Data Modelling

- **Board**
  - No Collection or Model because there are no information to store in a Board other than a name
  - Used a View object to store any event handlers
- **List**
  - Represented as a Collection, a Model and a View because there are information to be stored into them like its title, id and its collection of Cards.
  - I have embedded Card collection into the List Model. The reason is because if I collection cards under one BB collection, the positioning of the model makes no sense.

- **Card**

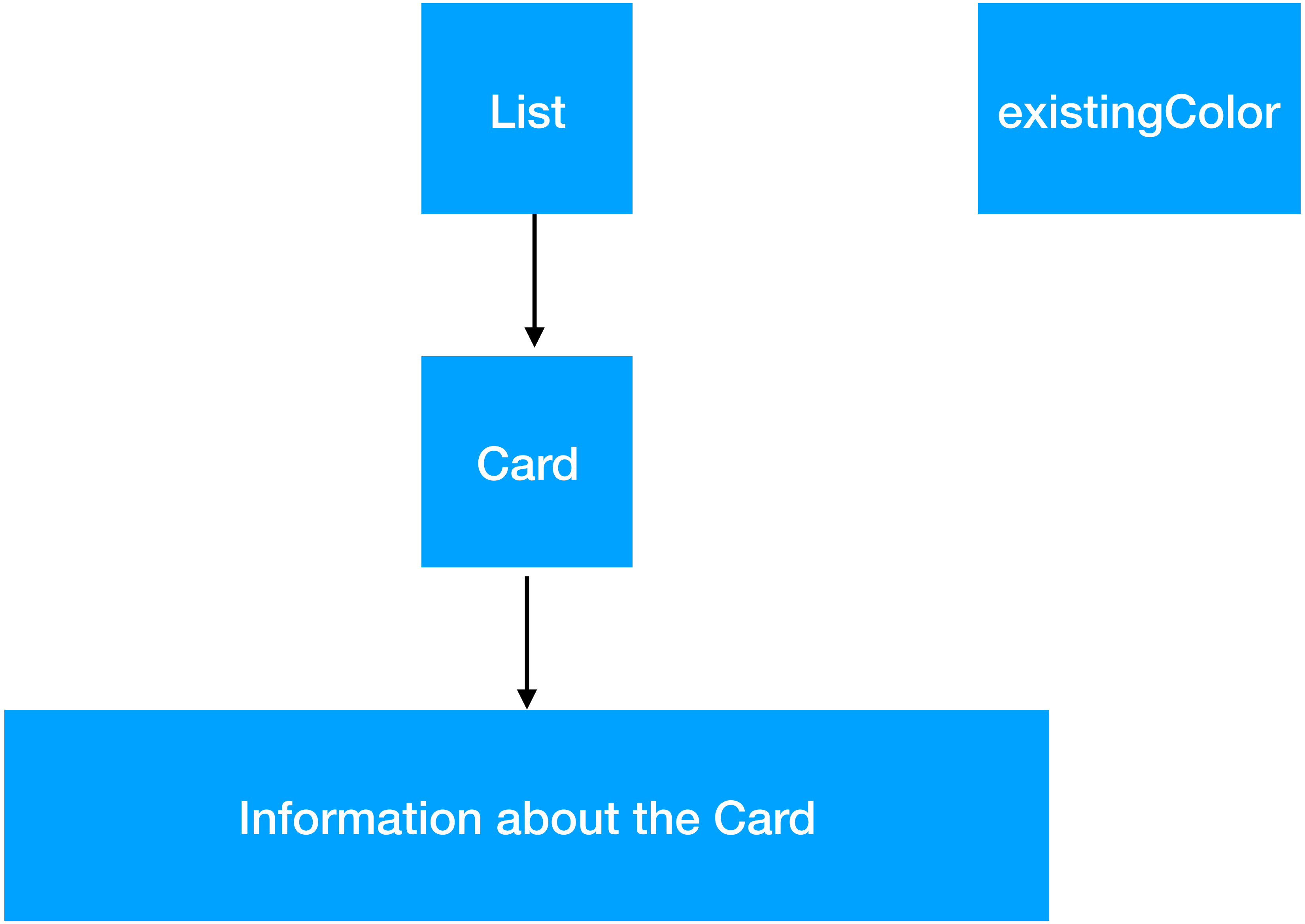
    - Represented as a Collection, a Model and multiple View. A Card need to have many View because it will be too cluttered to have all events regarding changes of title, description, labels etc cramped into one view.

    - I have store all information regarding the Card like comments, title, due date, description and labels inside the Card Model as a pure string, arrays or object. I don't think there are additional benefit in representing these information in BB model or BB collection. Also, my JSON data stored in my back-end has only three properties which is “lists” , “cards” and “existingColors”. So I can replace those item “one at a time” instead of altering its properties.

- **Colors**

    - I have represented the colour label as a BB Collection and Model because the label tag available for tagging is different from those labels already tagged in each Card. Therefore, I decided to store them separately in my backend JSON file.

Diagram for Data Modeling



List

existingColor

Card

Information about the Card

Assumptions

- There is only one users. So all the comments will have a single user as the commentor.
- I assume that users will be interested in knowing all the Card’s activities in a single session. So I stored all my notification into window.localStorage.
- There is only a single due date, and there is no progress bar, that is a card will be either done or not done, no 50% done.
- There is only a single Board, so I don’t have to embedded those List into a Board Model.

Client and Server Communication

- Card
    - All Ajax communication regarding a single Card like edit of title , description and label will be done by the BB Model.
    - All Ajax communication for the Collection is done by the Collection such as adding, removing and copying a Card.
    - The communication made to the server will be as often as user changes the information made to the card.
    - The data sent to the server will usually be the JSON file of a single Card. However, sometimes a whole array of Cards needs to be sent to the server in order to communicate the order of Cards within the Collection.

Client and Server Communication

- List
  - As for a single List , there is not much of changes to be made to the List information. But one thing that needs to be done is to create a Card Collection each time a List is created
  - As for List Collection, the usual adding and removing of List is communicated to the server via Ajax.
  - The server will communicate to the front-end at a single point, which is when index page is loaded. This is where “route/index.js” will pass in List , Card and Colors to the App object.

Structure

- All Backbone Model, View and Collection will have its own folder. This is much more organised and clear.
- Application.js will serve as a main file that act as a controller/ coordinator.
- handlebars_utility.js file that will store all my handlebars helper function.
- Express will reference a single all.js route file, which served as a linkage to all my other route file such as index.js, list.js and so on. This will uncluttered by express App.js file. Anytime I want to add a new route file, I just have to add the name into the array in all.js file

Libraries

- Drag and Drop
  - Draggable by Shopify and Jquery UI
- Date picker
  - Pikaday by dbushel and Jquery UI
- I have evaluated the above libraries. In the end I have chosen Jquery UI as the documentation is much more robust. Also, Jquery UI has both functionality, so I can combined the vendor file as one. Beside, I am more familiar with Jquery functionality.

Challenges

- I can’t embed the “popup” html into the parent element. Because anywhere I click within the “popup” will trigger the parent element event. However, how do I absolutely position the “popup” relative to the list item’s . Solution: anytime a parent is clicked, stored its position in Javascript and then pass in the position to the “popup”

- I can’t pass a Card collection as a JSON to the server. “Maximum call stack error” will be raised. Therefore, each time a new list is created, I will pass in an empty array as the Card instead of passing a full BB collection as the Card.

- I initially store Card and List separately. But it doesn’t make sense to re-order them if I put them all into a single BB Collection. Solution: put the Card Collection under a List

Future Improvement

- I would improve the validation of date and time for the due date. Also, I will store time information in each comment like in the real Trello.
- I would definitely refactor my code, seems like a lot inefficiency, redundancy and unreadable parts.