

App URL

<https://sleepy-river-56934.herokuapp.com/>

Pristine Master Repo (Margaret)

<https://github.com/MargaretDobson/oneoff3>

Forked Repos

Danny - <https://github.com/DanBytes/oneoff3>

Matt - <https://github.com/Manik747/oneoff3>

Oneoff Description

Problem definition / purpose

One off is designed to be a two a beautiful two sided marketplace where artist's, collectors, craftsmen and antiques dealers can sell beautiful one-off items. Users are able to follow other users so that their favourite items and artists are always just a click away. Beyond just being a simple marketplace One off is meant to be a source of inspiration and entertainment. Borrowing key ideas from the likes of Pinterest and Instagram where users can simply browse content for their leisure.

Functionality / features

The app has a number of features that allow users to connect, be inspired and purchase products from one another.

- User Authentication through devise
- The ability to follow other users through the global feed
- A specialized feed of users that the current user is following
- Make secure payments for goods through Stripe which automatically makes the product unavailable on the platform
- Users are able to message other users socially or to organise shipping
- All users are able to post products and make them available for sale or just share what they have worked on

Screenshots

One-Off

Products

[Log out](#) | [New Product](#) | [Conversations](#) | [Followed Feed](#)



come2papa

Apple Artwork

Excellent artwork for your home decoration!

Available : \$ 2000.0

[Buy with card](#)

[Follow](#)



Banana_man

A Beautiful piece of banana art

A visually appealing piece of art

Available : \$ 200.0

[Edit](#)

[Destroy](#)

One-Off

Products

[Log out](#) | [New Product](#) | [Home](#)



Apple Artwork

Excellent artwork for your home decoration!

Available 2000.0

[Following](#)

Mailbox

[Home](#)


[Papa John](#)
All Users

Whiteford [Message me!](#)
come2papa [Message me!](#)

One-Off

ts

oduct | Conversations | Followed Feed



ork
ork for your home

000.0

Banana_man

A Beautiful piece of banana art

A visually appealing piece of art

Available : \$ 200.0

[Edit](#) [Destroy](#)

Apple Artwork

banana@banana.com

Card number

MM / YY CVC

☐ Remember me

Pay AS\$2,000.00

[Terms](#) | [Privacy](#) | [i\\$teef](#)

One-Off

Conversations

come2papa 11/08/18 at 2:42 AM

Hey I am in your yellow banana.

Banana_man 11/08/18 at 3:28 AM

Woot I love your apple work!!!!

Add Reply

Tech stack

Ruby on Rails

Framework for creating app

PostgreSQL

Database for managing records

Heroku

Used for deploying the app to the web

Set-up instructions

For local deployment

Requirements needed for app

- Rails installed
- Cloudinary account with relevant API keys
- Stripe account with relevant API keys

Ensure you have configured with passwords your *db/database.yml*, API keys for cloudinary *db/storage.yml* and API keys for *config/initializers/stripe.rb* with an ENV file or by placing keys straight into file.

In command line

Bundle install

Rails db:setup

Rails server should now start an instance of the app.

Consult documentation from relevant cloud based hosting service for web deployment.

Design documentation.

Design process

We wanted to build an elegant two sided marketplace so we started to look around for well crafted web applications and we ended up with instagram as our main inspiration.

The challenge was how do we add an marketplace to an image sharing platform. So we started doodling. See process below:

Defining User Actions

How does the user interact with the application.

Anonymous (Unregistered User)

- Can see an example of what the marketplace looks like
- As a user I can set up my profile
 - Name
 - Bio / Description
 - Avatar
 - Banking Details-Stripe should handle this

User (Registered User)

- As a user I can post a product online
 - Can edit
 - Can delete
 - Can toggle sold status
- As a user I can edit profile
 - Bio / Description
- As a user I can buy a product
- As a user I can follow other users.
- As a user I can like and comment on a product
- As a user I need to see if the product is available for purchase if not it will say unavailable.
- Flag for inappropriate posts and comments for review by an Admin
- A user can delete own account

Admin

- Can delete comments and post
- Can delete accounts
- view and review list of flagged content

How does user experience the site

If user when not logged in they are presented with index page.

- index page links to log in or sign up pages.
- log in page has username and password
 - when authenticated the user is redirect to global explore page
- sign up page has form containing:
 - username
 - password
 - email
 - avatar
 - bio

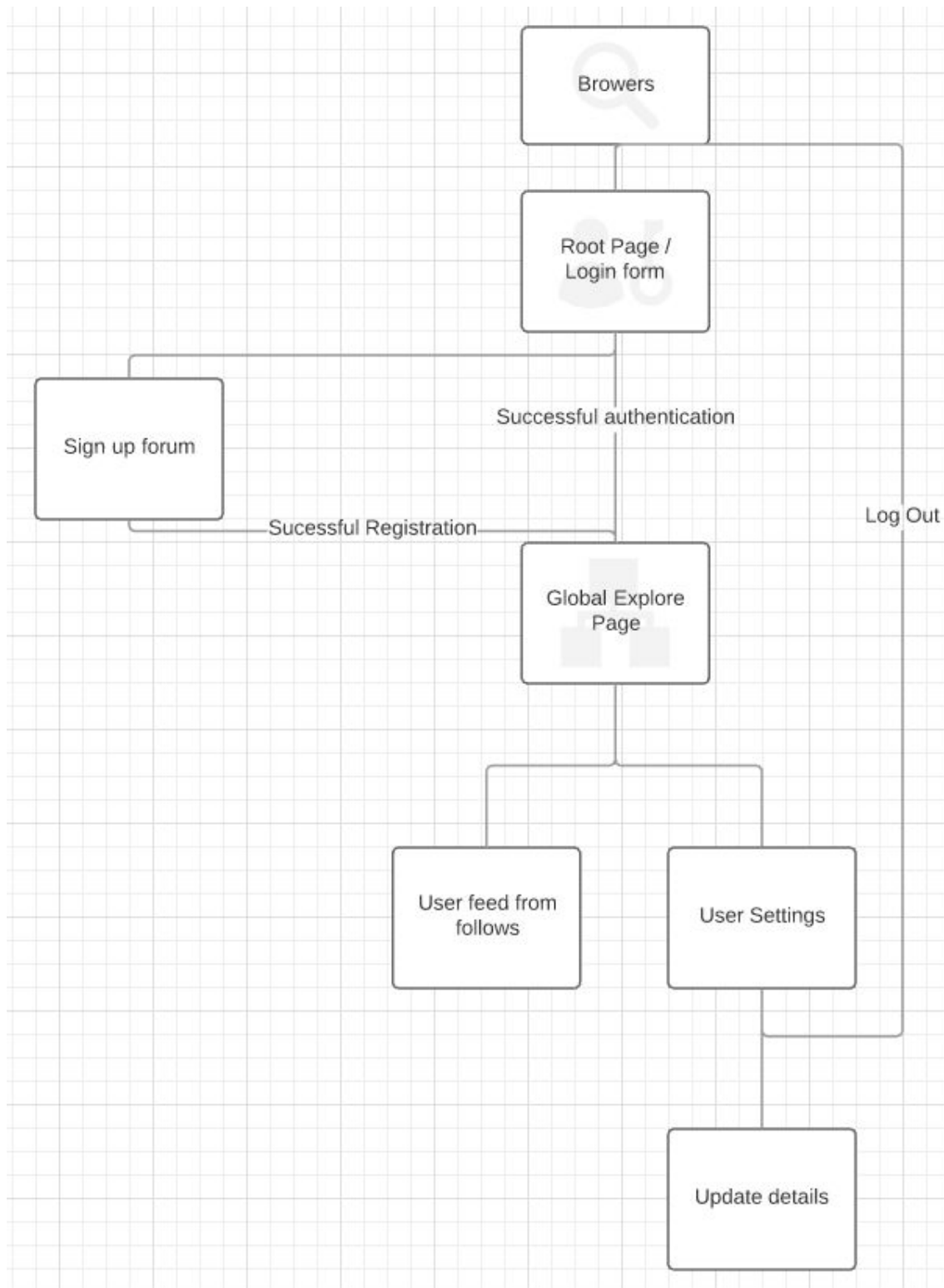
Else direct them to global explore page.

- global explore page contains all posts from the users
- each post can be like by users
- user can follow another user

- global explore page has links to personal feed
- personal feed page has posts from users that the user follows
- global explore page has link to settings
- user can create new post
 - post has image
 - post has description
 - post has price

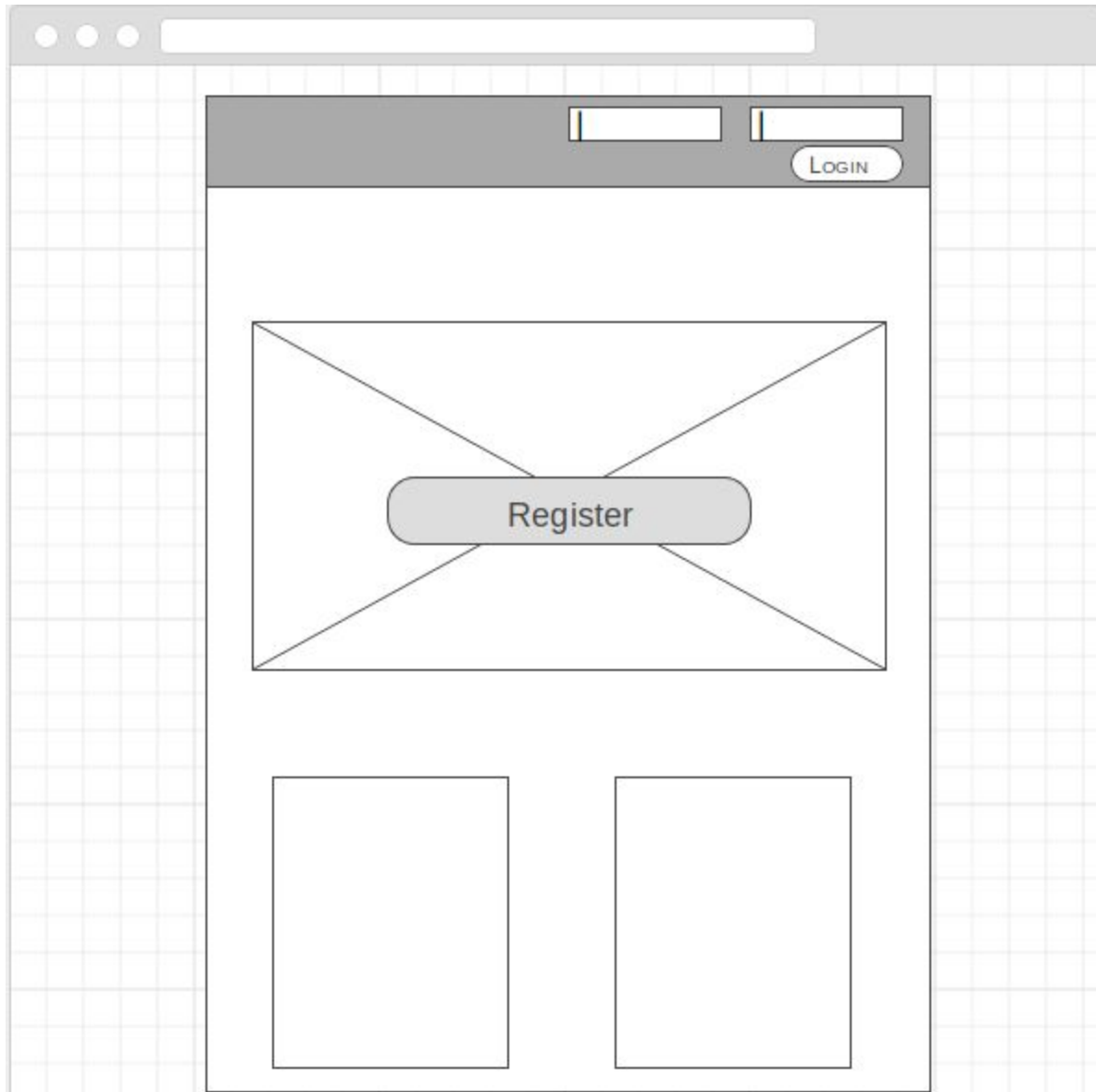
- can edit existing post
- delete existing post
- user can edit bio
- user can change avatar

Rough Application Flow Sketch

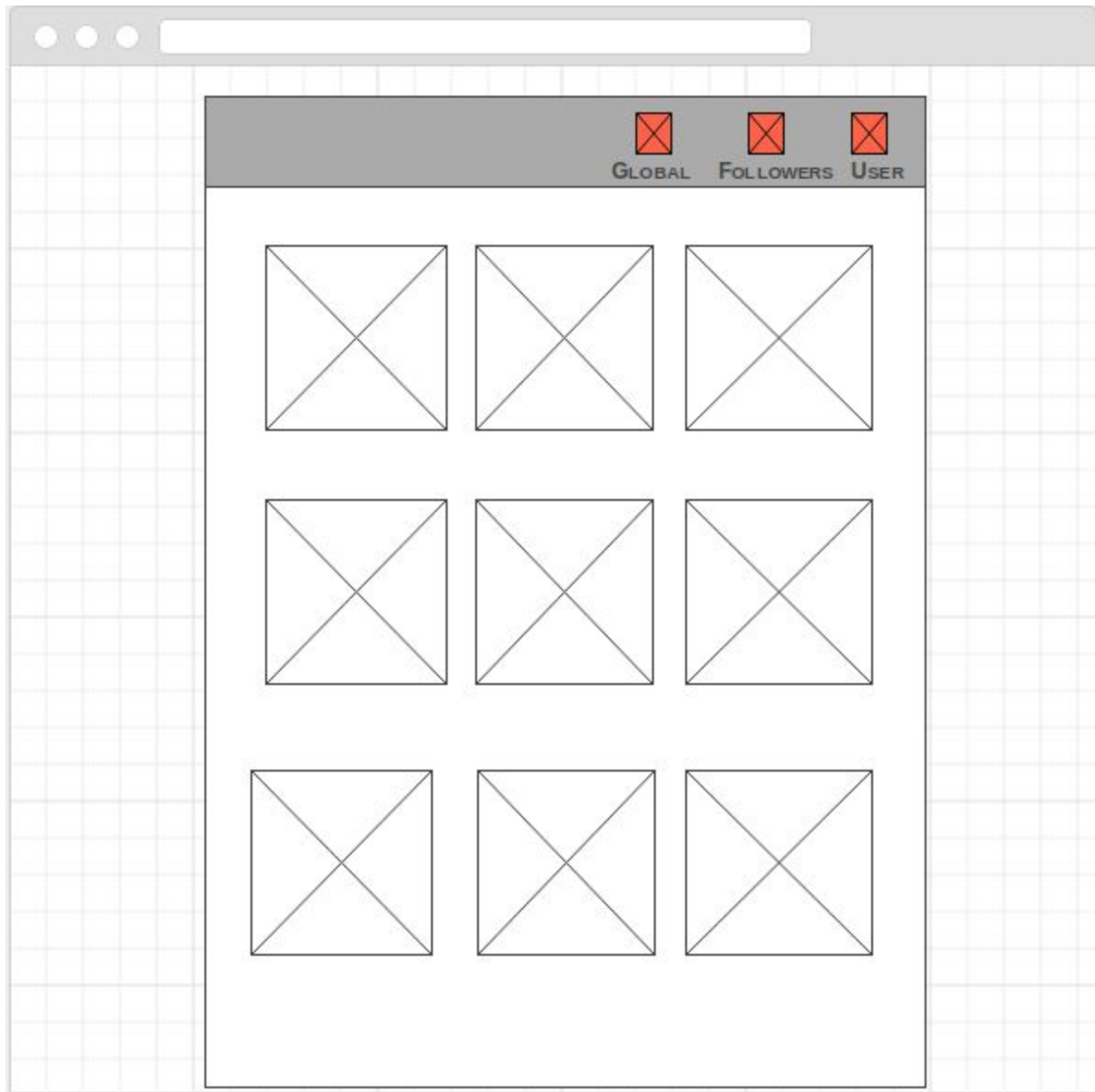


Wireframes

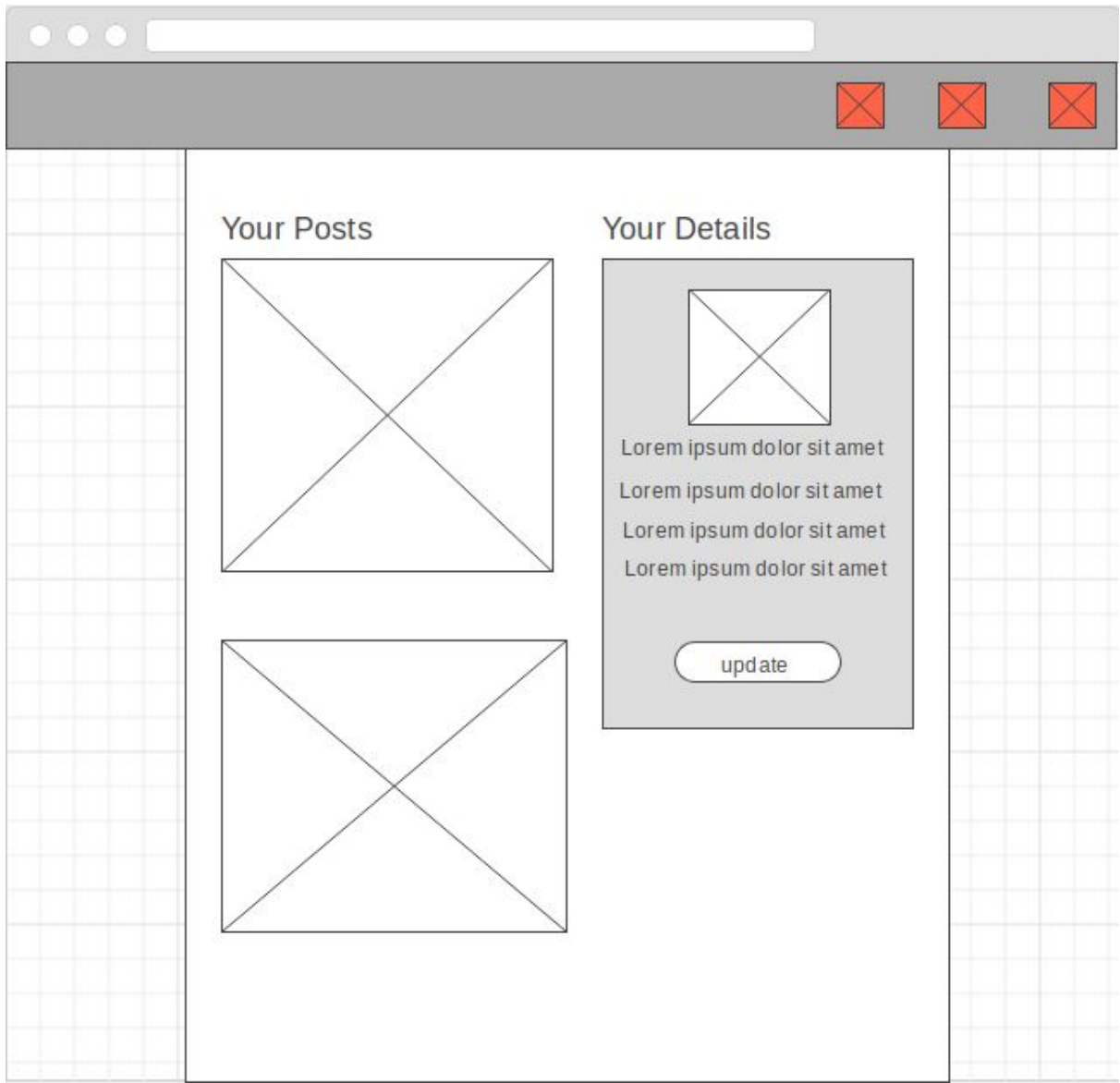
Root page



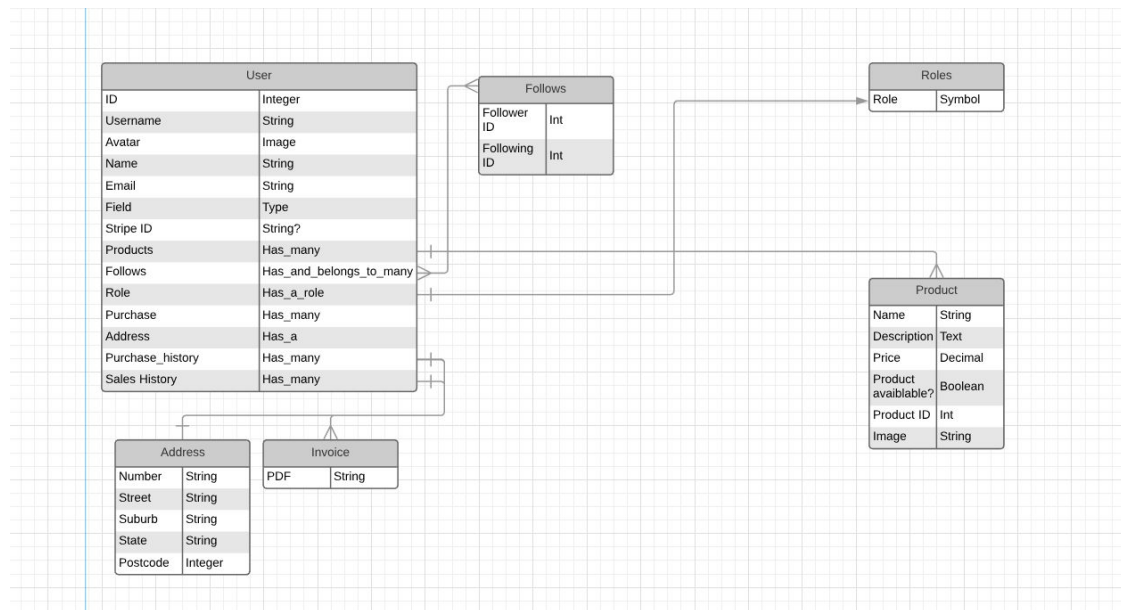
Global Page / Followers Page (shared layout)



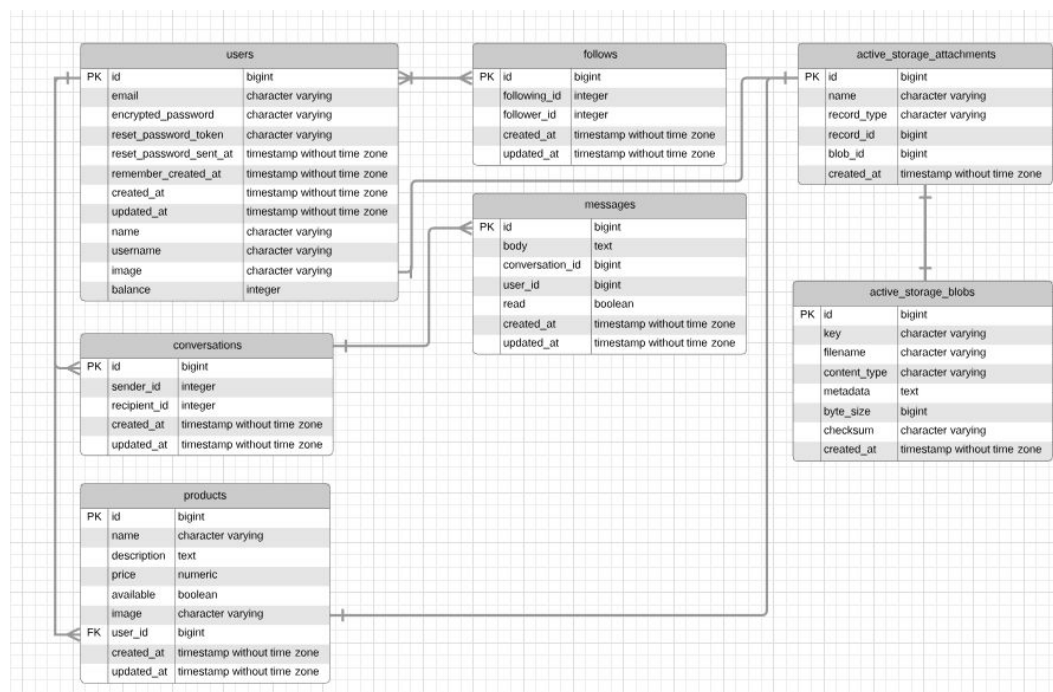
Wireframe - User's Profile View



Database Entity Relationship Diagrams (Plan)

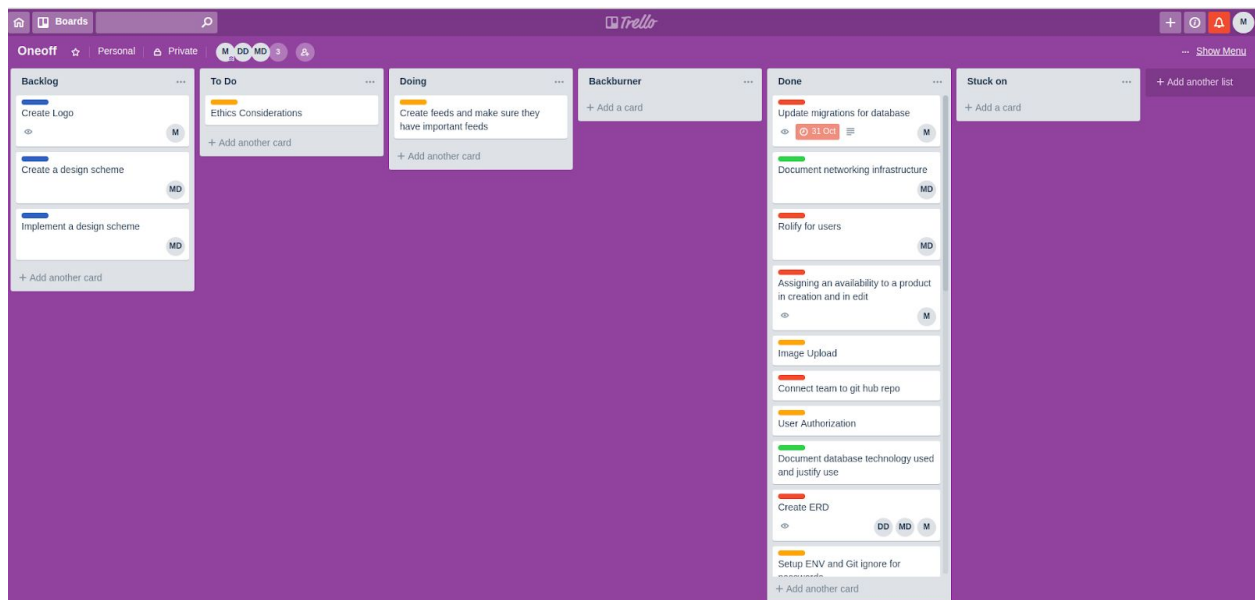


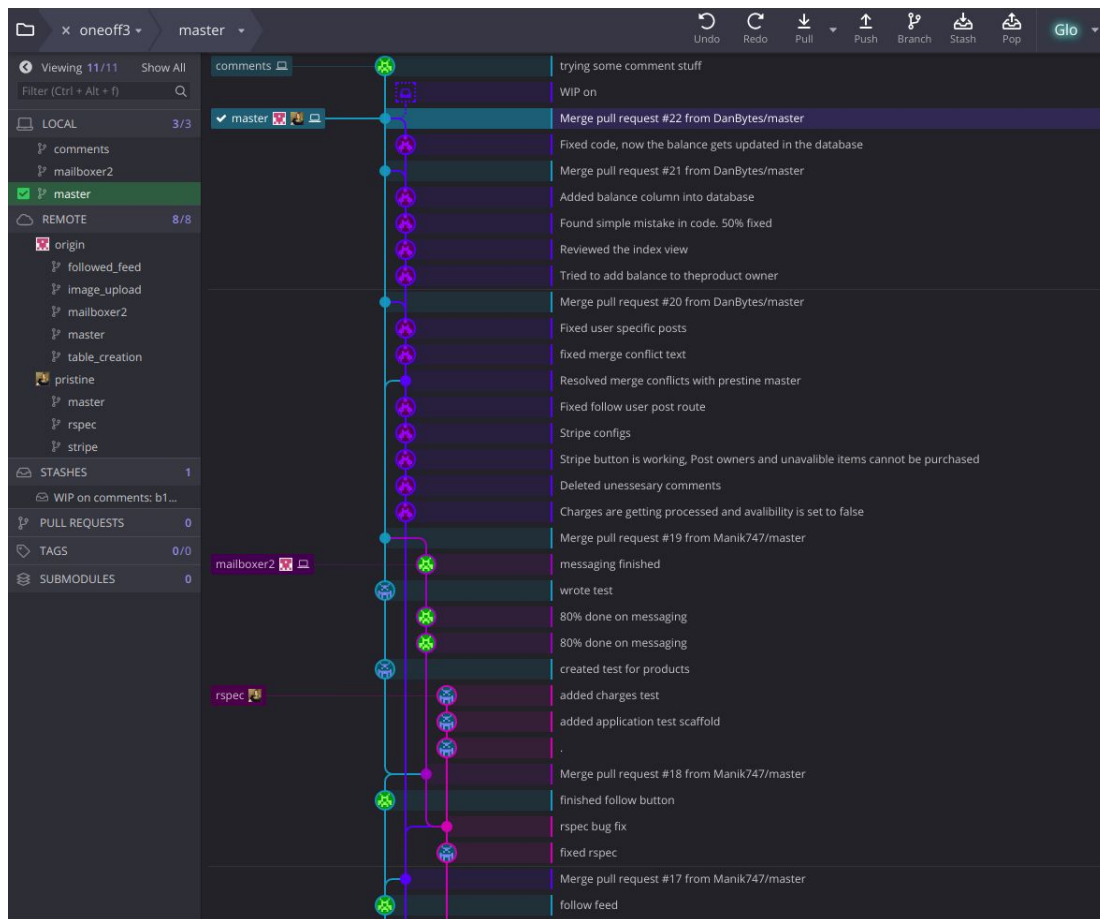
Database Entity Relationship Diagrams (What we ended up with)



Details of planning process.

Our initial plan for our app was arranged by creating cards of features we want to arrange along with other tasks that we want to arrange. Code pairing was done by all three group members to create the skeleton of the app. Once this had been arranged we created a git repo and used forking workflow to work on separate parts of the app using pull requests to merge to review and merge changes back into the pristine master branch. We completed all initial features and added more like messaging





Short Answer questions

What is the need (i.e. challenge) that you will be addressing in your project?

One off is designed to be a beautiful two sided marketplace where artist's, collectors, craftsmen and antiques dealers can sell beautiful one-off items. Users are able to follow other users so that their favourite items and artists are always just a click away. Beyond just being a simple marketplace One off is meant to be a source of inspiration and entertainment. Borrowing key ideas from the likes of Pinterest and Instagram where users can simply browse content for their leisure.

Identify the problem you're trying to solve by building this particular marketplace App? Why is it a problem that needs solving?

The world is becoming more and more cluttered with generic mass produced products from the likes of IKEA, KMART and Amazon. ONEOFF aims to bring back sentiment, beauty around unique and interesting one-off products that have character and history.

Describe the network infrastructure the App may be based on.

The app is firmly based in Heroku where database management hosting and redirecting is managed by this service.

Identify and describe the software to be used in your App.

- Visual studio code – for coding
- ruby on rails – as the web framework
- postgres – as the database
- Heroku – as the deployment platform
- Rspec – as the method of testing the application
- rails command line – as a terminal to carry out some commands
- GitHub – As our way version control and back up
- Trello – As the method of assigning tasks and keeping up to date
- Slack – As a method of communication
- Google hangouts - as a way to share screen with each other for code pairing
- Lucid charts- Creation of ERD which could be worked on together

Identify the database to be used in your App and provide a justification for your choice.

For this app we will be using postgres as opposed to SQL or MySQL. We have made this decision because postgres is widely used and flexible. Postgres is also an object relational database, meaning that databases can have and there contents can be given more definition by the developer.

Identify and describe the production database setup (i.e. postgres instance).

Describe the architecture of your App.

```
~/projects/rails/oneoff3$ rails new project --database=postgresql
```

Inside rails project > config > database.yml

Define default database parameters. Test, Development and Production environments inherit the defaults.

```

default: &default
  adapter: postgresql
  encoding: unicode
  # For details on connection pooling, see Rails configuration guide
  # http://guides.rubyonrails.org/configuring.html#database-pooling
  pool: <%= ENV.fetch("RAILS_MAX_THREADS") { 5 } %>
  password: <%= ENV['DB_PASSWORD'] %>
  username: <%= ENV['DB_USERNAME'] %>
  host: localhost
  port: 5432

```

Our values are stored inside .env file with the same variable names. That insures that our different setups won't conflict with each other.

Explain the different high-level components (abstractions) in your App.

The components of this app are:

- Messaging - Useful for arranging pick up or shipping
- Products - displayed in a global or selective followed feed
- Global feed
- Following feed
- Crud products
- Purchase products

Describe (in general terms) the data structure of marketplace apps that are similar to your own (e.g. eBay, Airbnb).

- Cloudinary – for payment storage
- Devise – for user authentication and initial creation of the user model
- Stripe – for secure, more stable payment

Discuss the database relations to be implemented.

The primary table that our database was based around was a user. As we wanted to create the opportunity for users to follow other users we created a join table to help create the many to many relationship needed for this. Stored in this was the unique ID of each user as a follower and following. The other main table that was added was products which a user could have many of. Below is our initial ERD. As Gems and functionality was added this was changed and modified.

Discuss how Agile methodology is being implemented in your project.

Agile methods were implemented in a number of ways. Firstly our group took part in a daily stand up with interested parties (our instructors and classmates). This allowed us to get ideas and solutions from many sources and let people we know where we are at. We also did our own very quick (sitting down) stand ups where we discussed what we were stuck on, broke problems

down into simple trello cards and split up tasks. When we were struggling with an issue or a bottleneck was created by a feature we did code pairing using google hangouts.

Provide an overview and description of your Source control process.

Github was used for version control and backing up. We used a forking workflow where a member was selected to have the pristine master and be in charge of deploying to heroku. Each team member had a forked copy of the pristine master. On each of our forks we would make feature branches and merge that to our own master once satisfied that it was working. From here we could create a pull request to the pristine master allowing the code to be reviewed before merging the changes in. When changes have been made to the pristine master we pulled them down to our local machine and then pushed them to our own repos.

Provide an overview and description of your Testing process.

Testing using rspec is more reliable because tests can be transported, the test we made would make sure that users were filling in the required fields and the usernames and emails weren't already in the database.

```
dobbo@DESKTOP-FMJ2067:/mnt/c/Users/Charlotte/projects/rails/oneoff3/oneoff/oneoff3$ rspec spec/models/product_spec.rb
*
Finished in 0.15035 seconds (files took 3.47 seconds to load)
1 example, 0 failures
```

Discuss and analyse requirements related to information system security.

We used a basic git ignore .env to make sure passwords were not uploaded to github and also used devise, it was a better option to go with a third party software rather than do our own because devise specialises in security and is constantly updating and keeping ahead, in line with the current market. It's better to let security specialists deal with security issues

To protect this information we are using devise encryption and ignoring all passwords to the database encryption, although we could have made our own encryption using ruby, we felt it was better to go with a third party.

Queensland legal obligations regarding the collection of user data

PRIVACY

Any customer information that you collect must comply with privacy laws, whether you use this information or not. The laws also cover how you can store and use the information.

- The business cannot use unfair means to collect the information. So it cannot trick the person into giving the information or spy on them. See APP 3.5.

- The business can use the information for building relationships with the clients but if it wants to use the information for some other purpose, it can only do so if the clients would reasonably expect that to happen (see APP 6.2) or if the clients have consented (see APP 6.1(a)).
- If the business is collecting sensitive information (racial origin, political opinions, religion, philosophical beliefs, sexual preferences, criminal record, or health information) it will need the consent of the individual. See APP 3. Sometimes it may not be obvious whether the individual has consented to all the uses of personal information that the business has in mind. The Privacy Act only states that consent can be either 'express' or 'implied'. Please find tips for compliance and other commentary in the APP Guidelines.

If a small business is covered by the Privacy Act, the potential purchasers will have to take care to protect individuals privacy rights in the due diligence process. Inspecting purchasers will need to comply with the Australian Privacy Principles if they collect personal information. Taking notes which include personal information or taking a copy of a document, which has personal information in it, is collecting personal information.

Only personal information necessary to assessment of business should be reviewed. De-identified information should be relied on wherever possible. Generally, purchasers would be able to review:

- financial information
- contractual documents with trading partners, suppliers and contractors
- information about key employees relevant to their employment relationship
- aggregated information about employee entitlements (long service leave etc)
- aggregated statistical customer information

Prospective purchases must take reasonable steps to protect the privacy of any personal information and comply with privacy clauses included in confidentiality agreements. Generally purchasers should only inspect and not copy documents. Personal information collected by the

Dear diary

29/10/18

- First day of the assignment
- Concentrating on planning, documenting, delegating tasks

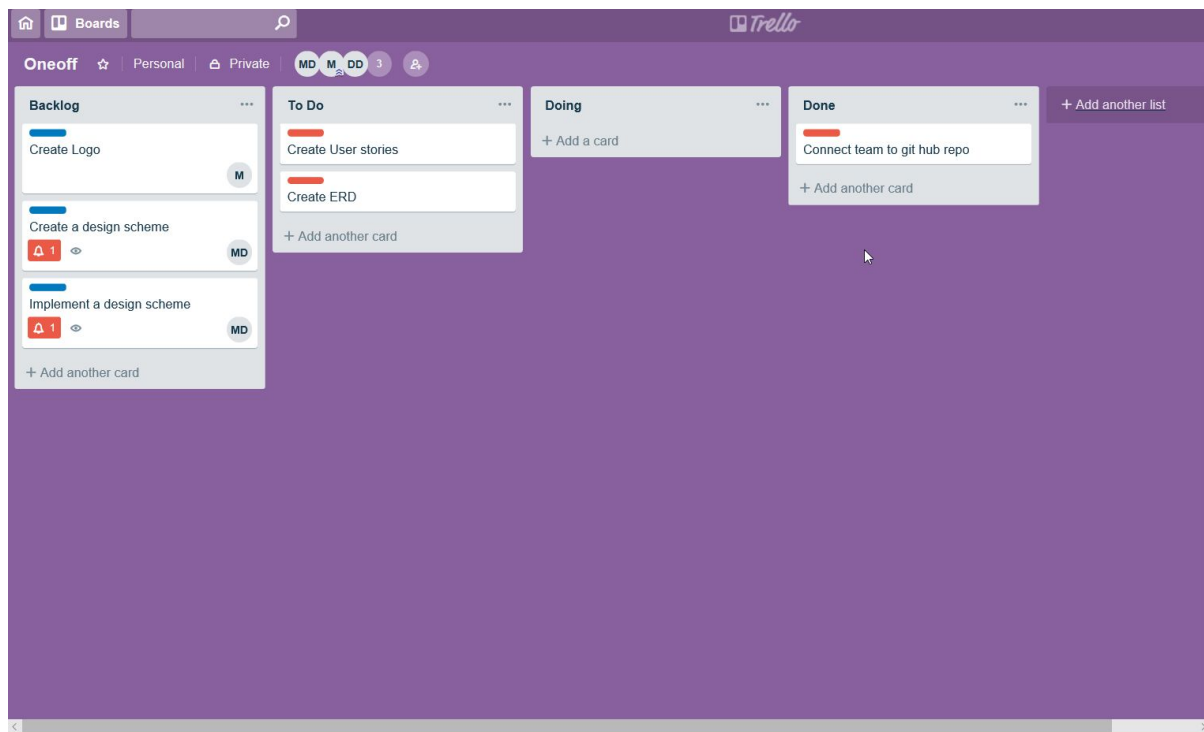
What we got done today.

We established communication method through trello and came up with a schema for importance of tasks

We got everyone connected to trello

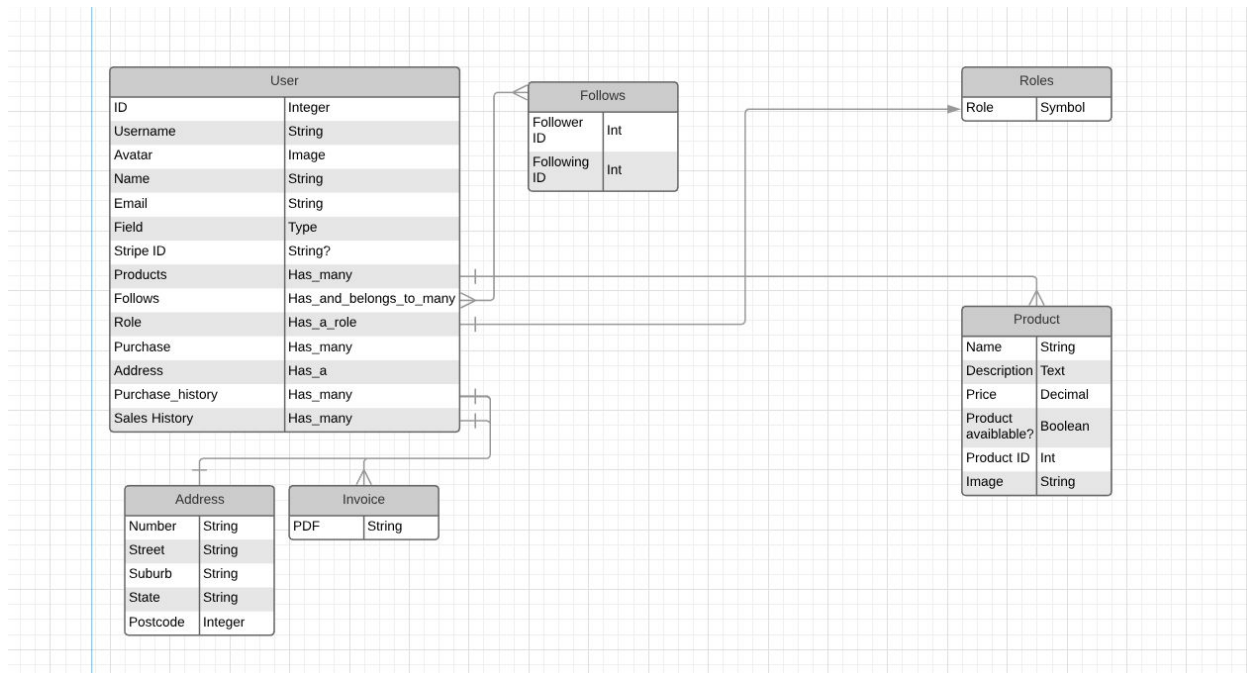
We got our databases connected

We got our user stories done

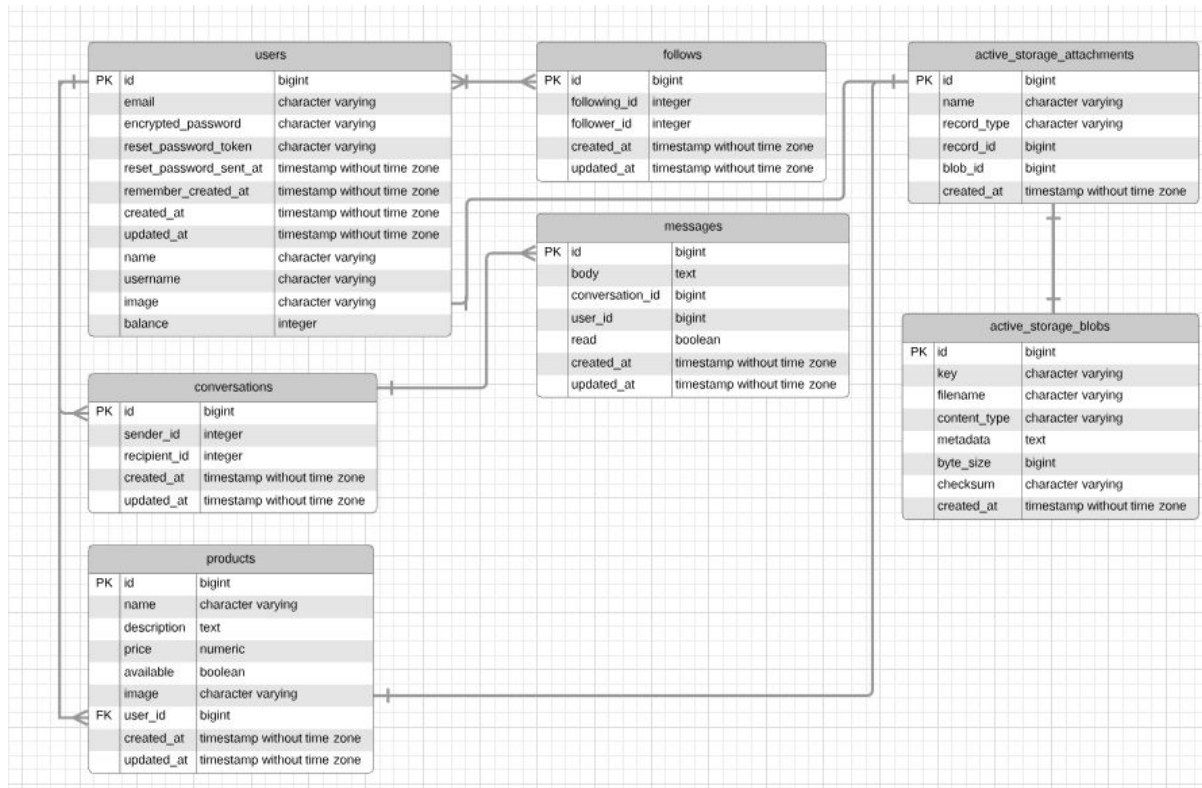


We got everything connected on GitHub

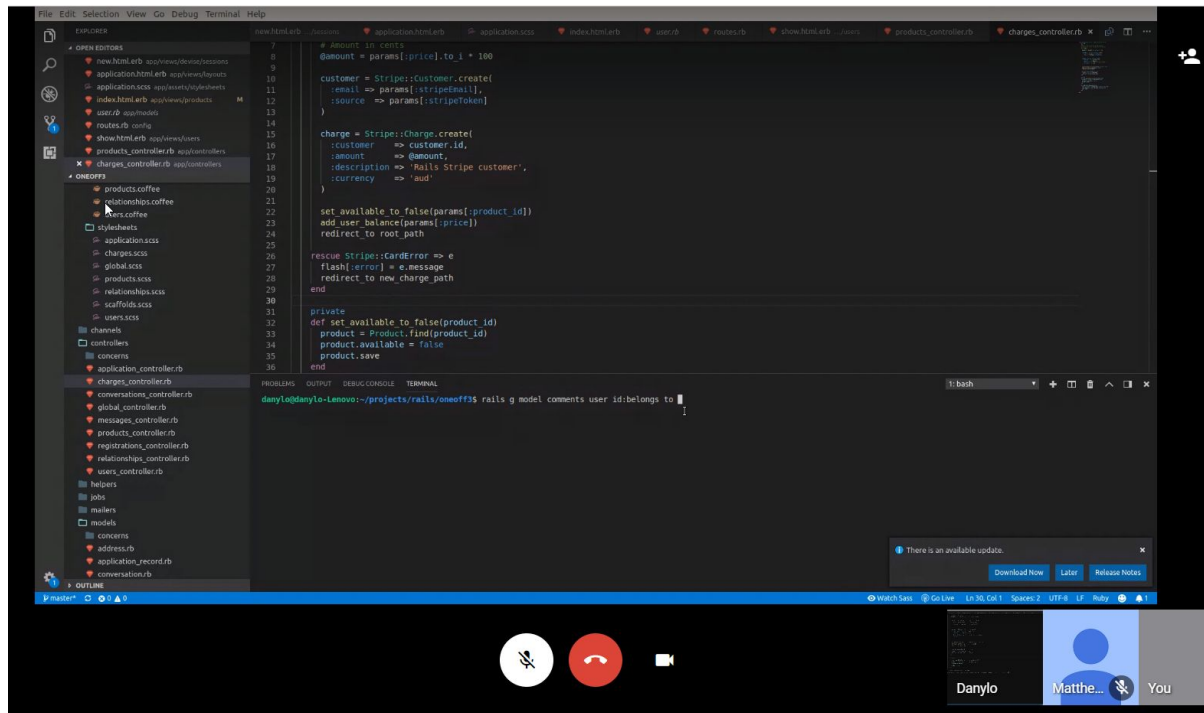
ERD



Final ERD



We also communicated through google hangout screensharing so we could watch each other code



30/11/18

We needed to start again as we somehow managed to mess up our database relationship. We setup google docs to help do the wireframes

1/11/18

We got our profiles done and did some relational work on the database we had some errors and bugs that took up a fair bit of time, we struggled to test rails and decided to just do rpsec

5/11/18

We did some bootstrap styling and redeployed to Heroku

Bootstrap styling

One-Off

You need to sign in or sign up before continuing.

Log in

Email

Password

☐ Remember me

[Forgot Password?](#)



Heroku deployment

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
-- add_column(:users, :balance, :integer, {:null=>false, :default=>0})
D, [2018-11-08T00:33:16.332474 #4] DEBUG -- : (17.6ms) ALTER TABLE "users" ADD "balance" integer DEFAULT 0 NOT NULL
-> 0.0211s
== 20181106131313 AddBalanceToUsers: migrated (0.0213s) =====

D, [2018-11-08T00:33:16.340804 #4] DEBUG -- : ActiveRecord::SchemaMigration Create (5.6ms) INSERT INTO "schema_migrations" ("version") VALUES ($1) RETURNING "version" [
["version", "20181106131313"]]
D, [2018-11-08T00:33:16.348603 #4] DEBUG -- : (7.4ms) COMMIT
D, [2018-11-08T00:33:16.360724 #4] DEBUG -- : ActiveRecord::InternalMetadata Load (1.0ms) SELECT "ar_internal_metadata".* FROM "ar_internal_metadata" WHERE "ar_internal
_metadata"."key" = $1 LIMIT $2 [{"key", "environment"}, [{"LIMIT", 1}]
D, [2018-11-08T00:33:16.373005 #4] DEBUG -- : (0.8ms) BEGIN
D, [2018-11-08T00:33:16.375907 #4] DEBUG -- : ActiveRecord::InternalMetadata Create (1.6ms) INSERT INTO "ar_internal_metadata" ("key", "value", "created_at", "updated_at
") VALUES ($1, $2, $3, $4) RETURNING "key" [{"key", "environment"}, [{"value", "production"}, [{"created_at", "2018-11-08 00:33:16.373447"}, [{"updated_at", "2018-11-08 00:33
":16.373447"}]]
D, [2018-11-08T00:33:16.378216 #4] DEBUG -- : (2.0ms) COMMIT
D, [2018-11-08T00:33:16.381567 #4] DEBUG -- : (3.1ms) SELECT pg_advisory_unlock(1480539515911191120)
dobbo@DESKTOP-FM02067:/mnt/c/Users/Charlotte/projects/rails/oneoff3/oneoff/oneoff3$ heroku bundle install
> Warning: bundle is not a heroku command.
Did you mean console? [y/n]: y
> Error: Unexpected argument: install
> See more help with --help
dobbo@DESKTOP-FM02067:/mnt/c/Users/Charlotte/projects/rails/oneoff3/oneoff/oneoff3$ heroku open
> Error opening web browser.
> Error: Exited with code 3
```

6/11/18

We were able to set up a messaging system for the app. We also wrote an Rspec test

7/11/18

We got comments functioning and finished the rest of the documentation and started working on the presentation

8/11/18

We decided to do more work on the documentation using google docs

1

File Edit View Insert Format Tools Add-ons Help All changes saved in Drive

100% Normal text Arial 11 B I U A

Editing

postgres – as the database

- Heroku – as the deployment platform
- Rspec – as the method of testing the application
- rails command line – as a terminal to carry out some commands
- GitHub – As our way version control and back up
- Trello – As the method of assigning tasks and keeping up to date
- Slack – As a method of communication
- Google hangouts - as a way to share screen with each other for code pairing
- Lucid charts- Creation of ERD which could be worked on together

Heroku Devise documentation is further down

4. Identify the database to be used in your app and provide a justification for your choice

For this app we will be using postgres as opposed to sql or MySQL. We have made this decision because postgres is widely used and flexible. Postgres is an object relational database, meaning that databases can have and there are contents

5. Identify and describe the production database setup (i.e. postgres instance) Inside rails project > config

6. Describe the architecture of your app

The app lands on a login page in which you have the option to sign up, one logged in or signed up, you can create products to sell and buy products from other sellers

7. Explain the different high-level components (abstractions) in your app

The components of this app are: