Link: Milestone 2 - Requirements Checklist

Project Features List: View calendar events on app, view friends' calendar events, add schedules to app, see common free time between you and your friends, add friends, user authentication, user onboarding

- ❖ #1 View Calendar Events
  - ➢ The app should allow users to view an itemized list of events in their calendar
  - ➢ They should be able to view this information on the mobile application
- ❖ #2 Add Schedules to Link
  - ➢ The app should be able to add schedules to a temporary storage folder.
  - ➢ Using mobile phones, they should be able to add a schedule based on a predetermined format that would be universal for all users of the application
- ❖ #3 Add Friends
  - ➢ The app should be able to display a comparison between you and your friends calendars.
  - ➢ Using an algorithm to compare, the common time could be highlighted for example on the mobile application to show the user when to expect the other person to be free
- ❖ #4 User Creation
  - ➢ The user needs to be able to identify current users, and allow them to log into their personal schedules
  - ➢ Using username and password, maybe even an email, the user should be able to sign into their account from the mobile application login page
- ❖ #5 User Authentication
  - ➢ The user needs to be able to create a new account, and save unique user/password information for authentication purposes.
  - ➢ Through the app itself, it should ask for a certain number of input parameters like email, username, name, college even maybe and other information that may make searching other users easier as well.
- ❖ #6 View Common Free Time between you and your friends
  - ➢ The app needs to allow for the ability to add friends to compare schedules to
  - ➢ Friends should be able to be searched on the app and then "followed" to gain access to their schedules
  - ➢ People that are not your friends on Link will not be able to view your schedule either
- ❖ #7 Aesthetics/Bug Fixing
  - ➢ The app should be pleasing to the eye and well formatted to make navigation simple and ease of use elegant.
  - ➢ The style should be moderately minimalist like many popular applications now
  - ➢ When switching between displays, transitions should be smooth and have animation instead of being a hard cut like opening a link or an image
  - ➢ No glitches when overloaded with input or extraneous cases

Functional and Nonfunctional Requirements

Feature	Functional	Nonfunctional
View Calendar Events	- mark calendar with events and classes that are listed in your schedule	- easily distinguish between various types of events, like classes vs office hours vs recitations
Add Schedules to Link	- Be able to upload a document or some other format of a schedule to app and allow users to view their schedules on the app	- It would be nice to have some type of universal schedule format that would allow you to upload your class information to Link
View common times between friends	- Be able to compare schedules with other people to see when you can meet up with them	- Different color schedules for both you and your friends would distinguish between the two - Highlighting the time when both parties are free would be useful
Add friends	- Be able to search for friends on Link - Once you send a friend request and the request is accepted, then both can view each other's schedules	- searching for criteria like school or state would be helpful to find classmates or friends - maybe some friends are in two different time zones so have to account for that as well
User authentication	- simple username and password authentication for users logging in to their phones for the first time	- Using 2 factor authentication is becoming increasingly popular as a form of extra security, so using that to further authenticate users
User creation and onboarding	- Process where user can fill out the form on their app to create their account with all the pertinent information	- Getting some extra information like college or high school would allow the search to function better and match friends easier

We plan on all working together in the design and evaluation stages of each sprint, and then dividing workload amongst us during the implementation stage. The workload will be divided fairly evenly, but it may skew towards skill level in a certain area. We are currently not familiar enough with each others strengths to split up the workload specifically, which is why we allocated extra time to the first sprint so that we can familiarize ourselves with the process.