Team Q
Cloq
Spring 2018

Overview:

Cloq is an innovative timeclock that is designed to be at its heart simple to use for both regular employees, but also their managers. Towards its most basic purpose Cloq supports the entire reason of having a time clock, that is recording the time when an employee has clocked in and clocked out. We openly admit, we are not the first to create a digital time tracking solution like this. A quick search reveals that applications exist for this purpose. However, we found a few common issues, they were either dedicated software requiring an application to be installed. They were somewhat non-intuitive to use, and lastly they were fixed in nature. In the process of developing Cloq, we hope to fully address these issues. By making Cloq a web application, the need for dedicated software is replaced by any browser with internet access. With this in mind, Cloq becomes easily adaptable to more situations, and can fit into a variety of workflows. Throughout the development process we also strived to make Cloq as intuitive as possible, by simplifying our interfaces and refining them to have the features one might expect, where they would expect it. Lastly, although Cloq is relatively complete for the purposes of this assignment, we believe that Cloq would be easily adaptable as a foundation to support different needs of organizations.

Team Members:

Team Member Name	**Github Usernames**
Arun Dunna	adunna
Darren Farrelly	dfarrelly
Isabel Hagberg	ihagberg
Jane Tangen	onlyNexusHere
Shane Parr	sparr1
Troy Mayrand	tmayrand

Github Repository:

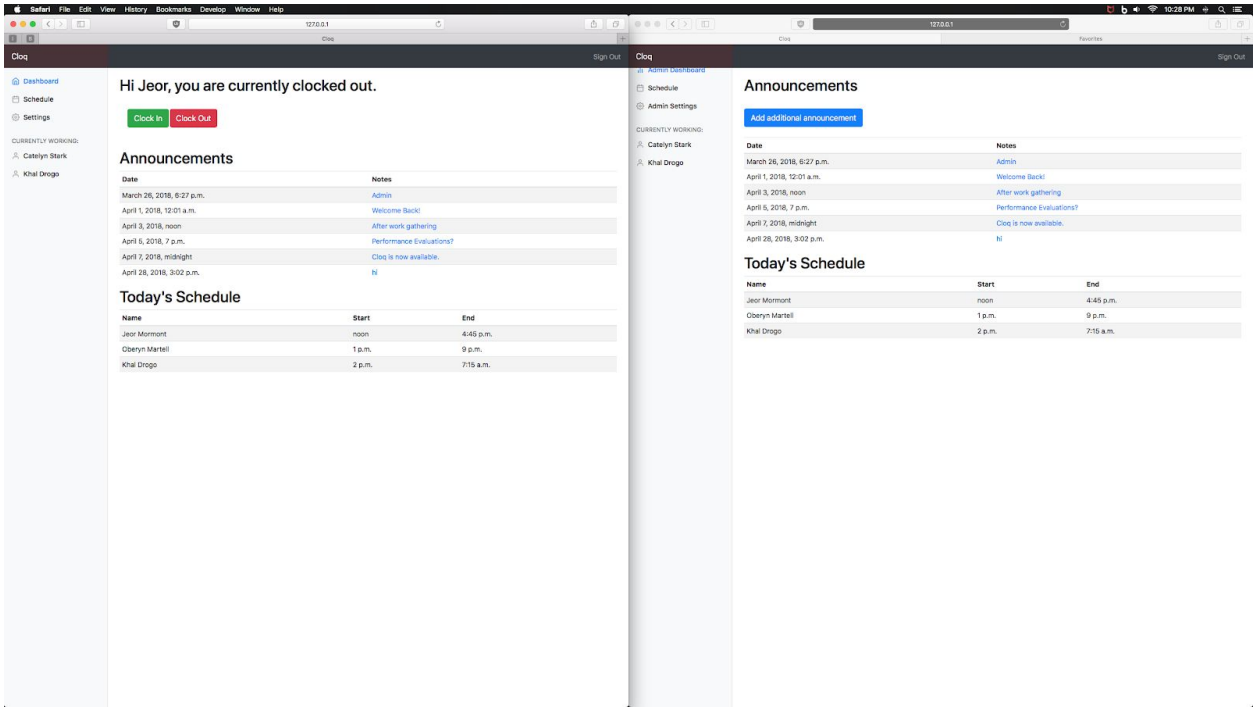
A link to our code for Cloq may be found on Github here:
https://github.com/tmayrand/COMPSCI-326-TEAM-Q

