



# buyDeis

## Final Presentation

Team 6 - Fatumata Jalloh, Yalda Mauj, Sophia Wang



# buyDeis Team



Fatumata Jalloh



Yalda Mauj



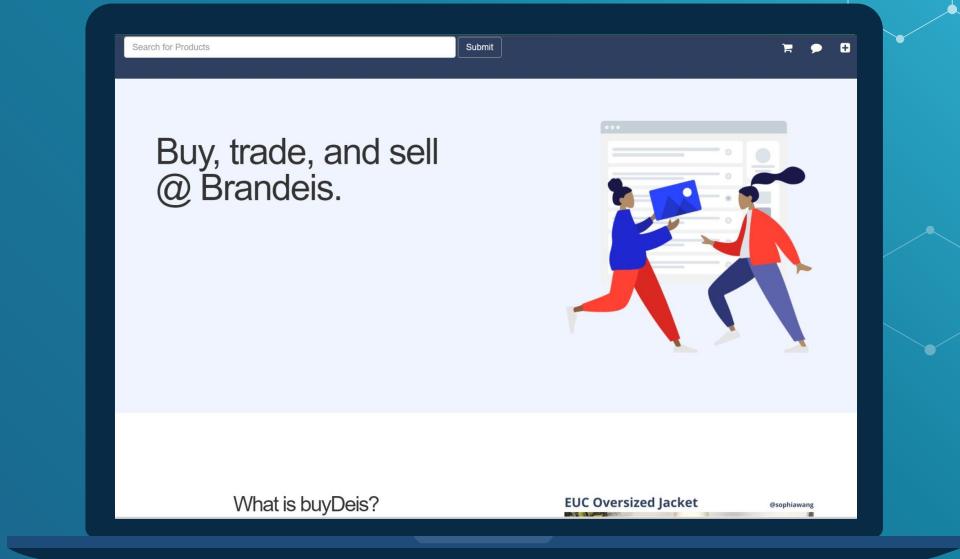
Sophia Wang



1

# What is buyDeis?

App introduction



“

buyDeis provides a platform  
for Brandeis students to  
practice sustainability by  
selling goods that they don't  
use/need

# buyDeis...

- ◆ Allows users to create posts and see the feed for items that are available for other users to purchase
- ◆ Gives users the option to search for specific products and filter products on their feed
- ◆ Allow users to follow each other
- ◆ Allows users to perform transactions
- ◆ Allows users to leave reviews for other users that they have purchased from
- ◆ Allows users to like/unlike products and follow other users
- ◆ Allows users to message other users about their products
- ◆ Allows users to have a shopping cart and buy products from their shopping cart



3

What is the technical  
architecture of our  
application?

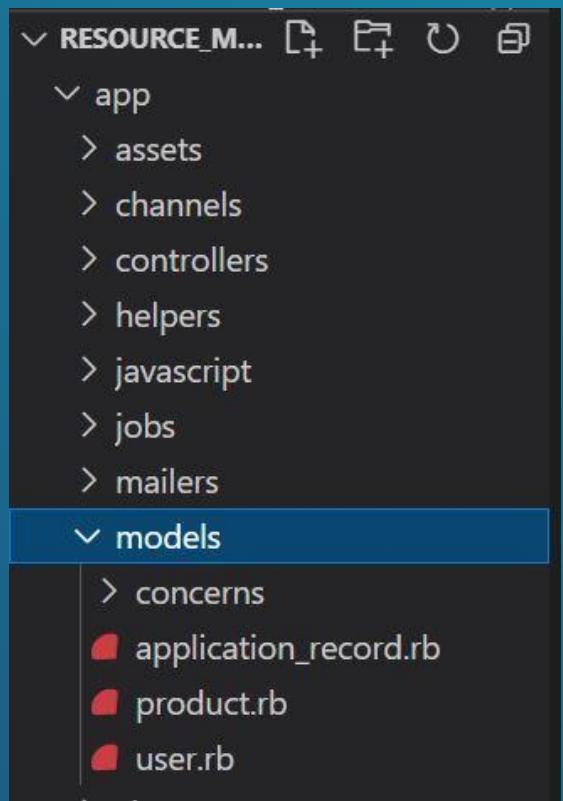
# Model

User
first
last
email
username
password_digest
List<> products

one-to-many

Product
name
description
price
quantity
image
user_id

# Model



# View - pt 1

\* views/conversations

index

Lightweight Concrete Gloves  
Lightweight Leather Watch  
Sample  
New Conversation

new

kendall\_rogahn ▾  
Subject  
Leave a comment  
  
Save changes

show

sophiaW I want to buy your product  
sophiaW hello fatumata!  
Fatumata Hey you can't buy it  
sophiaW :-(

Save changes



\* views/products

index

\_listAll

\_form

new

Create your Product

Name  Price   
Image  No file chosen  
Description   
submit

\* views/layouts

application

\_footer

\_navbar - top tab navigation  
partial

buyDeis Search for user or prod. Search

Messages createProduct You

# View - pt 2

\* views/**sessions**

-**new** - login page form

Sign in

Email

Password

**Sign In**

Don't have an account? [sign up](#)

\* views/**users**

**\_form** - partial for signup form

**new** - signup page, form partial is called here

**index, show, edit** - users to list, destroy, edit (admin)

Create your account

First  Last

Email

Username

Password

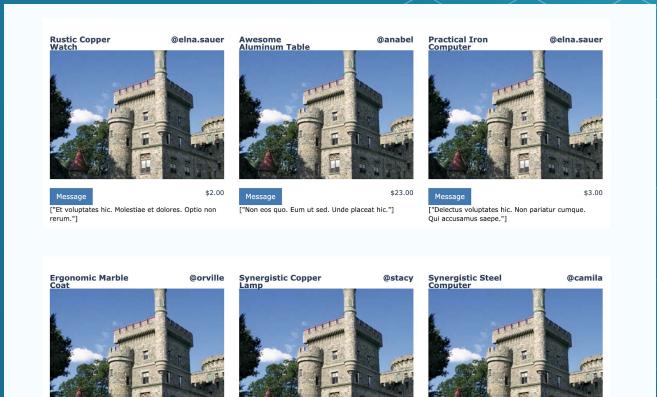
**submit**

Already have an account? [log in](#)

\* views/**welcome**

- **index** - Where users are directed shows all products posted with partial \_listAll in products

```
<%= render '/products/listAll' %>
```



# View

```
views
  conversations
    index.html.erb
    new.html.erb
    show.html.erb
  layouts
    _footer.html.erb
    _navbar.html.erb
    application.html.erb
    mailer.html.erb
    mailer.text.erb
  products
    _form.html.erb
    _listAll.html.erb
    index.html.erb
    new.html.erb
```

```
sessions
  new.html.erb
users
  _form.html.erb
  _user.json.jbuilder
  edit.html.erb
  index.html.erb
  index.json.jbuilder
  new.html.erb
  show.html.erb
  show.json.jbuilder
welcome
  index.html.erb
```



```
✓ RESOURCE_M...  ⌂ ⌃ ⌄ ⌅ ⌆
  ✓ app
    > assets
    > channels
  ✓ controllers
    > concerns
    application_controller.rb
    conversations_controller.rb
    messages_controller.rb
    products_controller.rb
    sessions_controller.rb
    users_controller.rb
    welcome_controller.rb
```

# Controller

- Application Controller - for welcome page.
- Welcome Controller - does not contain anything here but is utilized in the views folder as the home page
- Product Controller allows a logged in user to create a new post about an item to add to the app
- User controller allows a person to create a new account (sign up), delete an account or edit an account.
- Session controller allows the user with an existing account to log in and log out
- message controller create message that is part of a conversation. While the conservation controller creates a conversation between the user and the recipient using the mailboxer gem

5

# Technology: Gems, Integrations

## Models

Cart

Category

Condition

Conversation

line\_items

message

product

rating

relationship

user

## Controller

carts

categories

checkout

conversation

line\_items

messages

products

ratings

relationships

sessions

stripe

user

welcome

## Views

carts

categories

checkout

conversations

front

layouts

line\_items

products

ratings

relationships

sessions

stripe

users

# Gemfile

## ❖ Mailboxer

➤ Mailboxer allows any object created with a `acts_as_messageable` property to have a mailbox. This means they can send and receive messages. They can also have conversations with other users. This is implemented with the messages and conversations controllers and views.

## ❖ Active storage

➤ Active storage allows us to upload files, in our case images, and store the data of the files. By default, it stores the data locally, which is why we used Google Cloud to store it on the cloud.

## ❖ Google cloud storage

➤ We used Google Cloud Storage and created a bucket to store all images uploaded on the site. We added this to our `storage.yml` file with the specific encryption data given by Google so all images stored in the site are saved.

## ❖ Bcrypt

➤ Allows us to encrypt user passwords and ensures all user password created are correct



# Gemfile

- ◇ Stripe, omniauth-stripe-connect, oauth2
  - ◆ These gems allowed us to connect with the stripe UI to integrate transactions. The oauth2/omniauth-stripe-connect allowed for authorization with these transactions
- ◇ Searchkick (with Elasticsearch)
  - ◆ Searchkick allows for more advanced searching and filtering. It accounts for misspelled search input and incomplete input. It also allows for compound filtering with the products
- ◇ Acts\_as\_votable
  - ◆ Allows users to like/unlike a product and helps track number of likes and dislikes
- ◇ Cancancan, rails-admin
  - ◆ Allowed us to verify views. Users cannot see restricted views

6

# Testing



# Testing

- ◊ minitests for products and users
- ◊ Views testing
- ◊ TDD

7

What are we proud of the most?



# Integrating Stripe and Searchkick

- Stripe
  - Integrating Stripe UI and getting the functionality to have secure transactions on the website
- Searchkick
  - We used elasticsearch to integrate searchkick which allowed us to have search functionality that allows us to search for products even with misspelling errors or incomplete search input. This also allows us to filter the products by category, condition, price, and date. The filtering was also compounded.



8

Where can our product be improved or expanded?



# Features we could add in the future

- ◊ In app notification
  - ◆ New messages, follows and likes
- ◊ Email notification
  - ◆ When a new product is purchased
- ◊ Comments on posts
- ◊ Resetting password
- ◊ Using only Brandeis emails to sign up
- ◊ Trading



9

# Group Structure



# Deployment and Integration

- ◊ We deployed multiple times a week.
- ◊ We met every sunday and made sure we all had the same version of the app on our individual laptops.
- ◊ We met with Julian every Wednesday to make we were on track



# Working in a group

- ◊ We met every week and organized the trello to reflect the work we did
  - ◊ Sophia was in charge of figma
  - ◊ Fatumata was in charge of backend
  - ◊ We each came with different backgrounds and have different skill sets and experiences
- We always had a timeline for what we were working on and were expected to complete it by the beginning of each sunday



# THANKS!

