Project P0004:
Web Based Appointment System

IT 491 / CIS 492 - Senior Project

Fall 2002

Professor: Osama Eljabiri

Submitted by:

- Marius M. Acosta
- Johnny N. Lopez
- Glenys Pina
- Abbas Sheikh
- Toni Sopov
- Xoche B. Umagat
- Prashant Vadhadia

TABLE OF CONTENTS

1. PROJECT INITIATION

1.1 PROJECT ABSTRACT

In order to provide NJIT students a convenient and effective way of scheduling an appointment with their advisors, we will design a Web-based Appointment System. Replacing the current 'paper-based' appointment system with our web-based system will allow NJIT students to make an appointment for advising from the comfort of their home or from any computer with Internet access.

The appointment scheduling options being offer today are time consuming, unreliable, and unorganized. In order to schedule an appointment, students must walk into the office and search the appointment book for an available timeslot; or make a telephone call and wait for the secretary to tell them of the advisor's available timeslots.

The web-based appointment system will offer a user-friendly interface, which will allow students and advisors to manage appointment scheduling more effectively. Students can go online anytime, review the advisor's available days and timeslots, and schedule an appointment. To make the appointment, students must provide their name, ID, and email address. To complete the appointment, students must fill in the 'CCS Course Selection Form' online. Students can only schedule one appointment of 15 minutes timeslot limit. Once a timeslot is reserved for a student, its availability status will change. If students want to schedule another appointment, they must first delete the current one avoiding duplicate appointments. The system will send e-mail reminders a day prior to the students' appointment. On the other hand, advisors can go online to check their schedule at anytime, and make changes to their available timeslots, or cancel appointments. Overall, the system helps provide better service to students and advisors.

1.2 TEAM ROLES AND RESPONSIBILITES

	Name	Team Role	Responsibilities
1	Glenys Pina	Project Manager	Manage Team Members and Project deliverables. Prototype Contacting Sponsors and Getting Feedback Work Breakdown Structure (WBS) Ghantt Chart & Pert Chart Stakeholder Identification User Interface Design Protocol Analysis
2	Marius M. Acosta	1st System Analyst	Gathering system requirements, and project documentation including problem definition, and requirement reports, etc. Research on Business Model Student and Advisor Sequence Diagram & State Chart Diagram
3	Johnny N. Lopez	Front End Designer	Design the interface of the web based appointment system. ER-Diagram and Data Dictionary Database Design
4	Abbas Sheikh	Database Designer	Design the database Responsible for SPSS System for preparation of Questionnaire Database Design
5	Toni Sopov	2nd System Analyst	Work together with Marius in Gathering system requirements, and project documentation including problem definition, and requirement reports, etc. Research on DFDs Modular Decomposition
6	Xoche B. Umagat	Programmer	Implement, test and maintain the system. System Requirments COCOMO and Function Point Requirement Definitions and Modern Methods & Responsible for connecting Database to the System
7	Prashant Vadhadia	Architectural Designer	Design system's architectural model System Specification System Structuring, Structured English, Decision Tree and Decision Tables Research on other student appointment Systems Design System Control Model User Reference Manual

1.3 BACKGROUND

New Jersey Institute of Technology (NJIT) is a public, urban research university, which is committed to the pursuant of a higher level of education. New Jersey Institute of Technology offers undergraduate, graduate, and continuing professional education in such fields as Architecture, Engineering, Science and Liberal Arts, Management, and Computing Sciences.

Since New Jersey Institute of Technology is a technical university, it tries to "advance the uses of technology as a means of improving the quality of life." NJIT holds true to this statement by providing almost everything a student possibly needs on the World Wide Web. An active NJIT student is able to register for classes, view courses for the current semester, register for their courses, pay their term bill, etc.

Although NJIT is technologically wired, when it comes to scheduling an appointment with an advisor, a student is unable to go to the World Wide Web to schedule an appointment. For a student to schedule an appointment with an advisor, the student needs to do the following: 1st) Physically walk to the CIS Department offices, 2nd) Write their name in a given timeslot of a particular day and 3rd) Walk back to the CIS Department on the day of the appointment to speak to the advisor. Hopefully no one has scratched one's name out and put theirs in one's time slot. This procedure is tedious, time consuming, and unreliable. This method of scheduling offers no anonymity for the student who is scheduling the appointment. Students are able to see in the appointment book the name and time other students are meeting their advisors.

For a University that tries to "advance the uses of technology as a means of improving the quality of life", NJIT needs to change the way a student schedules an appointment with their advisor. NJIT faculty thought of a way to improve the process of a student scheduling a meeting with an advisor by offering an idea of a Web-based Appointment System.

It is necessary for New Jersey Institute of Technology to upgrade and implement a new Web-based Appointment System in order to accommodate a student's and an Advisor's needs. Also NJIT will be able to keep the competitive advantage of being the most technically wired university by providing this technology to improve the quality of life.

1.4 PROBLEM STATEMENT

As it might have been described previously, the process of scheduling an appointment to see an advisor at NJIT is a tedious, unreliable and time-consuming process. The problems with the current system include but are not limited to under-utilization of resources, not managing resources efficiently, being unreliable and unorganized.

Students must be present to make an appointment with their advisor, or inquire over the telephone about advisor's time availability. In this case, resources (time and people) are not being managed efficiently. It takes time from the student having to go to the department in order to schedule the appointment and then going back the actual day of the appointment to see the advisor. When inquiring over the telephone about advisor's time availability, the secretary is also involved in the appointment scheduling process. That makes two people involved in the process, taking time from the student to call and then hold for the secretary to get the available time slots, along with stopping the secretary from her/his duties to answer the telephone and looking in the appointment book, and then writing down the student's name. At the beginning of their workday, the secretary must provide the advisors with a hard copy of the particular page in the appointment book that holds their appointments for the day.

This whole process is unreliable and unorganized. Students can cross out other student's names to take their spot. Students can reserve more than one time-slot not giving a fair chance for other students to schedule an appointment. Also appointment records could be misplaced by a human error or lost if paper is ripped out of the binder.

To address the problems with the current paper-based appointment system, we will be working on an idea provided by an NJIT faculty member, Dr. Baltrush. His idea is to create a Web-based Appointment System, which will replace the current paper-based system. Dr.

Baltrush will be sponsoring our project along with the two CCS department advisors Mr. Thomas Moore and Ms. Kathy Sherman. Our goal is to develop a reliable, effective, accessible, and user-friendly web based system to help organize the advisors' agenda and the way students schedule their appointments.

With a Web-based Appointment System a student will be able to go onto a website and login with their name and student ID. Then from there a student will be able to view the advisor's name and available timeslots to schedule the appointment.

1.5 PREVIOUS WORK

A Web-based Appointment System is not a new or unique concept. There are many institutions that have a Web-based Appointment System incorporated into their information system to make life a little better for the institution and its customers. There is also Application Service Providers (ASP) that will design the Web-based scheduling system to work with your institution's infrastructure. We will be analyzing two different available software's that are out on the market today, so that we can possibly adopt and/or streamline an already working Web-based Appointment System so that it will cater to our specified requirements.

TimeTrade Systems, Inc. (Application Services Provider)

The first possible software solution we looked at was TimeTrade Systems, Inc. TimeTrade operates as an Application Services Provider (ASP) and as a traditional software provider for internally hosted solutions. TimeTrade has developed software that can be helpful in our development of the Web-based Appointment System.

With TimeTrade Systems, Inc.'s Web-based scheduling, a corporation will be able to allow everyone involved in the Web-based scheduling system to access the available timeslots in real-time. TimeTrade states that it can create a closed system accessible only by authorized staff members, or open access, via user login and password, for self-service scheduling by students and faculty. TimeTrade also states that if an institution chooses to offer self-service scheduling, which NJIT advisors wish the system to do, a student can go online at any time of day or night, check availability, and schedule a timeslot.

TimeTrade software will evaluate the available advisors and times, and apply appropriate constraints on the system based on the institutions business rules. There will never be any accidental double bookings because it is going to be a real-time scheduling application.

Whenever a timeslot is taken, that timeslot will be immediately removed from the available timeslots. This system will be easy to manage because it is going to be a centralized managed system.

This functionality in the software is the exact idea that we have in mind. The only function this software is missing is the e-mail reminder notification of the scheduled appointment to remind the student and the advisor that they have a meeting. Overall, TimeTrade Systems, Inc.'s software will provide a solid foundation for us to follow in developing NJIT's Web-based Appointment System.

Switch I.T. (Web and Software Development)

The next possible software solution we viewed was a software development concept from Switch I.T. Switch I.T. has the same views and concepts that we have in mind. Switch I.T. designed and implemented a better approach to appointment scheduling and communication between different employees at different physical locations. Switch I.T. developed an online centralized scheduling system to provide staff members' access to appointment and contact information via the Internet.

Switch I.T. is providing us with some good ideas and examples like: *Browsable daily calendars*: The system allows users to browse upcoming appointments on a day-by-day basis. The clean, usable interface allows simple access to any date in the calendar, *Permission based access*: The system will have different access levels depending on your account status (student-advisor), and *Automatic notification of schedule conflicts*: If a user attempts to add an appointment over an existing appointment, the system will notify the user and provide different options to work around the conflict. These examples were exact views of what we created in pencil during our team meeting on how the graphical layout of the system should look like.

The one feature that was not offered by Switch I.T. was a way for the software to send an automatic meeting appointment message reminder, via e-mail, to notify the parties involved that they have a meeting scheduled on a particular time of day.

From studying this software we will be able to gain an even greater understanding about developing and implementing our Web-based Appointment System so that it will meet our stakeholders' requirements.

1.6 METHODOLOGIES

1.6.1 EXTREME PROGRAMMING (XP)

The Agile approach to software development focuses on fast delivery of quality products through which the project life cycle can be reduced. Agile principles include but are not limited to A) Customer satisfaction through early and continuous delivery of product; B) Allow requirement changes at any stages of the development process for customer advantage; C) Provide customer with frequent working-product delivery allowing testing and viewing how the product will perform and looks; D) Developers and customers 'must work daily through the project'; and E) 'Working software is the primary measure of progress' .

Extreme Programming (XP) is an agile methodology which focuses in team work, customer satisfaction and fast delivery of needed product. "XP improves a software project in four essential ways: communication, simplicity, feedback and courage". XP allows changing customer requirements at any stage of the development life cycle. The product is delivered to the customer as early as possible, making it easier to implement any changes in requirements. The team roles in XP are Developers, Customers, and Management.

XP Practices include:

1. *Planning Game*: there is a close interaction between the programmers and the customer to estimate the effort needed for implementation (programmers) and scope and timing of releases (Customer)
2. *Frequent Small Releases*: Release system as often as possible
3. *On-site Customer*: Customer has to be present and available full-time for the team
4. *Testing*: Test and run system continuously

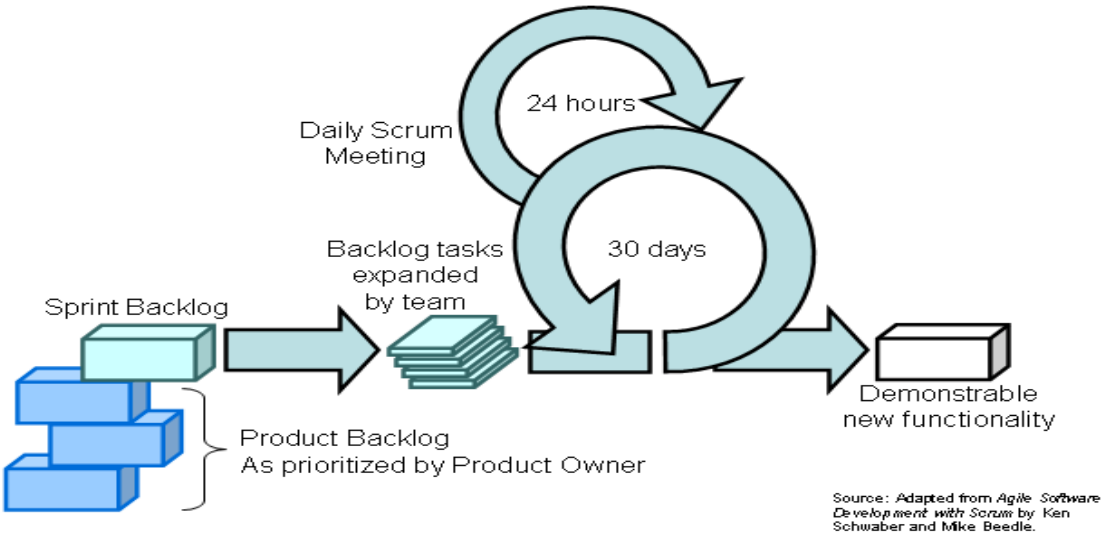
5. *Simple Design*: Design simplest possible solution that is implementable at the moment; avoid complexity, extra/duplicate code
6. *Refactoring*: Improve design of existing code by removing duplication, improving communication, simplifying and adding flexibility
7. *Pair Programming*: Two people get to develop all code concentrating on the same task; one person focuses on the task at hand 'driving', and the other focuses on the big picture 'navigating'
8. *Collective Code Ownership*: Any team member can change the code
9. *Continuous Integration*: Code is added as soon as it is ready
10. *Coding Standards*: Programmers must follow the coding rules; emphasizes communication through the code
11. *Metaphor*: System is defined by a metaphor/set of metaphors between the customer and programmers, which guide all development by describing how the system works
12. *40-hour Week*: A maximum of 40-hour working week. No two overtime weeks in a row are allowed

Every development process is different from one another. When using XP methodology, not all practices have to be selected. Practices should be tailored to suit the needs of the individual project.

1.6.2 SCRUM

SCRUM is an agile process use to manage and control the development process. SCRUM is a team-based approach to develop systems and products iteratively and incrementally. SCRUM methodology is flexible throughout the project development process. In the SCRUM process there are four phases:

1. *Planning*: Deep analysis and conceptualization if developing a system from scratch. Limited analysis if enhancing existing system.
2. *System architecture*: Design how backlog items will be implemented. System architecture modification and high level design.
3. *Development Sprints*: Sprints are short development phases where product is delivered incrementally. The variables of time, requirements, quality, cost, and competition are addressed in this phase. Multiple iterative sprints are used to evolve the system.
4. *Closure*: Usually completes the development process. Product is prepared for release, including final documentation, pre-release testing, and final release.



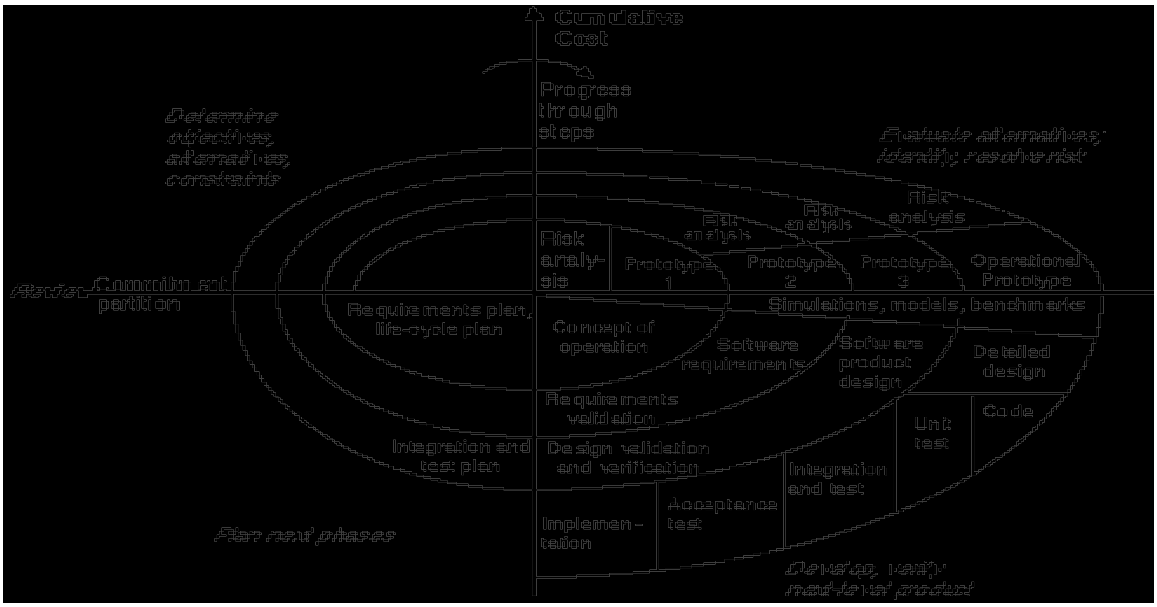
1.6.3 SPIRAL MODEL

The idea of a spiral model is to use the waterfall model for each step as an evolutionary development process. The primary reason to use the spiral model is to help manage risks. When using this model it is not necessary to define every detail of the new system being planned. Just define and implement the high priority features and give them to the customers for feedback. If needed then redefine/fix the features and then proceed to develop and implement more features in small units. This way the team is constantly receiving feedback from the end-users, helping the team deliver a better overall product that will meet all the expectations of the customers. With such constant feedback, the spiral model turns into an evolutionary development rather then your standard incremental development cycle.

Using this model does not give the development team a clear vision of the project due to the fact that this is an evolutionary process. The team only concentrates on the most important features and work their way from there to the less important features. Also since feedback is constantly coming back to the team, the project will sometimes go through transformations, meaning the team will not have a clear project vision until the very end of the development cycle.

Speed is one aspect that is not strictly enforced because the spiral model's main focus is risk management. Sometimes either by trying to avoid or deal with current risks the team may take a bit longer to finish the project. Measuring the progress could be done with the spiral model by looking at the finished features and the ones that have yet to be completed. The quality of the project is very high due to user feedback. One area that the spiral model is also good at is the ability to manage change. At any given time, the team is only focusing on one aspect of the

project, not the entire system, meaning if changes take place that effect later features it does not hurt the team or the progress.



1.6.4 UNIFIED PROCESS MODEL (UPM)

The Unified Process Model is designed for delivering various software development needs of organizations in order to finish the project on time, on budget and on target (10). It follows rapid application development (RAD) requirements to build a system that handles how to utilize UML. The concept follows process to product approach, in which the different components of SDLC are joined with the Rational set of software development tools. The UPM follows well-defined standard structure that uses an object-oriented approach. It covers modern software development technique to best practice and manages different projects in organizations.

It has many common characteristics with software products that use UML. Any upgrades related to UPM are available on line so that it is easier for any project manager to follow changes in the process. Regular software upgrades are released by Rational Software approximately twice a year. Therefore, the process is never obsolete, and its users benefit from the each of the newest developments. All team members can access the same version of the process. According to Rational's website, "Because it is modular and in electronic form, it can be tailored and configured to suit the specific needs of a development organization, something that's hard to do with a book or a binder. It is integrated with the many software development tools in the Rational Suites, so developers can access process guidance within the tool they are using."

1.6.5 RAPID APPLICATION DEVELOPMENT (RAD)

Rapid Application Development (RAD) is an iterative process for producing software. It is a very flexible methodology that stresses upon prototyping and user feedback. Since it implies a user driven design, it can also be referred to as the process or method of creating a new software system by involving the user community in all phases of the system creation. This methodology also facilitates early user acceptance, and ensures system functionality to all its users, before its final delivery.

The RAD methodology can be broken down into four phases. (5)

1. *Requirements Planning*: Managers, executives, and users determine and reach an agreement on the system requirements.
2. *User Design*: Users and analysts work closely and quickly to create prototypes that capture system requirements, which become the basis for the physical design of the system being developed.

3. *Construction*: Information Systems professionals generate codes using the CASE tools code generator.
4. *Cutover*: Delivery of the new system to its end users.

RAD supports the iteration and flexibility necessary for building robust business process support, emphasizing development speed. Since speed plays a significant part in this methodology, the goal of this process is to rapidly develop a high quality, low maintenance product, with a lot of participation as well as approval from the users.

There are many benefits of using the RAD methodology, associating with risks involved and quality of the product. As the phases of this methodology are completed and refined, gaining a broader knowledge of the system significantly reduces the risks. RAD realizes an overall reduction in project risk. Apart from the risk reduction this methodology has to offer, RAD projects also give a high priority to the quality of the product. The quality of the product may be linked to its availability, reliability, and maintainability, and therefore, the quality is guaranteed and assured under the project deadline.

RAD relies heavily on iterative prototypes in order to specify and document requirements. The iterations involved in this methodology give the process its flexibility to evolve and change, satisfying the end-users at every step of its phases. In conclusion, the RAD methodology shortens all phases of the life cycle into intensive work carried out by small functioning and highly participating teams.

1.6.6 METHODOLOGY METHOD SELECTION MATRIX

	MODEL / CRITERIA	*XP*	*UPM*	*SCRUM*	*RAD*	*Spiral*
15	***Project Vision***	25	20	25	20	5
15	***Iterative Nature***	25	25	25	25	20
15	***Speed***	20	20	25	25	10
5	***Project Integration***	10	10	15	10	15
5	***Ability to Measure Progress***	15	20	20	20	15
15	***Focus on Quality***	25	15	25	20	25
5	***Distribution of Duties***	15	20	20	25	15
10	***Risk Management***	20	15	20	25	25
10	***Ability to Manage Change***	20	5	25	25	25
5	***Establish Finish Line***	10	10	25	15	15
100		185	170	225	210	160

1.6.7 METHODOLOGY SELECTION

We researched a number of methodologies that we can incorporate into our *Web Based Appointment System Project.* After a very intensive and informative literature review, we chose to adopt a Rapid Application Development method, the Agile approach to software development. More specific we will be utilizing SCRUM as our software development methodology.

We took into consideration other methodologies including, XP, UPM, and Spiral. Once we established our criteria necessary to plan and arrange our workload amounts, we developed a Matrix system to evaluate which methodology would best suit our interests and our project goal. The Matrix assessed all the different methodologies against the criteria that we provided, giving SCRUM the highest points on the matrix.

SCRUM is an iterative process for managing system development processes. It is a very flexible methodology that stresses upon time, requirements, quality, prototyping and user feedback. SCRUM helps improve communications and maximize cooperation and productivity.

The Web-based appointment system requires a methodology that reduces cycle time and risks, encompass a high iterative nature and overall produce a system that will fulfill the users' needs and requirements. We are stressing a lot on the time and quality variables being we are developing our system within a short period of time. We like to deliver a quality product which will exceed our customer's expectation in a short period of time and all these traits are best suited for SCRUM.

1.7 GLOSSARY

Abstract: The degree to which a system or component performs only the necessary functions relevant to a particular purpose.

Analysis: A set of activities that attempt to understand and model customer needs and constraints.

Brainstorming: The unrestrained offering of ideas or suggestions by all members of a committee, conference, etc. in an effort to find a solution to a problem and generate fresh ideas.

Client/Server: Client / server describes the relationship between two computer programs in which one program, the client, makes a service request from another program, the server, which fulfills the request.

Context Diagram: An overview of an organizational system that shows the system boundaries, external entities that interact with the system, and the major information flows between the entities and the system.

Data Dictionary: Is a collection of descriptions of the data objects or items in a data model for the benefit of programmers and others who need to refer to them. It is a database that contains definitions of all data items defined during analysis.

Data Flow Diagram (DFD): A modeling notation that represents a functional decomposition of a system. The further decomposition of the Context Diagram.

Entity – Relationship Diagram: Is a data modeling technique that creates a graphical representation of the entities, and the relationships between entities, within an information system.

Feasibility study: Determines if the information system makes sense for the organization from an economic and operational standpoint.

Gantt chart: A graphical representation of a project that shows each task as a horizontal bar whose length is proportional to its time for completion.

Iteration: Is the repetition of a sequence of instructions.

Methodologies: Are comprehensive, multiple-step approaches to system development that will guide your work and influence the quality of your final product: the information system.

Milestones: A point in time that is used to indicate the progress during a project**.**

PERT chart: A diagram that depicts project tasks and their interrelationships. PERT stands for Program Evaluation Review Technique.

Prototype: The main deliverable of a phase that includes a working version of a product's technical features.

Security: The ability of a system to manage, protect, and distribute sensitive information.

Stakeholders: A person who has an interest in existing or new information system.

TCP/IP: The connection-oriented protocol built on top of Internet Protocol (IP) and is nearly always seen in the combination TCP/IP (TCP over IP).

Telnet: Is a user command and an underlying TCP/IP protocol for accessing remote computers.

Use Case: is a description of an interaction between an actor and a system.

Work Breakdown Structure (WBS): the set of work tasks required to build the software; defined as part of the process model.

2. PROJECT MANAGEMENT

2.1 RESOURCES MANAGEMENT

2.1.1 WORK BREAKDOWN STRUCTURE (WBS)

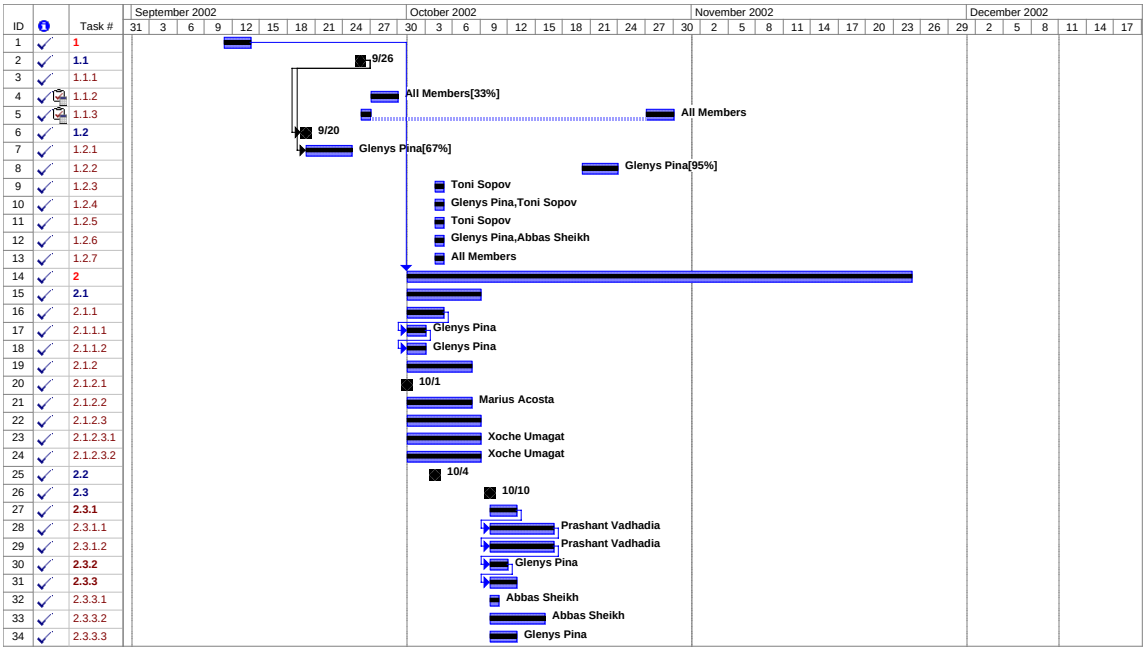
Task#		Name	Number of days	Start Date	End Date	Predecesors	Resources
1	**1**	**Project Initiation**	**3 days**	**09/11/02**	**09/13/02**		
2	1.1	Initiation	4 days	09/26/02	09/29/02		
3	1.1.1	Interview Soponsors	0 days	09/26/02	09/26/02		Glenys Pina, Abbas Sheikh, Toni Sopov, Xoche Umagat
4	1.1.2	List Project Needs	3 days	09/27/02	09/29/02	3	All Members
5	1.1.3	Define Project Goals	4 days	09/26/02	10/29/02	3	All Members
6	1.2	Planning	15 days	09/20/02	10/04/02	2	
7	1.2.1	Project Abstract	5 days	09/20/02	09/24/02	2	Glenys Pina
8	1.2.2	Team Roles and Responsibilities	4 days	10/20/02	10/23/02		Glenys Pina
9	1.2.3	Background	1 day	10/04/02	10/04/02		Toni Sopov
10	1.2.4	Problem Statement	1 day	10/04/02	10/04/02		Glenys Pina, Toni Sopov
11	1.2.5	Previous Work	1 day	10/04/02	10/04/02		Toni Sopov
12	1.2.6	Methodologies	1 day	10/04/02	10/04/02		Glenys Pina, Abbas Sheikh
13	1.2.7	Glossary	1 day	10/04/02	10/04/02		All Members
14	**2**	**Development Process**	**55 days**	**10/01/02**	**11/24/02**	**1**	
15	2.1	Project Management	8 days	10/01/02	10/08/02		
16	2.1.1	Resource Management	4 days	10/01/02	10/04/02		
17	2.1.1.1	Work Breakdown Structure	2 days	10/01/02	10/02/02	16	Glenys Pina
18	2.1.1.2	Project Milestones	2 days	10/01/02	10/02/02	17	Glenys Pina
19	2.1.2	Feasibility Study	7 days	10/01/02	10/07/02		
20	2.1.2.1	Economic Feasibility	7 days	10/01/02	10/07/02		Marius Acosta
21	2.1.2.2	Technical Feasibility	7 days	10/01/02	10/07/02		Marius Acosta
22	2.1.2.3	Cost Estimation	8 days	10/01/02	10/08/02		
23	2.1.2.3.1	Function Points	8 days	10/01/02	10/08/02		Xoche Umagat
24	2.1.2.3.2	Cost Estimation Model (COCOMO)	8 days	10/01/02	10/08/02		Xoche Umagat
25	2.2	1st Prototype Release	4 days	10/04/02	10/07/02		Glenys Pina
26	2.3	System Analysis	7 days	10/10/02	10/16/02		
27	2.3.1	Business Models	3 days	10/10/02	10/12/02		
28	2.3.1.1	As-Is System	7 days	10/10/02	10/16/02	27	Prashant Vadhadia
29	2.3.1.2	To-Be System	7 days	10/10/02	10/16/02	28	Prashant Vadhadia
30	2.3.2	Stakeholders Identification	2 days	10/10/02	10/11/02	29	Glenys Pina
31	2.3.3	Gathering Requirements	3 days	10/10/02	10/12/02	30	
32	2.3.3.1	Interviewing	1 day	10/10/02	10/10/02		Abbas Sheikh
33	2.3.3.2	Survey/Questionnaire	6 days	10/10/02	10/15/02		Abbas Sheikh
34	2.3.3.3	Use Cases	3 days	10/10/02	10/12/02		Glenys Pina

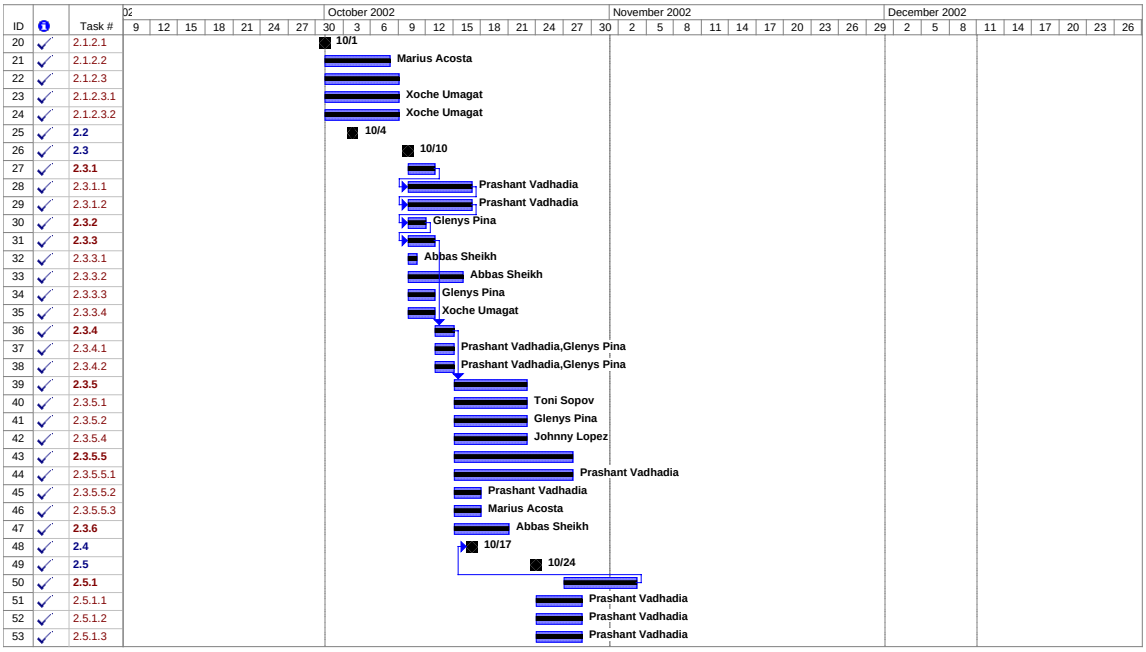
35	2.3.3.4	Brainstorming	3 days	10/10/02	10/12/02		Xoche Umagat
36	2.3.4	Documenting Requirements	2 days	10/13/02	10/14/02	31	
37	2.3.4.1	Functional Requirements	2 days	10/13/02	10/14/02		Prashant Vadhadia, Glenys Pina
38	2.3.4.2	Non-Functional Requirements	2 days	10/13/02	10/14/02		Prashant Vadhadia, Glenys Pina
39	2.3.5	Modeling Requirements	8 days	10/15/02	10/22/02	36	
40	2.3.5.1	Data Flow Diagrams	8 days	10/15/02	10/22/02		Toni Sopov
41	2.3.5.2	Use Case Diagram	8 days	10/15/02	10/22/02		Glenys Pina
42	2.3.5.4	Data Dictionary	8 days	10/15/02	10/22/02		Johnny Lopez
43	2.3.5.5	Process Specifications	13 days	10/15/02	10/27/02		
44	2.3.5.5.1	Structured English	13 days	10/15/02	10/27/02		Prashant Vadhadia
45	2.3.5.5.2	Decision Trees	3 days	10/15/02	10/17/02		Prashant Vadhadia
46	2.3.5.5.3	Decision Tables	3 days	10/15/02	10/17/02		Marius Acosta
47	2.3.6	SPSS Analysis	6 days	10/15/02	10/20/02		Abbas Sheikh
48	2.4	2nd Prototype Release	5 days	10/17/02	10/21/02	50	Glenys Pina
49	2.5	System Design	7 days	10/24/02	10/30/02		
50	2.5.1	System Structuring	8 days	10/27/02	11/03/02		
51	2.5.1.1	Repository Model	5 days	10/24/02	10/28/02		Prashant Vadhadia
52	2.5.1.2	Client-Server Model	5 days	10/24/02	10/28/02		Prashant Vadhadia
53	2.5.1.3	Abstract Machine Model	5 days	10/24/02	10/28/02		Prashant Vadhadia
54	2.5.2	Control Models	8 days	10/24/02	10/31/02		
55	2.5.2.1	Sequence Diagram	12 days	10/24/02	11/04/02		Marius Acosta
56	2.5.2.2	State Chart Diagram	12 days	10/24/02	11/04/02		Marius Acosta
57	2.5.3	Modular Decomposition	8 days	10/24/02	10/31/02		
58	2.5.3.1	Structure Charts (To-Be Systems)	14 days	10/24/02	11/06/02		Toni Sopov
59	2.5.4	Database Design	10 days	10/24/02	11/02/02		
60	2.5.4.1	ER Diagram	6 days	10/10/02	10/15/02		Johnny Lopez
61	2.5.4.2	ER Model	10 days	10/24/02	11/02/02		Johnny Lopez
62	2.5.4.2	Database Description	1 day	10/24/02	10/24/02		Johnny Lopez, Abbas Sheik
63	2.5.5	User Interface Design	3 days	10/24/02	10/26/02		Glenys Pina
64	2.5.6	3rd Prototype Release	5 days	10/20/02	10/24/02		Glenys Pina, Xoche Umagat
65	2.6	Software Detailed Design	7 days	10/31/02	11/06/02	64	
66	2.6.1	Document each unit, database and tests	20 days	11/07/02	11/26/02	48	Glenys Pina, Xoche Umagat, Marius Acosta, Abbas Sheikh
67	2.6.2	Conduct and document unit testing	2 days	11/07/02	11/08/02		Abbas Sheikh
68	2.6.3	Update user documentation	2 days	11/07/02	11/08/02		All Members
69	2.6.4	Update integration test requirements	2 days	11/07/02	11/08/02	68	All Members
70	2.6.5	Evaluate code and test results	2 days	11/07/02	11/08/02	10	All Members
71	2.6.6	Design User Manuals	20 days	11/21/02	12/10/02		Abbas Sheikh
72	2.6.7	Design Developer Manual	13 days	11/28/02	12/10/02	70	Glenys Pina, Xoche Umagat

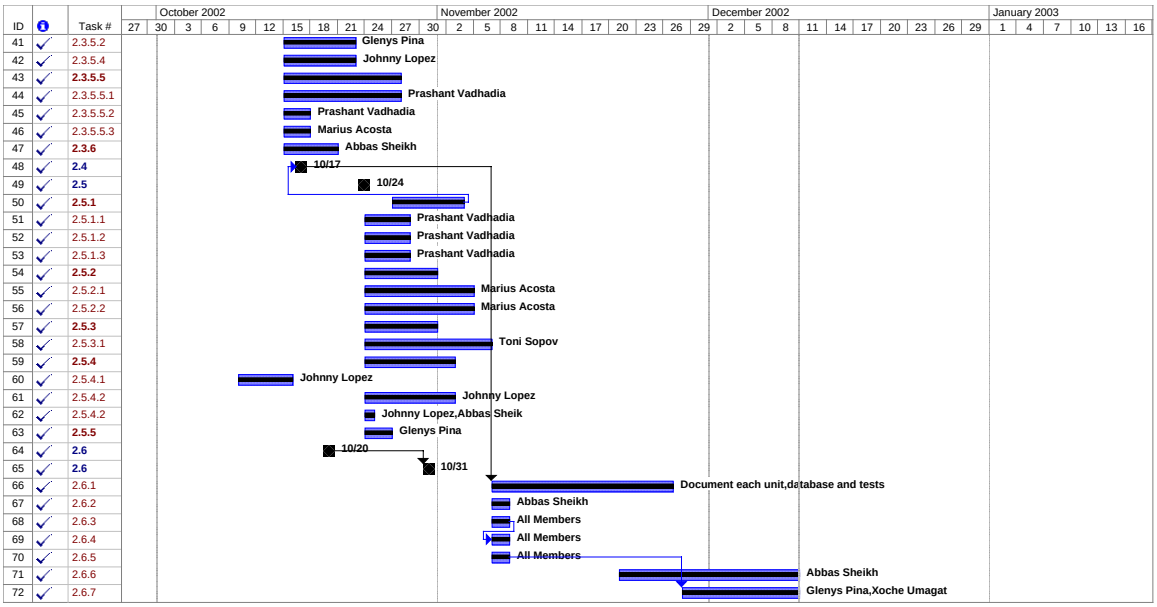
2.1.2 PROJECT MILESTONES

	PHASE (DATE)			
PHASE IMPLEMENTED	**START**	**END**	**ALLOCATED RESOURCES**	**PHASE IMPLEMENTED**
PHASE I - Introduction	09/11/02	09/25/02	Marius M. Acosta Johnny N. Lopez Glenys Pina Abbas Sheikh Toni Sopov Xoche B. Umagat	PHASE I - Introduction
PHASE IIA - Project Planning	09/26/02	10/09/02	Johnny N. Lopez Glenys Pina Abbas Sheikh Toni Sopov Prashant Vadhadia	PHASE IIA - Project Planning
PHASE II B - Project Feasibility	09/26/02	10/09/02	Marius M. Acosta Johnny N. Lopez Glenys Pina Toni Sopov Xoche B. Umagat	PHASE II B - Project Feasibility
PHASE II C - First Prototype	10/07/02	10/09/02	Glenys Pina	PHASE II C - First Prototype
PHASE IIIA - Requirements Engineering Application	10/10/02	10/23/02	Prashant Vadhadia, Glenys Pina, Abbas Sheikh, Xoche B. Umagat Johnny N. Lopez, Toni Sopov, Marius M. Acosta	PHASE IIIA - Requirements Engineering Application
PHASE IIIB - Second Prototype	10/10/02	10/23/02	Glenys Pina	PHASE IIIB - Second Prototype
PHASE IV - Architechtural Design	10/24/02	11/06/02	Marius M. Acosta, Glenys Pina, Abbas Sheikh, Johnny Lopez, Prashant Vadhadia	PHASE IV - Architectural Design
PHASE IVB - Third Prototype	10/24/02	11/06/02	Glenys Pina Xoche Umagat	PHASE IVB - Third Prototype
PHASE VA - Full Documentation	11/07/02	12/11/02	Toni Sopov, Prashant Vadhadia	PHASE VA - Full Documentation
PHASE VB - Final Software Product	11/07/02	12/11/02	Glenys Pina, Xoche B. Umagat	PHASE VB - Final Software Product
PHASE VC – Software Testing	11/07/02	12/11/02	Abbas Sheikh, Prashant Vadhadia Glenys Pina	PHASE VC – Software Testing, and user documentation

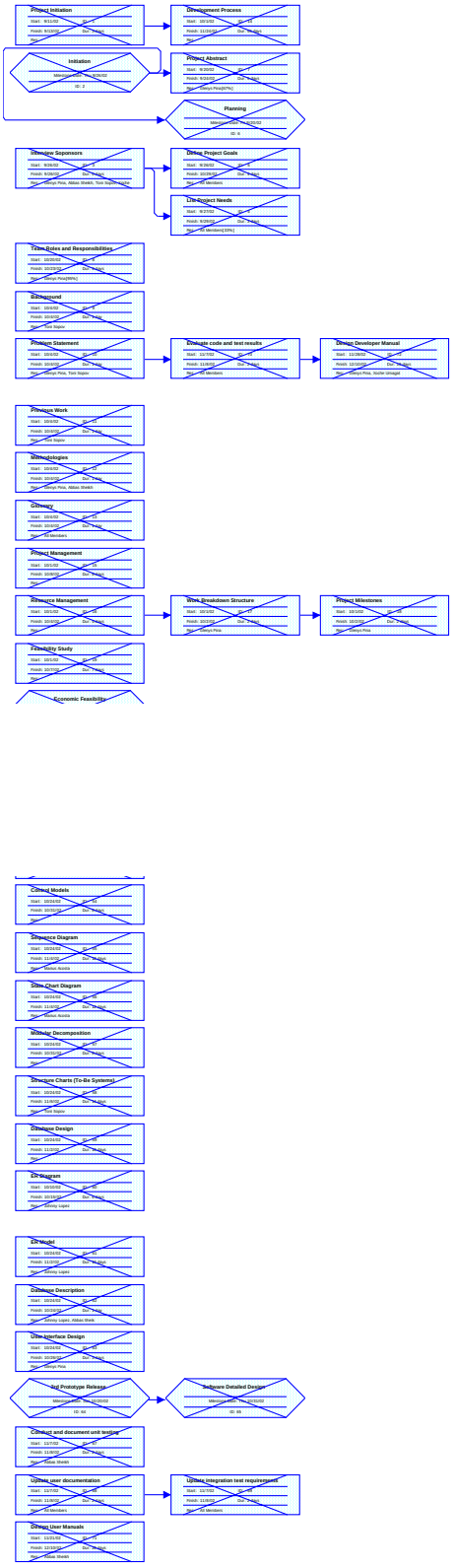
2.1.3 GANTT CHART






October 2002
November 2002
December 2002
January 2003
27 30 3 6 9 12 15 18 21 24 27 30 2 5 8 11 14 17 20 23 26 29 2 5 8 11 14 17 20 23 26 29 1 4 7 10 13 16
ID
Task #
41 2.3.5.2 Glenys Pina
42 2.3.5.4 Johnny Lopez
43 2.3.5.5
44 2.3.5.5.1 Prashant Vadhadia
45 2.3.5.5.2 Prashant Vadhadia
46 2.3.5.5.3 Marius Acosta
47 2.3.6 Abbas Sheikh
48 2.4 10/17
49 2.5 10/24
50 2.5.1
51 2.5.1.1 Prashant Vadhadia
52 2.5.1.2 Prashant Vadhadia
53 2.5.1.3 Prashant Vadhadia
54 2.5.2
55 2.5.2.1 Marius Acosta
56 2.5.2.2 Marius Acosta
57 2.5.3
58 2.5.3.1 Toni Sopov
59 2.5.4
60 2.5.4.1 Johnny Lopez
61 2.5.4.2 Johnny Lopez
62 2.5.4.2 Johnny Lopez,Abbas Sheik
63 2.5.5 Glenys Pina
64 2.6 10/20
65 2.6 10/31
66 2.6.1 Document each unit,database and tests
67 2.6.2 Abbas Sheikh
68 2.6.3 All Members
69 2.6.4 All Members
70 2.6.5 All Members
71 2.6.6 Abbas Sheikh
72 2.6.7 Glenys Pina,Xoche Umagat

2.1.4 PERT CHART



2.2 FEASIBILITY STUDY

2.2.1 COST BENEFIT ANALYSIS (ECONOMIC FEASIBILITY)

One-Time Cost Worksheet:

	Year 0
New hardware	$15,000
New (purchased) Software:	
Package application software	$5,000
Frontpage 2000	$130
Communication equipment installation	$7,500
Website Development Team	
a. Development Project Manager	$35,000
b. Architect Developer	$25,000
c. Creative Designer	$15,000
Total	$102,630

Tangible Benefits Worksheet:

Project Initiation Fee		
Department	$50,000	
Top-level admin.	$40,000	
Cost reduction or avoidance (1)	$5,000	
Technology based system (2)	$3,000	
(error reduction)		
Federal Funding	$10,000	
State Funding	$15,000	
Improvement in management	$1,000	
Other (3**)**	$40,000	
Total	$164,000	
Note: (1) cost reduction or avoidance is calculated by the		
cost of reprinting appointment manual sheets and binders.		
(2) technology based system means that the overhead of		
creating hard copies of appointment sheets is replaced		
by advanced technology systems.		
(3) 2 alternating front desk secretary position that can be		
eliminated.		

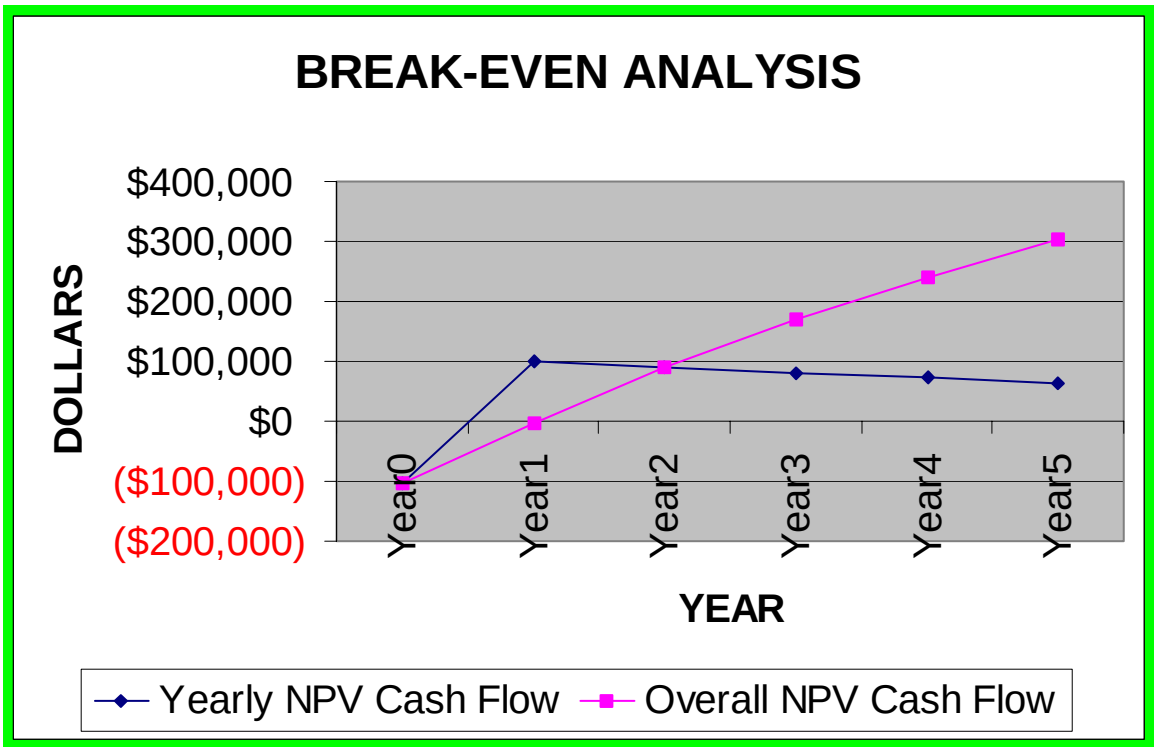
Recurring Cost Worksheet:

	Year 1 through 5
Website Maintenance	
a. Technical Project Manager	$35,000
b. Student Workstudy	$5,000
Web server Upgrades and Maintenance	$5,000
Supplies	$1,000
Managing development	$5,000
Total	$51,000

Economic Feasibility Analysis

	Year 0	*Year 1*	*Year 2*	*Year 3*	*Year 4*	*Year 5*	TOTALS
Net Economic Benefit	$0	$164,000	$164,000	$164,000	$164,000	$164,000	
Discount Rate (12%)	***1.0000***	***0.8929***	***0.7972***	***0.7118***	***0.6355***	***0.5674***	
PV of Benefits	$0	$146,435.60	$130,741	$116,735	$104,222	$93,054	
NPV of all BENEFITS	$0	$146,436	$277,176	$393,912	$498,134	$591,187	**$591,187**
One-time COSTS	($102,630)						
Recurring Costs	$0	($51,000)	($51,000)	($51,000)	($51,000)	($51,000)	
Discount Rate (12%)	***1.0000***	***0.8929***	***0.7972***	***0.7118***	***0.6355***	***0.5674***	
PV of Recurring Costs	$0	($45,537.90)	($40,657.20)	($36,301.80)	($32,410.50)	($28,937.40)	
NPV of all COSTS	($102,630)	($148,167.90)	($188,825.10)	($225,126.90)	($257,537.40)	($286,474.80)	**($286,474.80)**
Overall NPV							**$304,712.20**
Overall ROI (overall NPV / NVP all COSTS)							**1.064**
Break-even Analysis							
Yearly NPV Cash Flow	($102,630)	$100,897.70	$90,083.60	$80,433.40	$71,811.50	$64,116.20	
Overall NPV Cash Flow	($102,630)	($1,732.30)	$88,351.30	$168,784.70	$240,596.20	$304,712.40	
Project break-even occurs bet. Yr. 1& yr.2							
Break-even fraction - ((90,083.60-88,351.30 / 90,083.60) = 0.02							

Break-even Analysis



2.2.2 TECHNICAL FEASIBILITY

According to Hoffer, technical risks associated with any projects usually include factors such as project size, project structure, development group experience in the subject, and the user group's experience with development project and application area. Our project is a relative small project. We will be developing the system for one particular NJIT department, which might be extended to other departments upon successful project completion. The requirements for the project have been clearly specified by our sponsors. Our group consist of seven members each one with experiences in different areas, such as programming, web development, database design, etc. As a group, members are not experts in the subject but are familiar with the specified functions for each one of the roles, as well as with the most of the software development environment tools.

There are currently companies providing web based appointment system services to different organizations. These will be deeply research and its technology will be used as a guide/framework for our project. Each member is willing to learn new tools to be used to enhance and facilitate our project development process.

This project can be viewed as having medium-low risk according with our technical feasibility study.

2.2.3 COST ESTIMATION

2.2.3.1 FUNCTION POINTS ANALYSIS

Accurately planning and estimating software projects is an extremely difficult software management function. One way to estimate the size of a project uses Function Points. A Function Point is a measure of program size that is based on the number and complexity of inputs, outputs, files, and program interfaces.

Function Points give software engineering researchers a way of sizing software through the analysis of the implemented functionality of a system from the user's point of view. They provide a way to predict the number of source code statements that must be written for a program or system. Languages have varying, but characteristic, levels. The level is the average number of statements required to implement one Function Point.

There are a variety of factors that can impact the complexity of the project. Fourteen factors such as data communication, reusability, end-user efficiency, etc., are considered and assessed in terms of their effect on the project’s complexity.

In determining the ultimate size of the project, the Function Points Analysis computes the Total Adjusted Function Points (TAFP). The TAFP can then be used in the Cost Construction Model (COCOMO) to estimate the effort on the project.

Function Point Count

Function Counts	Complexity			Total
	Low	Average	High	
External Input	*3	3*4	*6	12
External Output	*4	4*5	*7	20
Internal Logical File	3*7	*10	*15	21
External Interface File	*5	*7	7*10	70
External Inquiry	*3	5*4	*6	20
			Total Unadjusted Function Points (TUFP)	143

Value Adjustment Factor

Factor	***Value****
Data Communications	3
Distributed Functions	1
Performance	3
Heavily Used Configuration	2
Transaction Rate	4
Online Data Entry	4
End User Efficiency	4
Online Update	4
Complex Processing	2
Reusability	3
Installation Ease	2
Operational Ease	3
Multiple Sites	0
Facilitate Change	3
Project Complexity (PC)	38

* 0 = No Effect on Process Complexity; 5 = Great Effect on Process Complexity

Adjusted Project Complexity (PCA) = 0.65 + (0.01 * PC)

0.65 + (0.01 * 38) => **1.03**

Total Adjusted Function Points (TAFP) = PCA * TUFP

1.03 * 143 => **147.29**

Function Points

Language	*Approximate Number of Lines of Code per Function Point*
C	130
COBOL	110
Java	55
C++	50
Turbo Pascal	50
Visual Basic	30
Power Builder	15
HTML	15
Packages (Access, Excel, etc.)	10-40

The following will be used in the Web Based Appointment System

Visual Basic = 50 * 30 = 1500
HTML /ASP = 40 * 15 = 600
Access = 60 * 30 = 1800

Total = 3900 Lines of Code

2.2.3.2 Cost Construction Model (COCOMO)

COCOMO is a simple cost model for estimating the number of person-months required to develop software. The model also estimates the development schedule in months and produces an effort and schedule distribution by major phases. Barry Boehm designed and developed this model. COCOMO is applicable to the large majority of software projects.

COCOMO is good for rough order of magnitude estimates of software costs, but its accuracy is necessarily limited because of its lack of factors to account for differences in hardware constraints, personnel quality and experience, use of modern tools and techniques, and other project attributes known to have a significant influence on costs.

COCOMO uses the following three modes to estimate cost.

- **Organic**

  It is for a relatively small software teams develop software in a highly familiar, in-house environment. Most people connected with the project have extensive experience in working with related systems within the organization, and have a thorough understanding of how the system under development will contribute to the organizations objectives.

- **Semi-detached**

  It represents an intermediate stage between the organic and embedded modes. "Intermediate" may mean either of two things: an intermediate level of project characteristic or a mixture of the organic and embedded mode characteristics.

- **Embedded**

  It operates within tight constraints. The product must operate within (is embedded in) a strongly coupled complex of hardware, software, regulations, and

operational procedures, such as an electronic funds transfer system or an air traffic control system.

COCOMO

Effort Adjustment Factors		
Project Constraints	**Complexity**	**Adjustment Factors**
Database Size	Medium	1.0
Product Complexity	Medium	1.0
Main Storage	Very High	1.21
Programmer Capability	Very High	1.25

Semi-Detached

Let PM = Person-Month

PD = Project Duration
S = Staffing
KSLOC = 3900 SLOC/1000 = 3.9 KSLOC

Effort:
PM = EAF * 3.0 * ((KSLOC) /\ (1.12))
= (1.0 * 1.0 * 1.21 * 1.25) * 3.0 * (3.9 /\ 1.12)
= 20.84 => 21

Project Duration:
PD = 2.5 * (PM)/\ 0.35
= 2.5 * (21)/\0.35
= 7.26

Staffing:
S = PM/PD
= 21/7.26
= 2.89 => 3

There are 3 people needed to finish the project in 7.26 months or

A group of 7 people could finish the project in 3.11 months.

2.3 RISK MANAGEMENT

Project Size

A team of 7 individuals will create this online appointment system for the Computer Science Academic department. Due to the fact that is system is not overly complex and the needed tools are readily available by various vendors, the creation will not be a large undertaking. The target users of this system once it is implemented will be the students, and the advisors. A system administrator will also be needed to maintain and update the system on a regular basis, but since the software is not complicated, the school can use existing personnel for this task.

Project Structure

The project will involve taking the existing paper based arrangement and revamping it to be an online system. The analysis of the current system led to the realization that it is still functional, but not very efficient for neither the students, the advisors, nor the sectaries that have to put aside time from their busy schedules to make sure the appointments are made accordingly. The information that needs to be gathered form the end users, for the system to work properly, are straightforward and very simple. This leads to a very low risk of failure, which makes the project attain the expected benefits with ease.

Familiarity with Technology or Application Area

The group members are knowledgeable enough on the various tools needed to create this system that it should not lead to any sort of big problem in the near future. At the same time, the members are either brushing up on key concepts they have learned from past experiences or learning new techniques that will help out the team achieve the desired goal(s). In addition, the end-users also should not have any problems with the system because it will use a very simple GUI. The technologies needed to be integrated onto the system; are database systems software,

applications developments software, web and active server pages programming. Right now, keeping database integrity seems to be the greatest technical risk.

Time Constraint

This is one aspect that unfortunately we have no control over. Right now, the time frame is adequate enough to create and implement the new system. However since unforeseen problems may rise in the future, the current time frame does not have much room for adjustments. As a result, this project is very time dependent, giving the team a total of about 13 weeks to work on all the documentation, analysis, the creation and implementation of the system. However we are countering this by using the process of a rapid development cycle, which allows the team to have prototypes at the different stages of the cycle, thus permitting the team to correct any problems that could have come up toward the end of the project. This gives plenty of time to deal with such tribulations.

Systems Interdependence

Since this a standalone system, the team feels this should not pose any integration problems. The system will have links to other systems but these are just requests for information nothing more.

Allocation of Tasks

The project contains many different tasks related to different phases that are essential to be completed in order to have well comprehensive project. All tasks require through familiarity in specific area that each task demands. It is well taken in to account that different tasks are assigned with respect to team member's experience related to specific task. When the team members are familiar with the specific assigned task; then the final product will be developed *on time, on budget, and on target*.

In order to achieve above-mentioned criteria; Regular meeting session ranging from 2-4 hours, Continuous Process Improvement and Team Collaboration have been done extensively throughout the System Development Life Cycle. Before assigning any task it is made sure that the members are thoroughly familiar with the specific assigned task. The System depends heavily on way the different tasks have been distributed among the team members.

Project Size	Small project with already existing tools available for the creation of the system. Also the system will not be a large undertaking. ***Risk: Low***
Project Structure	The requirements are well defined and the data needed for the system to function are simple and structured. The existing paper based system made the analysis simple allowing the team to gather the system requirements easily. ***Risk: Low***
Familiarity with Technology or Application Area	The group members are knowledgeable enough on the various tools needed to create this system that it should not lead to any sort of big problem in the near future. Members are either brushing up on key concepts they have learned from past experiences or learning new techniques that will help out the team achieve the desired goal(s). ***Risk: Low***
Time Constraint	Time frame is adequate enough to create and implement the new system, however leaves no room for unforeseen problems. Countering this by using the process of a rapid development cycle, which allows the team to have prototypes at the different stages of the cycle, ***Risk: Medium***
Systems Interdependence	Since this a standalone system, the team feels this should not pose any integration problems. ***Risk: Low***
Allocation of Tasks	When the team members are familiar with the specific assigned task; then the final product will be developed *on time*, *on budget*, *and on target*. In order to achieve above-mentioned criteria; Regular meeting session ranging from 2-4 hours, Continuous Process Improvement and Team Collaboration have been done extensively throughout the System Development Life Cycle. ***Risk: Medium***

3. SYSTEMS ANALYSIS

3.1 PROBLEM STATEMENT

Academic advising is an essential component of student's success in higher education. Students are guided with course selection and appropriate prerequisites. This way, students are prepared and well equipped to progress from one course to another. It also makes the learning valuable and fun.

The constant yearly increase in enrollment has posed a problem with academic department's limited resources. To make an appointment for advising becomes a difficult task. This is coupled with students' busy schedules, and advisors' responsibilities outside of advising. Each semester, a constant long file of students can be seen lining up to see their advisors. Oftentimes, students find themselves late or unable to register with their courses. Advisors find themselves overwhelmed trying to accommodate students. The result defeats the very purpose of academic advising.

The advent of Internet and other telecommunication technologies can shed light on this problem. A web-based appointment system will enable students to schedule appointments wherever Internet connection is available.

3.2 BUSINESS MODELS

3.2.1 AS-IS SYSTEM

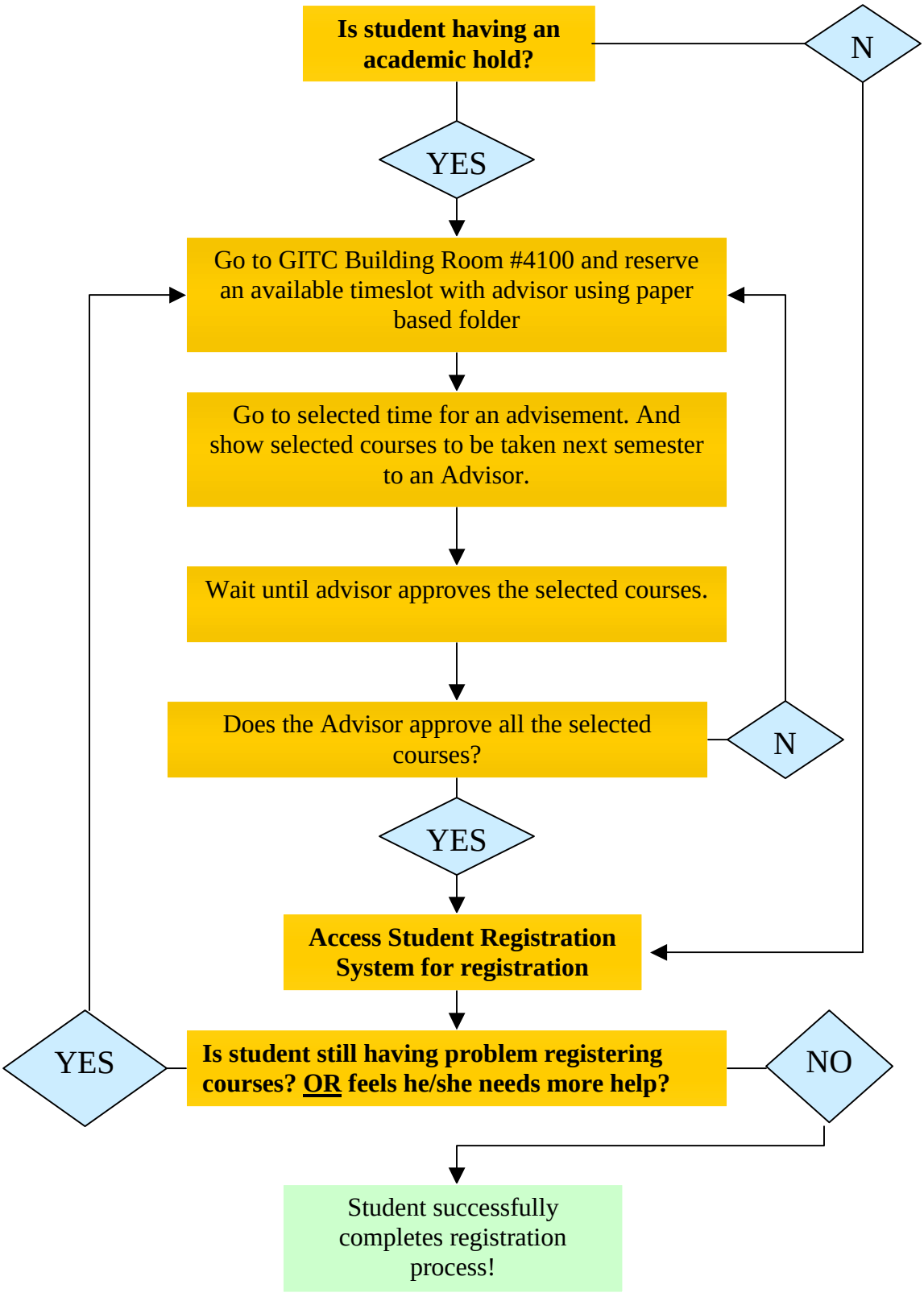
The As-Is System being utilized today for scheduling appointments with NJIT advisors is a “paper-based” system. The options being offer for scheduling appointments are time-consuming, unreliable and unorganized. In order to schedule an appointment, students must walk into the GITC building room 4100 office and search the appointment book for an available timeslot; or make a telephone call and wait for the secretary to tell them of the advisor’s available timeslots. During advisement period, students must manually fill out a course selection form to bring to the advisor the day of the appointment so the advisor can approve the selected courses. Finally after that is done, students are able to access the Student Registration System to complete the registration process.

3.2.2 TO-BE SYSTEM

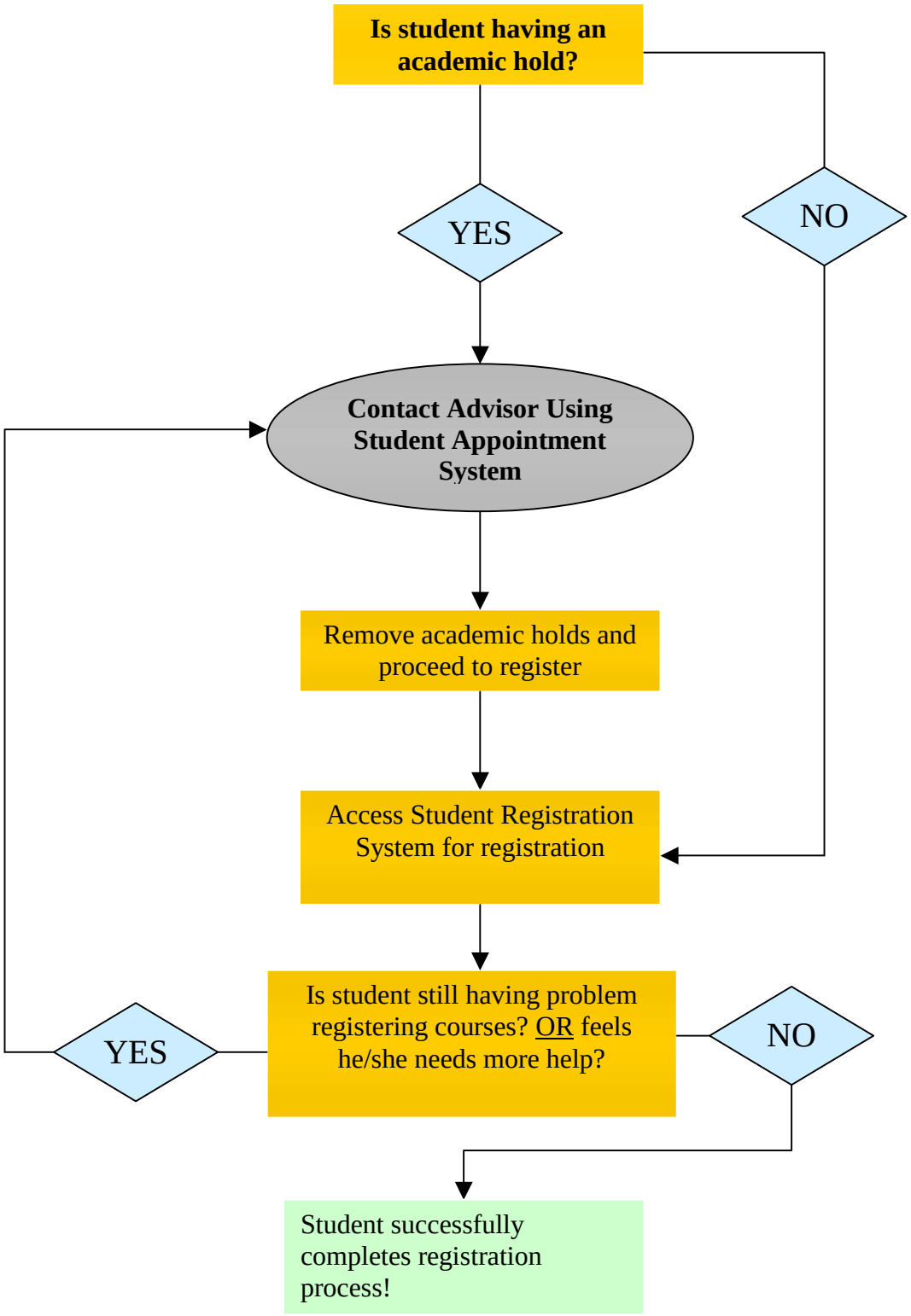
The To-Be System, which is the web-based appointment system, will offer a user-friendly interface, which will allow students and advisors to manage appointment scheduling more effectively. Students can go online anytime, review the advisor’s available days and timeslots, and schedule an appointment. To schedule an appointment, students must provide their name, student ID and email address. To complete the appointment, students must fill out the ‘CCS Course Selection Form’ online. Students can only schedule one appointment of 15 minutes timeslot limit. Once a timeslot is reserved for a student, its availability status will change. If a student wants to schedule a second appointment, he/she must first delete the current one avoiding duplicate appointments. The system will send an automated email to remind students of their appointment a day prior to the scheduled day. On the other hand, advisors can

go online at any time to check their schedule, make changes to their available timeslots, or cancel appointments. Overall, the system helps provide better service to students and advisors.

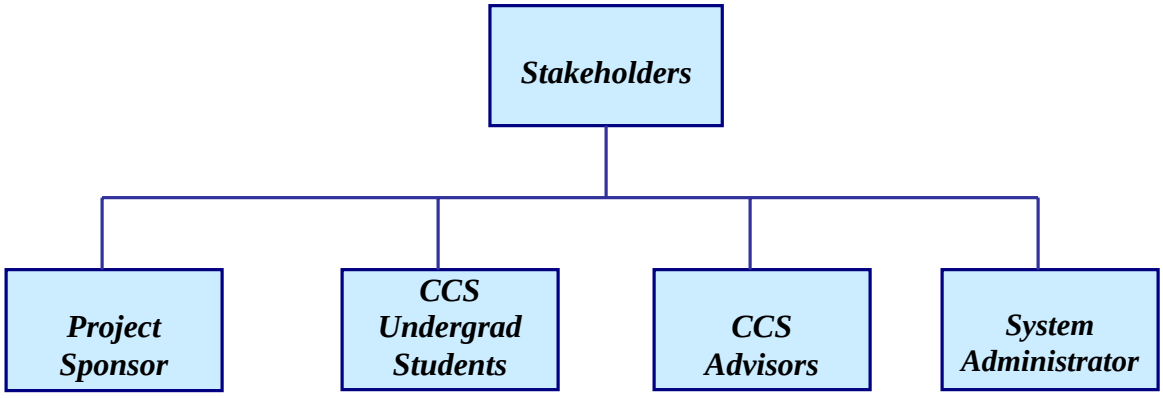
AS-IS System Business Model Diagram



TO-BE System Business Model Diagram



3.3 STAKEHOLDERS IDENTIFICATION



Project Sponsor (Michael Baltrush)	Dr. Baltrush has provided the idea of creating a Web based appointment system for the College of Computing Sciences (CCS). He is an active player during the whole development process who will be working closely with the project development team giving feedback and helping out. His requirements are really important and must always be taken into consideration. This project is his idea and therefore we must meet his requirements and expectations. If his requirements are not fulfilled, the system will be considered a failure and will not be implemented in the future by the CCS advisors.
CCS Undergraduate Students	CCS' undergraduate students are very important stakeholders to the system. The system will be restricted to those students which majors are *BSCS, BACS, BSIS, BAIS, BSIT, or BSHCI.* Students will be using the system to schedule appointments with their advisors. Their requirements have to be fulfilled entirely. The User-Interface Design should follow their requirements, so they will feel comfortable with the system and use it to the max.
CCS Advisors (Tom Moore and Kathy Sherman)	Moore and Sherman are very important stakeholders to the system. They will be interacting constantly with the system. Their requirements must be completely fulfilled. The User-Interface Design should follow their requirements. The system itself must be user friendly and help them manage their appointments in a more organized way. They have to specify their available time so students can schedule their appointments.

System Administrator The System Administrator will help the CCS maintain the system, including backup and recovery, adding and deleting user accounts, performing software updates, supplying the days and times for appointments, being able to rearrange or cancel appointments. System administrator will deal with security issues.

* For future system enhancements, the system will have more stakeholders including advisors from the different colleges among the school and graduate advisors. The system can be upgraded to allow advisors from other departments to use the system.

3.4 GATHERING REQUIREMENTS

3.4.1 TRADITIONAL METHODS

3.4.1.1 INTERVIEWING

After conducting a number of interviews, we concluded that most people had similar views and inputs about the traditional paper-based system and the future web-based appointment system. Upon completion of the interviews, we had the following results (All results are quoted):

1.) Sponsors

Tom Moore's reply to prototype:

"Consistency in the time slots with wording – possibly Available and Reserved throughout the schedule – including lunch which will simply read reserved…Overall, the model looks excellent…good job. Once this is implemented, it will be very helpful."

Dr. Baltrush's reply to prototype:

"Your best bet for testing is to create a bunch of names and student ids…It may be easier to add records as the student registers for an appointment…Putting all the student names in beforehand might entail extra effort."

Kathy Sherman's reply to prototype:

"I think that its great that students would need to submit a course selection form, if their purpose for requesting the appointment is pre-registration…I also agree with Tom that, as a whole, this looks great, and I'm looking forward to when we can use it. Thanks."

2.) Students

Replies to the current paper-based system:

"I do not like it because it is a very long process of making an appointment."

"If you want to change your appointment, then you can just scribble out your assigned time and make another appointment for yourself. But other students that come in at that time tend to get confused between the scribbled part and assigned part, and don't get to make their appointments because they think that the slots are taken."

Replies to the future web-based appointment system:

"I would like to see a stable website that can handle the appointments correctly and I think it will."

"The web-based appointment system will be very convenient for all students because you would be able to access your appointments from any computer…also, people won't have to walk to the GITC department just to make an appointment."

3.4.1.2 SURVEY/ QUESTIONNAIRE

Evaluation of the Advising Appointment System

1. What is your major?

CS	IS	IT

2. What is your current status?

Freshman	Sophomore	Junior	Senior

3. How familiar are you with the current process of making appointments with your advisor?

Extremely Familiar	Strongly Familiar	Familiar	Unfamiliar	Strongly Unfamiliar	Extremely Unfamiliar

4. How satisfied are you with the current paper-based system of making appointments?

Extremely Satisfied	Strongly Satisfied	Satisfied	Dissatisfied	Strongly Dissatisfied	Extremely Dissatisfied

5. Do you believe that the paper-based system is reliable?

Extremely Agree	Strongly Agree	Agree	Disagree	Strongly Disagree	Extremely Disagree

6. Do you believe that the paper-based system is time efficient?

Extremely Agree	Strongly Agree	Agree	Disagree	Strongly Disagree	Extremely Disagree

7. Do you believe that the paper-based system is organized?

Extremely Agree	Strongly Agree	Agree	Disagree	Strongly Disagree	Extremely Disagree

8. Would you prefer a web-based appointment system to the traditional paper-based system?

Extremely Agree	Strongly Agree	Agree	Disagree	Strongly Disagree	Extremely Disagree

9. Would you like to access the web-based appointment system through the NJIT Highlander Pipeline account?

Extremely Agree	Strongly Agree	Agree	Disagree	Strongly Disagree	Extremely Disagree

10. Would you like to access the web-based appointment system through your department's website?

Extremely Agree	Strongly Agree	Agree	Disagree	Strongly Disagree	Extremely Disagree

11. Would you like the system to display the advisor's calendar weekly?

Extremely Agree	Strongly Agree	Agree	Disagree	Strongly Disagree	Extremely Disagree

12. Would you like the system to display the advisor's calendar biweekly?

Extremely Agree	Strongly Agree	Agree	Disagree	Strongly Disagree	Extremely Disagree

13. Would you like the system to display the advisor's calendar monthly?

Extremely Agree	Strongly Agree	Agree	Disagree	Strongly Disagree	Extremely Disagree

14. Do you have a preference over the advisors when scheduling an appointment?

Extremely Agree	Strongly Agree	Agree	Disagree	Strongly Disagree	Extremely Disagree

15. Would you like to view all the necessary information to schedule an appointment on one page, consisting of the advisor's calendar, time availability, CCS course selection form, etc.?

Extremely Agree	Strongly Agree	Agree	Disagree	Strongly Disagree	Extremely Disagree

16. Would you like to view all the necessary information to schedule an appointment, through separate links in an organized format?

Extremely Agree	Strongly Agree	Agree	Disagree	Strongly Disagree	Extremely Disagree

17. Do you believe that there should be a gap between every appointment (each appointment would be 15 minutes)?

Extremely Agree	Strongly Agree	Agree	Disagree	Strongly Disagree	Extremely Disagree

18. Would you like to receive a confirmation that your appointment has been made, once you arrange one in an available time slot?

Extremely Agree	Strongly Agree	Agree	Disagree	Strongly Disagree	Extremely Disagree

19. Would you like to receive an email a day before your appointment, as a reminder of the scheduled appointment with your advisor?

Extremely Agree	Strongly Agree	Agree	Disagree	Strongly Disagree	Extremely Disagree

20. Would you consider the web-based appointment system as an effective approach to make appointments and replace the traditional paper-based system?

Extremely Agree	Strongly Agree	Agree	Disagree	Strongly Disagree	Extremely Disagree

3.4.2 MODERN METHODS

3.4.2.1 USE CASES

Actor: Student

1. WBAS website: The student visits the WBAS website which displays the option to login as a student or advisor.
2. Student Login Page: After selecting the 'login as a student' option on the main page, the student login page displays the option to schedule an appointment and edit course selection form or cancel a current appointment. The student provides the required information and select option.
    - Edit Course Selection Form/ Cancel Appointment option allow student to cancel scheduled appointment, or make changes to the submitted Course Selection Form. The student uses the keyboard to enter the SID. If SID is in the database, a page displaying student information such as appointment day and time and an option to edit the Course Selection Form will be display.
    - Schedule an appointment option. The student enters his/her first and last name, SID, and selects major and advisor. Student will be then redirected to a page with the advisors' calendar.
3. Schedule Appointment: The page display the advisor's calendar and advisor's available timeslots. Once student clicks on the preferred timeslot, he/she is required to fill out the Course Selection Form to confirm the appointment.
    - Course Selection Form page: The student enters his/her name, SS#, NJIT email address, major, semester and courses planning to take that particular semester. Upon submission of the form, the appointment is confirmed.

Actor: Advisor

1. WBAS website: The advisor visits the WBAS website which displays the option to login as an advisor or student.
2. Login to Database: The advisor utilizes the keyboard to enter his/her user name and password. The password is compared to the one stored on the database. If wrong password is entered, an error message will be display and he/she will be able to reenter the password. If both passwords match, the advisor's page will be displayed which includes the advisor's calendar.
3. View advisor calendar: Advisor calendar is display. When advisor clicks on the calendar to view a particular day's scheduled appointments, the student name will be display as a link. The student personal information will be displayed when the advisor clicks on the student name.
4. Cancel appointment: Advisor clicks on the cancel appointment option which will display the advisor's calendar. Advisor selects day and time he/she wants to cancel the appointment. Student names with scheduled appointment will be displayed. Advisor clicks on student to canceled appointment and an option to send an email is displayed. Advisor personalizes the email and sends it to the student.
5. Reserve time: Advisor clicks on the calendar day he/she wants to reserve time and blocks the timeslots to prevent students from reserving those timeslots.

Actor: System Administrator

1. Maintains Network: Maintains external and internal web presence. Performs system backups. Is able to cancel student appointment, and edit advisors' time availability.

Actor: Web Designer

1. Design/Construct Website: designs and constructs the website and its user interface features. Maintains and provides ongoing design for the website.

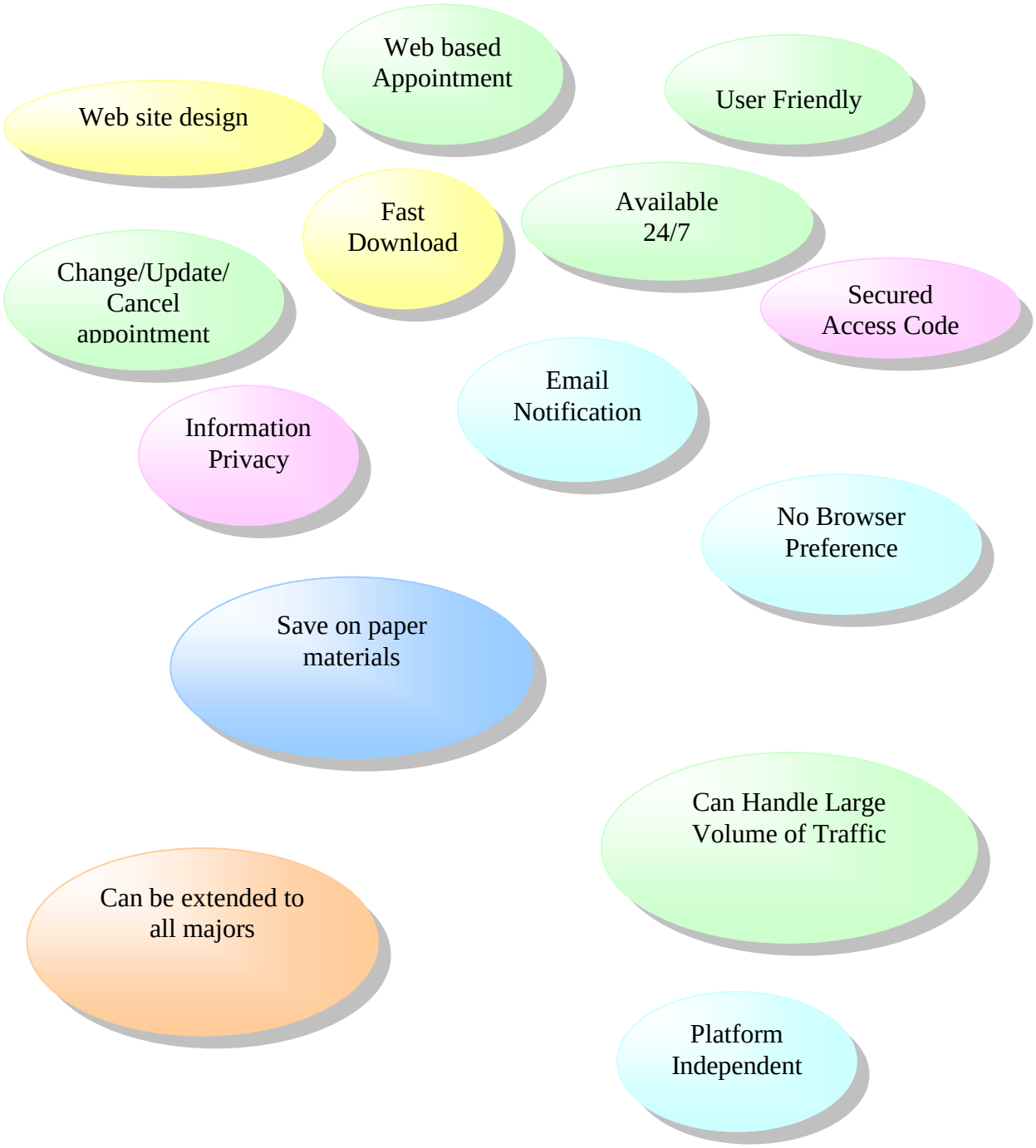
Actor: Web Programmer

1. Design/develop applications and scripts: designs and develop applications and scripts for the system. These applications and scripts will be interactive and will allow students and advisors to interact with the database.

Actor: Database Administrator

1. Plan, implement, and configure the system database: plans, implements, and configures the database to be used by the system. The database will hold student's records including student name, SID, Major, Course Selection Form, appointment date and time, as well as email address.

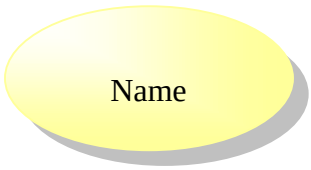
3.4.2.2 BRAINSTORMING



Categorization

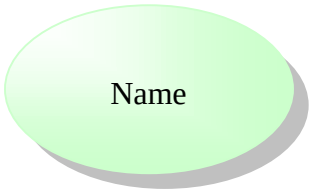
Speed

- Website design
- Fast download



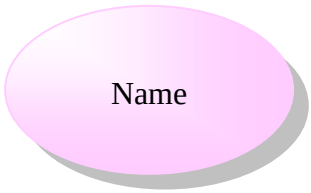
Convenience

- Web based Appointment
- User Friendly
- Email notification
- Available 24/7
- Change/Update/Cancel appointment online



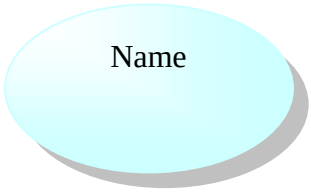
Security

- Secured Access Code
- Information Privacy



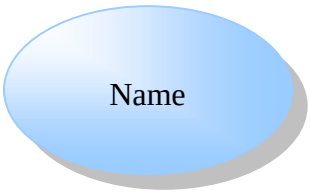
Efficiency

- Platform independent
- Can Handle Huge Volume of Traffic
- No Browser Preference



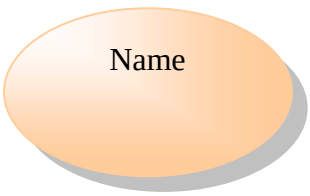
Benefits

- Save paper materials



Future Enhancement

- Can be extended to all majors



Prioritization

	Convenience	*Security*	*Efficiency*	*Speed*	*Benefits*	*Future Enhancement*	
Abbas	20	20	20	10	20	10	100
Glenys	40	30	20	10	0	0	100
Johnny	20	20	20	20	10	10	100
Marius	20	20	20	20	10	10	100
Prashant	20	20	20	20	10	10	100
Toni	20	20	20	20	10	10	100
Xoche	30	20	30	10	5	5	100
	170	160	150	110	65	55	

Ranking of Priorities

1. Convenience
2. Security
3. Efficiency
4. Speed
5. Benefits
6. Speed
7. Future Enhancements

3.5 DOCUMENTING REQUIREMENTS

3.5.1 FUNCTIONAL REQUIREMENTS

1. The Web based appointment system should be user friendly
2. The system should be updated frequently
3. The system should provide up-to-date information regarding advisors' time availability
4. The system should keep students' information secure and confidential
5. The system should facilitate the process of scheduling appointments with advisors
6. The system should be scalable
7. The system should be easy to manage and update
8. The system administrator should have full access and privileges to manage the system including editing advisor's time availability and adding new users
9. The system should be able to send automated email reminders to students
10. The system should allow advisors to view students' information before meeting
11. The system should allow advisors to reserve needed time, cancel appointments and email students to let them know of any changes
12. The system should allow students to schedule, cancel, reschedule appointments, and edit the course selection form

3.5.2 NON-FUNCTIONAL REQUIREMENTS

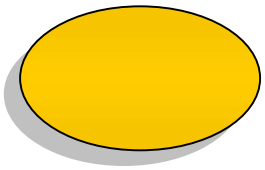
1. The Website should follow the NJIT web designing standards, so its appearance should be uniform to other NJIT websites
2. The system should operate on the following platforms and operating systems
    - Microsoft Windows 2000
    - Microsoft Windows NT
    - JAVA enabled web browser, ex. IE 5.0+
3. The system should have the following minimum hardware requirements
    - RAM Memory: 1 GB (recommended) 592 (minimum)
    - Disk Space: 20 GB disk space (recommended) 100 GB (recommended)
    - Dual or Quad Pentium III or IV Processors
4. Supports multiple communication protocols including HTTP, FTP, OFTP, SMTP, ARP, RARP and POP
5. Enables local and remote management of system administration
6. Supports ASP.NET
7. Manages data over the Internet and NJIT private Networks

Data Flow Diagrams

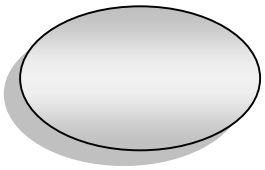
- Glossary of symbol representation:

- DeMarco and Yourdan style of symbols.

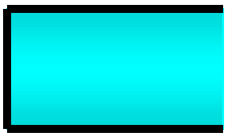
As-Is System Process



To Be System Process

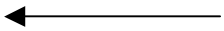


Data Store

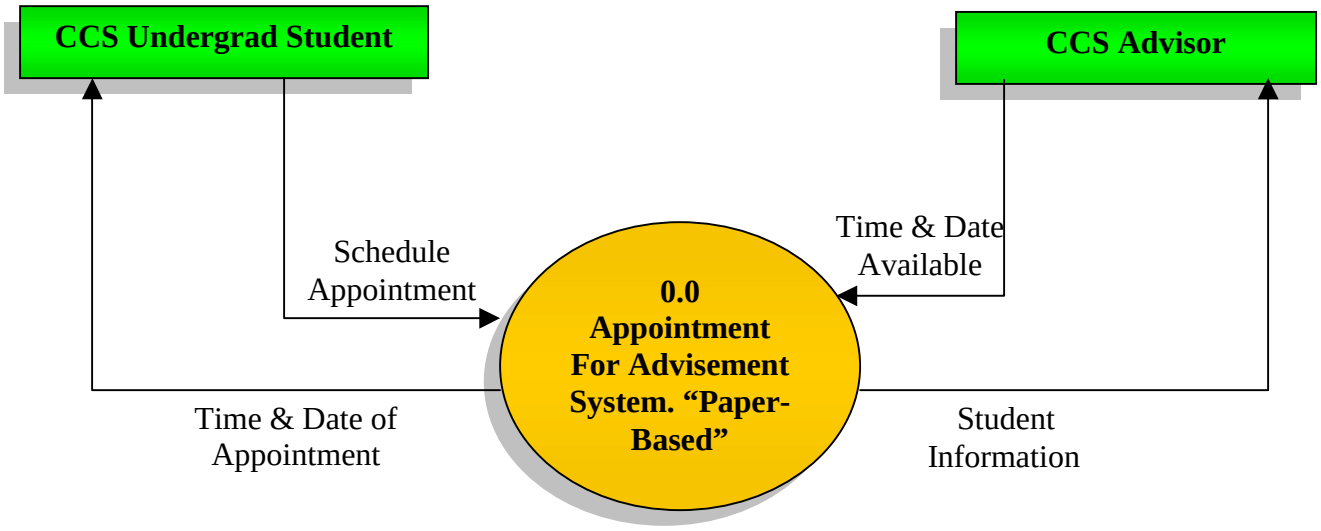


External Entity

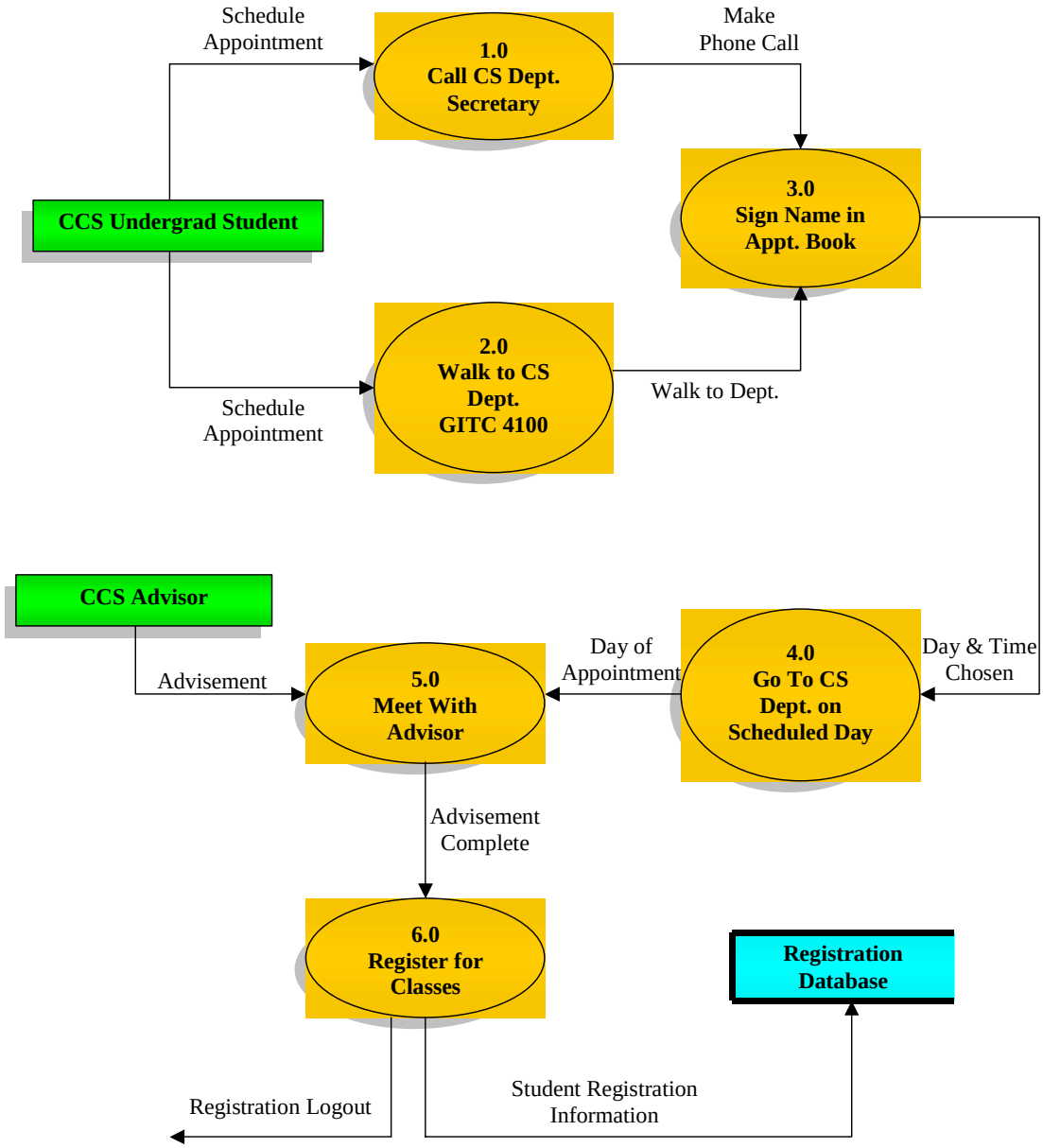
Data Flow



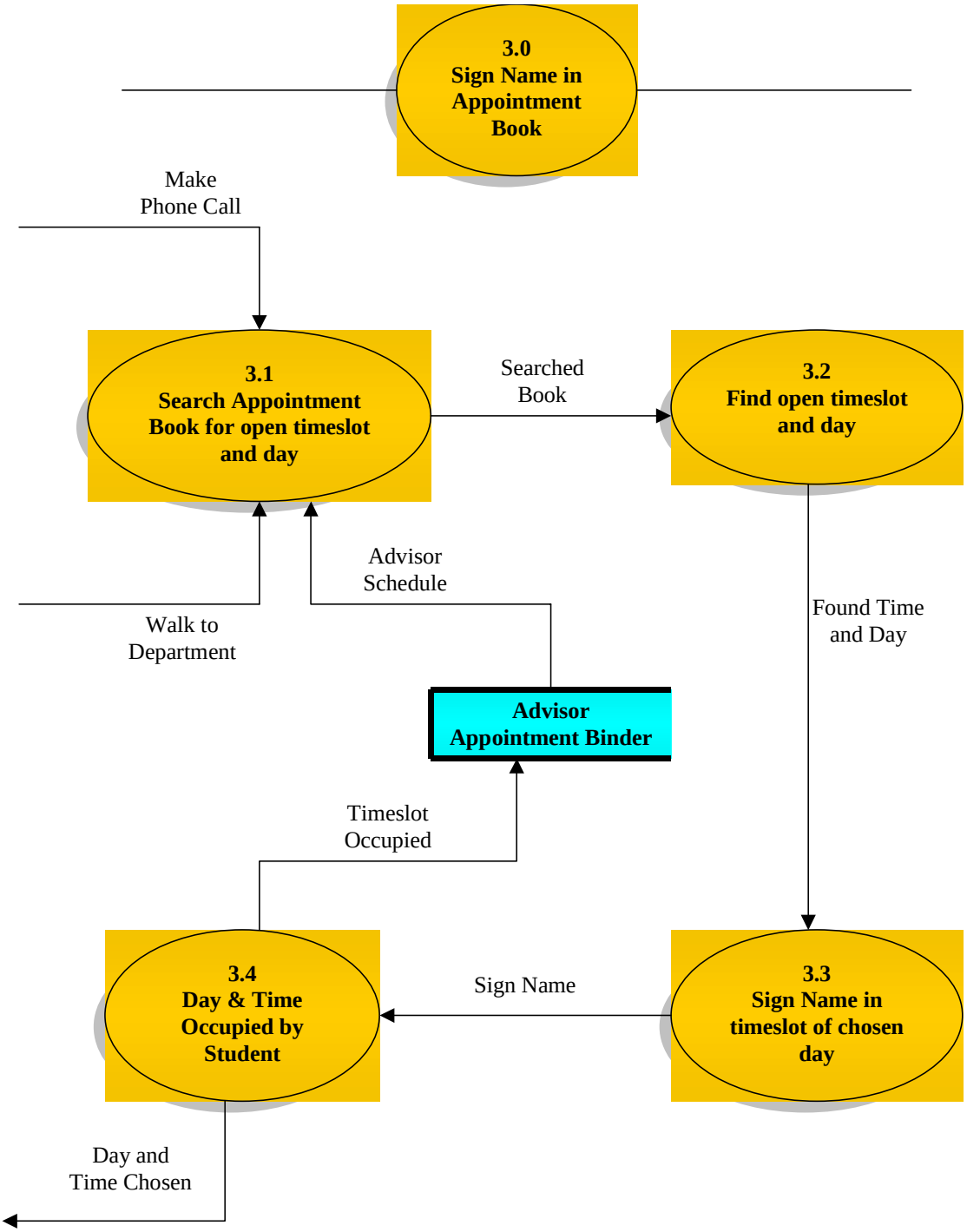
Context Diagram: As-Is System



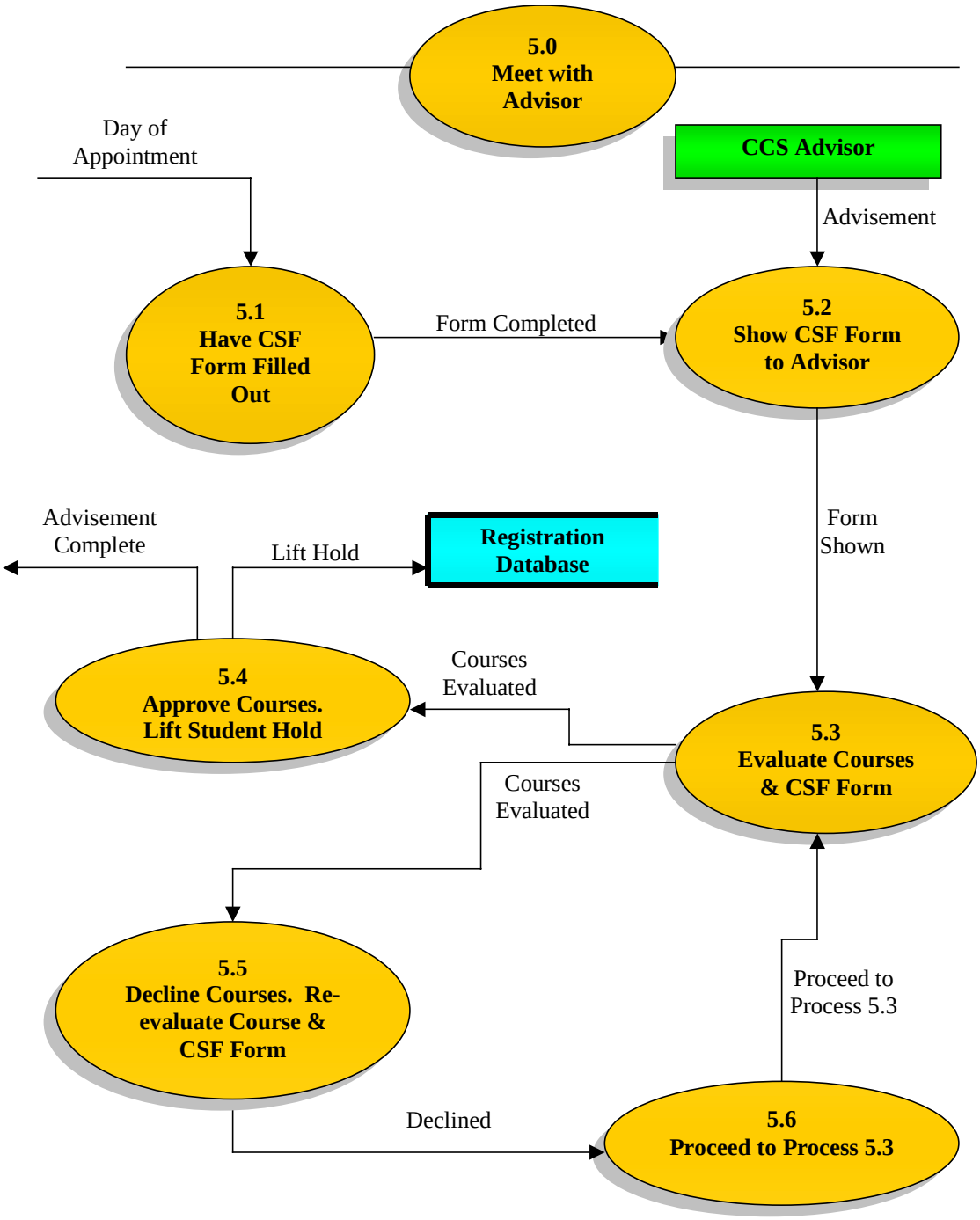
General Diagram: As-Is System



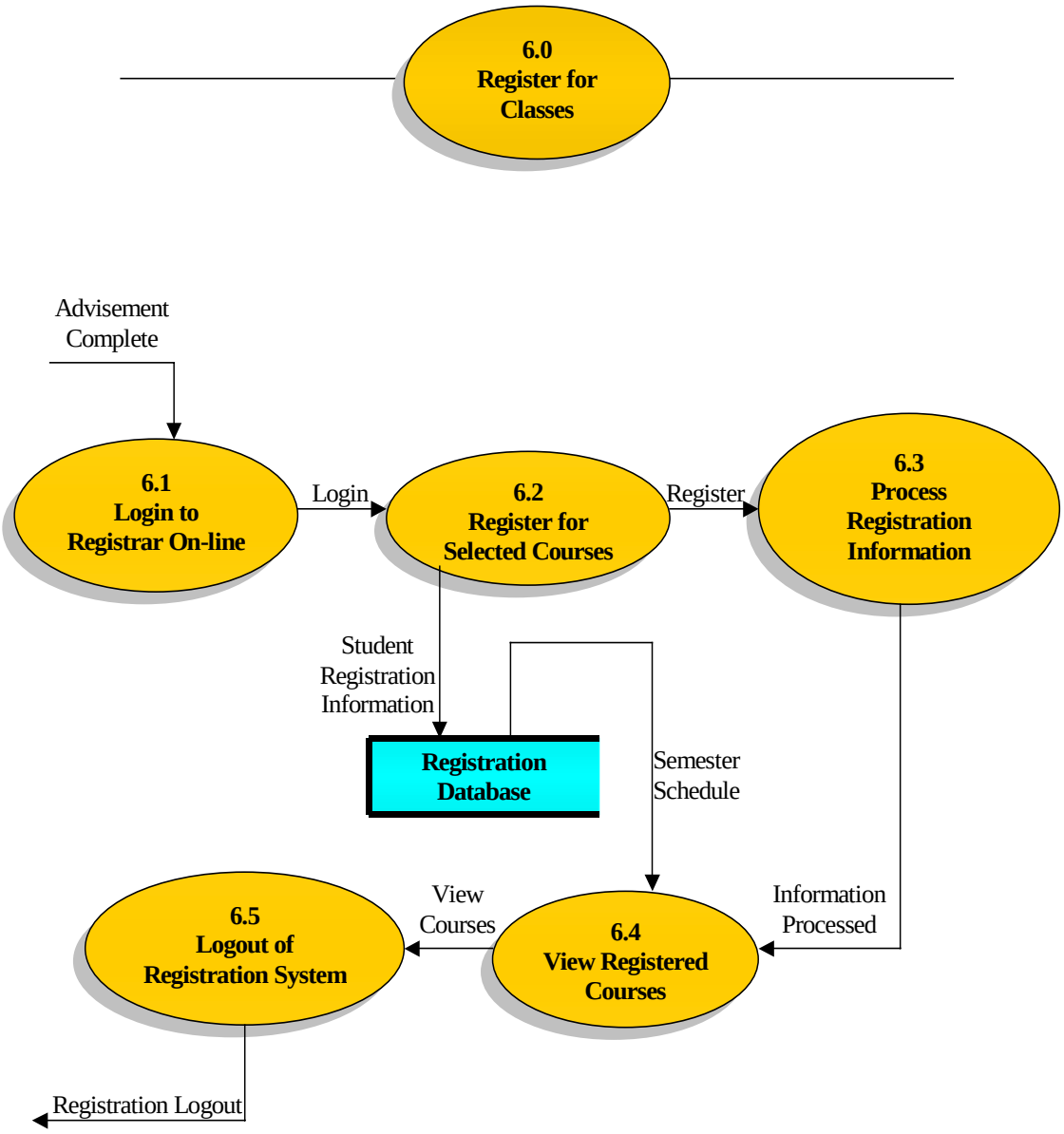
Decomposition of 3.0: As-Is System



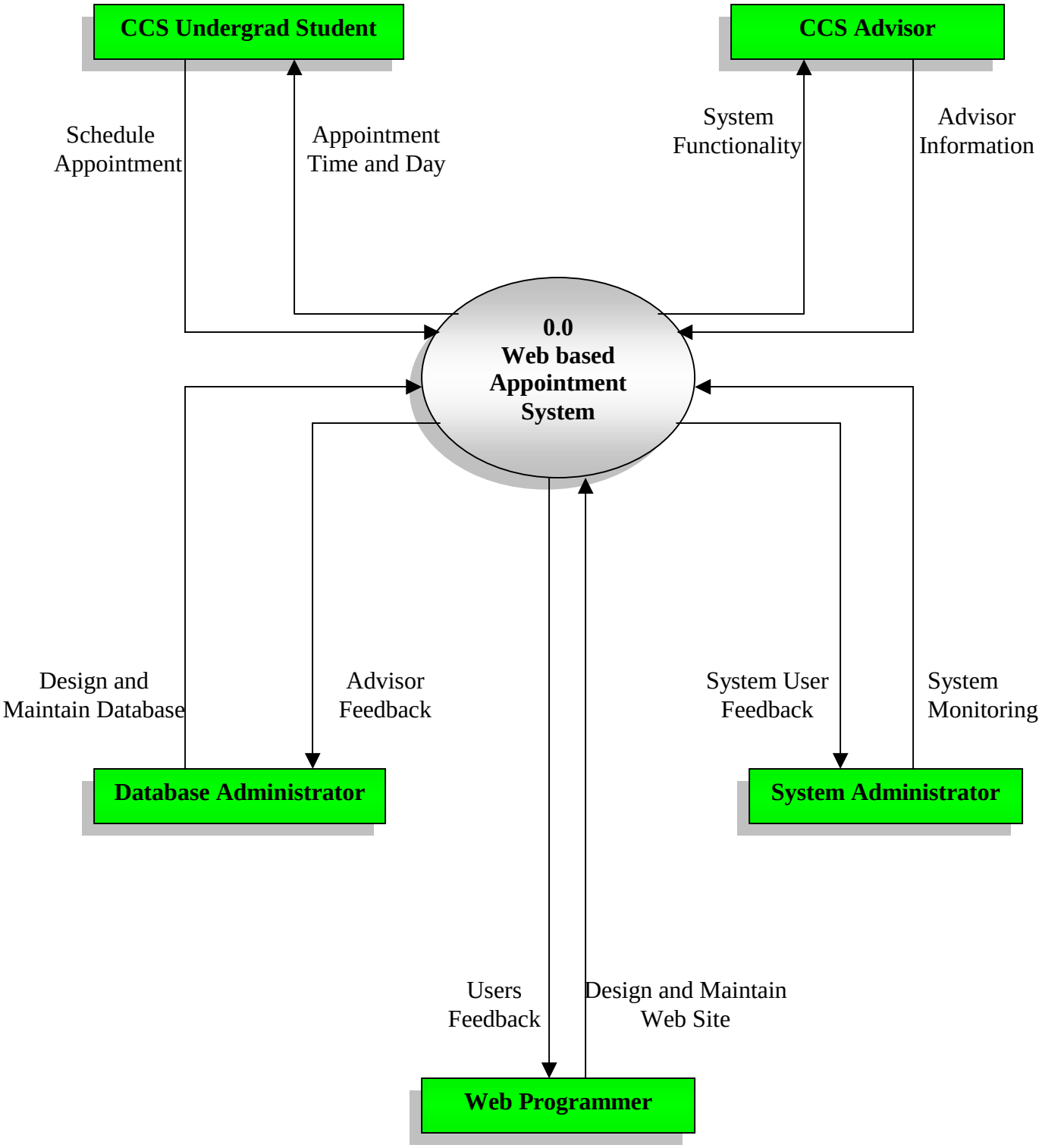
Decomposition of 5.0: As-Is System



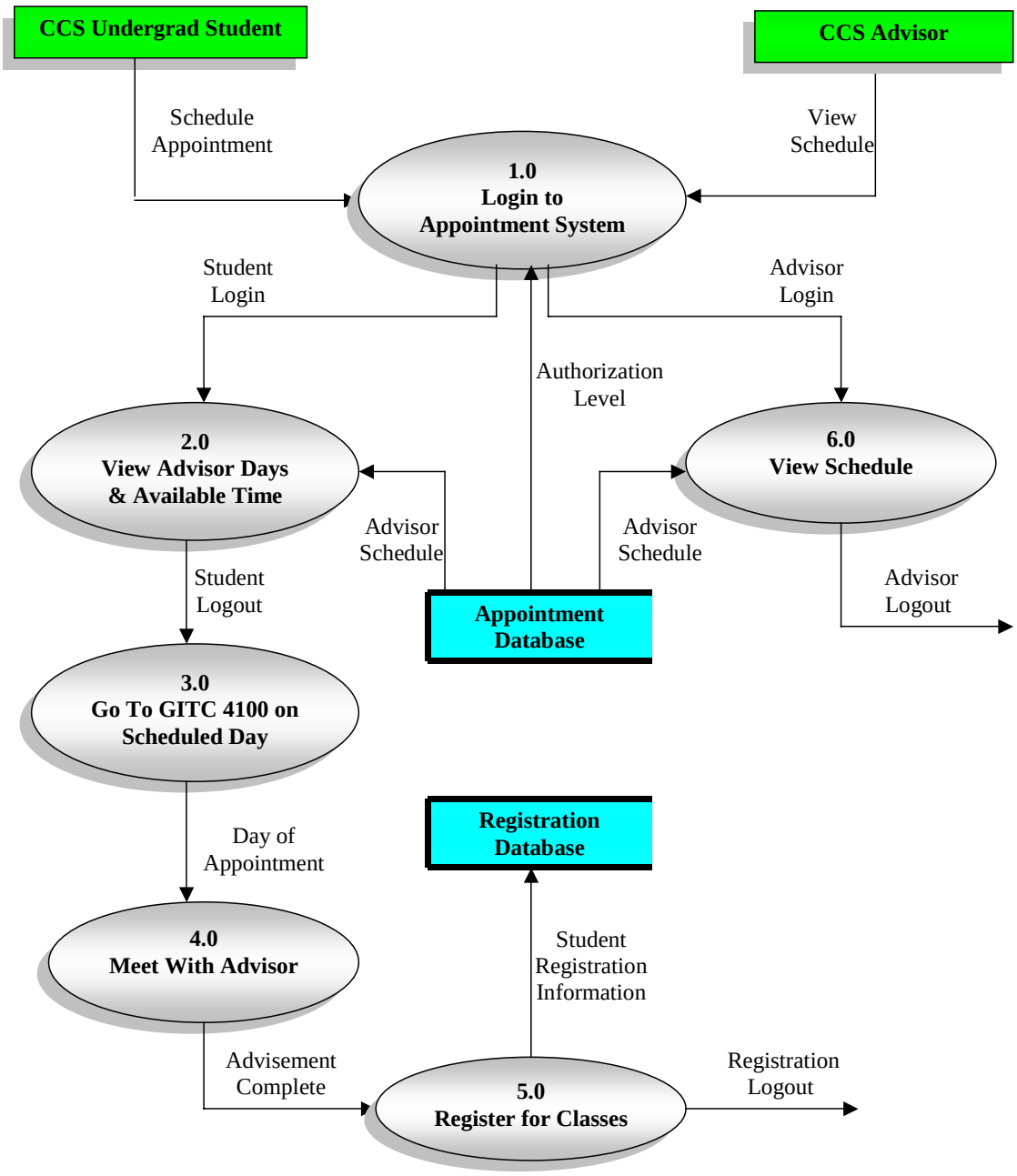
Decomposition of 6.0: As-Is System



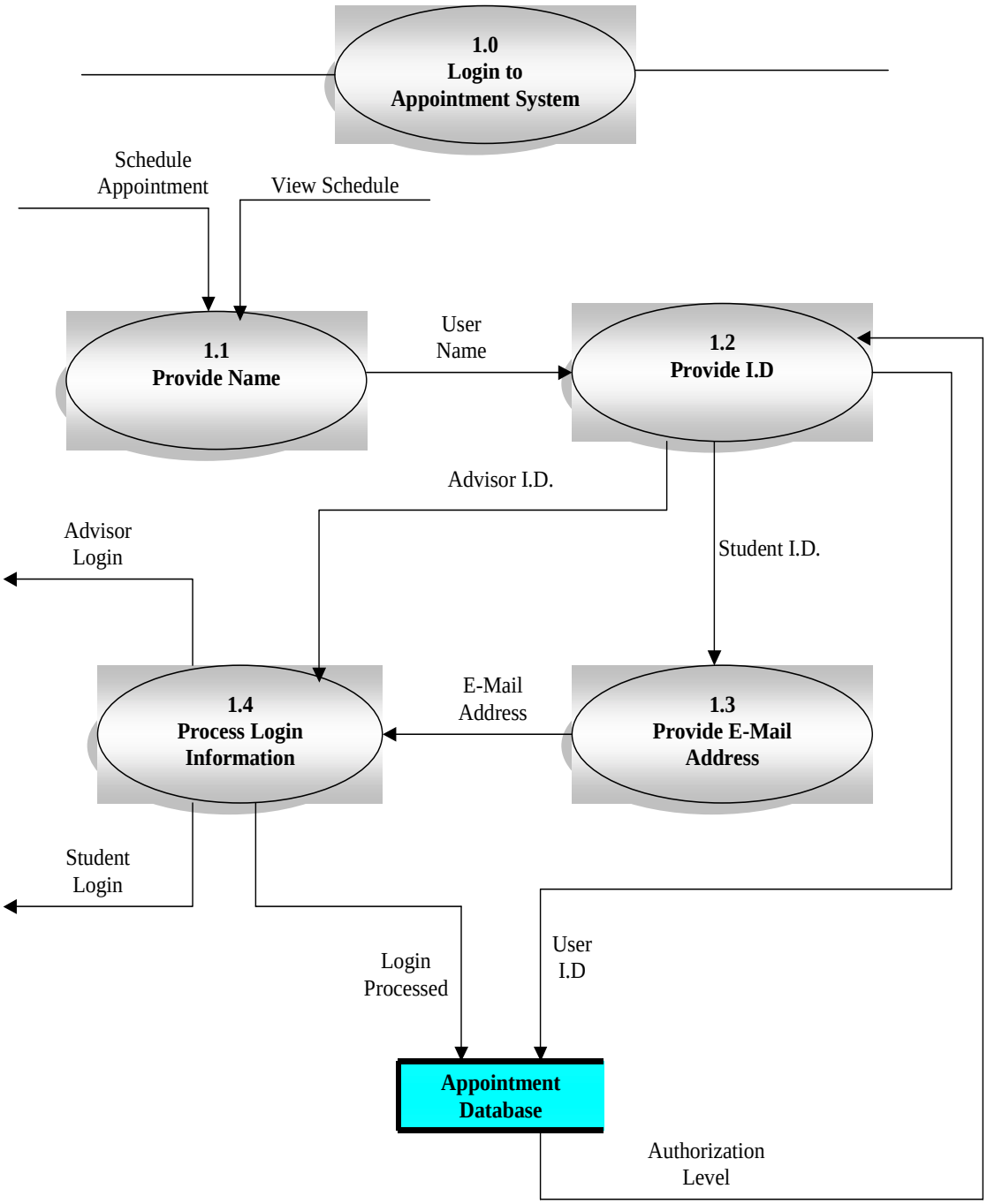
Context Diagram: To-Be System



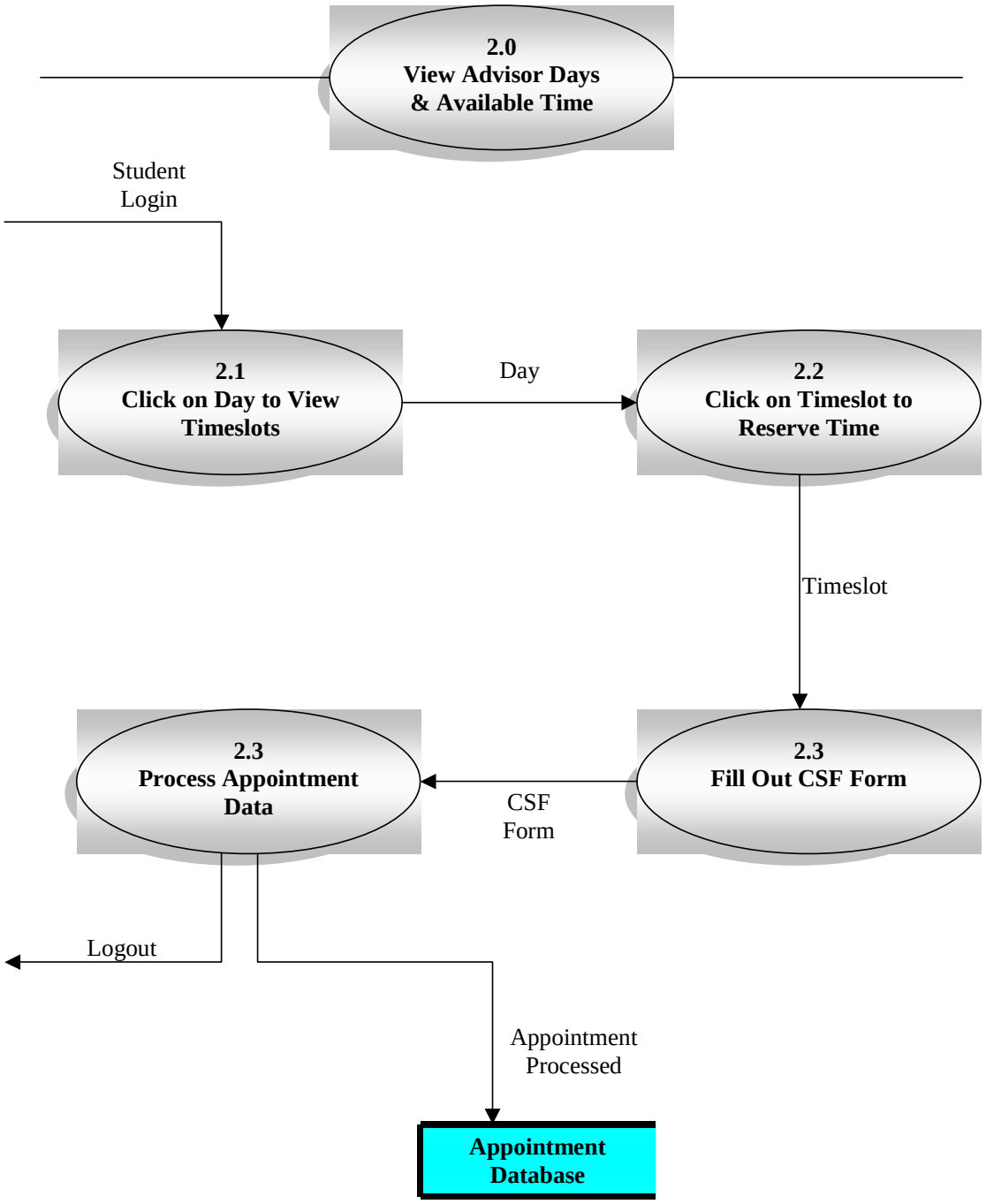
General Diagram: To-Be System



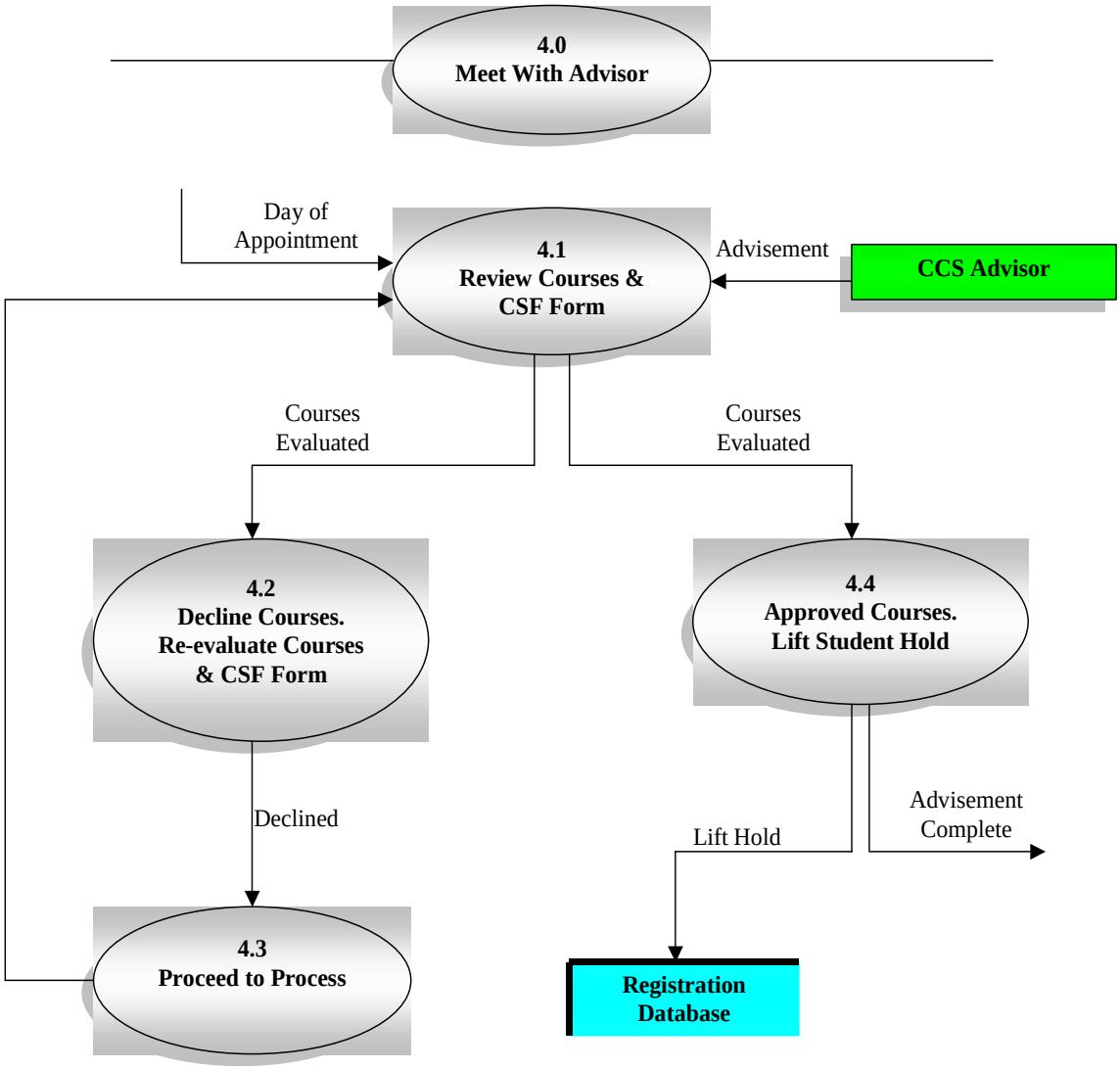
Decomposition of 1.0: To-Be System



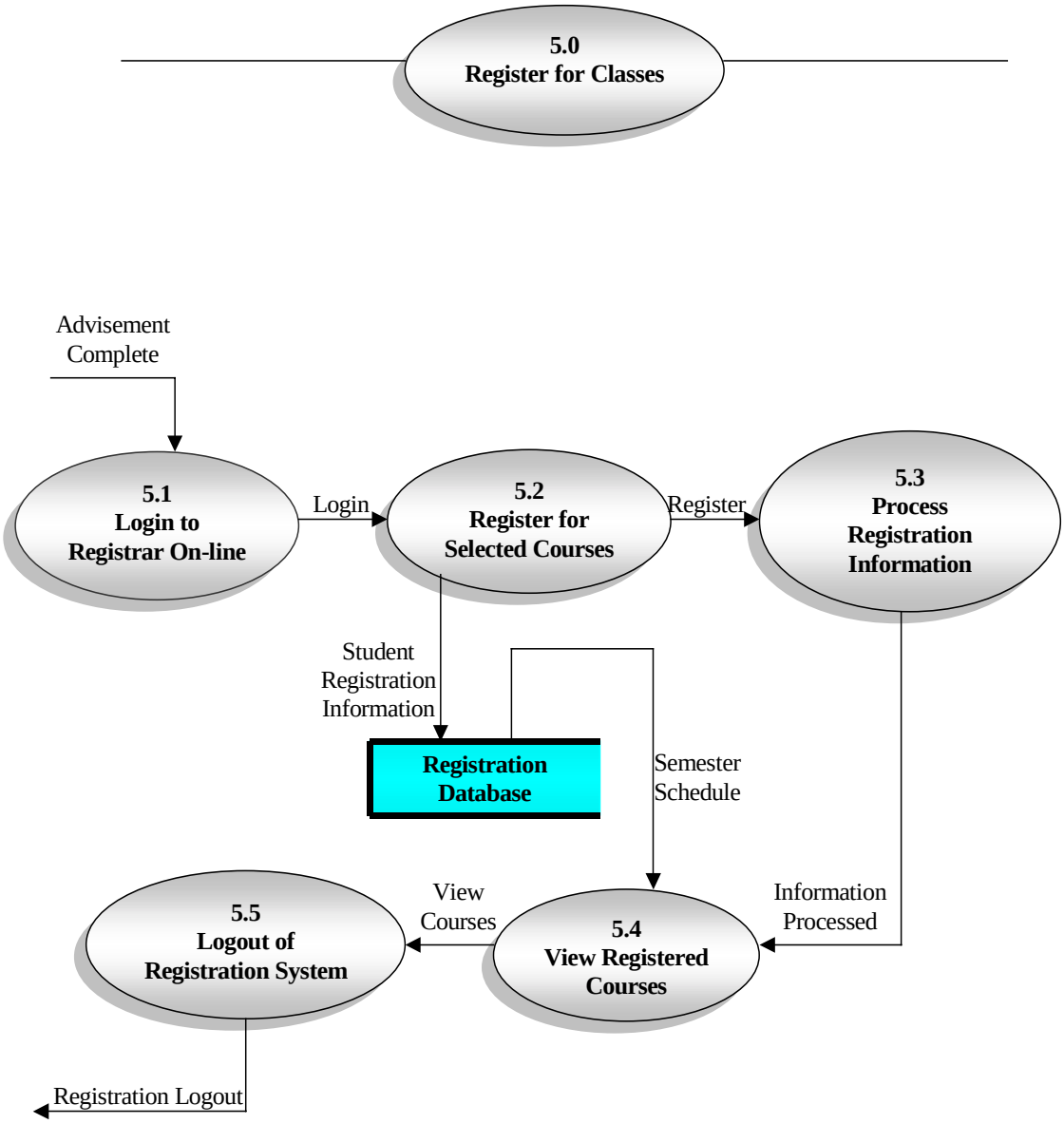
Decomposition of 2.0: To-Be System



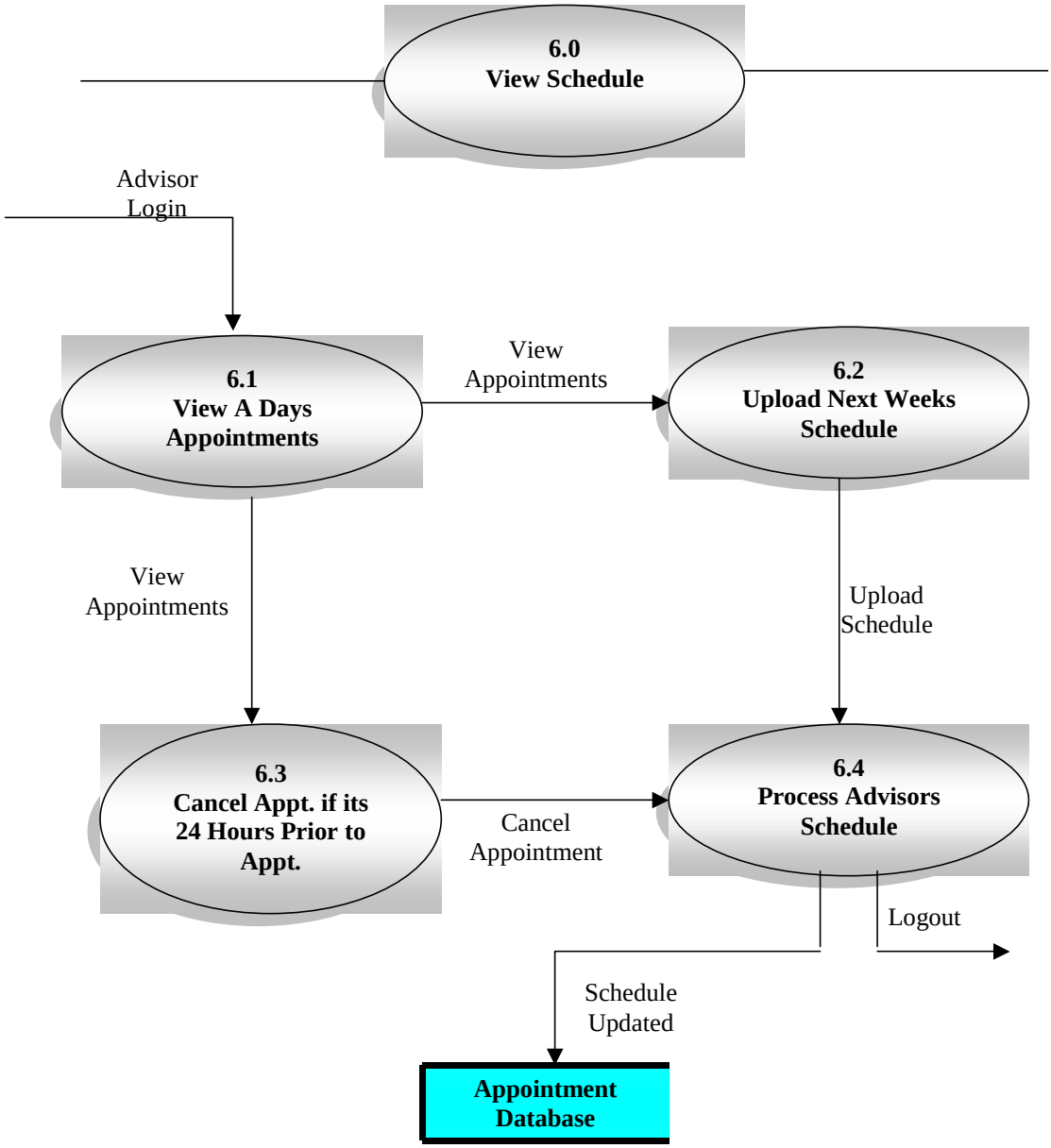
Decomposition of 4.0: To-Be System



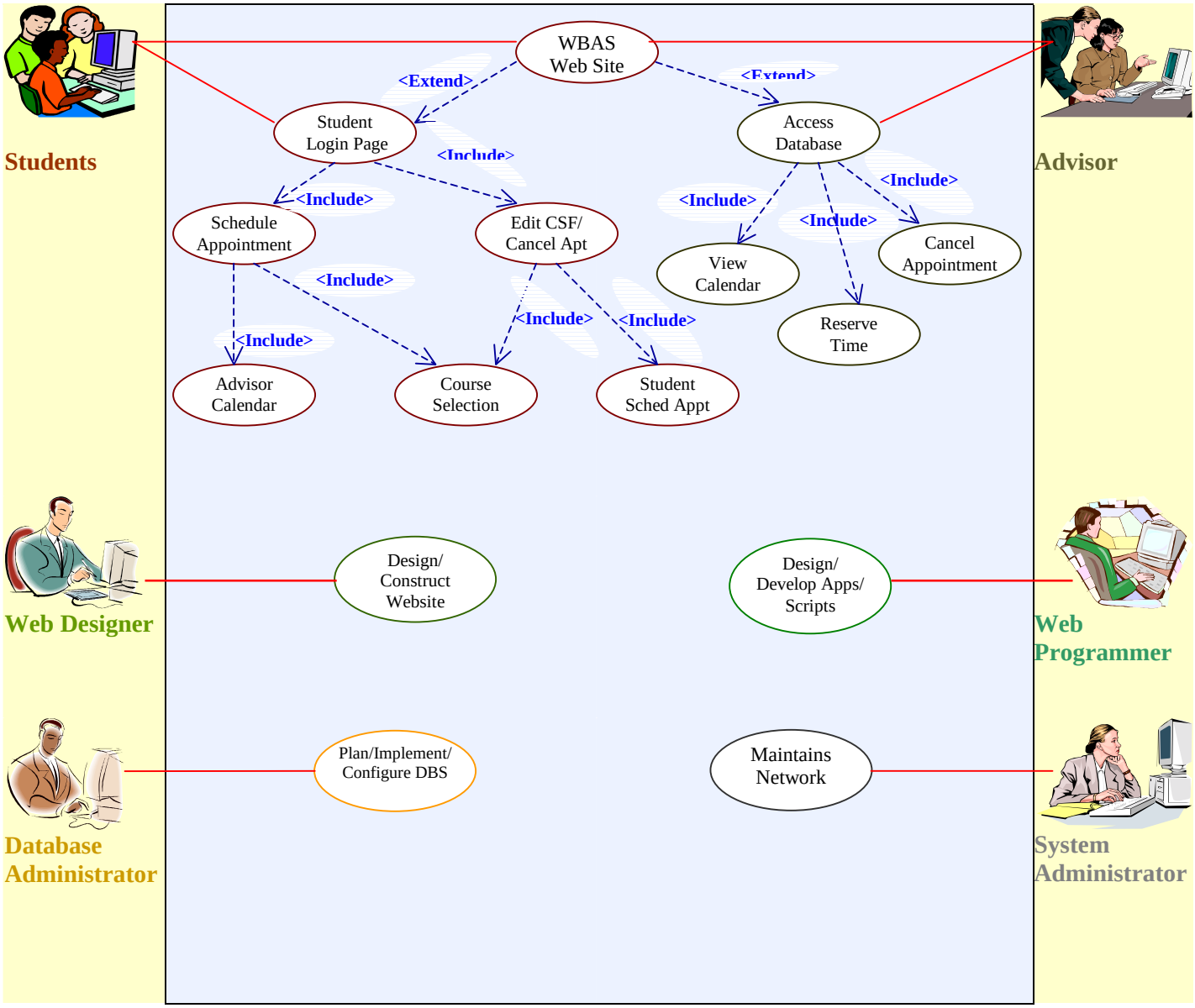
Decomposition of 5.0: To-Be System



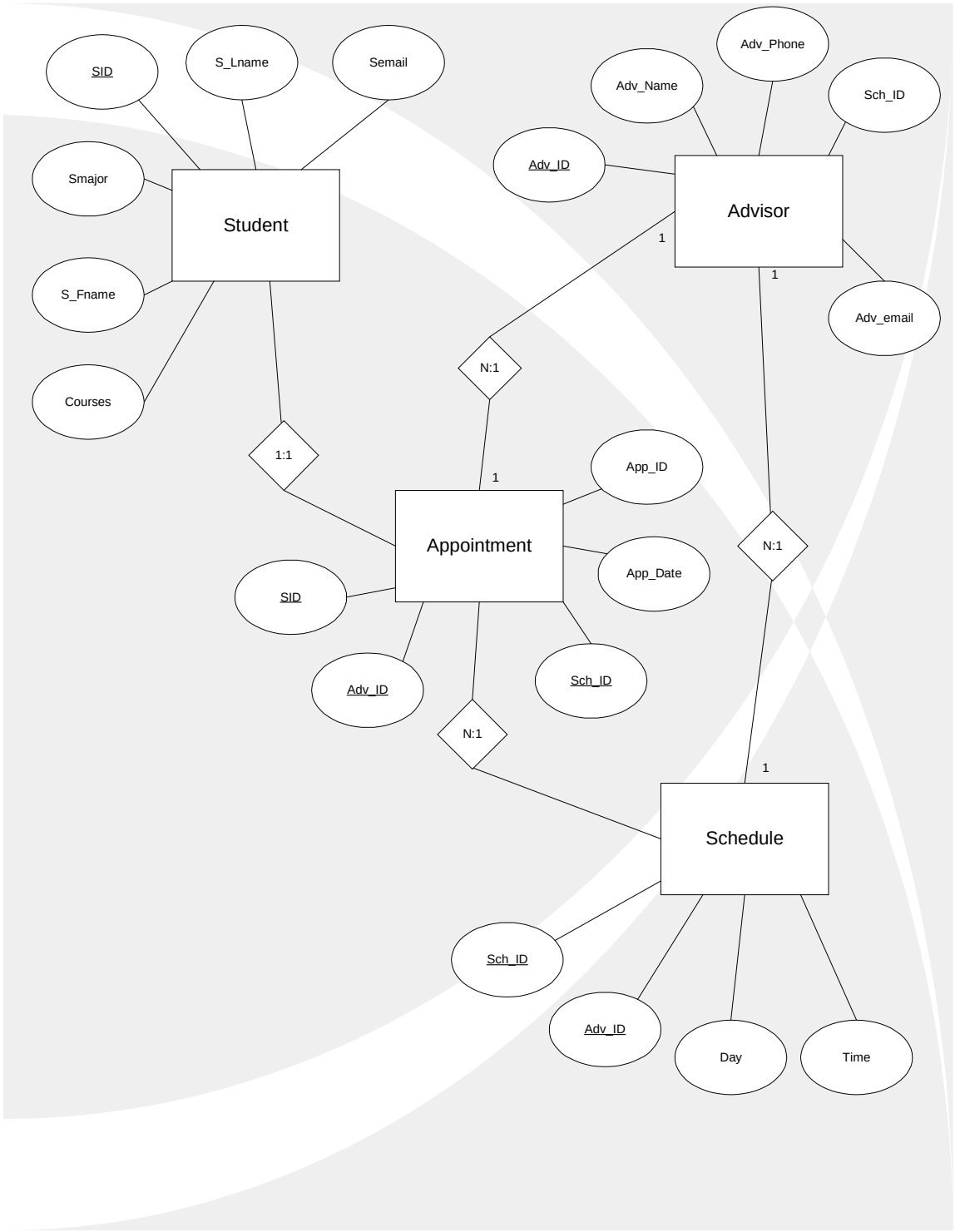
Decomposition of 6.0: To-Be System



3.6.2 USE CASE DIAGRAM



3.6.3 ER-DIAGRAM



3.6.4 DATA DICTIONARY

1.) StudentCSF

SYSTEM: Online Appointment System
NAME: StudentCSF
DESCRIPTION: NJIT students enrolled for a degree in either: CIS, IS, or IT. Also serves as the arranged engagement between the student and the advisor
PURPOSE: Holds vital information about the students and their appointment
RANGE OF VALUE: none
FORMAL STRUCTURE: SID + Studentfirst + Studentlast + Studentmail + SelAdvisor + AptReserved + TimesSlotID + TimeSlotReserved + StudentMajor + ITConcentration + Semester + Courses + Credits

SID → {number}+9
StudentFirst → {char}+15
StudentLast → {char}+25
Studentemail → {char}+20
SelAdvisor → {char} +20
AptReserved → {char} +20
TimesSlotID → {char} +10
StudentMajor → {char} +10
ItConcentration → {char} +10
Semester → {char} +10
Courses → {char} + 5
Credits → {char} +3

2.) AdvisorInfoTable

SYSTEM: Online appointment system
NAME: AdvisorInfoTable
DESCRIPTION: The information for the undergrad CS advisors
PURPOSE: They input the schedules into the system
RANGE OF VALUE: none
FORMAL STRUCTURE: SerialNUM + AdvisorLogID + AdvisorFirst + AdvisorLast + Office + Phone

SerialNUM → {Long Interger}
AdvisorLogID → {char}+15
AdvisorFirst → {char}+50
AdvisorLast → {char}+50
Office → {char} +20
Phone → {char} +11

3.) AdvisorLogin

SYSTEM: Online appointment system
NAME: AdvisorLogin
DESCRIPTION: The table for advisor usernames and passwords.
PURPOSE: Holds security information such as the password and username for a advisor.
RANGE OF VALUE: none
FORMAL STRUCTURE: AdvisorLogID + AdvisorPassw

AdvisorLogID → {char} +10
AdvisorPassw → {char} +10

4.) TMooreWeeks

SYSTEM: Online Appointment system
NAME: TmooreWeeks
DESCRIPTION: This are the available time slots posted by the Thomas Moore for the students.
PURPOSE: To present the student with available times to register for
RANGE OF VALUE: Different
FORMAL STRUCTURE: SerialNum + Tmoore + TimeSlot1 + TimeAv1 + TimeSlot2 + TimeAv2 + TimeSlot3 + TimeAv3 + TimeSlot4 + TimeAv4 + TimeSlot5 + timeAv5 + TimeSlot6 + TimeAv6 + TimeSlot7 + TimeAv7 + TimeSlot9 + TimeAv9 + TimeSlot10 + TimeAv10

SerialNUM → {Long Interger}
Tmoore → {char} +50
TimeSlot1 → {char} +10
TimeAv1 → {char} +10
TimeSlot2 → {char} +10
TimeAv2 → {char} +10
TimeSlot3 → {char} +10
TimeAv3 → {char} +10
TimeSlot4 → {char} +10
TimeAv4 → {char} +10
TimeSlot5 → {char} +10
TimeAv5 → {char} +10
TimeSlot6 → {char} +10
TimeAv6 → {char} +10
TimeSlot7 → {char} +10
TimeAv7 → {char} +10
TimeSlot8 → {char} +10
TimeAv8 → {char} +10
TimeSlot9 → {char} +10
TimeAv9 → {char} +10
TimeSlot10 → {char} +10
TimeAv10 → {char} +10

5.) KshermanWeeks

SYSTEM: Online Appointment system
NAME: KShermanWeeks
DESCRIPTION: This are the available time slots posted by the Kathy Sherman for the students.
PURPOSE: To present the student with available times to register for
RANGE OF VALUE: Different
FORMAL STRUCTURE: SerialNum + KSherman + TimeSlot1 + TimeAv1 + TimeSlot2 + TimeAv2 + TimeSlot3 + TimeAv3 + TimeSlot4 + TimeAv4 + TimeSlot5 + timeAv5 + TimeSlot6 + TimeAv6 + TimeSlot7 + TimeAv7 + TimeSlot9 + TimeAv9 + TimeSlot10 + TimeAv10

SerialNUM → {Long Interger}
KSherman → {char} +50
TimeSlot1 → {char} +10
TimeAv1 → {char} +10
TimeSlot2 → {char} +10
TimeAv2 → {char} +10
TimeSlot3 → {char} +10
TimeAv3 → {char} +10
TimeSlot4 → {char} +10
TimeAv4 → {char} +10
TimeSlot5 → {char} +10
TimeAv5 → {char} +10
TimeSlot6 → {char} +10
TimeAv6 → {char} +10
TimeSlot7 → {char} +10
TimeAv7 → {char} +10
TimeSlot8 → {char} +10
TimeAv8 → {char} +10
TimeSlot9 → {char} +10
TimeAv9 → {char} +10
TimeSlot10 → {char} +10
TimeAv10 → {char} +10

3.6.5 PROCESS SPECIFICATIONS

3.6.5.1 STRUCTURED ENGLISH

Process 1.0 – User Login Selection

ENTER Student ID and PIN#
VERIFY Student ID and PIN# with respect to Student Database System
IF
Student ID and PIN# matches records in Student Database System
THEN

Allow Student to register for meeting.
ELSE

Send message: "Access Denied – Invalid Student ID and PIN#

Process 2.0 – Student's appointment Page

ENTER Student ID and PIN#
VERIFY Student ID and PIN# with respect to Student Database System
IF
Student ID and PIN# matches records in Student Database System
THEN

Allow Student to register for meeting.
SELECT
Name of Advisor from Dropdown Menu
: Advisor's appointment with Student page comes up with Calendar
SELECT
Day from the calendar

: Selected Day shows advisor's Scheduled availability of timing for specified Day
SELECT
Time from the Specified Day's Schedule

Click on "SUBMITT" Button at the bottom of the page

PRINT
Print Scheduled appointment Date/Time and conformation message
SEND

Conformation Page with selected Date and Time of the meeting.

ELSE

Send message: "Access Denied – Invalid Student ID and PIN#

Process 3.0 – Information Page for Advisors ONLY.

ENTER Advisor's ID and Password

IF
ID and Password Correct

Allow Access to Advisor Information Page

SELECT
Date form the Calendar
: Appointment Page Shows up with specified Time for individual Student's appointment

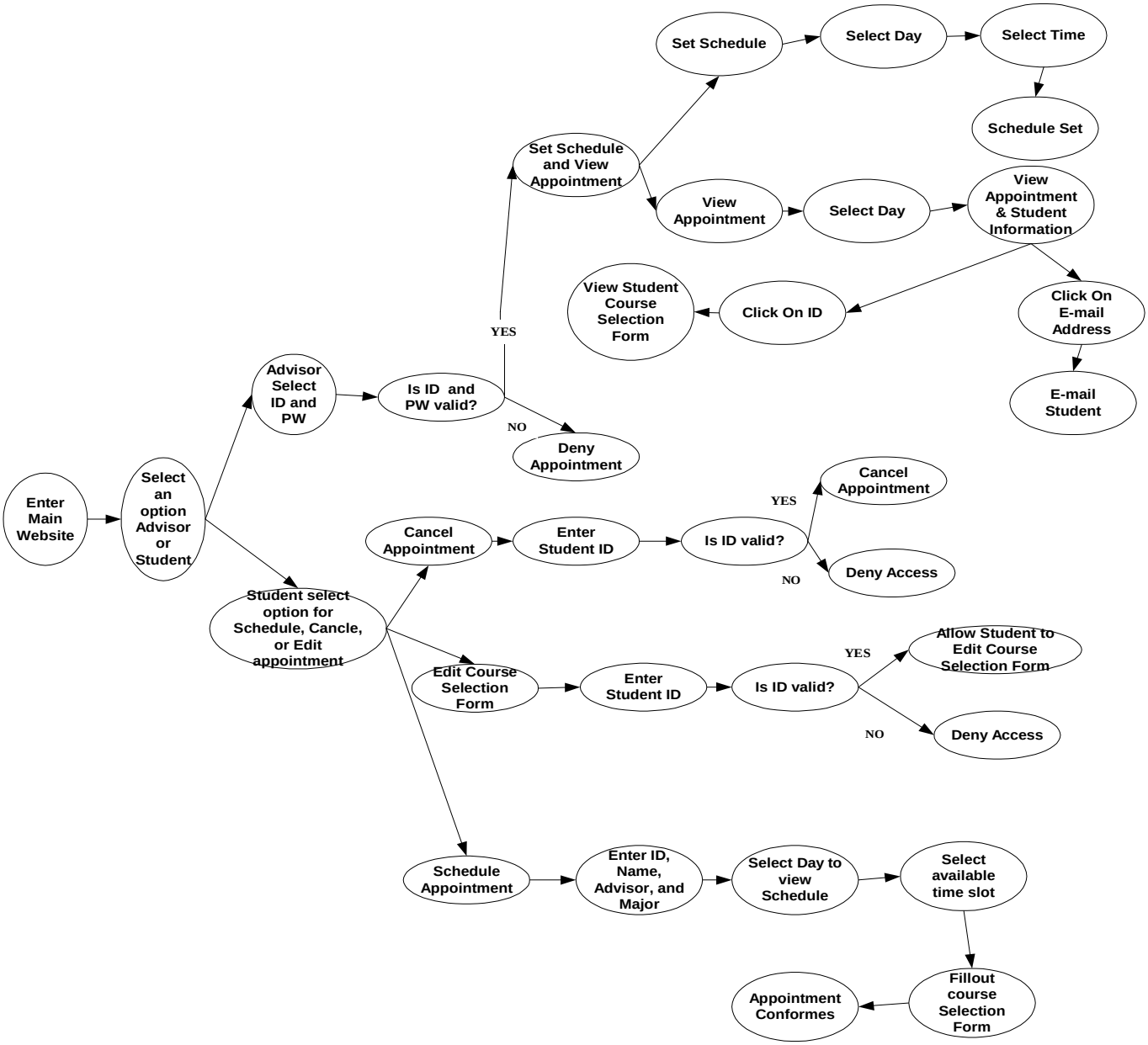
SELECT
Send Mail to Student Option
:Compose e-mail page shows up with
Write any important message regarding appointment

Click on "SEND MAIL" Button

SEND
Conformation Message to advisor

ELSE
Send Error Message: ID or Password Incorrect.

3.6.5.2 DECISION TREES



3.6.5.3 DECISION TABLE

CONDITION	**RULES**							
	1	***2***	***3***	***4***	***5***	***6***	***7***	***8***
Is user an Advisor	Y	Y	Y	N	N	N	N	N
Is user a student?	N	N	N	Y	Y	Y	Y	N
Valid User Login?	Y	Y	N	Y	Y	Y	N	N
ACTION								
Provide Access	*X*	X	-	X	X	X	-	-
Schedule App	-	-	-	X	-	-	-	-
Cancel App	-	X	-	-	X	-	-	-
Edit CSS	-	-	-	-	X	X	-	-
View timeslots	-	-	-	X	-	-	-	-
Send Automated email	-	-	-	X	X	-	-	-
View Scheduled appointments	X	X	-	-	-	-	-	-
Fill in Schedule	X	-	-	-	-	-	-	-
Reserve time	X	-	-	-	-	-	-	-
View Student Info	X	X	-	X	X	X	-	-
Email Student	X	X	-	-	-	-	-	-

3.7 ANALYTICAL ANALYSIS (SPSS)

(To view detailed SPSS Test Results, see attached document at the end)

Research Design Details:

The dependent variable that we have chosen in this project is the future "Web-Based Appointment System". Since this is a project geared for students in the future, it would not be recommended to analyze it based upon unexploited factors. Therefore, we decided to undertake these statistical studies by comparing the Web-based appointment system with the current "Paper-Based Appointment System" that many students are familiar with. In this case, the paper-based appointment system acted as our independent variable. We also used students' "Major" and "Status" to act as moderating variables that can influence our study.

Once we established this relationship, we designed a questionnaire to investigate our variables as well as analyze student's results. After completion of a total of 20 questions, the questionnaire was handed out to a select group and the data received was then entered in the SPSS System. Once all the data was entered into the SPSS system, we ran a number of specific statistical tests to attain a clear understanding of our variables. These tests included Cronbach's Alpha Test, Pearson Correlation Matrix Test, as well as Frequencies and Descriptive Statistics Tests. The results we achieved from the SPSS system explained how the independent and dependent variables varied. The moderating variables were used to better define the population.

Methodology Section

Study Population:

Our study population was drawn from NJIT, since NJIT students would be accessing the Web-based Appointment System in the future. Attaining results or surveys from any other population or group would be futile.

Study Sample:

The survey of students at NJIT was taken via systematic sampling. However, our population was limited to a certain group because only IT, IS, and CS undergrad students were allowed to answer the questionnaire. The Web-based appointment system is only intended for these specified Undergraduate majors. Since the population was heterogeneous, demographics such as Major and Status were used. Although all students were familiar with the paper based system, we expected the higher status students to have more of an understanding of the paper-based system as well as the need for a future web-based appointment system, since they have endured enough time making appointments with their advisors through the paper-based system.

Fifty surveys were distributed, from which 40 Undergraduate IT, IS, and CS students had participated. From the 40 students that had participated, 2 were not familiar with the paper-based system, and therefore did not qualify to complete the survey. Therefore, a total number of 40 NJIT students were surveyed.

Data Collection Methods:

The questionnaire was designed to analyze and explore the relationship and comparison between the current independent variable (Paper-based System) and the future dependent variable (Web-based System). The collected data was entered into the SPSS system after the questionnaire was passed out to the above stated sample population.

Sample Characteristics & Descriptive Statistics

After conducting a descriptive and statistical study of the SPSS system, it was established that 57.5% of the NJIT students surveyed were CS majors. 30% of the students surveyed were IS majors, and only 12.5% of the students surveyed were IT majors.

The statistical study further showed that 65% of the NJIT students surveyed had a status of seniors. Only 7.5% of the students surveyed were freshmen. The reason for such a difference could be that the seniors are better informed of the current paper-based system.

Under the descriptive statistics, the minimum, maximum, mean, and standard deviation was also computed for the dependent variable, Web-based Appointment System, and the independent variable, Paper-based System. The mean for the independent variable was 3.7288, which suggests that the students are not satisfied with the current paper-based system. The majority of students have strongly disagreed that the paper-based system is reliable, efficient, and organized. The mean for the dependent variable was 2.2794, which suggests that the students were strongly in favor of a future Web-based appointment system.

Statistical Methods Used

1.) Cronbach's Alpha Test for Reliability

2.) Pearson Correlation Matrix Test to determine the variance between different variables

Reliability Analysis

This analysis test is crucial to a statistical study because it tests whether the respondents answered the questions in a questionnaire consistently and more importantly, reliably. A strong and well-phrased questionnaire would result in a consistent response from the respondents. We achieved an Alpha value of .7535 as our reliability scale, which means that our respondents were 75.35% reliable, and responded very well and consistently.

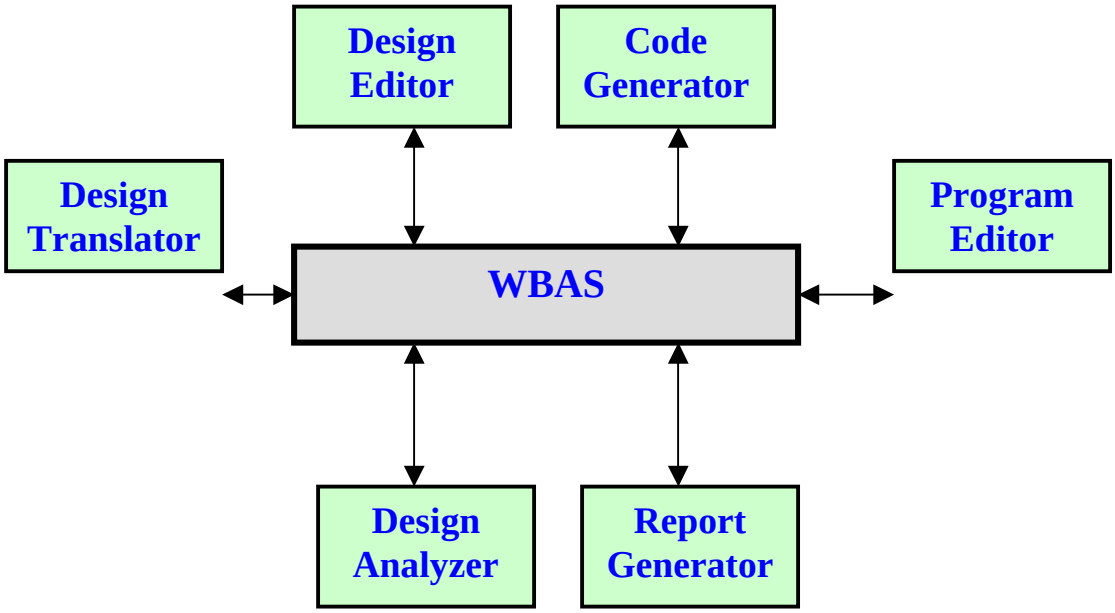
Pearson Correlation Matrix

The Pearson Correlation Matrix reveals that there is a negative correlation between the Web-based Appointment System and the Paper-based system. This suggests that as the Paper-based system comes into use and goes up, the web-based system goes down, and vice versa. The reason this might happen is because the systems are quite the opposites and do not exist with each other. The paper-based system is currently used, and the Web-based system is being made for the future to ultimately replace the paper-based system.

4. SYSTEMS DESIGN

4.1 SYSTEM STRUCTURING

4.1.1 THE REPOSITORY MODEL



The repository model shares information through sub-systems that communicates directly with local databases.

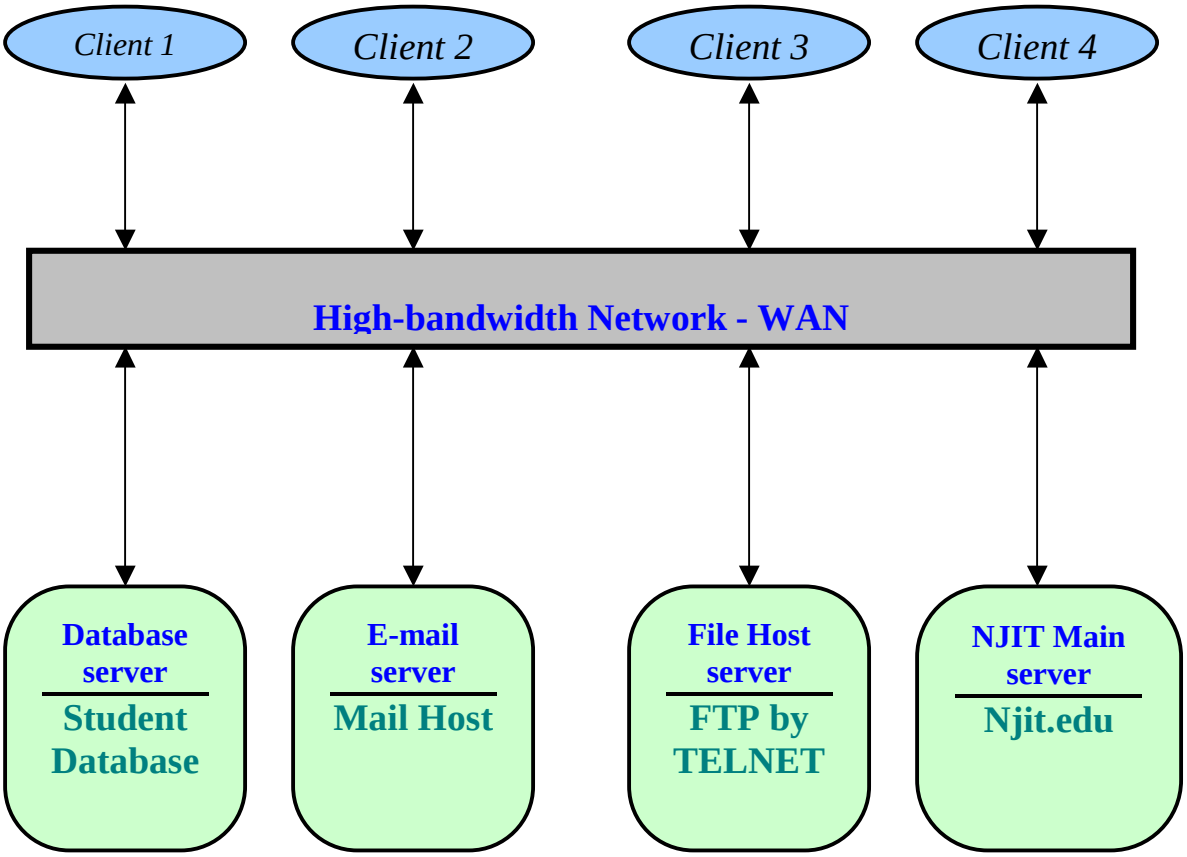
Advantage

- This type of modeling technique will efficiently share large amount of data that cannot be comprehended through other models.

Disadvantages

- Sub-System must share same repository data model
- System evolution is difficult
- Fault-tolerance problem
- Different sub-system may have different requirements

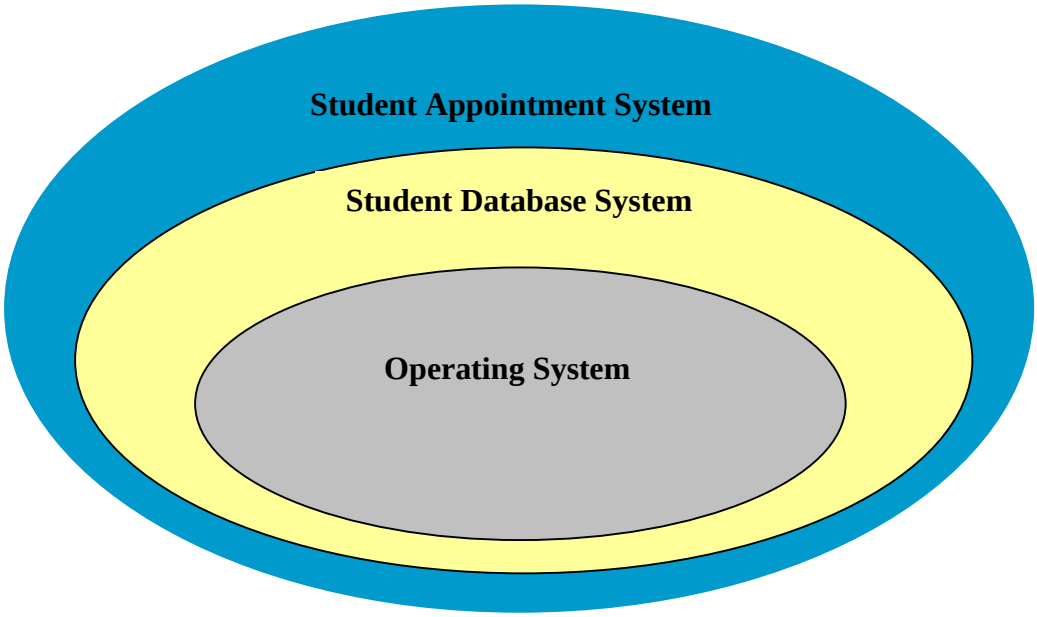
4.1.2 THE CLIENT-SERVER MODEL



Functionalities

- A set of stand-alone "servers" which offers services to other sub-system
- A set of "clients" that call on the services offered by servers.
- A "network" that allows the clients to communicate with servers and access their services.

4.1.3 THE ABSTRACT MACHINE MODEL (A LAYERED MODEL)



The abstract machine model organizes a system into a series of layers each of which provides a set of services. Each layer defines an abstract machine. The abstract machine model supports incremental development of systems

Advantage

- Changeable and portable

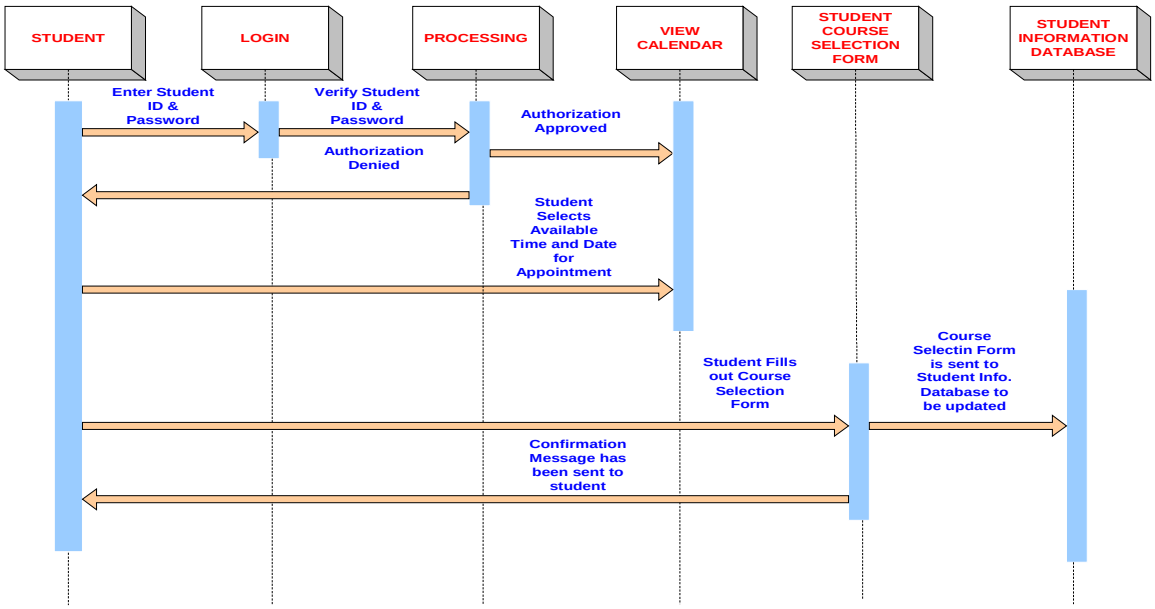
Disadvantages

- Structuring system is difficult
- Performance may be a problem

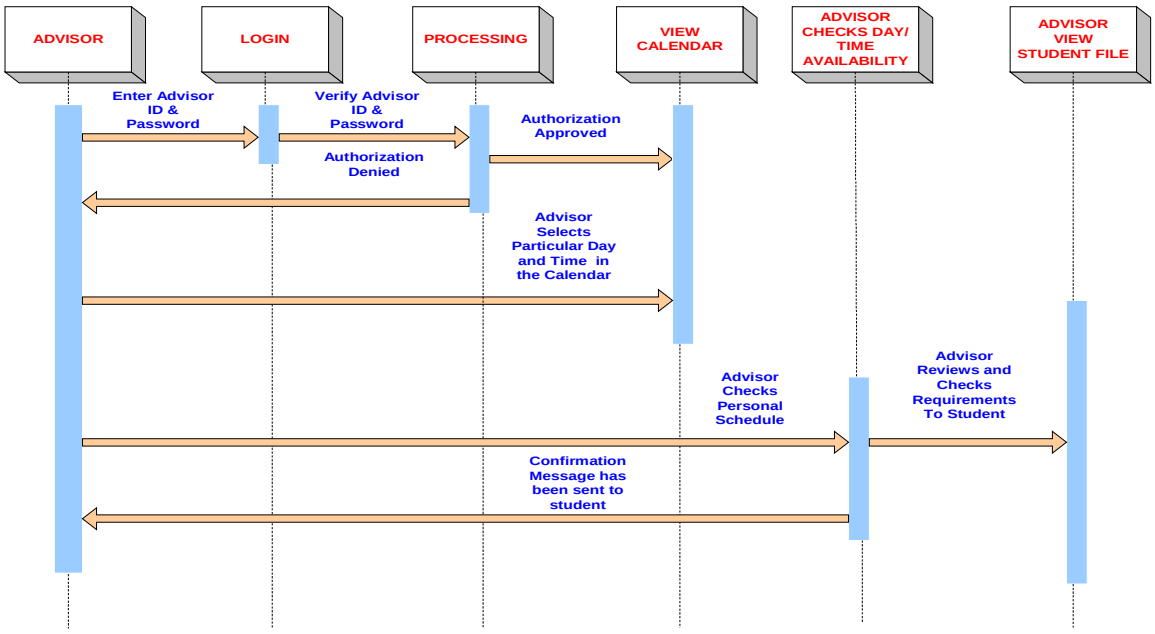
4.2 CONTOL MODELS: EVENT-DRIVEN DESIGN

4.2.1 SEQUENCE DIAGRAM

Students' Sequence Diagram



Advisors' Sequence Diagram



Purpose

The sequence diagram (SD) specifies the time and control aspects of a system. Typically, you use the SD to analyze only the more complex business events. Simple business events rarely require an SD.

A collaboration diagram shows the objects and relationships involved in an interaction. It shows the messages exchanged among the objects during the interaction, numbering them to show the sequence of messages. It does not show the timing of the messages.

Object

An object is an instance of a class. The object line represents a portion of the lifetime of an object in the system. Time flows from top to bottom on the symbol. The spacing between event symbols is not important. It does not indicate the length of time between events.

In Scope region or focus of control

The In Scope region is a white rectangle that can be placed on top of an initiator, object, or active object to show that the object is in scope. You can move the In Scope region up and down the axis of the object, but not away from it. When you move the In Scope region, the nearby messages move with it. When you move messages, the In Scope region does not move. This allows you to move messages from one In Scope region to another.

Messages

A message represents an individual stimulus from one object to another. A message can signal an event or transmit data. It occurs at a point in time.

Multiple messages can be sent from the same point on the initiator or object line; that is, sent at the same point in time. An object can send a message to itself. Such a message must arrive at the object at another point in time.

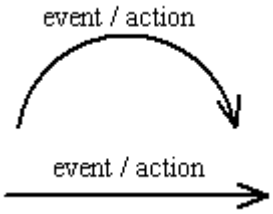
Object lifeline

A Role is a slot for an object within a collaboration that describes the type of object that may play the role and describes its relationships to other Roles. Within a sequence diagram the existence and duration of the object in a role is shown, but the relationships among the roles is not shown. There are Classifier Roles and Association Roles.

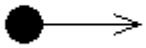
4.2.2 Statechart Diagram

Basic Statechart Diagram Symbols and Notations

States
States represent situations during the life of an object. You can easily illustrate a state in SmartDraw by using a rectangle with rounded corners.



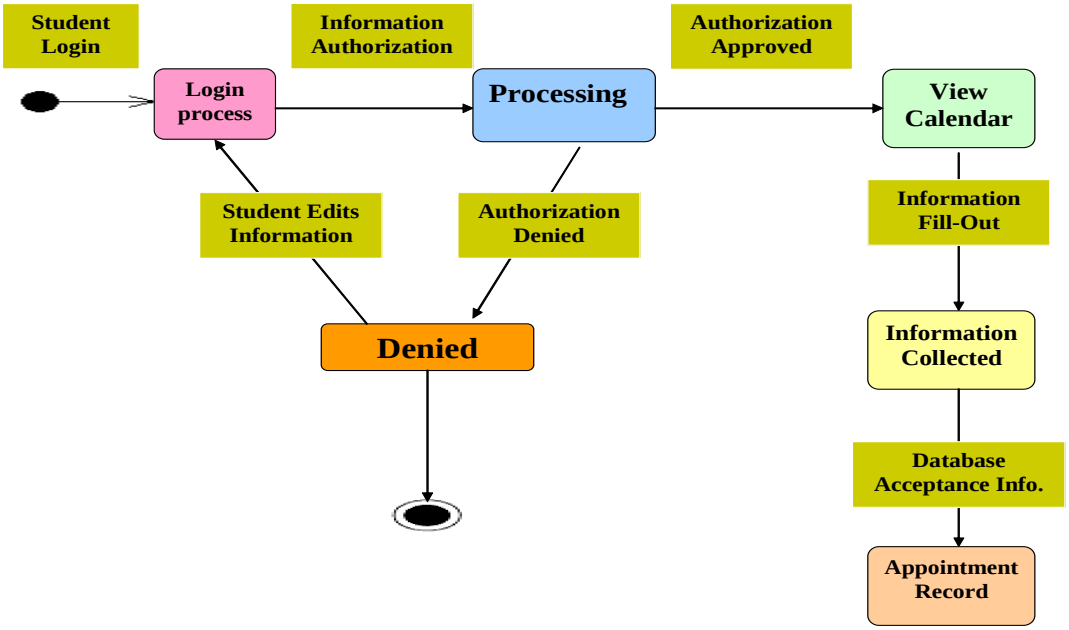
Transition
A solid arrow represents the path between different states of an object. Label the transition with the event that triggered it and the action that results from it.



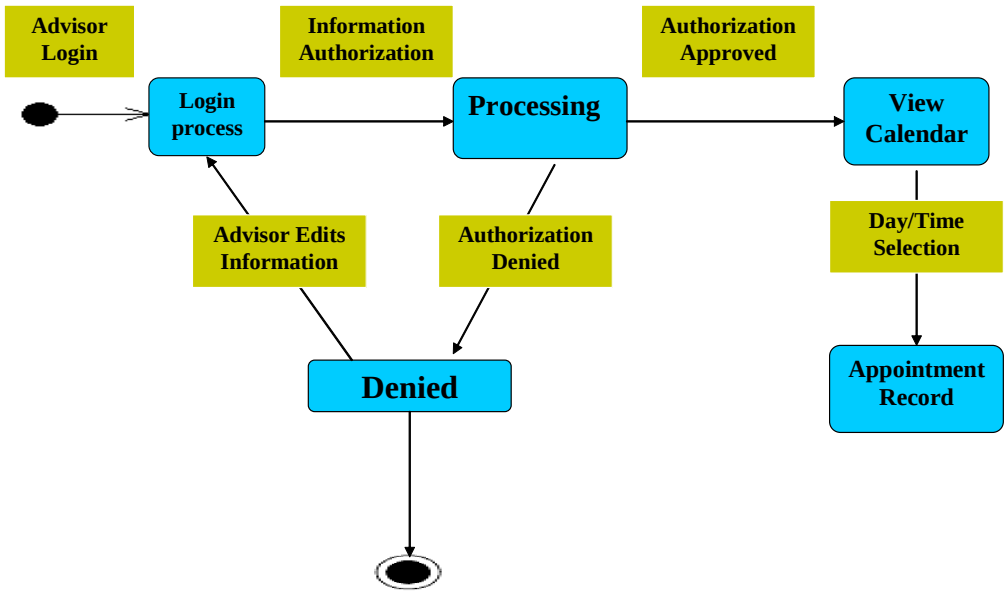
Initial State
A filled circle followed by an arrow represents the object's initial state.

Final State
An arrow pointing to a filled circle nested inside another circle represents the object's final state.

Students' Statechart Diagram



Advisors' State Chart Diagram



Purpose

A statechart diagram (STD) describes the possible states of a single class and the events that cause state transitions. They are useful for showing the life cycle of the class.

Statechart and activity diagrams both describe state transitions and share many of the same elements. An activity diagram is most effective for describing processes that involve more than one object.

Types of STDs

There are two kinds of STDs:

- One-shot life cycles describe objects with finite lives. They have initial and final states:
    - The initial state, in which the object is created, is a solid circle.
    - The final state, in which the object is destroyed, is a bull's eye.
- Continuous loops describe objects that move through a series of states. Where, how, and when the object is created or destroyed is unimportant.

Contents

A state diagram defines:

The external stimuli of the system: the events

The values of objects: the states

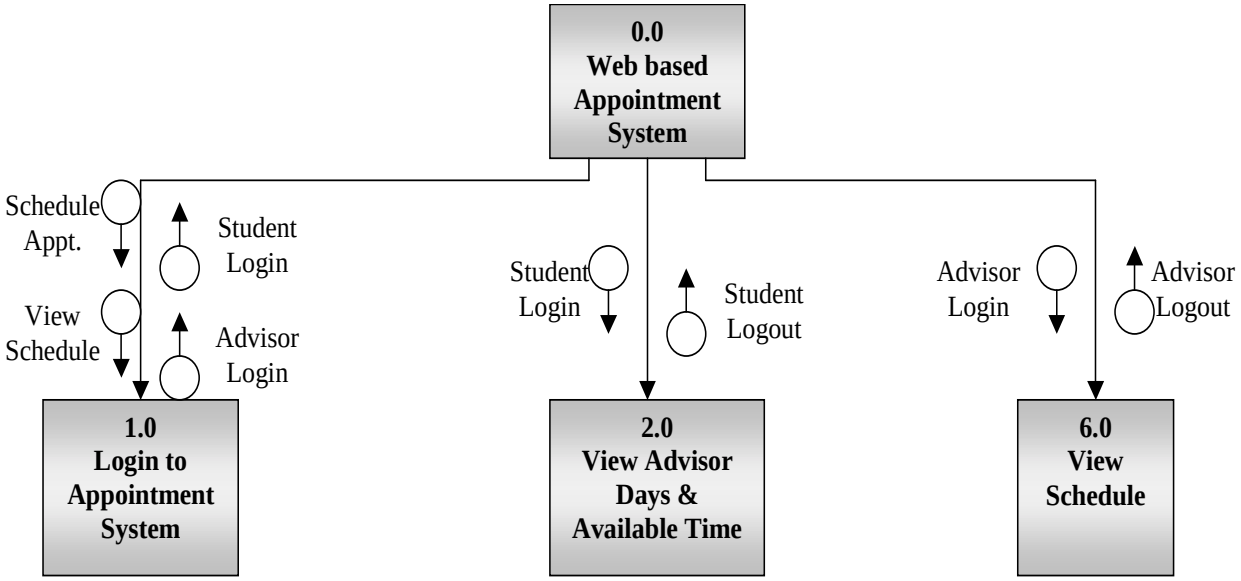
The changes of the object values: the transitions

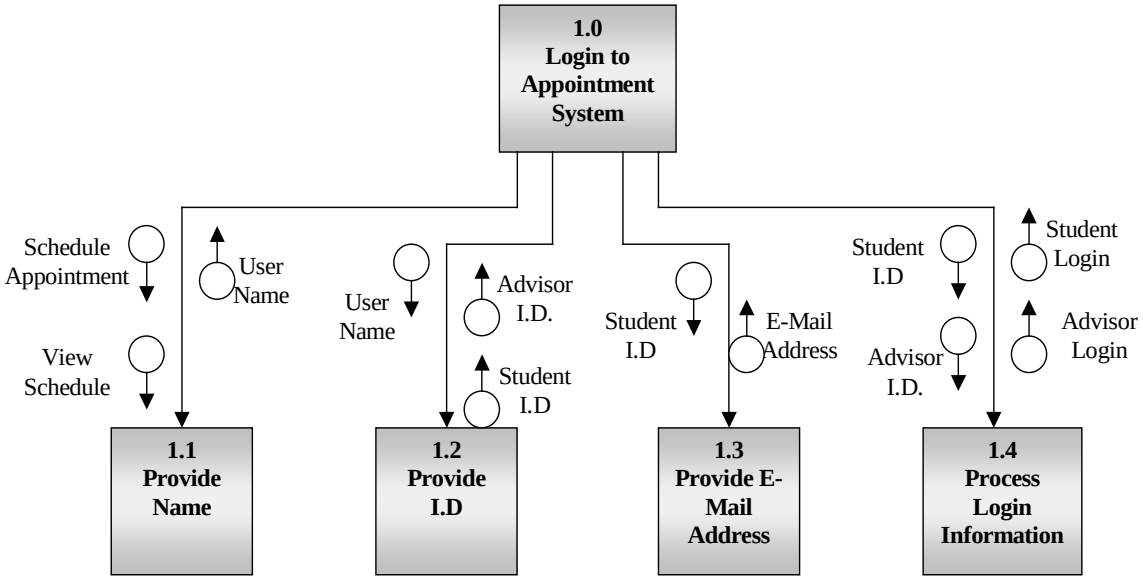
More than one transition may leave a state. The first event to occur causes the corresponding transition to fire. Transitions are guarded by conditions, that is, an event can only fire a transition if the condition of the event is true. A condition is valid over an interval of time.

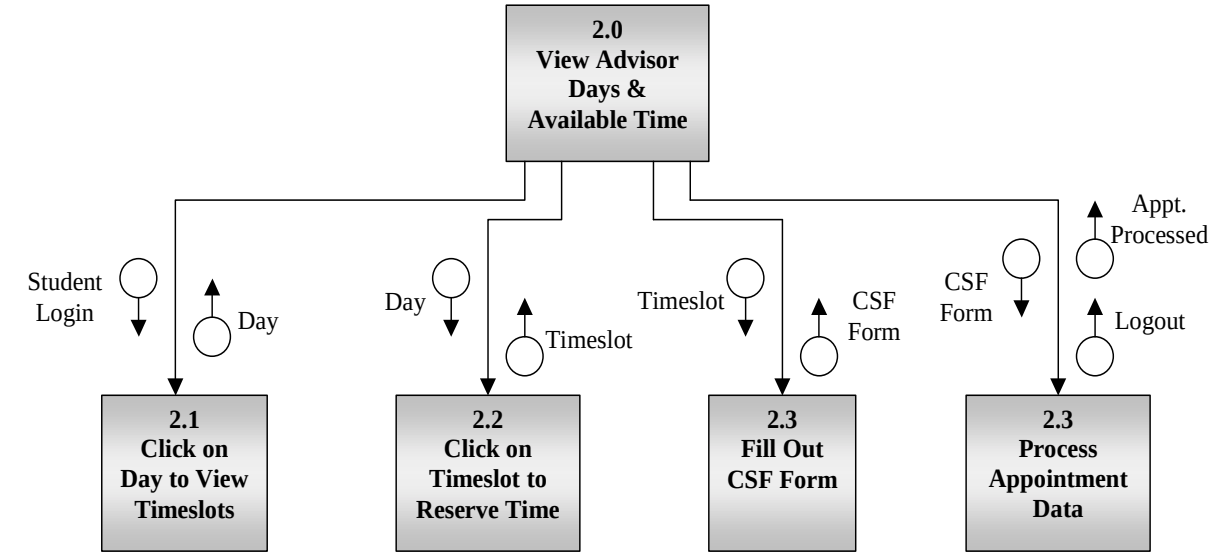
- The messages sent to classes: the event messages

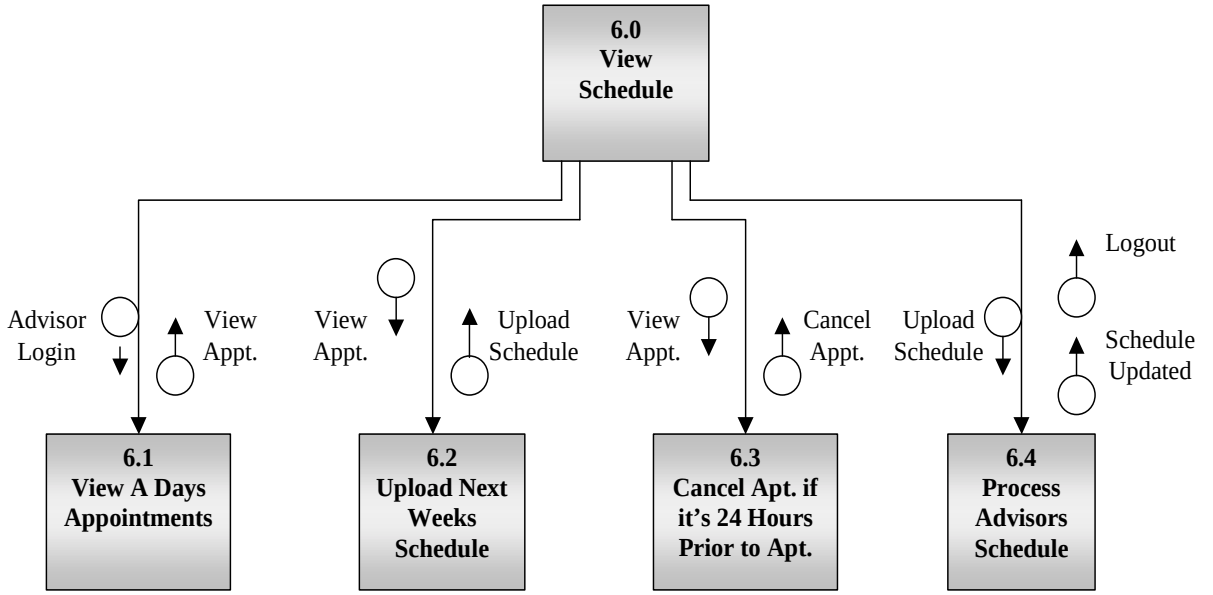
4.3 MODULAR DECOMPOSITION

4.3.1 STRUCTURE CHARTS - TO-BE SYSTEM





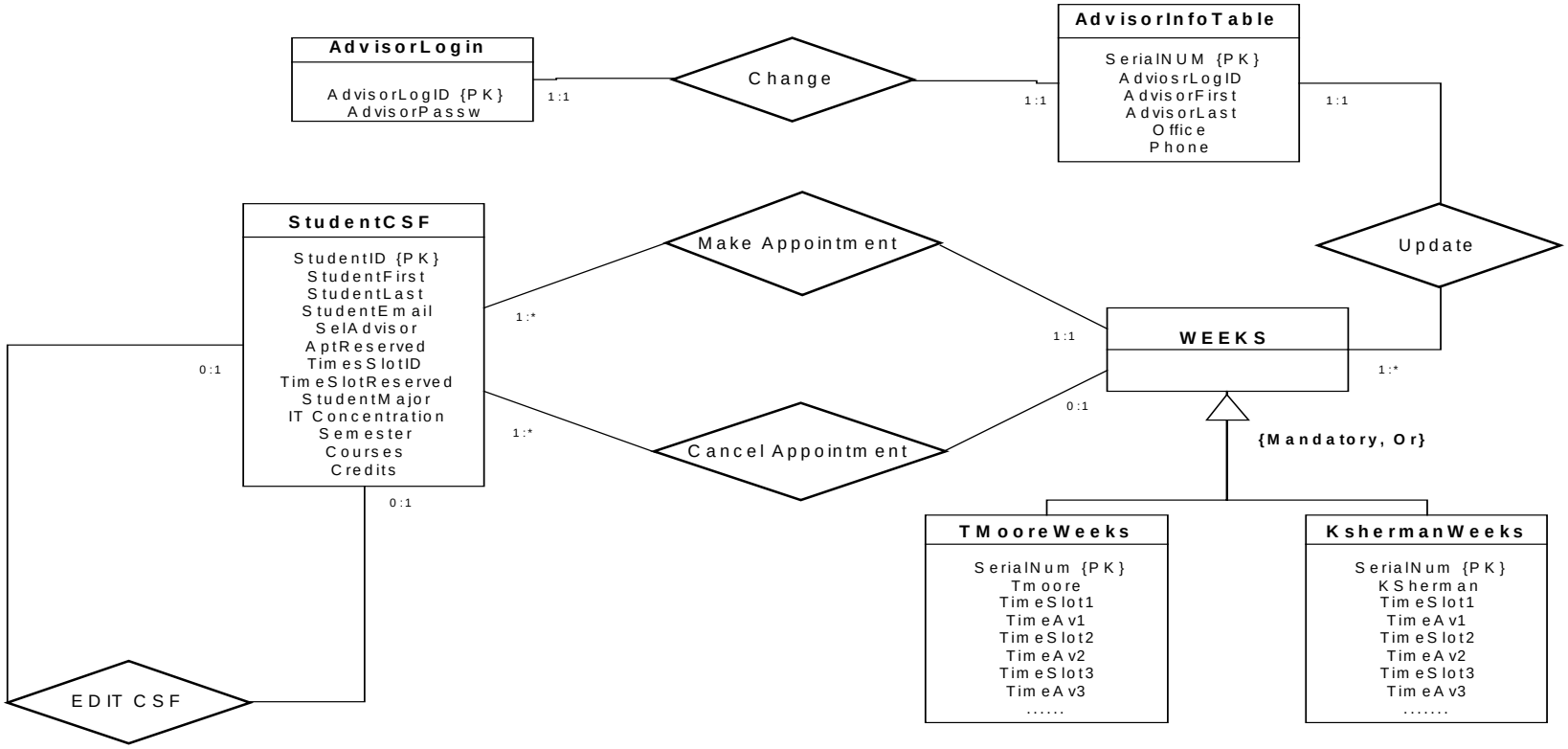

2.0
View Advisor Days & Available Time
Student Login
Day
Day
Timeslot
Timeslot
CSF Form
CSF Form
Appt. Processed
Logout
2.1
Click on Day to View Timeslots
2.2
Click on Timeslot to Reserve Time
2.3
Fill Out CSF Form
2.3
Process Appointment Data


6.0
View Schedule
Advisor Login
View Appt.
View Appt.
Upload Schedule
View Appt.
Cancel Appt.
Upload Schedule
Logout
Schedule Updated
6.1
View A Days Appointments
6.2
Upload Next Weeks Schedule
6.3
Cancel Apt. if it's 24 Hours Prior to Apt.
6.4
Process Advisors Schedule

4.4 DATABASE DESIGN

4.4.1 ER MODEL

Entity-Relationship (ER) Diagram
for
The Web Based Appointment System



4.4.2 DATABASE DESCRIPTION

This system describes the final version of a database for NJIT's Web-Based Online Appointment System. The first version of our database system was initiated as a test subject for normalization. The second version was more detailed and included more information about the entities and their relationships. The third version of our database is quite a transformation from what we had started in the first version and it helps develop clear entities and strong relationships amongst the different fields in a normal form. The final version saw changes in order to for the front-end to connect with the database. It is a departure from the first three versions. As we moved along in the implementation phase we saw that we needed to change tables and attributes around.

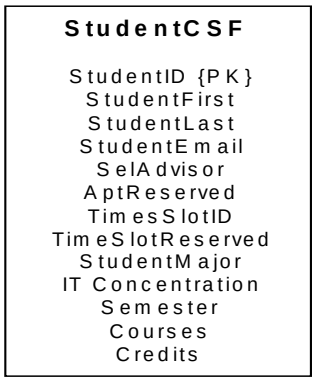
This new version provides considerably more detail, tables, and relationships. The first two versions included 4 tables with minimal attributes, while the third version covers all the necessary information the users, in this case the students and advisors, could possibly need to conduct their scheduled meetings as hassle-free as possible. The final version incorporates all the functions of the third version but it is designed to work better with the online system. Also we had to update the timeslots because of the fact the sponsors updated us on their schedules.

There are a total of 19 tables latest database in order eliminate redundancy and to better organize the information for the database administrator and the end-users. In comparison, the third version only had a total of 7 tables. Cleary these improvements made in each version highlight the fact that this needs to be seen as a process of continuous improvement rather than a one-shot effort. As the project moves deeper into the implementation phase, the database could see major changes and updates that were unexpected, but as of now, we think it is sufficient enough and easy to follow and understand.

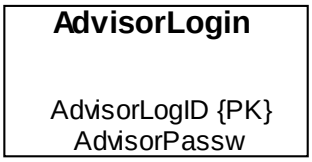
Database Design

In total there are seven tables currently in our database, each one with their own purpose. They are as follows:

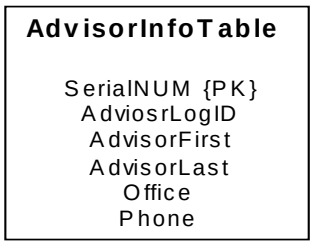
1.) StudentCSF



2.) AdvisorLogin

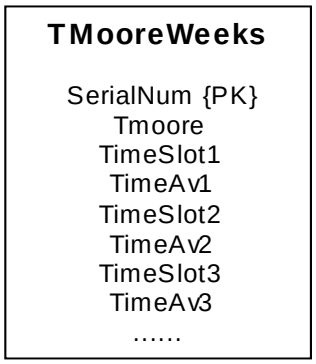


3.) Advisor InfoTable



4.) KshermanWeek (1 –8)

5.) TmooreWeek (1-8)



The StudentCSF table we have created contains 13 attributes that will uniquely identify each of the students in the database. The first attribute is Student ID, which will serve as this table's primary key. Also in this table is the first and last name for the student. There are two foreign keys that create a relationship with two other tables; one is SelAdvisor, which will join each student with their advisor. Next is AptReserved, which will point to the student's individual appointments in the corresponding week. The TimeSlotID will be used to better organize the timeslots and allow the advisor to search for particular students, or free timeslots. Also this table will hold the CSF, which holds the student's information regarding courses they are planning to take in the following semester. Such information as the student's email, major, ITConsentration, and the courses will be stored within StudentCSF.

The AdviosrLogin table will have only two attributes. The first one is AdvisorLogID, which is their login ID and AdivsorPassw, which will serve as the advisor's password.

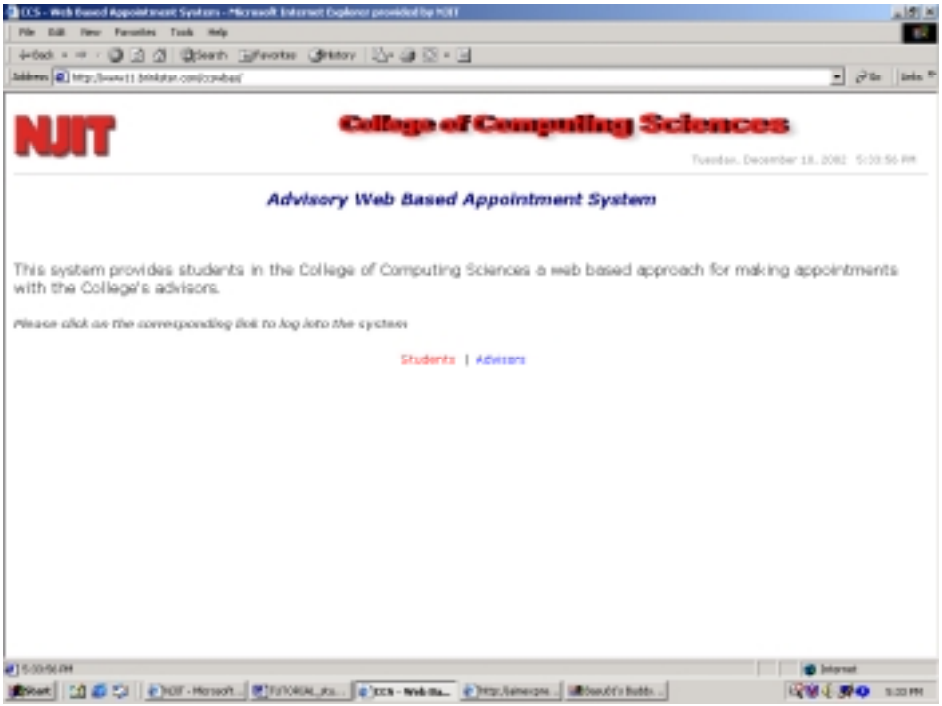
The AdvisorInfoTable table will serve a similar purpose as the StudentCSF table. It is just to store the advisor's personal information that a student can access, such as email, phone, office room, etc. The primary key will be the SerialNum attribute, and the foreign key is the AdvisorLogID that will connect the advisor with the AdvisorLogin table. We separated them to ensure that the advisor's usernames and passwords would be safe from the system's end-users.

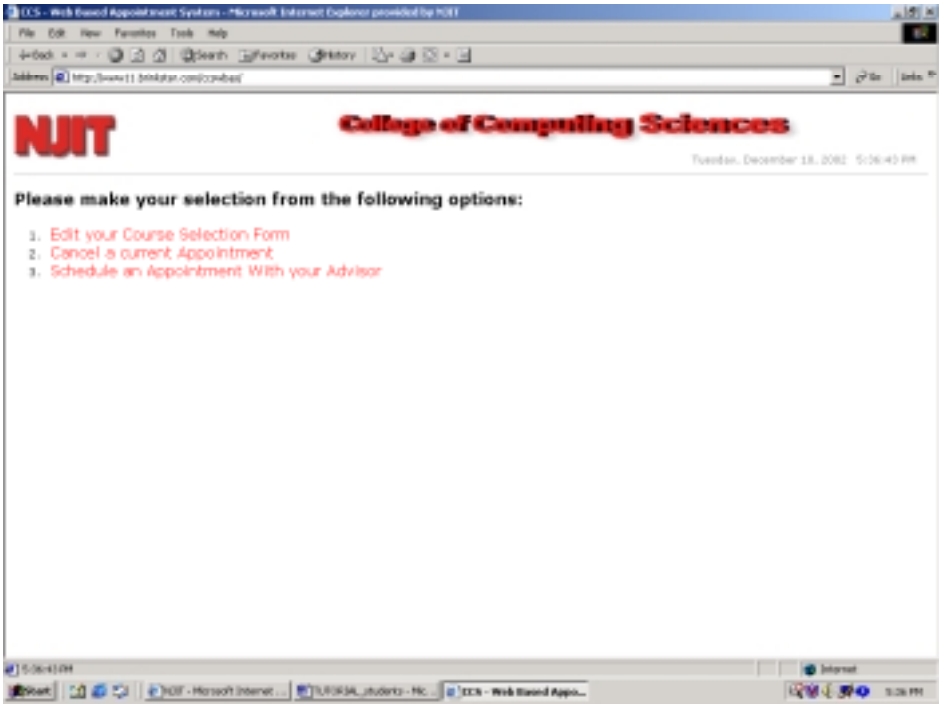
Both the KshermanWeeks and the TMooreWeeks will act as the core tables for this database. There will be 8 tables of each within the system in order to reflect the 8 weeks in the pre-registration phase. These tables will allow the student to make an appointment with their advisor. The primary key is the SerialNum, which is one of the primary keys in the AdvisorInfoTable. When a student makes an appointment, the database will generate a TimeSlotID that will join that student with the appropriate week table. Other attributes inside this table are: Ksherman or Tmoore that will allow the system to uniquely identify which advisor is responsible for that week. The other attributes for both tables are Timeslot and TimeAv. Each one will be repeated numerous times in order to reflect an advisors schedule. Each Timeslot will store a 15-minute slot for the students to choose. The TimeAv will hold if that particular timeslot is available or unavailable, that information will be passed onto the system and displayed for the students to see.

In conclusion, all the tables discussed above are fairly easy to understand and they all serve a major point, as mentioned above. We used primary keys and foreign keys to further simplify the process between the tables. We also used 1st, 2nd, and 3rd normalization tests to cut down on redundancy. Of course as each new function is added to the system, the database could see some more changes.

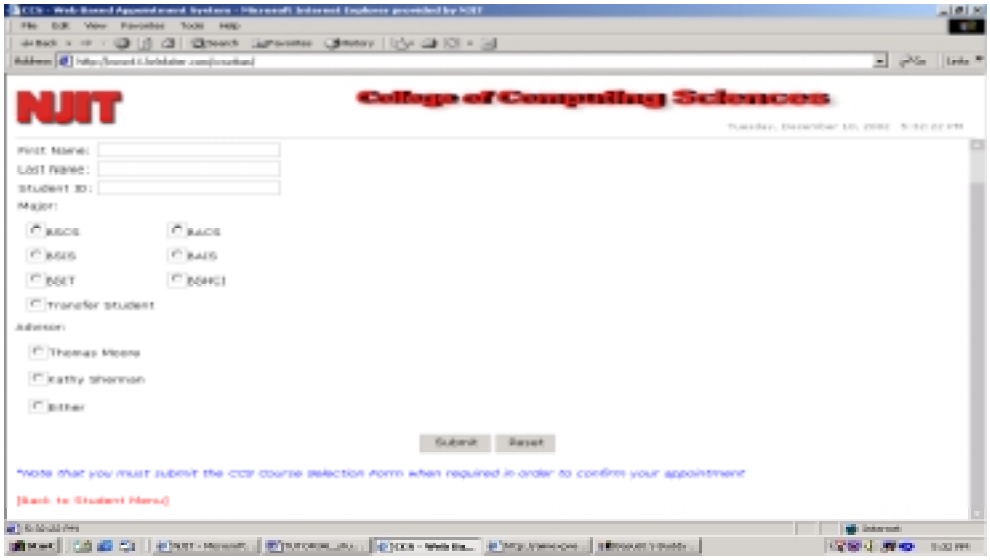
4.5 USER INTERFACE DESIGN

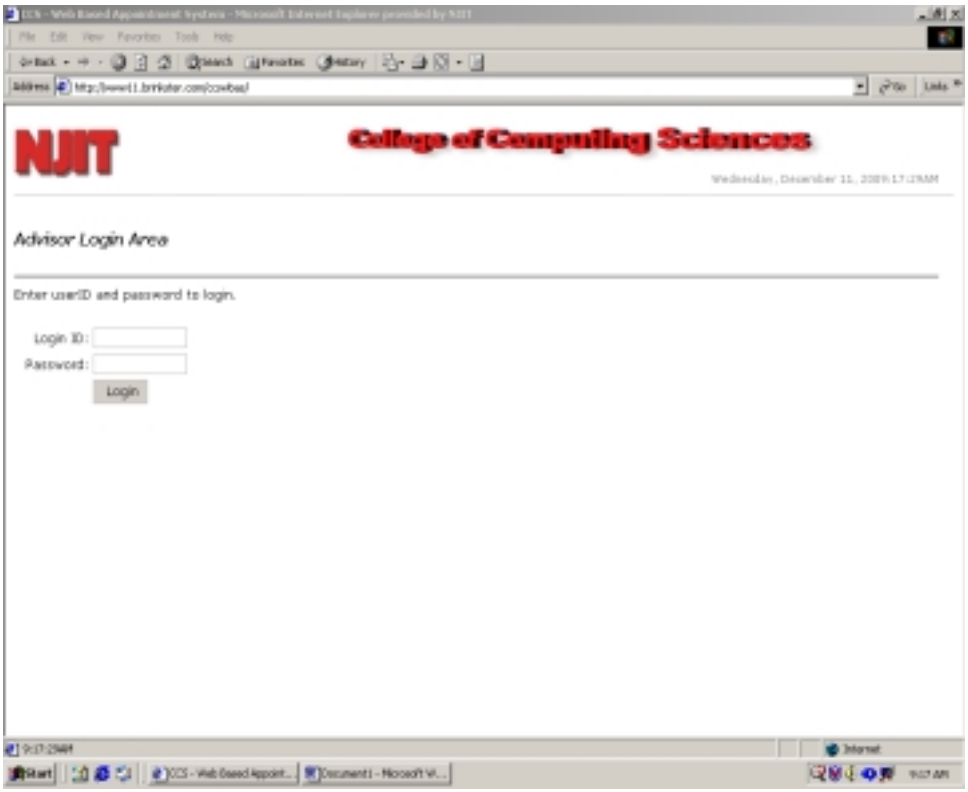
4.5.1 SCREENSHOTS OF THE USER INTERFACE

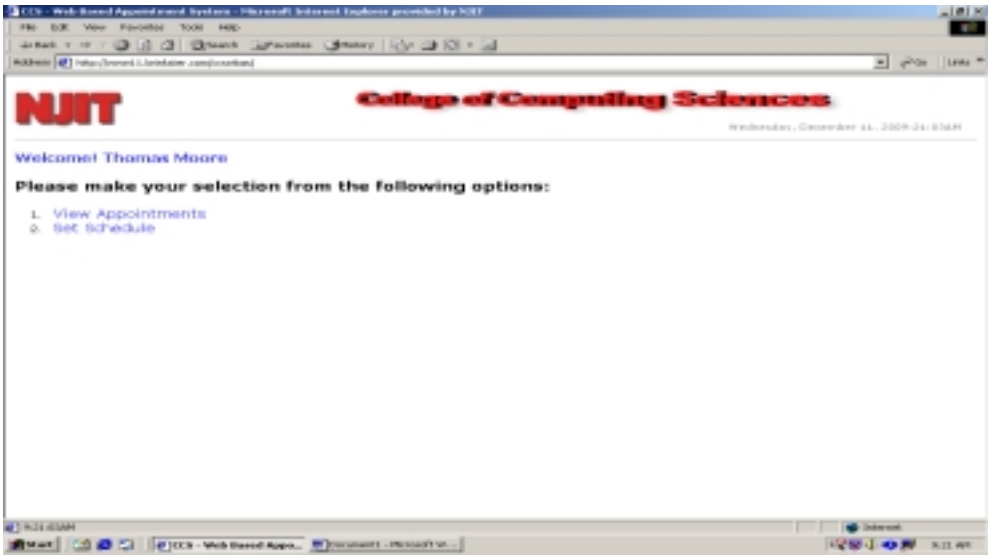


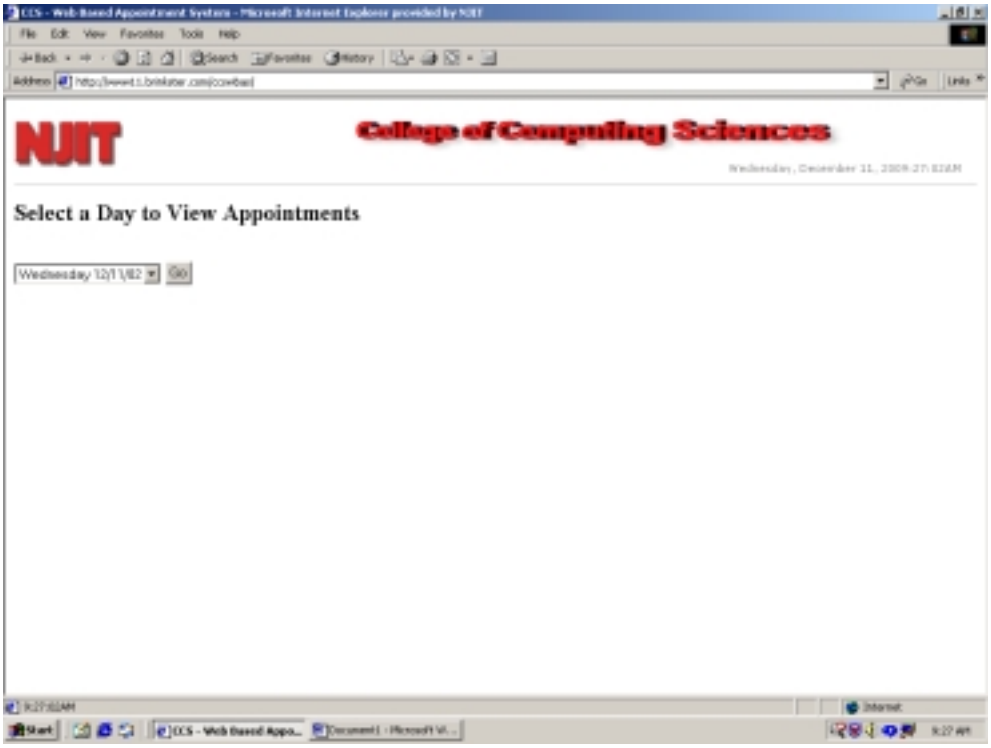


P0004


NJIT
College of Computing Sciences
First Name:
Last Name:
Student ID:
Major:
Transfer Student
Advisor:
Thomas Moore
Kathy Sherman
Either
Submit
Reset
*Note that you must submit the CCS course selection form when required in order to confirm your appointment
[Back to Student Menu]


NJIT
College of Computing Sciences
Advisor Login Area
Enter userID and password to login.
Login ID:
Password:
Login


NJIT
College of Computing Sciences
Welcome! Thomas Moore
Please make your selection from the following options:
1. View Appointments
2. Set Schedule


NJIT
College of Computing Sciences
Select a Day to View Appointments
Wednesday 12/1/82
Go

5. SYSTEM GUIDE

5.1 USERS' TUTORIAL

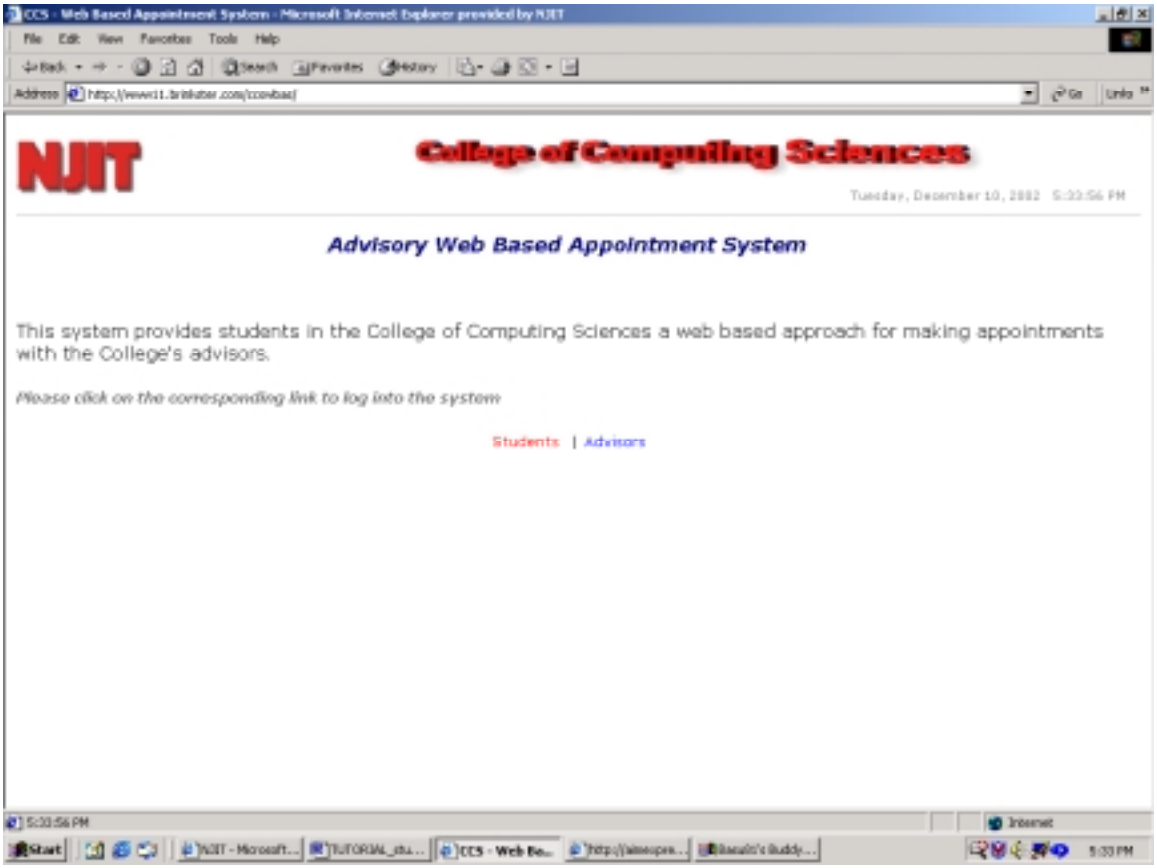
ADVISORY WEB BASED APPOINTMENT SYSTEM

The main function of an Advisory Web Based Appointment System is to enable students to make/schedule appointments with their advisors via online. Likewise, this system also enables advisors to check/reschedule their appointments, as they seem fit. This tutorial will give you a thorough understanding of our Advisory Web Based Appointment System.

5.1.1 Student's Tutorial

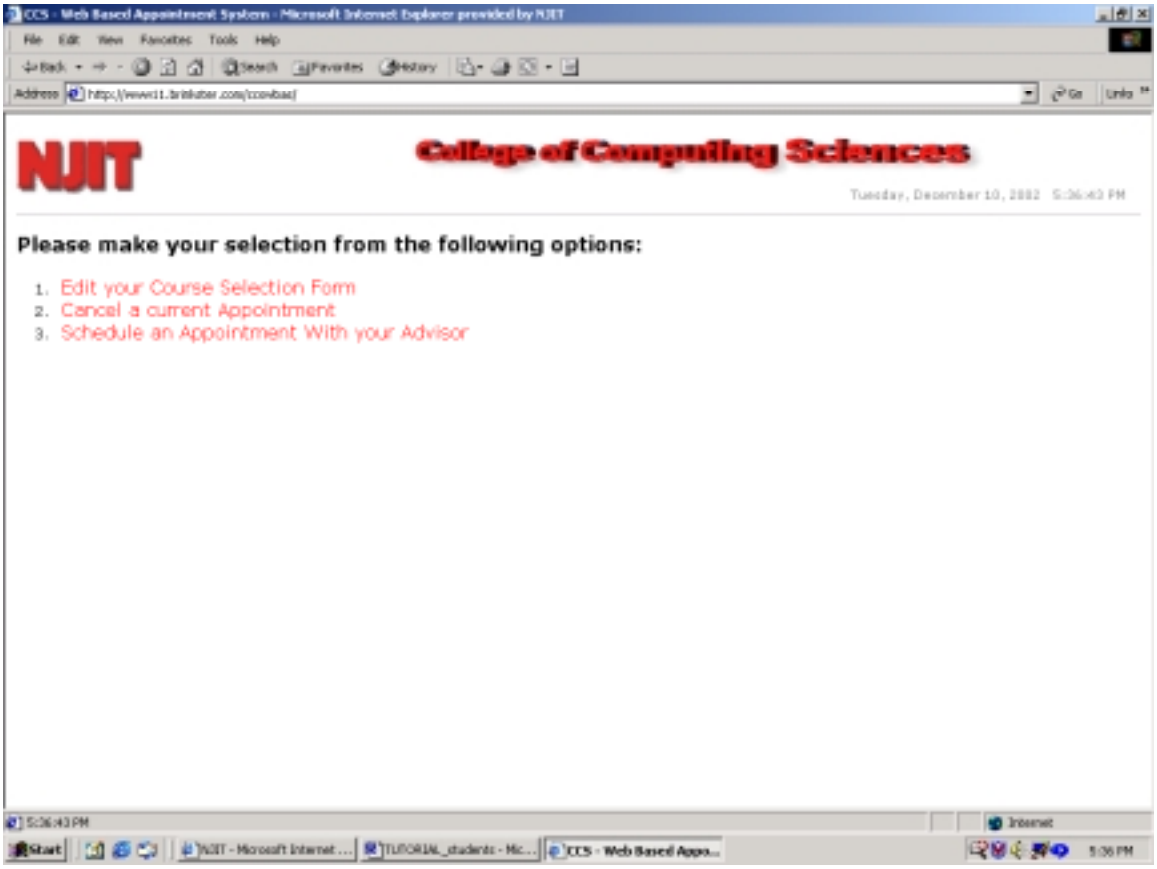
The main page in the “Advisory Web Based Appointment System” will allow you to login to the system as a Student or Advisor. This page will comprise of two links, one for students and one for advisors, as shown below:

(Figure 1)



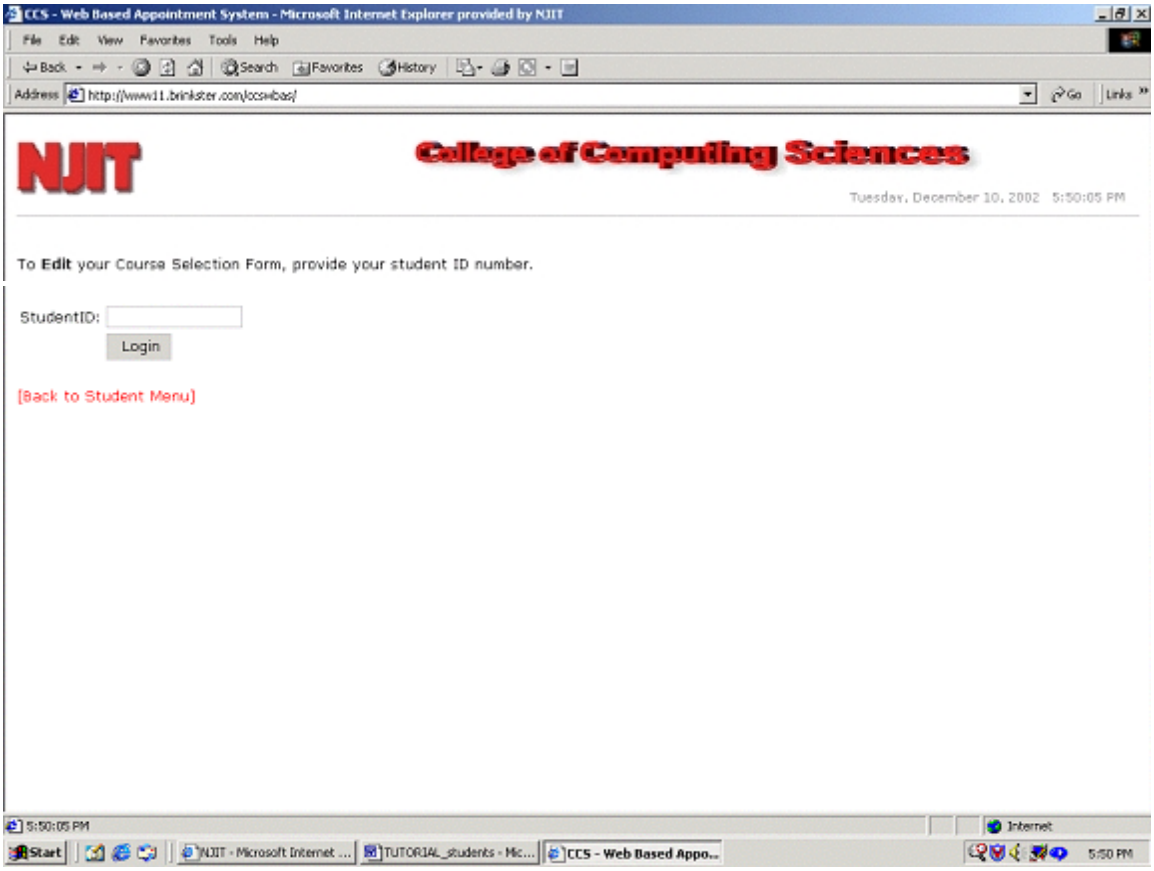
If you are a student, then click on the ‘Students’ link, and you will be directed towards an Options page where you will have three options. These options will be listed as shown below:

(Figure 2)



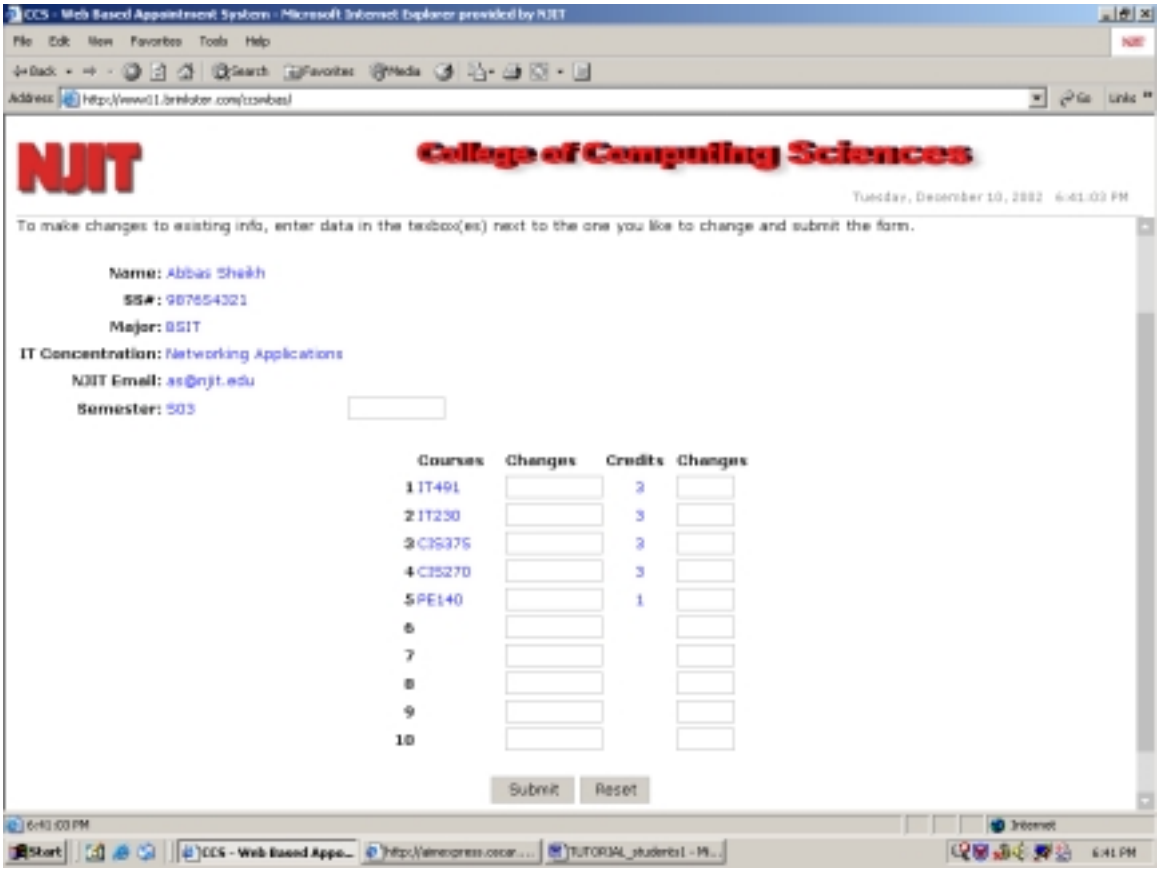
If you are a student and wish to edit your course selection form, you can make the selection from the Options page. In this page, you will click on the first option listed as 'Edit you Course Selection Form' to make any changes. After clicking on the link, you will be sent to a login page, where you will be asked to provide your student ID number, as shown below:

(Figure 3)



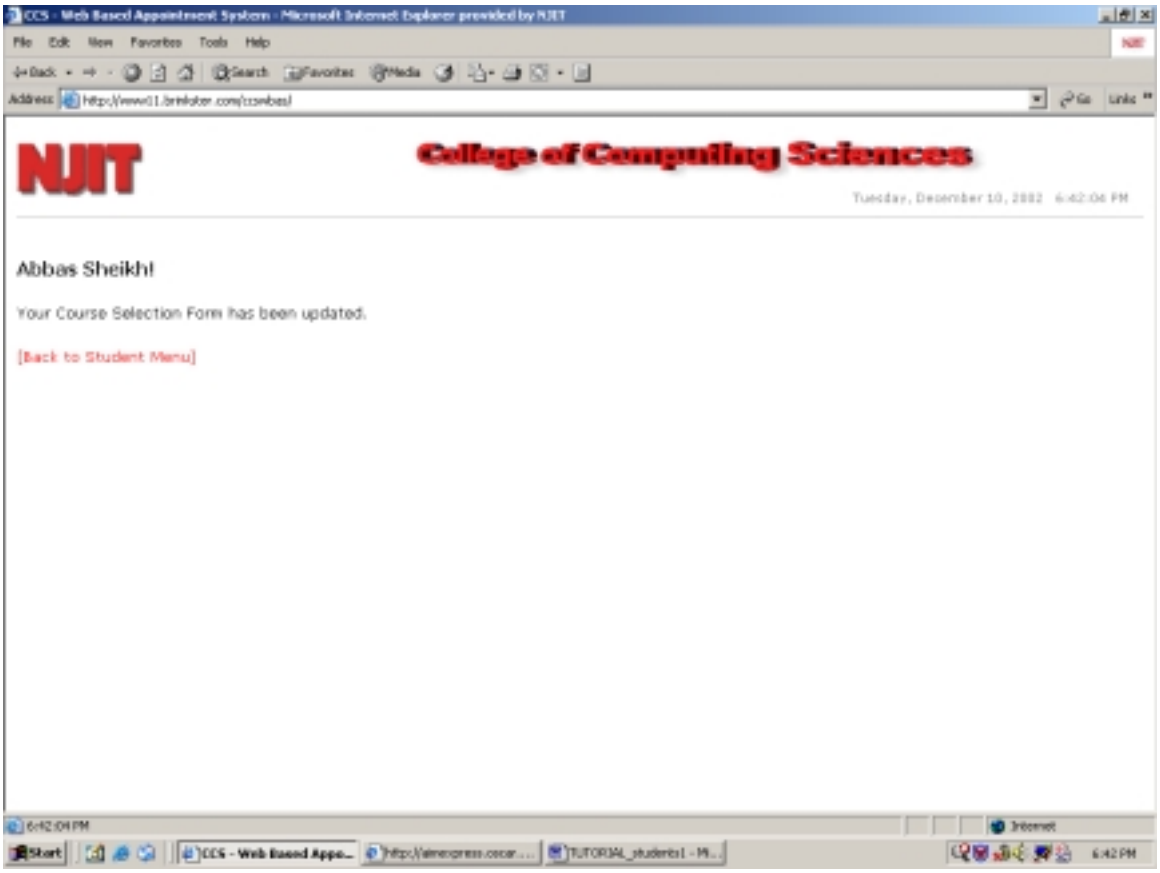
Upon entering your student ID number, click on the 'login' button to enter your course selection form. Once you have entered the course selection form, you can make any changes or corrections onto the form, as shown below.

(Figure 4)



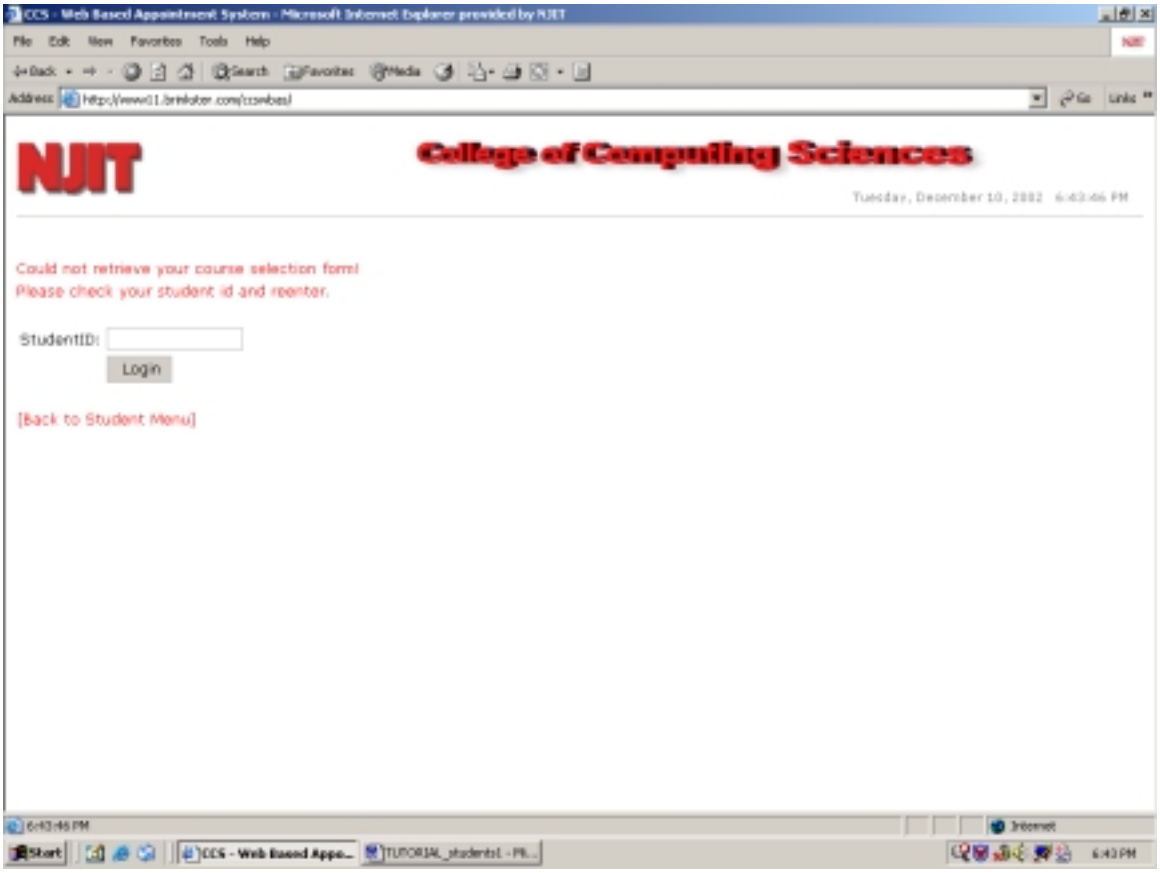
The Edited Course Selection Form will be displayed with your Name, SS#, NJIT E-mail, Major, and Semester. You may change data in the text boxes as shown above. Upon completion of the form, you will click on the 'Submit' button to submit the edited form. You will be directed to a page where you will receive a confirmation stating that your information has been updated as shown below.

(Figure 5)



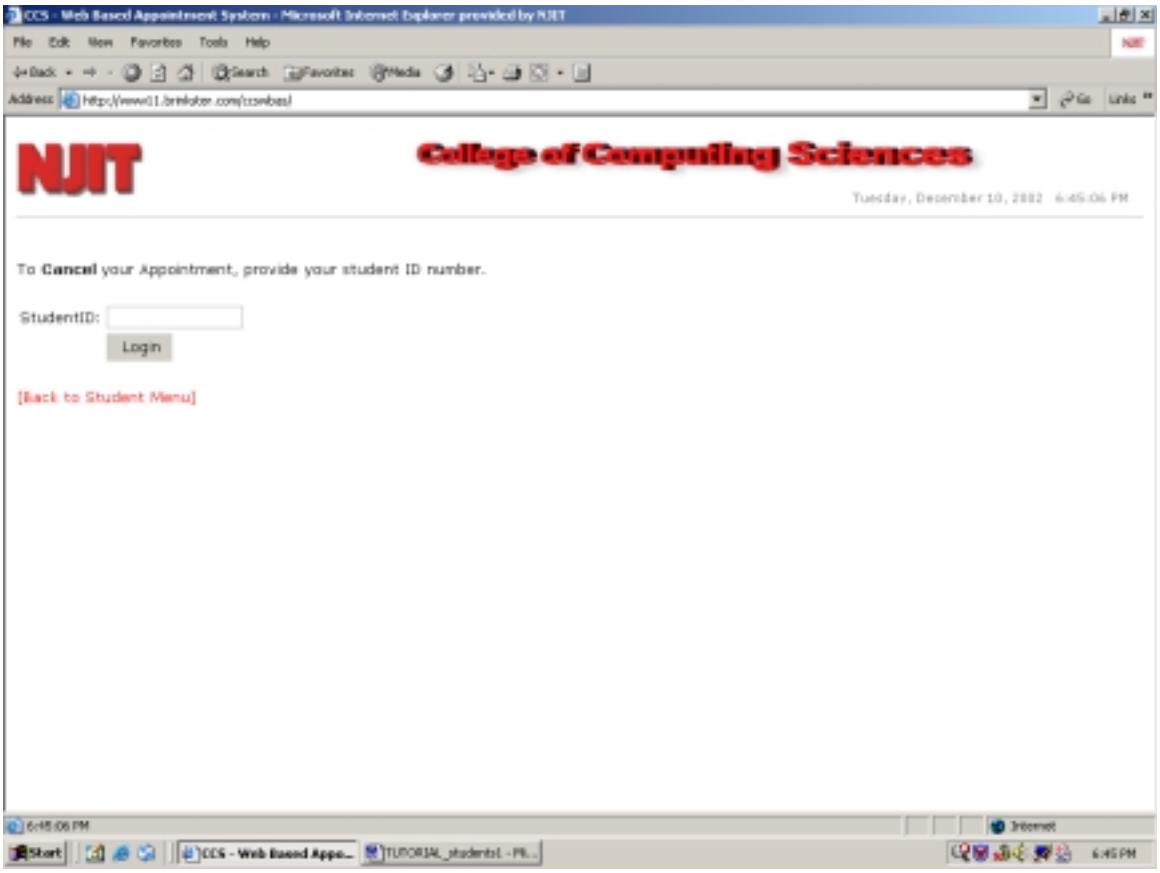
If your student ID number is not valid or recognized, you will be asked to reenter your student ID and will receive the following statement:

(Figure 6)



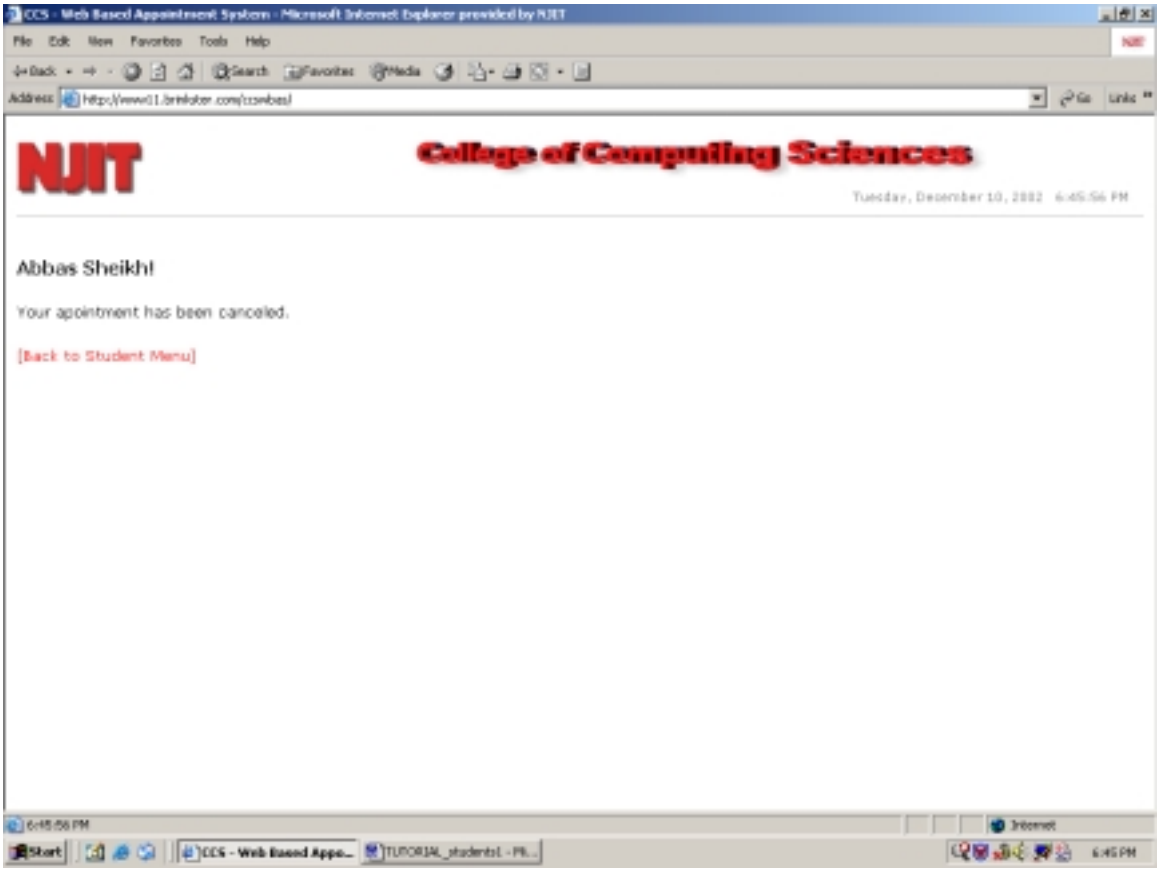
If you are a student and wish to cancel an appointment that you have scheduled with your advisor, you can make the selection from the Options page. In this page, you will click on the second option listed as 'Cancel a current Appointment' to cancel your appointment. After clicking on the link, you will be sent to a login page, where you will be asked to provide your Student ID number, as shown below:

(Figure 7)



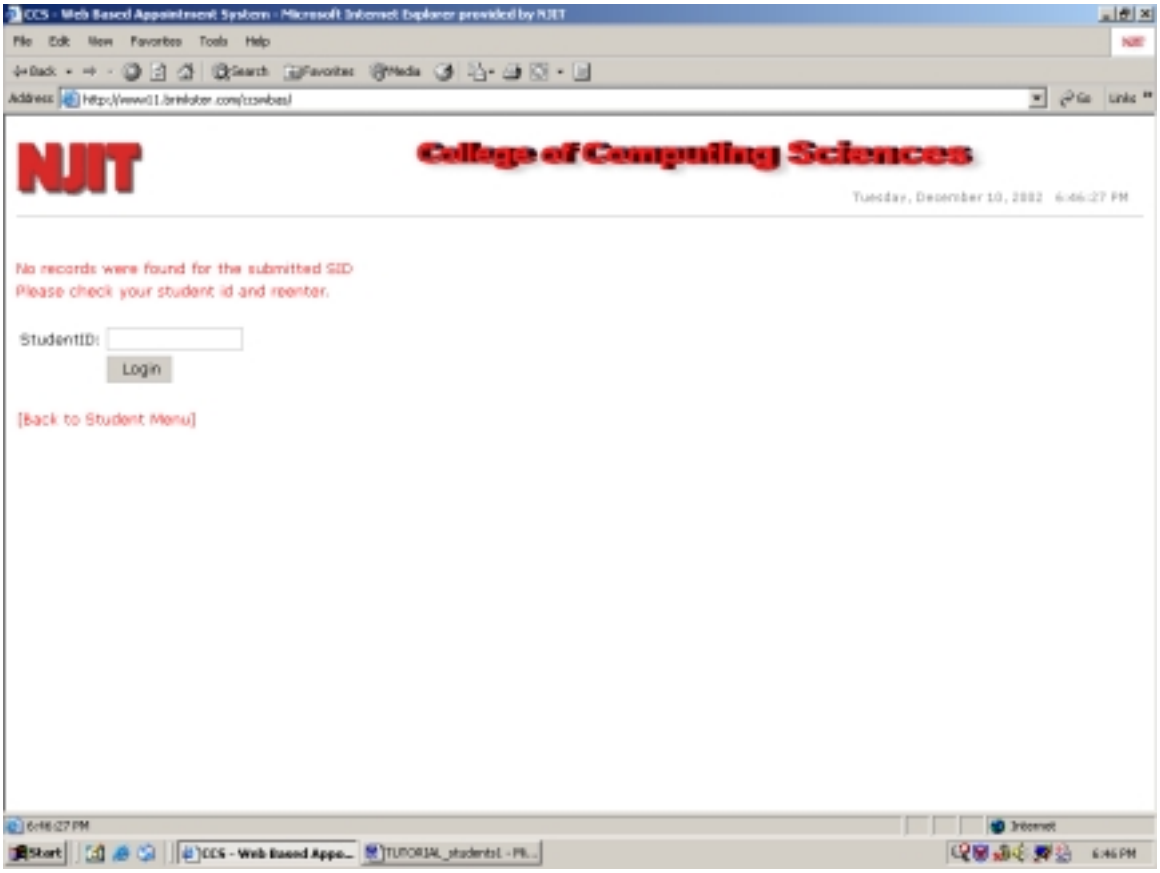
Upon entering your Login ID number, click on the ‘Login’ button. Once your student ID is recognized, you will be directed towards a page that will remind you of your appointment and ask you if you wish to cancel it, in the following format:

(Figure 8)



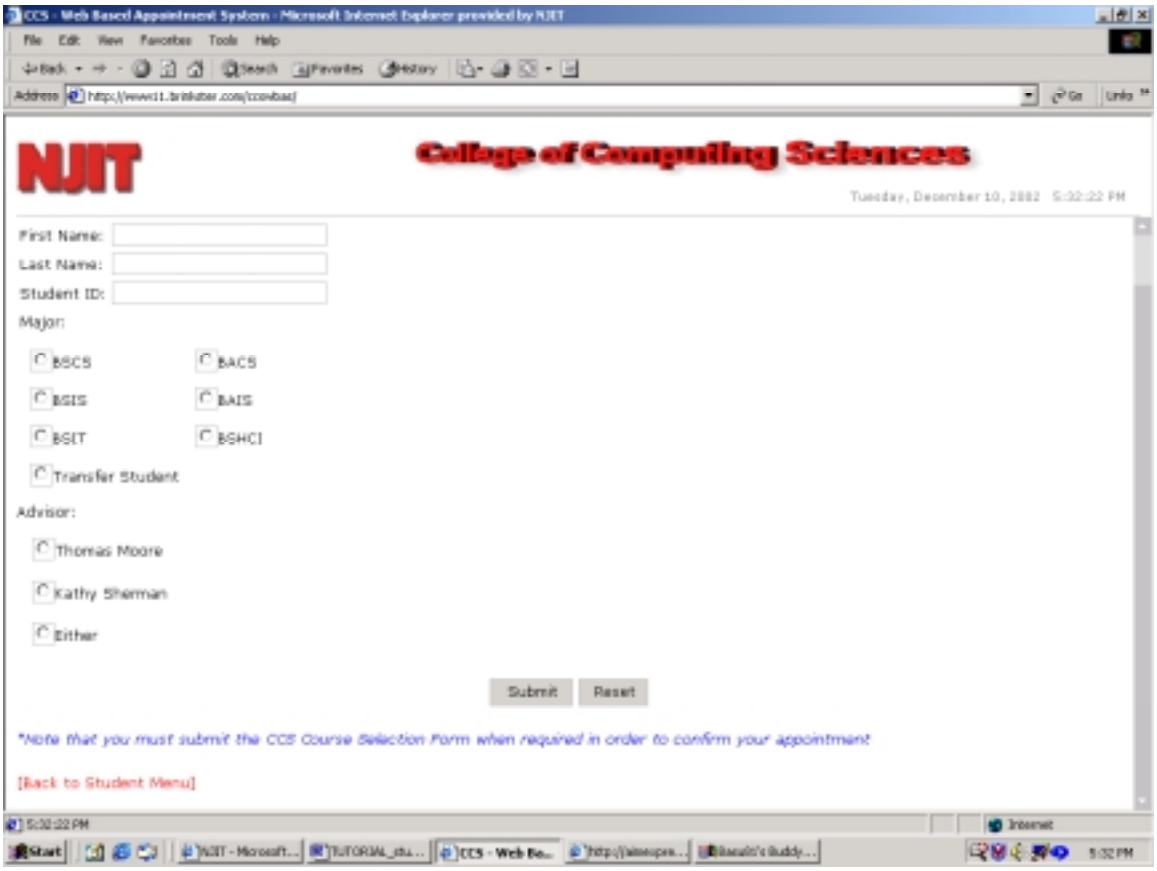
However, if your student ID number is not valid or recognized, you will be asked to reenter your student ID and will receive the following statement:

(Figure 9)



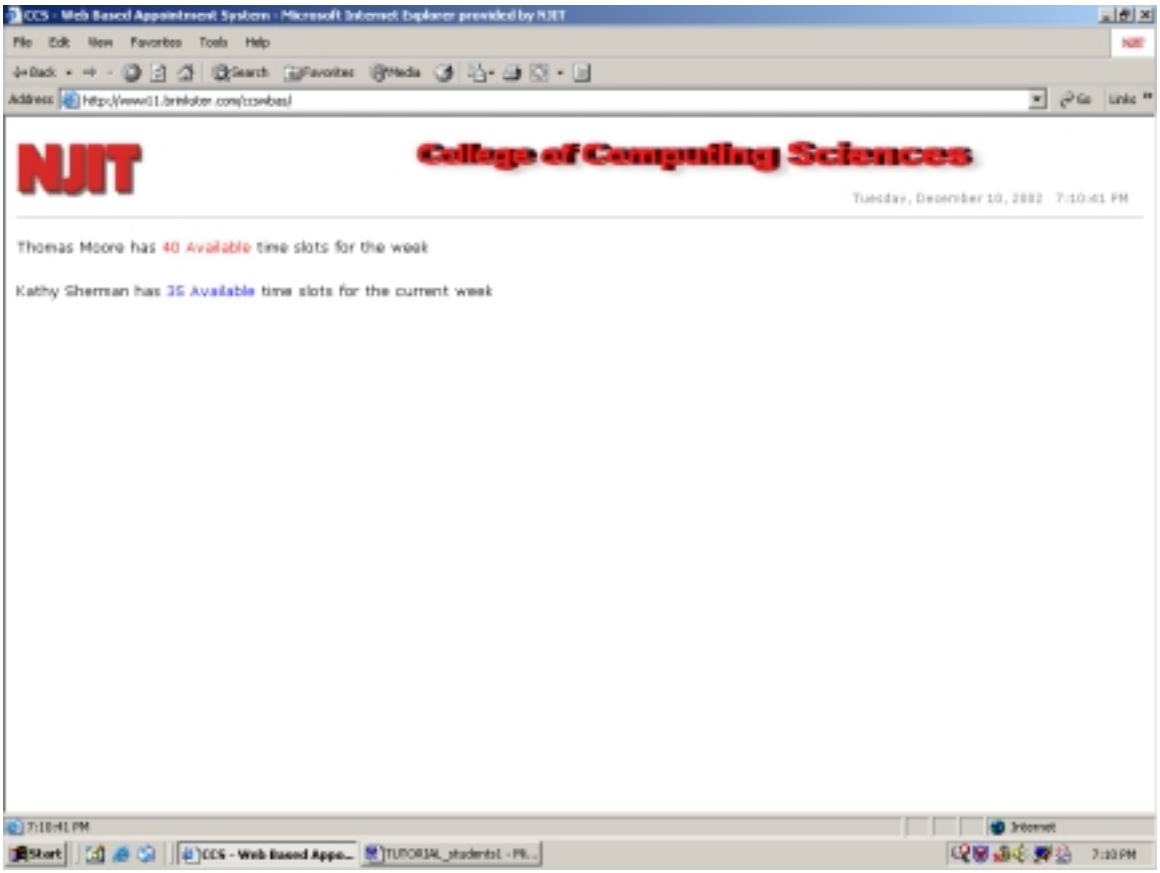
If you are a student and wish to schedule an appointment with your advisor, you can make the selection from the Options page. In this page, you will click on the third option listed as 'Schedule an Appointment With your Advisor' to make an appointment. After clicking on the link, you will be sent to a page where you will be asked to provide all the necessary information to see the advisor, as shown below:

(Figure 10)



If you are unsure of your advisor, then click on ‘Either’ in Figure 10. Clicking on this option will direct you to a page that lists the number of appointments that each advisor is scheduled for, as shown below:

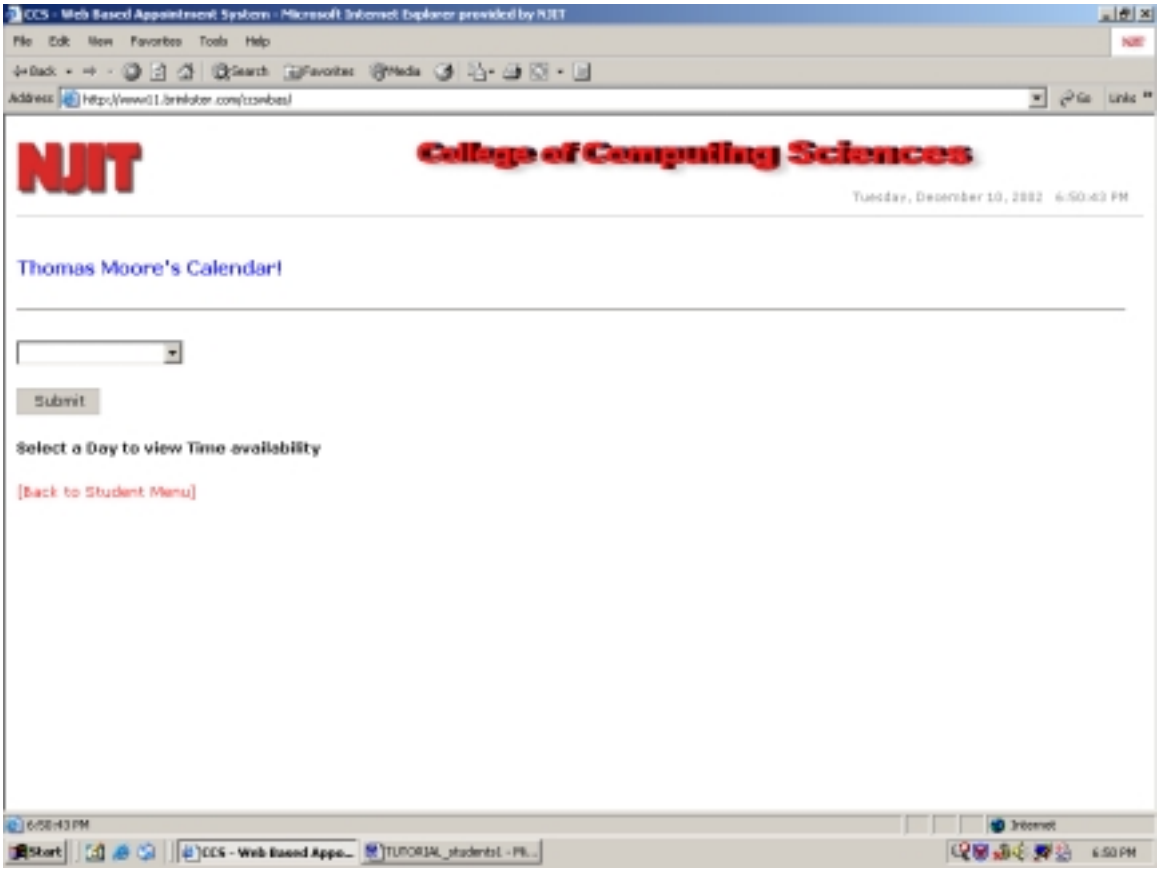
(Figure 11)



After viewing the number of slots available for each advisor, you can make your appointment accordingly.

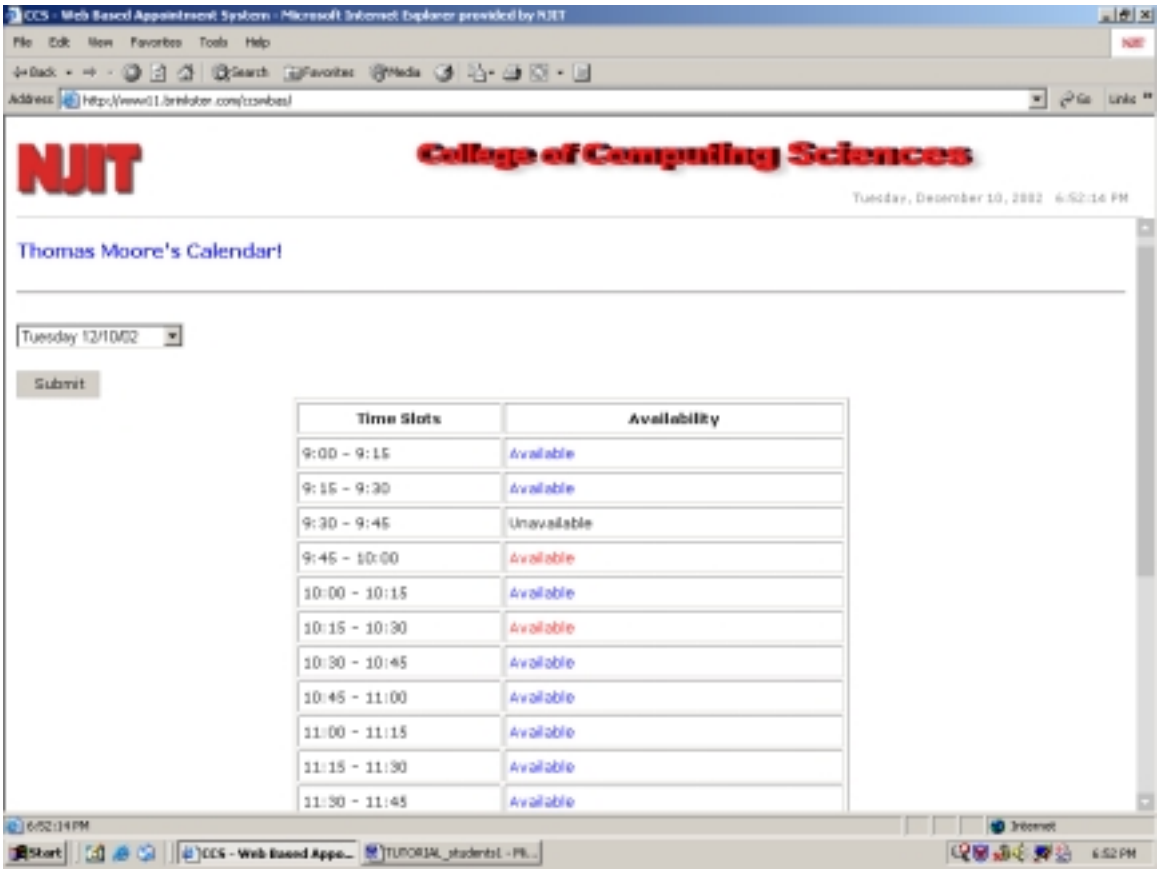
Once you have entered all the required information in Figure 10, you are given two options, ‘Submit’ and ‘Reset’. Clicking on the Reset button will clear all the fields of the information that you provided, leaving you to complete the form again. Clicking on the Submit button will direct you towards your selected Advisor’s Calendar, as shown below.

(Figure 12)



The drop down menu includes everyday of the week from Monday on through Friday. Clicking 'Submit', after selecting a day of the week, will direct you towards the following page:

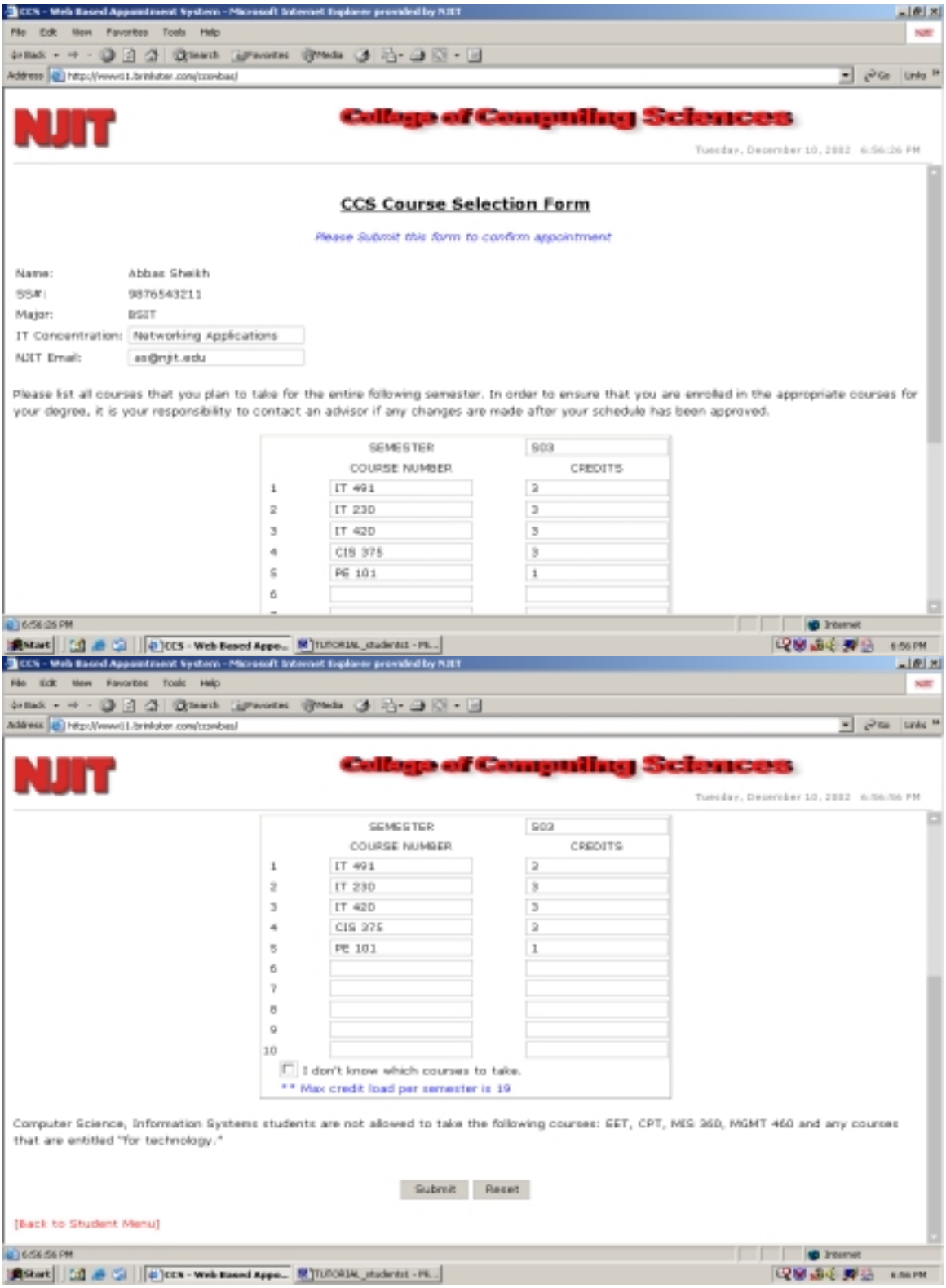
(Figure 13)



This page will consist of all the timeslots and availability periods of the advisors, indicated for that particular day, from 9am till 5pm. The 'Unavailable' period of the timeslot indicates that the period is not available for students to schedule and appointment with their advisor. However, the 'Available' period of the timeslot indicates an opening for students to schedule an appointment.

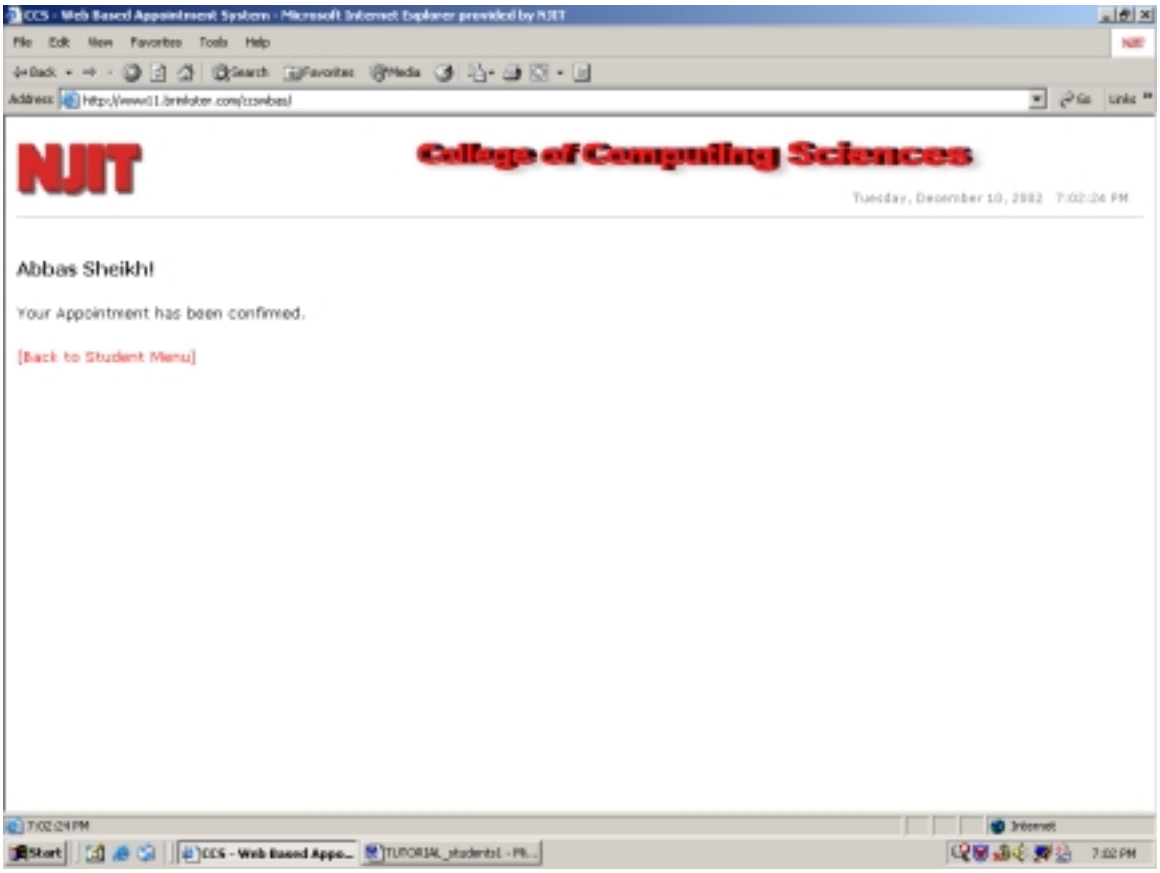
If you wish to make an appointment, click on the 'Available' hyperlink. Clicking on 'Available' should direct you towards the 'CCS Course Selection Form', which needs to be filled out before an appointment can be made, as shown below:

Course Selection Form:
(Figure 14)



The Course Selection Form will state your Name, SS#, and Major on the top, as you had entered in Figure 10. After completing the course selection form, you will have two options of either 'Submit' or 'Reset'. If you click 'Submit' and all the fields in the course selection form are filled out properly, then your appointment will be confirmed for the assigned day and time. You will receive a confirmation in the form of a page stating:

(Figure 15)

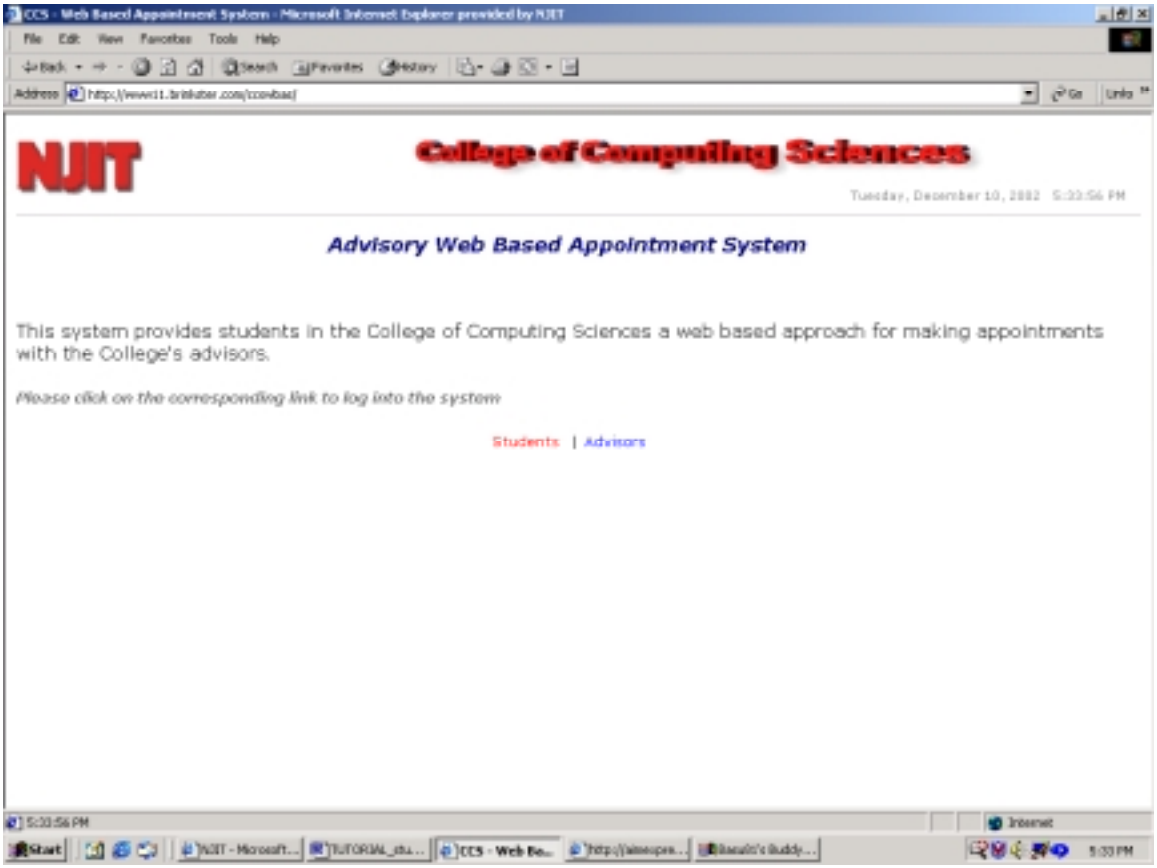


If you click 'Reset', the Course Selection Form will clear itself of all the text you input, and you will have to start entering the required fields again.

6.1.2 Advisor's Tutorial

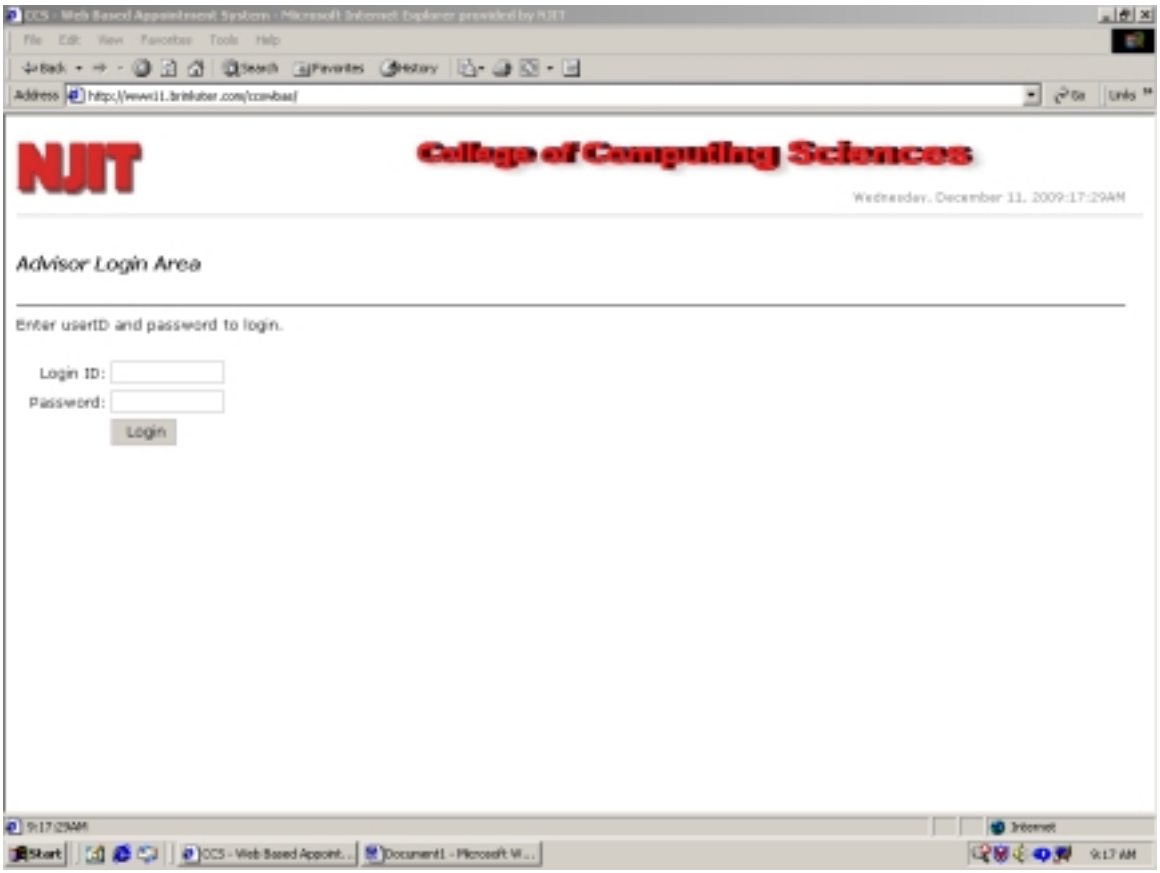
The main page in the “Advisory Web Based Appointment System” will allow you to login to the system as a Student or Advisor. This page will comprise of two links, one for students and one for advisors, as shown below:

(Figure 1)



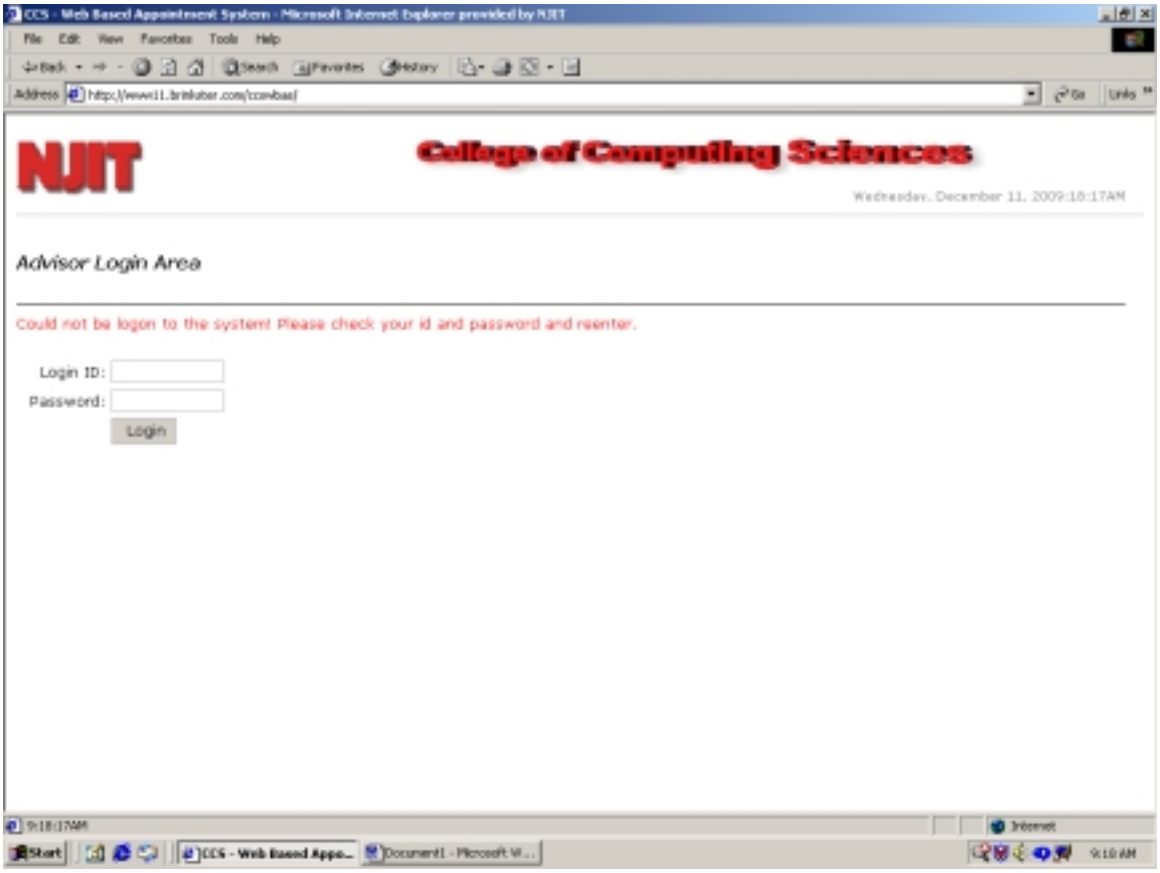
If you are an advisor, click on the ‘Advisors’ Link, and you will be directed towards an Advisor’s Login Area, where you will be asked to enter you Login ID and Password, as shown below:

(Figure 2)



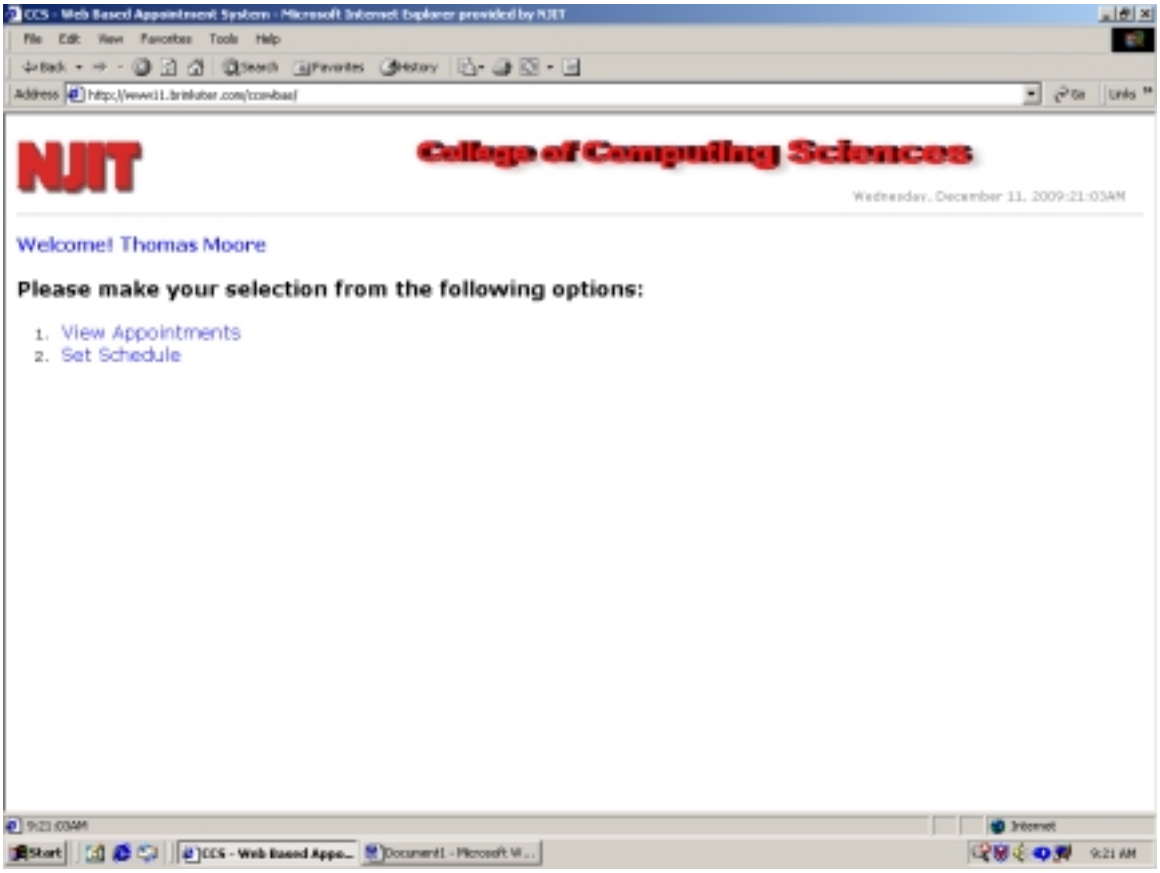
If your login ID or password is not recognized, you will be asked to reenter your ID again, as shown below:

(Figure 3)



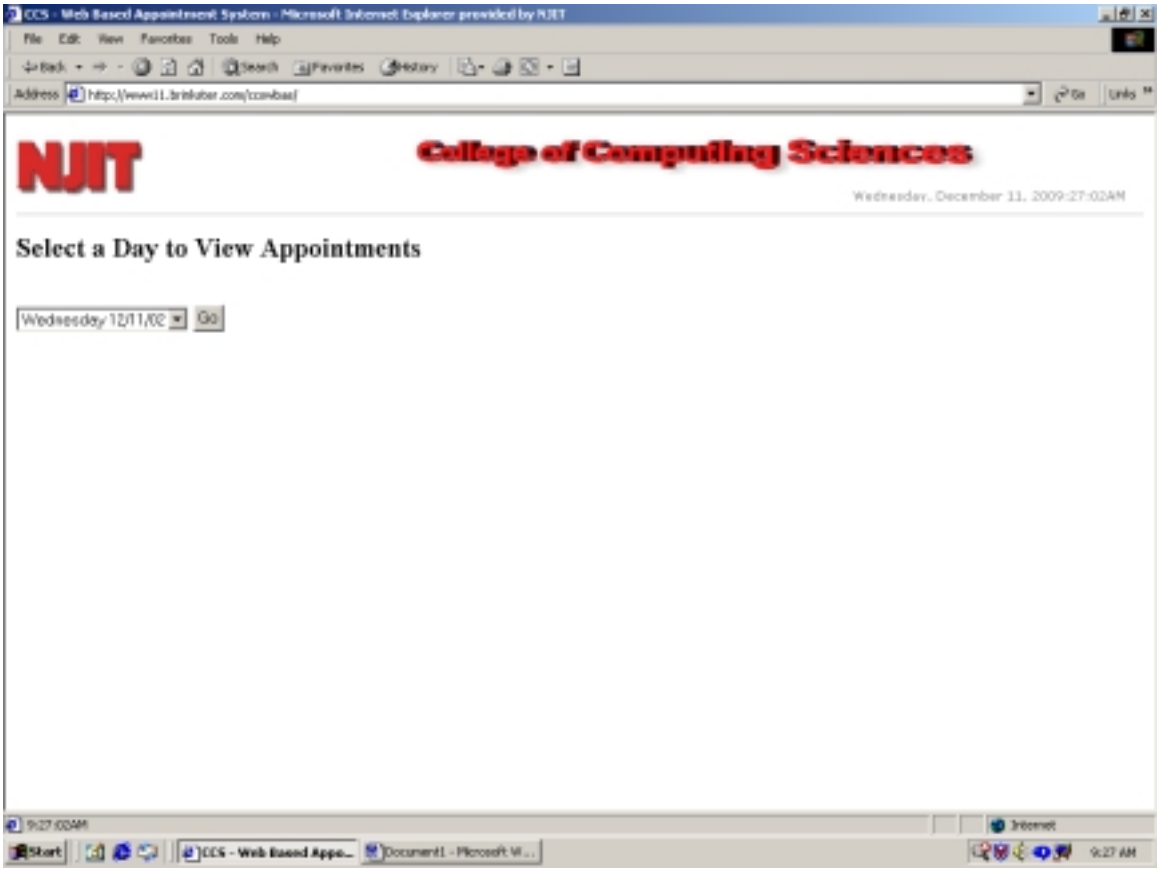
Once your Login ID and password are recognized, you will be directed towards an Options page, where you will have two options. These options will be listed as shown below:

(Figure 4)



If you wish to view your appointments, you can make the selection from the Options Page in Figure 4. You will click on the first option, labeled 'View Appointments', to see a listing of all the appointments you have within the following days. After clicking the link, you will be sent to a page where you will be asked to select a day to view appointments, as shown below:

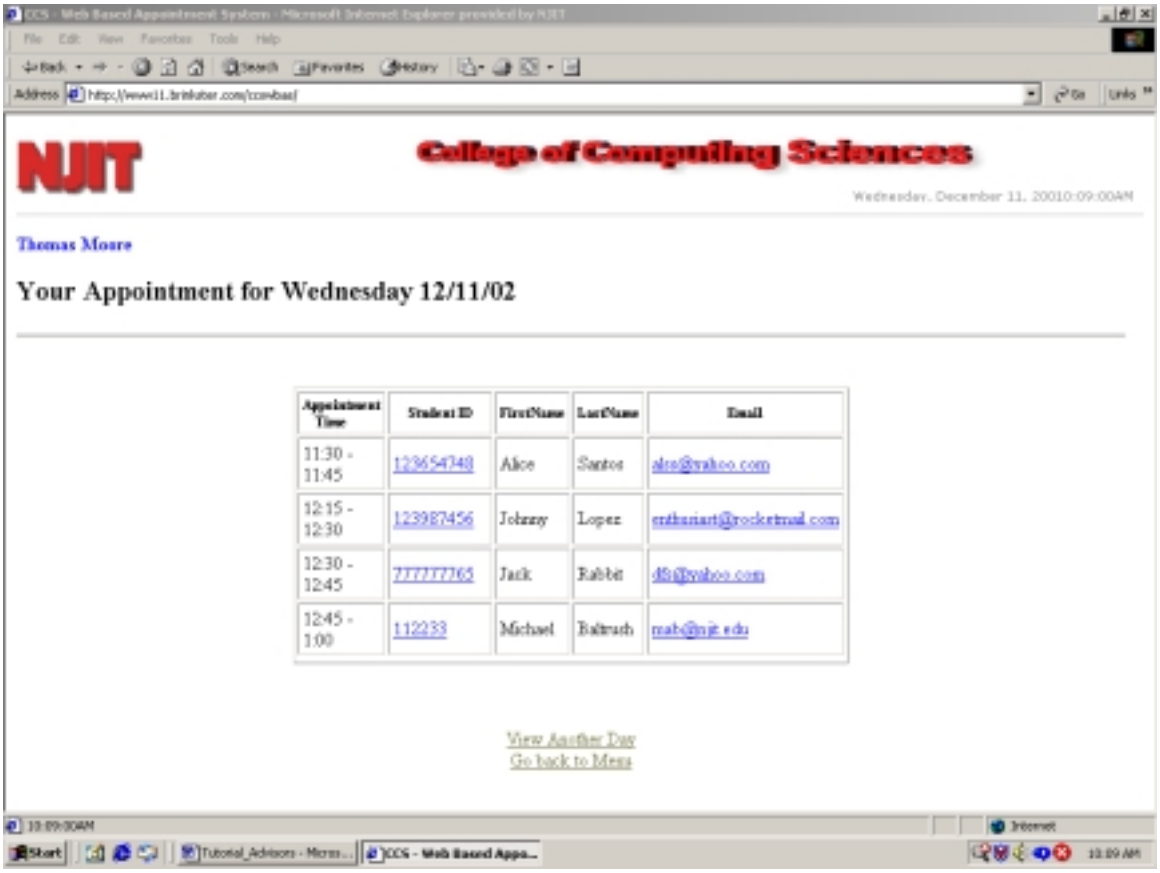
(Figure 5)



The drop-down menu shown above will give you access to view your appointments from the current day to the end of the following week.

After selecting a day to view appointments, you will click on ‘Go’ to view the appointments scheduled for that particular day, as shown below:

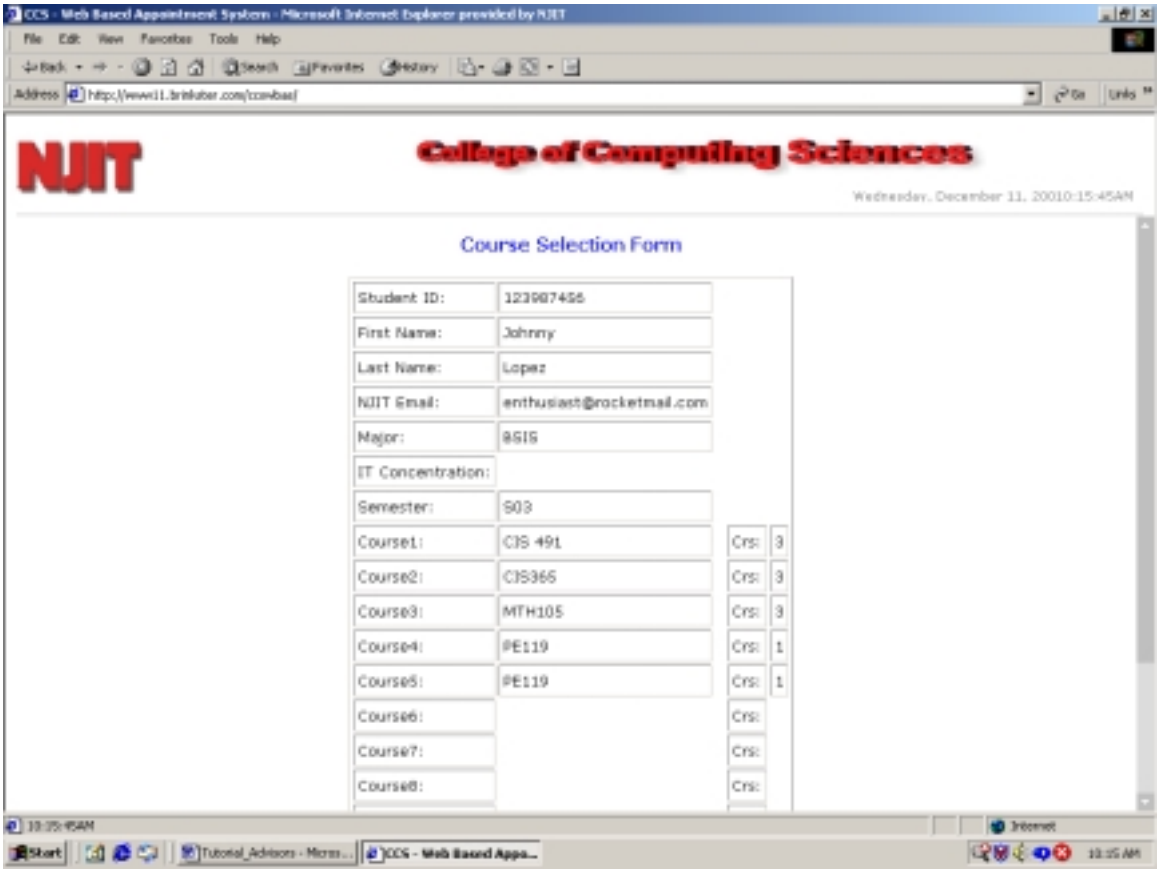
(Figure 6)



The table shown above displays all the appointments an advisor is scheduled for that particular day. Clicking on the student ID column will display the respective student’s Course Selection Form, while clicking on the email column will display an email box to email that particular student.

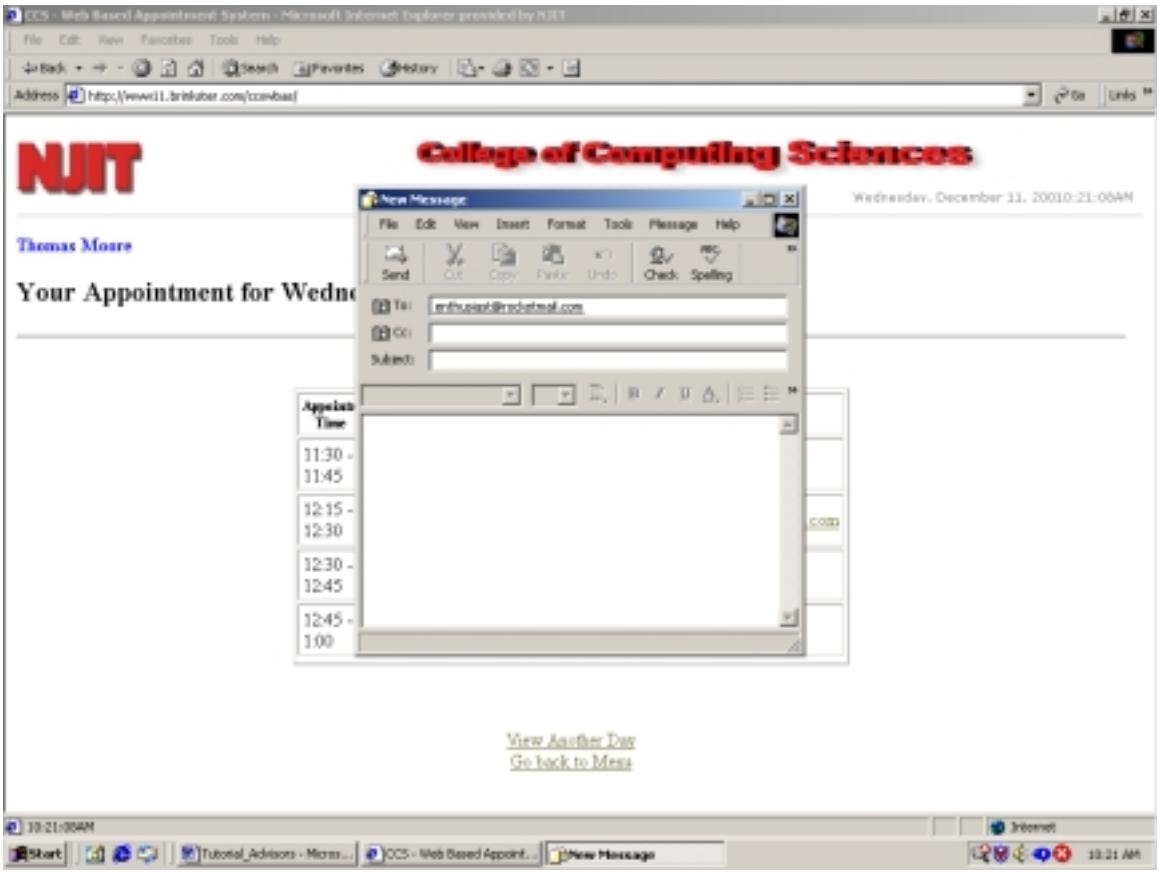
If you wish to view a student's Course Selection Form, then click on that particular student's ID in Figure 6, and you will be directed to his/her Course Selection Form, as shown below:

(Figure 7)



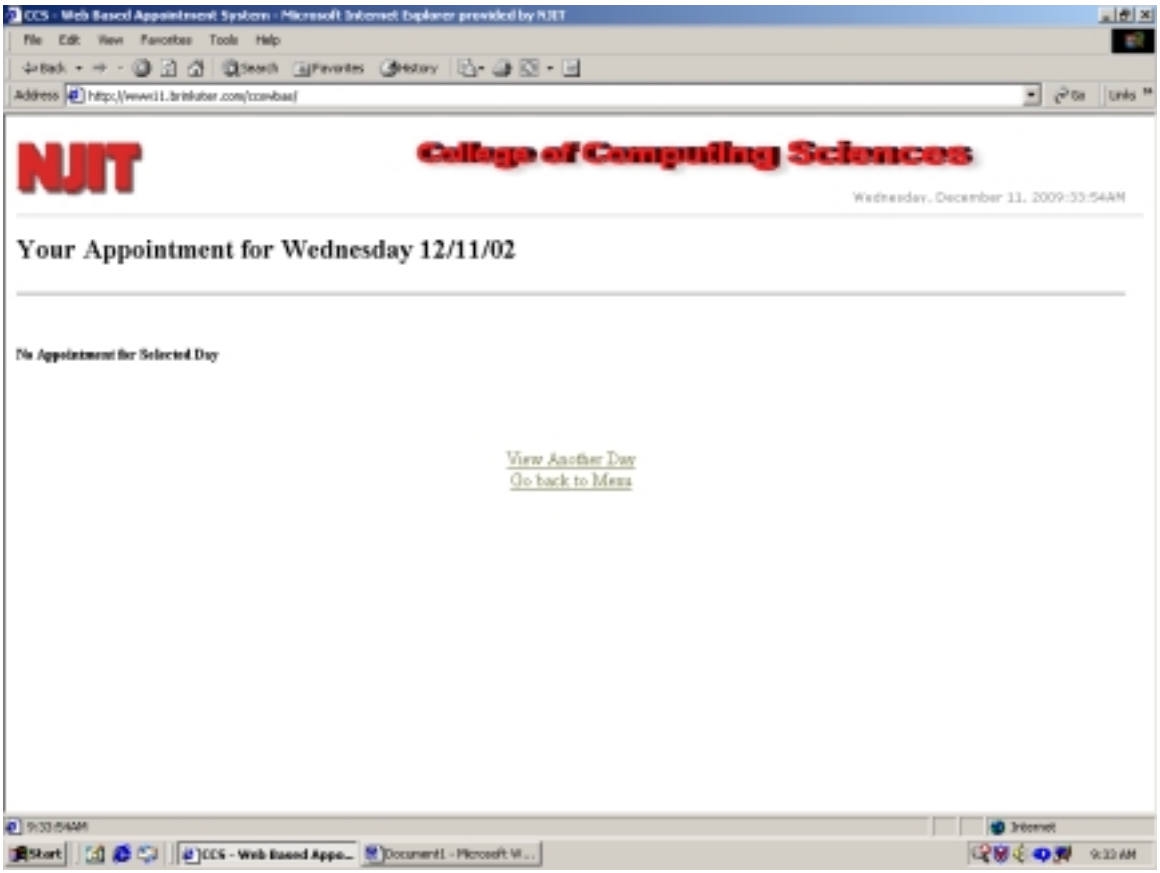
If you wish to email a student who has scheduled an appointment, click on that particular students email address. Clicking on an email address will open up a Microsoft Outlook Mailbox that will enable you to email that particular student, as shown below:

(Figure 8)



If there are not any appointments scheduled for that particular day, you will be asked to view appointments for another day, as shown below:

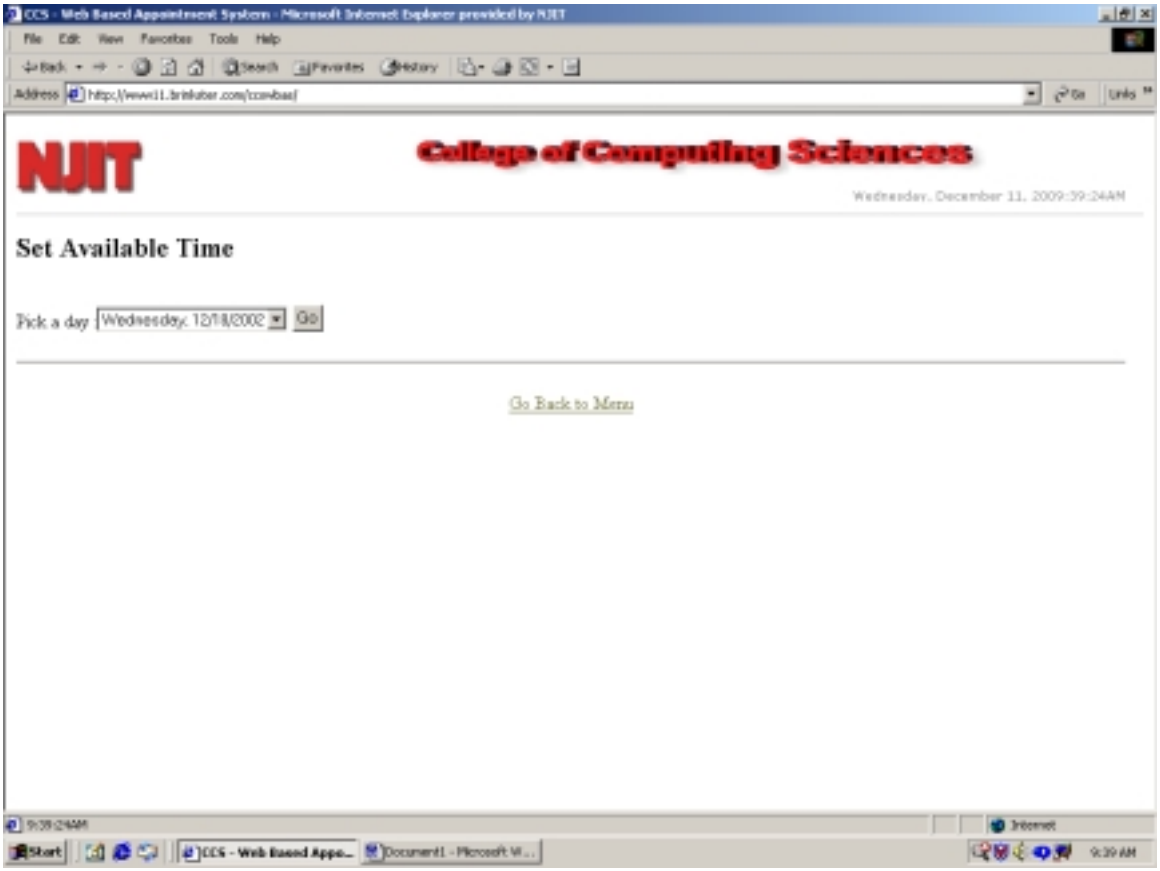
(Figure 9)



Selecting the following, **Go back to Menu**, will direct you back to the options page.

If you would like to set your schedule according to your available and unavailable timeslots, you can make the second selection, labeled ‘Set Schedule’ from the Options page in Figure 4. Clicking on the second selection will direct you to a page that asks you to select a day to set schedule, as shown below:

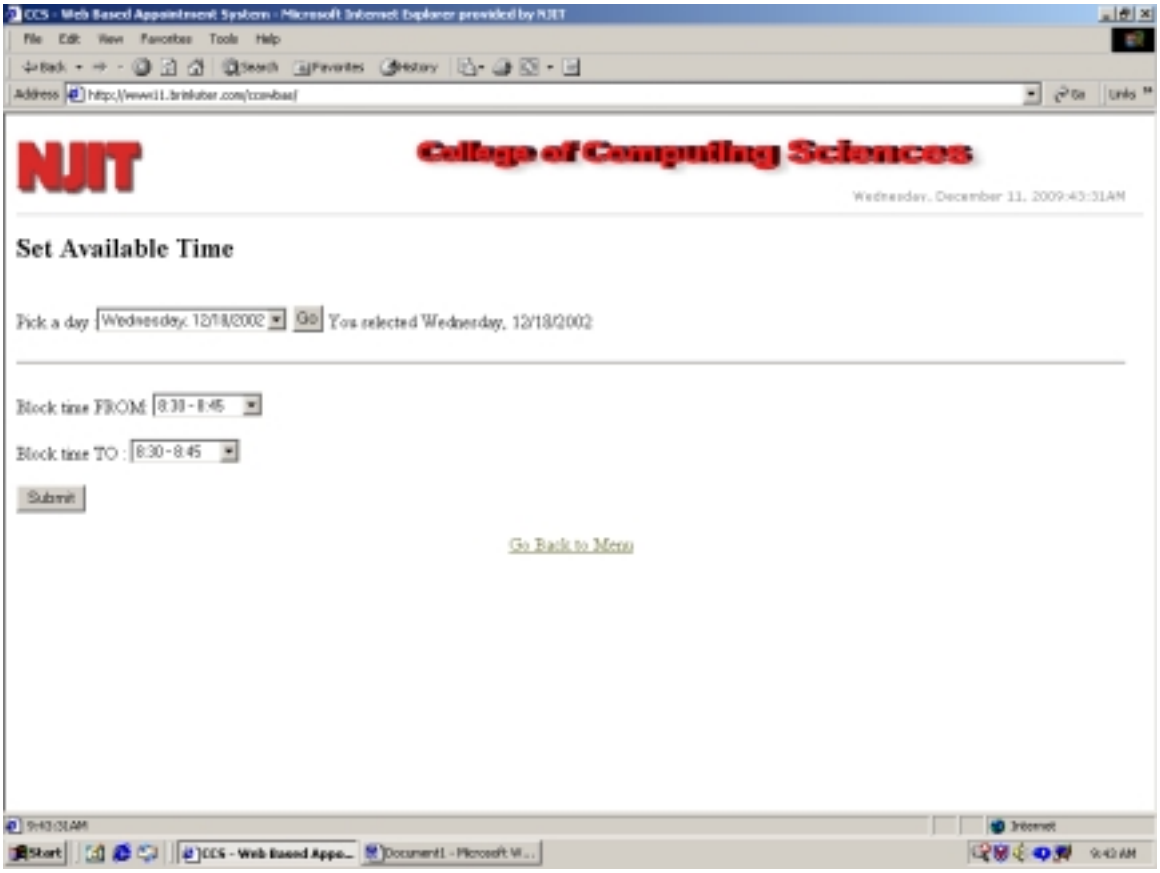
(Figure 10)



The drop-down menu allows you to select a day and set your schedule from next week to next month, leaving you with sufficient time to arrange your available and unavailable hours.

Once you pick a day, you will click on ‘Go’ to block your time on that particular day, leaving that time period blocked for all students to make appointments, as shown below:

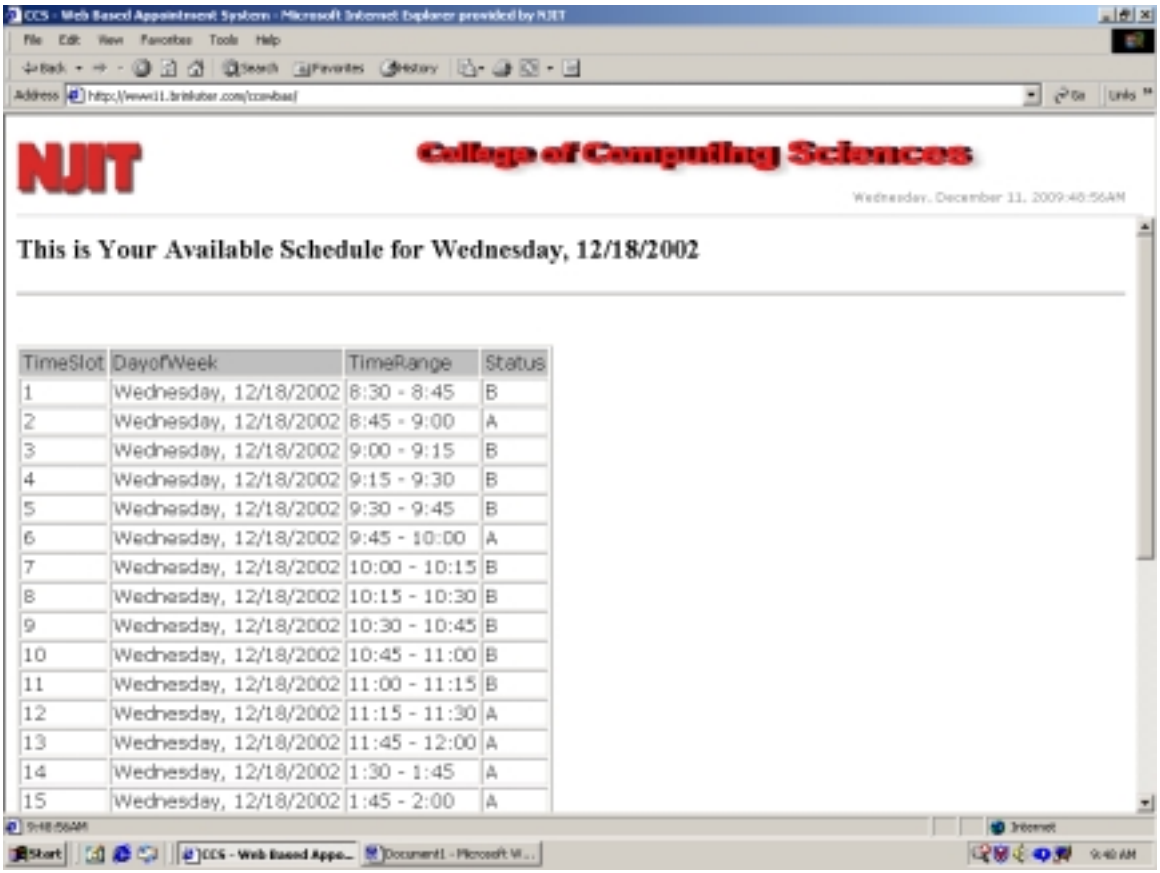
(Figure 11)



In this page, you will be given the option, under the drop-down menu, to block your schedule any time from 830am to 5pm.

Once you make a final selection of your blocked/Unavailable time, click on ‘Submit’ to update your appointment book. Clicking ‘Submit’ will send you to a page where you will be able to view your Available schedule for that particular day, as shown below:

(Figure 12)



TimeSlot	DayofWeek	TimeRange	Status
1	Wednesday, 12/18/2002	8:30 - 8:45	B
2	Wednesday, 12/18/2002	8:45 - 9:00	A
3	Wednesday, 12/18/2002	9:00 - 9:15	B
4	Wednesday, 12/18/2002	9:15 - 9:30	B
5	Wednesday, 12/18/2002	9:30 - 9:45	B
6	Wednesday, 12/18/2002	9:45 - 10:00	A
7	Wednesday, 12/18/2002	10:00 - 10:15	B
8	Wednesday, 12/18/2002	10:15 - 10:30	B
9	Wednesday, 12/18/2002	10:30 - 10:45	B
10	Wednesday, 12/18/2002	10:45 - 11:00	B
11	Wednesday, 12/18/2002	11:00 - 11:15	B
12	Wednesday, 12/18/2002	11:15 - 11:30	A
13	Wednesday, 12/18/2002	11:45 - 12:00	A
14	Wednesday, 12/18/2002	1:30 - 1:45	A
15	Wednesday, 12/18/2002	1:45 - 2:00	A

The table shown above will show you all your timeslots from 830am to 5pm, indicating the status as well, where A represents the time available for students to make appointments and B represents the blocked time that you selected in Figure 9. Students will not be able to make any appointments under the blocked time slots.

6.2 DEVELOPERS' MANUAL

6.2.1 INTRODUCTION

The WBAS (Web-Based Appointment System) project was originally developed by a group of seven students at the New Jersey Institute of Technology to fulfill the requirements of their senior project. The goal is to create a Web based appointment system for two undergraduate advisors and undergraduate students of the College of Computing Sciences. Currently the site is located at http://www11.brinkster.com/ccswbas/ and consists of a database, which holds the advisors' calendar, and the students' scheduled appointments. The site resides in the Brinkster's server at http://www.brinkster.com and is developed under the ASP.Net framework, using VB.Net as the programming language. Other tools used to create the pages are HTML coding, Microsoft Front Page and SQL statements to access and interact with the database. Files are saved under the '.htm' and '.aspx' file extensions.

The system allows students to scheduled appointments with their advisor online. First, they view the advisors' time availability and after selecting a day, students are required to fill out the course selection form. Upon submission of the course selection form appointments are confirmed. The system also allows student to edit their course selection form and cancel appointments once their appointment has been confirmed.

The system advisors to view scheduled appointments by day. This option displays the student's name, ID, and email address. The email and ID are hyperlinks, which allow advisors to email students by clicking on the email address and view the student course selection form by clicking on the Student ID. Advisors are also allowed to blocked time for the upcoming week.

5.2.2 SYSTEM FILES

5.2.2.1 Main Page

Default.htm

Index page for the site which calls the topframe.htm and main.htm files

TopFrame.htm

Pages Header file

Main.htm

Displays the options to use the system as a student or advisor

5.2.2.2 Student Pages

StudentLogin.aspx

Displays students options (Schedule appointment, edit course selection form and cancel appointment) and redirect student according to selected option

CancelConfirmation.aspx

Displayed once student cancel his/her appointment

EditCSF.aspx

Search the database for student records. If the submitted student ID matches a record in the database, it then calls the CSFToEdit.aspx file, else, it displays an error message and allow the student to reenter their student ID

CSFToEdit.aspx

Displays student information stored in the database, and allows students to make changes. Then displays a confirmation page, EditConf.aspx

EditConf.aspx

Confirmation page displays upon successful edit of the course selection form.

GeneralSCal.htm

Displays available timeslots for the week for both advisors

SchApt.aspx

Accepts the student information, student name, last name, id, major, and advisor. It then redirects the student to either KshermanCalStView.aspx or TMooreCalStView.aspx depending on the selected advisor.

KshermanCalStView.aspx

Displays advisor Kathy Sherman calendar. The timeslots and timeslots availabilities are displayed. Available is a link to StSchApt.aspx. Upon student selecting from available times, it saves the time slot id, time availability and timeslots to StSchApt.aspx

TMooreCalStView.aspx

Displays advisor Thomas Moore calendar. The timeslots and timeslots availabilities are displayed. Available is a link to CourseSelForm.aspx. Upon student selecting from available times, it saves the time slot id, time availability and timeslots to CourseSelForm.aspx

CourseSelForm.aspx

The course selection form to be submitted by the students. Stores student information in the database, and calls the UpdateMooreCalendar.aspx or UpdateShermanCalendar.aspx, depending on the selected advisor.

UpdateMooreCalendar.aspx

Using the saved information for timeslot and availability, it changes the time availability in the database from available to unavailable when student appointment is confirm

UpdateShermanCalendar.aspx

Using the saved information for timeslot and availability, it changes the time availability in the database from available to unavailable when student appointment is confirm

SchAptConfirmation.aspx

Confirms student appointment displayed once student submit the course selection form

StCancel.aspx

Searches the database for submitted student ID, and if found calls the "UpdateCancelMooreCalendar.aspx" passing the value of the advisor and time slot id

UpdateCancelMooreCalendar.aspx

Using the previous saved information for advisor and slot idit changes the time availability in the database from Unavailable to available when student cancels appointment

UpdateCancelShermanCalendar.aspx

Using the previous saved information for advisor and slot idit changes the time availability in the database from Unavailable to available when student cancels appointment

TmooreCalStView_multiple.aspx

Will display advisor Moore calendar in multiple weeks. The timeslots and timeslots availabilities are displayed. Available is a link to StSchApt.aspx. Upon student selecting from available times, it saves the time slot id, time availability and timeslots to StSchApt.aspx.

5.2.2.3 Advisor Pages

AdvisorLogin.aspx

It serves as the login page for advisors. It checks and validates for authorized users, and calls and passes user information AdvisorMenu.aspx.

AdvisorMenu.aspx

It is a simple menu for advisor to either view appointments or set daily time availability. It either calls AdvisorViewAppt.aspx or AdvisorSetSch.aspx.

AdvisorSetSched.aspx

Displays list of dates starting one week from present date. It allows advisors to set daily time availability. It also updates the database for the selected time on a specific day.

AdvisorViewAppt.aspx

Diplays the list of dates starting from present date and allows advisor to select a date of appointments. After date is selected, it calls DispayAppt.aspx.

DisplayAppt.aspx

It receives advisor name and selected date. Using those two information, it queries the database and displays the results. Two displayed information are hyperlinked: 1) Student ids allow view of more detailed information; 2) Student email for notification purposes.

DisplayCSF.aspx

Receives student id, queries the database and displayed detailed information which matches the student id.

DisplaySetSch.aspx

After advisor set their daily time availability, it displays the results for viewing and confirmation.

5.2.3 FREQUENTLY USED STATEMENTS

- **Creates the database Connection:** strconnection = "Provider=Microsoft.Jet.OLEDB.4.0; Data Source=" + Server.MapPath("db/wbas.mdb")
- **Selects advisor log id and password and compare if they match:** strSQL = "SELECT * FROM AdvisorLogin, AdvisorInfoTable WHERE AdvisorLogin.AdvisorLogID = '" + Request("AdvLoginID") + "' AND AdvisorPassw = '" + Request("AdvPassword") + "'"
- **Creates the Database Connection:** objConnection = New OledbConnection(strConnection)
- **Creates the command to be executed:** objCommand = New OledbCommand(strSQL, objConnection)
- **Connects to the database:** objConnection.Open()
- **Execute the SQL query:** objDataReader = objCommand.ExecuteReader()
- **Save User information into session variables to be used throughout the linked pages:**

  Session("SessionName") = objDataReader("UserInformation")

  Session("SessionName") = Rquest("UserInformation")
- **Redirects user to the next page:** Response.Redirect("AdvisorMenu.aspx")
- **Closes the database reader:** objDataReader.Close()
- **Closes the database connection:** objConnection.Close()
- **Referencing cascading style sheets so pages will have the same format:** <link rel="STYLESHEET" type="text/css" href="style1.css">. This is used inside the <head> tags.

- **Creates a label to display message send by the server:** <asp:label id="Message" runat="Server" />. This is used inside the <body> tags.
- **Sends the message to the body label be displayed on the screen:** Message.Text = " Could not be logon to the system! Please check your id and password and reenter."
- **Creates a password textbox: <**asp:textbox id="AdvPassword" runat="server" columns=12 textmode="Password" /></td>

6. REFERENCES

Methodologies

- Rising L and Janoff Norman. The Scrum Software Development Process for Small Teams. IEEE Software: July/August 2000
- Larry Trussell, "*Essential Software Development Methodology*",IEEE Software, 1998, Page 357-361.
- http://agilemanifesto.org/principles.html
- http://www.extremeprogramming.org/what.html
- http://www.agilealliance.org/articles/articles/IntroToXpTomKubit.pdf
- http://www.inf.vtt.fi/pdf/publications/2002/P478.pdf

System Structuring

- www.ics.uci.edu/~redmiles/ics121-FQ99/lecture/ten.pdf
- http://www.engr.sjsu.edu/gaojerry/course/ise165/Handouts/Architecture.pdf

Protocol analysis

- http://www.flukenetworks.com/us/LAN/Handheld+Testers/Enterprise+LANMeter/Features/Protocol+Analysis.htm

Control Models

- http://www.ifi.uio.no/in219/verktoy/doc/html/doc/user/mg/mgTOC.html
- Systems Analysis and Design, 2000. Alan Dennis and Barbara Haley Wixom
- Hoffer, Jeffrey A., George, Joey F., Valacich, Joseph S. *Modern Systems Analysis and Design; Third Edition* , Prentice Hall, Upper Saddle River.
- www.er.doe.gov/production/er-60/98sdm3.htm

- http://www.switchit.com/work/making_changes.asp
- http://www.timetrade.com/solutions_education.asp
- http://www.rational.com/products/whitepapers/101057.jsp