User Interface:

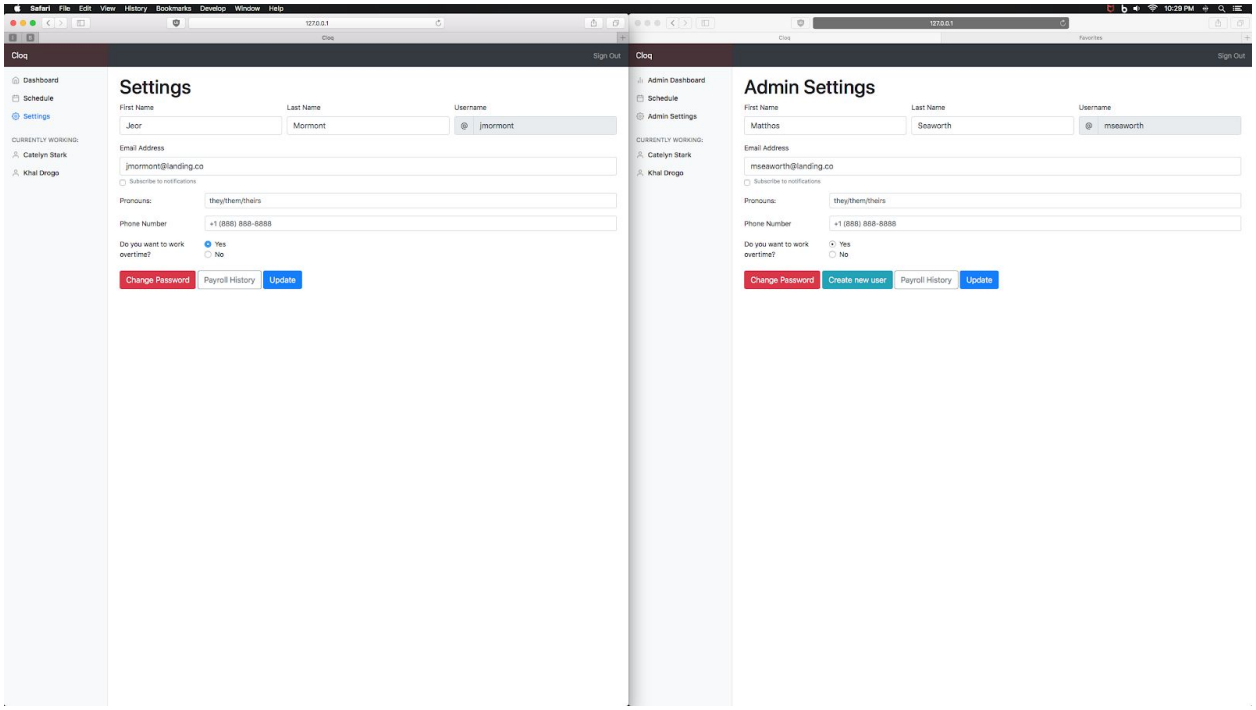
As one of our overall design goals, we designed out UI to be as clean and organized as possible. Toward this goal, when looking at our application, each user is presented with three major “views”, ‘Dashboard’, ‘Schedule’, and ‘Settings’. Each one of these pages essentially has slightly

different content rendered on the page, depending on which usertype the currently logged on user is. One additional design choice of ours was to keep interactions within each of these views “within” the sidebar category. Most interactions with Cloq occur inside of pop-up modals. For example, two interactions that are classifiable as settings, occur within on the ‘Settings’ view. By having these briefly used interactions contained within a modal, we can avoid showing extra content when not necessary, while also keeping the application to a minimal number of pages.

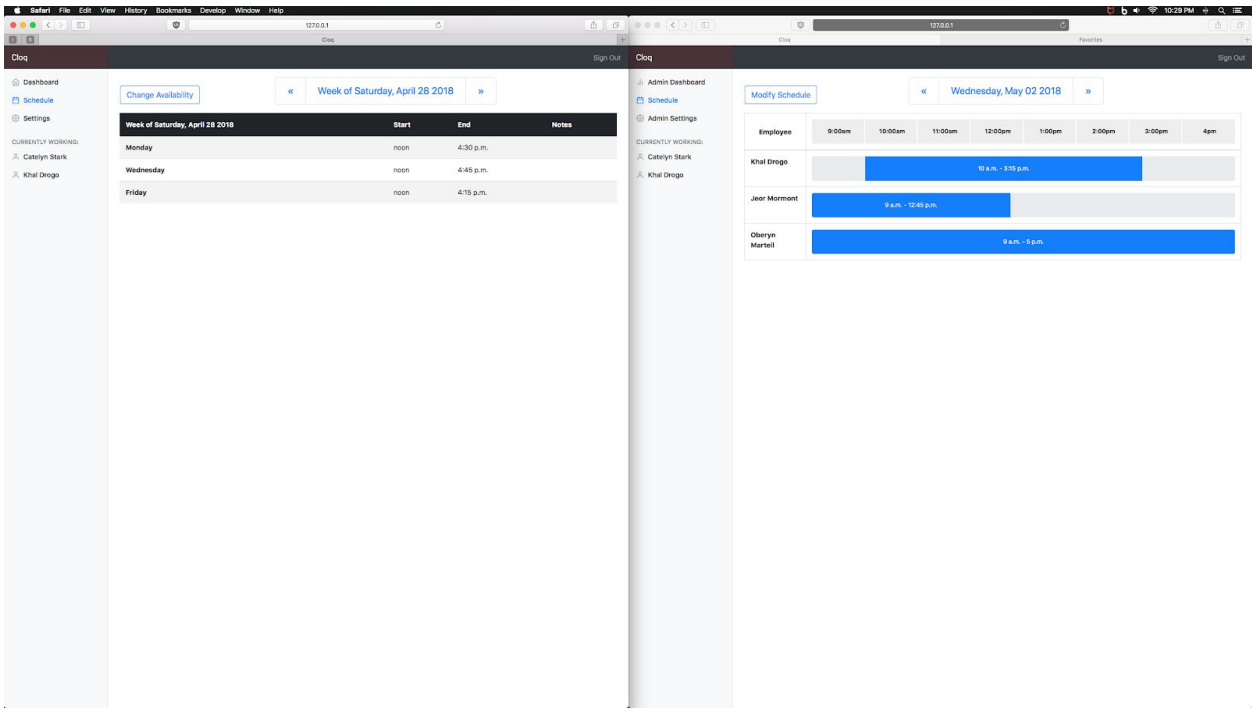
Despite these differing views, the content on the sidebar and top bar remains constant, ensuring navigation remains obvious and efficient. Returning to the differing views. The ‘Dashboard’ view is designed to show off the current status of the workplace, and provide quick functionality to users the following images show our UI for each usertype, regular users on the left, and admins on the right.



In the case of the ‘Dashboard’ views, the table of shifts, and announcements is common across both the user and admin views. However, when signed in as an admin (a user with usertype 2) the buttons that an normal user would need to clock in and out are replaced by a button for adding announcements to clock it.



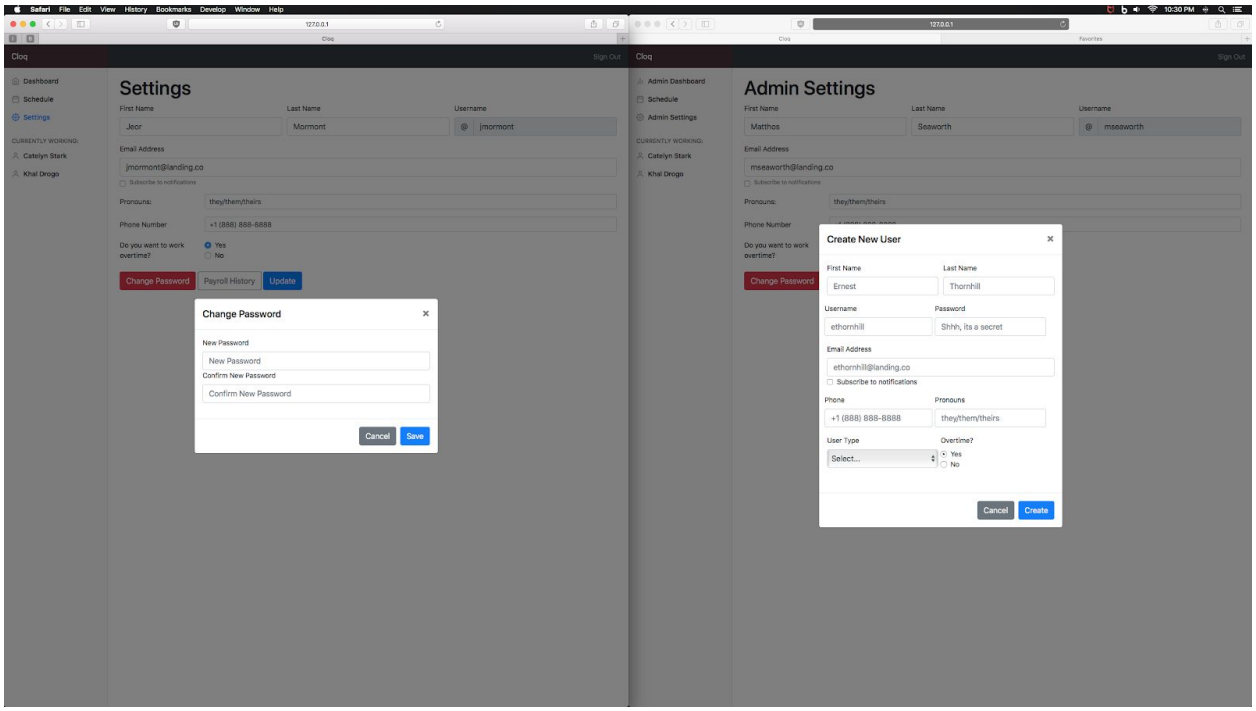
In a similar way, the settings pages for both usertypes is largely identical. The only difference is currently the addition of an extra button on the admin view to add new users.



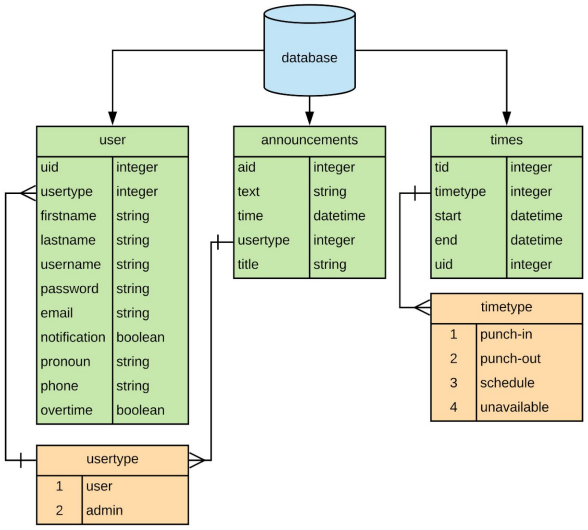
Lastly, the ‘Schedule’ view has the largest changes between the two usertypes. When signed in as a regular user a user will see table that shows their schedule for a weekly period of time. If an admin signs as loads the schedule view and sees an empty table because they are not scheduled would be a waste of time. Rather, admins see a daily series of colored bars that show when

individual users have scheduled shifts, while a users just needs to quickly see when they need to show up and can leave, the admin needs to be able to ensure that there is adequate coverage, this becomes obvious when overlaps between shifts are now actually visible, rather than inferred from a list of times. In a similar way, both users and admins need to be able to make changes to a schedule, however, admins need a way to create shifts, while users only should be able to change availability.

An example of some of the modals used for more in-depth settings, the left page is used to change passwords, and the right is designed for admins to add a new user to Cloq:



Data Model:



Our model consists of three unique entities (the green tables above, the orange tables represent keys that certain values can take on). Our user model is designed to encapsulate all the information Cloq maintains for a user such as their name, email, and preferences. Announcements consist of a title, and body of text, date of creation, unique ID, and the user types that are able to view the content. Lastly our most complex model, each times object can serve one of four purposes depending on how it was created or needs to be used by Cloq. Timetypes 1 and 2, are generated by users when they create punch events, type 1 signify a punch to the start of a shift, whereas type 2 mark a punch at the end. Time objects of type 3 are used to represent official scheduled shifts (those rendered by the scheduler), and type 4 represents what is essentially an anti-shift, times a user submits that they are explicitly unable to work. Times objects of type 1 and 2 set both the same value in the start and end datetime fields, as a punch knows nothing about how long the duration of a shift might be. Times objects of type 3 and 4 have differing start and end datetime fields so as to fully represent a block a time.

