**Legend:**

**MVP requirements**
**High priority requirements**
**Low priority requirements**

**SCU Schedule Helper User Requirements**

Ability to view quantitative data from RateMyProfessor and SCU course evaluations, within Workday and elsewhere (3.x)

Ability to decide whether to control access over personal data (2.1.x)

Ability to add friends, manage their social network, and view their enrollment in courses (2.2.x, 2.3.x)

Ability to express certain preferences and sort courses by most preferences matched (1.x)

Ability to filter courses by time range (4.1)

Ability to export a schedule (4.2)

Ability to see RateMyProfessor comments and tags from students (3.2.2, 3.2.3)

Ability to generate a short-term schedule—upcoming quarter only (5.1, 5.3)

Ability to generate a long-term schedule (5.2, 5.3)

Ability to save a generated schedule (5.4)

**SCU Schedule Helper Functional System Requirements**

1. Preferences
    - 1.1. The user should be able to input preferences
        - 1.1.1. The user should be able to enter time preferences for courses
        - 1.1.2. The user should be able to enter preferred professor difficulty
    - 1.2. Courses that match preferences should be visually prominent
        - 1.2.1. Courses that match preferences should be sorted to top
        - 1.2.2. Courses should be color coded according to preferences matched
2. Social Network
    - 2.1. The user should be able to sign in with their scu.edu account
        - 2.1.1. User must be able to opt-in or opt-out of sharing any sensitive information, grant access to certain—likely OAuth—scopes (name, profile photo, course history)

2.1.2. User should also be able to choose to share course history data with friends
2.1.3. User should be able to easily or automatically input all of their course history into the platform
2.1.4. Courses should be automatically tracked over time, if course data sharing is enabled by user
2.1.5. Data sharing preferences must be able to be updated at any time

2.2. Users can manage a network of friends on the platform
2.2.1. Users should be able to look up friends by name or email
2.2.2. Users should be able to add friends
2.2.3. Users should be able to remove friends
2.2.4. Users should be able to invite friends by email

2.3. Information regarding friends of users should be displayed
2.3.1. Friends who are interested in the class should be visible
2.3.2. Friends currently enrolled in the class should be visible
2.3.3. Friends who have previously been enrolled should be visible

3. Access to third party data and SCU course evaluations data

3.1. Information from course evaluations should be displayed in Workday, when possible (Done)
3.1.1. Professor overall rating should be displayed
3.1.2. Professor difficulty rating should be displayed
3.1.3. Workload expectations should be displayed

3.2. Information from RateMyProfessor should be displayed in Workday, when possible (Done)
3.2.1. Professor rating should be displayed
3.2.2. Professor tags should be displayed
3.2.3. Comments should be displayed, when possible
3.2.4. Comments should be sorted by relevance to the course listing

3.3. Information from RateMyProfessor and SCU Course Evaluations should be generally available on the platform (not just within the page embeddings)
3.3.1. Ability to query a professor, and see RMP rating, as well as course evaluation rating data
3.3.1.1. Ability to see which quarters they have historically taught a class in
3.3.2. Ability to query a course, and see potential professors, as well as their ratings
3.3.2.1. Ability to see a “best recommendation” for a professor
3.3.2.2. Ability to see in which quarters a course is taught
3.3.2.3. Ability to see which friends have taken a course (with professor granularity)

4. Other Workday Additions

4.1. Add an ability to filter classes by time range, within course registration page
4.2. Add an ability to export class schedule to Google Calendar

5. Schedule generation

5.1. Ability to intelligently generate a schedule for the upcoming quarter, based on a multitude of factors: initial preferences entered, professor ratings and course evaluation data, major/minor requirements, past course history, courses offered shown on Workday, text input from user
5.2. Ability to intelligently generate a high-level year-by-year plan, subject to change but providing a general direction, based on same factors as above, but excluding courses offered on Workday (instead, factor the historical data to guess when a certain course would be offered), and also exclude specific time preferences for courses
5.3. Ability to generate back-up plans, for both types of schedules
5.4. Ability to save and come back to generated schedules later

**SCU Schedule Helper Non-Functional System Requirements**

Detailed expectations with testable constraints are given below, but subject to change based on empirical trials.

1. Management
   1.1. Cost of maintaining project-related infrastructure (i.e. cloud providers, 3rd-party services, *not* labor costs) should be no more than $5/month
   1.2. Project should be easy to maintain, and should generally require no more than small environment variable changes, or a few (3-5) hours of engineering per year
   1.3. New engineer onboarding should take between 1-2 weeks
2. Performance
   2.1. Getting the statistics for each class listing (i.e. RMP data, course eval data) on Workday should take no longer than 500ms to fetch in the 99th percentile
   2.2. The initial download of the course evals data should take no longer than 1 minute in the 99th percentile
   2.3. All other API operations hosted on our end (adding/removing friends, etc), should take no longer than 300 ms in the 99th percentile
   2.4. The application should not add significant latency / performance drops to the Workday course browsing experience
3. Reliability
   3.1. The deployed service should have a very low rate of internal API errors (i.e. error code 5XX), less than 1%
4. Other
   4.1. Generally, the extension should make the course registration experience easier, not overwhelm students with too much data. Any user-facing interface should not be confusing to use. Learning to use the platform should occur naturally and obviously, in less than minutes or seconds
   4.2. First-time set up with the platform should take no more than 3 minutes in the 99th percentile