

out_Veggin

Images I used and saying thanks to: https://www.printablesigns.net/preview/Recyclables_No_Food_Waste

Criteria

- R1 Create your app using Ruby on Rails. - Done
- R2 Use Postgresql database in development and production. - Done
- R3 Your app will have authentication (eg. Devise). - Done
- R4 Your app will have authorisation (i.e. users have restrictions on what they can see and edit). - Done
- R5 Your app will have some type of file (eg. images) uploading capability. - Done
- R6 Your app is to be deployed to Heroku (recommended) or AWS. - Done

R7 Identification of the problem you are trying to solve by building this particular marketplace app.

I've built this market-app because I have a lot of neighbours around me that really love working in their gardens, some are unfit for work, but produce some really nice fruit and vegetables. They would highly benefit from creating an account with out_Veggin and then others can find the items they are after and in the location, and instead of buying from the big companies and making them richer and richer, every one can make some money and have a small business for some decent income. It will bring the community together again like it used to be when every one would have small businesses and help each other.

R8 Why is it a problem that needs solving?

It needed solving because there are many people unable to work and this gives them an opportunity to work and make some side income. I also notice a lot of fruit and vegetables go to waste because they simple cant consume it all them selves, so it is great for many different reasons.

R9 A link (URL) to your deployed app (i.e. website)

https://outveggin.herokuapp.com/users/sign_in

R10 A link to your GitHub repository (repo).

https://github.com/AaronNovakDev/out_Veggin

R11 Description of your marketplace app (website), including:

- Purpose

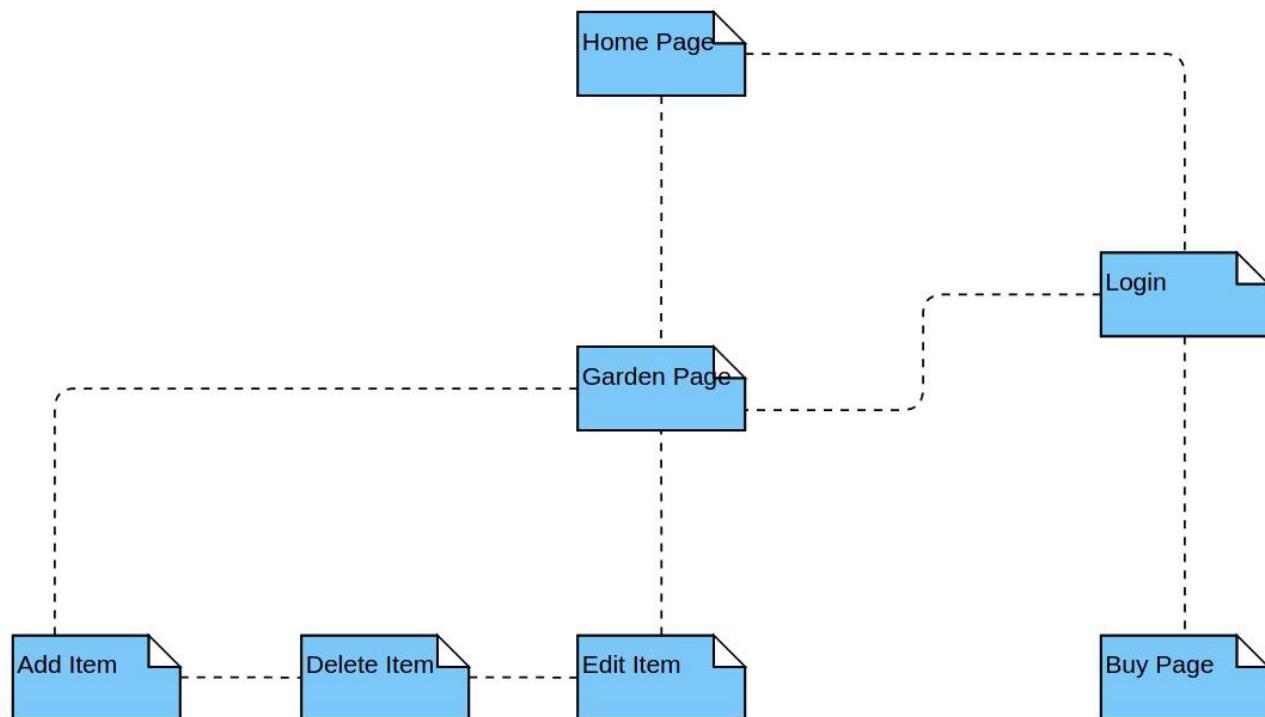
The purpose of my application is to fill in the void in the marketplace, where you can only buy from either the large chains, or directly from farmers. I would like to make it available for every one to be able to sign up and sell there fruits and vegetables. I understand there will be more insurances etc involved, but that will be set up later down the track.

- Functionality / features

It features the full CRUD capabilitites. A user can log in to their User account safely with the use of the devise gem, they can then add the items they would like to sell, update or destroy. A buyer can login and enter what they would like to buy, and with the simple search button at the top of the screen it will filter out who and where the item they have entered is. There will be a messaging system so that people dont have to give out their phone numbers just in case they want to drop off the item or they cant pick it when they said they would

- Sitemap

Main page / home page Create user Login Sellers page Add items Edit items Destroy items Buyers page Buy item page Payment page



- Screenshots

Here are my Screenshots of the working app, it works perfectly locally, but it for some reason when pushed to heroku had some old errors thats i fixed up pushed to it, so here are them. It is also fine to clone off github, I used all the standard gems. Authorisation access fail

out_Veggin

[Home](#) [Log out](#)

[Garden](#)

You are not authorized to access this page.

Welcome to out_Veggin!

Whether you're here to make
a "celery", or

"Leek" your lips over these
delish Veggies,

"Lettuce" show you a great
time!

Check out what's trending right now!

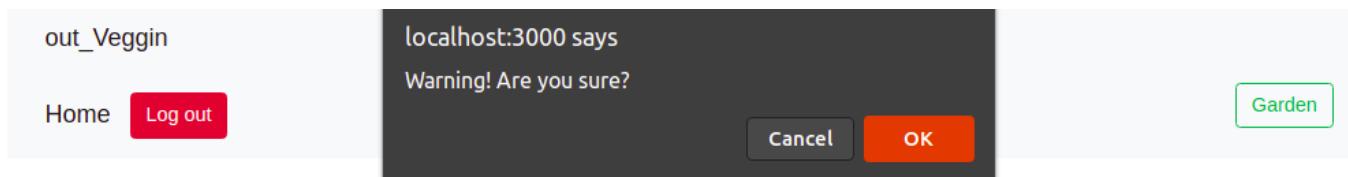


[Buy Now!](#)



[Buy Now!](#)

[delete item](#)



Items

Type	Description	Price	Quantity	User			
lemons	yellow round things with bite	5.0	5	2	Show	Edit	Destroy
carrot	long orange thing - yummo	5.0	10	1	Show	Edit	Destroy

New Item

delete item success

out_Veggin

Home Log out

Garden

Item was successfully deleted.

Items

Type	Description	Price	Quantity	User
lemons	yellow round things with bite	5.0	5	2

New Item

edit item

out_Veggin

Home Log out

Garden

Edit Your Item!

Category

carrot ✓

Description

long orange thing ✓

Price

5.0 ✓

Quantity

10 ✓

Choose file No file chosen

Update Item

Show | Back

edit item complete

out_Veggin

Home [Log out](#)

[Garden](#)

Item was successfully updated.

Type: carrot

Description: long orange thing - yummo

Price: 5.0

Quantity: 10

User: test@test.com



garden page

out_Veggin

Home [Log out](#)

[Garden](#)

Items

Type	Description	Price	Quantity	User	
lemons	yellow round things with bite	5.0	5	2	Show
carrot	long orange thing	5.0	10	1	Show

[New Item](#)

home page

out_Veggin

Home Log In

Garden

Welcome to out_Veggin!

Whether you're here to make
a "celery", or

"Leek" your lips over these
delish Veggies,

"Lettuce" show you a great
time!

Check out
what's trending
right now!



Buy Now!



Buy Now!

add item successful

out_Veggin

[Home](#) [Log out](#)[Garden](#)

Item was successfully created.

Type: carrot

Description: Long orange thingys

Price: 4.0

Quantity: 10

Email: test@test.com



new item screen

out_Veggin

[Home](#) [Log out](#)[Garden](#)

Add Your New Item Here

Category

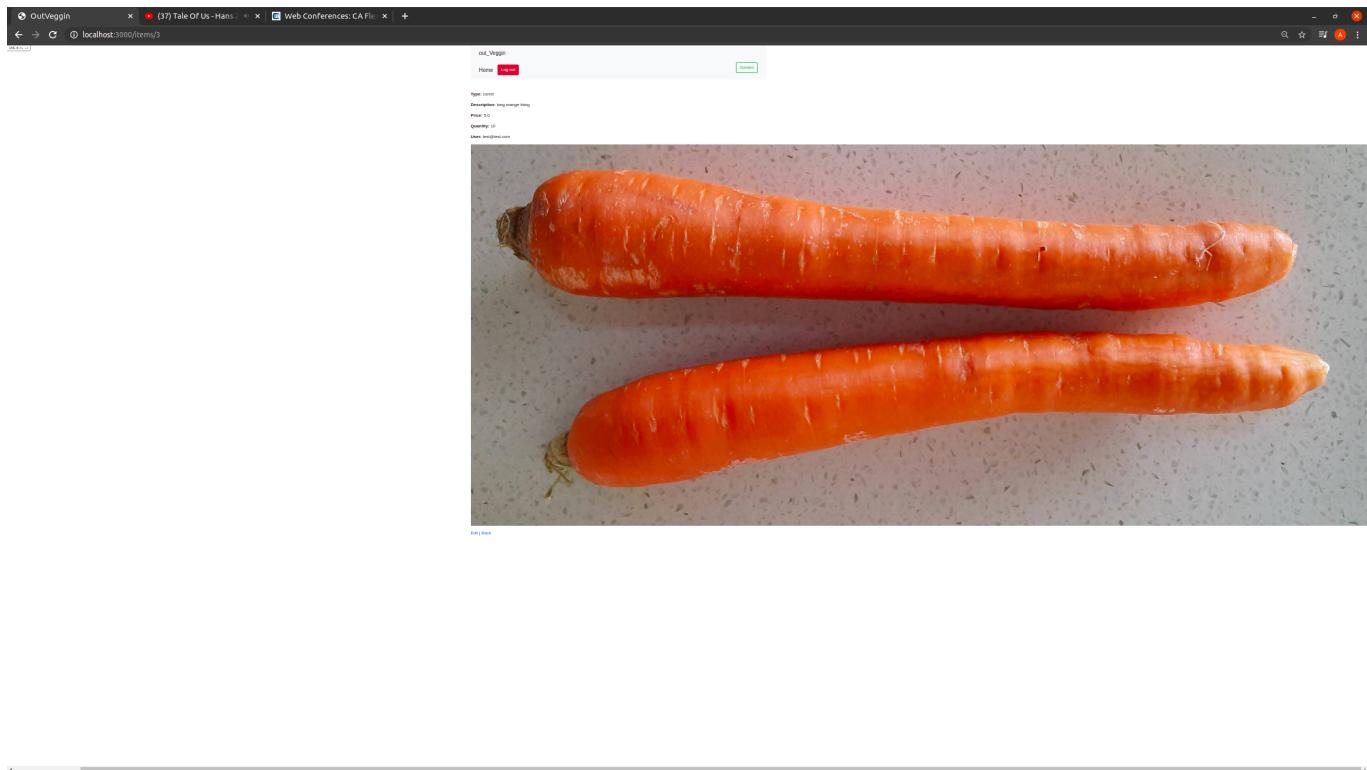
Description

Price

Quantity

[Choose file](#) No file chosen[Create Item](#)[Back](#)

show items



signup screen

out_Veggin

Home [Log In](#) [Garden](#)

Sign up

Email *

Password *

 6 characters minimum

Password confirmation *

[Sign up!](#)
[Log in](#)

- Target audience

My target audience is every one who would like to make some extra cash and enjoy growing fruit and vegetables. I am mainly targetting the stay at home people that either have no work, are out of work, or unable to work. This way they can still work and stay fit and healthy while staying home.

- Tech stack (e.g. html, css, deployment platform, etc)

I used Heroku to deploy my application. I used bootstrap and scss to style it. I used Ruby to write the whole code and a bit of Javascript.

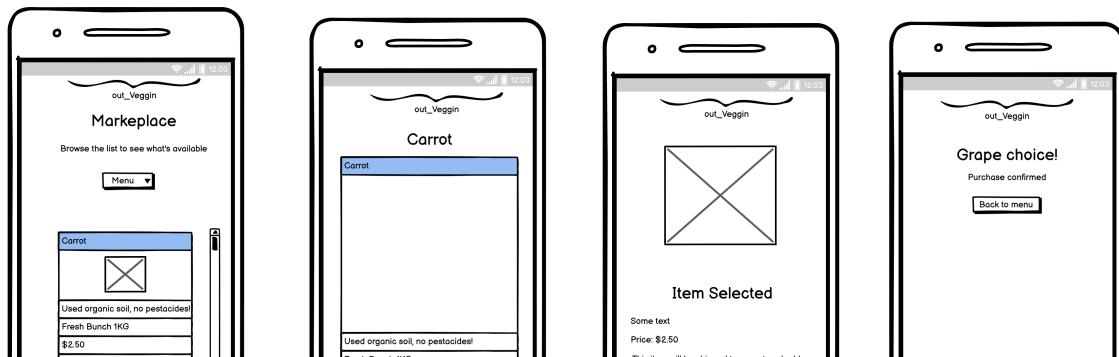
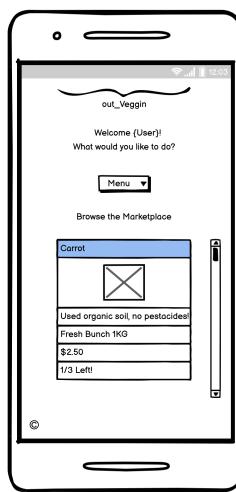
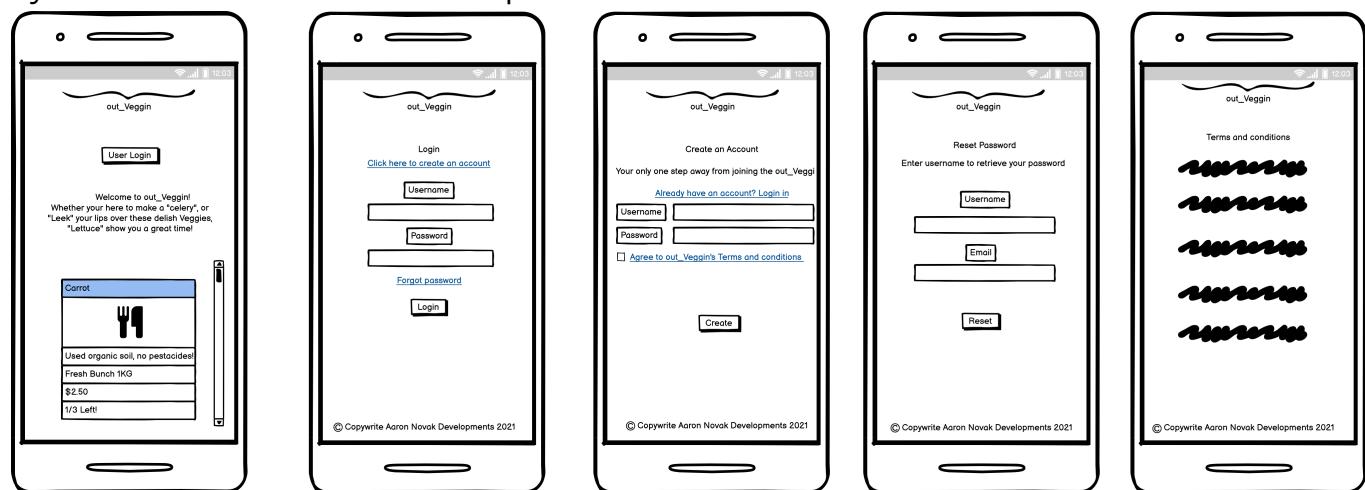
R12 User stories for your app

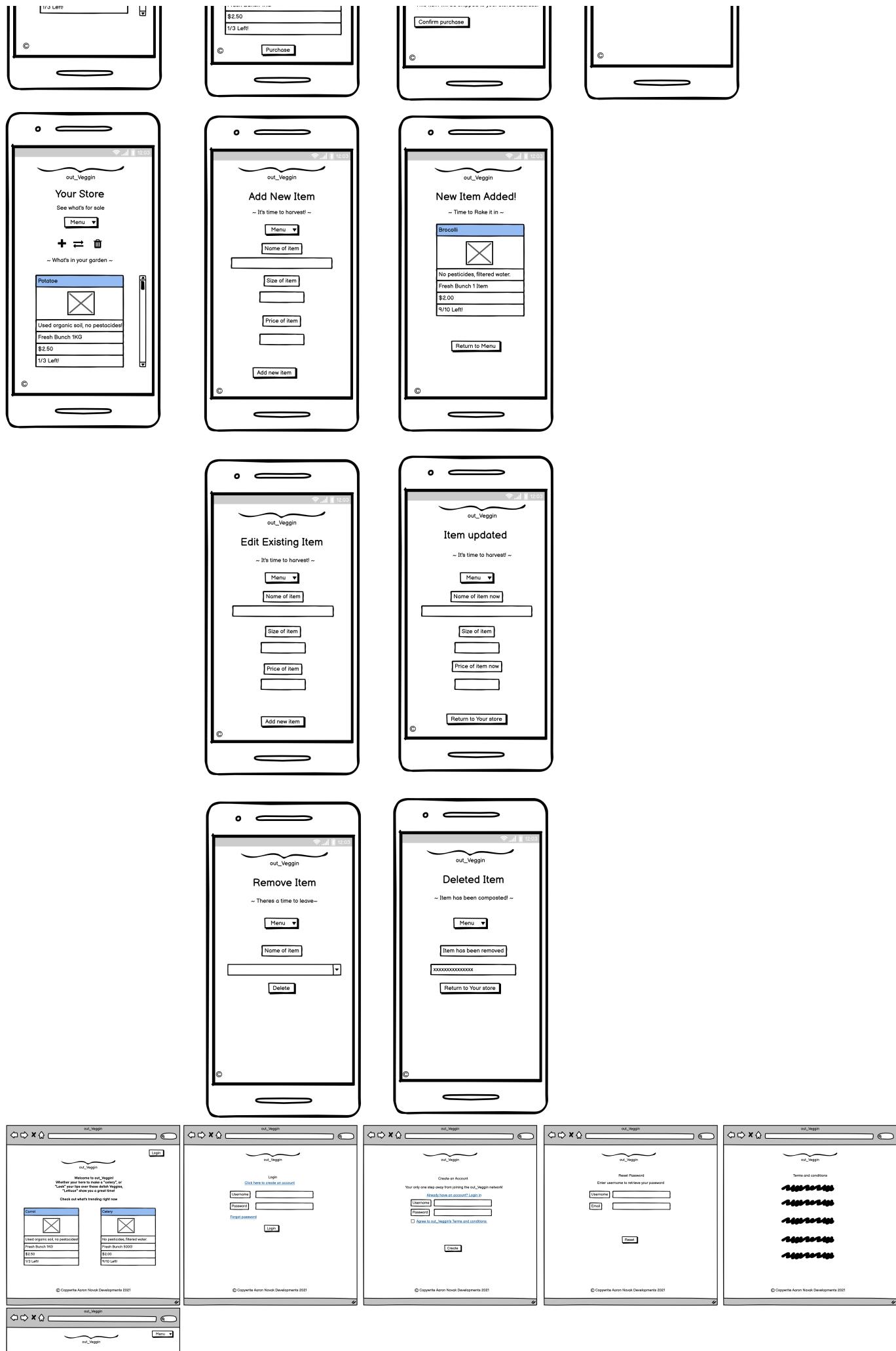
As a user, I can create an account on out_Veggin, so that I can browse and buy/sell produce As a user, I can login to the platform, so that I can access my private account and transact As a user, if I forget my password, I can reset it, so that I can continue to access my account As a user, I can log out of the platform, so that my account stays safe

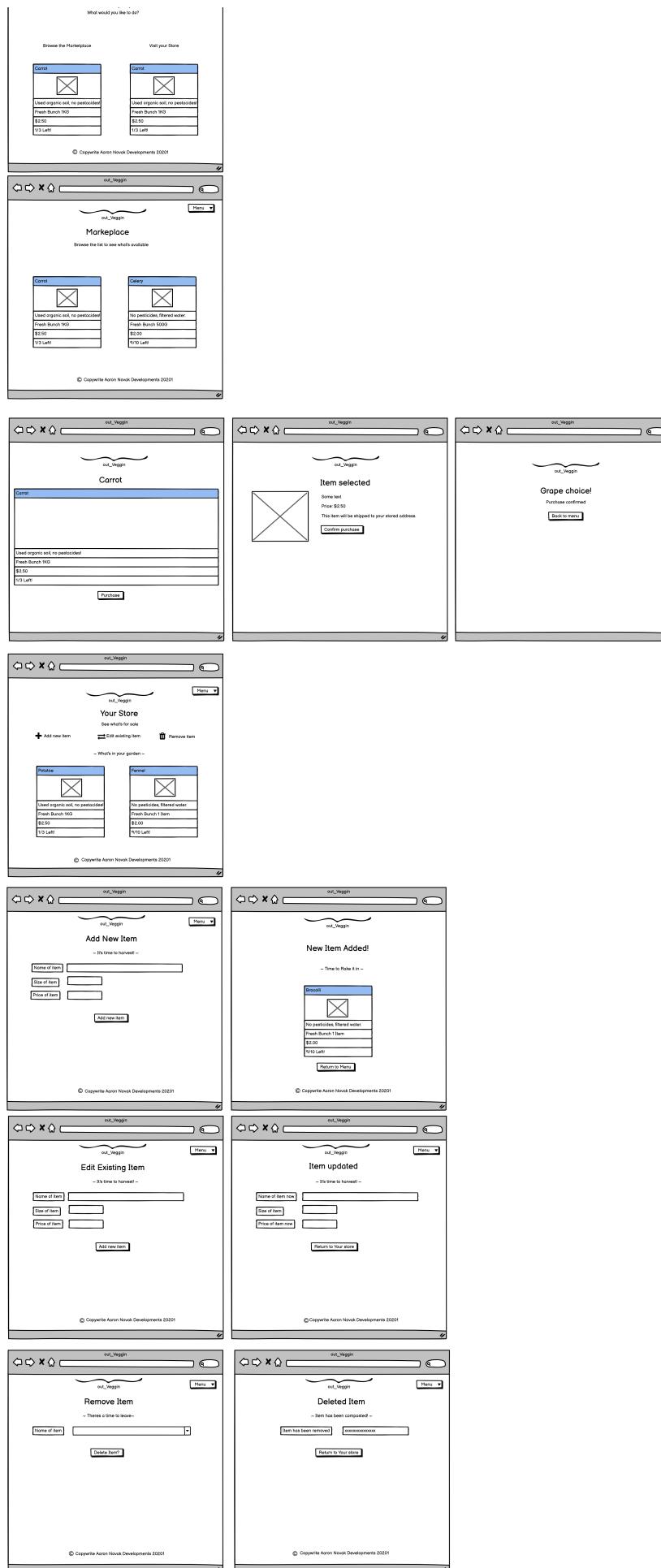
As a buyer, I can browse what type of produce is available and see basic information about each item, so that I can decide whether or not to buy As a buyer I can purchase items from the store using my credit card As a seller, I can list new items (with an image) for sale so that buyers can see them and decide to buy As a seller, I can see a full list of what I have for sale, so that I can monitor stock and update it as needed As a seller, I can edit the details of my items so that they are correct and don't mislead buyers As a seller, I can delete produce so that items no longer available are not displayed

R13 Wireframes for your app

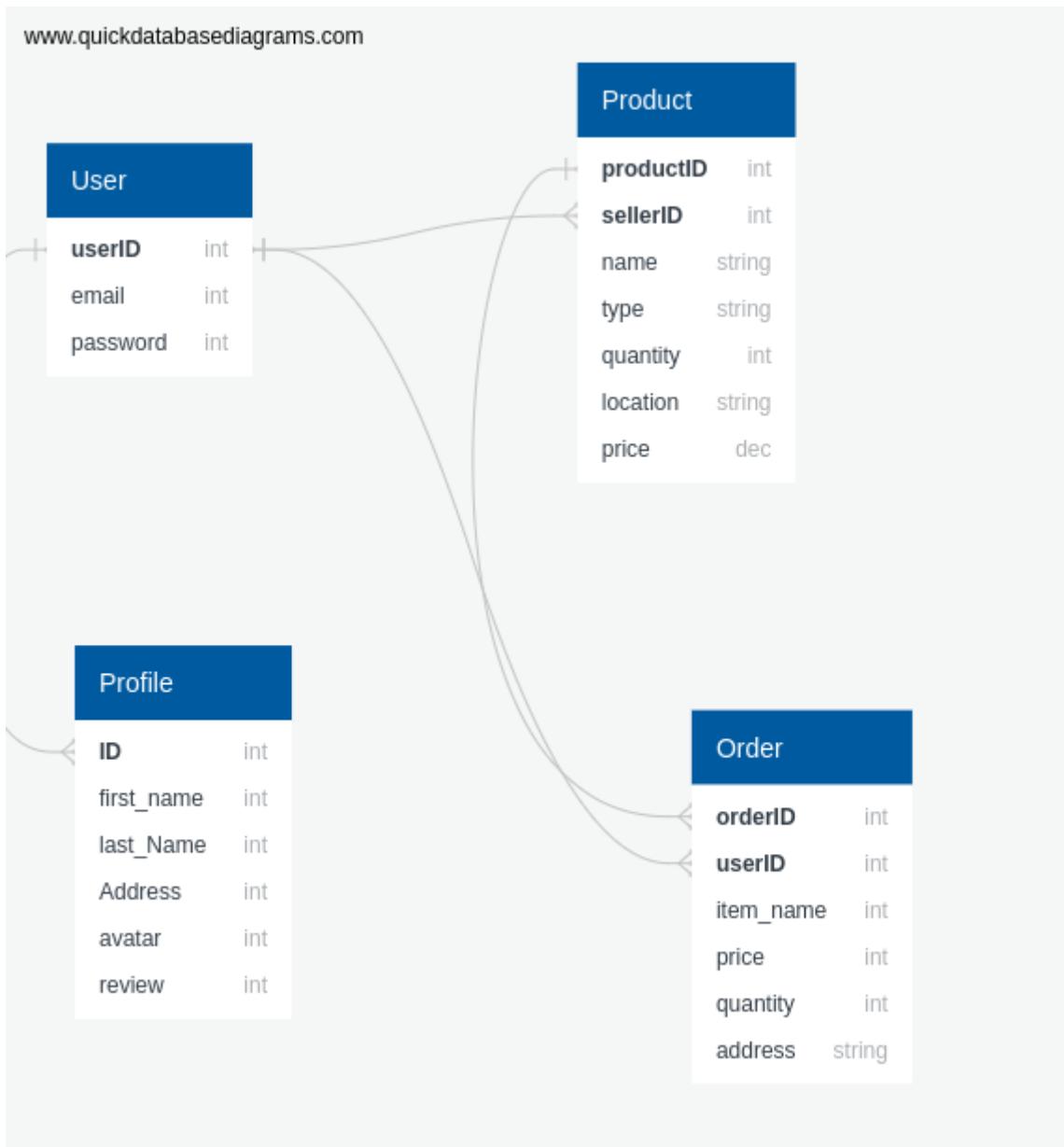
My wireframes for mobile first and computer.







R14 An ERD for your app



My ERD

R15 Explain the different high-level components (abstractions) in your app

My high level components im using are Postgresql – its a great easy to use DB software that's free, When you couple this with rails, it creates a very powerful database application maker. I first of all created a postgresql account and linked it to my rails application by putting my credentials into the database.yml file. Once I completed that I then created the DB integration with “rails db:create” once that was done I ran “rails db:migrate”, by doing this is creates a connection in the Schema.rb file so that Rails knows the DB and Rails are linked. You can then create what ever tables you want with what ever colums you want with simple syntax such as “create_table(:users_roles, :id => false) do |t| t.references :user t.references :role end” This then once Migrated will create the tables and columns you want in the DB. Ruby has an inbuilt set of functions that you can call on that make database making a breeze. They call it active record associations. What is does is creates the data base entity links behind the scenes so you dont have to do all the nitty gritty. They use terms like belongs_to, has_one, has_many etc, which link 2 tables together so they work in unison

R16 Detail any third party services that your app will use

The application is uploaded to Heroku and fully operational (was) at the time of writing this with full Auth and CRUD. I will be hosting my images on Cloudinary, it's free to use and you have an extremely reasonable amount of space for a free account.

R17 Describe your projects models in terms of the relationships (active record associations) they have ## with each other

My models are:- -Item -Order -Profile -Role -User

I have a User that has many Items, but Items have only one User. A User can have many Orders, but a single Order can only have one User. A Profile belongs to the User and has many Items. A User can have one Role, but there are many Roles that a User can be. (Later down the track, super Edit, Super Delete, Super user access to all CRUD.)

R18 Discuss the database relations to be implemented in your application

My data base implementations that are linked to my PostgresQL are Items, Orders, Profile, Role and User. These all link with the Postgresql when you run a DB:create and then a DB:migrate. I have created a profile model which links to their items, they can also link to their order if they were to buy something. Because I've used devise I only needed to create one user which is able to buy or sell. I firstly wanted to create 2 types of User Roles, I wanted to keep them separate and then I thought harder and didn't think it was worth it, I can create one User model and they can buy or sell depending on what they click on. I would of liked to have another model that was connected to a review type system and that will be implemented in future.

R19 Provide your database schema design

I've added in comments in my Schema file as per CoderAcademy instructions. It shows what each line of code does on each table. The reason we have t.index is just to make the code much more quicker to write and draw up a full table with only a simple few words of syntax thanks to our ruby gems. The schema file is created once we run the migrate to migrate all the tables and let them all talk together in the way they are entered, where it be one to one, one to many, many to many, etc.

My schema was updated a few times as I learnt different ways on implementing what I wanted to do and how I wanted the DB to run.

<https://app.quickdatabasediagrams.com/#/d/Dh9l84>



www.quickdatabasediagrams.com

Free Diagram Documentation

Link to schema: <https://app.quickdatabasediagrams.com/#/d/Dh9l84>

Profile

Field	Description	Type	Default	Other
ID		int		FK
first_name				
last_name				
Address				
avatar				
review				

User

Field	Description	Type	Default	Other
userID				FK
email				
password				

Product

Field	Description	Type	Default	Other
productID				FK
sellerID				FK
name		string		
type		string		
quantity		int		
location		string		
price		dec		

Order

Field	Description	Type	Default	Other
orderID				FK
userID				FK
item_name				
price				
quantity				
address		string		

R20 Describe the way tasks are allocated and tracked in your project

<https://trello.com/b/vL7XZoO8/project-management> I've used the Trello board to track my tasks with. Its really easy to use and a great all round task tracking app. It has helped me by reminding me what to do and how much time i should be spending on each part. It also helps me by having an all in one place to go to, to see what tasks i have left.

HOW TO RUN THIS PROGRAM

To run this application all you need to do is open the website that ive attached above. Once youve clicked that, it will take you straight into the login screen. Since your a new User you click on "Sign Up" You create your user name, use your email address, then enter a password thats at least 6 characters long. Then click on Sign Up!. Now youve made it to the Home screen, from here its as simple as adding in your items via the Add Item button, Or you can click on Garden. Garden will take you to every ones listed Items for sale. From here you can add, update or destroy your own items, but youll notice the other users you can only view.