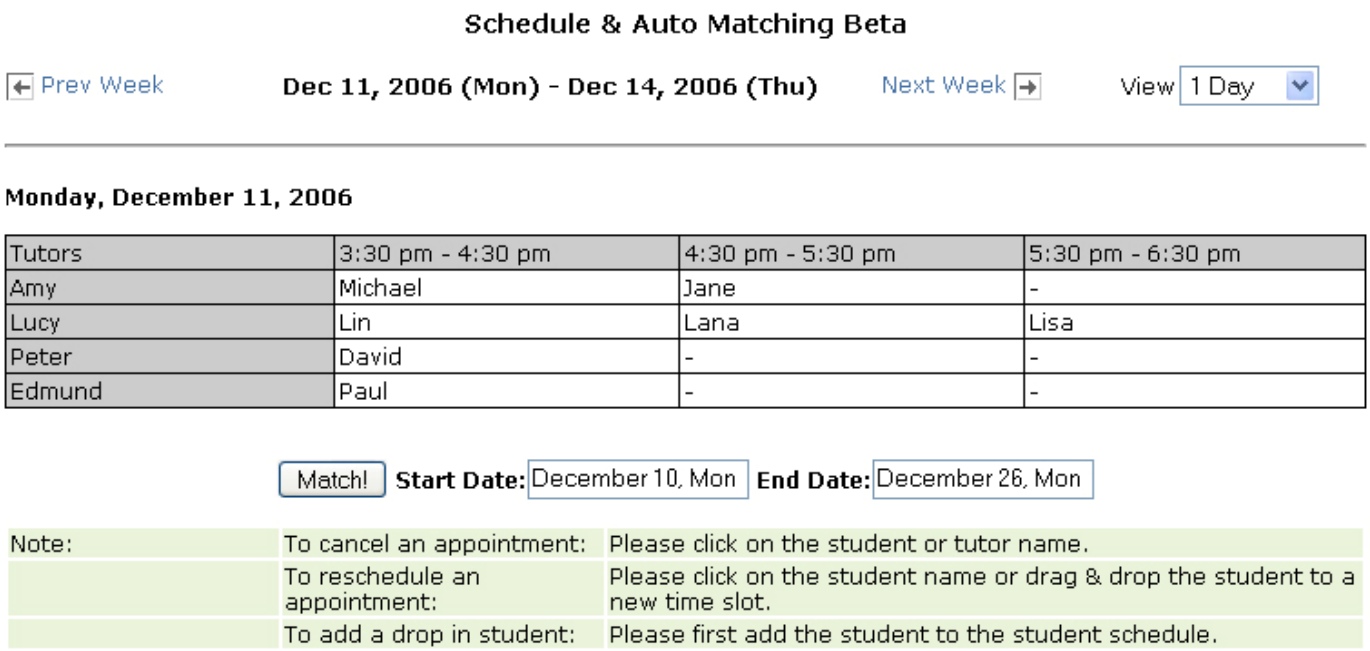
iSchool project (BA Info Redesign) to give a sense of completeness to the design. Thus the first interactive prototype had the main navigation features similar to the previous project with TAMS user interface displayed in the central content area. The purpose of the first interactive prototype was to test the interaction flow of the scheduling system, and to provide a demonstration of the features to our client, and to set up a context in which ideas about the tutoring matching system can be discussed, and imagined. Therefore, the look and feel of the interface was not focused on at this iteration, but the existing UI design from the BA Info Redesign's final prototype was adopted. Since our client is familiar with the UI of the BA Info Redesign prototype, this would also minimize the learning curve and help our client to focus more on the functionality of the schedule matching system

Some of the screen shots of the first interactive prototype are shown below and the remaining are in Appendix [E].



	Monday	Tuesday	Wednesday	Thursday
3:30 pm - 4:30 pm	Michael Lin David Paul	Lin	-	Lin
4:30 pm - 5:30 pm	Jane Lana	Paul	Paul	Michael
5:30 pm - 6:30 pm	Lisa	Michael	David	David

• Figure 12: Student schedule depicted in the first interactive prototype



Schedule & Auto Matching Beta

Prev Week Dec 11, 2006 (Mon) - Dec 14, 2006 (Thu) Next Week View 1 Day

Monday, December 11, 2006

Tutors	3:30 pm - 4:30 pm	4:30 pm - 5:30 pm	5:30 pm - 6:30 pm
Amy	Michael	Jane	-
Lucy	Lin	Lana	Lisa
Peter	David	-	-
Edmund	Paul	-	-

Match! Start Date: December 10, Mon End Date: December 26, Mon

Note:	To cancel an appointment:	Please click on the student or tutor name.
	To reschedule an appointment:	Please click on the student name or drag & drop the student to a new time slot.
	To add a drop in student:	Please first add the student to the student schedule.

• Figure 13: Schedule & Auto Matching (First Interactive Prototype)

Differences From Low-Fi Prototype

A side menu is added to allow TAMS to become more of an integrated part of the existing tutoring application. This includes allowing the tutoring admin to easily access the payment page from the Tutoring Scheduling page.

What Was Left Out And Why

- Payment Summary was kept as a placeholder for future work. No screen was designed for any payment system as we had identified the integration of the scheduling tool with the payment feature as out of scope for this version of TAMS.

- One-to-one tutoring scheduling was focused on in this version. The features like group tutoring were not considered due to lack of time and resources. However we do realize the necessity of the group scheduling feature and have taken into consideration the possibility of future extension of this project to include group scheduling.

First Interactive Prototype Testing & Feedback

We conducted usability testing on the first interactive prototype before we started the creation of the databases for the back end. We went to

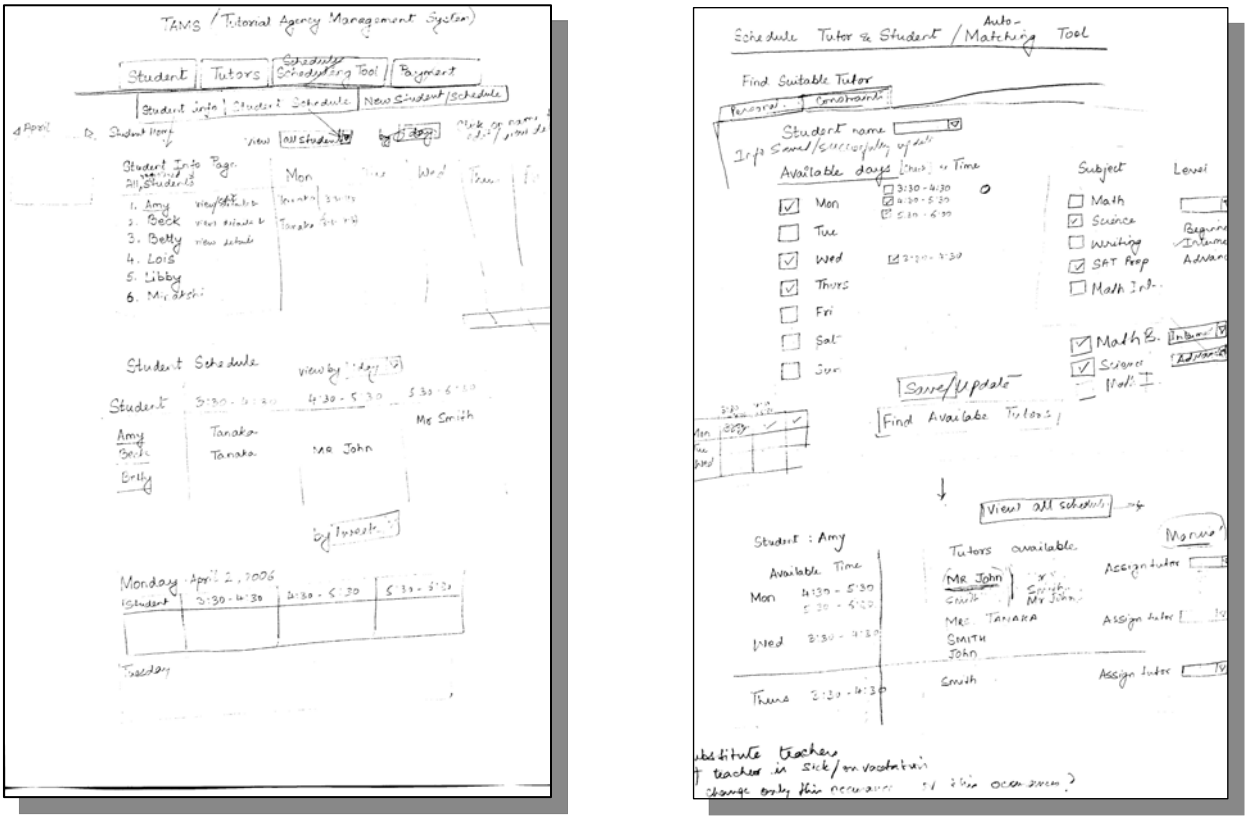
the TA administrative staff to conduct our test on the first interactive prototype. This time as the prototype had a more 'real' feel to it with the features like checkboxes and submit buttons working. The key feedback point from the user testing was:

- Main navigation on the left with the secondary navigation on the top seemed to be confusing.

USER INTERFACE DESIGN > SECOND INTERACTIVE PROTOTYPE

From the usability test results of the second interactive prototype we realized the need to redesign some pages and change the navigation positions.

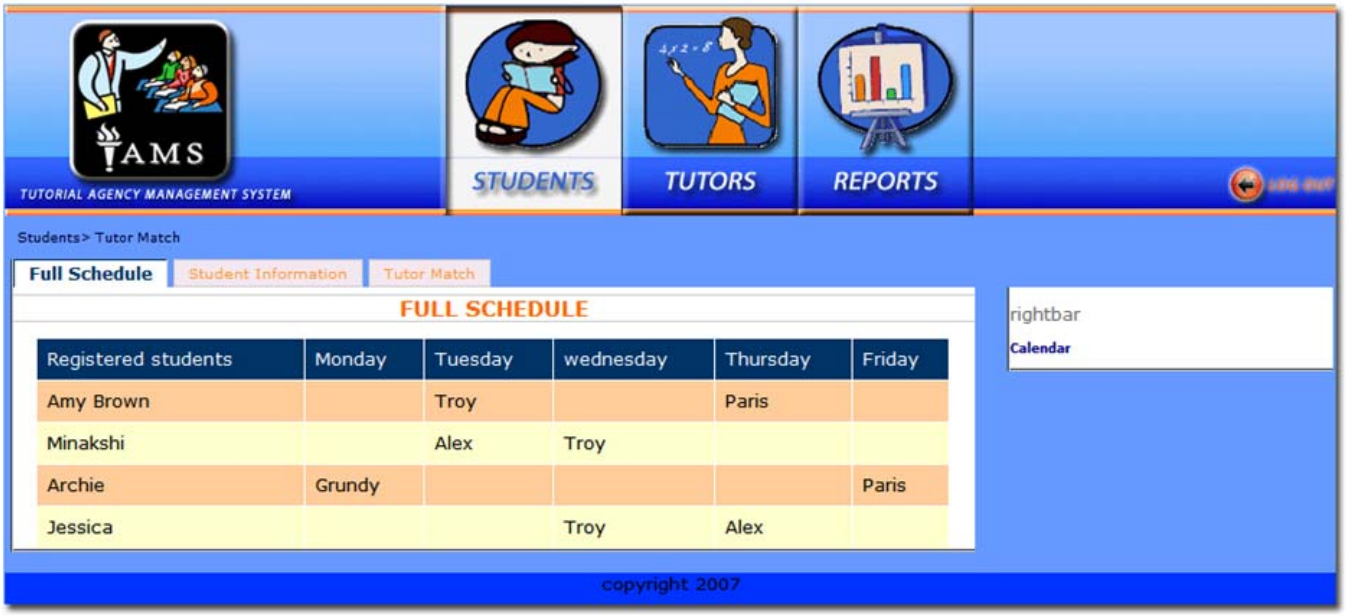
As part of the interactive design process we had to go back to the drawing board to sketch out new interfaces for some pages and also changed some more features from the feedback and suggestions that surfaced from the first low-fi prototype testing.



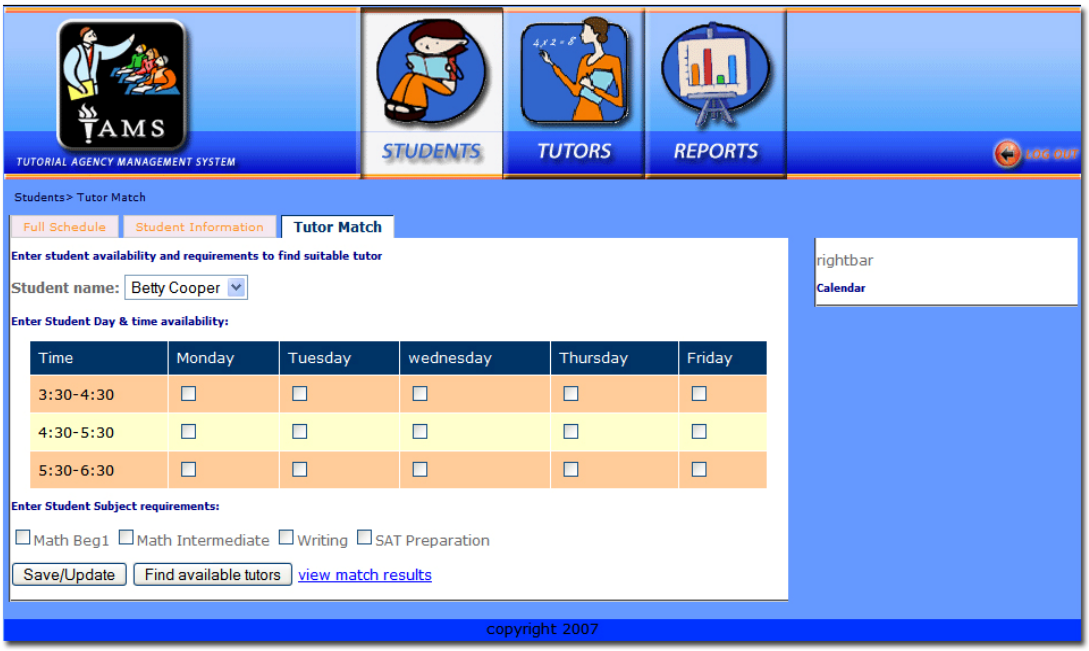
• Figure 14: Revised sketches

Second Interactive Prototype Description

A couple of screen shots are reproduced below.



