Hayden Shelton
CS4962 Spring 2015

Final Project Design - boscc Abstract

The name of my app is boscc. boscc stands for best optimizer for scheduling college courses. The purpose of my application is to make it easier for college students to decide what courses to take and when.

The problem:
Each semester, college students must register for courses. Deciding what courses to take and when to take them can be a frustrating, confusing, and difficult process because there are several things you must consider before enrolling. Your considerations typically include: making sure you're taking the right required courses in a timely order, and making sure your class schedule is compatible with your life schedule. By nature, humans aren't great at thinking about several considerations at once, so when we do our schedules in an non systematic way we make mistakes that cost us time, money, and quality of life.

The solution:
bossc is an iOS application that will make deciding what courses to take easy. boscc will provide students with a visualization of their courses so that they can view the courses they have taken and the courses they need to take. The visualization will be displayed in a way that makes their path to graduation clear and simple. The app will allow the student to select their major and their information will be saved online so that they can view their progress on any iOS device.

The visualization will be a graph, it will show required courses and dependencies (prerequisites) in a tree. After selecting a major, the application will build the graph based on data which has been preloaded into a database. Initially, the graph will consist of a node for each course. At the root level will be courses that require no prerequisites, each ascending level of nodes will represent the next level of required courses. As students complete courses, they can tap on a grey node; the app will then color the node green, marking the course as complete. Each semester, as students decide what courses to take, they can tap on a grey node and a detail view will be displayed for the course, letting them know what times the course is available as well as a course description.

The course information will be pulled off of the University of Utah CIS system and loaded into a database so that the application can access it through a RESTful web api. The app will initially support full course information for CS and EAE undergraduate students at the University of Utah.

Initial Features:

1. Online account management. Students can set up an account online, or through the app. Account management will be done through a simple C# web application. The url to log in will be http://boscc.io. Their login information, major, and starting date will be saved into a database. The database will either be MYSQL or MongoDB.

2. Major progress visualization and interaction. The student can visualize and interact with their course visualization as described above in the abstract. The app will present the data in a polished, visually pleasing and intuitive way. The app will support zooming and scrolling in a natural way.

3. Web api interaction. The app will pull data from the database through a RESTful web api, written in C#.

4. Full major requirements for CS and EAE undergraduate majors at the University of Utah. The web portion will be built so that more majors and more Universities could be added at a later date.

Further Features (as time allows)

1. Ability to optimize courses according to concerns. For example, a student could enter the days and times that they would prefer to be at school and the app would color course nodes that match their desired schedule.

2. O-Auth integration for login.

3. Facebook integration to allow the student to schedule according to classes their friends may be taking.

4. Extended major support at the University of Utah and other Universities.

Time Line and Estimated Work Hours

Server side code for account management, web api, and database integration: I am well versed in writing C# web applications. I would use the .net web api framework to get this going. The web application will likely be hosted on Microsoft Azure or AWS.
8-20 hours

Loading course information into the database: I will enter course dependencies manually and may choose to write a web crawler or investigate CIS api functionality to pull course info and enter it into the database.
4 - 8 hours

