

Gardener's Exchange

...

By: Chris Arpin, Brandon George, Adam Hursey, Noah Lindsey, Tim
Ours, and Mitch Whedon

Business Proposal

Motivation

- Most people buy from local stores
- Connected, local communities through gardening
- Improved health, less waste
- More interest in gardening

Non-Customer Stakeholders

Sellers

- Post listings
- Accept/decline Trade offers

Buyers

- Buy from listings
- Offer trades to sellers

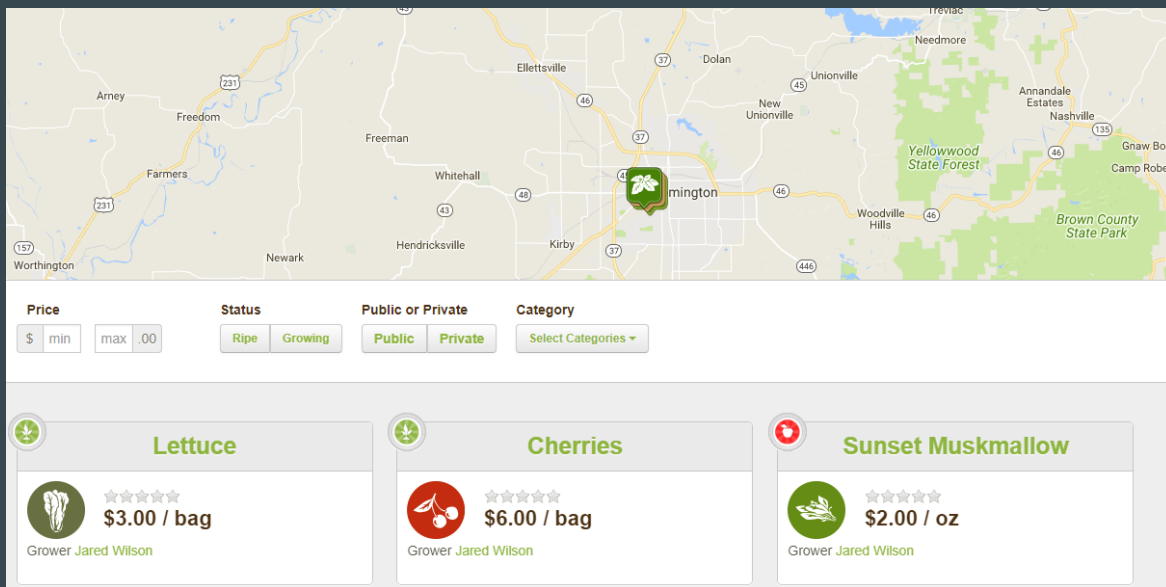
KT Situation Appraisal

List Concerns	Clarify Concerns	Severity	Urgency	Growth
Mindset	<ul style="list-style-type: none">- Their mindset of income might hurt the start- The location of Upland would be harder to start an app than a larger setting	High - Still early, but with could affect the way that the app brings in revenue	Low - we haven't started	Low - their minds are set on what they want
Encouraging for other to grow crops	<ul style="list-style-type: none">- This is not currently the focus of the application- User interaction and user experience would be what would guide this encouragement	Medium - If the app doesn't launch well, it won't grow	Low - again, we haven't started, but also we don't have much influence over potential crop growers	Low - this is early and something that would be added on later
Loss of buyer interaction	<ul style="list-style-type: none">- Customers are what the application needs to grow and produce profit	High - Once there is a connection between the buyer and the seller they won't require the app	High - This is an obstacle for the app growing initially	Low - This is a static issues

Solution Alternatives

RipeNear.Me

- Web application for buying/selling home-grown foods online
- “Growing” and “Ripe” statuses
- “Friend” other users
- Interactive map area



Budget/Cost-Benefit Analysis

- Free to use
- $NPV = \$0$
- No income from a charge per transaction

The Farmer's Garden

- Web application for buying/selling/swapping home-grown foods
- Requests foods, offer free items
- Minimal clicks to get tasks done, easy to use

Budget/Cost-Benefit Analysis

- Free to use
- $NPV = \$0$
- No income charge per transaction

Custom Build

...

Gardener's Exchange

- Mobile application for buying/selling/trading home-grown foods
- Home Feed of listings in a default search radius
- Card listing style
- Easy to use interface

Budget/Cost-Benefit Analysis

- Initial investment: \$540 maintenance, \$25 publishing on Google Play Store
- \$540 yearly maintenance cost
- $NPV > 0$
- Benefits based off of a small charge per transaction

Year	0	1	2	3	4
Discount Rate	4%				
Costs					
Initial Investment	-\$565				
Maintenance		-\$540	-\$540	-\$540	-\$540
Benefits	\$0.00	\$700.00	\$700.00	\$700.00	\$700.00
Future Value	-\$565	\$160.00	\$160.00	\$160.00	\$160.00
Present Value	-\$565	\$153.85	\$147.93	\$142.24	\$136.77
Cash Flow	-\$565	-\$411.15	-\$263.22	-\$120.99	\$15.78
NPV	\$15.78				

KT Decision Analysis

		Gardener's Exchange			RipeNear.Me			The Farmer's Garden		
Musts		Comments		Y/N	Comments		Y/N	Comments		Y/N
User Posted Content				Y	Users post items		Y	Users post items		Y
User Buying and Selling				Y	Buying and selling from user		Y	Buying and selling from user		Y
Wants	Wgt	Comments	Score	Wgt'd Score	Comments	Score	Wgt'd Score	Comments	Score	Wgt'd Score
Offer and Accept Trades	25%		10	2.5	No trading within the app	0	0	Yes, barter transaction	10	2.5
Charge per transaction	15%		10	1.5	No	0	0	No, the app is owned by someone else	0	0
Stream is the home page	15%		10	1.5	Yes Homepage is given after typing in the ZIP code	8	1.2	Yes, home page lists item for sale with default search parameters	10	1.5
Search by Keyword	15%		10	1.5	No, able to filter after ZIP search	5	0.75	Search by transaction type and category	9	1.35
Filter between closeness and proximity	15%		10	1.5	Search by ZIP code	10	1.5	Yes, ZIP and search radius	10	1.5
Transaction History	5%		10	.5	No	0	0	No	0	0
Buyer can look at items without setting up an account	5%		10	.5	Yes	10	.5	No	0	0
Simple layout according to material design standards	5%		10	.5	Yes	10	.5	Simple and easy to use website	9	0.45
Totals	100			10	Very well put together		4.75	Simple web app for finding other foods		7.35

Our Recommendation

We recommend the Gardener's Exchange

- Meets all requirements without sacrificing highly weighted wants
- No development costs
- Involvement through development

Software Requirements Specification

User Roles

- Users
- Unregistered User/Guests
- Customer
- Seller

Collaborating Systems/Software

- PayPal Expresses Services
- Amazon Web Services S3

Schedule

Date	Milestone
January 30th	Start product development
February 15th	Backend complete, app development begin
March 15th	External systems integrated
March 30th	Start testing
April 20th	Polish product
May 5th	Documentation and preparation for app release
May 17th	App released on App Store

Functional System Requirements

System Response Table

Event ID	Source	Trigger	Response	Major Outputs	External Destinations
L1	Guest	Enters the app	Shows a list of items within 25 miles	Home feed	None
L8	User	Pressing the Distance Button	Pop up box where customer can edit the distance that the app looks for items	Changes searching distance to the set amount	None
C2	Customer	Buys items	Adds an order to the database and notifies the seller	Order data, Email to seller, Home Feed	Project database, seller email
C4	User	Views the details of a listing	Generates page with more information about the item	Item page	None
T1	Customer	Offers Trade	Adds a trade to the database, notifies seller	Trade data, email to seller	Project database, seller email
T2	Seller	Accepted Trade	Update trade data to close trade and email offering customer	Trade data, email to offering customer	Project database, offering customer email
S1	Seller	Lists an item for sale	Adds new listing to database	Listing data	Project database
S4	Seller	Clicks button to view trade offers	Sent to trade offers page	Trade offers page	None