• Figure 15: Second Interactive Prototype – Full Schedule Page



• Figure 16: Second Interactive Prototype – Tutor Match Page

The following points enumerate the key features that were added or improved in the second interactive prototype:

- Primary and secondary navigation were placed on the top
- A calendar feature was added although the integration of the scheduling system with the calendar was listed as future work
- Login with a user name and password that would only enable the administrative staff to access the application database
- Reports section that displayed trend graphs.
- The entire tool was given the look and feel of a stand alone application.
- Icons for student, tutors and reports were added.
- The blue color was used to give the application a professional look along with shades of warm orange to add a friendly feel.

Second Interactive Prototype Testing

Before we conducted the usability testing on the second interactive prototype, we had the backend matching algorithm and the MySQL database working. We integrated the back end with the second interactive prototype to check the actual working of the prototype. We then tested the integrated prototype with the tutoring agency administrative staff. We prepared a testing script which had a list of tasks that the participant was asked to carry out. The participant was instructed to think aloud as they went about carrying out the tasks. We used a tape recorder to record the entire session after we asked if she feels comfortable with the session being tape recorded. The studies were conducted using the laptop of one of our team members.

Key Feedback On The Second Interactive Prototype

- Separate overall schedule views for tutors and students
- Complete view of students in the student details page.
- A feature to add relevant notes about a student in the student info page.

USER INTERFACE DESIGN > FINAL PROTOTYPE

- Figure 17: Final Interactive Prototype – Students Full Schedule Page

- Figure 18: Final Interactive Prototype – Student Details Page

A few final changes were added to the second interactive prototype as a result of feedback from our client in the user testing and also from our faculty advisor.

Changes done:

- A separate tutor match tab removed

- A different view for full schedule designed in the tutor schedule and student schedule page. The student schedule page contains the view from the student's perspective where as the tutor schedule view displays the schedule from the tutor's perspective

- The axes in the graphs in the report section changed to follow the standard norms.

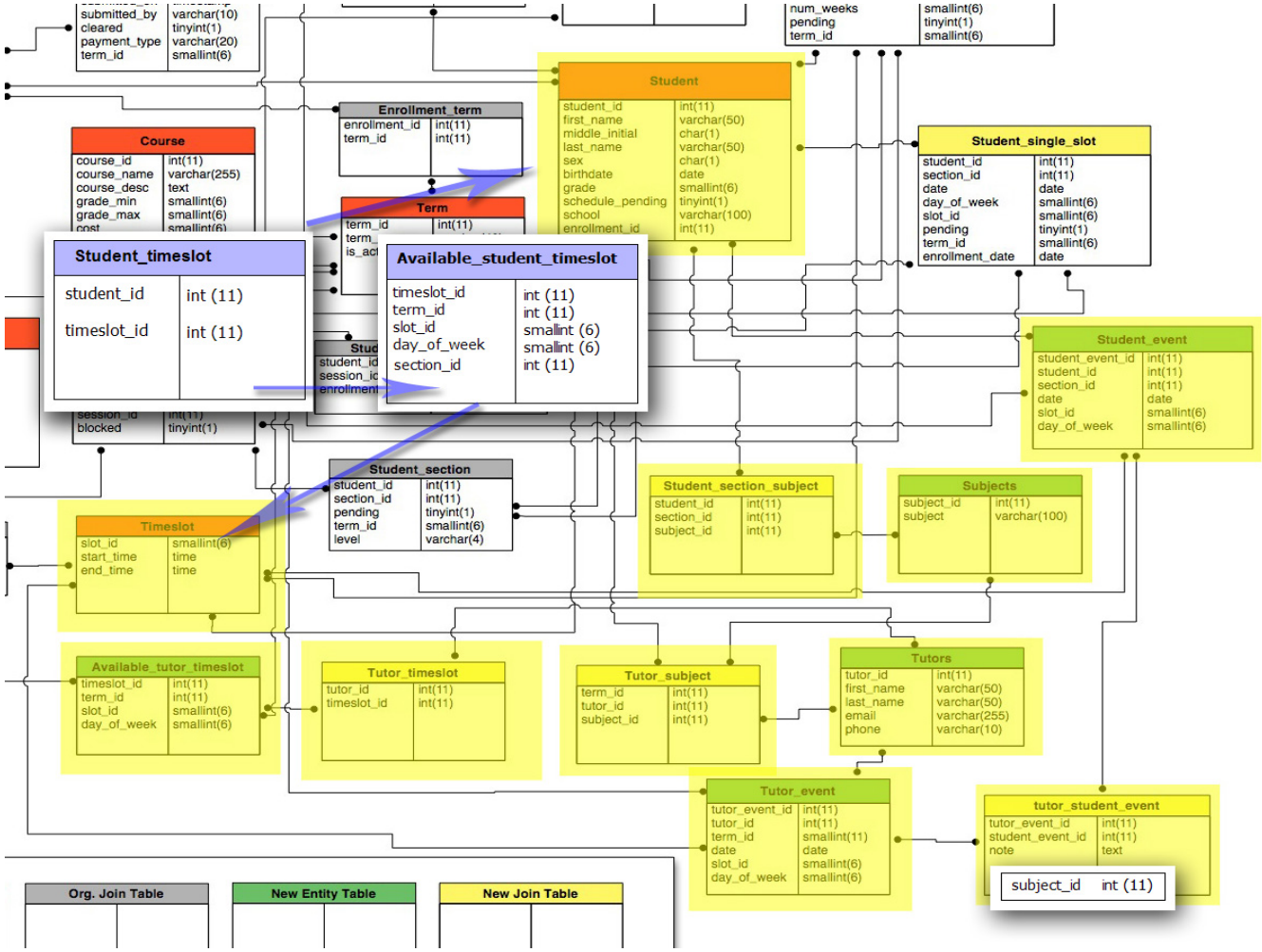
Features that could not be implemented due to lack of time:

- Notes section feature

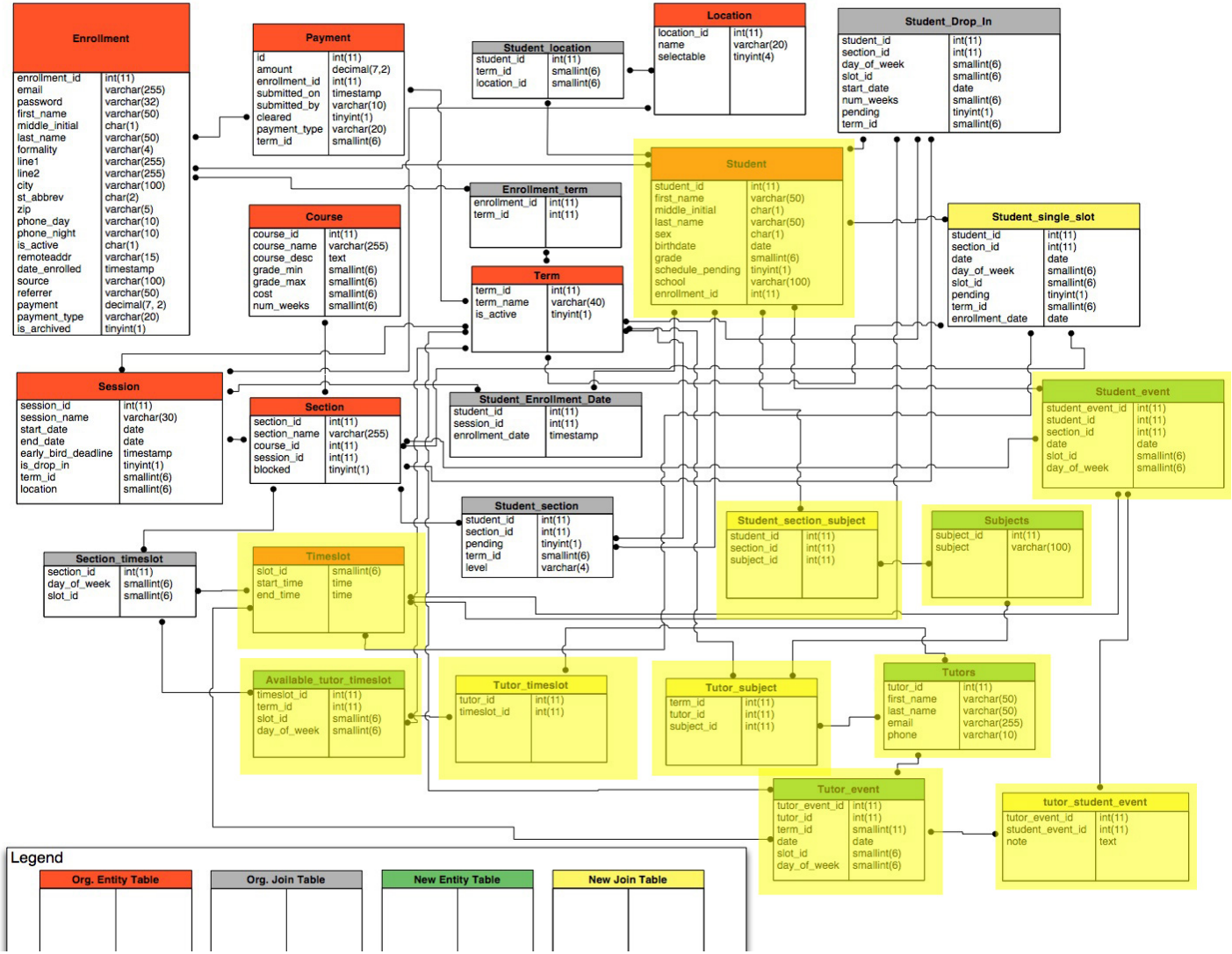
DATABASE DEVELOPMENT

Database development was an important component of the TAMS application. While two new tables were created specifically for TAMS, and one existing table was modified, we mostly relied and used earlier tables which contained details on students and tutors. This section details the database design and modification process.

DATABASE DEVELOPMENT > E-R DIAGRAM



- Figure 19: ER Diagram for TAMS Database (note the pop up images indicate the new additions to the existing TA database)



- Figure 20: Database tables used in TAMS with Relation to the Overall TA Database (the highlighted tables are the ones used in TAMS)

DATABASE DEVELOPMENT > DESCRIPTION & DATA DICTIONARY

The database tables are created in MySQL, an open source relational database management system (RDBMS). Since TAMS is meant to be used as one component of the tutoring agency's current information management system, the database tables are integrated with the already existing database tables.

The tutoring agency contains more than 20 database tables that store information in the areas of students and tutors' profiles; student registration information; the courses offered at the tutoring agency; and the payment information for the tutoring sessions.

TAMS uses a subset of the existing database tables that are currently used by the tutoring agency to organize and retrieve information on tutors and students. During the process of building the TAMS matching tool, we also realized the need to create new tables to store the necessary information, as well as to modify one existing table.

DATABASE DEVELOPMENT > TABLES CREATED

The "Available_Student_Timeslot" Table

The "available_student_timeslot" table stores all possible combinations of the day and time information that any student can potentially indicate as a session that they are available to receive tutoring. This table is primarily used for printing out the basic weekly schedule availability input table.

A detailed description of all the fields in the "available_student_timeslot" table is as follows:

```
mysql> describe available_student_timeslot;
+-------------+-------------+------+-----+---------+----------------+
| Field       | Type        | Null | Key | Default | Extra          |
+-------------+-------------+------+-----+---------+----------------+
| timeslot_id | int(11)     |      | PRI | NULL    | auto_increment |
| term_id     | int(11)     |      |     | 0       |                |
| day_of_week | smallint(6) |      |     | 0       |                |
| slot_id     | smallint(6) |      |     | 0       |                |
| section_id  | int(11)     | YES  |     | NULL    |                |
+-------------+-------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

- Table 2: Available_Student_Timeslot Table

The "Student_Timeslot" Table

The "student_timeslot" table is a relationship table that stores the connection between a student and a time that this student is available to receive tutoring. More specifically, it stores the relationship between an entry in the "student" table and an entry in the "available_student_timeslot" table. This table is used, along with "student", and "available_student_timesot" table to pull out information on each student's tutoring availability schedule.

A detailed description of all the fields in the "student_timeslot" table is as follows:

```
mysql> describe student_timeslot;
+-------------+---------+------+-----+---------+-------+
| Field       | Type    | Null | Key | Default | Extra |
+-------------+---------+------+-----+---------+-------+
| student_id  | int(11) |      | PRI | 0       |       |
| timeslot_id | int(11) |      | PRI | 0       |       |
+-------------+---------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

- Table 3: Student_Timeslot Table

DATABASE DEVELOPMENT > TABLES MODIFIED

The "Tutor_Student_Event" Table

Change made: A new column "subject_id" is added to store the subject information for each tutoring sessions.

The "tutor_student_event" table is a relationship table that stores all the connections between a student and a tutor that has been matched for a tutoring session. More specifically, it stores the relationship between an entry in the "student_event" table, and an entry in the "tutor_event" table. This table is used, along with "student_event", and "tutor_event" tables to retrieve the current tutoring matching information.

