Criterion B: Record of Tasks

Task number	Planned Outcome	Planned Action	Time estimated	Target Completion Date	Criterion
1	Get my proposed client approved	Get possible client approved from computer science teacher.	10 mins	August 20	A
2	Find solution for client	Consultation with client about possible software solutions	20 mins	August 21	A
3	Document communication with client	Write a transcript for the consultation I had with Mr. Dickinson	30 mins	August 21	A
4	Define success criteria	Consult with client about the criteria that will determine the success of the project.	20 mins	August 21	A
5	Finalize success criteria with client	Clarify any problems with the criteria	15 mins	August 22	A
6	Respond to client's feedback	Ensure success criteria and functionality of the planned application meet the client's needs	10 mins	August 22	A
7	Outline overview of implementation	Show how, in general, the program will be made in terms of programming languages and broad structure	3 hours	August 24	B
8	Research relational databases	Research implementation of a relational database of which will be used in the application	1 hour	August 25	B
9	Make preliminary interface designs	Design the interface in accordance to the success criteria	3 hours	August 25	B
10	Prototype interface design	Set out a preliminary design for the interface of the application	2 hours	August 26	B
11	Prototype design of relational database	Consider then design the relational database to be used for the application	1 hour	August 26	B
12	Consult with client about prototype	Discuss the accuracy of the application to the success criteria	30 mins	August 28	B
13	Make changes to prototype	Make relevant changes to improve the accuracy of the application to the success criteria and additional	1 hour	August 28	B

		client feedback			
14	Research the iBeacon protocol and its API	Plan necessary classes and methods for iBeacon server	1 hour	August 30	C
15	Prototype beacon communication in Swift for IOS	Program a simple bluetooth server-client system using Apple's iBeacon protocol.	2 hours	August 30	C
16	Prototype potential client side communication strategies	Consider how the client will communicate with the server	2 hours	September 2	C
17	Make a method or find a library for sending HTTP requests from swift	FIgure out how HTTP will be used as a communication protocol	30 minutes	September 3	C
18	Begin middleware server production	Program bottle.py server that accepts post requests with the form.	1 hour	September 3	C
19	Develop the handler for requests to the bottle server	Program the class that will handle post requests and direct them to the proper method	30 minutes	September 3	C
20	Develop the request handler for student requests	Add all possible requests made by students	30 minutes	September 4	C
21	Develop the request handler for administrators	Add all possible requests made by administrators	30 minutes	September 4	C
22	Develop the universal request handler	Add login and create account methods	30 minutes	September 4	C
23	Debug the request handlers	Make sure the request handlers work as expected	20 minutes	September 5	C
24	Learn how to use the pyscopg2 library for postgresql	Learn proper use of the methods and how to use SQL queries	15 minutes	September 10	C
25	Add a class to interface with postgresql	Create the PostgresConnector	45 minutes	September 10	C
26	Debug postgresql class with the handlers	Make sure the program can update, read, write and insert.	10 minutes	September 12	C

27	Add a class to interface with the configuration file	Create the ConfigConnector	15 minutes	September 15	C
28	Learn how to use the Xcode GUI design tool	Learn how to use the different tools to create a GUI	20 minutes	September 20	C
29	Create GUI design for login page	Create the GUI for the login page	30 minutes	September 20	C
30	Add login functionality to login page	Add the submit button; with server interfacing.	20 minutes	September 25	C
31	Data validate response from server	Ensure that the password is a string, username is a string, etc.	5 minutes	September 26	C
32	Add security functionality in middleware	Add the hashing functionality in a separate file.	10 minutes	September 27	C
33	Testing of criterion one against testing plan	Make sure that the application so far allos students and admins to login to the application from a single page.	5 minutes	September 28	C
34	Debug ViewController and ensure login functionality	The controller must successfully interface with the middleware server without bugs. Ensure that all potential errors are handled.	10 minutes	September 30	C
35	Learn how to use tables in swift	Make sure I understand the functions and attributes necessary for use tables in swift.	15 minutes	October 10	C
36	Create GUI design for StudentViewController	Using the GUI designer, design the StudentViewController	20 minutes	October 10	C
37	Add beacon detection functionality to the StudentViewController	Use built in libraries as well as custom logic to detect beacons	40 minutes	October 24	C
38	Add ability to update location from the StudentViewController	Use information from the beacons to update location to the server and then to the database	30 minutes	October 24	C
39	Add ability to receive information about student in the StudentViewController	Add labels to display information about the student (name and email). Accomplished with a server request.	10 minutes	October 25	C
40	Test criterion two	Ensure that criterion two is met: we are able to update the location	5 minutes	October 25	C

		of a student relative to beacons			
41	Create GUI for AdminViewController for beacons	Using the GUI designer, design the AdminViewController	20 minutes	November 5	C
42	Add tab in AdminView for viewing and editing beacons	Using the GUI designer, design and add a tab for viewing beacons	20 minutes	November 6	C
43	Add ability to edit, add, and delete beacons	Add the functionality by implementing the necessary logic and various methods	30 minutes	November 6	C
44	Debug beacon view tab	Make sure beacons update consistently, and the user is able to edit, add, and delete beacons	10 minutes	November 7	C
45	Test criteria six and seven	Test if criterion six is met: searchable table of beacons; and that seven is met: the ability to edit, add, and delete beacons	5 minutes	November 8	C
46	Add tab in AdminView for viewing students	Using the GUI designer, design and add a tab for viewing students	20 minutes	November 15	C
47	Create detail view for students for viewing location	Using the GUI designer, design the detail view for viewing student locations. Then implement this programmatically with tables.	1 hour	November 16	C
48	Test criterion four	Make sure criterion four is met: a searchable table of students.	5 minutes	November 16	C
49	Create modal view: fulfilling the map criterion	Create a modal detail view when a cell in the locations table is clicked.	30 minutes	November 20	C
50	Test criterion nine	Make sure criterion nine is met: a map of the school with the zone highlighted is displayed	5 minutes	November 20	C
51	Add settings tab in AdminView	Using the GUI designer, design and add a tab for administrator settings	30 minutes	December 1	C
52	Add logout functionality for AdminView	Add the ability to logout of the application with a logout button	30 minutes	December 1	C
53	Add functionality to add an administrator to the database	Add another button that adds administrators to the database, in a similar way to creating a new	20 minutes	December 5	C

		account			
54	Add functionality to create a new account	Add the create account button on the login page	20 minutes	December 7	C
55	Resume development on StudentView	Move back to development on the student view, and remember where I left off	10 minutes	December 10	C
56	Add function for detecting beacon using CoreLocation	Add a function for detecting beacons using the core location library	40 minutes	December 11	C
57	Debug beacon detection	Debug beacon detection and make sure it works consistently	1 hour	December 11	C
58	Add logout functionality for StudentView	Add the ability to log out of the student view with a logout button at the bottom.	10 minutes	December 13	C
59	Begin development of MacOS beacon emitter	Start working on the MacOS emitter; consider criterion B for plan.	10 minutes	December 20	C
60	Learn how beacon emitters work and their construction	Learn how iBeacon emitters work, and how they are programmed.	40 minutes	December 21	C
61	Learn how bluetooth packets are constructed	Learn how the bluetooth packets are structured and how they can be structured in swift.	20 minutes	December 22	C
62	Test beacon emitter function	Test and debug beacon emitter and ensure that it is consistently detected by the student IOS app.	1 hour	January 2	C
63	Add networking to display beacons in a list	Network the application using the Alamofire library and display beacons in a list.	30 minutes	January 3	C
64	Add the ability to select a beacon from the list to emit	Make each of the items in the list selectable	20 minutes	January 3	C
65	Debug MacOS application	Debug and test the application to ensure that it works properly	40 minutes	January 5	C
66	Test criterion eight	Make sure criterion eight is met: Have a macos application that is able to function as a emitter.	5 minutes	January 5	C
67	Debug the system together	Debug the system all together, make sure all parts interface	30 minutes	January 10	C

		correctly.			
68	Test criterion three	Test criterion three: make sure location is updated and is viewable by the administrators	5 minutes	January 10	C
69	Debug the system together more	Debugging system	30 minutes	January 11	
70	Debug aggregate system more	Debugging system	30 minutes	January 12	C
71	Test criterion three again	Test criterion three again.	5 minutes	January 13	C
72	Documentation and commenting of code	Document the code with comments explaining how and what it does.	30 minutes	January 14	C
73	Documentation and commenting of code	Continue commenting.	30 minutes	January 15	C
74	Video of developed application	Create the video of the functional application.	3 hours	January 17	D
75	Installation of program on client's phone	Install the application on the client's phone.	20 minutes	January 22	E
76	Installation of program on student's phones	Install the application on the student's phones	30 minutes	January 22	E
77	Installation of program on computers for use as beacon	Installation of the application on macs throughout the school, adding beacons for all the relevant room numbers.	30 minutes	January 22	E
78	Installation of server on a school-wide server	Installation of the python server and database that is accessible throughout the school	20 minutes	January 22	E
79	Evaluation of application with client	Sit down with client and work through the success criteria. Also, find areas for improvement in the application.	20 minutes	January 23	E
80	Final handover of software	The development and documentation of the software have finished.	10 minutes	January 24	N/A