coding_
academy

The misterEmail *App*

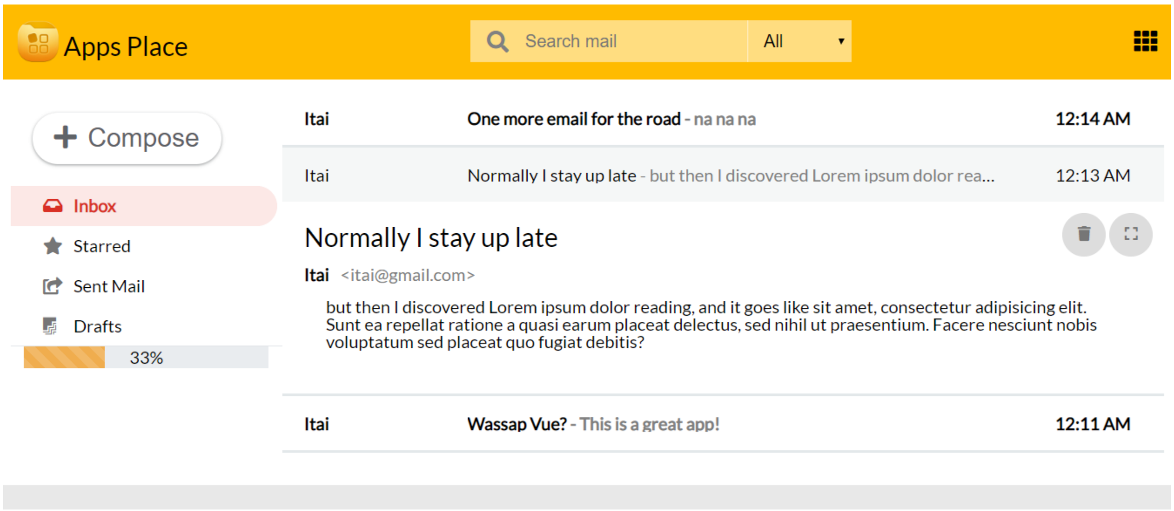
Requirements

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



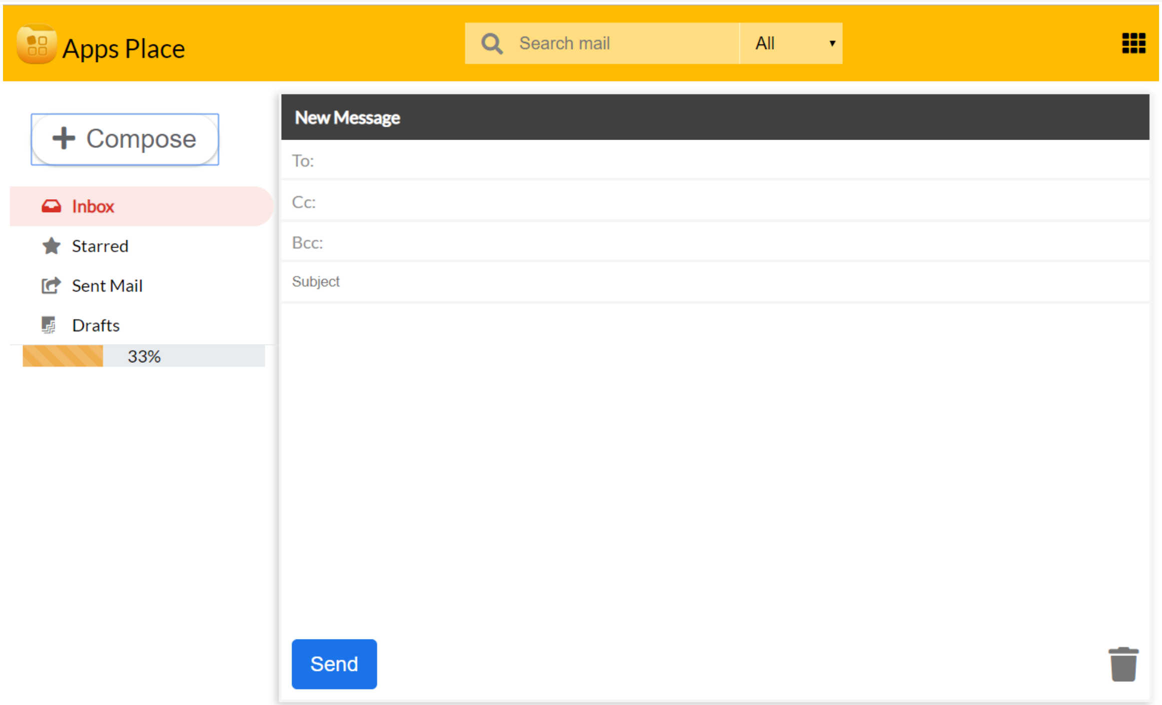
Itai	**One more email for the road** - na na na	**12:14 AM**
Itai	Normally I stay up late - but then I discovered Lorem ipsum dolor rea...	12:13 AM

Normally I stay up late

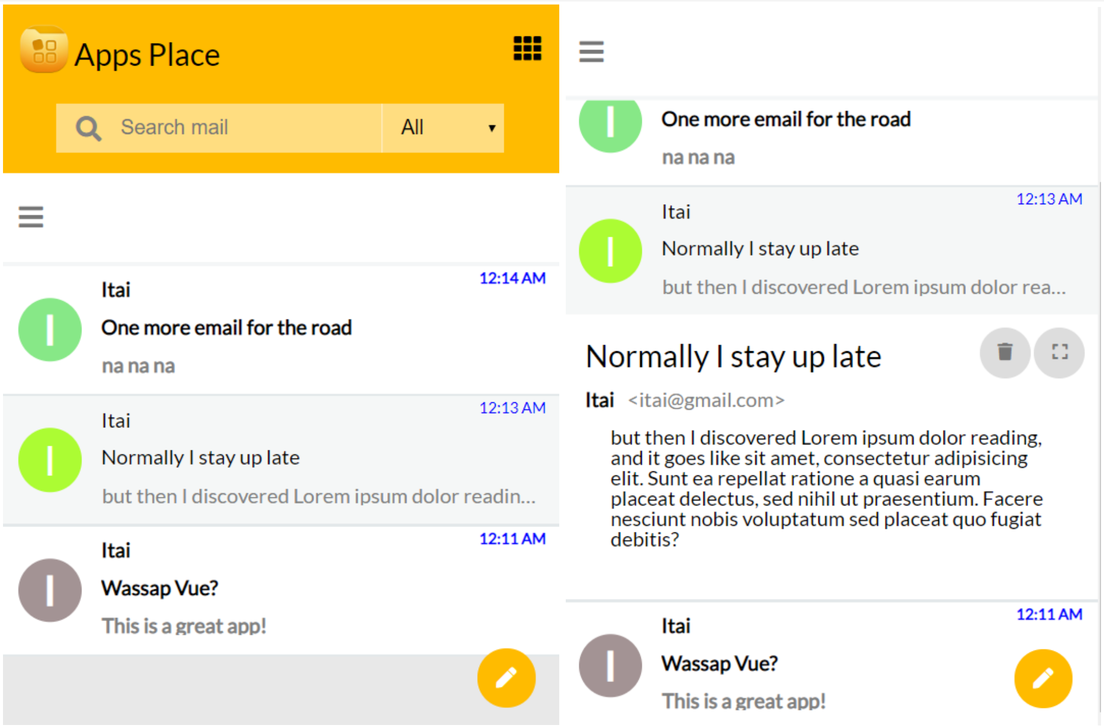
Itai <itai@gmail.com>

but then I discovered Lorem ipsum dolor reading, and it goes like sit amet, consectetur adipisicing elit. Sunt ea repellat ratione a quasi earum placeat delectus, sed nihil ut praesentium. Facere nesciunt nobis voluptatum sed placeat quo fugiat debitis?

Itai	**Wassap Vue?** - This is a great app!	**12:11 AM**



Mobile Version



Itai 12:14 AM
One more email for the road
na na na

Itai 12:13 AM
Normally I stay up late
but then I discovered Lorem ipsum dolor readin...

Itai 12:11 AM
Wassap Vue?
This is a great app!

One more email for the road
na na na

Itai 12:13 AM
Normally I stay up late
but then I discovered Lorem ipsum dolor rea...

Normally I stay up late
Itai <itai@gmail.com>
but then I discovered Lorem ipsum dolor reading, and it goes like sit amet, consectetur adipisicing elit. Sunt ea repellat ratione a quasi earum placeat delectus, sed nihil ut praesentium. Facere nesciunt nobis voluptatum sed placeat quo fugiat debitis?

Itai 12:11 AM
Wassap Vue?
This is a great app!

Functional Design

- Model - Start with a basic model of emails:

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

<EmailApp>

- EmailApp Gets emails from service (asynch)

<EmailList>

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

<EmailPreview>

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

<EmailDetails>

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

<EmailStatus>

- Renders how many read from the emails

<EmailFilter>

- Allow filtering by text and Read / Unread

<EmailCompose>

- 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 <LongText> component - gets the text to format as a prop. (this component is used by both apps)
- Use a <UserMsg> component for showing success / error messages
  (this component is used by both apps)