

The WPI Scheduler Project

An Interactive Qualifying Project Report

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

Degree of Bachelor of Science

by

Cyle Anthony Hawkins

Peter Nicholas Lucia

Graciela Rubio

Date: December 13, 2007

Professor Michael Ciaraldi, Advisor

## **Abstract**

The WPI Scheduler Project aimed to develop a new course scheduling system for the students of Worcester Polytechnic Institute by modifying and extending the open source University Scheduler originally developed by Keith Lea. This course scheduling system will provide students and their advisors an easier, more convenient method of creating course schedules for an entire academic year.

## **Acknowledgements**

We would like to thank Keith Lea for granting us permission to expand the University Scheduler to meet the needs of Worcester Polytechnic Institute.

We would also like to thank the administration of Worcester Polytechnic Institute, especially Dale Snyder and Charles Kornik, for endorsing this project as a great new tool for all WPI students and their advisors.

Special thanks to Paula Delaney of the Information Technology division for helping us develop this system using current course schedules and aiding us to integrate this system with the Banner Web Information System.

We would also like to thank our advisor, Professor Michael Ciaraldi, for all of his encouragement, guidance, and support throughout this project. We truly appreciate all that you do.

## Table of Contents

Abstract .....	i
Acknowledgements .....	ii
Table of Figures .....	vi
1. Introduction .....	1
2. Methodology .....	3
Software Engineering Techniques .....	3
2.1 Requirements Management .....	3
2.2 Programming Techniques .....	4
2.3 Design Patterns .....	5
3. Research .....	6
3.1 Survey Questions .....	6
3.2 Survey Results .....	7
3.2.1 Question One: Ease of Current Course Scheduling System .....	7
3.2.2 Question Two: Class Meeting Time versus Class Availability .....	8
3.2.3 Question Three: Entire Academic Year Verses Individual Terms .....	9
3.2.4 Question Four: Time Spent Creating Schedules .....	11
3.2.5 Question Five: Changing Course Schedules after Registration .....	11
3.2.6 Question Six: Personal Computers and Java .....	12
3.2.7 Question Seven: Consulting Advisors .....	13
3.2.8 Question Eight: Calendar Views of Course Schedules .....	15
3.2.9 Question Nine: Terms Shown in a Calendar View .....	16
3.2.10 Question Ten: Class Occupancy and Course Selection .....	17
3.3 Survey Feedback .....	18
3.3.1 Experiences with Previous Course Scheduling Systems .....	19

3.3.2	Suggested Features for the WPI Scheduler .....	19
3.4	Meetings with WPI Administrators .....	20
4.	Design.....	22
4.1	Scheduler Window and Default Sub Panel .....	22
4.2	Course Selection Panel .....	23
4.2.1	Department Box .....	23
4.2.2	Search Box .....	23
4.2.3	Possible Courses List .....	23
4.2.4	Course Information Pane.....	24
4.2.5	Selected Courses Table .....	24
4.3	Narrow Down Panel .....	25
4.3.1	Year Time Selector.....	26
4.3.2	Time Selector .....	26
4.3.3	Section Tree.....	26
4.4	View Schedules Panel .....	27
4.4.1	Mini Schedules Pane.....	27
4.4.2	Schedule View.....	28
4.4.3	Year Detailed Schedule Grid.....	28
4.4.4	Detailed Schedule Grid .....	28
4.4.5	Legend Pane .....	28
4.4.6	Section Details Pane.....	29
4.5	Listing Panel.....	29
4.5.1	Term Information Panel .....	30
5.	Implementation.....	31
5.1	How to Build .....	31

5.2	How to Run.....	31
6.	Future Work .....	33
	Appendix A: Survey Questions.....	34
	Appendix B: Configuration File and Java Web Start JNLP File .....	37
1.	wpi.schedui File: .....	37
	Appendix C: Sample ScheduleDB File.....	44
1.	Explanation File .....	44

## Table of Figures

Figure 1: Ease of Use of Current Scheduling System.....	7
Figure 2: Class Time Verses Class Availability .....	9
Figure 3: Planning Classes for the Entire Academic Year Verses Each Term .....	10
Figure 4: Hours Spent Creating Schedule.....	11
Figure 5: Number of Times Schedules Were Changed.....	12
Figure 6: Java and Personal Computers .....	13
Figure 7: Students Who Consulted with Their Advisors .....	14
Figure 8: Calendar View of Course Schedules .....	16
Figure 9: Calendar View Options.....	17
Figure 10: Class Occupancy.....	18
Figure 11 - Course Selection Panel.....	25
Figure 12 - Narrow Down Panel .....	27
Figure 13 - View Schedule Panel.....	29
Figure 14 - Listing Panel.....	30

## **1. Introduction**

The WPI Scheduler Project aimed to develop a new course scheduling system for the students of Worcester Polytechnic Institute by modifying and extending the open source University Scheduler developed by Keith Lea. The current scheduling system at WPI presents students with a textual listing of course offerings for the entire academic year and allows students to develop their class schedules manually before registering using the Web Information System, also known as Banner Web. Although students can find the course schedules online, this system is not automated.

The problem with the current system is that students must first determine in what term the specific course is offered and what class times are available for each of their courses. Students must then create a class schedule by determining which class sections do not conflict with other classes. Many students create spreadsheets or tables to visualize when a class meets and which other classes will fit into their schedule. This process is a tedious exercise in trial and error: guessing what courses might fit, drawing those courses out on a schedule, before picking a few more courses only to find out that there are time conflicts between the courses you have chosen.

As a graduate student at Rensselaer Polytechnic Institute, Keith Lea developed the University Scheduler to help fellow students create their course schedules. Although RPI does not support or advertise this program for student use, on November 15, 2007, Keith Lea reported that students ran the RPI Scheduler 24,000 times within the previous four weeks. This course scheduling system allows students to select courses for a single semester while notifying them of which class sections conflict with other sections. After selecting the desired courses for the semester, the program displays the possible schedules and the course registration numbers for those schedules. Students can then register for the schedule they like most.

In extending and adapting the University Scheduler for Worcester Polytechnic Institute, the WPI Scheduler has become a course scheduling system that handles schedules founded on a multi-term system. This allows students to select classes for an entire academic year instead of for each individual term. Through future work and collaboration with faculty members and the



administration, the WPI Scheduler may allow students to consult with their advisors electronically when developing their course schedules.

The process of calculating schedules for multiple terms as opposed to a single term or semester presents many additional challenges. Whereas the largest block of time that must be considered in a single term system is a week, an entire year composed of multiple terms, each with their own weekly schedule must be considered when doing a multi-term system. Also, the sheer number of schedule possibilities tends to increase when you allow a longer period of time to be scheduled at once. Finally, the complexities of allowing a user to determine which courses need to be taken during a specific term or before another course introduces an additional challenge

The WPI Scheduler Project also aimed to extend the University Scheduler so that it can be easily configured and used by other colleges and universities. The scheduling system can be configured to have a different number of terms and work with the registration and course instructions of that university.

## **2. Methodology**

To develop a successful course scheduling system, the WPI Scheduler Team utilized several tools. The team met with their advisor on a weekly basis, asked other students to test the application, and surveyed the student body. The team heavily relied on software engineering techniques and the use of good software design patterns when developing this application.

### **Software Engineering Techniques**

#### **2.1 Requirements Management**

Good software engineering techniques necessitate frequent meetings with the customer to ensure that the product is on track and if it is not, that cuts are made to the appropriate areas. Since the team, as WPI students, was both the primary customer and the developers, their advisor provided an excellent sounding board as a ‘customer’. Monday afternoons were set aside by the WPI Scheduler Team to meet with their advisor, Professor Michael Ciaraldi. The team presented their current progress with the project and areas where they were experiencing difficulties. Professor Ciaraldi provided suggestions and guidance on how to proceed with the project as well as suggestions regarding what considerations should be made in the design of the program. This time allowed the team to reflect on what had worked during the past week, and what should be focused on during the next week.

This process was aided greatly by a tool known as SourceForge. SourceForge is a web application that provides a code repository and many features for running a project such as task organization and requirements tracking. The team used these resources as a way to keep track of features needed in the long term or problems that we needed to address later. We took entries that we had added to this system, and ranked them with a priority and an estimate of how long we thought they would take to implement. This allowed us to realistically track what could be accomplished and what needed to be reconsidered.

Many of the requirements and tasks put into SourceForge were derived by testing with WPI students. Throughout the development process, the WPI Scheduler Team asked other students to use the WPI Scheduler and provide feedback. These users were drawn from among

our friends, coworkers, and other students who were working in the lab. The consistent feedback allowed the team to make subtle changes to enhance usability.

In addition to the feedback obtained from small scale application testing, the WPI Scheduler Team surveyed the entire student body to determine which features and functionality would be most desirable in a scheduling tool. This was done early in the development process to ensure that this project was actually needed and that it would address the most frequently requested features. The survey and its results are discussed in further detail in section 3.1.

## **2.2 Programming Techniques**

The WPI Scheduler is rather large program, and a program that is likely to be extended by others in the future. As such, it was very important that good programming techniques be followed. These techniques included but were not limited to: following good design principles, always having a working codebase, and utilizing pair programming. How the team attempted to follow good design principles is discussed in the next section, but first a look at the other techniques used.

The WPI Scheduler Team attempted to always have a working version of the code available in their version control system. This was not an easy task and in fact the team was not able to stick to this goal. Twice during the project, there was an extended period of time when the code in the version control system was not working. The first time was when the objects representing time were reworked from having a week as their largest unit to having a year as their largest unit. This transition was not a simple one and required hundreds of small changes before the program would properly compile. After this was done, the scheduler worked properly with the exception that all terms were displayed in the same week view. The second time this process broke down was when the switch was made from week to year in the user interface. This too required hundreds of changes before the code would once again compile.

The complexity of the Scheduler and the number of changes that had to be made to it lent itself very well to the technique of pair programming. A majority of the work done on the WPI Scheduler by the team was done during the team's daily meetings, where they set out a task and pursued it, as a team working at one computer, until completion. Pair programming allows one programmer to "drive" for a time while the other developer "navigates" pointing out errors,

coming up with ideas, and maintaining a perspective on the big picture. Pair programming also allows one developer to focus on the design of the code while the first continues to complete the lower level here and now programming. This technique is most useful when the two programmers frequently switch roles so both can maintain a grasp on the big picture and each have a chance to do low level programming of the kind they do best. Nearly all of the work done to the WPI Scheduler by the team was done through pair programming.

The technique of Pair Programming also facilitates the learning aspect of the Interactive Qualifying Project. One developer, more experienced in a particular area, teaches the other using code that must be written anyway and has a visible purpose, which is far greater than learning the theory from a manual or even through examples in isolation. It is also important to note that this is not one developer exclusively teaching the other, this happens between developers who have approximately the same experience levels and who both take turns teaching and learning. This allows the team members not only to be more knowledgeable about the code base but also allows the team to have a more knit feel where working late, is not as much of a chore as a communal feeling of accomplishment on the horizon.

### **2.3 Design Patterns**

Good design patterns are absolutely necessary for any program that is larger than a trivial example. A design pattern is a known solution to a common problem. The University scheduler was written in Java, which is a very object oriented language. Design patterns for object oriented programs include the idea that each object should only have one purpose and do that purpose well. This idea was already woven into the Scheduler and made the extension of it much easier. The design patterns we encountered and used are far too numerous to include in this paper, so one will be expounded as an example.

The WPI Scheduler Team developed a large section of the revamped multi-term code using data stored in a Singleton object implemented with the Singleton Pattern. The Singleton Pattern allows a developer to ensure that only one copy of an object ever exists. This is utilized in the storing and recalling of the how the year is configured, ie: how many terms there are, and how many part of terms each term has. This information is not going to change once it is setup and we can save memory by simply having one class store this information.

### **3. Research**

As work began extending the University Scheduler a survey was sent to all students, both undergraduate and graduate, via e-mail. This survey attempted to find out if the need really existed for a new scheduling system and as well as to provide insight regarding how students currently create their course schedules. The survey results strongly confirmed the need for an improved course scheduling system at WPI. The survey results also offered many suggestions and ideas as to what WPI students would like in a course scheduling system.

In addition to students, administrators, staff, and faculty members also provided insight and suggestions to what features they would like to see in a scheduling system. In particular, suggestions were geared toward ways to allow for greater and easier collaboration between students and their faculty advisors.

#### **3.1 Survey Questions**

The survey, which can be found in Appendix A, included ten multiple choice questions and three optional questions. The optional questions were used for student feedback and classification purposes. The survey asked students what they think of the current course scheduling system, how they plan their class schedules, what importance they place on class availability and class meeting times, and how many times students changed their schedules. Students were also asked if they had met and consulted with their advisors regarding their course schedules, what they would like to see in a calendar view of course schedules, and if they have Java installed on their personal computers.

The three optional questions asked students if they have used other course scheduling programs, what they liked or disliked about them, and what they would like to see in a scheduling program for WPI. The final classification question asked students if they were an undergraduate or graduate student and how many years they have been studying at WPI as an undergraduate student.

### 3.2 Survey Results

The survey was sent to all WPI students, and the response was encouraging. Approximately 10% of students responded to the survey and the analyses of the three hundred thirty-two responses strongly support the need for a new course scheduling system. The following sections discuss the response for each individual question and the resulting analysis.

#### 3.2.1 Question One: Ease of Current Course Scheduling System

The current course scheduling system at WPI involves determining what courses a student would like to take within the academic year, finding out which terms each course is offered in, before looking up the course meeting times from the Office of the Registrar’s website. Students must then create their course schedules by hand, manually avoiding time conflicts. To make this task somewhat easier, students tend to create spreadsheets, tables, and other calendars to graphically view possible schedules.

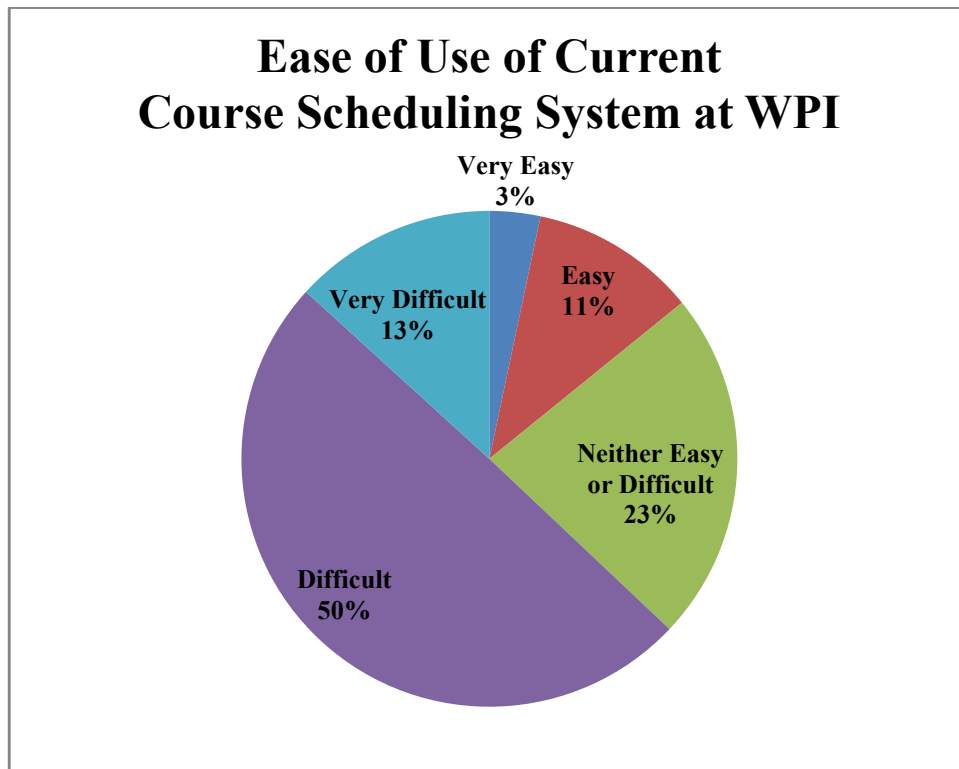


Figure 1: Ease of Use of Current Scheduling System

Students were asked to rate the ease of using this current course scheduling system as being either very difficult, difficult, neither easy nor difficult, easy, or very easy. As seen in Figure 1, one hundred and sixty-two students, approximately fifty percent of the total responses, rated the current system as being difficult. Forty-seven students, or fourteen percent, said that the current system was either easy or very easy to use. With such a large percentage calling the current scheduling system difficult to use, it is clear that students need an easier solution.

### **3.2.2 Question Two: Class Meeting Time versus Class Availability**

At WPI, each course is categorized as being either a Category I or Category II course. A Category I course is offered at least once a year, while Category II courses are only offered every other year. Many introductory, Category I, courses are offered multiple times in a year. For example, the four introductory physics courses are offered twice a year, and mathematics courses such as statistics and differential equations, are offered every term. However, there are several introductory courses that are only offered once a year.

Since courses are only offered a specified number of times a year, students must determine what courses they must absolutely register for and when the course is offered. Students can then use a course's available meeting times to determine which sections will fit into their schedule. Many courses, especially introductory mathematics courses and foreign language courses, are split into sections with different meeting times so that class sizes remain small. Other courses may only consist of large lecture classes, while other courses will consist of large lectures with smaller lab and conference sections.

Students were asked how a class' meeting time and class availability affect course selections for their schedules. In this case, class time refers to the time of day a class meets and how many times it meets per week, while class availability refers to the number of times a course is offered in one academic year. Students selected one of the following six options: mostly based on class time, somewhat based on class time, equally based on class time and class availability, somewhat based on class availability, mostly based on class availability, and neither based on class time or class availability. The graphical distribution of responses can be seen in Figure 2.

As hypothesized, students tend to strongly consider a course's availability when creating their course schedules. Approximately thirty-five percent, or one hundred and nineteen students,

said that class availability impacted their course schedules the most, while twenty-four percent of students surveyed said they equally consider class time and class availability when creating their course schedules.

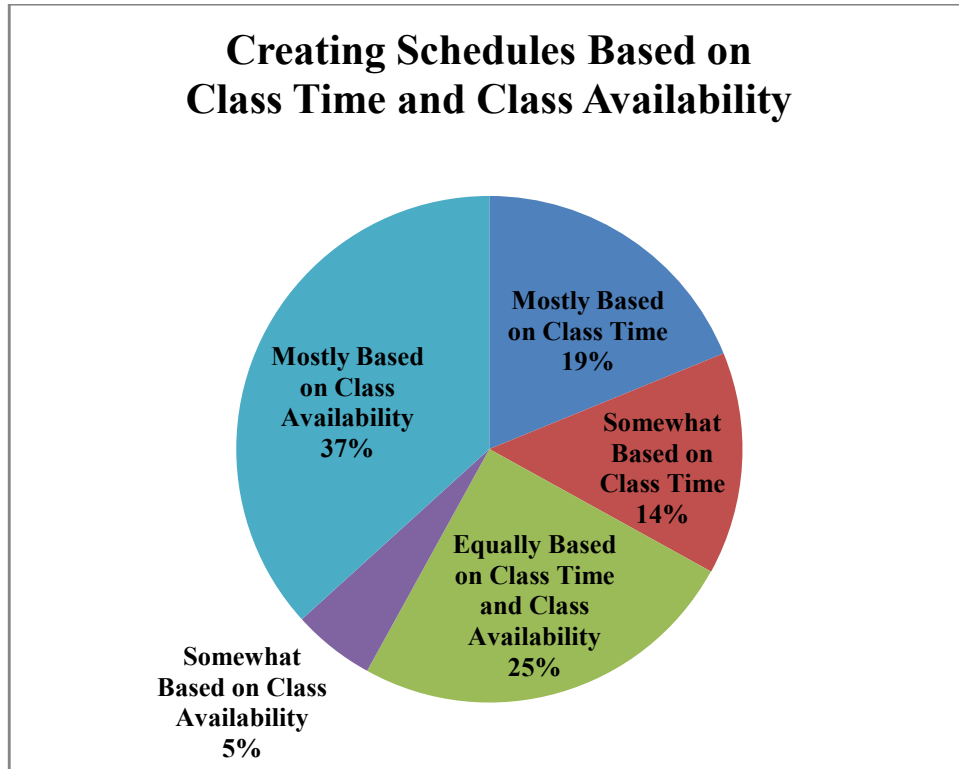


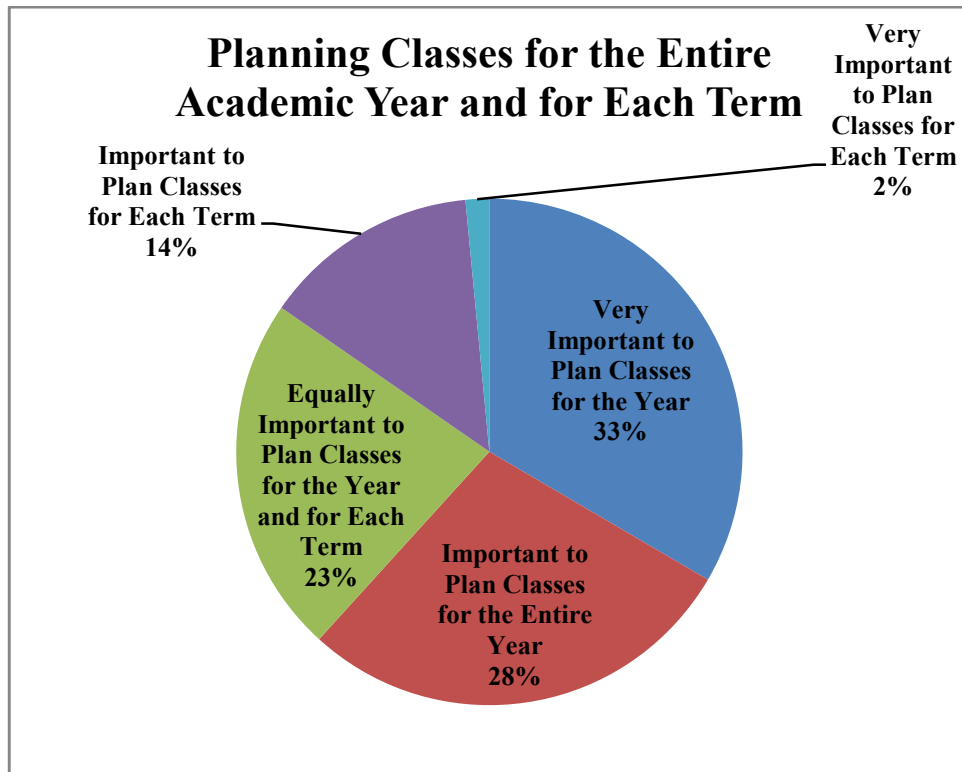
Figure 2: Class Time Verses Class Availability

### 3.2.3 Question Three: Entire Academic Year Verses Individual Terms

Many students tend to plan their course schedule for the academic year while others create their schedules for each individual term. Since many classes are only offered once a year, students typically create their course schedule for the entire academic year. To verify this information, students were asked to rate the importance of developing a course schedule for the entire year verses planning a course schedule for each individual term. Students responded with one of five options: very important to plan classes for the entire year, important to plan classes for the entire year, equal importance to plan classes for the year and for each term, important to plan classes for each individual term, and very important to plan classes for each individual term. As seen in Figure 3, the resulting data and graphical analysis clearly shows that the majority of



students prefer to create their course schedules for the entire year rather than for each individual term.



**Figure 3: Planning Classes for the Entire Academic Year Verses Each Term**

The reasons for planning a course schedule for the entire academic year are plentiful, yet include two distinct reasons. First of all, many of the courses at WPI are only offered once a year or once every other year. This forces students to register for that course when it is offered instead of planning their schedule for the entire year and assuming that the courses they selected are offered at least once that year and do not conflict with other desired classes.

A student's Interactive Qualifying Project (IQP) and Major Qualifying Project (MQP) can also affect a yearly course schedule, especially if the student wishes to complete the project off-campus. If students are studying at an off-campus project center, they must devote an entire seven week period to their projects. However, students who complete their projects within Worcester can span their project over two to three terms which allows them to concurrently take classes that may only be offered in a specific term or every other year.

### 3.2.4 Question Four: Time Spent Creating Schedules

Students tend to spend many hours developing their course schedules for the upcoming academic year. Students were asked how many hours they spent creating their course schedules and responded using one of the five following options: less than an hour, one to three hours, three to six hours, six to eight hours, and more than eight hours.

As seen in Figure 4, the majority of students, over sixty-eight percent, spent between one and six hours creating their course schedules. Approximately twenty-five percent of students spent six or more hours creating their schedules. These results strongly indicate that students need a more efficient course scheduling system.

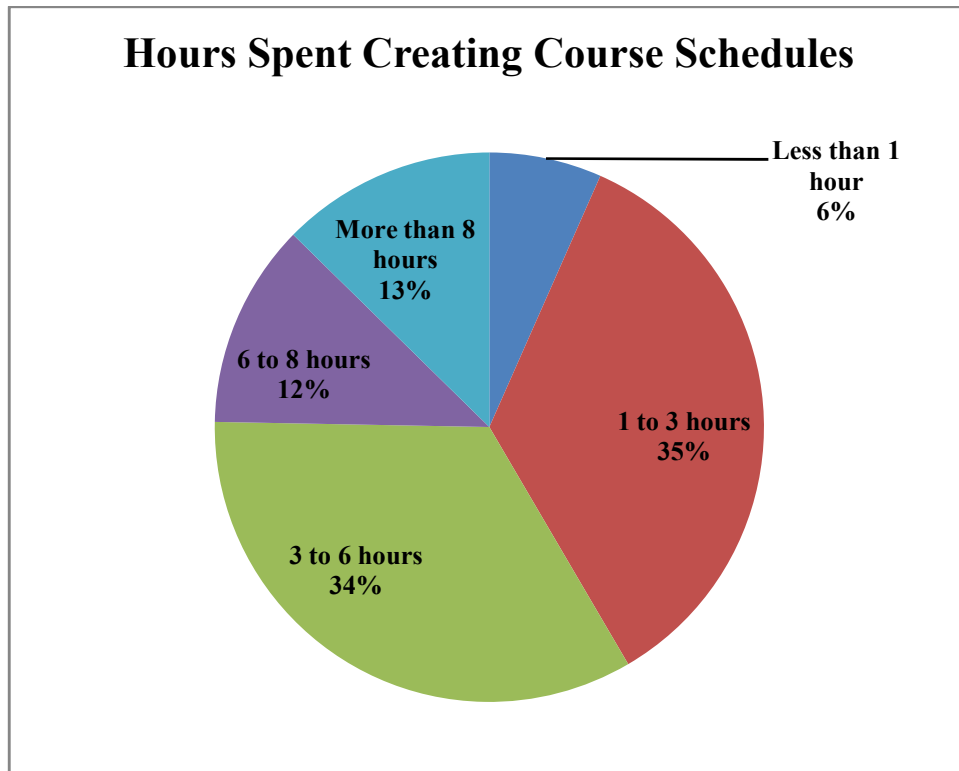


Figure 4: Hours Spent Creating Schedule

### 3.2.5 Question Five: Changing Course Schedules after Registration

After registering for classes for the upcoming academic year, students may change their course schedules either at the beginning of each term or at the beginning of each semester. Only sixteen percent of the respondents stated that they did not change their course schedules

after registration. Ninety-eight students, or thirty percent, stated that they have changed their course schedules at least once. As shown in Figure 5, seventy-three students, or twenty-two percent, said that they changed their schedules three times last year, while seventy-one students, or twenty-one percent, changed their course schedules twice.

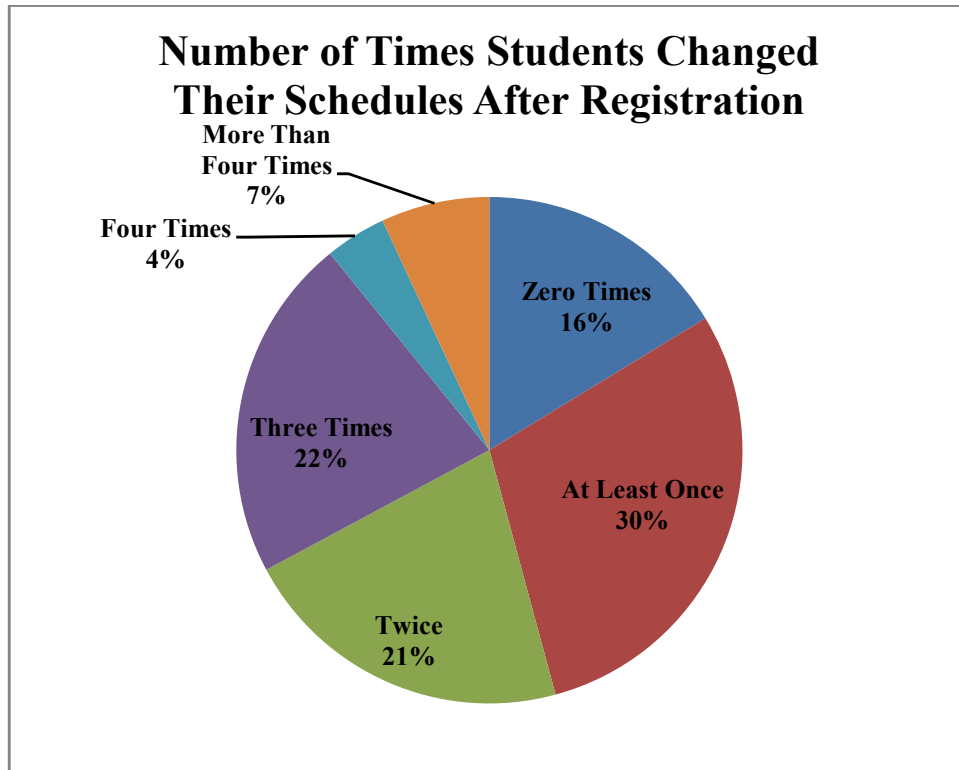


Figure 5: Number of Times Schedules Were Changed

### 3.2.6 Question Six: Personal Computers and Java

Because the WPI Scheduler is written in Java, students will need to have Java installed on their personal computers in order to run the scheduling program. Although Java is used in many web-based applications, it is possible that not all students will have Java installed. The Team hypothesized that most students will have Java installed, but to verify that students will be able to use the WPI Scheduler on their personal computers, students were asked if they have Java installed. Students could choose from the following four answers: yes, no, I believe so, and “What’s Java?”

Over sixty-six percent of students responded as having Java installed on their personal computers, while seven and a half percent do not have it installed. As seen in Figure 6, almost

eighteen percent of students believe they have the software installed, while the remaining eight percent, or twenty-seven students, do not know what Java is or what it does.

Since the majority of students do have Java installed on their personal computers and since Java is a commonly used web application, the WPI Scheduler will run properly on the majority of students' computers.

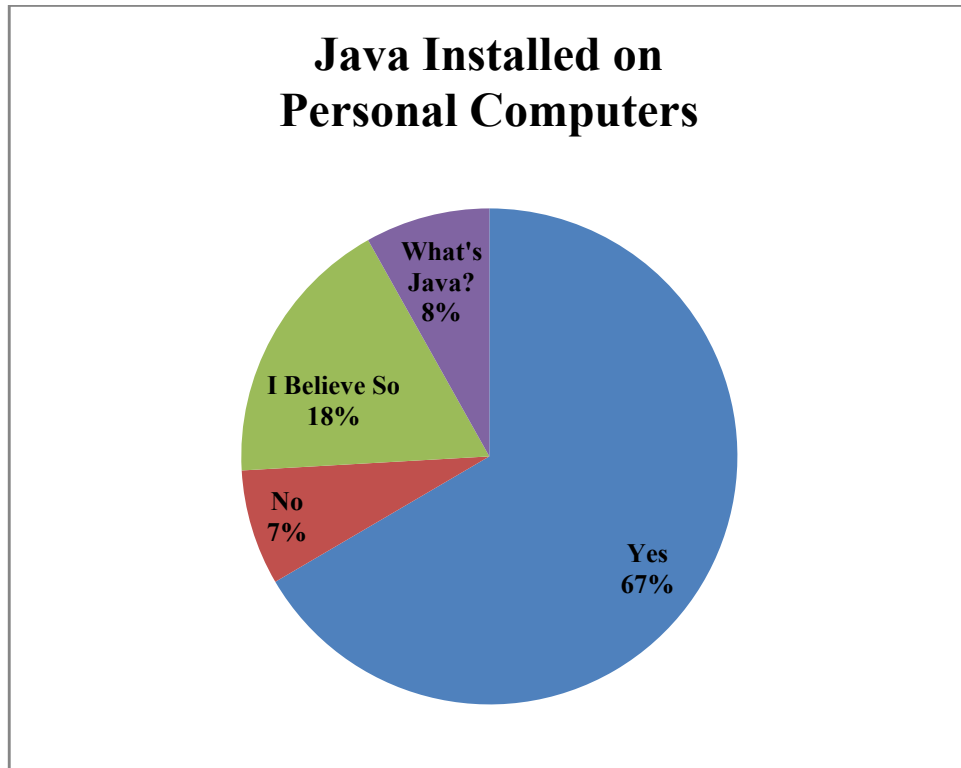
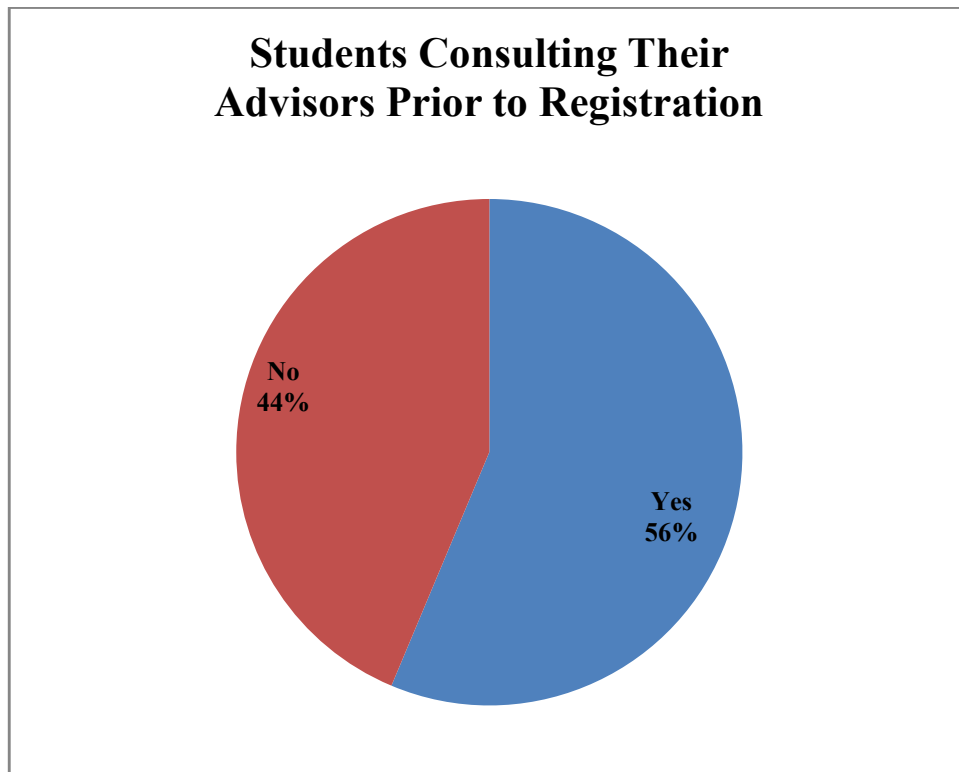


Figure 6: Java and Personal Computers

### 3.2.7 Question Seven: Consulting Advisors

Every year, the administration of WPI designates a day where classes are not held so that students and their advisors can meet. Academic Advising Day is typically held in February so that students can consult with their advisors and discuss potential course selections for the upcoming academic year. While freshmen are typically obligated to meet with their advisors, many upperclassmen also take advantage of this day to verify that they are fulfilling the graduation requirements, especially those related to their major.

The electronic survey asked students if they consult with their advisors prior to registering for classes for the upcoming school year. As seen in Figure 7 below, approximately fifty-six percent of students said that they do meet with their advisor, while the remaining students do not meet with their advisors.



**Figure 7: Students Who Consulted with Their Advisors**

The WPI Scheduler aims to help facilitate meaningful conversations with advisors by allowing students to investigate possible schedules quickly and easily before meeting with their advisor. As seen in earlier results, students tend to spend a long time trying to determine what schedule will work; often this process cannot be completed before meeting with an advisor. With the WPI Scheduler, the process hopes to be so quick that changes can be made on the fly in the room with the advisor, and alternatives to conflicting classes can be discussed on the spot.

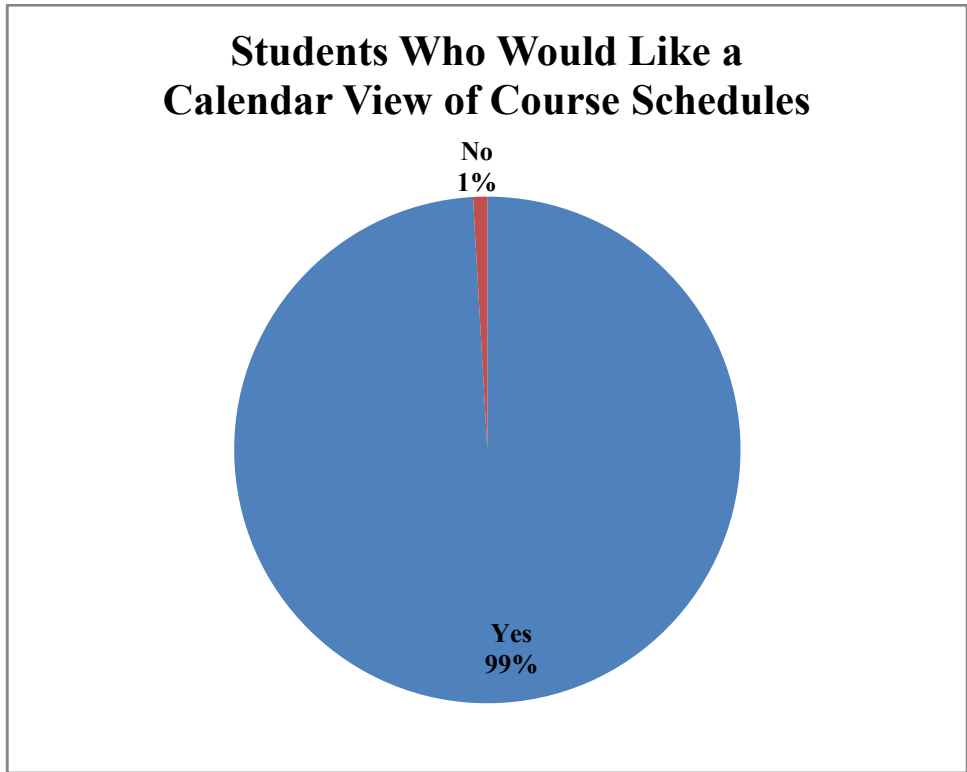
In the future, the WPI Scheduler also hopes to strengthen the communication between advisors and advisees by providing an electronic means of sharing course schedules. Within the WPI Scheduler, students can save their preferred schedule to an XML file. This file could be sent to the advisor who could view the advisees' proposed schedule in a graphical format and

then make comments regarding the schedule. The increased communication will greatly benefit students, advisors, and many members of the administration.

### **3.2.8 Question Eight: Calendar Views of Course Schedules**

One of the most difficult aspects of the current course scheduling system at WPI is that students cannot view a calendar that displays the days and times a class meets during a typical week. It is the WPI Scheduler team's belief that many students work better when they graphically see a schedule instead of simply looking at a list of numbers. A calendar view would also represent each section of the course since each section may have a different lecture, a different conference time, or a different lab time. The downside to having a calendar view of every course and its sections is that there are many different sections with different schedules.

To verify that students would appreciate a weekly calendar view of a course and its sections, students were asked if they were interested in such a graphical interface. Of the students surveyed, over ninety-nine percent of students agreed that a calendar view would be beneficial. As seen in Figure 8, the response was overwhelming with only three students responding with the answer no.



**Figure 8: Calendar View of Course Schedules**

### **3.2.9 Question Nine: Terms Shown in a Calendar View**

The original University Scheduler was only able to generate schedules for one term. The problem with only being able to view one term at a time is that students cannot create their schedules for the entire year and see what options are available to them. For example, if a student wanted to take a mathematics course that is offered twice a year, he would need to know which term he would take the course instead of seeing all of the options for the entire academic year. If this course conflicted with another course this term alternatives of taking it another term would not be considered. This is particularly true for courses that are only offered once a year or once every other year because students must register for that course when it is being offered.

Since students want a graphical view of their schedules and after an overwhelming approval for a calendar view of schedules, students were asked how many terms they would like to see in a calendar view of the course schedules. Students were able to choose from the following three options: one term at a time, two terms, or a semester, at a time, and all four terms at once. As seen in Figure 9, approximately fifty percent of students would like to view their

course schedules for the entire academic year, or all four terms, at once. Yet, approximately twenty-six percent of students would still appreciate viewing each individual term by itself. The main conclusion is that students want a new graphical course scheduling system.

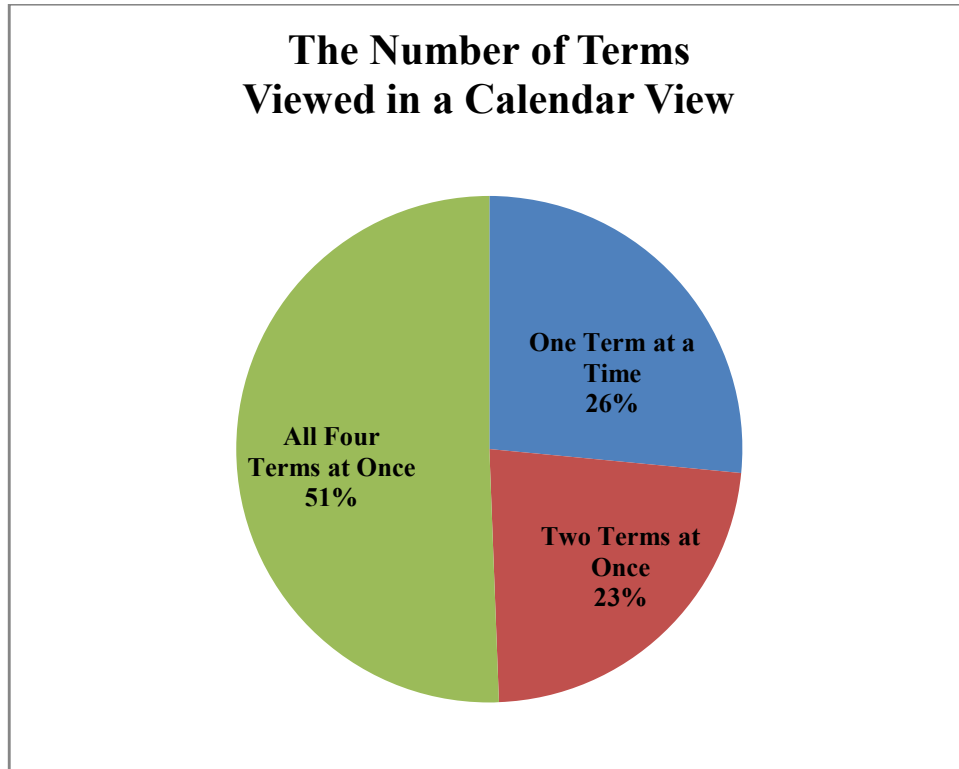


Figure 9: Calendar View Options

### 3.2.10 Question Ten: Class Occupancy and Course Selection

While WPI boasts a thirteen to one student to faculty ratio, many introductory courses have up to two hundred seats available. The final multiple choice question asked students how often a course's occupancy, or number of students registered for a class, affects their decision to enroll in the course. Students were asked to choose from the following four options: at least once a term, at least once a semester, at least once a year, and never. Approximately seventy percent of students said that a class' occupancy has never affected their decision to enroll in the course. However, as seen in Figure 10, approximately fifteen percent of students said that the occupancy of a course affects if they will enroll in the course at least once a year.

If a course has reached its maximum enrollment capacity, a student may be waitlisted for the course and may need to select another class for that term. Unfortunately, waiting lists are



inevitable due to scarcity of classrooms and the maximum capacity of a classroom. The WPI Scheduler has the capacity to display the number of seats in a section and how many of them are occupied, but this functionality has not yet been implemented.

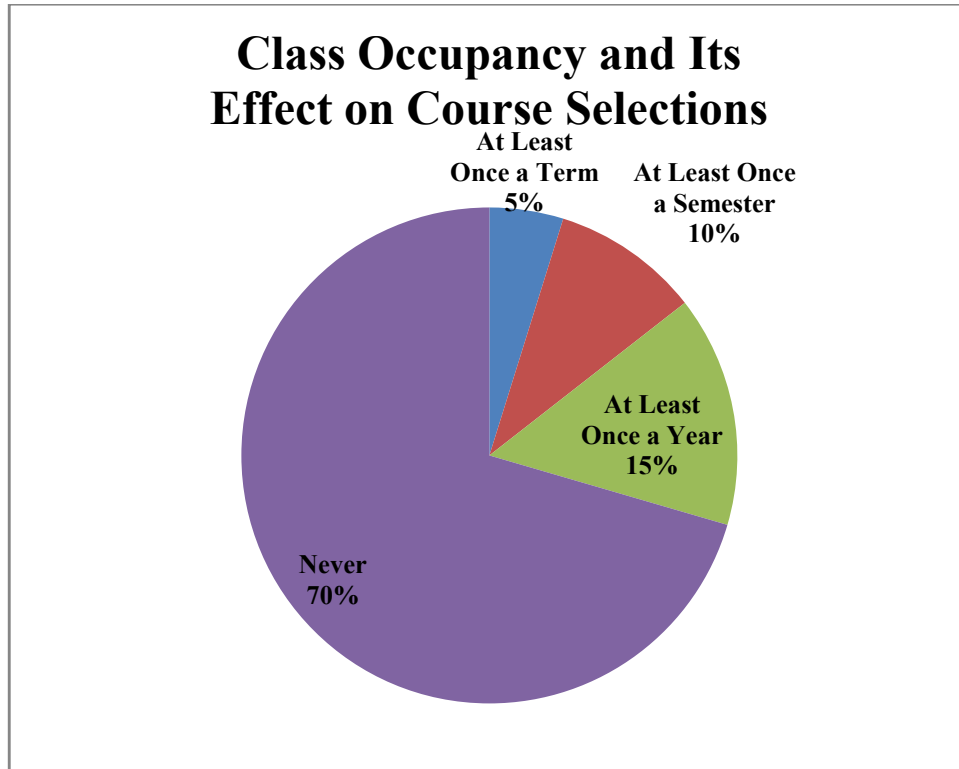


Figure 10: Class Occupancy

### 3.3 Survey Feedback

Of the three hundred and thirty-two students who completed the electronic survey, approximately eight percent are graduate students while the remaining three hundred six students are undergraduates. The undergraduate students can be broken into each class year, including freshmen, sophomores, juniors, seniors, and super seniors. (Super seniors are students who have been attending college for more than four years.) Seventy-two freshmen, eighty-six sophomores, eight-five juniors, fifty-six seniors, and seven super seniors completed the survey.

After answering the ten multiple choice questions, which are available in Appendix A, students were able to provide feedback regarding what they do and do not like in course scheduling systems that they have used in the past. Students were also able to provide suggestions regarding what features they would like to see in the WPI Scheduler.

### **3.3.1 Experiences with Previous Course Scheduling Systems**

Fifteen students provided information regarding course scheduling systems that they used at other universities. The majority of systems were very similar to the current scheduling system at WPI in that students needed to manually create their course schedules and then register for courses via Banner Web. Two students described systems that would not allow students to select courses that conflict with previously selected courses. This feature has been implemented within the WPI Scheduler so that students will not have to manually avoid these conflicts.

### **3.3.2 Suggested Features for the WPI Scheduler**

Over one hundred and twenty students provided suggestions for features they would like in a course scheduling program. Many of the responses were duplicates of each other, so the list was compiled into thirty specific features. Many of the suggested features have been implemented in the WPI Scheduler, and other features may be added in the future.

Of the features that have been implemented in the WPI Scheduler, students stressed that the system must be easy to use, time efficient, and have a graphical interface to view course schedules. Students also suggested that the system should allow the user to select courses and then automatically generate possible schedules. The WPI Scheduler is based on this premise, so schedules are automatically generated.

Many students want a graphical, calendar view of class times and another view to view their schedule for the entire academic year. Other students would like to block specific times that they do not wish to have class, and others want an easier way to view time conflicts between courses. The WPI Scheduler does allow students to block specific periods of time and time conflicts are displayed when initially selecting courses. Students also asked to have the course registration numbers (CRN) displayed; these numbers are currently displayed on the final screen once a final schedule has been selected in the WPI Scheduler.

Although many of the suggested features have been implemented in the WPI Scheduler, several other features have been suggested and may be implemented in the future. These features include color coding courses based on the department and to specifying which courses are Category II and only offered every other year. Suggested calendar features include the

ability to export a calendar to Microsoft Exchange or Google and tabbed browsing between calendars.

The majority of the requested but unimplemented features relate courses to degree requirements. For example, several students would like the program to have a current degree evaluation and indicate which requirements still need to be met and which courses will fulfill those requirements. With regards to courses, students would like to view the course description and the recommended background courses from within the program. Finally, students would like to register for courses through the WPI Scheduler. All of these requests cannot be fulfilled because the data required to implement them is either not available in Banner or they would require that information be written to Banner which is not allowed.

### **3.4 Meetings with WPI Administrators**

To gauge the impact of a new scheduling system for the WPI faculty and administration, we met with three key administrators: Charles Kornik, Dale Snyder, and Paula Delaney. Charles Kornik is the Administrator of Academic Programs and is responsible for scheduling course meeting times, managing the classrooms and the occupancy schedules, and overseeing the Interactive Qualifying Projects (IQP) and the Major Qualifying Projects (MQP). When we met with Mr. Kornik, we described how the original University Scheduler program worked and how we would like to modify the University Scheduler to meet the needs of the WPI community. Mr. Kornik was very excited about the expansion of the University Scheduler, and he agreed to recommend this tool to students. He also agreed, on behalf of the Office of the Registrar, to endorse this tool and to post a link to download the software on the Registrar's website.

We also met with Dale Snyder, the Director of Academic Advising. She informed us that a system is currently being developed for advisors and advisees to communicate via Banner Web. She, like Mr. Kornik, believes that the WPI Scheduler will be beneficial for students, as well as faculty advisors. Ms. Snyder recommended that we include a feature that allows students to save their schedules within the WPI Scheduler so that it can be sent via email or uploaded to Banner Web. This would allow for greater communication between advisees and their advisors. She also agreed to endorse this program and provide links to the software on the Office of Academic Advising's webpage.

Prior to beginning this project, we met with Paula Delaney who works with the Computing and Communications Center as a systems administrator in the Banner division. She has been excited about this project since our initial proposal and believes that the WPI Scheduler will help to ease the load on the servers during peak registration times. Ms. Delaney agreed to provide our team with the necessary course schedule information that is commonly found on the Registrar's website. She has also agreed to provide an XML dump of the scheduling data required to make this system work. Finally, the team has also been working closely with Ms. Delaney to implement features that would be beneficial to faculty advisors.

Through student research and interviews with key administrators, the need for an improved course scheduling system is evident. The WPI Scheduler will not only provide an easier way for students to create their course schedules, but it will also aid communication between advisees and their advisors.

## **4. Design**

The WPI Scheduler is an ambitious expansion of the University Scheduler project, originally developed by Keith Lea. Aside from the core functionality of the University Scheduler, the WPI Scheduler Team rewrote the majority of the data structures and the graphical user interface. This monstrous undertaking was accomplished through the use of good software design as explained in the Methodology section of this report. The use of good software design principles ensured that each part of the program does one thing well and all these parts can then work together to create an excellent program.

The overall purpose and design concept of the WPI Scheduler is to facilitate the easy creation of course schedules by both students and faculty advisors at WPI. The WPI Scheduler reduces the load on the Banner Information System, and it improves student life by eliminating the hassle and the frustration of manually developing course schedules.

The WPI Scheduler transforms the scheduling process that, in some cases, took over twenty hours, of into a process that only takes about five to twenty minutes depending on how many course options are available. The WPI Scheduler is a Java Web Start application that uses one configuration file and one database file, to display everything below and much more.

### **4.1 Scheduler Window and Default Sub Panel**

The Scheduler Window is the primary frame of the Scheduler application. Its purpose is to give the entire application a container as well as provide a file menu. Much of the Scheduler Windows is to get the configuration files and keep track of the child panels via the default sub panel.

Default Sub Panel is never directly seen, but instead contains information and functionality used by nearly every other panel. The Default Sub Panel contains all the information that allows the following panels to operate properly within the WPI Scheduler program. The Default Sub Panel demonstrates the object oriented nature of the WPI Scheduler and University Scheduler projects. The fact that this one class contains functionality that can be used in many contexts is proof that the rest of the classes are well designed.

## **4.2 Course Selection Panel**

The Course Selection Panel allows end-users to select courses that they either need or would like to take in an academic year. In the WPI Scheduler, the Course Selection Panel gains additional functionality with the support for multiple terms and continues to maintain the ability to detect conflicts between course offerings. Users can also specify which term they would like to take a specific course. This allows the user to optimize their schedule so that they can complete the recommended background for a course and it also allows the user to create a more balance schedule.

### **4.2.1 Department Box**

The Department Box limits the number of courses that appear in the Possible Courses List. This panel lists each of the academic departments of the university. When the user selects a department, the courses offered within that department are displayed. This allows for faster end-user searching and an overall improved user experience.

### **4.2.2 Search Box**

Similar to the Department Box, the search box improves the sorting and overall ease of use of the program. Possible search key words include “Professor”, “Location”, “Section CRN”, and “Course Number”. An end user of the program using this tool can save time from scrolling through the two thousand or so courses offered at WPI.

### **4.2.3 Possible Courses List**

The Possible Courses List is one of the two main components of the Course Selection Panel. The Possible Courses List uses a conflict detector and a scheduling engine to show which courses will no longer fit into a user’s schedule based on the courses that have already been selected.

The multi-term interface provides additional functionality to the Possible Courses Panel. It is no longer as important to know if a course cannot be taken during the academic year, but it is now important to see if the course can fit in the user’s schedule with respect to individual part of terms.

Each part of term being scheduled (as configured in the configuration file) is listed before the course number for each course. Each part of term is colored and displays which terms have been blocked due to course conflicts. These conflicts represented by a color code: black equates a valid course and term, light gray means the course is not offered that term, and red means that a previously selected course conflicts with the time the course is offered in the specified term.

#### **4.2.4 Course Information Pane**

The Course Information Pane displays specific details related to the course. These details include, but are not limited to, the number of sections, the professors teaching the course, and the class meeting times and locations. This pane provides the user with pertinent information regarding the course so that the user can determine which term they would like to take the course. This pane can easily be extended to show additional information like the number of seats available.

Directly beneath the Course Information Pane is a sub-pane the WPI Scheduler Team refers to as the Button Panel. The three buttons in this pane are add, remove, and times. The add and remove buttons are used to add and remove selected courses, while the times button brings up a pop up window and displays the available sections and the pertinent information, such as the location, meeting times, et cetera.

#### **4.2.5 Selected Courses Table**

The Selected Course Table lists all the courses the user plans to take in the upcoming time period albeit semester, part of term, or even academic year. The Selected Courses Table allows the user the ability to quickly narrow courses on a term or part of term basis by using a button panel. The button panel uses green to distinguish a selected course, while red means that the course cannot be taken at that time, and white means that the course was not chosen.

An additional feature of this table is the extra check box, which allows the user to set a flag to make a course optional in the scheduling process. Choosing to have a selected course as being optional allows more schedules to be generated.

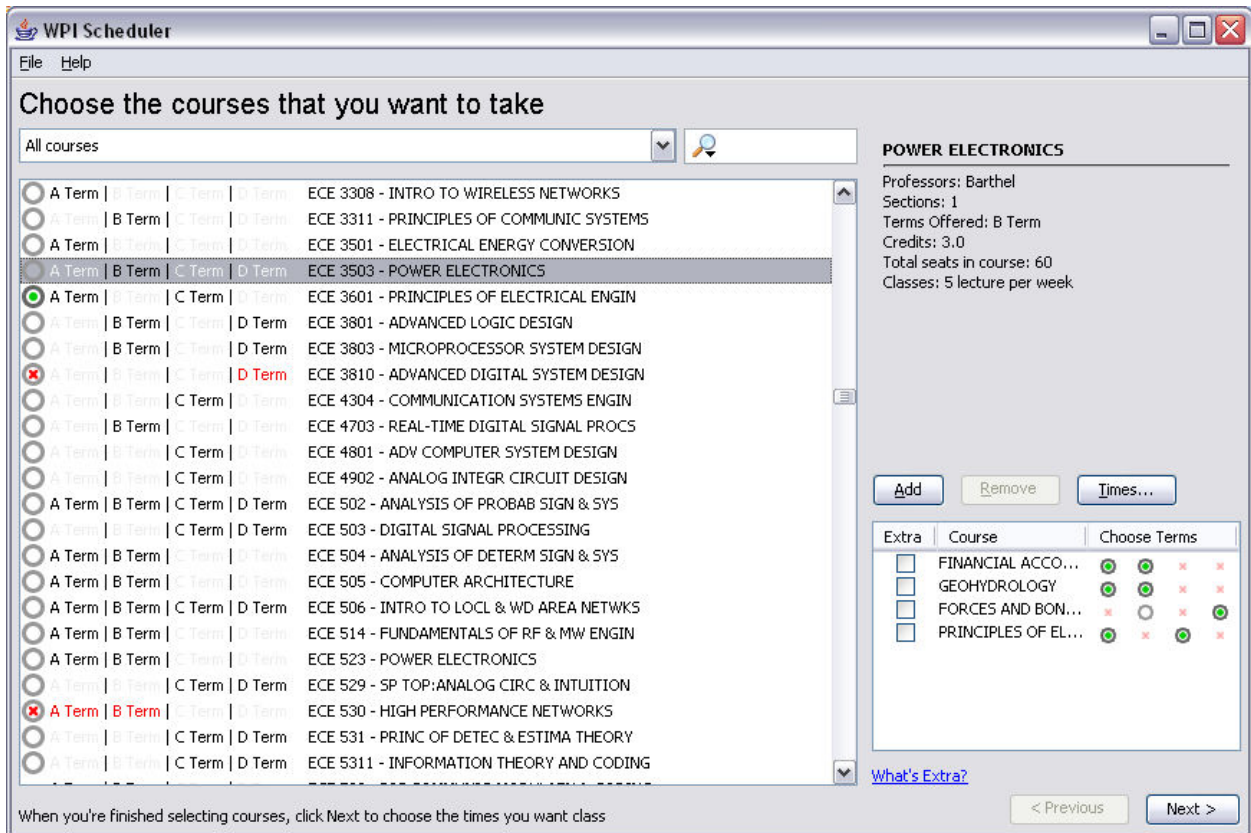


Figure 11 - Course Selection Panel

### 4.3 Narrow Down Panel

The Narrow Down Panel consists of multiple week blocks, one for each part of term. The user is allowed to select times in these blocks when they do not want class. For example, if the user is uncomfortable with an early RF & MW so they would block out the early morning hours eliminating the schedules that has sections in the early morning, while still leaving the rest of the courses. The main body of the Narrow Down Panel is a Year Time Selector which holds the smaller week based Time Selectors.

The other main component of the Narrow Down Panel is the Section Tree which displays all the sections of all the courses selected in the previous panel (Course Selection Panel). There are a couple validation functions built into the background of this panel, which accounts for blocking of too much time as well as indicating which sections were eliminated.



### **4.3.1 Year Time Selector**

The Year Time Selector is an object that the WPI Scheduler Team created in order to manage the Time Selector for each part of term, however it also has the functionality of keeping track of many variables and action listeners that allow the Time Selectors to work properly. The Year Time Selector also renders the panel in the grid pattern and can be expanded to any amount of rows and columns that could feasibly work in scheduler without running out of Java Heap Space.

### **4.3.2 Time Selector**

The Time Selector is a lower level component that displays a weekly view of each part of term. The Time Selector allows the end user to paint all the times they do not wish to have class. The Time Selector can also communicate with the parent object and the children objects that allow it to create additional time masks. In turn, the Time Selector effectively narrows down the possible courses.

### **4.3.3 Section Tree**

The Section Tree lists each course and all of its offered sections. The Section Tree's main feature involves the ability to strikethrough courses that cannot be chosen because it conflicts with another course's meeting time or because of times that have been blocked by the Time Selector.

There are also check boxes beside each of the sections. If one of the boxes is unchecked, the specified section will not be computed in the list of possible schedules. These checkboxes are controlled on an individual basis as well as in the Course Selection Panel, when one would like to not take a course in a given term.

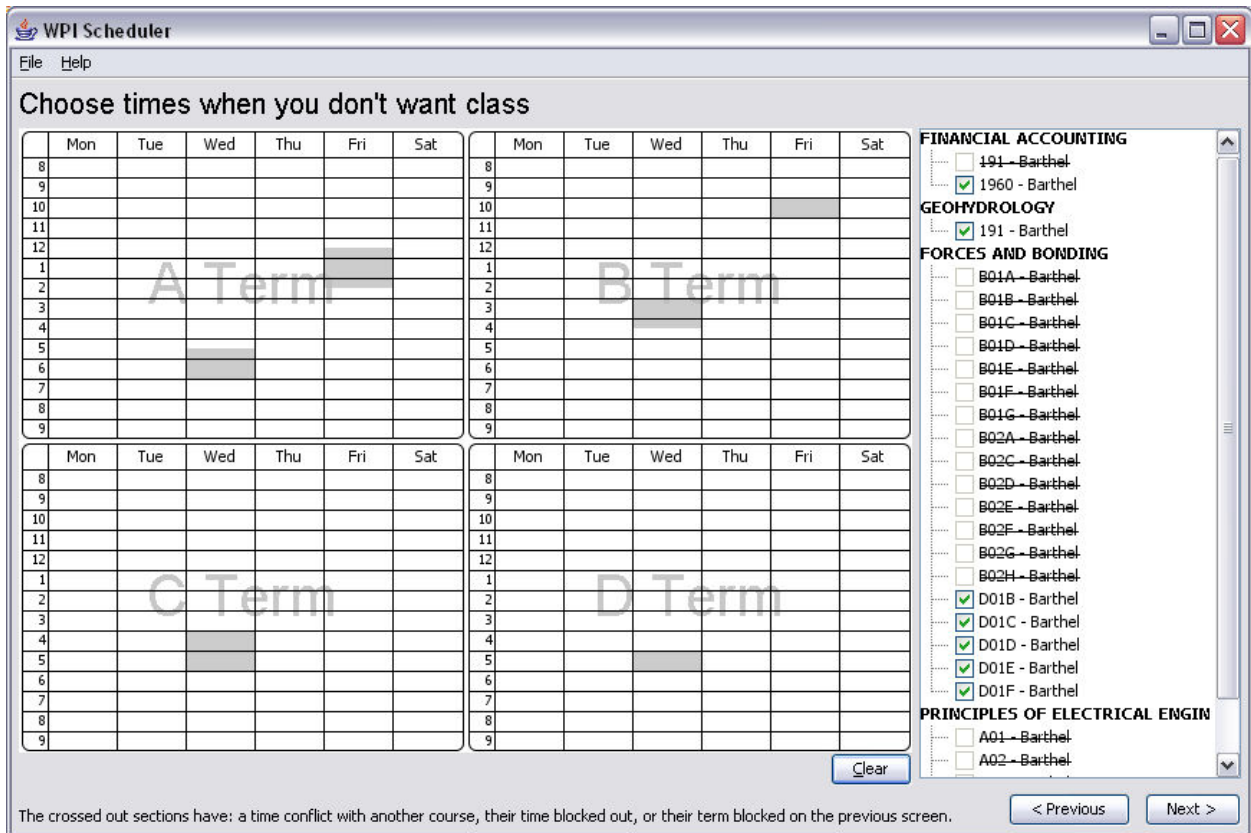


Figure 12 - Narrow Down Panel

## 4.4 View Schedules Panel

The View Schedules Panel displays all the possible schedules. The view is broken into three columns, the Mini Schedules Panel, the Schedule Viewer, and the Legend and Section Details Pane. As a whole, the View Schedules panel provides the user with a wealth of information regarding the available schedules and the differences between each of them.

### 4.4.1 Mini Schedules Pane

The Mini Schedules Pane allows the end user to sort the possible schedules in three formats: by the days the class meets, the shortest breaks between classes, and the times a class is offered. If the user would like additional time to practice for a sport or even has other commitments, the sorting function of this pane allows the user to sort through the possible schedules more efficiently.

#### **4.4.2 Schedule View**

The Schedule View displays a single schedule in a grid of weeks. This allows the end user to graphically view their possible schedules and allows them to comprehend further the blocking of schedules. Many students at Worcester Polytechnic Institute use a similar method to determine if the possibilities fit into the times of the student. This tool automates this process and saves the user countless hours of valuable time. The Schedule View uses a Year Detailed Schedule Grid to show the grid of Detailed Schedule Grids.

#### **4.4.3 Year Detailed Schedule Grid**

The Year Detailed Schedule Grid is the multi-term expansion that makes the WPI Scheduler unique from the University Scheduler before this version. From this expansion, it allows the end user to view as little as a single term or as much as a full year. Through this major development, the Year Detailed Schedule Grid was born. It maintains information about each of the children Detailed Schedule Grids. The most pentacle part of this object is that it is dynamically scalable. This allows the administrators of each university to configure the application to meet the needs of it the institution.

#### **4.4.4 Detailed Schedule Grid**

The Detailed Schedule Grid is an individual unit of the Year Detailed Schedule Grid. The Detailed Schedule Grid contains most of the same functionality of the Panel. In particular, the Detailed Schedule Grid includes the painters and the mouse listeners. These integral components of the graphical user interface provide the user with a more intuitive experience while using the application. The Detailed Schedule Grid also specializes in quickly comparing the amount of time one schedule absorbs verses another.

#### **4.4.5 Legend Pane**

The Legend Pane displays a color-coded list of all the required courses and any possible selected extra courses. This allows the end user to quickly determine the course's location and number of meeting times per week. The Legend Panel directly connects to the Detailed Schedule Grid and the Section Details Pane, with a combination of mouse listeners and the overall data structure of the WPI Scheduler.

#### 4.4.6 Section Details Pane

The Section Details Pane displays pertinent information related to a selected section of a course. This information includes the meeting location, the professor, the number of available seat, and more.

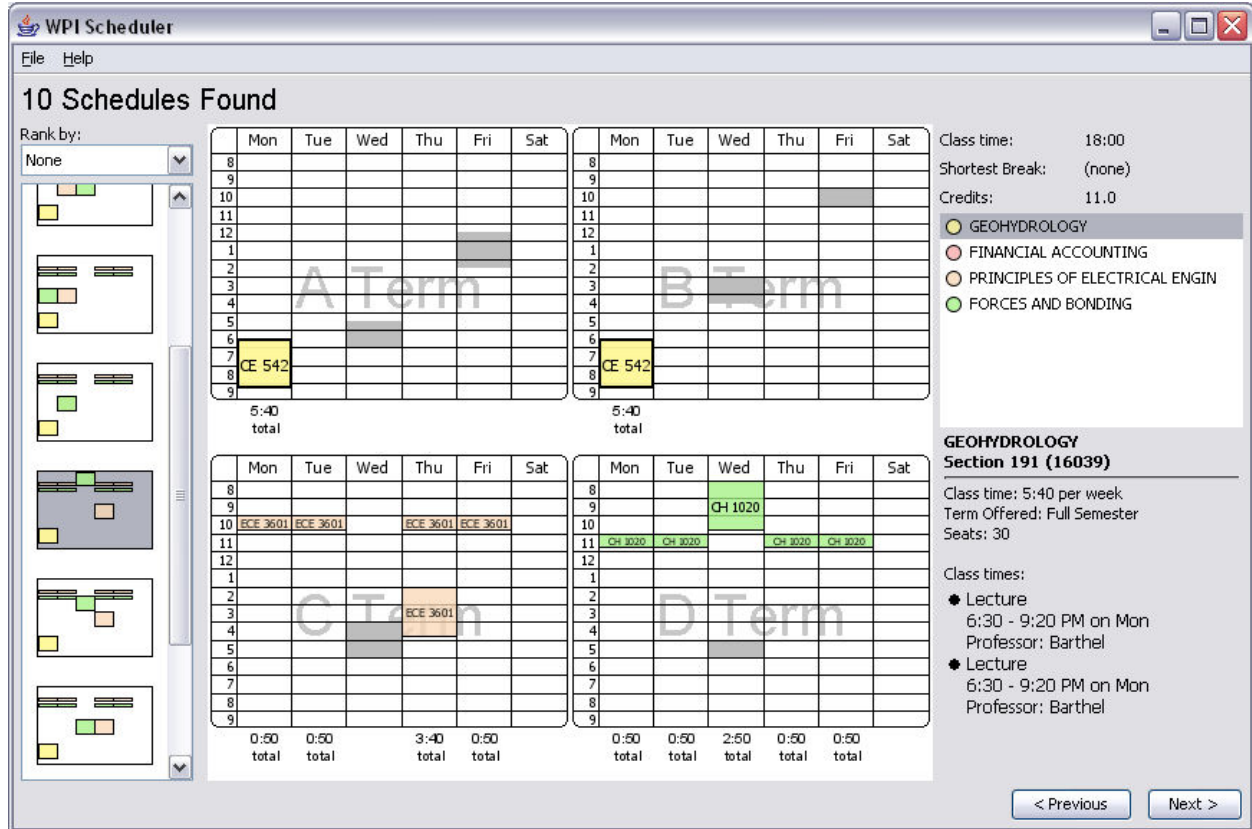


Figure 13 - View Schedule Panel

#### 4.5 Listing Panel

The WPI Scheduler Team added the Listing Panel to the application. The Listing panel displays the course registration numbers for a schedule that the user chooses in the View Schedules Panel. The panel's simple graphical interface provides the user with the pertinent information needed for registration using the Banner Web Student Information System.

Similar to other panels that display multiple terms, the Listing Panel can be modified via a configuration file. This file allows the institution to configure the Listing Panel instructions to specify the proper procedure to follow once a schedule has been chosen.

### 4.5.1 Term Information Panel

The Term Information Panel lists all the course, sections, and course registration numbers of the final schedule that the user selected. In future releases this panel may show the likelihood of being waitlisted for a course.

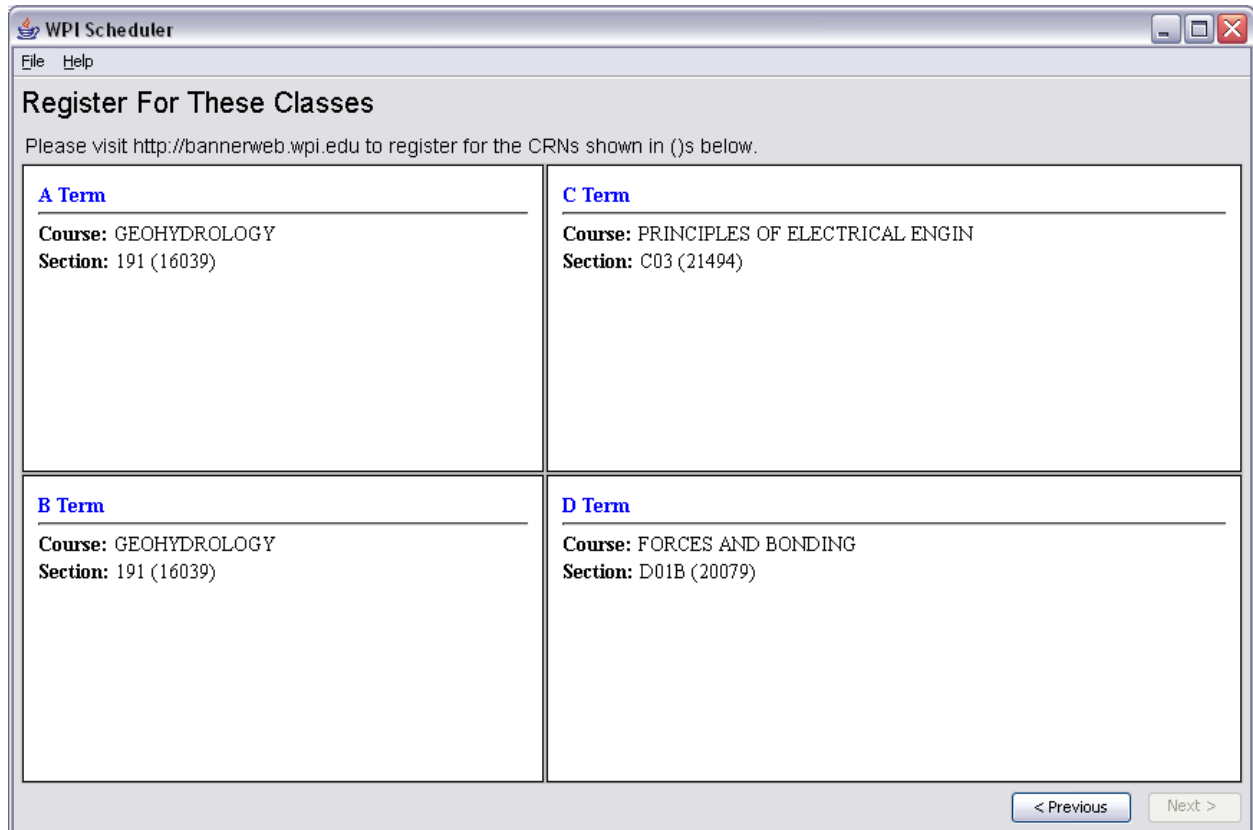


Figure 14 - Listing Panel

## **5. Implementation**

In order to run the WPI Scheduler the end user visits the website and downloads the application, this webstart program maintains a majority of the data within the jnlp file. This file specifically contains information about which JARs to use, their location, as well as all the flags and configuration files. Once the end user installs the application it automatically checks for updates to the class database file and the GUI configuration file, while actual updates to the program are not preformed at runtime it is rather easy to update the program by running the installation jnlp again by going to the scheduler website. A local copy of the configuration file, as well as the class database file is stored in the user's folder under the .Scheduler directory.

### **5.1 How to Build**

The process of building the WPI Scheduler is rather long. When using JetBrains IntelliJ, the developer can remove a sizable portion of time from this process because the project uses IntelliJ workspaces file (IWS), and IntelliJ project files (IPR) that contain all the build path information. The forms\_rt.jar file contains the runtime instructions of IntelliJ's graphical user interface.

To compile the WPI Scheduler, the following three libraries must be included in the build path: JDOM, which reads the XML configuration files and database files, JUnit which performs case tests on the codebase, and Forms\_rt, which is JetBrains IntelliJ ".form" runtime code. All library dependencies should now be satisfied.

To use the existing Java Web Start application, the following packages must be compiled separately: SchedulerEngine, Common, UI, and SchedulerDB. The other two packages are only intended for use at Rensselaer Polytechnic Institute. Once each package has been compiled, the JAR files and the proper Java web start file can be exported to the proper file folder on the designated web server.

### **5.2 How to Run**

There are two methods of running the WPI Scheduler, one of which is in an integrated development environment (IDE) and Java Web Start. When the program is in the development

stage, it is more common to launch the application directly from the IDE. The reason for running the application from the integrated development environment is that the application would need to use a web server with the web start environment.

The WPI Scheduler does not directly use any program arguments, however the Java Virtual Machine receives four arguments: the configuration file for the user interface, the scheduler database with course information, a flag to enable anti-aliasing, and a switch to increase the amount of Java Heap Space allocated to the application. The WPI Scheduler's current configuration file and JNLP file are in Appendix B, and the scheduler's schedule db file is in Appendix C.

## **6. Future Work**

Throughout the entire IQP the WPI Scheduler project team members determined a great many things that could be done to extend the functionality of the WPI Scheduler. While many of these are rather ambitious we broke them down into a few categories: Administrative, Informative, and Functionality.

Expansion of the application functionality will likely include adding iCal abilities. By “telling” the application the dates the classes or parts of term run it is rather easy to add the ability to export the existing data to format that would be readable by a wealth of applications from Outlook, to Google Calendar, and iCalendar. By adding this and other minor but useful functions it could allow the Scheduler to become an even more useful tool.

Allowing WPI students to know what course they need to take in addition to the scheduling would allow incoming students to take courses in the correct order as not to potentially complicate their course load before the end user is able to fully comprehend what the next course has to offer them. With this said the team has an idea to add a database of the recommended background in the scheduler as to attempt to aid the end user further.

Per meetings with the advising department at WPI we saw the ability to dump schedule information into the BannerWeb space useful such that your advisor could potentially see the schedule that you constructed for yourself and even editing and relaying the information back to the student.

There are many other possibilities for the project and lots of things that could be redone in order to improve efficiency or usability. It is the goal of the WPI Scheduler team to improve the overall ease of schedule creation at WPI, and potentially anywhere else this branch of the University Scheduler may be used.



## Appendix A: Survey Questions

1. How would you rate the ease of use of the current WPI course scheduling system? (The current system of looking up course times from the Registrar's website and manually avoiding time conflicts.)
  - a. Very Difficult
  - b. Difficult
  - c. Neither Easy or Difficult
  - d. Easy
  - e. Very Easy
2. Do you create your class schedule based more on class time (what time of day does the class meet or how many times does the class meet per week) or class availability (how many times a class is offered in a year)?
  - a. Mostly Based on Class Time
  - b. Somewhat Based on Class Time
  - c. Equally Based on Class Time and Class Availability
  - d. Somewhat Based on Class Availability
  - e. Mostly Based on Class Availability
  - f. Neither Based on Class Time or Class Availability
3. Do you usually plan an entire year of classes (all for terms) at a time or do you plan each individual term?
  - a. Very Important to Plan Classes for the Entire Academic Year
  - b. Important to Plan Classes for the Entire Academic Year
  - c. Equal Importance to Plan Classes for the Year and for Each Term
  - d. Important to Plan Classes for Each Individual Term
  - e. Very Important to Plan Classes for Each Individual Term
4. Approximately, how much time did you spend creating your class schedule last year?
  - a. Less than an hour
  - b. One to three hours
  - c. Three to six hours
  - d. Six to eight hours

- e. More than eight hours
5. Did you make any changes to your schedule after initially registering for classes? If yes, approximately, how many times did you change your schedule?
- a. Yes
    - i. At least once
    - ii. Twice
    - iii. Three times
    - iv. Four times
    - v. More than four times
  - b. No
6. Do you have Java installed on your home or personal computer?
- a. Yes
  - b. No
  - c. I believe so.
  - d. What's Java?
7. Did you consult with your advisor regarding your class schedule options for this academic year?
- a. Yes
  - b. No
8. Would you like to have a calendar view of a specific course's available meeting times for a week, including lectures, conferences, and labs? For example, there are many different sections available for a physics course and each section includes lecture, conferences, and labs. Would you like to see all of the available sections and their meeting times?
- a. Yes
  - b. No
9. In a calendar view of possible class schedules, which of the following would you like to see?
- a. All four terms at once (Entire Academic Year)
  - b. Two terms at a time (Entire Semester at Once)
  - c. One term at a time (A Term, B Term, C Term, and D Term)

10. How often does a class' size (the number of students enrolled in the course) determine your class choice?

- a. Never
- b. At least once a year
- c. At least once a semester
- d. At least once a term

## Appendix B: Configuration File and Java Web Start JNLP File

### 1. wpi.schedui File:

```
1. <?xml version="1.0"?>
2. <!DOCTYPE schedui PUBLIC "-//Peter Lucia//WPI Scheduler 2007//EN"
3. "http://kano.net/dtd/schedui-2005.dtd">
4. <schedui>
5.   <program-info>
6.     <!-- This is the name that will display at the top of your program -->
7.     <program-name>WPI Scheduler</program-name>
8.     <!-- This is the contact information that will be shown in error messages
9.     and
10.    in the help files. -->
11.     <admin name="Peter Lucia" email="peternl@wpi.edu" />
12.     <!-- This is the text that shows up in the About this Program box.
13.     Basic HTML formatting can be used. -->
14.     <about-box-text><![CDATA[
15.       <b>In order of development</b>
16.       <hr>
17.       The University Scheduler (formerly the RPI Scheduler) is developed by
18.       Keith Lea, a senior at RPI, with help
19.       from Anish Shah, an RPI alumnus, and Vinay Shah, a senior at RPI.
20.       <br><br>
21.       Thanks to Charlie Hayes, Sam Stephenson, Tom Manhart, Zif Horton, Aaron
22.       friend, for testing and ideas for the Scheduler.
23.       <br><br>
24.       Thanks to Charlie Hayes for icon design.
25.       <br>
26.       Thanks to Sam Stephenson for the RPI Scheduler website design.
27.       <hr>
28.       The University Scheduler was extended to be term based
29.       by an IQP Team at WPI during the Fall of 2007. This team consisted of:
30.       <br><br>
31.       Professor Mike Ciaraldi - CS - Advisor<br>
32.       Peter Lucia - CS/ECE - Class of 2009<br>
33.       Cyle Hawkins - Aerospace - Class of 2010<br>
34.       Gracila Rubio - ECE - Class of 2010<br>
35.       <br>
36.       Thanks to the whole ADP lab crew for testing and support!
37.     ]]>
38.   </about-box-text>
39.
40.   <!-- Each <help-link> shows up as an item in the Help menu which brings
41.   the user to a web page. You can add or remove as many as you want, and
42.   change the ones that are already here. -->
43.   <help-links>
44.     <help-link href="http://rpischeduler.com/help">Scheduler Help
45.     Website</help-link>
46.     <help-link href="http://rpischeduler.com/help">Report a Bug</help-link>
47.   </help-links>
48.   <!-- This is the text that shows up in the Registration Instructions box
49.   Basic HTML formating can be used, although unfortunately links do not
50.   yet work. -->
51.   <registration-instructions><![CDATA[
52.     <span style="font:Arial;">
53.       Please visit http://bannerweb.wpi.edu to register for the CRNs shown in
54.       ()s below.
55.     </span>
```

```

54.     ]]>
55.     </registration-instructions>
56.
57.     </program-info>
58.
59.     <ui-info>
60.     <!--
61.         These <icon> entries specify what window icon should be used for the
62.         four main windows of the application.
63.
64.         These can be PNG, BMP, JPG, or GIF files. We recommend that you use
65.         16x16
66.         images for these icons. The URL's are relative to the "codebase"
67.         defined
68.         in the scheduler.jnlp file. You may also use "absolute" URL's like
69.         "http://myschool.edu/scheduler/image.gif".
70.
71.         If you don't change these, the default Scheduler icon will be used.
72.     --><!--
73.     <icons>
74.     <icon type="main" href="icon-16x16.png" />
75.     <icon type="course-preview" href="icon-16x16.png" />
76.     <icon type="view-schedule" href="icon-16x16.png" />
77.     <icon type="loading-progress" href="icon-16x16.png" />
78.     </icons>-->
79.
80.     <!-- This sets the size of the grid to be displayed in all the
81.     scheduler panels. Currently there is a known bug with rendering
82.     if days is set to anything besides "mon,tue,wed,thu,fri,sat"
83.     assuming that classes are Monday - Friday. There is no known
84.     work around if classes occur on Saturday or you would like to
85.     display Sunday.
86.
87.     Things will break if you set this to range so it does not
88.     include all times that courses are offered.
89.     -->
90.     <default-grid-range days="mon,tue,wed,thu,fri,sat"
91.     start-time="8:00 AM" end-time="10:00 PM" />
92.
93.     <!-- This section sets default window sizes for various windows.
94.     Defaults are usually fine, but feel free to adjust. -->
95.     <window-sizes>
96.     <window-size type="main" width="780" height="590"/>
97.     <window-size type="course-preview" width="713" height="363" />
98.     <window-size type="view-schedule" width="678" height="439" />
99.     </window-sizes>
100.
101.     <!--
102.         The display-format entries specify how the application should
103.         identify
104.         a course to the user, in different contexts. Course names formatted
105.         in
106.         this way are seen throughout the application.
107.
108.         The formats provided here should be good enough, but if your school
109.         has a
110.         particular way of referring to courses, like
111.         "CSCI/1200/Programming Languages", you should change these formats.
112.
113.         The display-format fields {0} through {7} will be replaced
114.         according to
115.         these rules:
116.         {0} = Department abbreviation, like CSCI

```

```

111.          {1} = Department name, like Computer Science
112.          {2} = Course number, like "1200"
113.          {3} = Course name, like "Programming Languages"
114.          {4} = Course minimum credits, like "2" (if course may be worth 2
to 12
115.          credits), or "4" (if course is worth 4 credits)
116.          {5} = Course maximum credits, like "12" (if course may be worth 2
to 12
117.          credits), or "4" (if course is worth 4 credits)
118.          {6} = Course credits, like "2-12" (if course may be worth 2 to 12
119.          credits), or "4" (if course is worth 4 credits)
120.          {7} = Number of sections in course, like "6"
121.          -->
122.          <display-formats>
123.          <!-- The "very-short-course" format is for when there is not
enough room
124.          for the course name to be shown.
125.
126.          The standard very-short-course format, "{0} {2}", shows
courses like
127.          "CSCI 1200". -->
128.          <display-format type="very-short-course">{0} {2}</display-format>
129.
130.          <!-- The "short-course" format is for when there is more room, and
131.          more information about the course can be shown (like its full
132.          name).
133.
134.          The standard short-course format, "{0} {2} - {3}", shows
courses like
135.          "CSCI 1200 - Programming Languages".-->
136.          <display-format type="short-course">{0} {2} - {3}</display-format>
137.          </display-formats>
138.          </ui-info>
139.
140.          <!--
141.          Here you can define the types of course searches that the user can
perform
142.          using the search box in the upper-right corner of the initial course
143.          selection screen.
144.
145.          Each search-type definition contains a list of search-keys. A search
key
146.          specifies which part of a course's information should be searched.
There are
147.          several pre-defined search keys:
148.
149.          dept-name - searches department names, like "Computer Science"
150.          dept-abbrev - searches department abbreviations, like "CSCI"
151.          course-name - searches course names, like "Programming Languages"
152.          course-number - searches course numbers, like "1200"
153.          section-id - searches section ID's, like "64423"
154.          notes - searches note entries for a course, like "This course meets
with
155.          STSS 4963"
156.          location - searches class period location, like "Barton 103"
157.          professor - searches class period professor names, like "Showalter"
158.
159.          As an example, if your school lists the locations of class periods,
you
160.          could add a location search type like this:
161.
162.          <search-type name="Class location">
163.          <search-key name="location" />

```

```

164.         </search-type>
165.
166.     Then, students using your Scheduler could select the Location search
type,
167.     and type the name of a building, to find all courses which meet in
that
168.     building.
169.     -->
170.     <search-types>
171.         <search-type name="All">
172.             <search-key name="course-name" />
173.             <search-key name="course-number" />
174.             <search-key name="professor" />
175.             <search-key name="location" />
176.             <search-key name="section-id" />
177.         </search-type>
178.
179.         <search-type name="Course name">
180.             <search-key name="course-name" />
181.             <search-key name="course-number" />
182.         </search-type>
183.
184.         <search-type name="Professor">
185.             <search-key name="professor" />
186.         </search-type>
187.
188.         <search-type name="Class location">
189.             <search-key name="location" />
190.         </search-type>
191.
192.         <search-type name="Section CRN">
193.             <search-key name="section-id" />
194.         </search-type>
195.     </search-types>
196.
197.     <!--
198.         NEW!
199.         This section defines how your school's year is structured. Each
200.         Year is made up Terms, and each Term is made up of Part-Of-Terms.
201.         If your school uses a semester based system you probably only
202.         want two Terms with one Part of Term each. Each Term needs at
203.         least one Part of Term.
204.
205.         At many schools it may be appropriate to think of what the
206.         University Scheduler calls "Terms" as what you know as
207.         Semesters and "Part Of Terms" as Terms.
208.
209.         A Semester Based Example:
210.         <year>
211.             <term name="Fall Semester">
212.                 <part-of-term name="Fall" />
213.             </term>
214.             <term name="Spring Semester">
215.                 <part-of-term name="Spring" />
216.             </term>
217.         </year>
218.
219.         A Term Based (WPI) Example:
220.         <year>
221.             <term name="Fall">
222.                 <part-of-term name="A Term" />
223.                 <part-of-term name="B Term" />
224.             </term>

```

```

225.             <term name="Spring">
226.                 <part-of-term name="C Term" />
227.                 <part-of-term name="D Term" />
228.             </term>
229.         </year>
230.
231.     -->
232.     <year>
233.         <term name="200801">
234.             <part-of-term name="A Term" />
235.             <part-of-term name="B Term" />
236.         </term>
237.         <term name="200802">
238.             <part-of-term name="C Term" />
239.             <part-of-term name="D Term" />
240.         </term>
241.     </year>
242.
243. </schedui>
244.

```

## 2. scheduler.jnlp:

```

1. <?xml version="1.0" encoding="utf-8"?>
2. <!--
3.     This is the Java Web Start launch file. You should read the instructions
4.     inside the <!-- -> marks, like the ones that enclose this paragraph.
5.
6.     You should only modify the parts of this file which have associated
7.     instructions in <!-- -> marks. The other parts of the file do not need to be
8.     modified.
9. -->
10.
11. <!DOCTYPE jnlp PUBLIC "-//Sun Microsystems, Inc//DTD JNLP Descriptor 1.1//EN"
12.     "http://kano.net/dtd/jnlp-1.5.dtd">
13.
14. <!--
15.     You should change the "codebase" value below, to the URL for where your JNLP
16.     and JAR files are located.
17.
18.     This does not have to be the same URL as your main Scheduler site; we suggest
19.     placing the JNLP, JAR, and associated files in a subfolder called "run".
20. -->
21. <jnlp spec="1.5"
22.     codebase="http://localhost/sched/sched/run/"
23.     href="scheduler.jnlp">
24.     <information>
25.
26.         <!-- You should change the title and vendor to whatever your school's
27.             scheduler is called. -->
28.         <title>WPI Scheduler</title>
29.         <vendor>WPI Scheduler 2007</vendor>
30.
31.         <!-- You should change the "href" value to your main scheduler website, --
32.         >
33.         <homepage href="http://localhost/sched/sched/" />
34.
35.         <!-- You can change these descriptions to match your school's scheduler.
36.             They will be shown at different points in the installation process,
37.             and
38.             in the user's Start menu and desktop. -->

```



```

37.   <description>Makes scheduling classes at My School fast and
      easy.</description>
38.   <description kind="short">My School class scheduler</description>
39.   <description kind="tooltip">My School Scheduler</description>
40.
41.   <!-- You should change the "href" value to the name of your school's
42.         scheduler icon, in the folder you specified earlier in this file as
43.         your "codebase". This should be a GIF file.
44.
45.         If you don't change this, the default Scheduler icon will be used. -->
46.   <icon href="jnlp-icon.gif" width="32" height="32" />
47.
48.   <offline-allowed/>
49.
50.   <association extensions="schedule" mime-type="text/x-course-schedule" />
51.
52.   <shortcut online="false">
53.     <desktop/>
54.     <menu/>
55.   </shortcut>
56. </information>
57.
58. <security>
59.   <all-permissions />
60. </security>
61.
62. <resources>
63.
64.   <!-- You should change this "value" URL to point to the SCHEDB course
65.         database file for your school's scheduler.
66.
67.         The URL's are relative to the "codebase" defined earlier in this file.
68.         You may also use "absolute" URL's like
69.         "http://myschool.edu/scheduler/undergrad.schedb". -->
70.   <property name="scheduler.dburl" value="wpi.schedb" />
71.
72.   <!-- You should change the "value" URL here to point to the SCHEDUI
73.         scheduler UI configuration file for your school's scheduler.
74.
75.         The URL's are relative to the "codebase" defined earlier in this file.
76.         You may also use "absolute" URL's like
77.         "http://myschool.edu/scheduler/something.schedui". -->
78.   <property name="scheduler.ui.configfile" value="wpi.schedui" />
79.
80.   <!-- These four properties control how long the Scheduler will wait when
81.         downloading new SCHEDB (course database) and SCHEDUI (Scheduler
82.         configuration) files when the user starts up the program.
83.
84.         The first property ("mintimeoutdays") is measured in days; the last 3
85.         are measured in seconds.
86.
87.         For details on what these properties mean, see the "Scheduler at Your
88.         School" tutorial. -->
89.   <property name="scheduler.cache.mintimeoutdays" value="3" />
90.   <property name="scheduler.cache.extraTimeoutDelta" value="1.0" />
91.   <property name="scheduler.cache.mintimeout" value="3.0" />
92.   <property name="scheduler.cache.maxtimeout" value="10.0" />
93.
94.   <!-- This option toggles if we want to generate our schedule's time
      masks on the fly
95.         (CPU intensive) or if you want to preload them all (Memory
      intensive). This is

```

```
96.         only really an issue when you have a large number of potential
          schedules (>4,000).
97.
98.         Set to True to use the CPU intensive option, false to preload into
          memory
99.         -->
100.        <property name="saveMemory" value="true" />
101.
102.        <j2se version="1.5+" />
103.
104.        <jar href="ui.jar"/>
105.        <jar href="engine.jar"/>
106.        <jar href="common.jar" />
107.        <jar href="db.jar" />
108.        <jar href="forms_rt.jar" />
109.        <jar href="jdom.jar" />
110.        </resources>
111.
112.        <resources os="Win">
113.        <jar href="winlaf.jar" />
114.        <nativelib href="winlaf-native.jar"/>
115.        </resources>
116.
117.        <application-desc main-class="edu.rpi.scheduler.ui.Scheduler" />
118.    </jnlp>
```

## Appendix C: Sample SchedeDB File

### 1. Explanation File

```
1. <?xml version="1.0" encoding="UTF-8"?>
2. <!--
3.   This is a sample SCHEDB XML course database file. This sample file should
   help
4.   you to understand the Scheduler database format, and help you to make your
5.   own SCHEDB database file generator.
6.
7.   This has been updated for the WPI Scheduler release of the University
   Scheduler.
8.
9.   This file contains all schedule data for a the current time. It is organized
   as follows:
10.  -Department - Math, Physics, etc
11.  --Course    - MA101 Calculus 1
12.  ---Section  - A particular offering of this course (A01, etc)
13.  ----Period - A particular time that this offering of the course occurs.
14.
15.
16. -->
17. <schedb generated="Thu Nov 04 00:24:48 EST 2004" minutes-per-block="30">
18.
19.
20. <!-- A department takes a short abbreviation (abbrev) and a full name (name).
21.   The abbreviation needs to be unique to this department, but names can be
   repeated.
22.   This department abbreviation will be prepended to every course number
   for courses
23.   under this department.
24. -->
25. <dept abbrev="ADMN" name="Administration">
26.
27.
28. <!-- Next we have an entry for each of the courses that are offered in
   this
29.   department.
30.
31.   Each course has a course number (which will have the department
   abbreviation
32.   prepended to it) and a name. Additionally, a course has a grade type
   which
33.   can be either normal or pass-fail. Finally, each course has min-
   credits and
34.   max-credits attribute.
35.
36.   For most courses, the min-credits and max-credits will be the same,
37.   because most courses are only worth a single number of credits.
38.   However, at many schools, independent study courses are worth some
39.   number in a given range (like 2-12), so this option is provided.
40. -->
41. <course number="1010" name="ORAL COMM FOR TA'S II"
42.   min-credits="4" max-credits="4" grade-type="normal">
43.
44.
45. <!-- Since each course is often offered more than once, we introduce
   sections.
46.
47.   A section may have a CRN (Course Reference Number), and a section
```

48. number. Each section in the entire course database must have a  
unique

49. CRN which is shared by no other section of any course. A section's  
50. section number, however, must only be unique within that course. So,  
51. for example, there cannot be two sections of "ORAL COMM FOR TA'S II"  
52. with the section number "2".  
53.

54. The "seats" value corresponds to how many students can take this  
55. section of this course.  
56.

57. It is at the section level that we introduce when during the year  
the

58. course is offered. See the SchedUI file for a description of how  
59. Terms and PartOfTerms work. It is sufficient to say here that the  
60. Term and PartOfTerm information here needs to be exactly as  
specified

61. in the SchedUI file.  
62.

63. If a section spans multiple part of terms, include them all in order  
64. in a comma separated list. If a section spans an entire Term that  
has

65. multiple part of terms under it, enter "Full Semester" as the  
PartOfTerm.

66. -->

67. <section crn="30392" number="1" seats="15" availableseats="4"  
term="200801"

68. partofterm="A Term, B Term">

69.  
70.

71. <!--

72. Periods represent a single weekly meeting time for a course.  
73. Each section typically has multiple periods, or at least one for  
74. lecture and one for a lab.  
75.

76. The following period types are recognized:  
77. ind-study lab lecture recitation seminar studio  
78.

79. If a period occurs at the same time on multiple days they can all  
80. be entered in a comma separated list.  
81. The following abbreviations must be used for the "days" value:  
82. mon tue wed thu fri sat sun  
83. -->

84. <period type="lecture" professor="Barthel" days="mon"  
85. starts="4:00PM" ends="4:50PM" building="AK" room="233" />

86.  
87. <-- More example data is found below -->

88. <period type="recitation" professor="Barthel" days="tue,thu"  
89. starts="4:00PM" ends="4:50PM" building="AK" room="233" />

90. </section>

91.  
92. <section crn="30393" number="2" seats="30" availableseats="4">

93. <period type="lecture" professor="Barthel" days="mon"  
94. starts="6:00PM" ends="7:50PM" building="SL" room="105" />

95. <period type="recitation" professor="Barthel" days="tue,thu"  
96. starts="10:00AM" ends="12:50PM" building="SL" room="105" />

97. </section>

98. </course>

99.  
100. <course number="4050" name="PROFESSIONAL LEADERSHIP P"  
101. min-credits="0" max-credits="0" grade-type="pass-fail">

102. <section crn="33659" number="1" seats="45" availableseats="4">

103. <period type="lecture" professor="Persoon" days="tue"  
104. starts="7:00PM" ends="8:50PM" building="SL" room="105" />

```

105.         </section>
106.     </course>
107. </dept>
108.
109.     <dept abbrev="ARCH" name="Architecture">
110.         <!-- Don't forget, this is XML, so special characters like the & sign
must
111.             be specially formatted. You should use an XML code library to
generate
112.                 this file; such a library will automate things like this. -->
113.         <course number="2120" name="BLDG & THINKING ARCH 2"
114.             min-credits="2" max-credits="2" grade-type="normal">
115.             <section crn="30133" number="1" seats="90" availableseats="4" >
116.                 <period type="lecture" professor="Bell" days="mon,thu"
117.                     starts="10:00AM" ends="11:50AM" building="SL" room="105" />
118.             </section>
119.         </course>
120.     </dept>
121. </schedb>

```