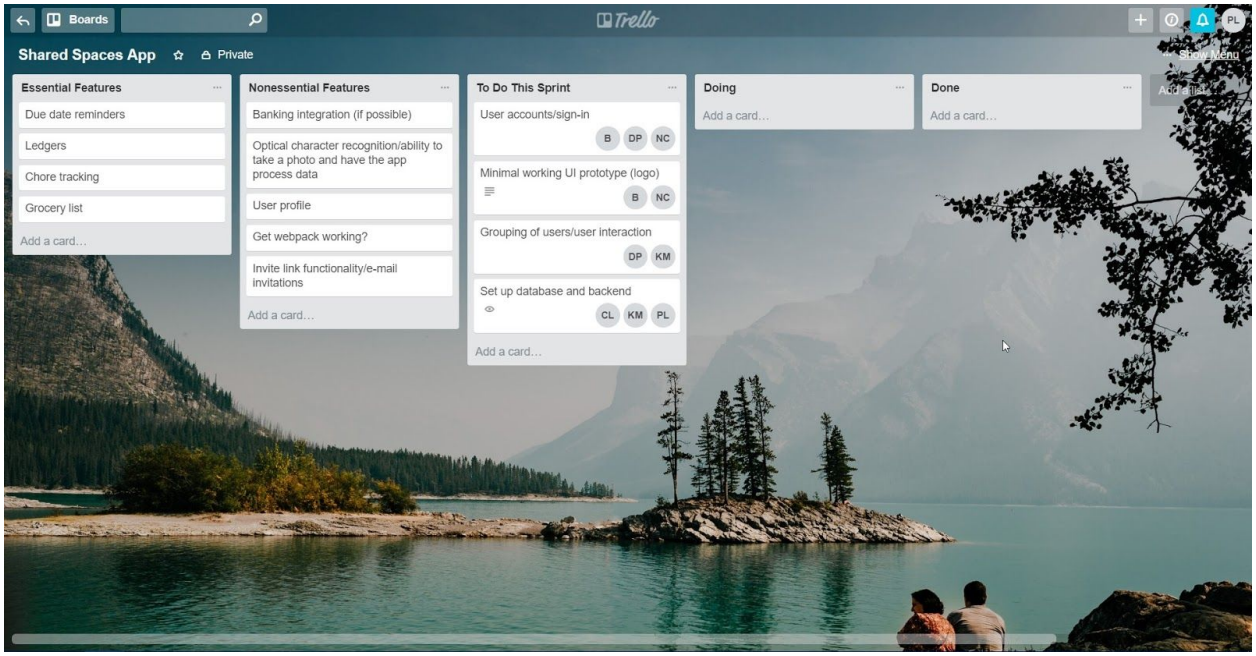
Project Management Tool: We will use Trello to plan and track the progress of our project. We have created a Trello board and captured the requirements of the project. A screenshot of the Trello board at the time of writing is included below.



The description of each item on the Trello board has been documented according to the Use Cases template.

The overall goal at the time of writing is to take 4 to 5 sprints to accomplish what needs to be accomplished for the project to work. The first sprint will cover the basic functionality: a minimal working example of the server and database schema, a basic UI prototype, the ability to group users in the database, and potentially user accounts. The second sprint will begin to flesh out the already established infrastructure, adding more features to the server side and the ability to host the user interface pages from the same server. User interface will also be drawn up for the main features of the app. The third sprint will begin integrating the user interface with the backend, such that the webpage/app that is served can actually communicate with the server, and so that any changes the user makes in the user interface are reflected accordingly in the database. The fourth and fifth sprints will be mainly polish, adding additional features that are described as non-essential, things that may be more challenging and might block progress on the project if set to a higher priority.

Plan Cycle within the Project Management Method: The requirements for this section have been documented in the Trello board above. For example, Paul Lucero, Kyle McDevitt, and Cayleb Langhals have been assigned to set up the database and backend, and the due date for this (not shown, but written in the description of the item) is the end of next week.

Agile Meeting Summary: During the past sprint, Kyle, Paul and Cayleb have managed to set up a small working server on AWS, which is able to serve web pages (both JavaScript from React and the like) as well as basic HTML. It is also able to serve a mySQL database, which we have fallen back to rather than work with MongoDB. On further reflection, MongoDB is not as much of a fit for our project, since we do not need the scaling capabilities that MongoDB's sharding offers, nor do we need a non-relational (also referred to as noSQL) database that would be offered from MongoDB. The syntax and setup procedures are also more confusing.

While the past couple of weeks have been hectic, the rest of the team has been preoccupied with studying the frameworks that we will need to know to be able to effectively develop this application, such as React and NodeJS. This means that we are in a good position to begin more formal and more intensive work on the actual feature set of the application in the weeks to come.