



Team Quartermen

Amit Narang, Clara Mohri, Damian Wasilewicz, and Jared Asch

El Barto: Bell Schedule 2.0

Introduction	1
Site Map	2
Component Map	3
Database Schema	4
Framework	6
Roles	6

Introduction

For our final project, our group decided to try to create a version of the Bert Bell Schedule fit for the modern era. We will add new features like customizable templates and patch bugs like flooring the amount of time remaining, but we will also keep the simplicity and usefulness that has ingrained it into our Stuyvesant society. In reference to the “flooring the amount of time remaining”- right now, when a user checks the schedule, the amount of minutes remaining will be rounded up to the nearest minute. This means that if 4 minutes and 2 seconds are remaining, the schedule will indicate that 5 min are left. This is an inconvenience to teacher planning, and is something we will remedy by rounding down the time remaining.

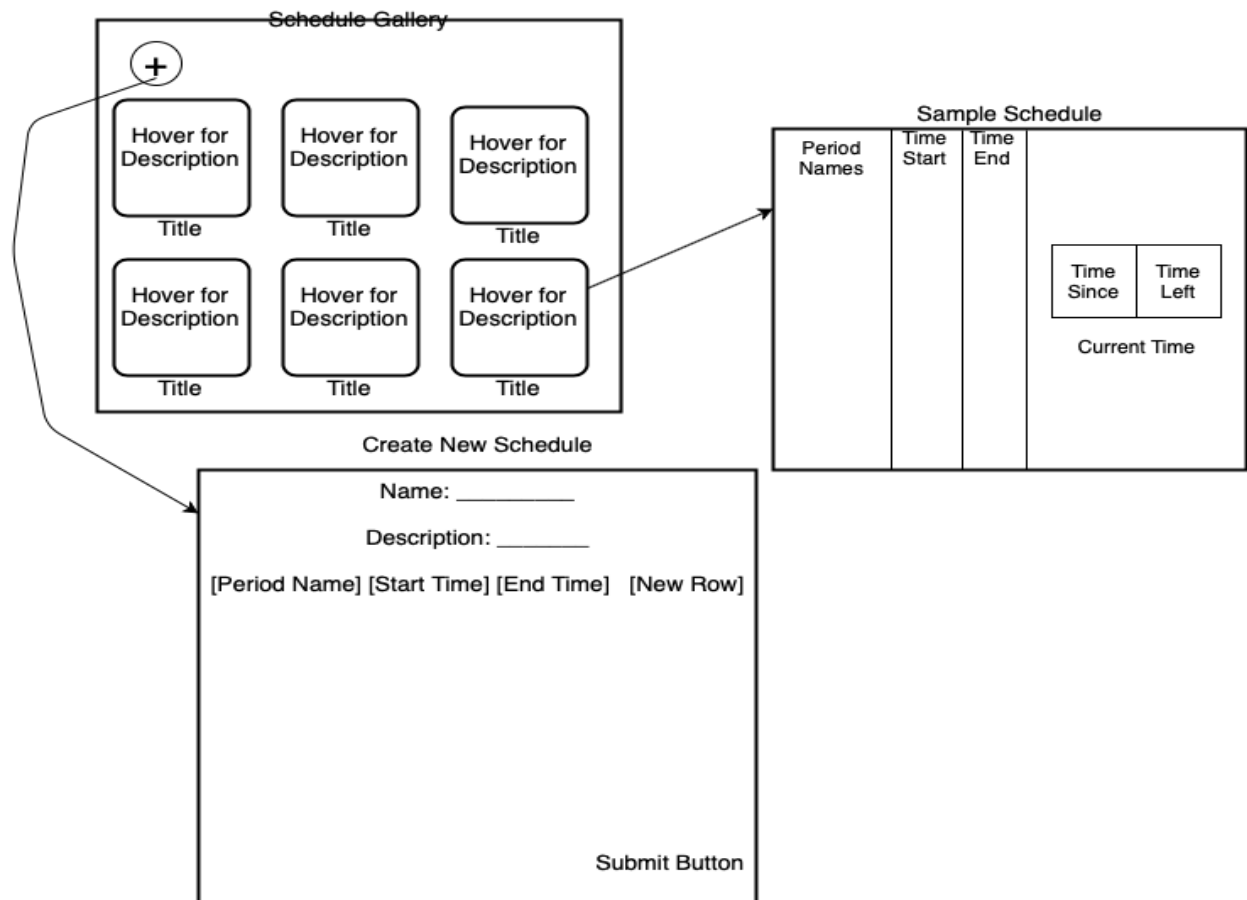


Team Quarterman

Amit Narang, Clara Mohri, Damian Wasilewicz, and Jared Asch

El Barto: Bell Schedule 2.0

Site Map



The landing page for our site will be the standard stuy schedule. It will be similar to the Bert schedule, adjusted depending on what day of the week (A, B, conference,



Team Quarterman

Amit Narang, Clara Mohri, Damian Wasilewicz, and Jared Asch

El Barto: Bell Schedule 2.0

homeroom, etc.) it is. Information about A,B, etc days is available courtesy of Pman. From there, users could visit the Schedule Gallery to select someone's premade schedule to use for themselves or log in to view their own private Schedule Gallery consisting of only their designs. We only included one version of the Sample Schedule and the Schedule Gallery as they will follow a similar format. The Create New Schedule template will be accessible both from the navbar and the Schedule Gallery page, and will consist of a form with inputs for name, description, and periods. The schedule gallery comes with a search bar, where you can close in on your desired schedules based on titles. Upon creation, the user will be redirected to their newly-made schedule and the schedule will be saved in their gallery. Users have the option of publishing their new schedule to the public gallery or to their own one.

Component Map

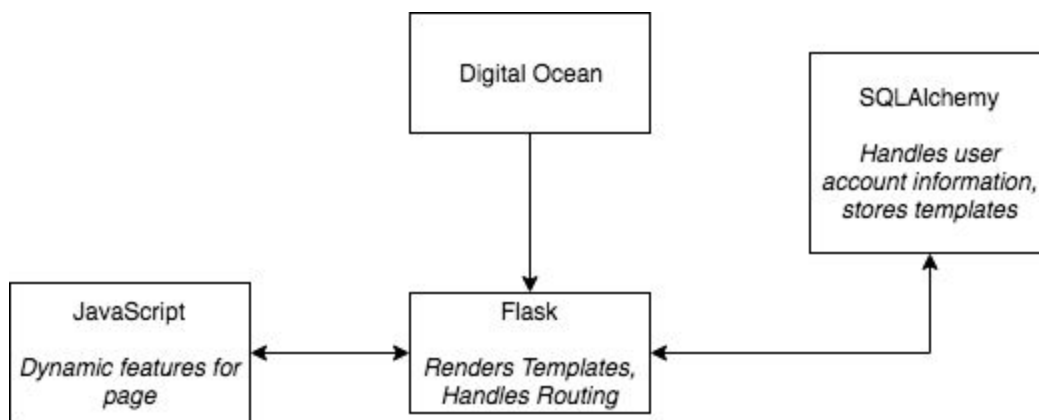
Our minimum functionality component map found below is quite simple, with only three distinct parts interacting with each other. However, we hope to add more if we have enough time, namely user input on schedules for the basic Stuyvesant schedule (such as A or B days).



Team Quarterman

Amit Narang, Clara Mohri, Damian Wasilewicz, and Jared Asch

El Barto: Bell Schedule 2.0



Our Flask app will handle routing between different templates and will render them to the user. Our templates will be rendering pages with HTML, Bootstrap, and JavaScript. It will also interact with the DateTime library to get current times. The SQLAlchemy database management system will store our user and schedule template data.

Database Schema

For our project, we only need two different tables. The database schema and explanation can be found below.



Team Quarterman

Amit Narang, Clara Mohri, Damian Wasilewicz, and Jared Asch

El Barto: Bell Schedule 2.0

Users	Schedules
<code>first_name, String</code>	<code>name, String</code>
<code>last_name, String</code>	<code>description, String</code>
<code>email, String</code>	<code>period_list, Periods</code>
<code>password, String</code>	
<code>color_scheme, enum</code>	
<code>Schedules, Database Relationship</code>	

The first one is the Users table, which contains:

- *first_name*, a String that represents the first name of the user
- *last_name*, a String that represents the last name of the user
- *email*, a String that represents the email of the user
- *password*, a String that is the hashed password of the user (for security purposes)
- *color_scheme*, an enum that contains the user's preferred color scheme for the website
- *Schedules*, a relationship to the other database which contains all of the user's created schedules (use SQLAlchemy relationships)

The second one is the Schedules table, which contains:

- *name*, a String that represents the title given to the schedule template
- *description*, a brief explanation of what this schedule is for
- *period_list*, a list of Periods. We would create a Period object for this, which is a tuple containing *period_name* (a String), *start_time* (a DateTime time object), *end_time* (a DateTime time object).



Team Quartermaster

Amit Narang, Clara Mohri, Damian Wasilewicz, and Jared Asch

El Barto: Bell Schedule 2.0

Framework

We decided to use Bootstrap for our project, as all three of our developers had used it extensively in the past. We agreed that creating something that looked good and could be done easily was our main priority, and using Bootstrap would free us to focus more on the intricacies of our project that might be difficult to solve with less time.

Roles

1. Amit Narang
 - a. Role: Project Manager
 - b. Executables: Updating the design document with relevant changes, ensuring the devlog is being updated, connecting the Flask app to Digitalocean, adjusting team vision as we encounter challenges, helping create database and back-end.
2. Clara Mohri
 - a. Role: Mid-stack Developer
 - b. Executables: Helping create and manage database, writing the underlying functionality of the Flask app.
3. Damian Wasilewicz



Team Quarterman

Amit Narang, Clara Mohri, Damian Wasilewicz, and Jared Asch

El Barto: Bell Schedule 2.0

- a. Role: Back-end Developer
 - b. Executables: Developing the shared library functionality, and integrating the database with the Flask app.
4. Jared Asch
- a. Role: Front-end Developer
 - b. Executables: Designing the templating functionality, and creating the basic templates for people to glance upon.