Use Cases

ID	L1			
NAME	Enters the App			
Primary Actor	Guest			
Other Actors	None			
Description	Guest enters app without a login			
Actor Action			System Response	
Typical Event Flow	1. Guest opens app on smartphone			
			2. System opens app to the home feed	
Alternate Event Flows	1. Guest opens app on smartphone			
			2. System crashes on opening the app	
Business Focus			System Focus	
Before event	Trigger	Enters the App	Preconditions	App installed on smartphone
After event	Conclusion	Shows listing of items within 25 miles	Postconditions	Display the home feed

ID	L8			
NAME	Search by distance			
Primary Actor	Guest			
Other Actors	None			
Description	User changes the distance parameter			
Actor Action			System Response	
Typical Event Flow	1. User presses the Distance button			
	2. User enters a new distance			
			3. System refreshes Home Feed with new listings	
Business Focus			System Focus	
Before event	Trigger	Pressing the Distance button	Preconditions	Application running
After event	Conclusion	Pop up box where user can edit the distance that the app looks for items	Postconditions	Home feed updated with new distance parameter

ID	C2			
NAME	Buying items			
Primary Actor	Customer			
Other Actors	None			
Description	Customer buys the items			
Actor Action			System Response	
Typical Event Flow	1.	Customer presses the Buy button		
			2. System asks for confirmation	
	3.	Customer confirms buy order		
			4. System create a new order entry in database	
			5. System notifies seller of buy order	
Alternate Event Flows	1.	Customer presses the Buy button		
			2. System asks for confirmation	
	3.	Customer cancels buy order		
			4. System loads item details page	
Business Focus			System Focus	
Before event	Trigger	Buys items	Preconditions	Customer logged in
After event	Conclusion	Adds an order to the database and notifies the seller	Postconditions	New order entry in database

ID	C4			
NAME	View listing details			
Primary Actor	Customer			
Other Actors	None			
Description	Customer views the details of a particular listing			
Actor Action			System Response	
Typical Event Flow	1. Customer presses on a listing			
			2. System loads the item details	
Business Focus			System Focus	
Before event	Trigger	Views the details of a listing	Preconditions	Customer logged in
After event	Conclusion	Generates page with more information about the item	Postconditions	Item details displayed

29

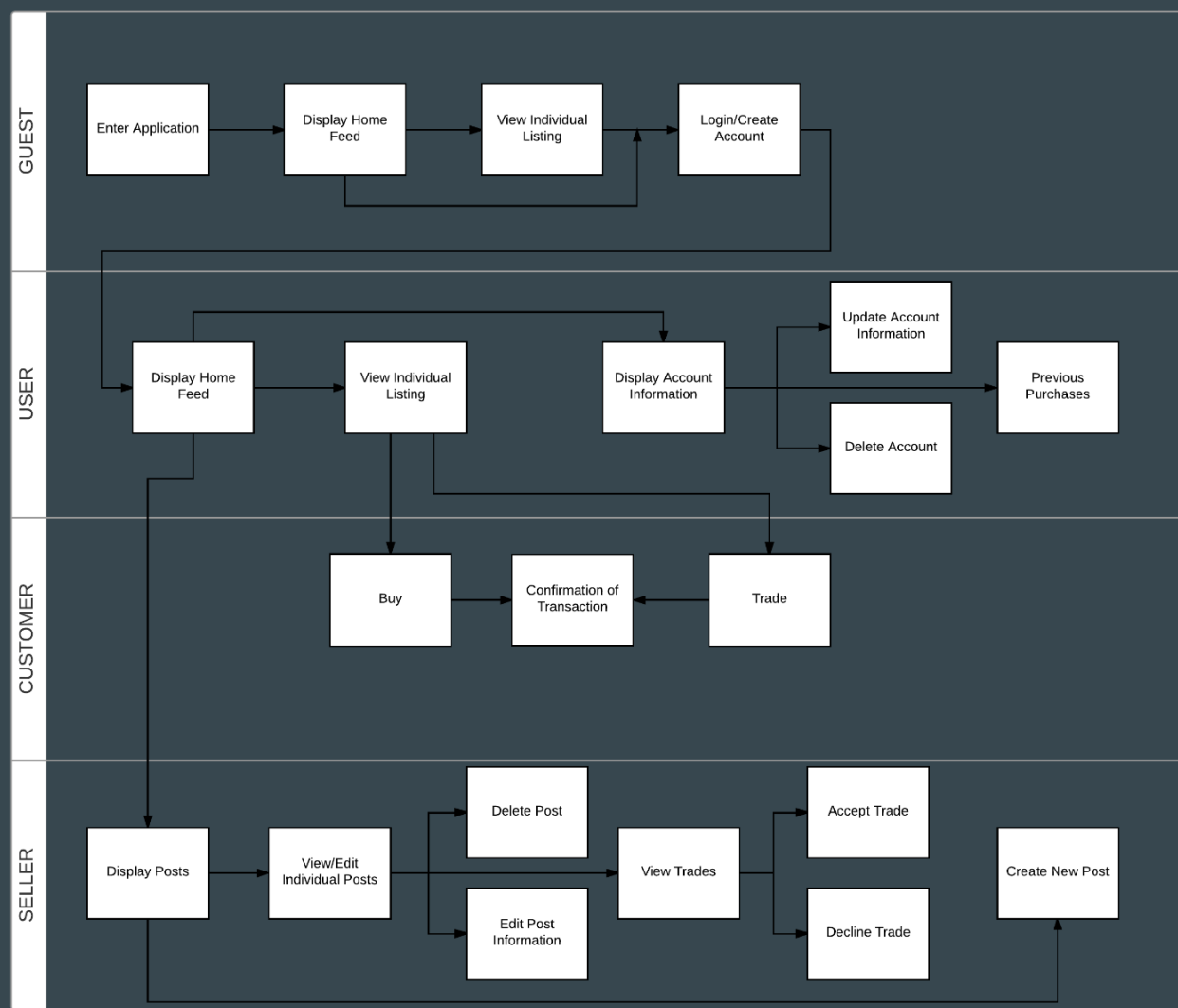
ID	T1			
NAME	Offer a trade			
Primary Actor	Customer			
Other Actors	Seller			
Description	Customer offers to trade their item for seller’s item			
Actor Action			System Response	
Typical Event Flow	1. Customer presses the Trade button			
	2. Customer enters items to trade			
			3. System confirms trade offer, adds trade entry to database	
			4. System notifies seller of trade offer	
Alternate Event Flows	1. Customer presses the Trade button			
	2. Customer enters items to trade			
	3. Customer cancels trade offer			
Business Focus			System Focus	
Before event	Trigger	Offers trade	Preconditions	Customer logged in
After event	Conclusion	Adds a trade to the database	Postconditions	Trade entry added to database

ID	T2			
NAME	Accept trade			
Primary Actor	Seller			
Other Actors	Customer			
Description	Seller accepts Customer's trade offer			
Actor Action			System Response	
Typical Event Flow	1. Seller accepts a trade offer			
			2. System confirms trade entry	
			3. System marks trade entry as completed	
			4. System notifies Customer of payment information	
Business Focus			System Focus	
Before event	Trigger	Accepted trade	Preconditions	Seller logged in
After event	Conclusion	Update trade data to close trade and email offering customer	Postconditions	Trade entry completed
				31

