Kali Burres Cheeley

Professor Garrison

Software Engineering

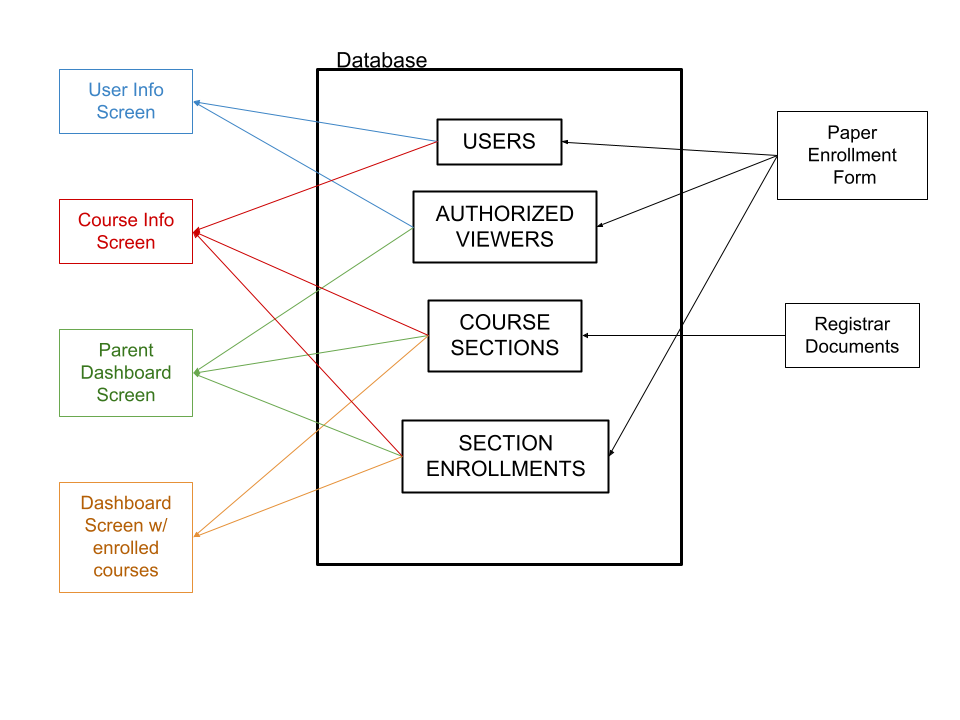
03/16/2022

Final Design

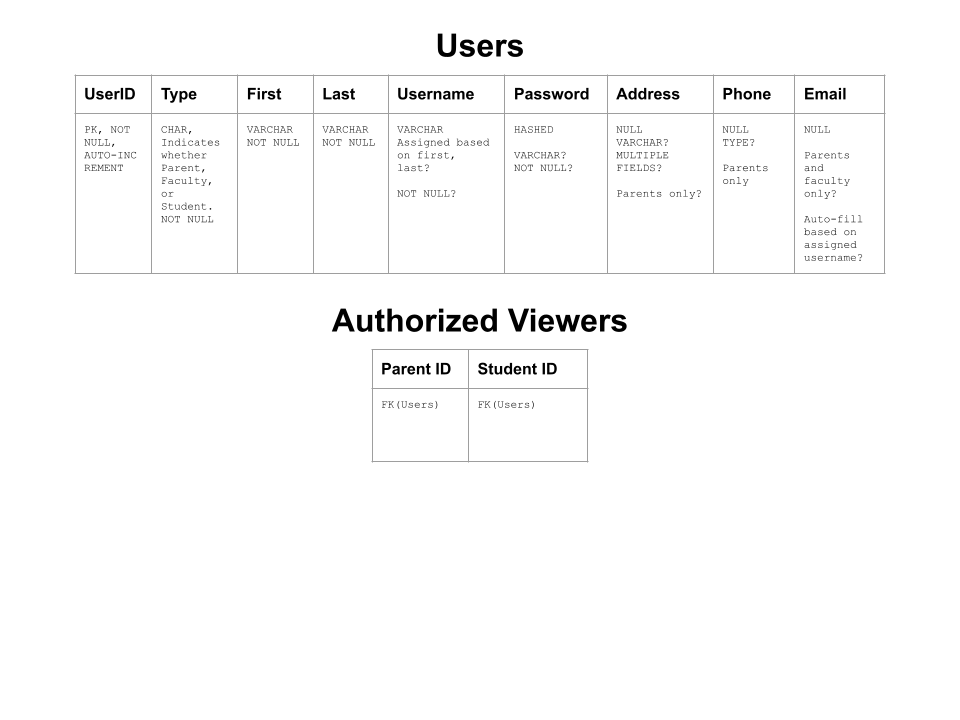
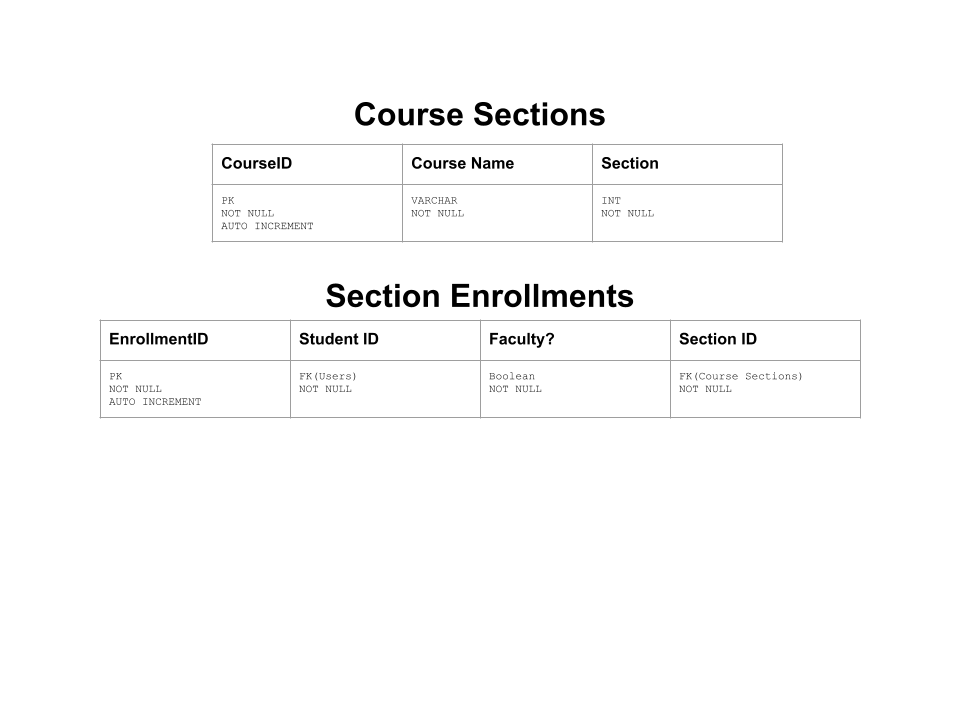
**Language/Tool Decisions:**

1. I would love to learn Node.JS, and trust the popular opinion about how great it is as a backend environment. If I had all the time in the world, I would probably undertake this project using Node.js, especially with how popular it is at this time and how useful it seems for scripting and how popular its use is currently. Realistically, however, I know I do not have the time to pick up a new language right now. I have a foundation in PHP, and would like to advance that to a basic proficiency anyway, and as what would have not long ago perhaps been the “obvious choice” for a project like this, I am confident that it will do the job well. The user also noted that eventually the application should be done in the Laravel PHP framework, so building the first edition with PHP is appropriate.
2. While MongoDB seems like a viable option for the storage of the SIS data, the user-given requirements require a sql or sqlite database, so phpMyAdmin (or something similar and free) will be used. (phpMyAdmin, being built in PHP, makes PHP also a good choice for the scripting language of the application.)
3. HTML and CSS will have to be used for the structural and design foundations of the web pages.
4. Tailwind CSS and Tailwind UI components will be used to construct the CSS.
5. JavaScript will be used to manipulate data within the page, including jquery and potentially axios.
6. InfinityFree will be used to host the Web Application.

