CS 194 Project Proposal: Schedulr

Hope Casey-Allen
`hcaseyal@stanford.edu`

Charles Mulemi
`cmulemi@stanford.edu`

Rebecca Wang
`rwang7@stanford.edu`

Wen Zhang
`zhangwen@stanford.edu`

Winter 2017

1 Description

Schedulr is an advanced calendar and organizer application, specifically catering to Stanford University students. It provides students with an efficient, friendly tool to manage their schedule upon input of a list of classes. The application will manage lecture, section, office hour, and exam times as well as locations. In addition, it allows students to both manually enter in events and automatically import relevant University events, such as career fairs. Lastly, Schedulr facilitates more effective life management and work-life balance by suggesting times to be set aside for meals, exercise, showers, and catching up with friends through helpful notifications. Users will have the option to opt in and out of specific features to fit their personal needs.

We envision the general feel of the calendar to look like Figure 1, with toolbars and settings on the side (not pictured).

2 Need for Product

To the best of our knowledge, there is currently no such scheduling application or tool that is tailored specifically for Stanford students. SimpleEnroll visually displays blocks of class times, but it does not let you customize your weekly schedule (for example, adding section or office hour times). It also does not have a printer-friendly version—you can see class information only by clicking and expanding one block at a time, and SimpleEnroll is only available while you are logged into AXESS. This is not helpful for Stanford students who want to glance at their weekly class schedules when deciding when to meet with friends, schedule an extracurricular, etc.

Current alternatives to Schedulr include using existing calendar applications (such as Google Calendar) or simply memorizing class schedules. The former involves manually inputting information that our tool could automatically generate, and the latter is prone to human error. By automatically getting infor-

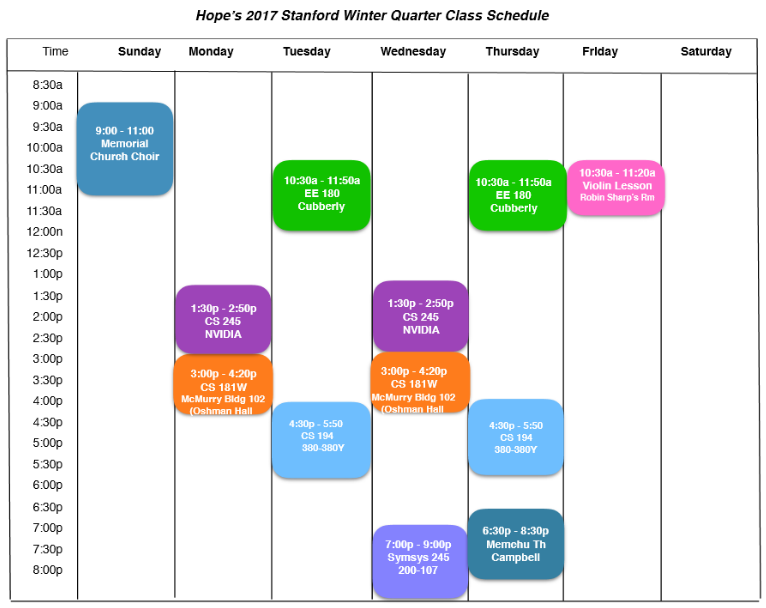
Hope's 2017 Stanford Winter Quarter Class Schedule
Time
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
8:30a
9:00a
9:30a
10:00a
10:30a
11:00a
11:30a
12:00n
12:30p
1:00p
1:30p
2:00p
2:30p
3:00p
3:30p
4:00p
4:30p
5:00p
5:30p
6:00p
6:30p
7:00p
7:30p
8:00p
9:00 - 11:00 Memorial Church Choir
10:30a - 11:50a EE 180 Cubberly
10:30a - 11:50a EE 180 Cubberly
10:30a - 11:20a Violin Lesson Robin Sharp's Rm
1:30p - 2:50p CS 245 NVIDIA
1:30p - 2:50p CS 245 NVIDIA
3:00p - 4:20p CS 181W McMurry Bldg 102 (Oshman Hall
3:00p - 4:20p CS 181W McMurry Bldg 102 (Oshman Hall
4:30p - 5:50 CS 194 380-380Y
4:30p - 5:50 CS 194 380-380Y
6:30p - 8:30p Memchu Th Campbell
7:00p - 9:00p Symsys 245 200-107

Figure 1: General feel of the calendar.

mation from ExploreCourses, the Stanford registrar, and the Stanford calendar, our tool would reduce time spent looking up class information, final exam times, and Stanford holidays/breaks. Moreover, we aim to provide a cleaner, simpler look with functionality centered around classes.