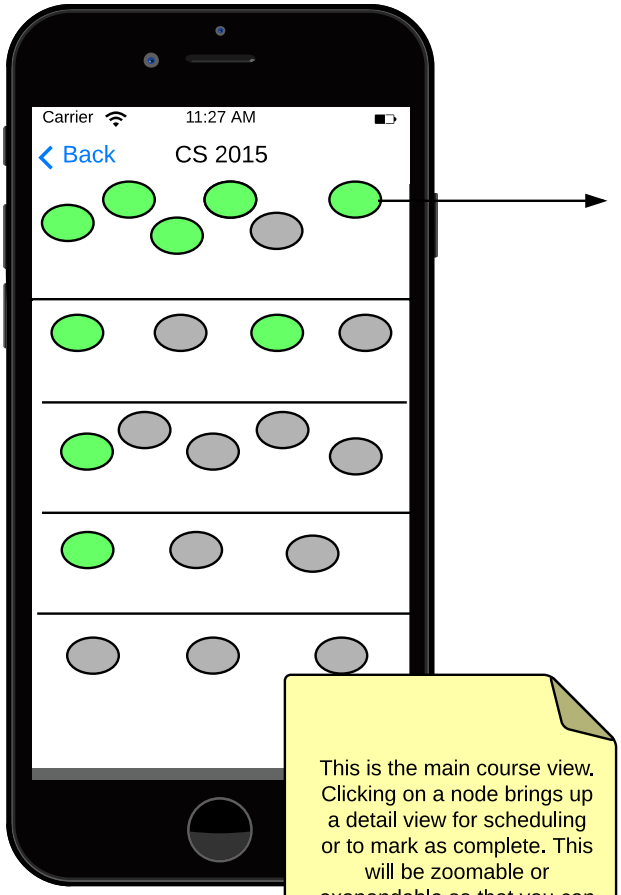
iOS application programming: I will spend the majority of my time coding the iOS application in SWIFT. The application will require a few custom views and possibly some OpenGL components. I will break up the development into these sections:
1. build out initial MVC architecture and data models.
2. build out server controller
3. build custom views and determine best graph representation
4. connect views to data models
20 - 30 hours

Art and graphic design components: I'm not a graphic designer but I think that I can make the splash screen and visual aesthetics pleasing.
4 hours

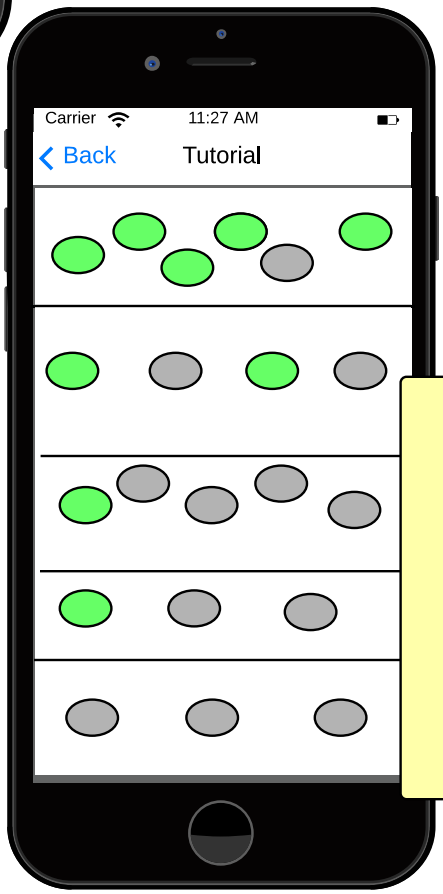
Testing and debugging: This will take place throughout the process. I will, of course, spend additional time testing the application once initial development is complete.
6 - 8 hours (additional)

Total: 40 - 70 hours

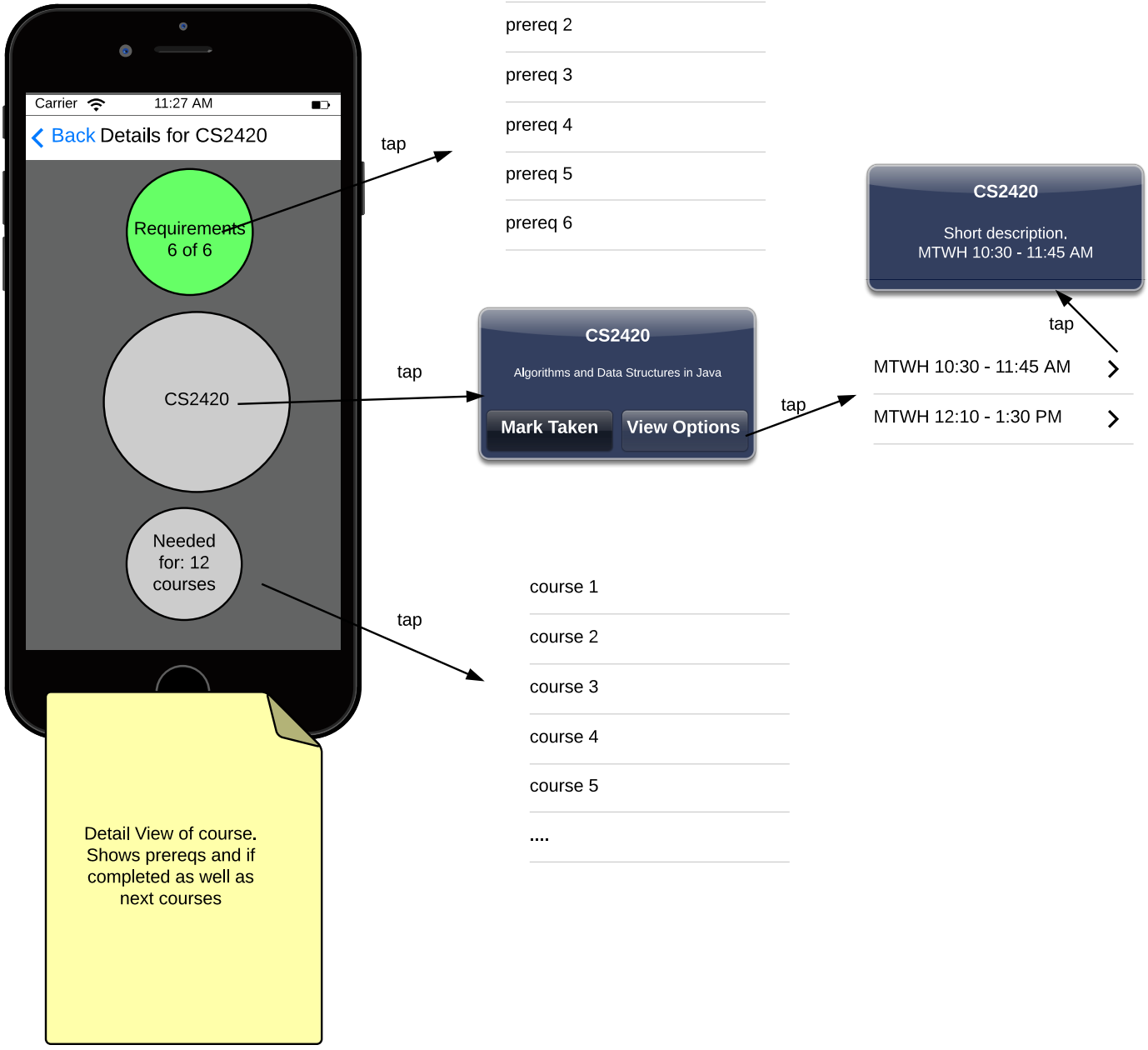
boscc - best optimizer for scheduling college classes



This is the main course view. Clicking on a node brings up a detail view for scheduling or to mark as complete. This will be zoomable or exapandable so that you can actually read the node text. See next page for what happens when you touch a node.



Turorial. Shows you how the graph works and how to use the app.


Carrier
11:27 AM
Back Details for CS2420
Requirements
6 of 6
CS2420
Needed
for: 12
courses
tap
tap
tap
prereq 1
prereq 2
prereq 3
prereq 4
prereq 5
prereq 6
CS2420
Algorithms and Data Structures in Java
Mark Taken
View Options
tap
CS2420
Short description.
MTWH 10:30 - 11:45 AM
tap
MTWH 10:30 - 11:45 AM
MTWH 12:10 - 1:30 PM
course 1
course 2
course 3
course 4
course 5
....
Detail View of course.
Shows prereqs and if
completed as well as
next courses