BEAR Plus
CS190 Top Level Design
Emily Leventhal: February 25, 2000

1 Document Overview

1.1 Introduction

This document is the top-level design for BEAR Plus. Bear Plus is a programmed aimed to save students time by allowing them to complete tasks online instead of having to physically go to the registrar's office. It will allow students to do the following: pre-register for courses, access grades from past semesters, change what courses they are registered for, access course registration information, use a schedule builder that assists and suggests as they create a possible schedule, and more!

1.2 Top Level Components

This program has been broken into four main parts. THe menu items(i.e. registration, written permission...), the databases, a process module and the HTML builder. This document will detail the way these parts interact. Please look at eldiagram in this directory to view the graphical version of the design.

2 User Interface

The user provides their input through a series of html forms the input is then taken and put through the process module and the output will be in the form of an html page. The user will login and be sent to a menu screen. Once at that screen the user can access their personal infomaton. (note: The Schedule Builder is something that combines the advisor with BEAR. It allows the user to try to arrange their courses and tells them what might be the best fit and suggests things that might be better. Also when they are registering for classes the advisor module can be turned on and make suggestions about the students schedule or be turned off as not to bother a user who is not interested.

2.1 User Input

The following three subsections are menus for the three different users: student, professor and administrator.

2.2 Student Menu Module

The following a places the menu will guide the user too:

- Registration
- Written Permission for a Course
- Declare Concentration
- Schedule Builder
- Transcript
- Credit done at other Schools
- Student Information
- Post a request for a Professor

2.3 Professor Menu Module

The following is the menu for a Professor:

- Read All Student Request, i.e. sign Emily Leventhal's permission for mcm170.
- Change Course Description
- Read Advisse Concentration Forms
- Written Permission Requests
- Credit Requests
- Schedule Builder, the professor might want to look at their schedule as well.
- Enter Student Grades

2.4 Administrator Menu Module

The Following is a menu for a Administrator:

- Add Course
- Enter Grades
- Transcripts, would have ability here to mail one out
- Enter Course Information, i.e. room info or description
- Accept Written Permission
- Accept Declaration of Concentration

2.5 Menu Modules

The following is a list of all the items available from all types of users menu's and how they will interact. These items all go through the HTML builder to display information to the web browser and recieve information from the process module.

- Registration- This module will only be used by students. The student will enter what classes they want to take. When the user finished entering the module will check in the database to make sure the courses are valid. At the same time it will have the ScheduleBuilder do a smart search on the courses. The user can decide whether they want the ScheduleBuilder mode on or off when registering. They can also retrieve a schedule they previously built from the student database to use while registering. When the student is finished the information will be saved in their personal student info and the course info will update its class roster. Whenever new information is displayed on the screen the registration module will do it by speaking to the HTML builder and receive updates through the Process Module.

- Written Permission - A student can request written permission from a professor. The student will enter what course they want the written permission for it will then be sent to the database of the professor whose course it is. When the professor logs in he/she will see requests for written permission and can either grant/deny the permission. In either case a status message will be sent back to the student and if the permission is granted that information will be saved with the students info.

- Transcript - The student can request to see their transcript. To see it online the transcript module will get the information from the database with the students information. If the student wants to request a copy to be mailed their will be a form to fill out and the information will be sent to the administration. When the administration has sent a copy of the transcript the student will receive a message that will be stored in their database.

- Credit at Other Schools - A student will request that a professor grant them equivalent credit for course work done at a different school. Similar to written request, a request will be posted to the professor. They will fill out the information from the student and grant/deny the request. The students information will be updated accordingly adjusting their transcript to show changes. As well the student will receive a message back, this message will be stored in the students database..

- Student Info - The student can review their personal information. This information will be stored under the students name in the database. To get the information this module will just ask the database for it and have the HTML builder display the information.

- Declare Concentration - This module will get the concentration forms from the database. The student will fill out the forms and the process module will send back the information. This module will then save that information in the students information and have a message posted to their advisor that the student is requesting to declare their concentration. The advisor will get the message and to review them the module will

retrieve the information from the students database. When the concentration has been accepted by the professor the student will receive notification which will be stored in the students database

- Enter Course Information - When a professor or and administrator wants to enter the course information. The current information will be taken from the course database. The user updates the information and the information is sent back to the database.

- Enter Student Grades - This module will be used by either a professor or an administrator. The information will be entered and saved in the student database, it will be entered into each students transcript. If the administration decided this feature needs to be extended to include saving the grades for a course for future use this information will also be saved in the course information.

- Schedule Builder - This will allow the user to try building a schedule of their courses. They can build from a one semester schedule to a four year plan. This module will do a smart search and suggest whether the students plan works. The module will get the information from the course database. It will take course reviews and times and evaluate what is best. The user can also do a search to find courses at a specific time that fits their needs. The student can save the schedule they built here and use it while registering. The information will be saved in the student database.

- Message Board - This module will be used by all users. It will get the information from the users message database and display all the messages. If a professor had a request to grant a student written permission the browser would display the request. The professor would click on read request, this would then get the information to display based on what type of request it was and tell the written request module to display a request from: Emily Leventhal for va125. Now it has sent the information to a different module to handle. When the module is finished though it will need to remove that request from the professor's database and post a message to the student this can all be taken care of by the Written Permission module though.

3 HTML Builder

There will be a module that given information it will know how to produce html code from that. This module will interact with the web browser. It will recieve information from each of the items from the above section.

4 Process Module

This will recieve information from the webpage and decide which module needs to deal with it. This module will be sending information to the menu items discussed in section 2.5.

5 Webpage

This module will recieve information to be displayed from the HTML builder and will send information to the process module to be processed.

6 Databases

The following is a list of all the different types of databases and the information they will store.

6.1 Student Information Database

- Personal Information
- Transcript
- Current Schedule
- Schedule Ideas
- Concentration Forms
- Current Requests
- Messages

6.2 Professor Information Database

- Requests
- Current Courses
- Advisees

6.3 Administrator Database

- Transcript Requests

6.4 Course Information Database

- Description
- Semester
- Course Reviews
- Professor
- Time

- Room
- Exam Date
- Webpage
- Course Roster
- Written Permission Grants

7 External Dependencies

8 Group Organization

- **Project Leader/Spirit Leader/Booster:** responsible of general organization and group dynamics. Most imporantly the leader will make sure the group is happy and communicating. Communication is key to getting a program working in this short amount of time. The leader will also delegate work, create a schedule and make sure group members are keeping to the schedule. The group leader will help build general spirit for the group so that we stay interested and on track. Their job will also include reviewing code with members and suggesting improvements and resolving any problems that may arise within the team.
- **Lead Architect/Design Master:** responsible for the overall design of the program. The person will update the design when their are changes and work with the project leader for delegation for roles. When new designs are suggested by team members the design master will review them and decide whether they can be included. At times the design master may need to call a team meeting or a meeting of small groups to go over changes in design. Before coding this person should make sure that everyone in the group understands the overall design and how things flow.
- **Lead Programmer-Co Pilot:**
- **Q+A - Scrutinizer:** responsible for creating test data for the program and testing the programmers code. The scrutinizer will examine code with the programmers to check for bugs. When a piece of code is checked in the scrutinizer will test it and make sure it works for everything the programmer said it would. This way we will be checking for bugs all along. This person is avaible always to go over peoples code with them, but other team members can also help fellow programmers find bugs.
- **UI Master:** responsible for GUI management. This person will make test protypes and show them to users to make sure they are easy to use. The UI Master will be someone with a knowledge of color theory, works well with people, and willing to make adjustments at other peoples suggestions.

- **Editor:** responsible for program documentation. Each programmer will keep an updated documentation of their piece of the program and design decisions they have made. The editor will review these pieces of documentation and give feedback to the team members. Finally the editor will make one united document from all the team members documentation. Also the editor will write the user manual and help sections of the program, because they should understand from the team members documentation how to use each component.

- **toolguy/language lawyer/resident techie:**

9 Possible Group Breakdown

The following is a possible group roster based on information from the cs190 webpage and from some personal knowledge of others in the group. This roster will be adjusted after the first group meeting as people may have changed their course goals since the orginal survey and/or they may want a different position. Following each persons name is part of what they wrote on their questionaire.

- **Scrutinizer**: Neelu Bedi, A responsible team member involved actively in all parts of the project.

- **Design Master**: Melissa Cheng, item-sharpen coding skills, -be a better team member.

- **Editor/Scribe**: Brian Chuck, I would like to play a role in which I would be interacting a lot with the members of the team I am in. I am personable and enjoy working with people.

- **Implementer**: Lisa Cozzens, I'd like to be just another programmer in a group, not a group leader or anything.

- **Lead Programmer**: Joe Fugua, Find out how close I want to work to the code. I have only had the opportunity to work as a programmer and want to see how far into management (and thus away from code) I want to get.

- **Rikard Grafstroem**: I hope to improve my organizational, designing, and problem solving skills in addition to actual coding.

- **Project Leader**: Emily Leventhal, I would like to be a designer and organizer in the project.

- **Implementer**: Bretton Heath-Wlaz, Creation and maintenance of low-level classes and data structures

- **UI Master**: David Yun, I have the most experience in user interfaces and databases, but i am not limited to those areas. ss

10 Development Schedule

2/25: Top-level Design Proposals
2/25-2/33: Sometime in here have an initial group meeting so that team members can meet each other and initally get to know one another without the stress of the program.
3/3: Initial Group Meeting in class - finalize group roster.
3/3-3/10: Two Group meetings to discuss Final Top-Level Design.
3/10: Final Top-level Design
3/10-3/15: Meet to discuss interface proposal.
3/15: Interface Proposals
3/20: Interface Comments
3/24: Final Interface Definition
4/5: Detailed Designs
4/17: Initial System Integration
4/28: Functionality Freeze, Full System Implementation
5/12 9am: Final Documentation/System Submission