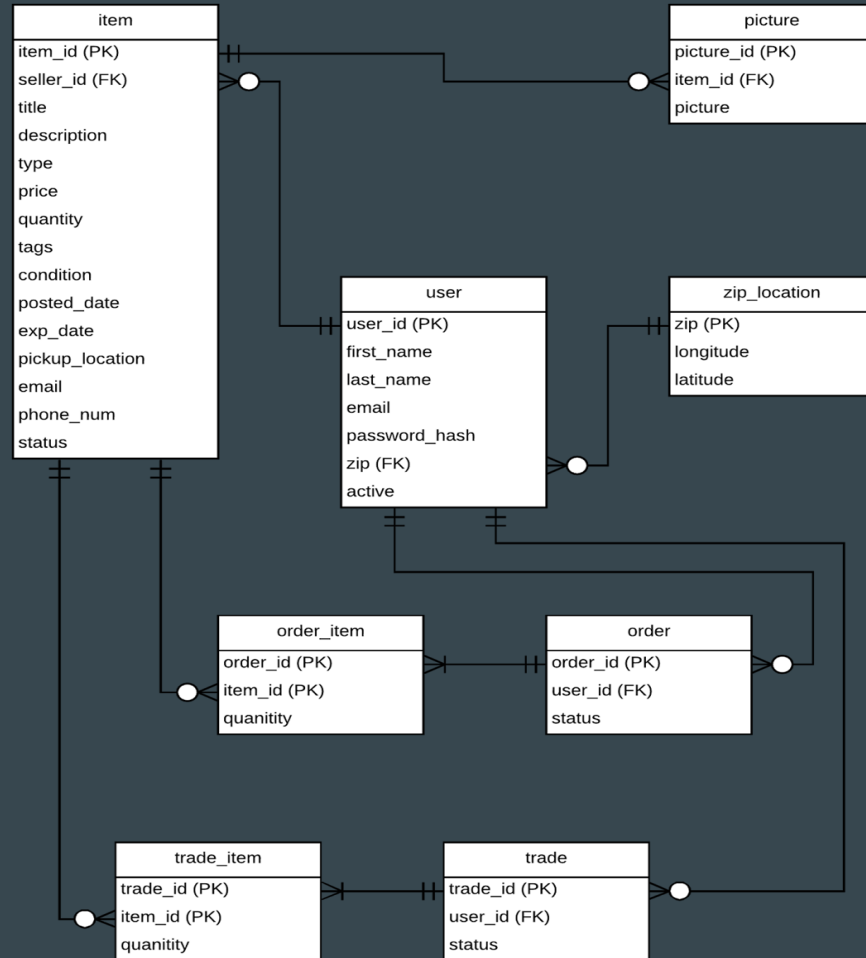
ID	S1			
NAME	List an item			
Primary Actor	Seller			
Other Actors	None			
Description	Seller posts an item for sale			
Actor Action			System Response	
Typical Event Flow	1. Seller submits form with item information			
			2. System adds item to database	
			3. System generates Home Feed	
			4. Returns to Home Feed	
Business Focus			System Focus	
Before event	Trigger	Lists an item for sale	Preconditions	Seller logged in
After event	Conclusion	Adds new listing to database	Postconditions	Item listed

ID	S4			
NAME	View trade offers			
Primary Actor	Seller			
Other Actors	None			
Description	Seller views current trade offers from customers			
Actor Action			System Response	
Typical Event Flow	1. Seller presses the Post tab			
			2. System loads the Post page	
	3. Seller presses the View Trades button			
			4. System loads trade offers	
Business Focus			System Focus	
Before event	Trigger	Clicks button to view trade offers	Preconditions	Seller logged in
After event	Conclusion	Sent to trade offers page	Postconditions	None

Cross Functional Map

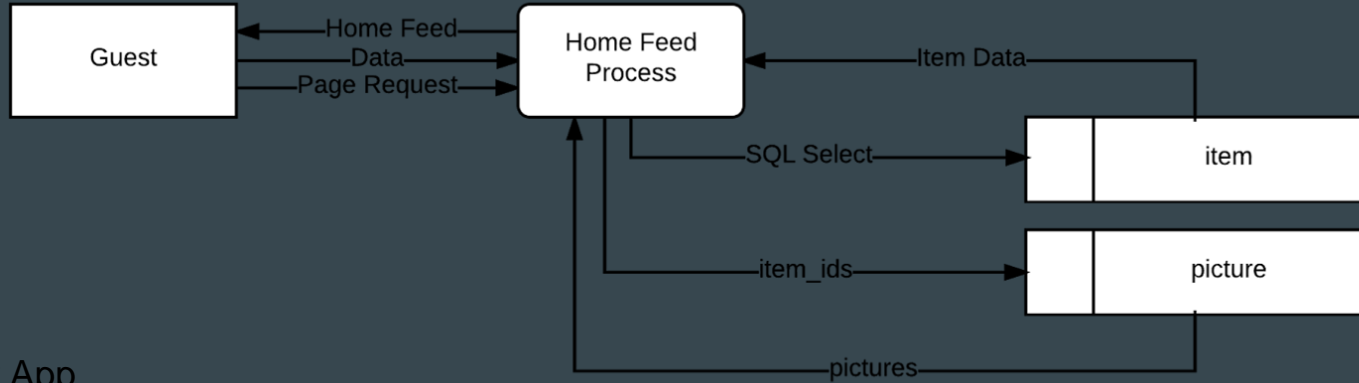


Logical Data Model



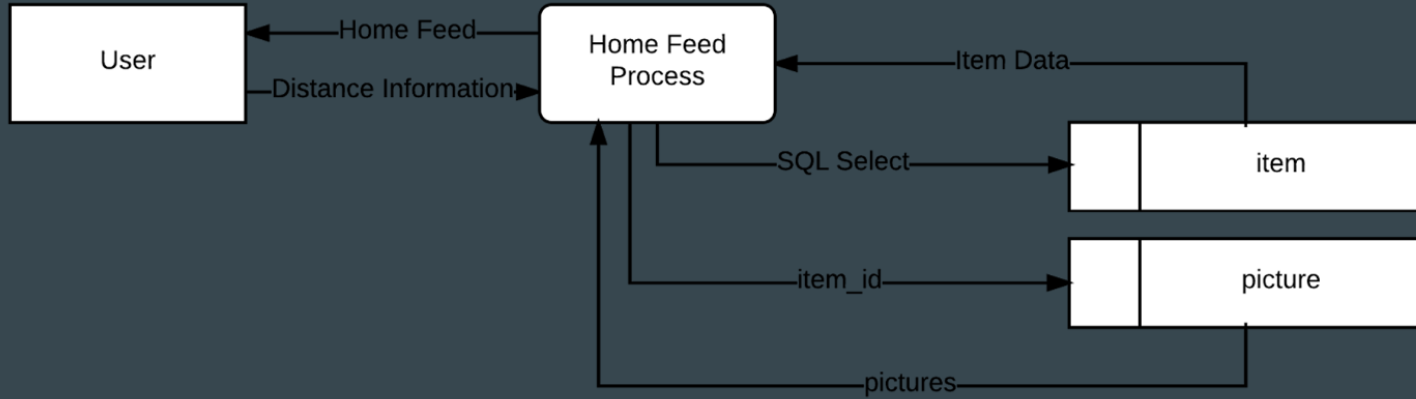
Logical Process Models

Entering the App



#Enter App
Request Home Feed
Get items within distance
Get item pictures
Generate page
Return page

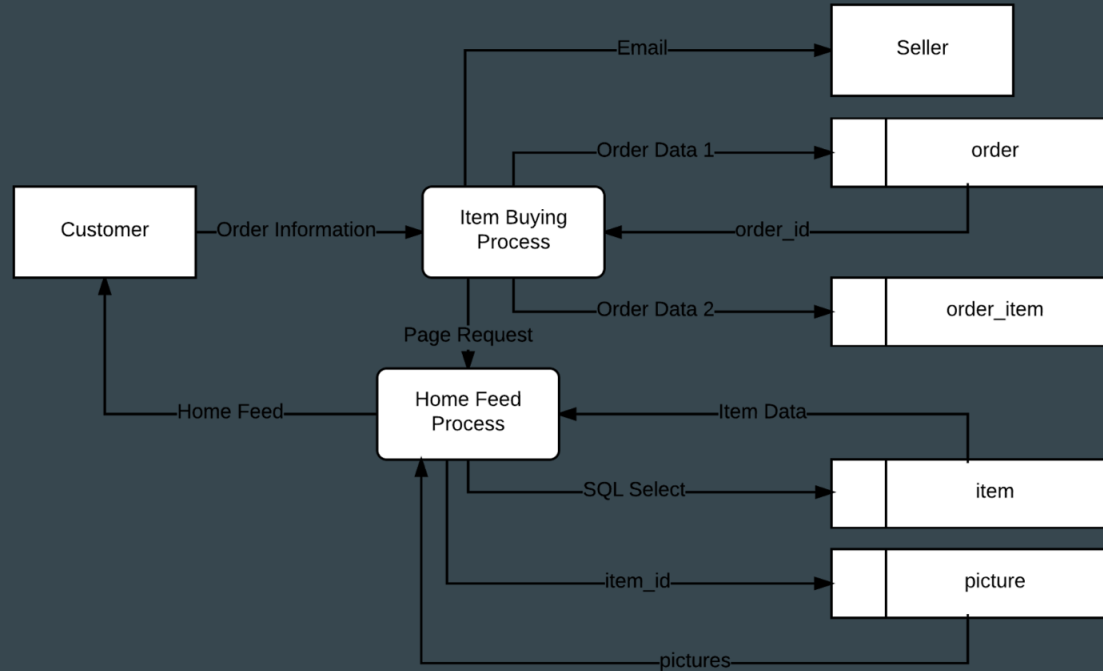
Changing Distance



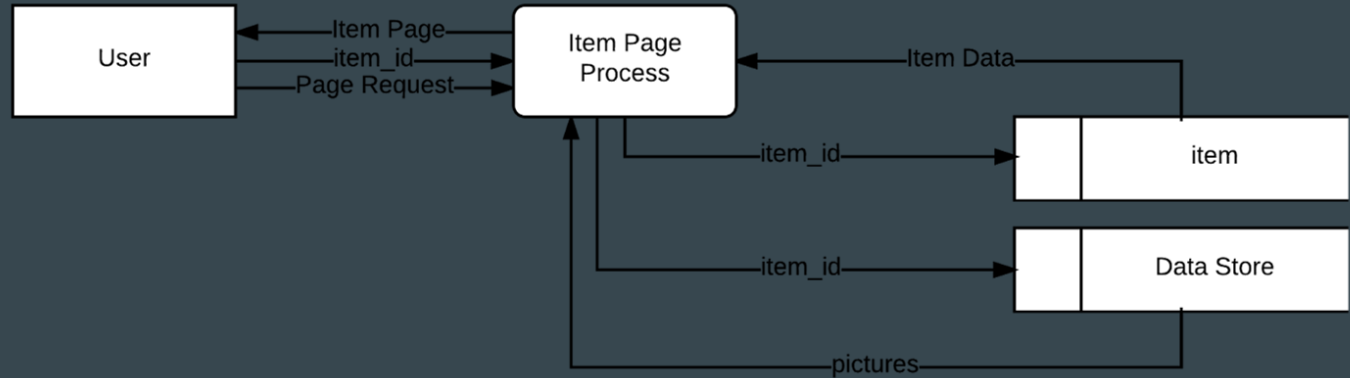
#Changing Distance
Get new distance
Generate Home Feed
Return Home Feed

Buying an Item

#Buying
Get order info
Create new order in order table
Get new order_id
Create associations of items to new
order in order_item table
Email the seller of the item
Generate Home Feed
Return Home Feed



Viewing Item Information



#Item Page
Get item_id
Get item data from item table using item_id
Generate Item Page
Return Item Page

Offering a Trade

#Setting Up a Trade

Get trade info

Create new trade in trade table

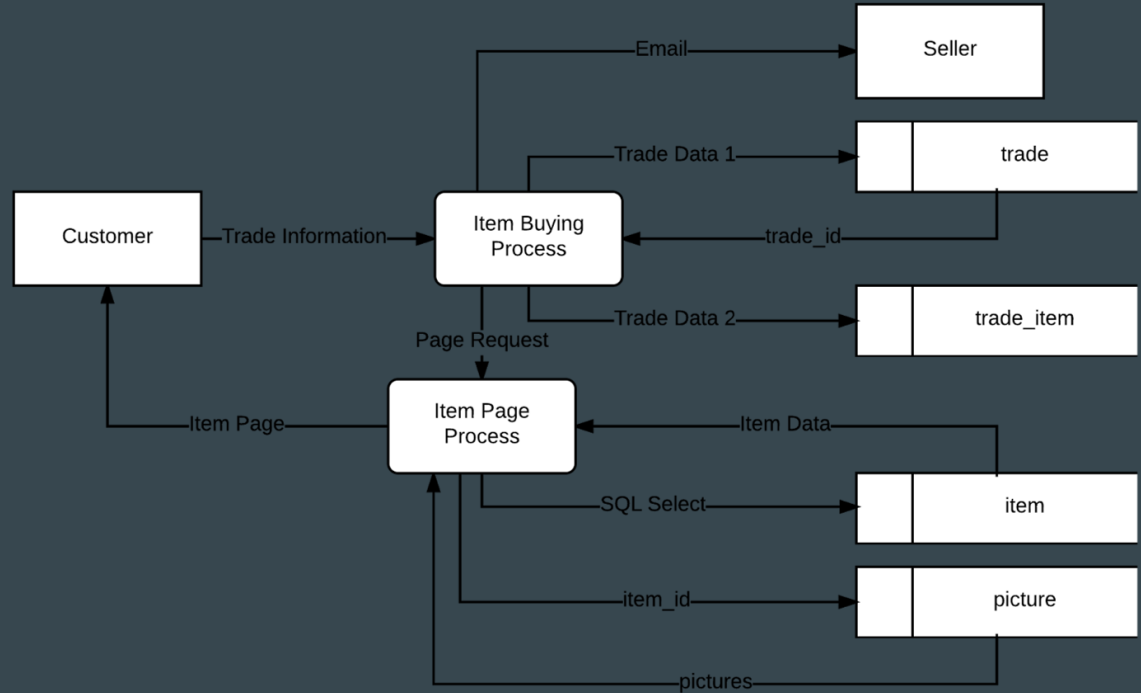
Get new trade_id

Create associations of items to new trade in trade_item table

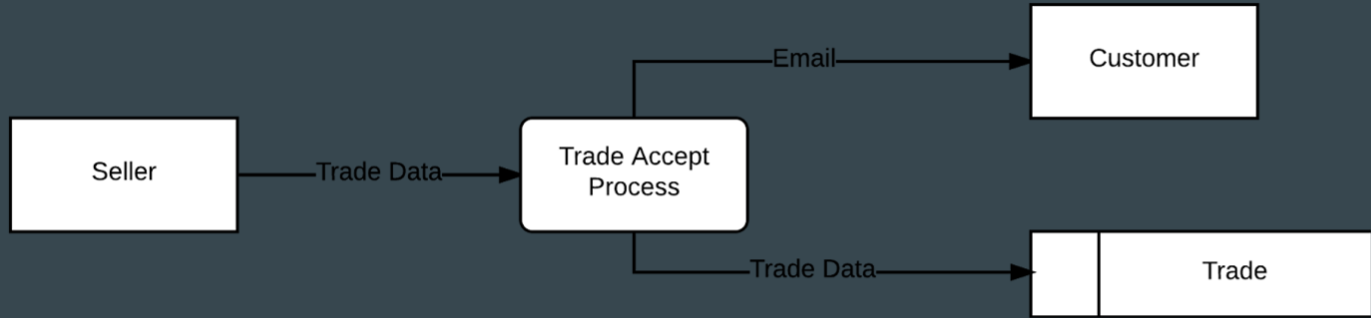
Email the seller of the item

Generate Item Page

Return Item Page



Accepting a Trade



#Accepting Trades

Get trade data

Change status of trade in trade
table

Email trading customer

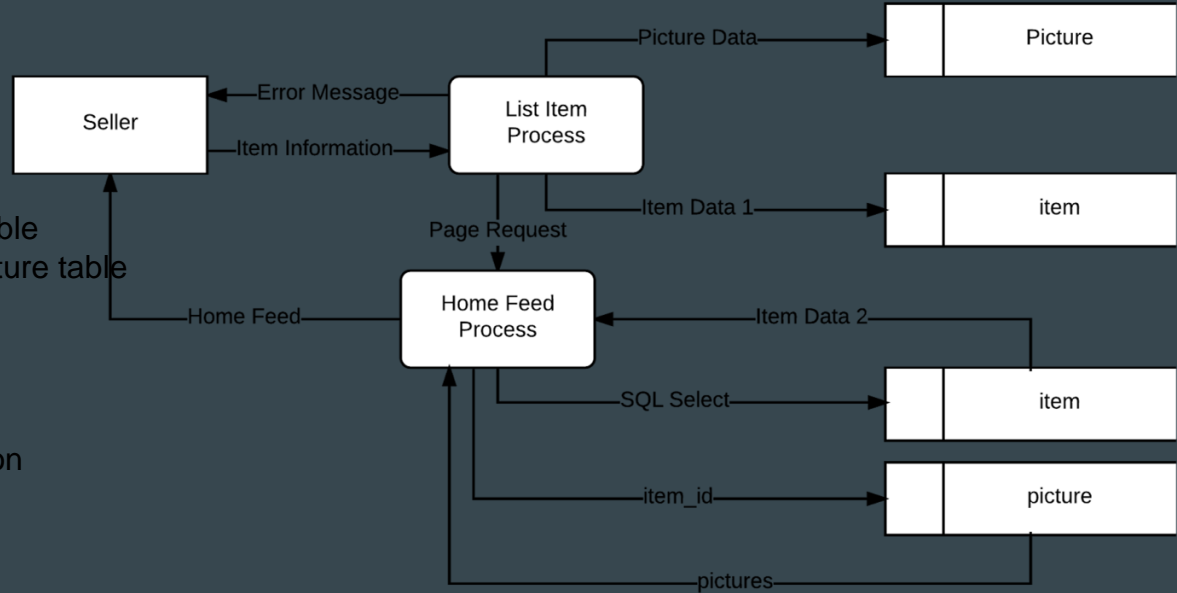
Listing an item

#New Listing
Get item info
if(info is valid):

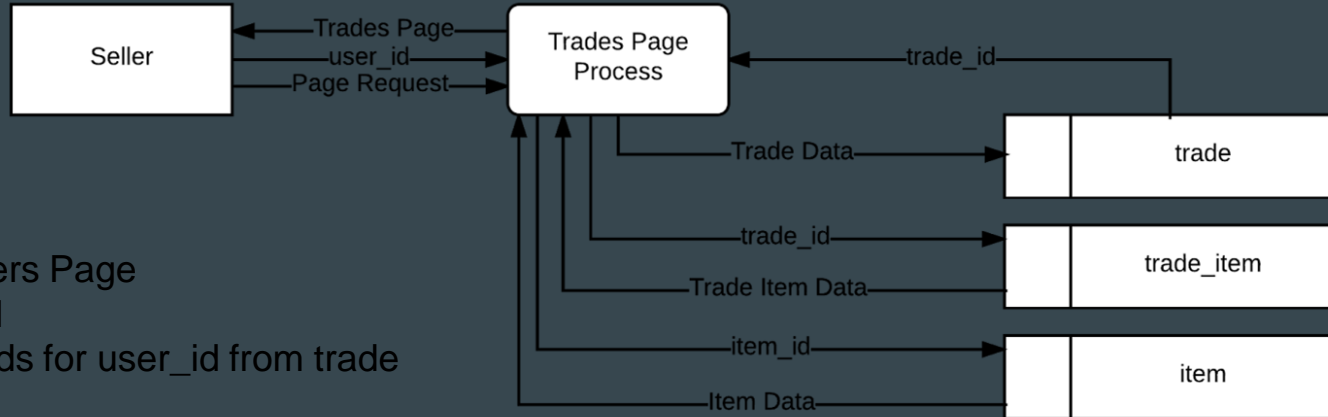
Add new item to item table
Add item pictures to picture table
Generate Home Feed
Return Home Feed

else:

Return error message
Request valid information



Viewing Trade Offers



#Trade Offers Page

Get user_id

Get trade_ids for user_id from trade table

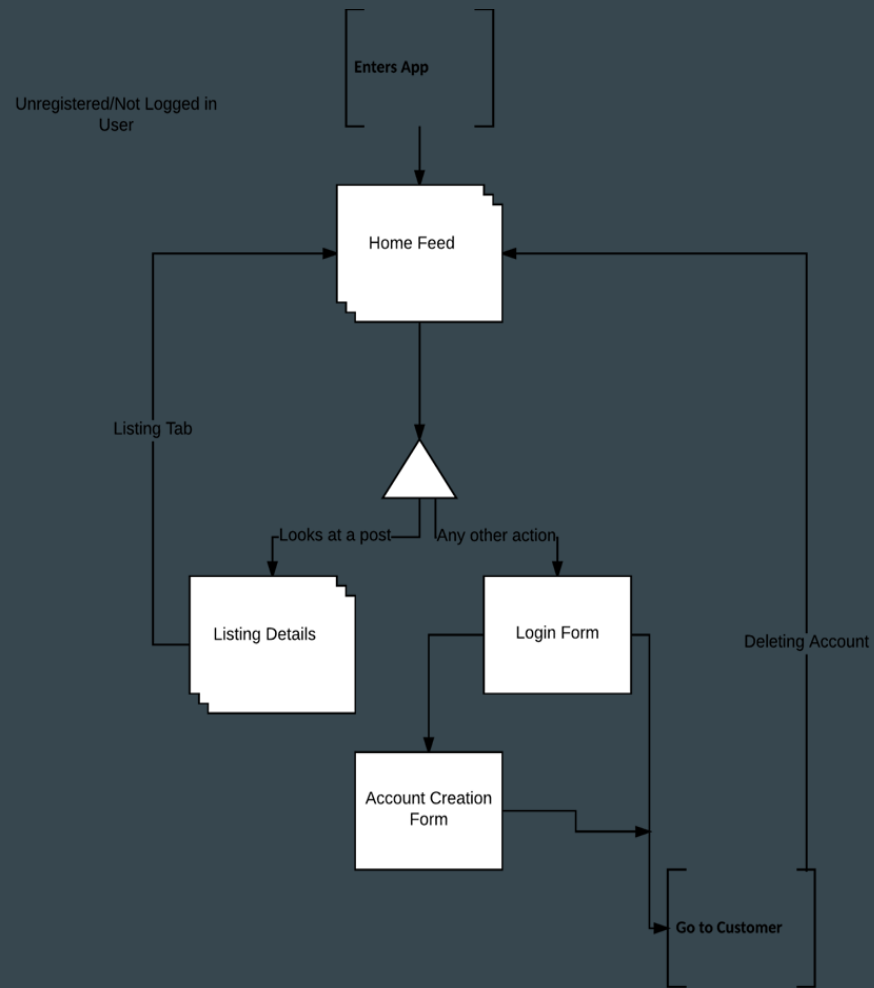
Get item_ids associated with trades from trade_item

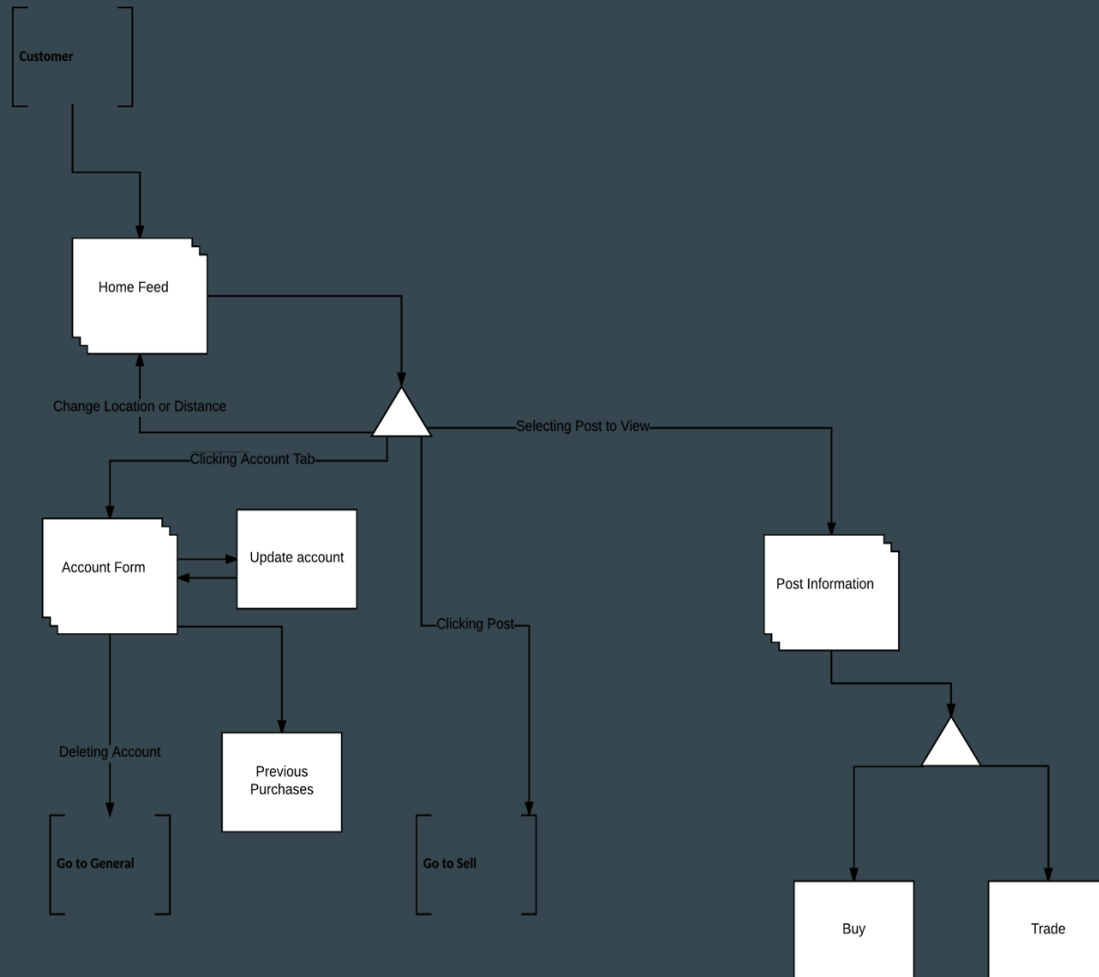
Get item data of item_ids

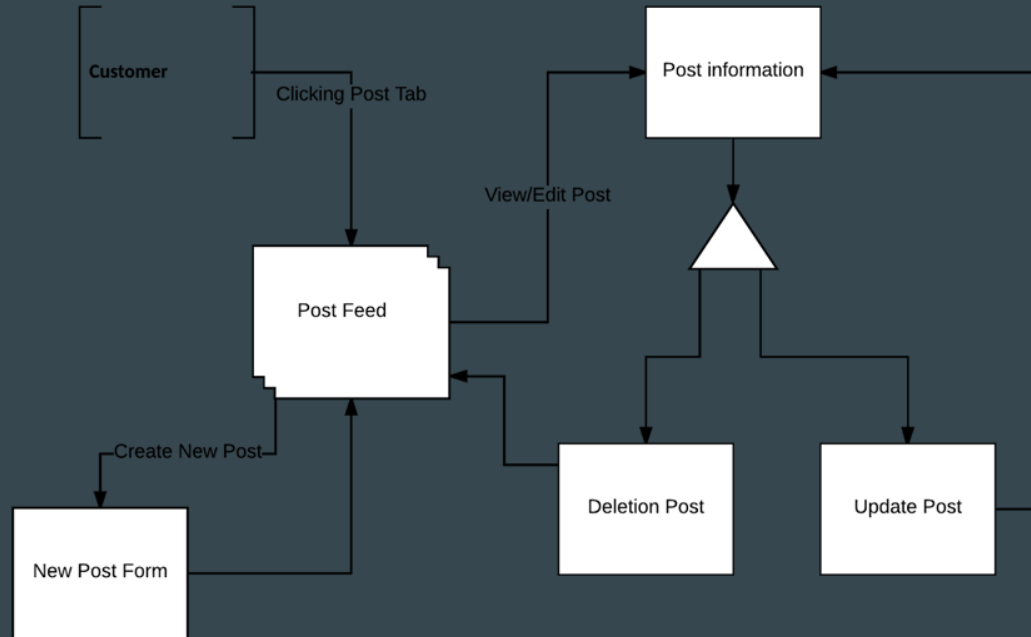
Generate Trade Offers Page

Return Trade Offers Page

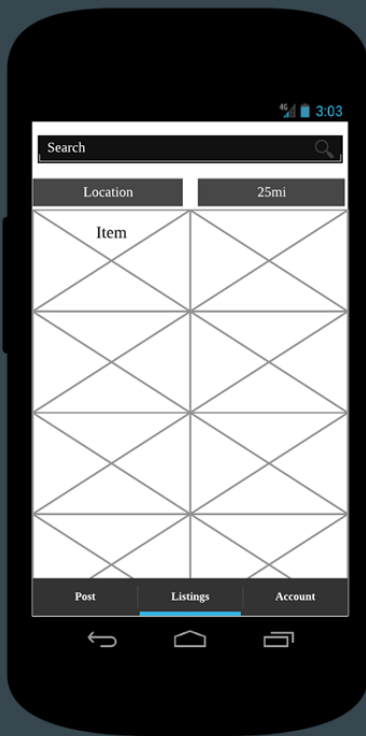
User Experience Diagram



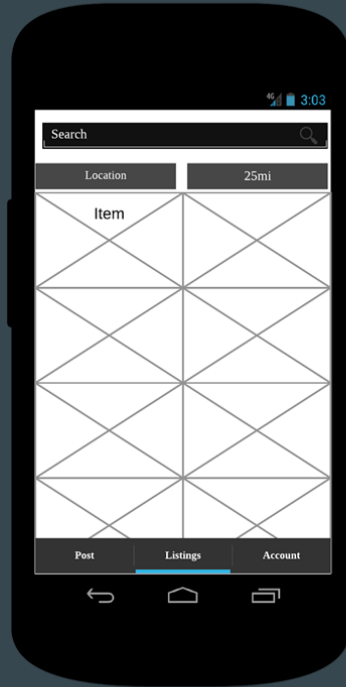




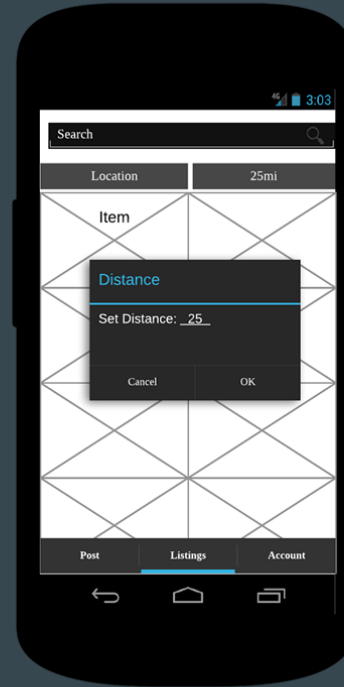
Wireframes



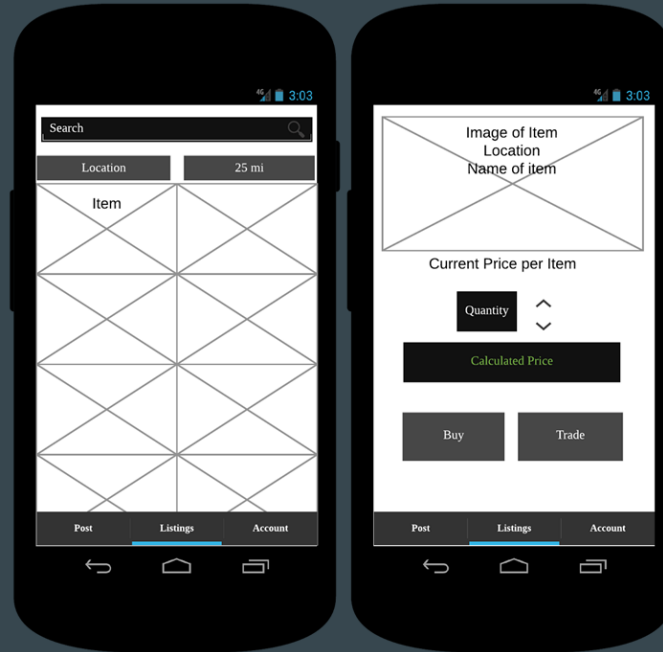
Landing Page for customer



Landing Page for Users

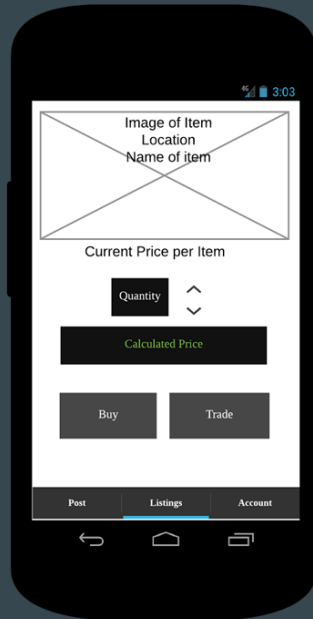


Pop Up for Distance

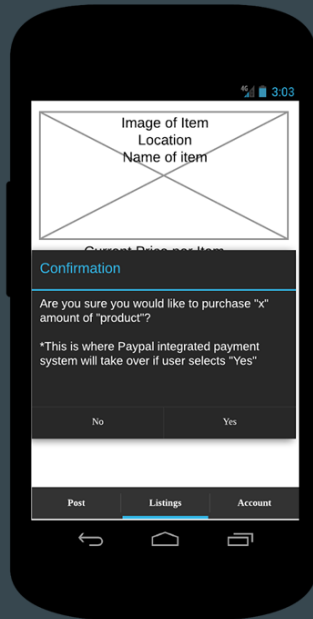


Customer Selects and Item
on the Listings Page

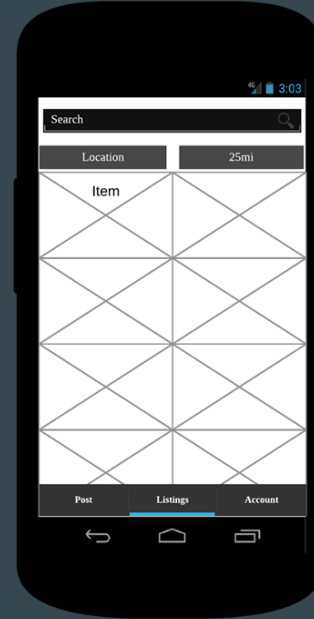
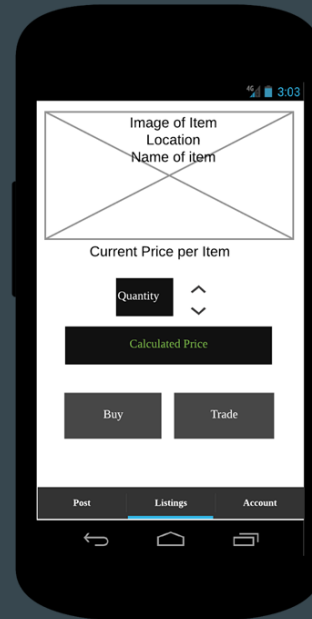
Customer is sent to a page
with more details about the
product and an option to Buy
or Trade



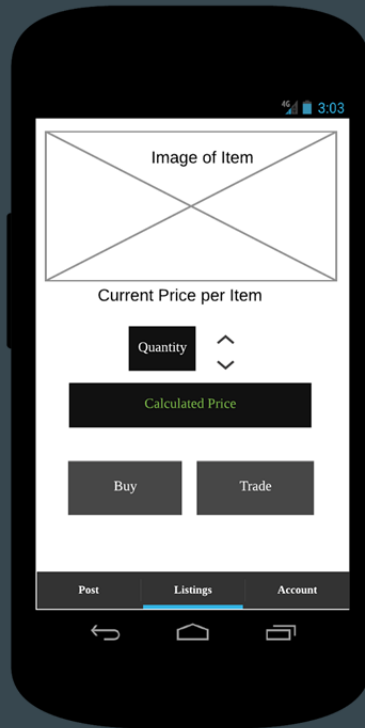
Customer Selects to buy
and item



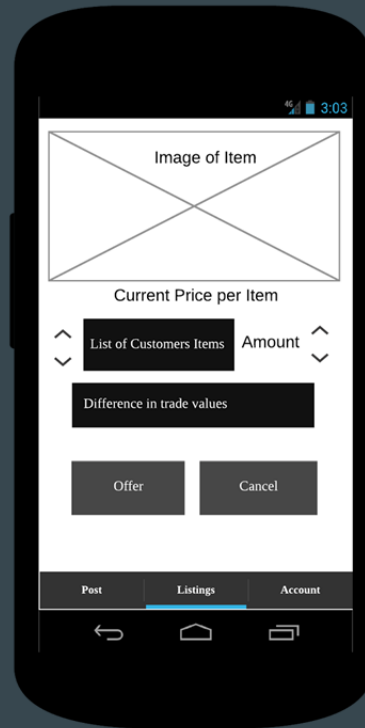
Customer Selects to buy
and item



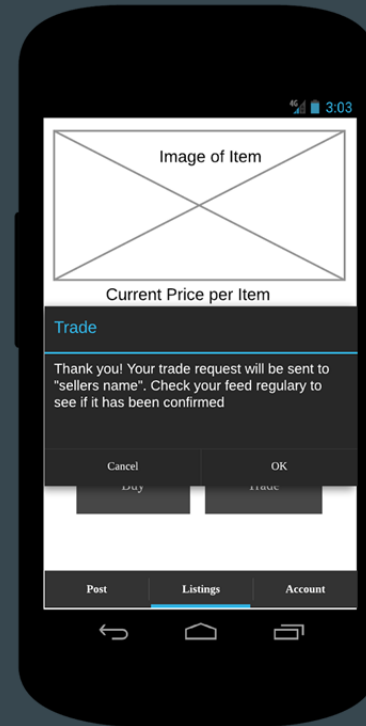
Customer is sent to the
Home Feed



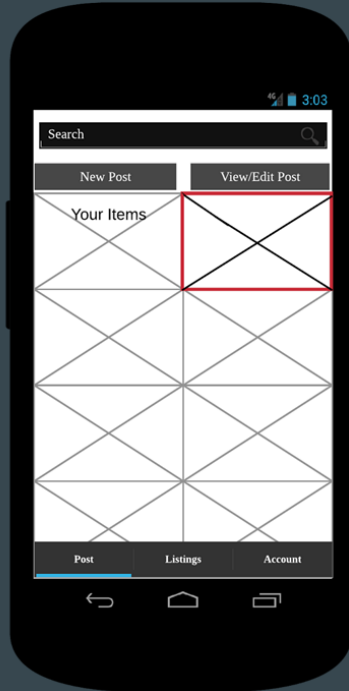
Customer selects option to trade



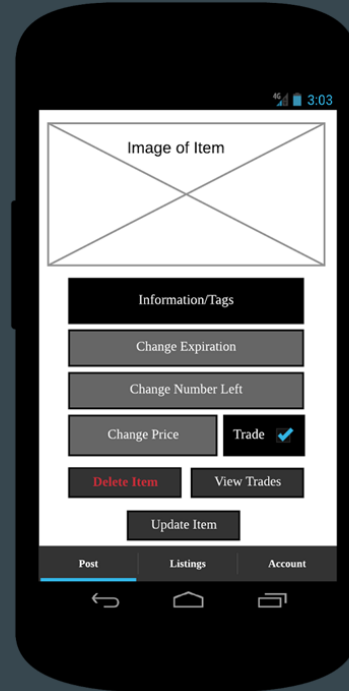
Trade Options



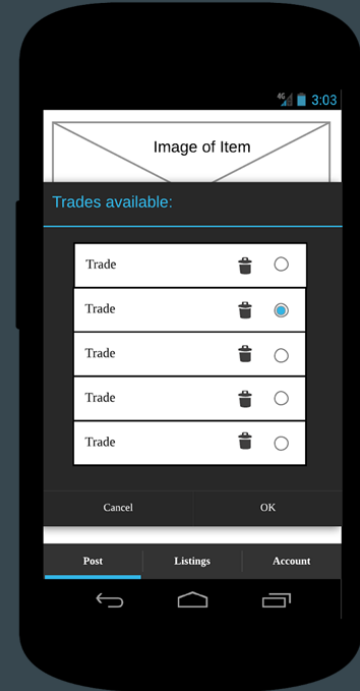
Trade is added to the database. Customer is reminded to check regularly for confirmation from the seller



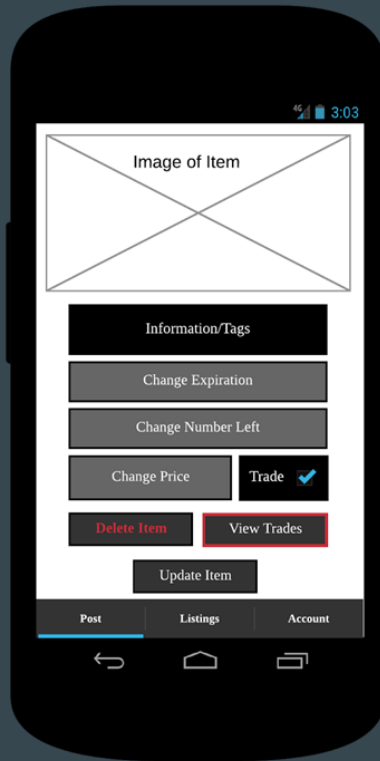
Selecting Item to trade



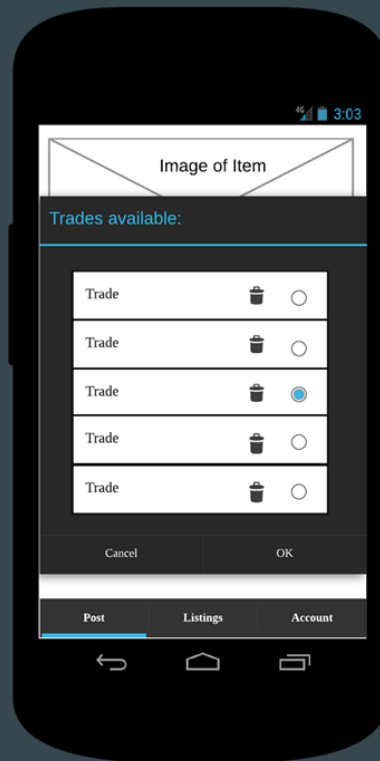
View/Edit Post Page



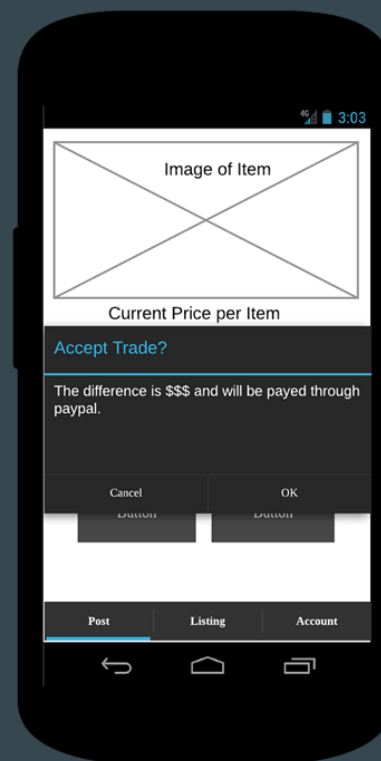
Select Trade



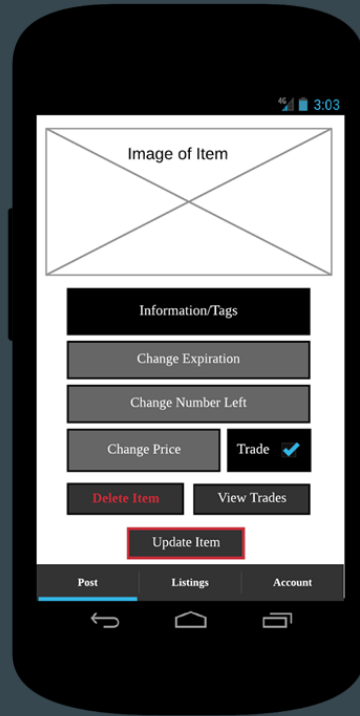
Seller View of Existing Post



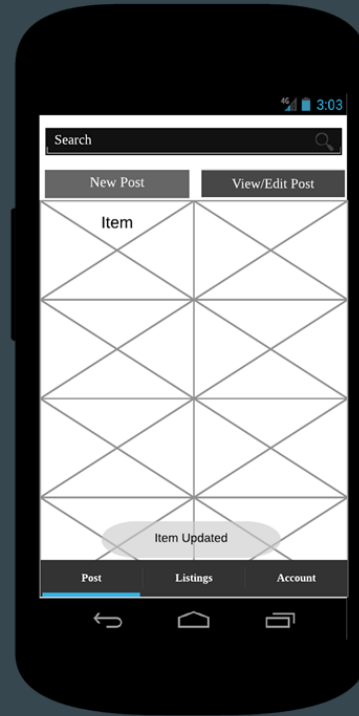
Trade selection



Trade confirmation



Seller View for new post



Item Updated

Non-Functional User Requirements

Usability

- Buy items from other users
- Users can sell items on the app
- Simple trading
- Easy browsing for unregistered users

Accessibility

- English language
- Usable for ‘non-technical’ users
- User friendly to colorblind users

Availability

- Available at all times
- Notification emails sent if system is down
- Page Error alerts

Documentation and Training

- Basic user manual
- Minimal training for the average user
- Database administrator

Non-Functional System Requirements

Performance

- Not intended to use a large amount of resources
- Not sluggish or slow
- Little battery life usage

Capacity

- Small server to start with
- 5GB standard storage
- Limited GET and PUT requests
- Future upgradability

Security

- User must be logged in to do certain tasks
- Limited session timers
- Secure passwords
- SQL sanitization

Longevity

- Continue growing in the local area, hopefully much larger
- Limited by small budget and server prices

Conclusion

- Custom Build - Gardener's Exchange
- Meets all customer requirements