A detailed description of all the fields in the "tutor_student_event" table is as follows:

```
mysql> describe tutor_student_event;
+------------------+---------+------+-----+---------+-------+
| Field            | Type    | Null | Key | Default | Extra |
+------------------+---------+------+-----+---------+-------+
| tutor_event_id   | int(11) |      | PRI | 0       |       |
| student_event_id | int(11) |      | PRI | 0       |       |
| note             | text    | YES  |     | NULL    |       |
| subject_id       | int(11) | YES  |     | NULL    |       |
+------------------+---------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

- Table 4: Tutor_Student_Event Table

DATABASE DEVELOPMENT > TABLES USED

The following tables already exist and are used by the tutoring agency prior to the creation of TAMS. The tables can be grouped into the following categories based on their content storage.

Information Personal Profile

The following tables store the general information on students and tutors.

- Student
- Tutors
- Student_location

Scheduling Information

The following tables store the schedule of availability of tutors, possible tutoring timeslots; and all the possible potential tutoring times for students.

- available_tutor_timeslot
- timeslot
- available_student_timeslot

Subject Information

The following tables store the tutoring subject information, as well as the subjects that students would like to get tutoring on, and the tutoring subject competency of the tutors.

- student_section_subject

- tutor_subject
- subjects

Matching Information

The following tables store information on the tutor and students' scheduled tutoring sessions.

- tutor_event
- student_event
- tutor_student_event

DATABASE DEVELOPMENT > TRANSACTIONS & SAMPLE QUERIES

1. Retrieve an individual student's tutoring availability

First, retrieve all the available tutoring sessions. Notices it is basically daily from 3:30 – 4:30 pm. This data can be expanded as the tutoring agency decided to make available more tutoring operation hours.

```
mysql> select att.timeslot_id as timeslot_id, att.day_of_week, time_format(start_tim
e, '%l:%i') as start_time, time_format(end_time, '%l:%i') as end_time from available
_student_timeslot att, timeslot ts where ts.slot_id = att.slot_id AND att.term_id =
106 order by att.slot_id, att.timeslot_id;
+-------------+-------------+------------+----------+
| timeslot_id | day_of_week | start_time | end_time |
+-------------+-------------+------------+----------+
|           1 |           1 | 3:30       | 4:30     |
|           4 |           2 | 3:30       | 4:30     |
|           7 |           3 | 3:30       | 4:30     |
|          10 |           4 | 3:30       | 4:30     |
|          13 |           5 | 3:30       | 4:30     |
|          16 |           6 | 3:30       | 4:30     |
|          19 |           7 | 3:30       | 4:30     |
|           2 |           1 | 4:30       | 5:30     |
|           5 |           2 | 4:30       | 5:30     |
|           8 |           3 | 4:30       | 5:30     |
|          11 |           4 | 4:30       | 5:30     |
|          14 |           5 | 4:30       | 5:30     |
|          17 |           6 | 4:30       | 5:30     |
|          20 |           7 | 4:30       | 5:30     |
|           3 |           1 | 5:30       | 6:30     |
|           6 |           2 | 5:30       | 6:30     |
|           9 |           3 | 5:30       | 6:30     |
|          12 |           4 | 5:30       | 6:30     |
|          15 |           5 | 5:30       | 6:30     |
|          18 |           6 | 5:30       | 6:30     |
|          21 |           7 | 5:30       | 6:30     |
+-------------+-------------+------------+----------+
21 rows in set (0.00 sec)
```

• Table 5: All student availability query result

Second, retrieve the availability of this student to receive tutoring. Notice for this student, the details of his tutoring availability can be pulled out from the available_student_timeslot based on the timeslot_id.

```
mysql> select timeslot_id from student_timeslot where student_id=1170;
+-------------+
| timeslot_id |
+-------------+
|           1 |
|           2 |
|           3 |
+-------------+
3 rows in set (0.00 sec)
```

• Table 6: Student availability query result

Note that retrieving tutor's availability works the same way, except we are pulling data from "available_tutor_timeslot" and "tutor_timeslot".

2. Obtain the subjects that the students wants to receive tutoring on

Retrieve all the tutoring subjects and the subjects that a student needs tutoring on.

```
mysql> select * from student_section_subject where student_id=1170;
+------------+------------+------------+
| student_id | section_id | subject_id |
+------------+------------+------------+
|       1170 |          1 |          1 |
|       1170 |          1 |          2 |
+------------+------------+------------+
2 rows in set (0.00 sec)

mysql> select * from subjects;
+------------+-----------+
| subject_id | subject   |
+------------+-----------+
|          1 | Math      |
|          2 | Writing   |
|          3 | Physics   |
|          4 | Chemistry |
|          5 | Reading   |
+------------+-----------+
5 rows in set (0.00 sec)
```

• Table 7: Requested subjects query result

Note that retrieving subject competency of tutors works the same way, except we are pulling data from the "tutor_subject" relationship table.

3. Retrieving all the students a given tutor is assigned to tutor on a given day.

```
mysql> SELECT student_id, e.subject_id, te.slot_id as slot_id FROM student_event as
se, tutor_student_event as e, tutor_event as te WHERE te.tutor_id = 12 AND te.tutor_
event_id = e.tutor_event_id AND se.student_event_id = e.student_event_id AND te.day_
of_week = 1 ORDER BY te.slot_id;
+------------+------------+---------+
| student_id | subject_id | slot_id |
+------------+------------+---------+
|       1106 |          2 |       1 |
|       1170 |          2 |       2 |
|       1106 |          5 |       3 |
+------------+------------+---------+
3 rows in set (0.00 sec)
```

• Table 8: Student-tutor assignment query result

Note this information is retrieved from three different tables (student_event, tutor_student_event, and tutor_event), and combined into one view.

STUDENT-TUTOR MATCHING ALGORITHM

The tutor student matching algorithm determines how the automatic tutor student assignment works. It automatically generates several matching configurations based on the following constraints, and it selects the ideal matching configuration to be saved and printed.

Constraints

a) A student must be matched to a tutor when both of them are available

b) A student must be assigned to a tutor who is capable of tutoring a specific subject that the student needs tutoring on

c) No tutor can tutor more than one student at a time (one-on-one tutoring)

d) No tutor can tutor multiple subjects in one session

For the sake of efficiency, the matching process terminates as soon as an ideal match is found. If no ideal match is found after all the tutoring configurations have been generated, then the best match among all the configurations is returned. Note that in this case, the best match may not be the ideal match.

An ***ideal match*** is a match in which the student is assigned to an available tutor for all the subjects that he/she wants to be tutored on.

A ***best match*** is a match in which the student can be assigned to an available tutor for the most number of subjects that he/she wants to be tutored on in comparison to all the other auto-generated tutoring matching configurations.

Assumptions

- The tutoring weekly schedule is consistent every week
- Tutoring session duration: one hour

Algorithm For Generating One Matching Configuration

A. Given this student subject (e.g. math), and this given time, find the first tutor that can tutor this student on this subject

B. If a tutor found, store it in match configuration array and move on to the next subject (e.g. writing)

C. If no available tutor can tutor this subject at this time, repeat this match search for the next time the student is available. Keep repeating until a tutor that can tutor this subject is found)

D. If no tutor can be found that can tutor this subject for this student at all, move on to find tutors for the other subjects.

E. Continue process A through D until the attempt to find tutors for all subjects have been completed.

TREND ANALYSIS TOOLS

Trend Analysis Tools are used to generate reports that can be used by the tutoring administrator to view the overall current growth trend of the tutoring agency. The reports can be used to assist tutoring agency director to make better business decisions. Therefore, the trend analysis tools can be seen as a business intelligence tool.

For trend analysis, we used PHP/SWF Charts, a powerful open source tool that can be used "to create attractive web charts and graphs from dynamic data".[1]

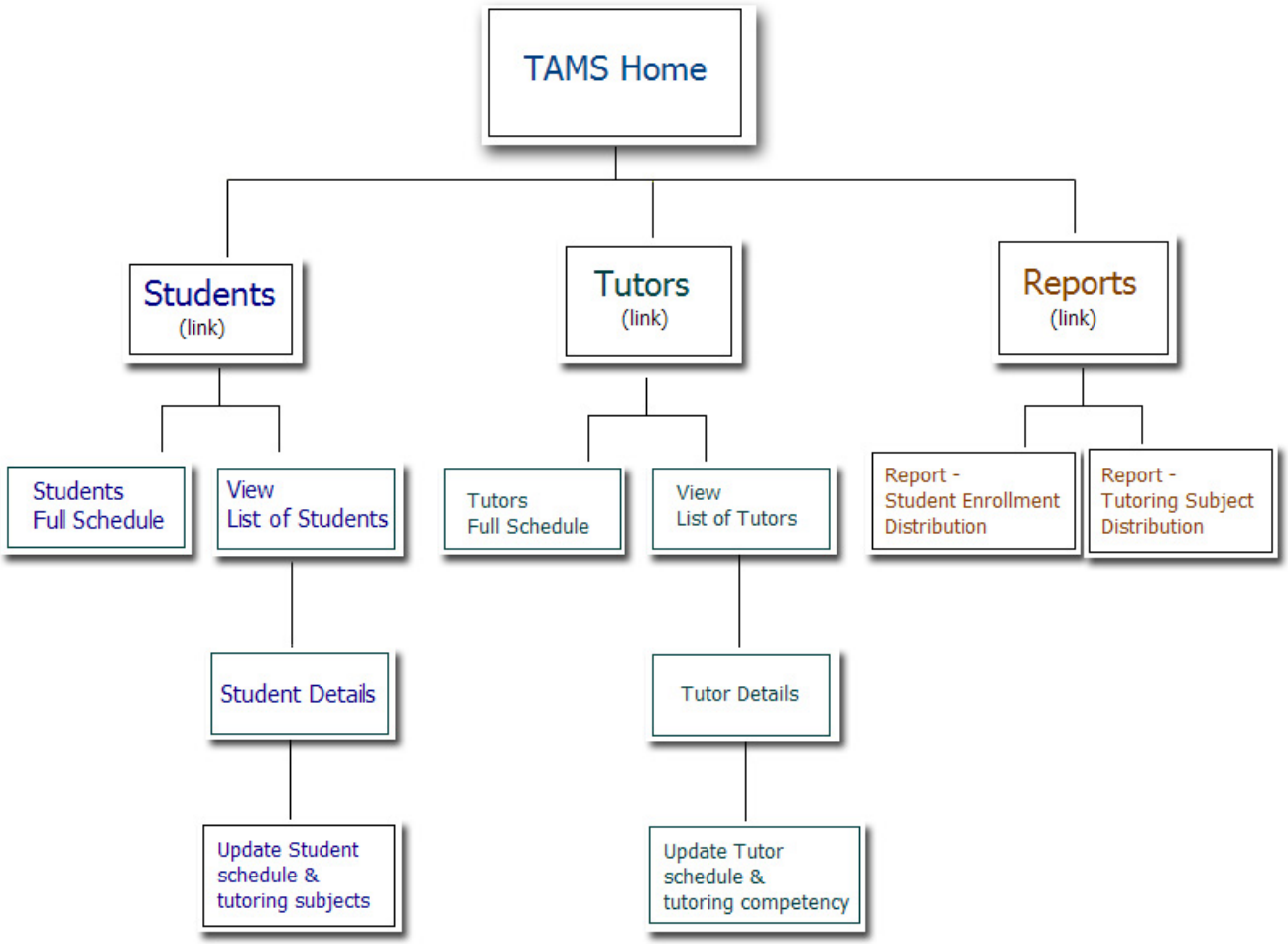
We use PHP scripts to generate data from our databases, and pass the data to the PHP/SWF Charts to generate Flash (SWF) charts.

"PHP/SWF Charts makes the best of both the PHP and SWF worlds. PHP scripts provide integration, and Flash provides the best graphic quality."[2]

[1] http://www.maani.us/charts/index.php?menu=Introduction, last access on April 27, 2007

[2] Ibid.

INTERACTION FLOW DIAGRAM



• Figure 21: Interaction Flow Diagram

FINAL APPLICATION DESCRIPTION

FINAL APPLICATION DESCRIPTION > OVERVIEW

Scheduling suitable tutors to incoming students based on the needs and availability of the student is one of the key functions of the scheduler or administrative staff of any tutoring agency. In the first version of TAMS, we have focused on one-to-one tutoring.

In the interface of TAMS Version 1.0, the main navigation bar is placed at the top and has tabs for *Students*, *Tutors* and *Reports*. Each tab is represented with an icon and text.

• Figure 22: TAMS Icons

The secondary tabs are placed on the top of the main content area. Each main tab of students and tutors has corresponding secondary tabs for full schedule and student and tutor information.

We assume that the student details are stored in the student databases when a new student registers. TAMS can be used to schedule a suitable available tutor to the already registered student for one-on-one tutoring.

We have described each page and the process of student-tutor scheduling in TAMS further in this section. The given description can also serve as a user manual for any new user of TAMS. In addition, we have also mentioned some features which we believe could enhance the functionality and usefulness of the page but could not be implemented as part of this version due to time and resource constraints.

• Figure 23: Students Full Schedule Page

The *Students Full Schedule* page is the default page which opens when the *Student*s tab is clicked in the top main navigation. This page displays the complete tutoring schedule of the agency for the current week. It contains a list of all the currently registered students and shows the days and time the student is scheduled to be tutored and also the tutors assigned to the student. This table takes the data from the MySQL database. Therefore, any change made in the database as a result of rescheduling is reflected in the students full schedule table. This facilitates convenient viewing of the overall updated schedule by the tutorial administrative staff at a single glance.

The table reflects the schedule for a single week.

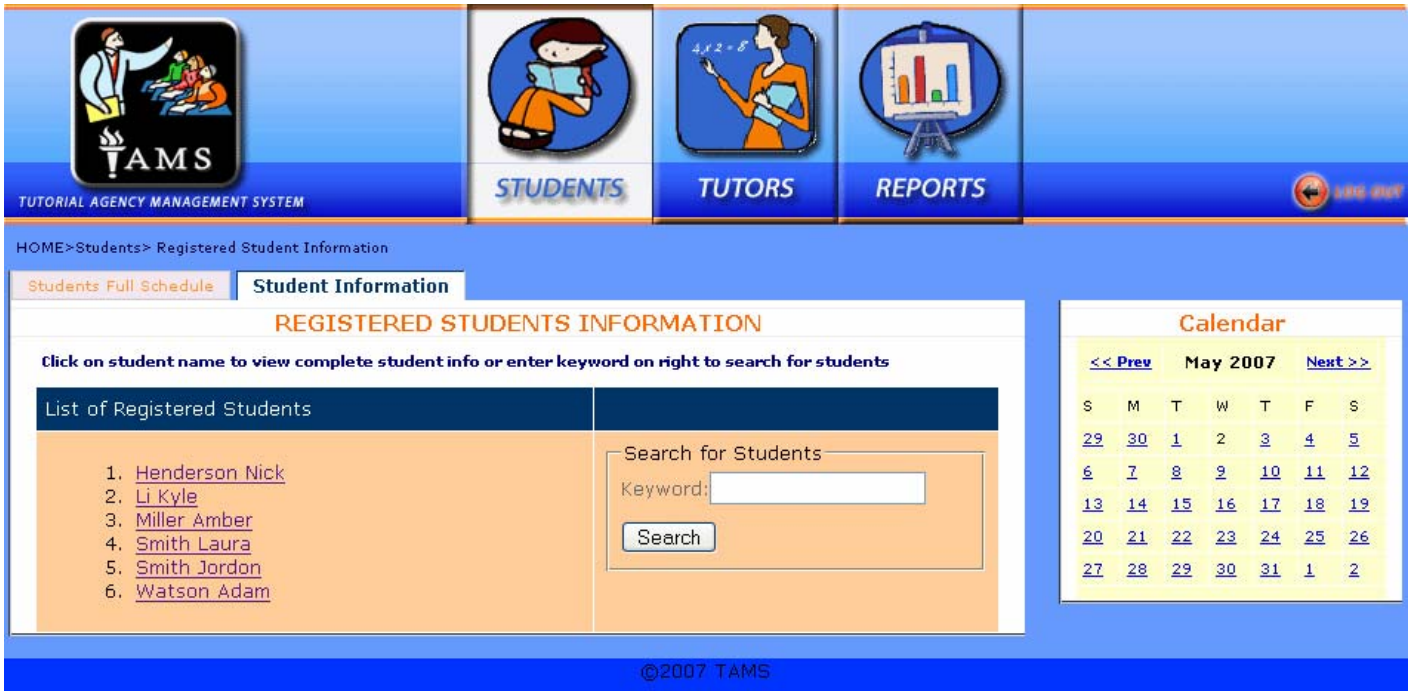
A monthly calendar is available at the right side bar to look up the dates while the schedule is being viewed.

Functions left out and future scope

- We understand that it would be even more convenient if the calendar on the right bar would be integrated with the display of the student schedule such that the user can click on a particular date on the calendar to view that week's schedule.

- Each name in the student schedule could be hyperlinked to the details page for quick viewing of individual details

FINAL APPLICATION DESCRIPTION > STUDENT INFORMATION PAGE



• Figure 24: Students Information Page

This page displays the list of currently registered students. Each name of the student is hyperlinked and the administrator can click on a particular name to see detailed information about the student. The page also has a search feature (not currently functional) to search for students quickly.

Functions left out and future scope

- The search feature (although not implemented in Version 1.0) will be useful when the list of students is very long and the administrator cannot remember the exact name of the student. In that situation, the administrator can enter any related keyword and the system will list the names of the students related to the entered keyword.

FINAL APPLICATION DESCRIPTION > STUDENT DETAILS PAGE

• Figure 25: Students Details Page

The *Student Details Page* displays all the details about a particular student. It shows the personal details of the student that is stored in the student database, the subjects the student is getting tutored on, the time of the tutoring session and the current tutor assigned to the student. In addition, it also displays a list of potential tutors. The list of potential tutor contains the list of tutors who teach the required subject and are available on the same days as the student but are scheduled to tutor some other student at that point of time. A link leads to the *Edit Student Schedule & Tutoring Subjects* page where any change of schedule can be inputted. The matching algorithm runs each time a change of schedule is made and the student details show an updated matched tutor list and potential tutors. So the student details page always shows the updated and current assigned tutors.

Functions left out and future scope

- We felt it be neat if the name of the student currently assigned to the potential tutor would show as a pop up when the administrator mouse over a tutor's name.

FINAL APPLICATION DESCRIPTION > EDIT STUDENT SCHEDULE & SUBJECTS PAGE

• Figure 26: Edit Student Schedule & Subjects Page

The *Edit Student Schedule & Subjects* page allows the administrator to reschedule a student or change the subject requirements of a student. Every time a change is scheduled or subject requirements is made in this page, the matching algorithm runs in the back end and the new tutors are assigned to the student which is reflected in the student details page.

FINAL APPLICATION DESCRIPTION > TUTORS FULL SCHEDULE PAGE

• Figure 27: Tutors Full Schedule Page

The *Tutors Full Schedule* page is the default page which opens when the *Tutors* tab is clicked in the top main navigation. This page displays the all of the tutors' schedule in a weekly view. As opposed to the *Students Full Schedule* page, this is a view from the tutors' viewpoint. It contains a list of all the currently registered tutors and shows the days and time the tutor is scheduled to tutor and also the student who is assigned to the tutor. This table takes the data from the MySQL database. Therefore, any change made in the database as a result of rescheduling is reflected in the tutor full schedule table. This facilitates convenient viewing of the overall updated schedule by the tutorial administrative staff at a single glance.

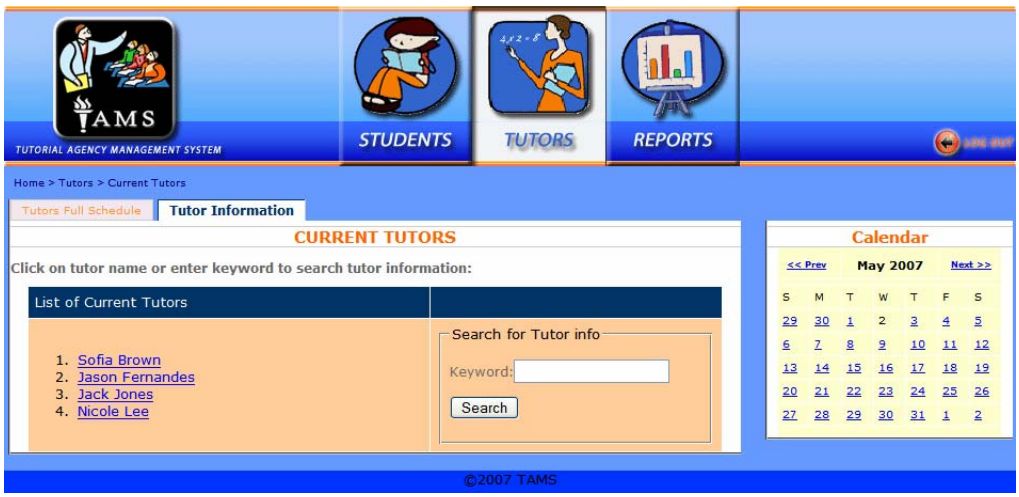
The table reflects the schedule for a single week.

A monthly calendar is available at the right side bar to facilitate looking up the dates at the same time the schedule is viewed.

Functions left out and future scope

- We understand that it would be even more convenient if the calendar on the right bar would be integrated with the display of the student schedule such that the user can click on a particular date on the calendar to view that week's schedule
- Each name in the tutor schedule could be hyperlinked to the tutor details page for quick viewing of individual details

FINAL APPLICATION DESCRIPTION > TUTOR INFORMATION PAGE



• Figure 28: Tutor Information Page

This page displays the list of currently registered tutors. Each name of the tutor is hyperlinked and the administrator can click on a particular name to see detailed information about the tutor. The page also has a search feature to search for students quickly.

Functions left out and future scope

- The search feature (although not implemented in version 1) will be useful when the list of students is very long and the administrator cannot remember the exact name of the student. In that situation the administrator can enter any related keyword and the system will list the names of the students related to the entered keyword.

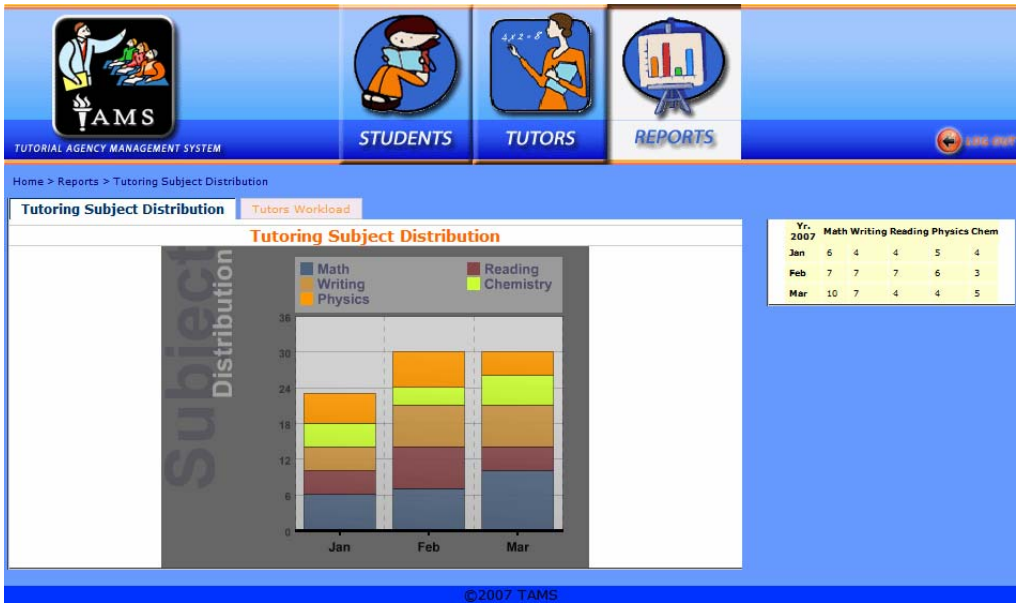
• Figure 29: Tutor Details Page

The *Tutor Details* page displays all the details about a particular tutor. It shows the personal details of the tutor that is stored in the tutor database, the subjects the tutor is proficient to tutor, the time of the availability and the current student assigned to the tutor. A link leads to the edit tutor schedule page where any change of schedule can be inputted.

Functions left out and future scope

- We plan to implement the actual re-matching algorithm when a tutor adjusts his/her schedule of availability, and tutoring competency. In this case, if a tutor is no longer available at a certain time, the students that are previously assigned to this tutor need to be paired up with new tutors.

FINAL APPLICATION DESCRIPTION > REPORTS



• Figure 30: Reports Page

The Report page can be seen as a business intelligence part of TAMS application that allows the tutoring administrator to view the overall current growth trend of the tutoring agency. The reports can be used to assist tutoring agency director to make better business decisions. The Report page also allows the user to view the business data in tabular format on the right hand side of the page. The graph is generated dynamically by putting the business data from the database in real time, the graph is plotted using a Flash based software.

The current report displays the distribution of the number of students getting tutoring in the five different subjects that are currently offered by the tutoring agency in the past several months. Our client has informed us that a graph for viewing the student enrollment distribution trend by month would be very useful.

Functions left out and future scope

- We plan to implement the section that generates student enrollment distribution graph. Also, a feature that lets user customize the parameters (e.g. domain and range of the data) of the business trend graphs would be useful.

IMPLEMENTATION TECHNOLOGIES & TOOLS

PHP

PHP is a widely-used general-purpose "server-side, cross-platform, HTML embedded scripting language that lets [us] create dynamic web pages."[3] PHP was selected as scripting language due to two major reasons – (a) the tutoring agency already had systems (including the earlier redesign project) which used this language and hence any future development, integration and changes can be easily implemented, and (b) PHP is open source with large amount of resources on the internet and hence amenable to projects such as TAMS.

MYSQL

For database management, we used MySQL, which is "an open source relational database management system (RDBMS) that uses Structured Query Language (SQL), the most popular language for adding, accessing, and processing data in a database. Because it is open source, anyone can download MySQL and tailor it to their needs in accordance with the general public license. MySQL is noted mainly for its speed, reliability, and flexibility."[4]

XHTML

XHTML is a family of current and future document types and modules that reproduce, subset, and extend HTML 4. XHTML family document types are XML based, and ultimately are designed to work in conjunction with XML-based user agents.[5]

CSS

CSS (Cascading Style Sheets) is a simple mechanism for controlling the style of a Web document without compromising its structure. By separating visual design elements (fonts, colors, margins, and so on) from the structural logic of a Web page, CSS gives Web designers the control they crave without sacrificing the integrity of the data - thus maintaining its usability in multiple environments. In addition, defining typographic design and page layout from within a single,

[3] www.acad.bg/beginner/gnrt/appendix/glossary.html; last access on April 27, 2007
[4] www.shop-script.com/glossary.html; last access on April 27, 2007
[5] www.w3.org/TR/xhtml1/#xhtml; last access on April 28, 2007

