

# kordio

system.in.idea

# TEAM MEMBERS

## **Julian Kuk: *Project Lead***

Responsible for managing the flow of the team and the distribution of tasks, with a primary focus on front-end development.

## **James Yanyuk: *Back End Lead***

Primarily focused on back-end development, and general integration of the front-end and database. Manages the application's dedicated server (currently running at <http://kord.io>).

## **Matthaus Wolff: *Back-End Developer and Marketing Manager***

Primarily focused on back-end development and marketing of the application.

## **Nam Phan: *Back-End Developer***

Primarily focused on database design and back-end development.

## **Sara da Silva: *Front-End Lead and Graphic Designer***

Primarily focused on front-end development and graphic design.

# OVERVIEW

Kord is a centralized hub for any human interaction over a digital medium, whether that be for friends, businesses, working partners, social groups, etc. Users will be able to chat and communicate in a room with an interactive board. Resources such as documents, videos, etc. can be placed onto the board. It can be drawn on. It can just be used as a space for saving important links, brainstorming, making timelines, and more.

It is a space where digital interactions have a sense of continuity and connect people to each other, rather than to the ether.

## *Disclaimer:*

Any designs and features discussed in this document are subject to change. UI designs will evolve as we test and feature lists will change as we run into technical problems and find inspiration. Technical details will be discussed in a future document. Note that also, the two wireframe versions of the UI illustrate the different layouts and set ups we will be experimenting with.

# SCENARIOS

## **A couple of friends want to do commentary of a sports game:**

There is a large sports event coming up and a group of friends really want to watch and talk about it together. However, they are all across the world in different places for their own reasons. They can use kord.io to create their own personal board and stream their favorite game. They can use a smooth embedded chat on the website or start a voice chat.

## **A TA/Professor wants to hold a screencast:**

The instructor can invite their students into a private board for a video lecture. This can be a pre-recorded lecture and be uploaded to the board during which the students can discuss the material chatting with each other. It can also be a live lecture, where the instructor organizes a time for the students to meet and then streams the lecture live on one of our boards.

## **Students doing Peer to Peer Essay editing:**

Students can upload their essay and sources to a board and invite their friends to join and comment on their work. Editing will be dynamic and users will be able to make suggestions by commenting directly on files, track changes, and communicate through voice or text.

# SCENARIOS

## **A group of GUI Developers Contributing to a design:**

Users will be able to work on images together with kord.io's dynamic drawing capabilities, as they simultaneously edit and discuss a particular interface or design. They will also be able to code together with context highlighting for hundreds of languages.

## **A presenter shows his ideas to board of executives:**

Users will be able to create presentations on kord.io in formats such as PowerPoint or keynote to show off ideas to potential donators and investors. Much like Google presentations this will also be easily shareable and presentable with multiple users.

## **Gaming Team Strategizing:**

A group of friends are preparing for a large tournament and they want to quickly and easily share spreadsheets and make diagrams for the game. They will be able to have their own board with various types of material as well as voice communicate in the background during the game.

# SCENARIOS

## **Large Team Organization:**

Project managers can create multiple boards linked to each other and assign members to specific tasks while managing and controlling the communication between them. They will be able to give rights to different members that are responsible for various jobs and documents to make working together in a large team smooth and efficient.

## **Mobile device Accessibility:**

Users will be able to use kord.io on various devices regardless of processing power and screen resolution. The web app will detect the system settings of a connection and display in various formats. A developer might unexpectedly run into someone interested in their idea and they want to be able to access their board from anywhere without needing a computer.

# NON-GOALS

## Never:

- \* Social networking
- \* Full applications such as Photoshop
  - Support can be added via remote desktop

## Possible Considerations:

- \* Large groups
- \* Remote desktop

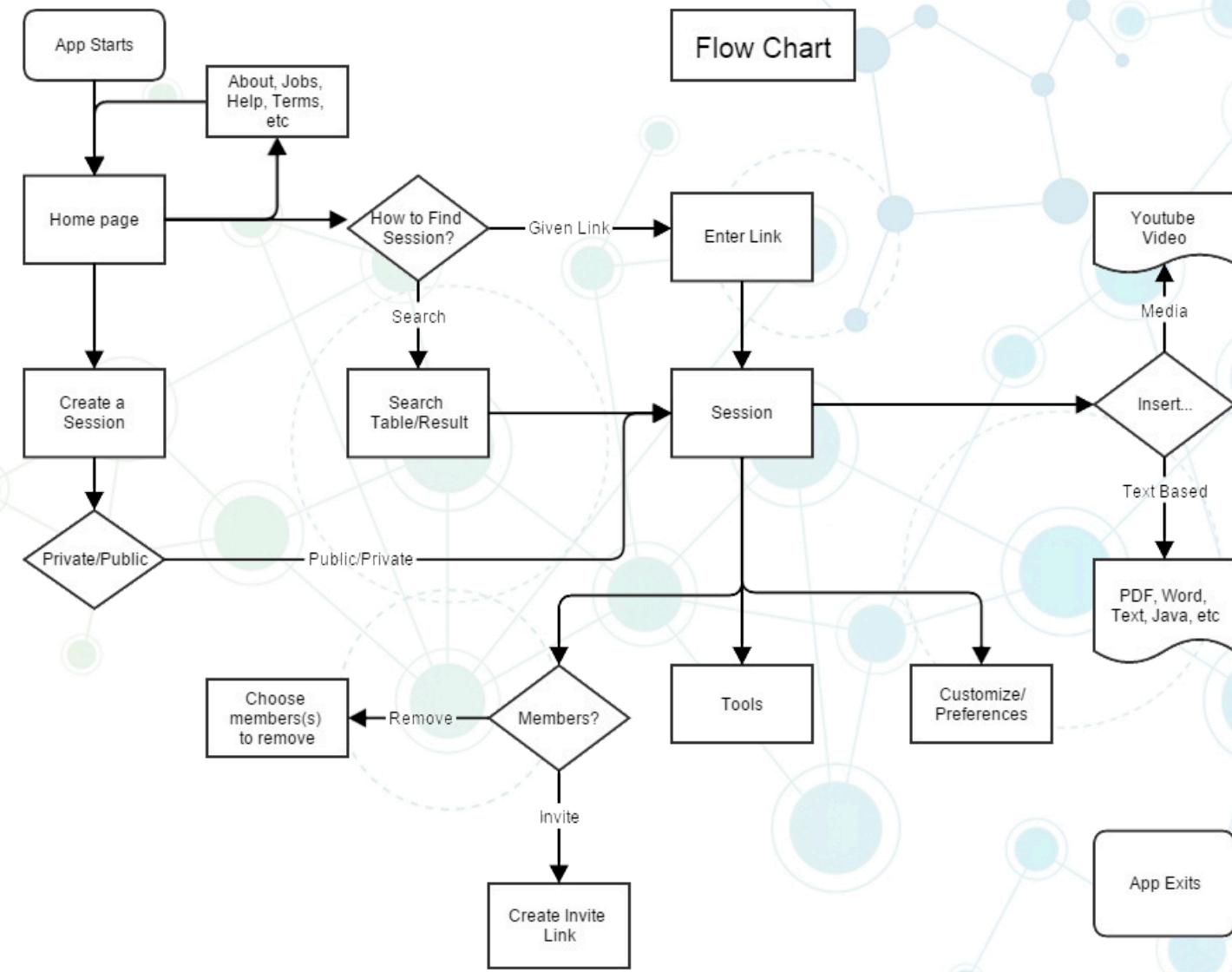
## Time Permitting:

- \* Accounts
- \* Youtube video embedding
- \* PDF embedding
- \* Image embedding
- \* Text Editing / Code Interpreter

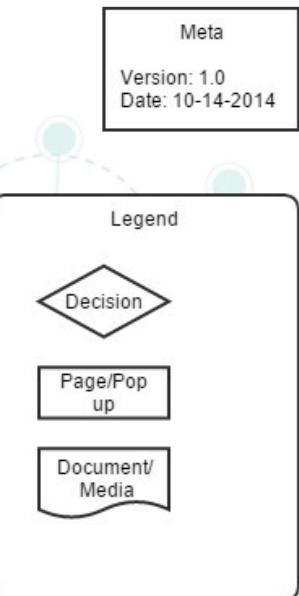
## Features for the Future:

- \* Voice chat
- \* Video conferencing
- \* Screen sharing
- \* Other file format support for embedding
- \* Monetization
- \* Desktop / offline / lan client
- \* Widgets such as calculator
- \* Board history saving

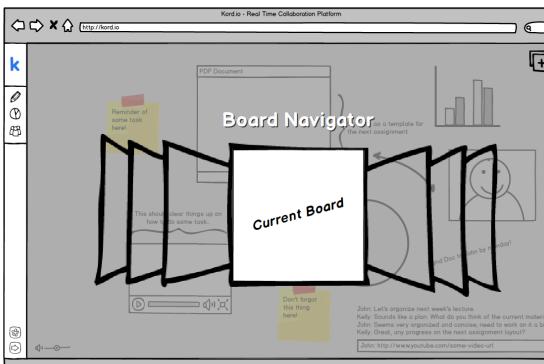
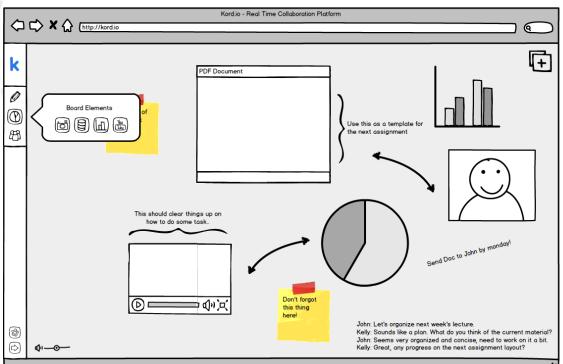
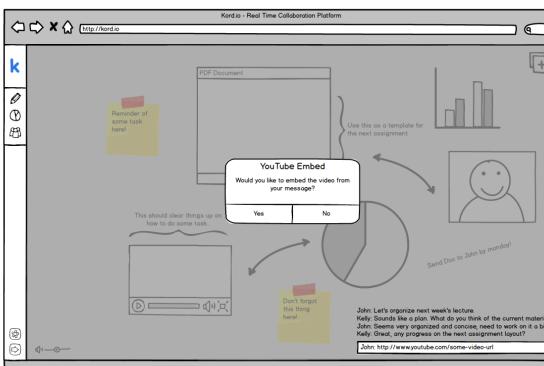
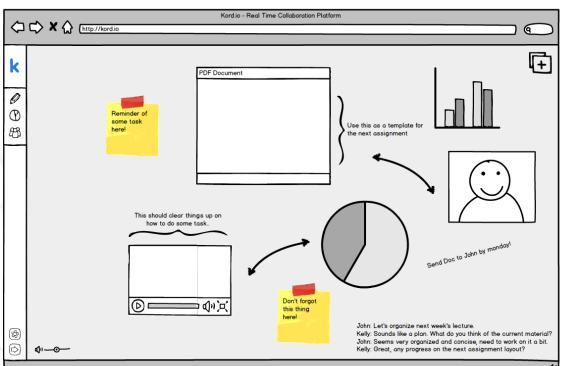
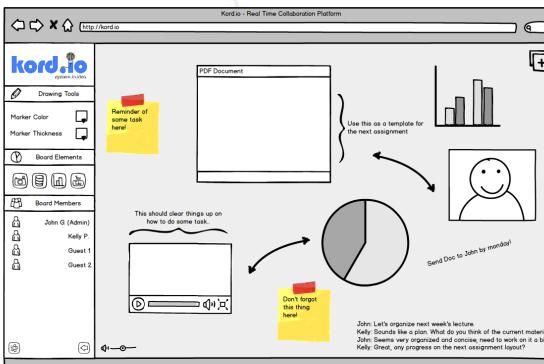
# SYSTEM FLOWCHART



Meta  
Version: 1.0  
Date: 10-14-2014



# SCREEN BY SCREEN



# SCREEN BY SCREEN



kord.io makes it easier than  
kobra join.me Google  
Realtime Board Trello Facebook slack  
to connect your ideas with anyone

create a board

When users connect to <http://kord.io>, they will be welcome by this home page which in its simplest form describes our application and compares it to its competitors. In addition, it will have one button which will take the user to create a new board. Since, for the time being, there will be no user accounts, everyone is a guest and can therefore create a board without logging in. In the future, however, this home page would include the functionality to login and sign-up for an account.

# SCREEN BY SCREEN



Clicking “Create a Board” brings users to this simple page where the functionalities of the board are displayed. Here they will encounter the ability to not only draw on the actual white board, but to also add widgets to it such as a widget where users collaborate to work on a code, share a video, add an image, include a .PDF and also any text file where they wish to work together on. In addition, they will encounter a chat where they will be able to communicate to all of the other users collaborating on the board.

# SCREEN BY SCREEN



Given that we expect users to use the board to its fullest potential and possibly make usage of any widgets they would like to, the menu will be displayed in a minimalistic form (using only the logos), as to not get in the way of what is really important: the work on the boards.



# SCREEN BY SCREEN



Collaborating with other users would not be possible without the possibility to actually chat about what you are doing. kord.io is very aware of this, and includes the feature of chatting with users currently on the board. The chat itself will not only support text, but the ability to include media in it, which ideally will be automatically added to the user's board.

# SCREEN BY SCREEN



github commands:  
- git add file  
- git commit -m "message"  
- git push

```
    };
    /**
     * Custom events
     */
    Cuckoo.Event = {
        bind: function(ev, callback) {
            var calls = this._callbacks || (this._callbacks = [
                var list = this._callbacks[ev] || (this._callbacks[ev] = []);
                list.push(callback);
                return this;
            ]),
            unbind: function(ev, callback) {
                var calls = this._callbacks || (this._callbacks = [
                    if (!ev) {
                        this._callbacks = {};
                    } else if (calls === this._callbacks) {
                        if (!callback) {
                            calls[ev] = [];
                        } else {
                            var list = calls[ev];
                            if (!list) return this;
                            for (var i = 0, l = list.length; i < l; i++) {
                                if (callback === list[i]) {
                                    list[i] = null;
                                    break;
                                }
                            }
                        }
                    }
                ],
                return this;
            }
        }
    };
}
```

need to create  
- http-server  
- http-client  
- user.csv



The chat will remain open as long as the user wishes to have it open. In the case that the chat is closed, users will receive message notifications on the bottom right of their screen which will allow them to quickly glimpse at the message and decide whether to open the message or ignore it. This allows users to stay in the loop about what is going on in the chat but to not be disturbed by it by keeping it open in the board.

Julian Kuk  
Not quite

# SCREEN BY SCREEN



github commands:  
- git add file  
- git commit -m "message"  
- git push

```
    /**
     * Custom events
     */
    Cuckoo.Event = {
      bind: function(ev, callback) {
        var calls = this._callbacks || (this._callbacks = [
          var list = this._callbacks[ev] || (this._callbacks[ev] = []);
          list.push(callback);
          return this;
        ]),
        unbind: function(ev, callback) {
          var calls = this._callbacks;
          if (!ev) {
            this._callbacks = {};
          } else if (calls === this._callbacks) {
            if (!callback) {
              calls[ev] = [];
            } else {
              var list = calls[ev];
              if (!list) return this;
              for (var i = 0, l = list.length; i < l; i++) {
                if (callback === list[i]) {
                  list[i] = null;
                  break;
                }
              }
            }
          }
        },
        return this;
      }
    },
```

need to create  
- http-server  
- http-client  
- user.csv



Here we see an example of a board being used to some of its full potential where a user is engaging with various widgets and collaborating with various users on a board. We can also see three different colored mouse pointers on the screen which belong to three different users currently interacting on the board. This feature allows all users to know what the other users are currently focusing on in the board.

# SCREEN BY SCREEN

The screenshot displays the Kord.io application interface. On the left, there's a vertical toolbar with icons for settings, tools, and sharing. The main area shows three boards labeled 'board 1', 'board 2', and 'board 3'. A fourth board, 'new board', is open in the foreground. This board has a network visualization and some text. To the right of the board is a red note: 'need to create - http-server - http-client - user.csv'. Below the board is a code editor window with the following JavaScript code:

```
Cuckoo.Event = {
  bind : function(ev, callback) {
    var calls = this._callbacks || (this._callbacks = [
      var list = this._callbacks[ev] || (this._callbacks[ev] = []);
      list.push(callback);
      return this;
    ],
    unbind : function(ev, callback) {
      var calls = this._callbacks;
      if (!ev) {
        this._callbacks = {};
      } else if (calls == this._callbacks) {
        if (!callback) {
          this._callbacks = {};
        } else {
          var list = calls[ev];
          if (!list) return this;
          for (var i = 0, l = list.length; i < l; i++) {
            if (callback == list[i]) {
              list[i] = null;
              break;
            }
          }
        }
      }
      return this;
    }
  }
};
```

Below the code editor is a GitHub integration window titled '1.4 - Adding Changes' showing a commit message from 'Geek Radio'.

**github commands:**

- git add file
- git commit -m "message"
- git push

Finally, but not least important, we see the option to switch among boards. Sometimes one board is not enough! By clicking the plus icon on the top right corner, users will be able to add new boards to work on or delete any boards not being used. Needless to say, the tools icon on the bottom left allows users to control the settings of the boards, such as whether a board is public or private, and also to create a link to share their board with other users.

# LIST OF OPEN ISSUES

## **Number of Members Limited to a Session**

Having to think about how much activity each member can perform in a session with x other members, finding a balance for limiting the number of members to a session will be left to trial and error.

## **Saving a Session for Future Use**

We want users to be able to save a session in some way, at first we may start with saving the board as a .PDF and then maybe eventually move to saving the board state.

## **Different Resolutions**

A big issue is figuring out how to handle users running our app on different resolutions. Whether we want a static window with scrolling or screen pulling, or a resizable window.

## **User Customization of Tools and GUI**

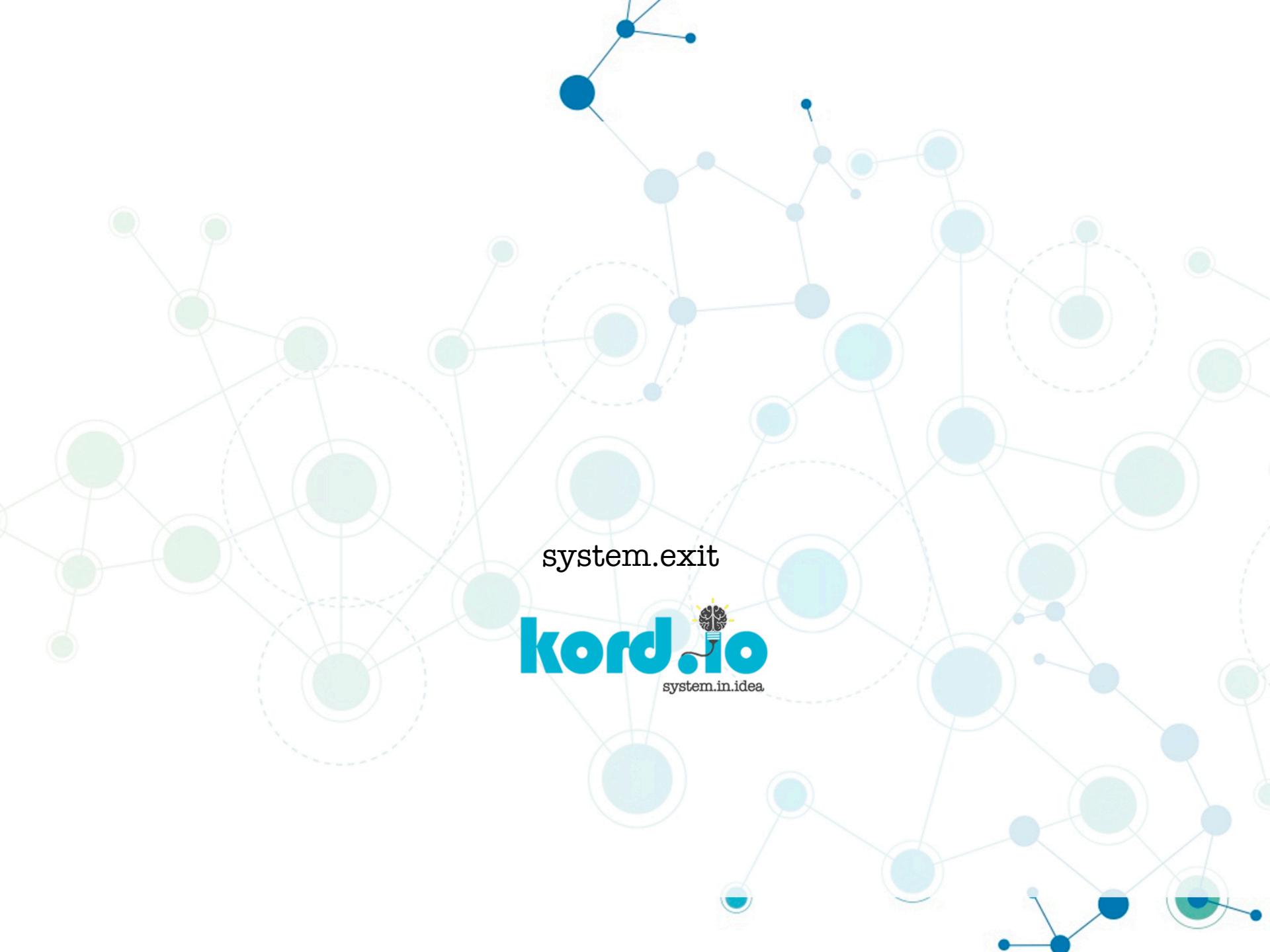
We want to give our users as much freedom as possible while keeping our UI very simple to operate, so deciding how much freedom to give to users will be an issue for trial and error.

## **Board Layers**

With multiple users means that we will run into a layering issue when multiple people interact with the board, deciding who gets priority and synchronization.

## **Exploit Prevention**

Making sure our app is less prone to exploits.



system.exit