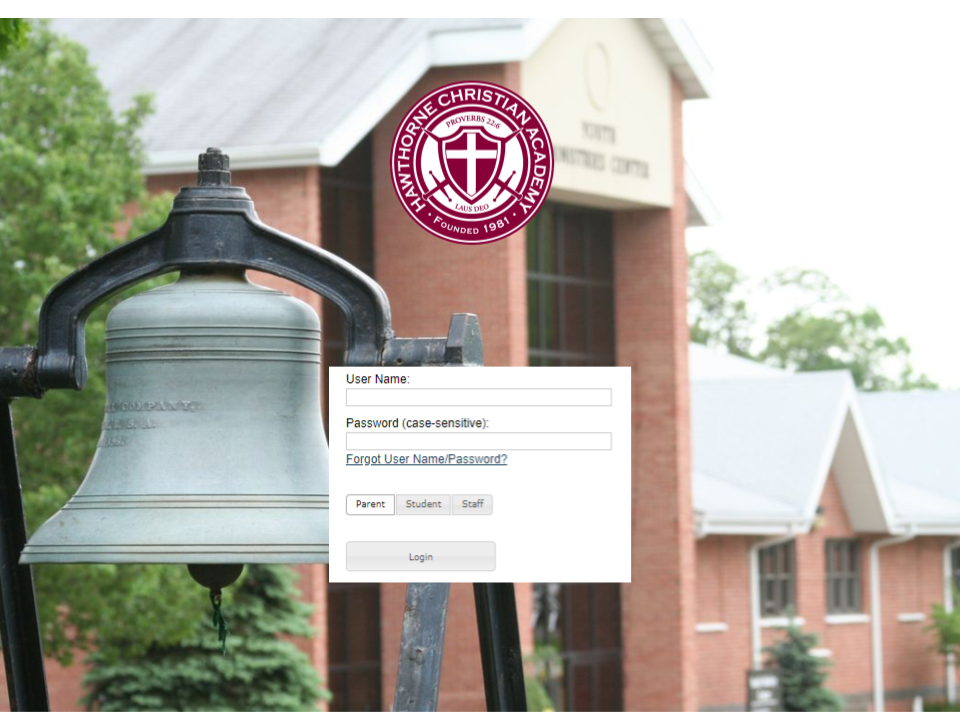
**Data Flow Diagram:**

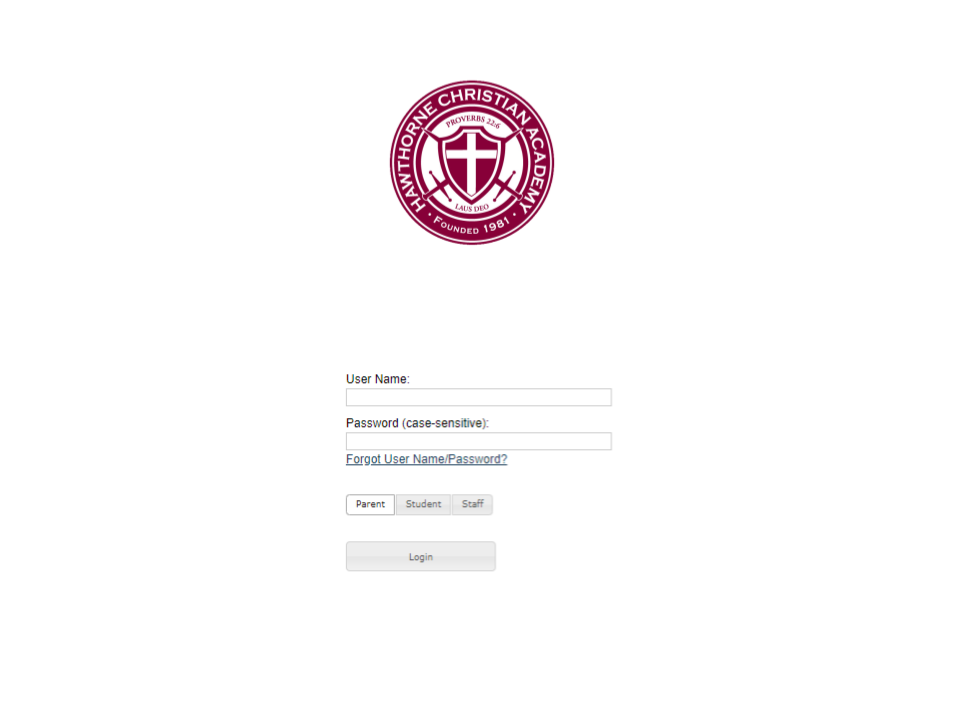
****

**Database Design Structure:**



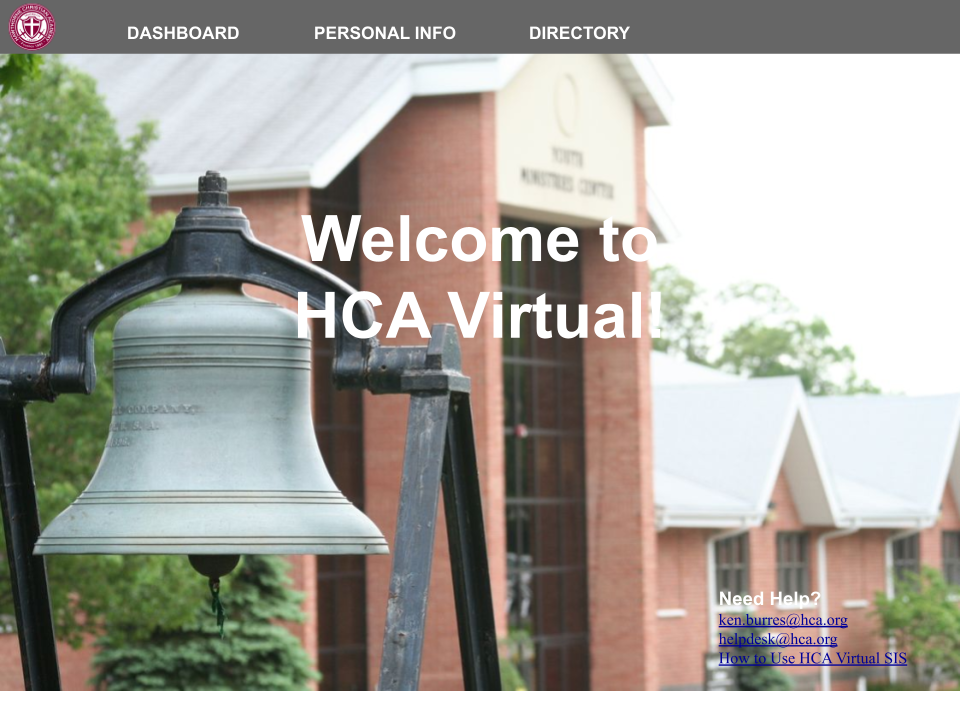
**Visual Screen Prototypes:**

**Login Screen:** 



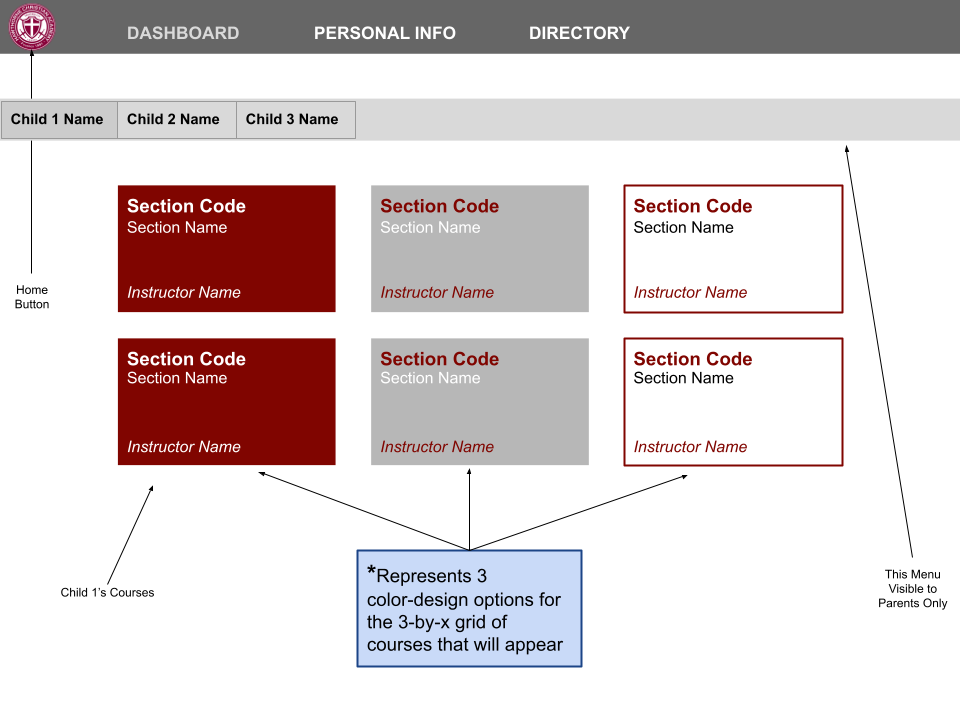
**OR:**

**Landing Screen:**

****

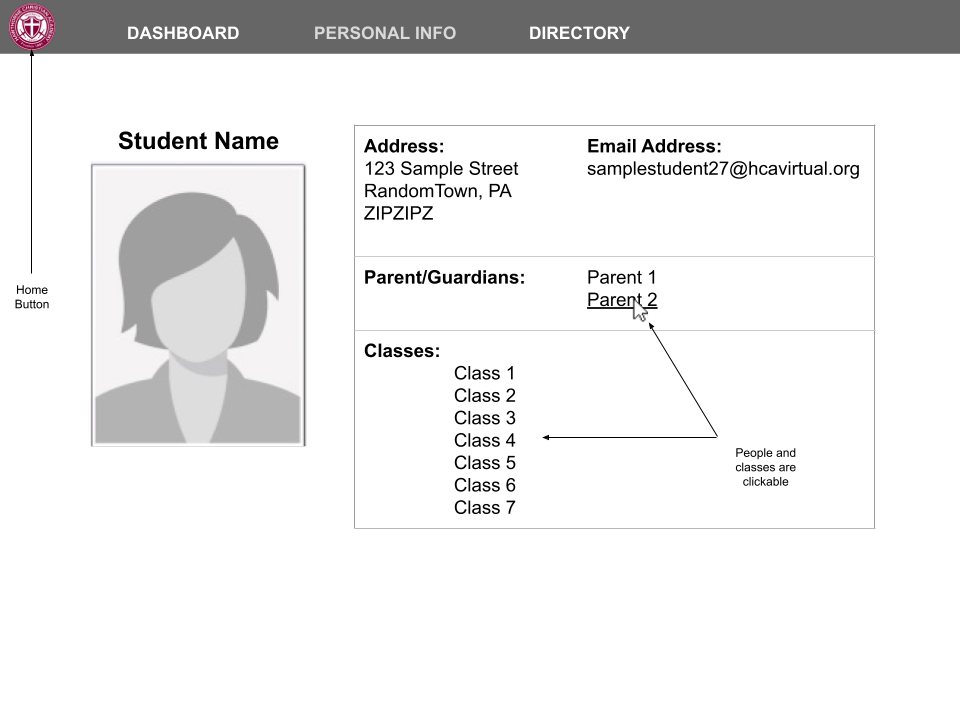
*I may or may not include this screen.*

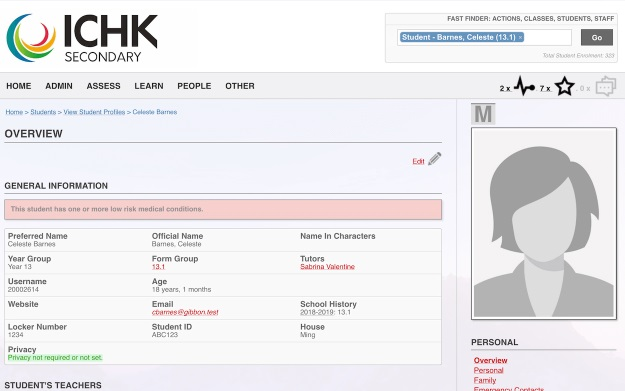
**Course Section View/Dashboard:**



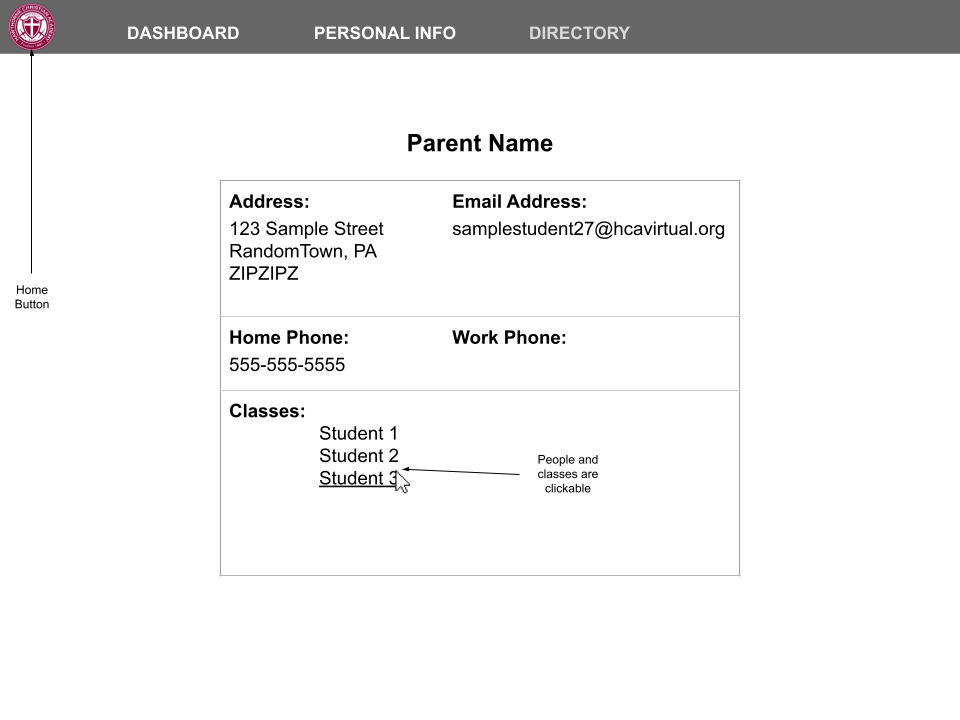
*Inspired by this image (below) of an LMS dashboard, this design aims to replicate the user-friendly modular design in the representation of courses. Clicking on a course will bring a faculty member to a screen specific to that class