distinct block of code - without having to resort to image maps, <font> tags, tables, and spacer GIFs - allows for faster downloads, streamlined site maintenance, and instantaneous global control of design attributes across multiple pages.[6]

MICROSOFT POWERPOINT 2003

Microsoft PowerPoint 2003 was used to create quick medium-fi prototypes to convey the look and feel of the application pages. While PowerPoint is a widely used and very powerful tool for presentations, it is equally used within the User Interface Designer community to develop and share quick blueprints of web based pages and applications.

ADOBE PHOTOSHOP CS2

Adobe Photoshop CS2 was used to create the TAMS logo and the icons used for the application.

CONCLUSIONS

CONCLUSIONS > LESSONS LEARNT

The TAMS project provided an excellent learning opportunity and brushing up most of the skills that we acquired over the 2 years in iSchool. The project involved all facets of complete project cycle from needs assessment, iterative user interface development, database design and development and final integration of database with the front end prototype. We relied heavily on the skills learnt in the IS213: User Needs Assessment, IS214: User Interface and Design, IS257: Database Management and UI tools. In addition, we utilized the project management and systems analysis skills learnt as part of the Effective Project management Class and IS207: Analysis of information Systems.

Not only did we utilize and hone our iSchool skills in working on this project, this served as a learning ground for new skills and techniques to manage the complexities of real life projects. Interacting with real clients added to the flavor of real world experience.

On a more personal note, both of our team members had come to iSchool from a technical background without much prior knowledge

[6] hotwired.lycos.com/webmonkey/reference/stylesheet_guide/; last access on April 28, 2007

about the field of Information Management, Usability, HCI and User Interface Design. Now at the end of our schooling here, this piece of project work where we successfully used new skills like Needs Assessment, Interface Design, Information Management and Project Management just to name a few, goes to prove the richness of the learning at iSchool.

For us the most rewarding piece of the project was the fact that our client was very happy with the product and is seriously considering one of our team members to work as a paid part time employee on integrating the current system with our project.

CONCLUSIONS > SCOPE FOR FUTURE WORK

Our client has been satisfied with our current development of the application. They are looking forward to the prospect of turning TAMS prototype to an actual fully functional, robust application that are integrated with the rest of the web applications used by the agency. The following are some of the major work to be completed for converting TAMS to a practical, efficient, robust application that provides optimal matching service tailored for our client.

Integrating with the real calendar system:

It would be convenient if TAMS could keep track of all the temporary rescheduling requests. For example, if a student is going on family vacation for two weeks in the first week of May, it would be nice that TAMS would remember that this student should resume tutoring when he/she comes back, and making sure that he/she is matched with tutors starting on the third week of May. Also, a feature to save the past tutoring matching sessions would be useful. This would allow the tutoring agency to keep track of tutoring history on students that can be used to evaluate student progress.

Integrating with the rest of the system:

The complete information management system of the tutoring agency contains several components: a registration system, a payment system, an enrollment database and a student tutoring status tracking system (This component has not been fully implemented. Currently, a tutor can store notes about a student using an online note-taking tool).

During the making of TAMS, we focused on the matching tool. However, the matching tool closely interacts with the other components. It takes in information about student profiles collected during the student registration process. Since the matching tool

generates tutoring matching configurations, it is closely linked with the information on tutoring sessions. As a result, our client has requested that the matching tool be integrated with the payment system, so they can easily access and view the payment information on the tutoring sessions.

The Matching Algorithm & Increasing Constraint Variables

The matching algorithm can be further optimized for more efficient processing. Further constraints could also be introduced into the system, such as matching ESL students to tutors who can speak their language. Another practical area is to allow matching of one tutor to multiple students, thus allowing the matching of group tutoring sessions.

APPENDIX [A]

Low Fi Prototype Interview Questions

The goal is to test to see how easy and intuitive this tool is for you to use. Please let me know if this tool is doing what you expected, and how I can improve this tool.

The Automatic Matching Tool

1. How do you think you can use this tool to:
    a. Cancel an appointment
    b. Reschedule an appointment
    c. Add a drop in student
    d. Adjust the Automatic Tutor/Student schedule match from December 2006 to January 2007?
2. How do you think the matching tool can be improved?

Student Section

1. How do you look up the tutoring availability schedule of a student?
2. How do you look up the tutoring history of a student?
3. How can the following pages be improved?
    a. Subject
    b. Schedule
    c. Tutoring History
    d. Miscellaneous
    e. Student Detail Page

Tutor Section

4. What do you look up the schedule information of say, a tutor named Lucy?
5. How can the following pages be improved?
    a. Subject page
    b. Schedule Page

Based on your observation of our interface, what additional things do you think you can do? Or the interface provides for you?

Do you have any other comments?

APPENDIX [B]

Details on Competitive Analysis

Scheduling Software

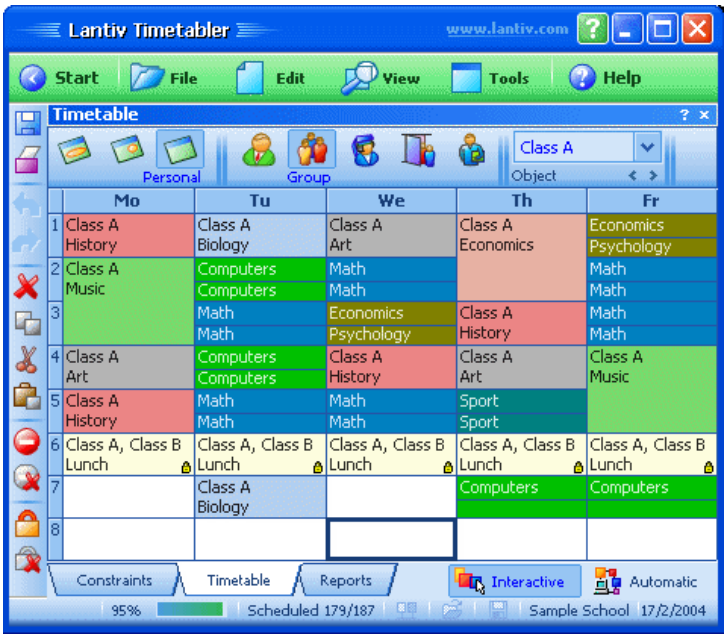
1. LANTIV Timetabler 6 - *www.lantiv.com*
2. Appointment Plus Tutor Scheduling - *www.appointment-plus.com*
3. Virtual Tutor Coordinator - http://www.tutorcoordinator.com/
4. Physician Scheduler® 6.0 - *www.physicianscheduler.com*

Scheduling Projects

5. Airline reservation
6. Campus connect

LANTIV Timetabler 6

http://www.lantiv.com/index.asp



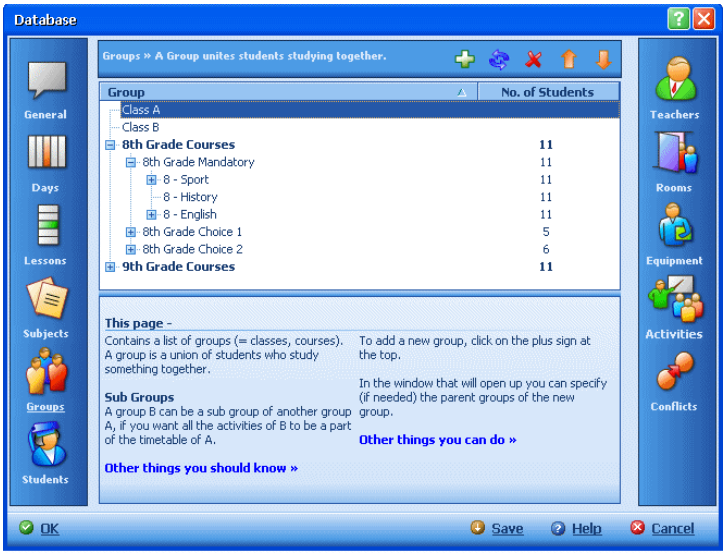
LANTIV Timetabler 6 is a fully automatic school scheduling software that caters to the timetabling needs of elementary and junior high schools. This software can be used to schedule teachers to classes to teach various subjects during a week, schedule courses according to student selection and enrollment, create custom study groups that combine several classes as a single lesson, create custom timetable for each student according to customized pedagogical constraints.

The Lantiv Timetabler 6 can work in 2 modes:

1. *Interactive mode* - The interactive mode facilitates manual scheduling of classes while the automatic mode schedules classes automatically using the Lantiv automatic engine.

2. *Automatic mode* - Although the automatic mode allows manual scheduling and indicates instances when there are scheduling conflicts, the user has to resolve the conflict manually. This in a way is similar to the paper based manual scheduling and requires a similar amount of effort. On the other hand the automatic mode can assign teachers to classes automatically depending on the constraints entered previously.

Cons:



- The interface makes use of icons for indicating various features. With the multicolor scheme used for distinguishing the classes and the large number of features and functions indicated by colorful icons makes the page look overwhelming.

- The various section tabs, function icons and navigation elements are distributed on the top, left and bottom of the page. This makes it a bit time taking to find any particular function tab for any new user. The class/teacher timetable is designed for the timetabling needs of elementary and junior high schools.

- The icons for the various database pages are distributed on the left and right bar of the page. Keeping all the icons/database pages clustered together would facilitate quick locating of any particular page.

What works well:

- The interface uses a multiple color scheme to indicate the various classes in the timetable. This color scheme technique worked well in this situation as it facilitates finding a particular class quickly by just looking at the color of the class. Although the colors used are not defined in any legend, the colors help in the distinguishing the classes and the use of any legend would be overkill in this case.

- All initial data is entered to the program in the Database window. The Database window includes the database pages for all the constraints like general details, subjects, groups, students, teachers etc. Based on the details and constraints entered in the database the timetable for the teachers and students is created by the software.

- The software stores data related to a large number of constraints.

Overall good features:

- Covers multiple domains (teacher/ classroom, tutor/group of students) and stores constraint details for various factors like students, teachers, lessons, subjects, rooms, groups, equipment, activities etc.

- Large number of useful features namely feature to fixate a particular activity such that the automation engine cannot move or delete a prescheduled activity, indication in cases were there are clash of schedules in the interactive more, indication of which teacher is free or which time slot is free etc.

- The interactive scheduling feature allows human intervention. This is one of the features that we would have loved to incorporate as part of this version of TAMS given more time. We understand that human intervention is an important part of scheduling.

Drawbacks:

- The Lantiv Timetabler is not web-based. It is available in CD or downloadable from the net on a particular machine. This limits the ability of this software to be accessed by an agency from multiple locations and having a central database repository for storing and updating information.

- This scheduling software cannot be easily integrated with the rest of the web applications used by the tutoring agency namely for viewing student history, tutoring history, payment system, student and tutor information.

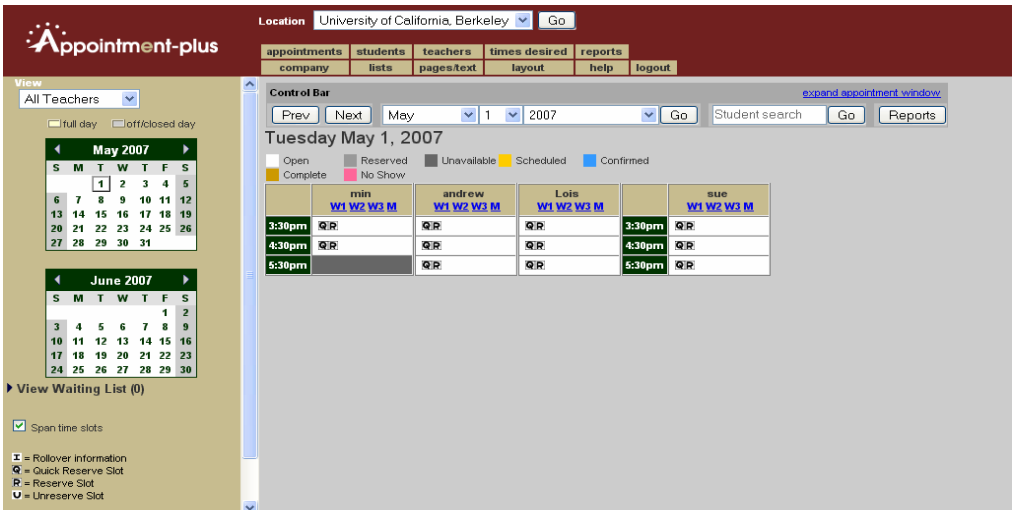
- The software appears to have a moderate learning curve due to the large number of features of the software along with the large number of terminologies and icons used.
- Although the software seems to consider multiple constraints, it was not clear if it actually does match individual students to tutors based on the availability, needs of the students and specialization of the teacher.

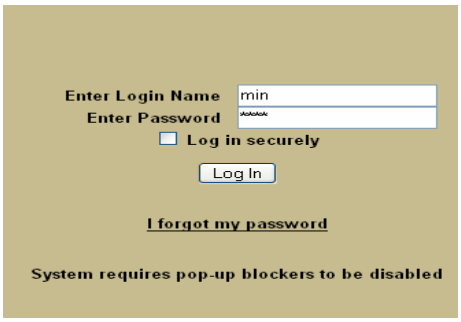
Appointment Plus Tutor Scheduling

http://appointment-plus.com/

Appointment plus tutor scheduling is an online appointment scheduling software for tutor and student scheduling. It is a part of Appointment-Plus Scheduling Software distributed by Contemporary Web Plus, Inc., a software development company based in sunny Scottsdale, Arizona. The online nature of the software allows tutors and students to access schedules via unique login and passwords and view their schedules and make new appointments.

To understand the features of the tool better, we also registered for the one month of free trial of the software.





Login name and password ensures privacy and security of the data.

Good features:

- This is a web-based software and thus facilitates easy access from multiple places and also maintain a central repository of information.
- Calendar on the left bar in the appointment view is very useful. Clicking on the individual dates on the calendar opens up the schedule for that day on the right.
- The schedule shown is linked with the calendar displayed on the left bar. This allows the user to see the schedule of any day just by clicking on the particular day on the calendar.
- This is another feature that we would like to integrate in our future work.

Drawbacks:

- Has only 'day-views' for schedules and does not provide a weekly view.
- User interface is confusing in many areas.
- Clickable links are not differentiated.
- Terms and keywords used in some sections are not clear: e.g "quick reserve slot".
- Uses color coding to differentiate between reserved, unavailable, unscheduled, open, confirmed slots in the schedule and the legend of the color codes is given at the top of the schedule. The color coding for this does not seem very useful and the user needs to remember the color in the legend to understand the particular slot in the schedule.
- Codes like I, q, R , U are used in the table with the legend described at the bottom. This again needs one to look back and forth or remember the particular code as the code names are not intuitive.

Conclusions:

Appointment plus is a very useful and flexible online software and has many features which are similar to the TAMS application. However, as opposed to the TAMS application this software assumes that the all the tutors have similar capabilities and teaching skills and therefore does not provide any comprehensive information about the tutor skills. The application and many features all packed into one but the use of many of them are not apparent.

Virtual Tutor Coordinator

http://www.tutorcoordinator.com/

Virtual Tutor Coordinator, an on-line program that automates and facilitates activities for most tutoring systems. It is designed to meet the scheduling needs in colleges, universities and tutoring centers. It automates and simplifies many tasks which are done manually by the scheduling coordinator in any tutoring agency. It gives students and tutors access to meeting information, personal details and meeting schedules and gives the tutor coordinator the complete control on the tutoring system.

(Partial Selection of the Student Sign Up Form)

(Partial Selection of Control Booth)

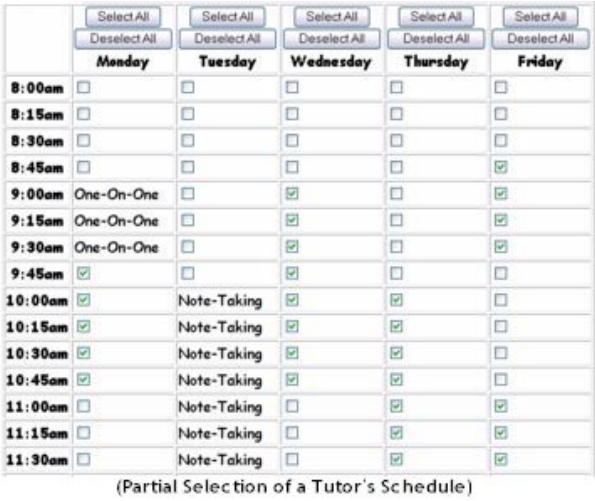
Pros:

- Virtual Coordinator is web based and works with all browsers.

- A number of features that would be very useful if integrated in TAMS namely Meeting options that fit a student's schedule, E-mails for tutors and students with weekly tutoring schedules.

Cons:

- High installation/maintenance cost, can't be easily integrated with the existing tutoring agency software. It cost $2,000 for the software and an additional $30 per month if the system is hosted on the virtual coordinator server.
- Interface is functional but not very user friendly. The first impression a user gets is of just text boxes and drop down boxes.
- The software assumes that all the tutors can teach all the subjects or can just be used for single subject tutoring. There is no provision to enter the tutor constraints.



	Select All / Deselect All	Select All / Deselect All	Select All / Deselect All	Select All / Deselect All	Select All / Deselect All
	Monday	Tuesday	Wednesday	Thursday	Friday
8:00am	☐	☐	☐	☐	☐
8:15am	☐	☐	☐	☐	☐
8:30am	☐	☐	☐	☐	☐
8:45am	☐	☐	☐	☐	☑
9:00am	One-On-One	☐	☑	☐	☑
9:15am	One-On-One	☐	☑	☐	☑
9:30am	One-On-One	☐	☑	☐	☑
9:45am	☑	☐	☑	☐	☐
10:00am	☑	Note-Taking	☑	☑	☐
10:15am	☑	Note-Taking	☑	☑	☐
10:30am	☑	Note-Taking	☑	☑	☐
10:45am	☑	Note-Taking	☑	☑	☐
11:00am	☐	Note-Taking	☐	☑	☑
11:15am	☐	Note-Taking	☐	☑	☑
11:30am	☐	Note-Taking	☐	☑	☑

(Partial Selection of a Tutor's Schedule)

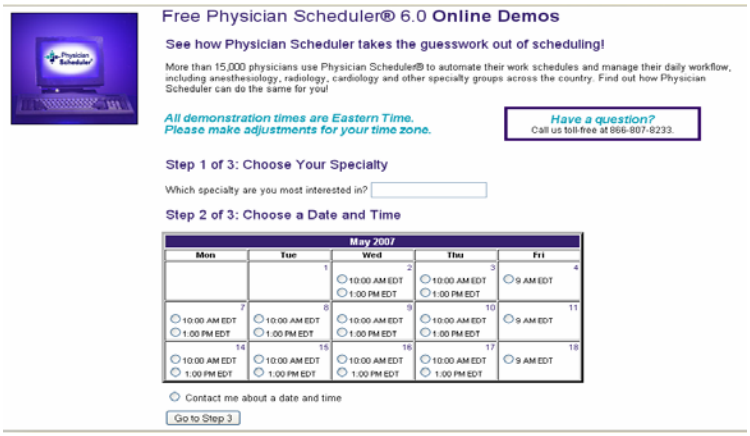
We believe that the scheduling problem is not just prevalent in tutoring agency activities but is a problem faced by many other organizations and there has been various attempt to find the optimal solution. In our attempt to look at solutions organizations other than tutoring agencies we looked at the physician scheduling software.

Physician Scheduler® 6.0

www.physicianscheduler.com

Physician scheduler is an automated physician scheduling software.

The heart of the system is the rules-based AutoStaffing engine, which makes each assignment taking into account fairness, coverage requirements, personnel conflicts, staff requests, and limits.



Pros:

- Optimizes coverage for all sites based on the unique skill sets of physicians, it also makes sure that each physician receives a fair rotation and workload.
- The system ensures accuracy and control through alerts that warn of any potential conflicts and errors (such as double-booking or under-staffed facilities) whenever schedule changes are made.

Cons:

- This is a one way scheduler in which the patients have to look up for the doctors available and schedule an appointment. This scheduling method maybe useful in doctor/patient scheduling environments but may not be optimal for tutoring agency administrators who need to schedule students to tutors.
- The interface is functional but not user friendly.
- Training required.
- Purchasing and continued maintenance, updating and training costs involved.

Airline reservation System

This was a database project done by 3 students in IIT Bombay as part of their Database course. The project attempted to model the working of an on-line airline reservation system.

This model finds the optimal flight schedule by taking into consideration multiple constraints depending on the passenger's needs and the availability of seats and schedule of the slight.

Pros:

- Was a good example of a multivariate scheduling system.
- Handles flight schedules across multiple airports with multiple stops.
- Although the constraints considered in this scheduling system was different than TAMS, the database description and SQL queries gave us an idea about how to approach our scheduling problem.

Cons:

- As it was a database project it did not provide any ideas for the user interface.

Campus connect

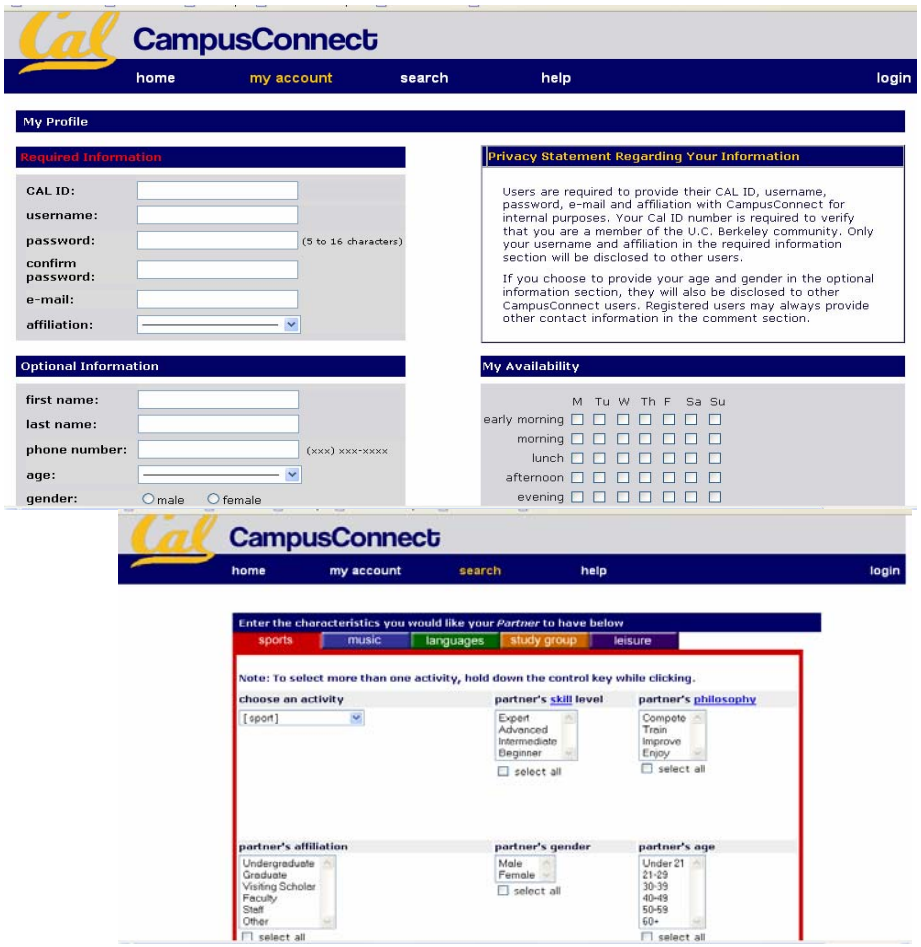
Campus connect is an activity partner/matching site done as a Masters project in the i-school a few years back. Campus students can search for other students to form a group for any collaborative activity such as sports , study, music etc.

Pros:

- Simple intuitive user interface

Cons:

- The match in this system is done based on some static variables (like interest, which once stored in the system will not need to be changed.



APPENDIX [C]

User Needs Assessment Questionnaire

TAMS (Tutoring Agency Management System) Project

Interview Questions

Questions about Tutoring Agency

1. How many kids are currently enrolled?
2. Can you tell us something about the Tutoring Agency - what kind of organization it is? About the general demographics of the tutors and students

Background Questions

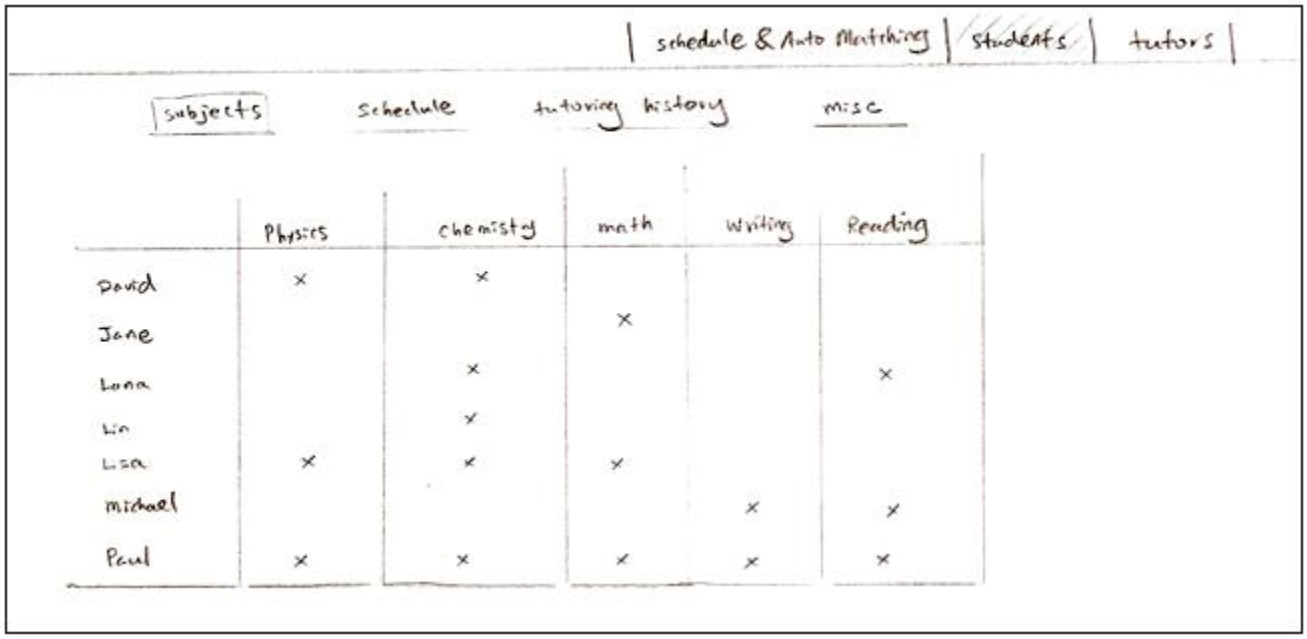
1. How long have you been working with the agency?
2. What is your comfort level/experience with computers? What is your educational background? – (Qualifications, certificates, degrees etc)
3. What are your responsibilities?
4. How many kids are generally in one group?
5. Can you walk us through your typical schedule in the agency?
6. How do you do scheduling of the classes?

Questions on the current system

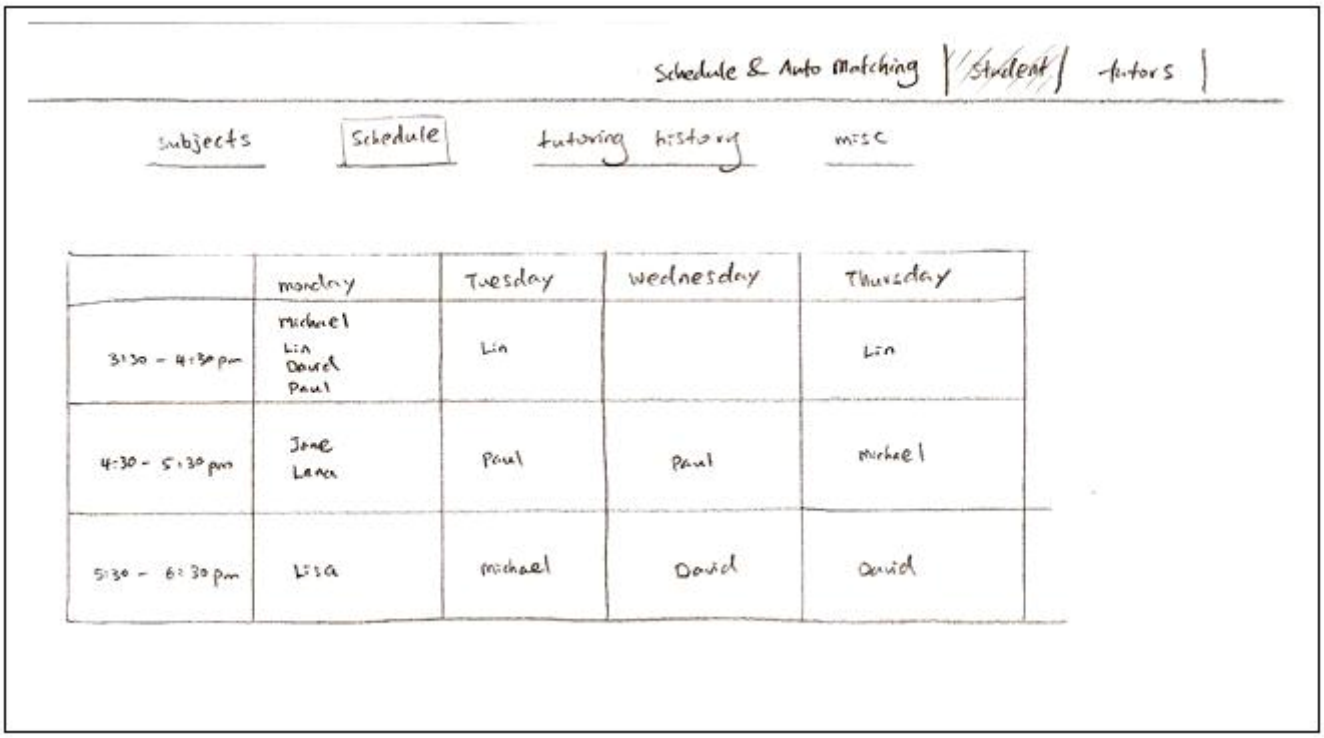
1. Can you describe the process – after a student registers online then how do you enroll the students in your system? is it manual?
2. How do you keep the information updated?
3. How do you schedule? How is a tutor assigned to a particular student? What are the general criterions that you take into consideration?
4. Problems that you face with the present manual system.
5. Frequency of errors? Any particular areas which are error prone?
6. Which task is most challenging/ time consuming/ error prone?
7. What do you think are the major issues with the present system? – (headache prone areas; bottlenecks)
8. Do you have any technical help or any technical team who takes care of your website and other technical issues that may arise?
9. What are the major areas that you think will be most convenient if automated?

APPENDIX [D]

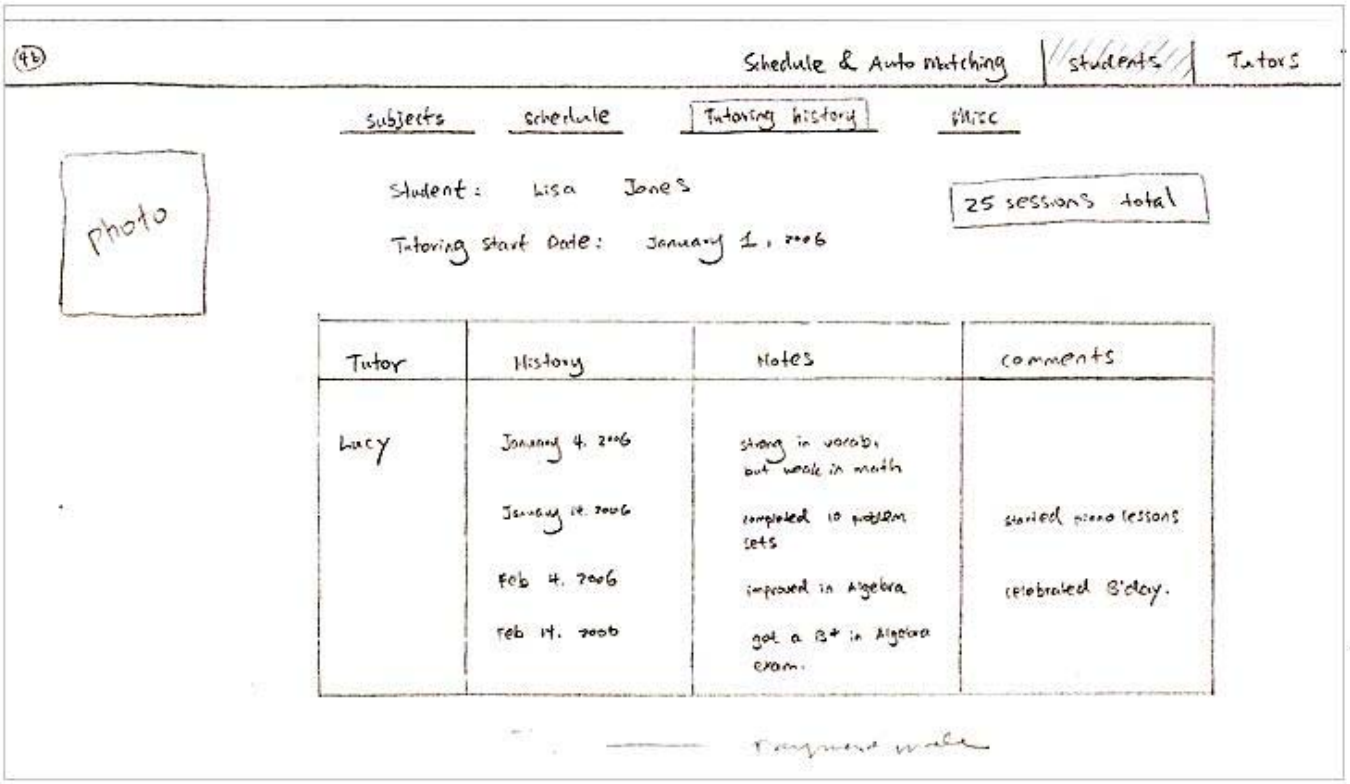
Initial Sketches & Low fi Prototype



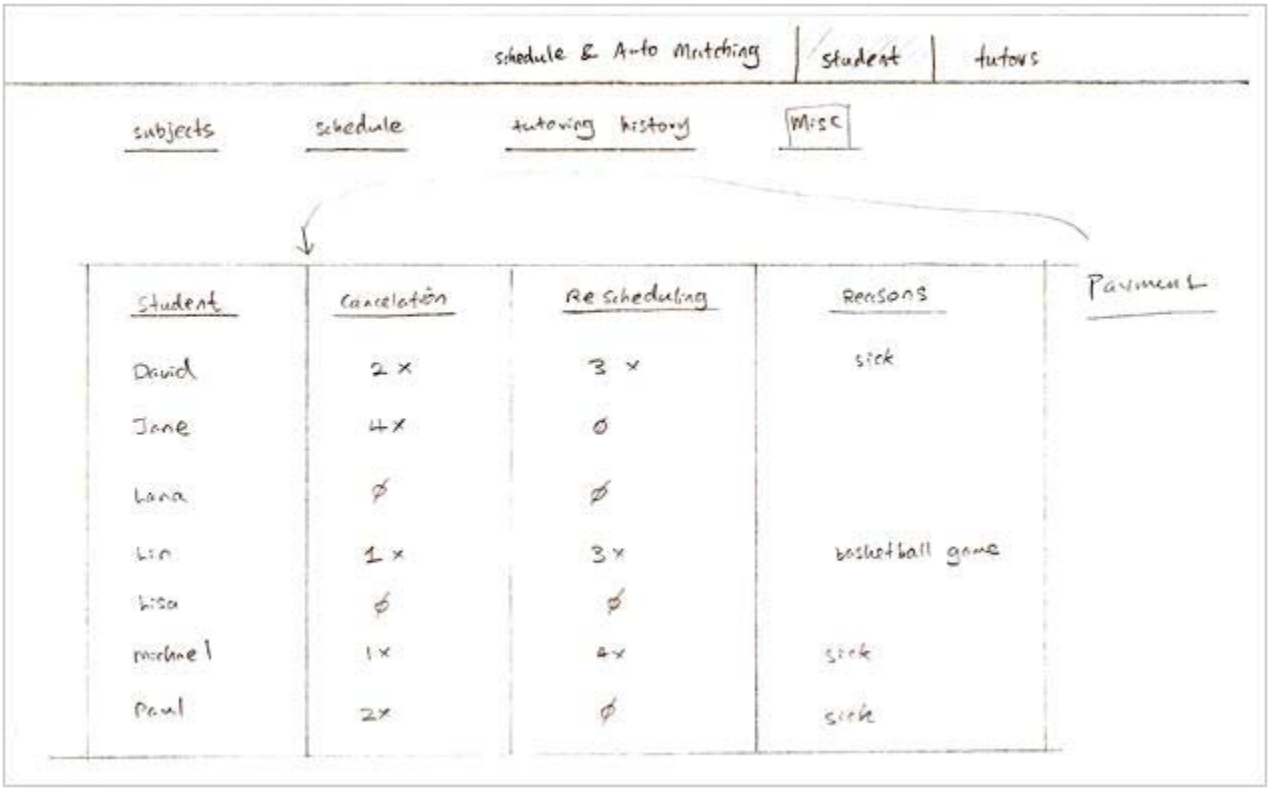
Collecting tutoring subject constraints from a student



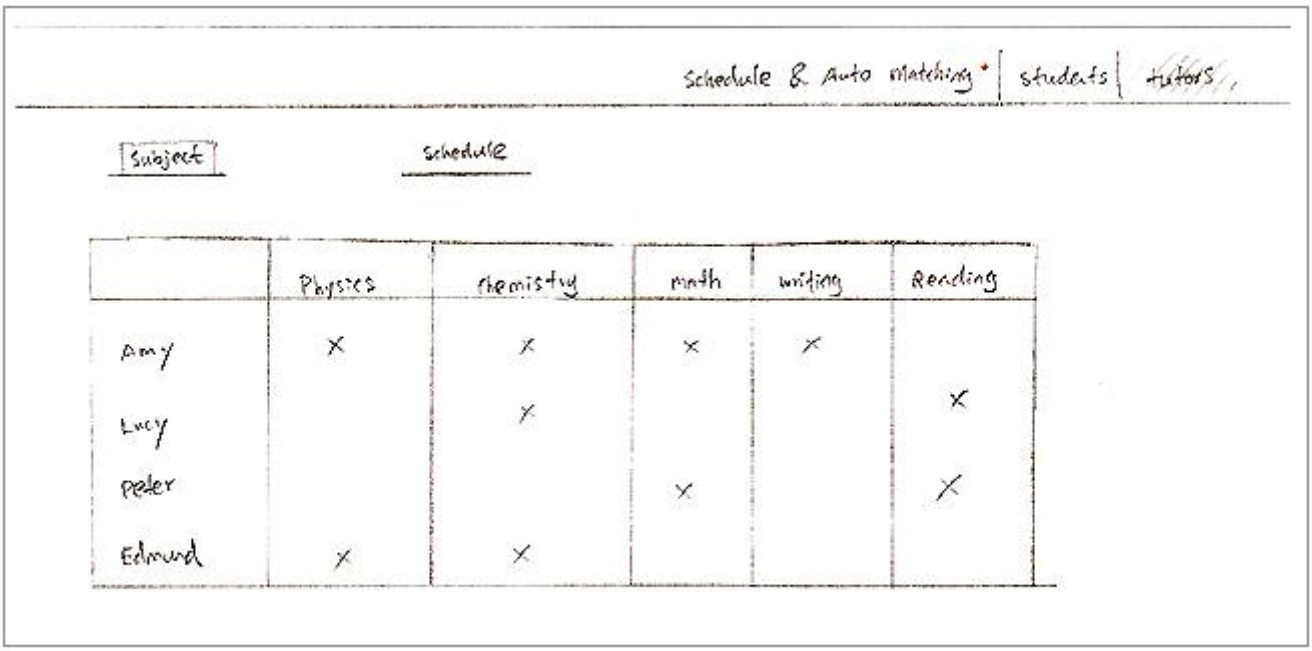
Collecting student schedule constraints



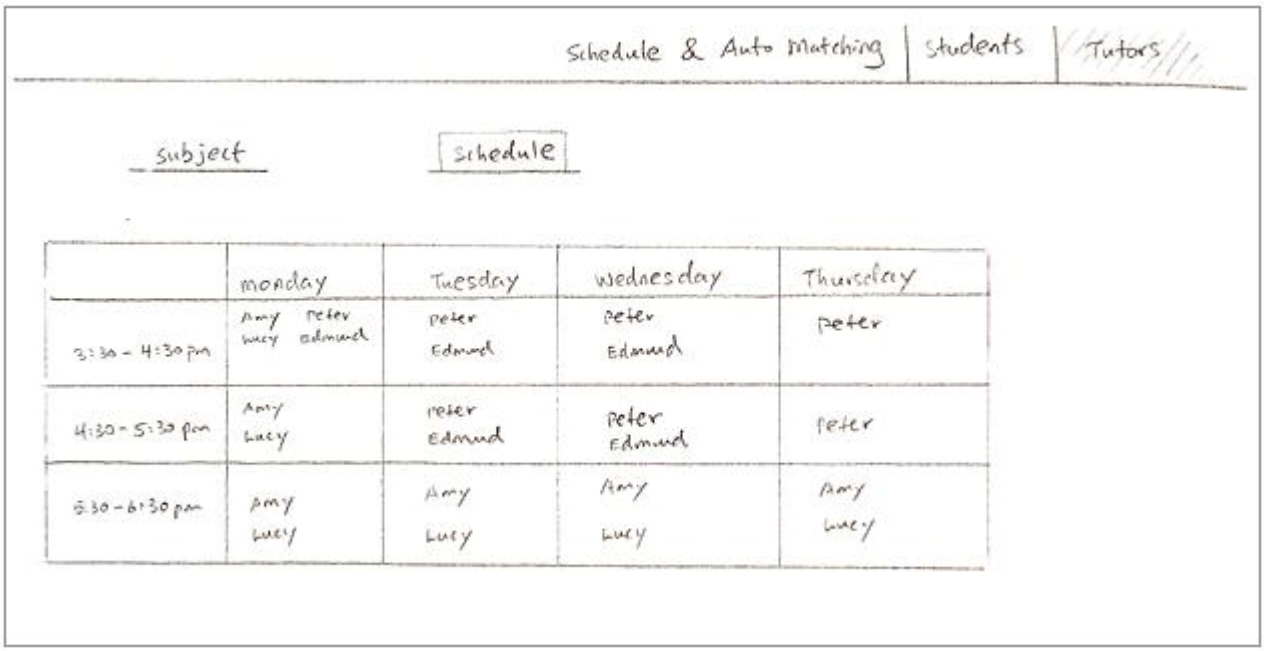
Tracking student history (per student)



Viewing student tutoring cancellation and rescheduling details



Collecting tutoring subject constraints from tutors



Collecting tutor's schedule constraints

APPENDIX [E]

First Interactive Prototype

Main Menu

Schedule & Auto Matching
Students
Tutors
Payment Summary

Students Schedule | Students Subject | Tutoring History | Misc

Student Subject

	Physics	Chemistry	Math	Writing	Reading
David	*	*	-	-	-
Jane	-	-	*	-	-
Lana	-	*	-	-	*
Lin	-	*	-	-	-
Lisa	*	*	*	-	-
Michael	-	-	-	*	*
Paul	*	*	*	*	*

Student Subject Details

Main Menu

Schedule & Auto Matching
Students
Tutors
Payment Summary

Students Schedule | Students Subject | Tutoring History | Misc

Student Tutoring History

Student	Tutors	History	Notes	Comments
David	Peter	3 hours	Need to do more physics problem sets	Seems to work with Peter well
Jane	Lucy	5 hours	Seems to have no problem with algebra, but need help with trig.	ESL student
	Amy	10 hours	Improved in the speed of doing problem sets	-

Student Tutoring History

Main Menu

Schedule & Auto Matching
Students
Tutors
Payment Summary

Students Schedule | Students Subject | Tutoring History | Misc

Student Tutoring History - Detail on a Student

Student: David Lee
Tutoring Start Date: October 4, 2006
Total Hours of Tutoring: 4

Tutors	History	Notes	Comments
Peter	October 4, 2006	strong in vocab, but weak in math	-
	October 14, 2006	completed 10 problem sets	Started basketball training
	November 4, 2006	improved in Algebra	celebrated birthday!
	November 17, 2006	got a B+ in Algebra practice test	

Student Details

Main Menu

Schedule & Auto Matching
Students
Tutors
Payment Summary

Tutor Schedule | Tutor Subject | Add/Edit Tutor

Tutor Schedule

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

Tutor Schedule

Main Menu

Schedule & Auto Matching
Students
Tutors
Payment Summary

Tutor Schedule | Tutor Subject | Add/Edit Tutor

Tutor Subject

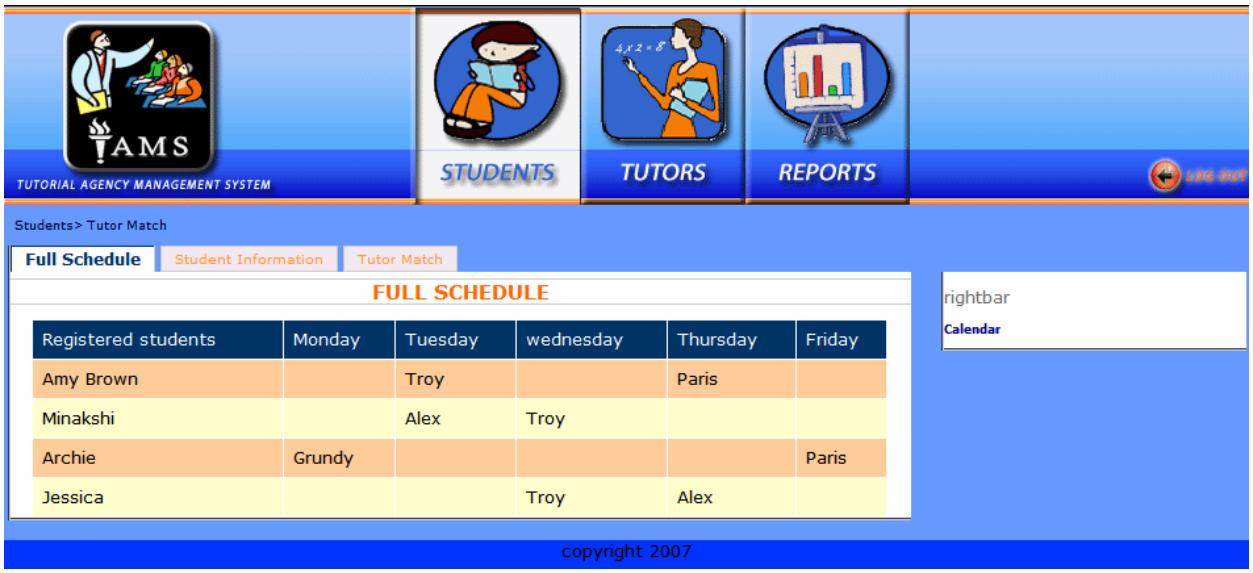
	Physics	Chemistry	Math	Writing	Reading
Amy	*	*	*	*	-
Lucy	-	*	-	-	*
Peter	-	-	*	-	*
Edmund	*	*	-	-	-

Tutor Subject Details

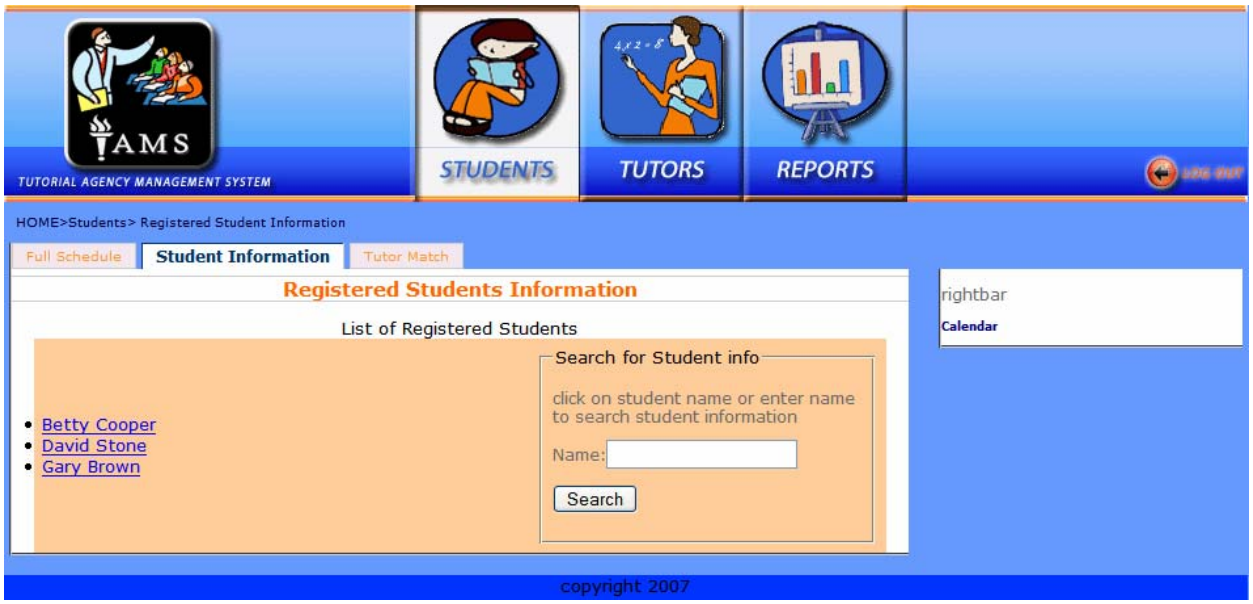
APPENDIX [F]

Second Interactive Prototype

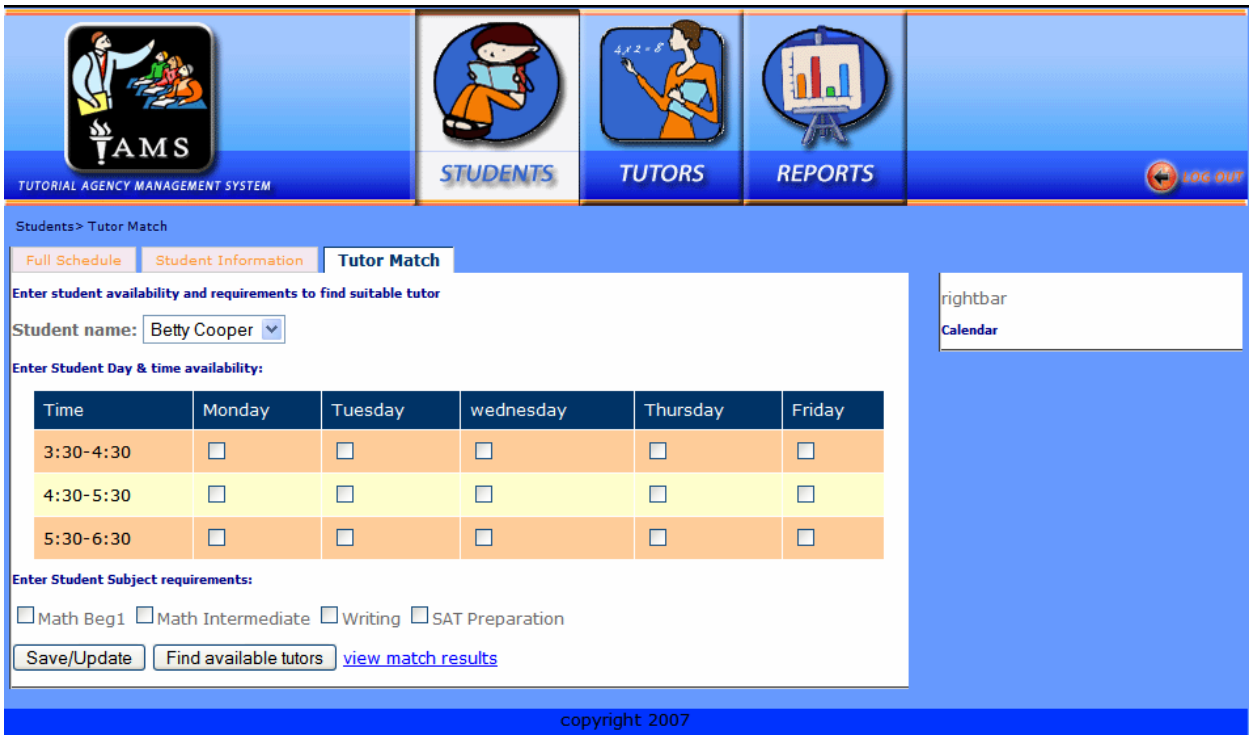
Home page with TAMS login feature



Students Full Schedule



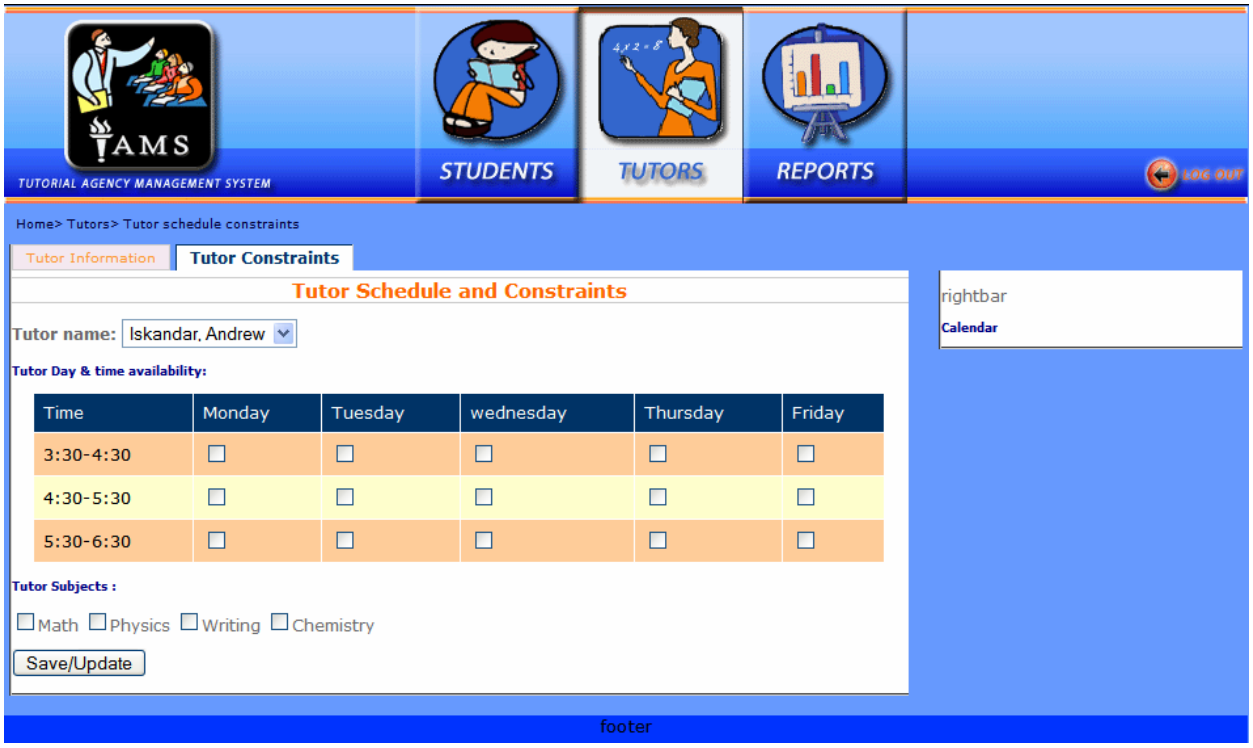
Registered Student Information



Tutor Match

Student Details

Current Tutor



Tutor Schedule & Constraints