



## Vi-Datec Frontend++ Seniority Test

# Summary

The test consists of the development of a dummy email client app.

The app must:

- Use Angular for UI rendering and Redux for state management (NGRX).
- Follow the [Eva Design guidelines](#). It may make use of existing implementations like [Nebular](#).
- Be responsive down to mobile phone screen sizes.
- Use CSS flexbox / CSS grid for structuring and positioning elements
- Use an ESLint config and a npm command to lint all files.
- Have a README file.
- Be stored in GitHub or GitLab.
- Be hosted in [Heroku](#), [Now](#) or [AWS](#).

# Screens

The app consists of a simple email client with 3 screens:

Note: only use the mockups as a reference point. Feel free to mix it up.

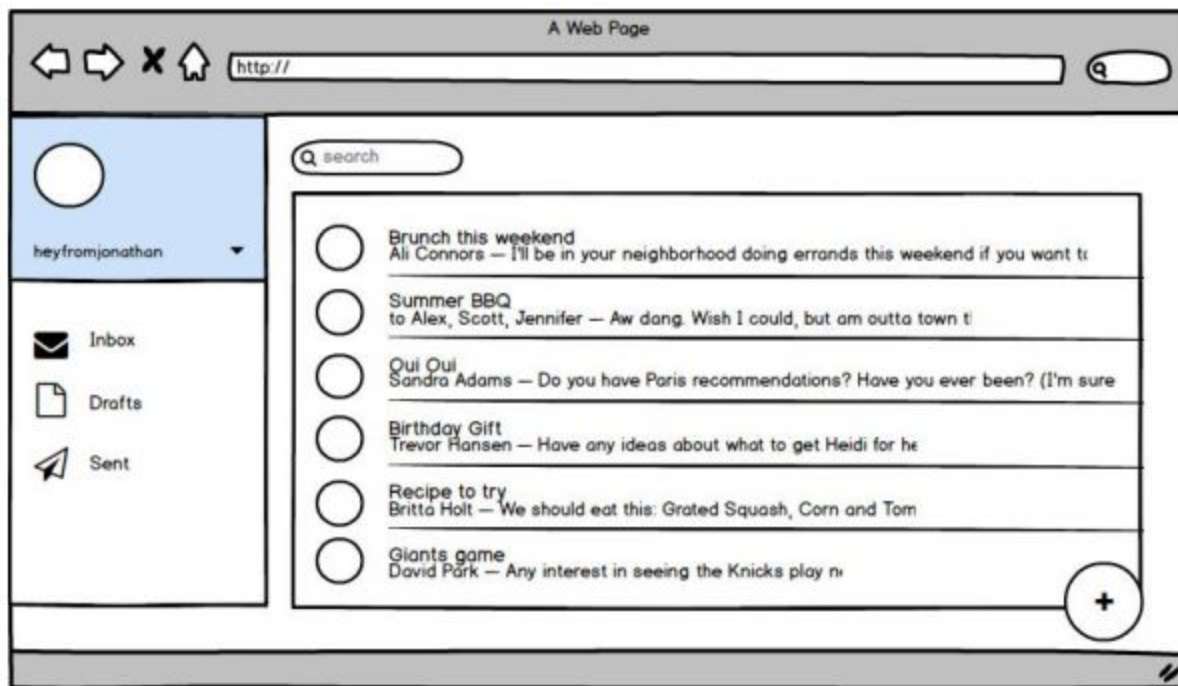
# Login

A hand-drawn diagram of a web browser window. The title bar at the top says "A Web Page". Below the title bar is a navigation bar with icons for back, forward, close, and home, followed by an address bar containing "http://". The main content area of the browser displays a "Login" form. The form is a light gray rectangle containing the following elements:

- The word "Login" in a large font at the top left of the form.
- An "Email:" label followed by a text input field.
- A "Password:" label followed by a text input field.
- A "Login" button below the password field.

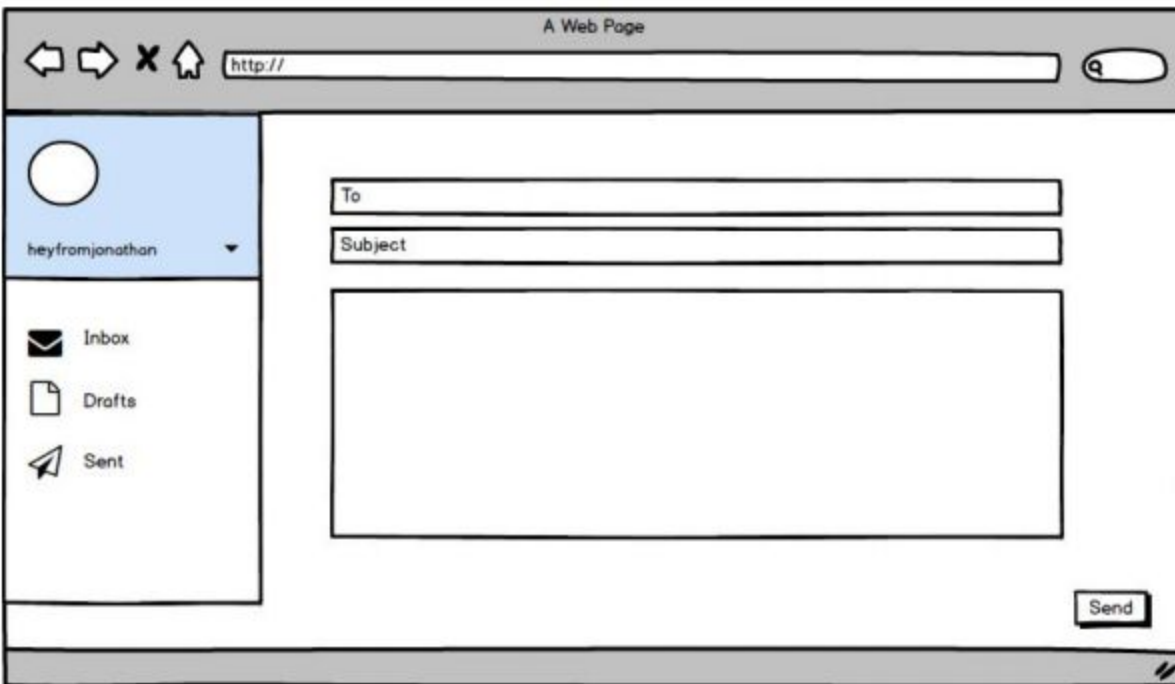
- Disallow empty fields.
- Disallow invalid emails.
- Reject email + password combinations as invalid, other than:
  - Email: user@vi-datec.com
  - Password: test
- Proceed to the *Inbox* section of the *Main* screen once logged in.

# Main



- On the left, the navigation menu allows switching between the *Inbox*, *Drafts* and *Sent* sections.
- On the right, a list of emails is displayed according to the current section.
- The list must initially show 20 messages and have a way to show all messages through pagination or infinite scroll.
- While in the *Inbox* and *Sent* sections, clicking on an email opens the *View* screen.
- While in the *Drafts* section, clicking on a draft opens the *Compose* screen.
- Clicking the plus button on the bottom right opens the *Compose* screen.
- Bonus points if the list can be searched by email and name.

## View / Compose



- On the left, the navigation menu allows switching between the *Inbox*, *Drafts* and *Sent* sections of the *Main* screen.
- On the right, the screen contains a form with the fields: *To*, *Subject* and *Body*.
- In the *View* screen, all three fields are read only and the *Send* button is hidden.
- The *To* field is auto-completed from the email data.
- While composing an email, the app must save a draft every 10 seconds.
- The *Drafts* section of the *Main* screen must only show the last draft of every unsent message.
- After clicking the send button the message should appear in the *Sent* section and disappear from the *Drafts* section.

## Bonus points

- Include [Storybook](#) as documentation of your components
- Use a real backend, preferably NesJS or another NodeJS framework
- Dockerize your app
- Use `@ngrx/entity` library

# That's it!

If all that was easy enough for you and you want to show us more... do something unique to wow us. **Good luck!**