Furthermore, none of these existing tools provide smart scheduling of tasks such as exercise, showering, meals, and sleep, as well as suggestions to catch up with certain friends on campus. Automatic scheduling of these tasks will ease the burden of logistical scheduling for busy students, who may otherwise neglect their health. For example, many students have times in the quarter where they stay up too late doing an assignment and forget they have a 9am class, or put off exercising in favor of cramming in an assignment last-minute. Other students may need to snack frequently throughout the day due to health concerns such as hypoglycemia or "hanger" (a common phenomenon where people get irritable when meals are spaced too far apart[1]). These students may forget to eat frequently or bring a snack to class due to their busy schedules or changes in dining hall hours. These are real concerns, especially for students just transitioning to college and adjusting to the sudden absence of stricter parental influence. Our app's gentle reminders to will help keep students on track and healthy during the busiest parts of the quarter.

[1] `http://www.cnn.com/2015/07/20/health/science-behind-being-hangry/`.

3 Potential Audience

The primary target audience for Schedulr will be the Stanford University student body. An additional use case is Stanford faculty members, who could have all the class times and office hours across all classes they are teaching automatically added to their schedules. The technical sophistication of Stanford students varies, but Schedulr aims to provide a simple, intuitive interface that is accessible to everyone.

4 Discussion of Competing Products

The most obvious competing product in this category would be Google Calendar. Similar products include Apple's Calendar, Microsoft Calendar, and other calendar applications. Our competitive edge would be the customizations made specifically for Stanford students, as mentioned in the "Need for Product" section, as well as smart scheduling of health-related aspects.

5 High-level Technical Design

We will first implement Schedulr as a web application so that it can be used on a wide variety of platforms (computers, cellphones, and any other device with a browser). Although the frontend we develop will likely be web-specific, our backend can expose API calls that can be accessed by not only web endpoints, but also future Android/iOS mobile endpoints if we expand Schedulr to mobile.

5.1 Backend

The Schedulr backend will be implemented using **Django**[2], a high-level Python web framework, in conjunction with the **PostgreSQL**[3] database. We will rely on the Django framework for the following standard web features:

User authentication Django manages account information in the database and cookie-based user sessions; it can also handle third-party authentication (e.g. Google) with the help of plugins. We are not currently considering integration with Stanford web authentication because (a) such integration doesn't give us much information about the user, and (b) we don't want a Schedulr account to become inaccessible after a Stanford student graduates.

Web security A well-written Django website is protected from SQL injections, cross site request forgery (CSRF) attacks, cross site scripting (XSS) attacks, etc.

[2]`https://www.djangoproject.com/`.
[3]`https://www.postgresql.org/`.

Admin interface Django's out-of-the-box admin interface will help manually manage website data (e.g. account information, schedules), which is not only helpful for development and debugging, but also useful for administrators when the site is in production.

The Schedulr-specific features will be implemented as follows:

Importing course schedules It does not seem feasible that Stanford would provide Schedulr with the user's course enrollment data because of privacy concerns. Schedulr will therefore ask the user to manually enter a list of classes they're enrolled in, from which it will generate the user's class schedule. To map a course number to its lecture/section schedules, Schedulr will consult its Stanford course schedule cache, which is scraped from the ExploreCourses website. We will use the **Beautiful Soup**[4] Python library to parse HTML and extract course schedules. The "About" page of ExploreCourses claims to expose a Course Data API, which would ease course data retrieval if we can access it.

Importing other course info Course data other than lecture/section times are more difficult to retrieve; to this end, we will adopt heuristics and manual methods:

- Final exam times, if not retrievable from ExploreCourses, can be generated from manually-programmed patterns (e.g. classes starting at 4:30pm will have their exams on Friday) according to the registrar's website; these rules are too complicated to machine-parse.
- Mid-term exam times and office hours will be scraped from course websites, with simple keyword matching (using the Beautiful Soup library) and date-time parsing (using the dateutil[5] library).
- Information that cannot be easily scraped will be entered manually by the user.

Smart scheduling Smart scheduling: Each time new information is added (e.g. an exercise slot is entered), Schedulr will schedule/reschedule related events if necessary (e.g. move shower time to after the exercise). Since a user is not expected to have too many events, it is feasible to do scheduling in a brute-force manner (e.g. enumerate and process all showering events for the user). The scheduling logic is specific to the type of information added.

The backend will expose a RESTful API which the frontend can access.

5.2 Frontend

The Schedulr frontend will be implemented using JavaScript with **AJAX** technologies. Standard web application features, such as user login, will be im-

[4]`https://www.crummy.com/software/BeautifulSoup/`.
[5]`https://pypi.python.org/pypi/python-dateutil`.

plemented using **Bootstrap**[6], which allows creating clean user interfaces in a mobile-friendly fashion.

For the calendar UI, we will use **FullCalendar**[7], an open-source JavaScript event calendar built with jQuery. Aside from displaying events in a Google calendar-like style, FullCalendar supports custom event logic (e.g. clicking on events) through its event hooks. We will use these event handlers to send AJAX requests to the server using **jQuery**[8] so that users can interact with the calendar in a persistent manner (e.g. create/modify events).

5.3 Deployment

Schedulr will be deployed on **Heroku**[9], a cloud platform for deploying and running web applications. The deployment process should be relatively effortless, as Heroku uses PostgreSQL natively and has good support for Django without need for much manual configuration. We will consider using AWS S3[10], a cloud datastore, to serve static assets if need be.

6 Resource Requirements

The development of Schedulr, a standard web application, does not require many special resources. Having access to the ExploreCourses Course Data API would be a plus, as explained in the Technical Design section, although not strictly required; we have emailed the point of contact for more information. If we end up deploying Schedulr on a large scale, we would need machines for web and database hosting, although that seems unnecessary for the purposes of CS 194.

7 Potential Approaches

Other potential approaches include developing our product for mobile first, rather than web. A mobile app would be more accessible on smartphones and can more easily push notifications. However, it is difficult to develop mobile apps in a platform-independent manner, e.g. an Android app does not run on iOS or Windows. Since making the app cross-platform would require significant engineering efforts beyond the timeline for this class, we would only develop our product for one platform, and this would limit our potential user base. Instead, we decided to first go with a web app, which can be accessed on any platform with a browser. We may develop a mobile version of Schedulr once it has been widely adopted.

[6] `http://getbootstrap.com/`.
[7] `https://fullcalendar.io/`.
[8] `https://jquery.com/`.
[9] `https://www.heroku.com/`.
[10] `https://aws.amazon.com/s3/`.

Our product could potentially be integrated with fitness tracking apps. However, we decided against this because this is not core to the intended user experience—indeed, it might distract from it. We also decided against integrating these functionalities into an already established Calendar like Google or Microsoft Calendar. This is because we intend a different aesthetic look to our product.

8 Assessment of Risks

There are very few associated risks to using Schedulr. One disclaimer is that information about classes may not be fully up to date on ExploreCourses, and the user should make sure to double check their course websites to resolve any discrepancies, especially for crucial things like exams. There are many potential options to reduce this risk. Upon generating crucial class information, we can display a message to remind the user to double check with their class websites. Alternatively, we can display a small screenshot of where the information was located to aid the user in identifying discrepancies. We can also occasionally send double-check notifications the week before key events, such as exams.

9 Next Steps

On the product management side, we have proposed numerous features for Schedulr, e.g. course information integration, smart scheduling, and reminders. We will prioritize them and design/implement them one by one.

On the technical side, we will set up a skeleton Django project and set up a development environment for each team member. We will then sketch out the interfaces for the most important features, e.g. AJAX endpoints (interface between frontend and backend) and database schemas (interface between backend and database), and implement the barebones so that each team member can work independently assuming the interfaces.