section, showing a class list. Eventually, for parents and students clicking on a class will also yield it’s own screen with relevant class information. The speed of progress on this project determines the level of development this aspect of the SIS will receive in the first iteration.*

**Student View Screen:**

**

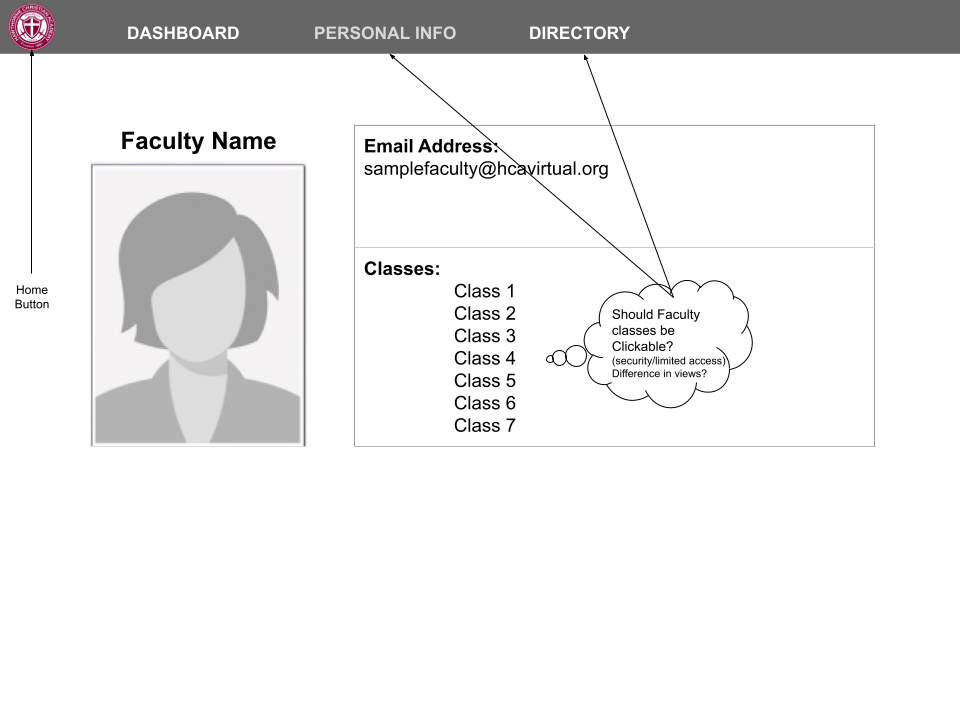
*Partially inspired by the image below, the grid format of information organization seems to work well. Bootstrap provides a good flexible grid structure that can be used. While currently pictures of the student will likely not make it into the first iteration of the software, including them in future iterations is advised and leaving space for such input on the page is advised.*****

