**Milestones**:

March 15

* Design document complete

March 21 (By End of Spring Break)

* Learn how to use proper source code management -> GIT
  + <https://www.cs.princeton.edu/courses/archive/spring16/cos333/lectures/12.pdf>
  + <https://www.youtube.com/watch?v=r63f51ce84A&index=84&list=WL>
* Learn JavaScript, HTML, CSS and (Python anyone?)
* Depending on design frameworks: learn Meteor (highly recommended), Bootstrap, ReactJS or Heroku
  + <https://www.meteor.com/why-meteor/features>
* Create website for timeline
* Have elevator speech planned starting March 21
* Create GITHUB repository

March 26

* App UI flow (storyboard) planned and sketched out.
* If not using meteor, set up **local database** and populate with data
* Design the **Log-in Page**
  + Allow people to learn about the service and consider whether they want to sign up.
  + Allow members who already signed up to log on
  + Allow people who want to sign up to create an account.

Week of March 28

* Design the homepage.
  + **calendar** or **queue** for students for each specific job or just for Lab TA’s.
* Create **student’s** **profile** to view all shifts.
* **Project Prototype completed on week of March 28**

April 4

* Implement **search** **functionality** to look for current shifts?
* Allow students to **create, delete and change** a shift(s).
* Create an **announcement header** to post recent updates on the top of website.

Week of April 18

* **Alpha – week of April 18**
* Have users to **test**

Week of April 25

* **Beta**

May 2 - 5

* **Demo**

Dean’s Date

* **Final Submission**

NOTE THAT WE WILL INCLUDE MORE FEATURES FROM ALPHA TO FINAL.

**Risks and Open Issues:**

Learning new languages and tools for web development: Everyone does not know JavaScript, HTML, CSS, GITHUB and (insert frameworks) very well.

One of the open issues is which audience we plan to target as well as the scope of the project. Are we going to make this only for Lab Ta’s in mind or for dining crew as well?

Another issue is how we will make the app visually as well as functionally smooth for the audience? Do we implement the homepage as a calendar or as a queue of shifts? The design of accessing and changing shifts will be essential.

Meteor and other tools may require monetary upgrade if there is more data or demand than expected.