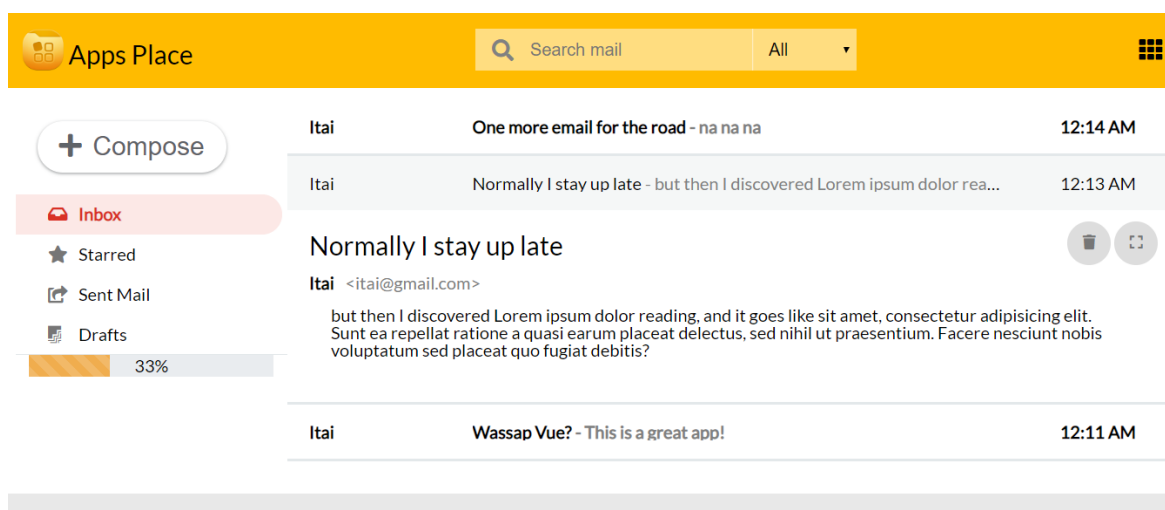## The misterEmail *App*

Requirements

Create an email client app. Note – this app will not work with a real server, so emails are not really sent, but the user experience will be the same.

Features:

- List of emails (inbox)
- Present the listed emails as read/unread
- Click email at list – opens the email for reading
- Show unread emails count on top (find the right place)
- Ability to mark as read/unread
- New email – ability to create a new email and send it.
- Search emails
- Filter read/unread
- Sort of emails by date, title
- Reply – allows editing, adding "Re:" to subject
- Integration with KeepApp (link to edit and save an email as a note)

UX

Mobile Version

Functional Design

- Model - Start with a basic model of emails:

```
{subject: 'Wassap?', body: 'Pick up!', isRead: false, sentAt : 1551133930594}
```

<email-app>

- Gets emails from service (asyc)

<email-list>

- renders a list of <email-preview> pass down an email prop

<email-preview>

- Has an email prop
- Renders the subject
- Gives visual indication for read/unread (i.e.: bold/unbold ; closed or open envelop)

<email-details>

- Routable component (page)
- show the entire email
- allow deleting an email (using a service)

<email-status>

- Renders how many read from the emails

<email-filter>

- Allow filtering by text and Read / Unread

<email-compose>

- Has a form with subject and body
- Use the service to add email to the list
- Yes, we are only supporting selfi-emails for now (-:

Integration

- Allow keeping an email as a note in missKeep - (use queryString Params)
- Use a <long-text> component - gets the text to format as a prop. (this component is used by both apps)
- Use a <user-msg> component for showing success / error messages
  (this component is used by both apps)