**Parent View Screen:**

****

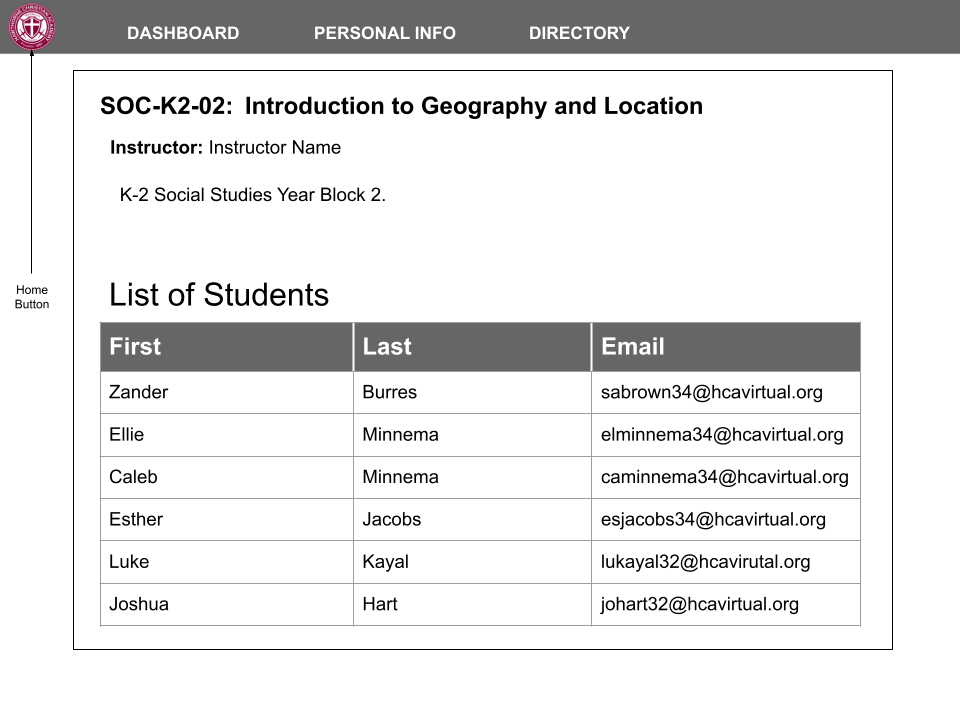
Needs further revision. Simplistic adaptation of student screen.

**Faculty View Screen:**

****

It still needs to be determined whether links should be clickable in certain circumstances. For example, if the same screen is going to show to the user themself under “Personal Info” as in the directory from random students and parents, then the links should not be clickable, because the users themselves have other ways to access class information (although parent links should still function). The directory also has not been fully decided on, so that will impact the decision.

**Class View:**



This draft needs revision, but is the beginning of a course overview screen. Who will it be available to (just those enrolled, teaching, or with viewing permissions of an enrolled student)? What other relevant information should be made available? Should parents who can view other students from here be able to surf through courses through student’s course lists? How should it be limited? BUT parents should be able to access the contact information for parents of students in the same class eventually through this list view. The class list format is inspired by the snippet below: