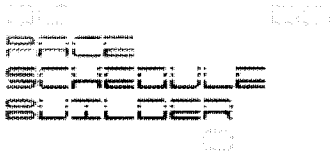

Pace University
New York, New York

CS 389
April 5, 2004

Viktor Geller Naseer Haniff Nikita Lukish

Feasibility Analysis v1.2

Pace Schedule Builder ("PSB")

Table of Contents

Introduction

All over the world, in many universities, every year and every semester each and every student shares a specific struggle. This struggle can be relieved with a simple solution that may make their lives much easier. Moreover, it will permit them to manage their time much more easily. What our group is referring to is the simple stressful season of registration times. This process can be very tedious and stressful to any student especially when given difficult scenarios and choices to deal with. What the Pace Schedule Builder ("PSB") will accomplish is the simplification of the process of putting together a compact college student's schedule. Furthermore, we will make registration fun, enjoyable and most of all simple and less stressful at the same time. Schedules will be previewed for the students at the touch of a button and all the manual work, such as paper and calculations will be avoided when many various schedules have been conjured up together by our software. In summary, the Pace Schedule Builder will:

- Eliminate the routine and tedious work of manually rooting out class conflicts in the schedule
- Automatically create personalized schedules based on constraints and preferences specified by the user
- Automatically generate a visual representation of the schedule(s) which can be sent to a printer or saved for later use

Right now, no similar software is publicly available, although some universities may have similar components in their internal systems.

Overview

This is the Feasibility Analysis second draft for the Pace Schedule Builder, due on Monday, Mar 8th. This document addresses four feasibility perspectives:

- Technical Feasibility
- Financial Feasibility
- Time
- Resource feasibilities

Technical Feasibility

This project will be built primarily using the Java programming language and will incorporate Sun's Java-based technologies and APIs such as JSP, Servlets and JDBC.

We will definitely make the interface as user-friendly as possible. Because the main purpose of the software was to relieve stress, the software itself shouldn't and will not be complicated or stressful to use or activate. It will simply be ineffective if the key aspect was to relieve tension but creates tension in the process of being used. Moreover, it will be enjoyable to use with many options that all have simplicity in their design meaning that it will be very easy to operate (user friendly), and nothing will be of extreme complexity that will confuse the users.

A database will be established to hold all login information for the users as well as any schedules that they have designated to save. The database connectivity will be done via JDBC, of Java API.

The system will be integrated with the Pace ClassSchedule system as a means of getting course data and registration information. It is entirely possible for other universities to write their own versions of this component to work with their databases.

Other Technical Issues

Although we will be using a variety of technologies to develop our software, it will not be too complex for anyone from a CS background to alter or maintain. We as the developers will work towards one of our goals to make the design, thus its' coding and creation modules as simple as possible to make the approach on improvement much easier. Although that will be one of our goals, most software that is simple to use and user friendly are not always simply coded or understood.

Financial Feasibility

Financial Feasibility is not particularly applicable to this project, since it is being developed as an academic undertaking.

We will use Pace University's Matrix system, which makes a variety of technologies and services freely available to Pace students. Open source technologies such as Apache web server and Tomcat servlet engine are well

suited for our needs, as the application that we intend to develop will not require high-performance commercial web servers.

However, there are some financial concerns about fulfilling software and hardware requirements for developing, executing and maintaining the project.

In terms of cost-benefit analysis, it would be cheaper and more efficient to utilize freeware tools that come with basic services such as *Tomcat* rather than spending money on commercial solutions such as *IIS* and *Web Logic* that provide analogous services.

Time Feasibility

Overview

The allocated project timeframe, which is approximately 3 months (one 13 week semester course), should be sufficient time for complete delivery of the application. The scope of the project and available resources make project development within this timeframe possible.

To meet the deadline the project will be broken into milestones so that each part of the application development life cycle is done within its allocated period. The milestone breakdown is as follows:

Monday 29th March:
Design document due

Monday 19th April:
Compilable code due
Informal demonstration of your software

Monday 26th April:
Tested code and test plans due
I must be able to test your software with the test plans you will submit without problems.

Wednesday 28th April:
Customer surveys - The impact of your software will be measured by surveying users of your software or/and your clients. You will have to find 4 students/faculty/staff with Pace emails who will use your software, answer a survey, and give their opinion on the software. They will send me an email directly. We will discuss this activity in class.

Wednesday 5th May:
Formal Presentation and Demonstration of the project in front of a committee composed of invited guests
As-built version of Feasibility Study, Requirement Document, Design Document, Test Plan Document, Customers surveys, Screen Snap shots of the software in use and Source Code

Time Concerns

The main time concerns are that team members:

- Hold part-time jobs that make harder to schedule meetings that are at convenient time for each member of the team.
- Have families and associated commitments

While no major problems are anticipated because of the above noted facts, they are important issues to consider when planning the development of the project and also when distributing the work amongst team members.

Resource Feasibility

Overview

The development team will consist of 4 people who will all share responsibilities of a project manager, developers and documentation writer. The team generally has experience with working on complex software projects in the past. The team's technical skills and commitment are the biggest factors that will ensure a timely and a successful delivery of the project.

The team will need to be familiar in the following areas/skills:

- JSPs and Servlets
- Tomcat or Weblogic web servers
- Databases (needed for storing intermediate data)
- HTML
- CASE tools (Visual)

Team Member Profiles

Viktor Geller

Viktor has been employed for 3 years with the NYC branch of Canadian Imperial Bank of Commerce as a J2EE developer, working with JSP/Servlets, and various database management systems (Sybase and Oracle). Prior to this he has worked as a Perl developer in the UNIX environment at Citigroup. At both companies, his duties focused mainly on porting mainframe and other legacy systems to the web. Viktor is a BS CS major and a junior at Pace.

Naseer Haniff

Naseer is a CS BS major, a junior and currently employed in the Computer Lab. He is the Assistant Programmer and also Senior Consultant. His knowledge is in technologies such as Java, Unix, Visual Basic, and Servlets. Naseer is also very experienced with Microsoft Office. Furthermore, he maintains and puts together networks of computers at Pace and is always updating OS, and software systems.

Nikita Lukish

For over a year, Nikita has been employed with TRADEPAQ Corp., a provider of collaborative supply chain execution solutions. He currently works as a Mumps developer. Skills needed for his work include Mumps programming language, Javascript and Cache relational database. Via his education at Pace, Nikita has some experience with Java, Servlets, Apache Web server and Tomcat servlet runner. He is a senior.