URL Routes & Mappings:

cloq/dashboard	The user dashboard.
cloq/admin_dashboard*	The admin dashboard.
cloq/settings	The user settings page.
cloq/admin_settings*	The admin settings page.
cloq/schedule/<int:year>/<int:month>/<int.day>/	Dynamic schedule page for users.
cloq/admin_schedule/<int:year>/<int:month>/<int.day>/*	Dynamic schedule page for admins.
cloq/availability	Page to show users availability data.
cloq/login	Login page to sign users into Cloq.
cloq/logout	Logout page that redirects to login.

* = Requires an account with usertype 2 to view. *See .../COMPSCI-326-TEAM-Q/project2/data_model_accounts.pdf for list of accounts with usertypes.*

All pages require a valid login to view, they will redirect to login if not signed in.

Authentication & Authorization & Team Choice:

We have decided to merge these two sections as our Team Choice contribution heavily contributes to our authentication for Cloq.

More specifically, we decided to write a custom backend. We wanted to separate django's authentication user ("AuthUser") from our application user model ("AppUser"). The new backend first searches through the users table to check if an account exists and then verifies the

password. If the password is correct, it then searches the AuthUser database to for the user and if it doesn't exist, creates a user with no privileges, otherwise it returns that user. If the user isn't found in the AppUser database, it means the user may be an admin without an AppUser account. So, it searches the AuthUser database and checks if the user exists, and if so, then checks the password and returns appropriately the user or nothing if the password was incorrect. Additionally, if a user is already logged in and visits the login page, it will redirect the user to their dashboard.

Given our modifications, and dynamic views to hide settings that are not part of what that user should be able to do. We have not explicitly defined authorization groups within django.

Conclusion:

In terms of developing Cloq, the features we wanted to implement were fairly straightforward. I think it is fair to say that we are all proud of the application we built over the course of this semester. One problem we did encounter though, was that we often underestimated to work required to actually implement some of the features we wanted to add. Our group while working also generally followed a “get it done” methodology without regard for the future. A few times this did come back to complicate matters. During some of our later development we realized our model was not the most ideally constructed for storing shifts (where each one requires two separate objects to demarcate the times of a single shift. One other case is that we store availability data (I.E. a user’s requested shifts and conflicts) with their schedule and punches, looking back on this though each of these different timetypes should have had a separate table created to simplify storage.

Outside of our development, our largest difficulties encountered as a team were in terms of clarity of communication, distribution of work, participation in the development of Cloq. To expand on the distribution of work issue, from project2 onward assigned tasks were noticeably broad, which is collectively a fault on our end, rather then assigning a full task without context, we probably should have rather focused on creating more lists of tasks to complete. Doing so would have definitely allowed for collaboration during development, especially considering most pages in our project as is were worked on by likely two at most of our group.

The time between project3 and the Pitch Party was a period of heightened tensions, as we realized that some of the content we thought was completed had yet to be fully implemented. But also at least one of our group members felt like they were being asked to complete only work that no one else wanted to touch, which resulted in some undesirable situations that appeared to be a result of this assumption surrounding the distribution of work. Reflecting on the way our group functioned though, this was another manifestation from the way we ended up assigning tasks rather than disregard of team members wishes.

Slide Presentation:

A link to our slide presentation for the Pitch Party may be found on Google Slides here: https://docs.google.com/presentation/d/1Af4tK0JLdRP6FvmJ8lxMA-tGZ-WiwiwJTz4h4wpd5b8/edit?usp=sharing

Single Slide:

A link to a slide that we are proud may be found at the following link: https://drive.google.com/file/d/1bJ2dK-fj-C6r8TUhtmtSQhsSY0-_-7qT/view?usp=sharing