# Slate – struct by\_lightning{}

Chad Nierenhausen

Homero Pawlowski

Timmy Garrabrant

## Introduction: What is Slate?

Slate is a modernized appointment scheduling application that will allow students to make appointments to visit with their academic advisers. We aim to create an experience that is quick and easy and gets students the information that they need, when they need it, about their appointments. We also hope to simplify the experience that advisers have when setting their availability in the system.

## Project Status: Where we are –

We have built out our database and created seeding mechanisms for testing purposes. The models and relationships for our database entities have been fleshed out and a layer of controllers are orchestrating the retrieval and delivery of the data to the client. Most of the user interface design decisions have been made and the student facing functionality is nearly complete. Currently a student is able to select the category for the appointment they are creating. Then they are presented with a list of the advisers that are available to meet regarding that category. Once an advisor is chosen their availability is pulled and we are in the process of displaying that information in an easy to consume way. We have implemented and tested the availability retrieval and appointment scheduling functionality but haven’t made it to the point of being able to schedule an appointment through the UI just yet. A majority of the underlying backend work and decisions have already been made and implemented. The time consuming portion of making an easy-to-use interface for the customer to interact with is currently our main focus.

## Project History: How we got here –

We met as a team to discuss what we felt were the big pain points in the current system and we set that as an initial base line to work from. We then met with one of the current advisers to discover what the pain points on the administrative side of the system were and added fixing those issues to our base line. We self-selected the parts of the system we each wanted to work on: Homero took the database design and implementation and model objects, Timmy focused on the controller API and business logic, Chad focused on the front end design and client side code. Most design decisions have been agreed on organically, through either as a discussion after class or in the group chat room we are all participants in. Most changes have been minor tweaks or additions nothing major has had to be altered from our initial goals. The most challenging aspects we have faced as a team are the co-located meetings outside of class time. We ended up working on what we could when we each have had time, and have used the group chat to keep up our communications. The roles haven’t been rigid and each of us has helped out in different areas as needed and the project is starting to come together nicely. We spent a lot of time initially on or design and framework, and while that hampered our initial MVP, it has given us a very strong foundation to work from where we haven’t needed to keep refactoring our code base.

## Going Forward –

We are choosing to focus on a self-contained product going forward. We have a solid foundation and framework that we have built and we feel that it is important to build a self-contained product that is strong and usable. Initially we felt that integration with existing university login and e-mail systems was a priority. Though these are still a features that we want to include, we feel that having a strong user interface and a full featured scheduling system will ultimately lead to better customer adoption. The user interface of any software product can be a major time-sink, but we believe that the user interface is a major component of our product and it is something we want to specifically focus on. We will continue to refine and refactor our backend framework, but going forward, creating a fluent user experience is our ultimate goal.

## Design Documents



