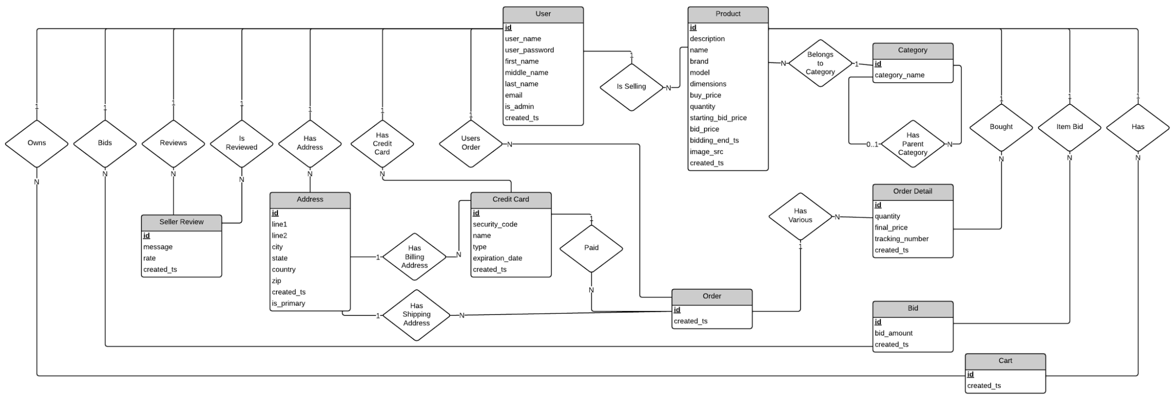
MOD Store Phase 1 Report

ICOM5016 Project - Dr. Manuel Rodríguez

Manuel Marquez
Daniel Santiago
Omar Soto

Index

Entity Relationship Diagram



(Full resolution ER will be included in a separated file)

Description of Entities:

1. **Address**
   This is an entity that contains this attributes: id, line1, line2, city, state, country, zip, created_ts, is_primary. The attribute line2 and state are optional. Finally, the attribute is_primary is a boolean attribute that will indicate is that is the default address for the users who belongs to.

2. **Bid**
   This is an entity that contains this attributes: id, bid_amout and created_ts. The attribute bid_amount has the amount that an User bidded for a particular Product, that quantity is cummulative, that means that will store the total after the User bidded (before price + increment).

3. **Cart Element**
   This is an entity that contains this attributes: id and created_ts. Besides it doesn't have relevant attributes, this class is a mapping between an User and a Product that the user wants to have in his own Cart. Each record of this entity for a user, will generate his cart at the moment, that means that its not historical, it will only contain the Cart Elements at the moment for a particular User.

4. **Category**
   This is an entity that contains this attributes: id and name. This will contain all the categories with his respective parent (using a relation to itself), so the record will work on a recursive way. The categories are important because they help us to filter the products

that are added into the Store.

5. **Credit Card**
   This is an entity that contains this attributes: id, security_code, name, type, expiration_date and created_ts. We ask for a name because an User can have Credit Cards with different names than his real name, for example if he is adding a Credit Card of one of his family members. The type is number based, we have a mapping between the types of credit cards and the name of the Credit Card that is specified in the code. The expiration date has the format as shown in the credit cards MM/YYYY.

6. **Order**
   This is an entity that contains this attributes: id and created_ts. Besides it has few attributes, this entity is very important because is the head of an Order, we do not wanted to replicate the credit card, shipping and user when a user is buying in a package (an Order). What will happen is that if the user decides to buy the items of his cart, an Order will be created containing the master information and each item in another entity called Order Detail, when a user win an auction, a single order is generated for that particular item. This entity has very important relations that contains the shipping address, the paying credit card and the user that has created the order.

7. **Order Detail**
   This is an entity that contains this attributes: id, quantity, final_price, tracking_number and created_ts.The attribute quantity is important because if a product supports buy out, the user can buy many items at once. The attribute final_price was added because the user is allowed to change the buyout price in the process of selling his items, and in that case we want to know wich was the price at the time that the user bought the item. The tracking number is what we use to know is the seller has shipped the product, also this is very important for the buyer to track his package; In this time almost every postal company has packages tracking within his services.

8. **Product**
   This is an entity that contains this attributes: id, description, name, brand, model, dimensions, buy_price, quantity, starting_bid_price, bid_price, bidding_end_ts, image_src and created_ts. Almost all the attribute are strings that will be used to describe the product that is being sold but there are other attributes that are important to comment out. The buy_price can be left blank. The quantity will be limited to one when the product has enable the auction, if not, the product can have quantities greater than one. The bid_price is also optional and the bidding_end_ts has the timestamp of the auction ending. The image_src will contain the path of the image of that Product.

9. **Seller Review**
   This is an entity that contains this attributes: id, message, rate and created_ts. The user that reviews another user can leave a message that describes his experience with that

seller. Also its required to leave a rate (from 1 to 5), using that we will get the Sellers Rate by averaging the rate in this records.

10. **User**
    This is an entity that contains this attributes: id, user_name, user_password, first_name, middle_name, last_name, email, is_admin and created_ts. The user_password will be encoded. There is a boolean attribute, is_admin, that contains the permission that the user has.

NOTE: Almost all entities has an attribute created_ts that has the timestamp of the time of creation of the record.

Description of Relationships:

1. Owns
   This relation connects between the entities **User** and **Cart Element**. This relation is one to many because the **User** can have many **Cart Elements**. The entity **Cart Element** will have all the **Products** that the **User** has at the moment in his cart.

2. Bids
   This relation connects between the entities **User** and **Bids**. This relation is one to many because the **User** can bid in different **Products** and also the **User** can bid a lot of times for a single **Product**, depending on the situation. The entity **Bids** has all the **Products** that the **User** has placed a **Bid**.

3. Reviews
   This relation connects between the entities **User** and **Seller Reviews**. This relation is one to many because the **User** can create a **Seller Review** for each Seller. The entity **Seller Reviews** has all the reviews that has been done by the **User**, in this case represented as the reviewer.

4. Is Reviewed
   This relation connects between the entities **User** and **Seller Reviews**. This relation is one to many because an **User** can have many **Seller Reviews**. The entity **Seller Reviews** has all the reviews that have been done to the **User**, in this case represented as the reviewed.

5. Has Address
   This relation connects between the entities **User** and **Address**. This relation is one to many because an **User** can have many **Addresses**. The entity **Address** has all the addresses that are associated to an **User**.

6. Has Credit Card
   This relation connects between the entities **User** and **Credit Card**. This relation is one to many because the **User** can have many **Credit Cards**. The entity **Credit Card** has all credit cards that are associated to an **User**.

7. User Order
   This relation connects between the entities **User** and **Order**. This relation is one to many because the **User** can have many **Orders**. The entity **Orders** has all the orders that have been done by the **User**, an **Order** specifies the **Address** and **Credit Card** that will be used in that **Order** and in another entity we have the elements that were ordered.

8. Has Billing Address
   This relation connects between the entities **Address** and **Credit Card**. This relation is

one to many because one **Address** can be used as Billing Address for many **Credit Cards** but a **Credit Card** can have only one **Address.** The entity **Credit Card** must have an **Address** associated to it.

9. Has Shipping Address
   This relation connects between the entities **Address** and **Order**. This relation is one to many because one **Address** can be used as Shipping Address for many **Orders** but an **Order** can have only one **Address**. The entity **Order** must have an **Address** associated to it.

10. Paid
    This relation connects between the entities **Credit Card** and **Order**. This relation is one to many because one **Credit Card** can be used to Pay many **Orders** but an **Order** can have only one **Credit Card**. The entity **Order** must have an **Credit Card** associated to it.

11. Has Various
    This relation is between the entities **Order** and **Orders Detail**. This relation is one to many because one **Order** can have various **Order Details** but an **Order Detail** can have only one **Order**. The entity **Order Detail** contains the information about the product that has been buyed or sold (depending the perspective), tracking number and the final price.

12. Bought
    This relation is between the entities **Order Detail** and **Product**. This relation is one to many because one **Product** can be associated to various **Order Details** but an **Order Detail** can have only one **Product**. The entity **Products** has all the information of the specified product, including price, starting bid price, duration of auction and other important attributes.

13. Item Bid
    This relation is between the entities **Bid** and **Product**. This relation is one to many because one **Product** can has many **Bids** but a **Bid** can have only one **Product**. The entity **Bids** has the information about what **Product** has been bidded, the user that bidded and the bid cost at that point.

14. Has
    This relation is between the entities **Cart Element** and **Product**. This relation is one to many because one **Product** can be in manny **Cart Elements** but a **Cart Element** can have only one **Product**. The entity **Cart Element** has the information about an specific **Product** that an **User** has in his cart, each record of the **Cart Element** represent an element in the user's cart.

15. Belongs to Category

This relation is between the entities **Product** and **Category**. This relation is one to many because a **Category** can be in many **Products** but a **Product** can have only one **Category**. The entity **Categories** contain the name and parent **Category** for each category.

16. Has Parent Category
    This relation is between the entity **Category** and itself. This relation is zero or one to one because a **Category** may have a parent **Category**. When the **Category** is a main **Category** the parent **Category** is represented using -1.

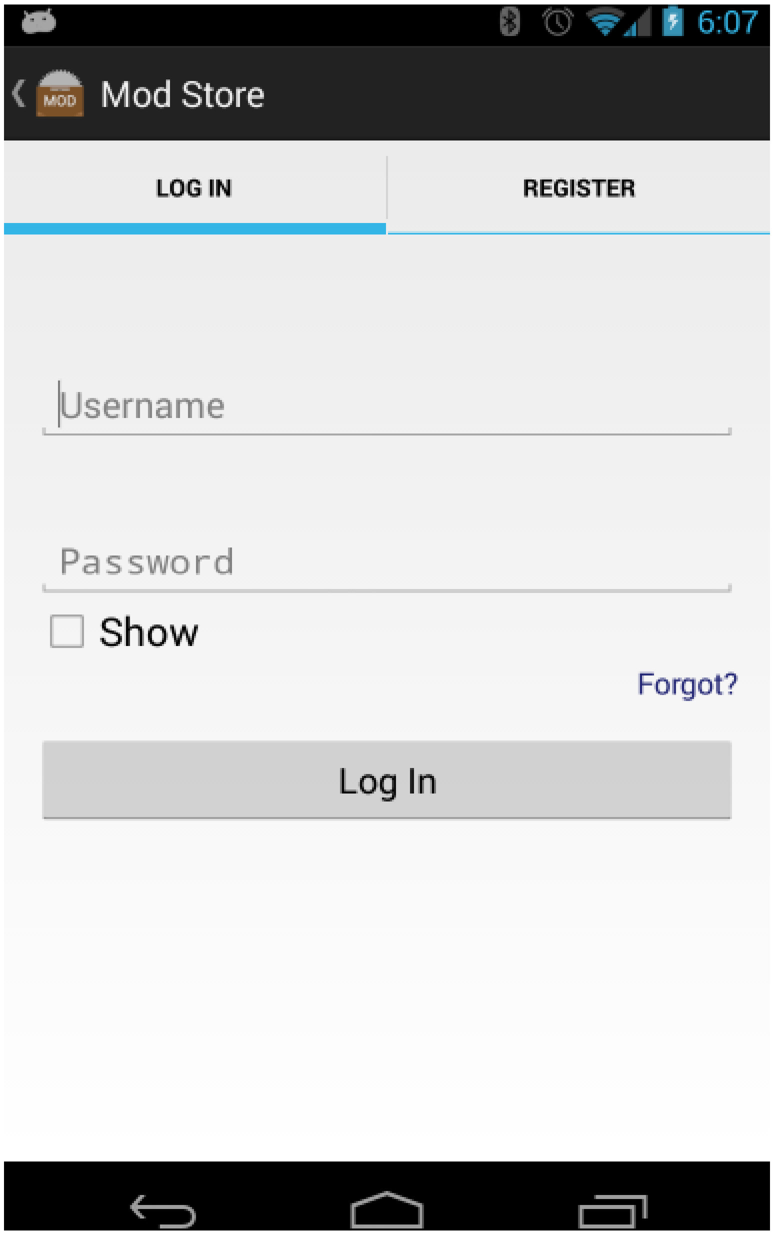
17. Is Selling
    This relation is between the entity **User** and **Product**. This relation is one to many because one **User** is selling many **Products** but the **Products** have just one **User** as seller.

Tasks Supported and Mockups

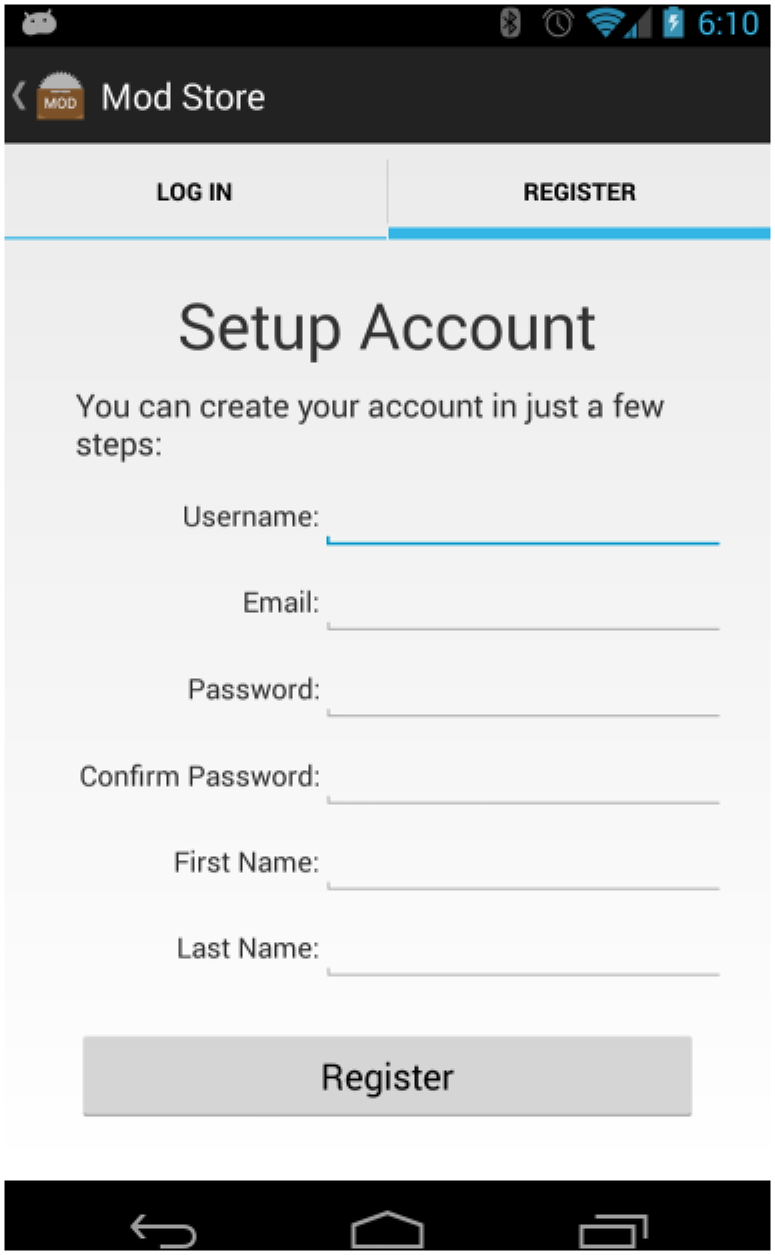
As specified by Phase 1 document on the class's website

1. Account

a. Login
b. Register
c. Profile Information
d. Address Information
e. Credit Card Information



User login view, includes password recovery option.



6:10

Mod Store

LOG IN

REGISTER

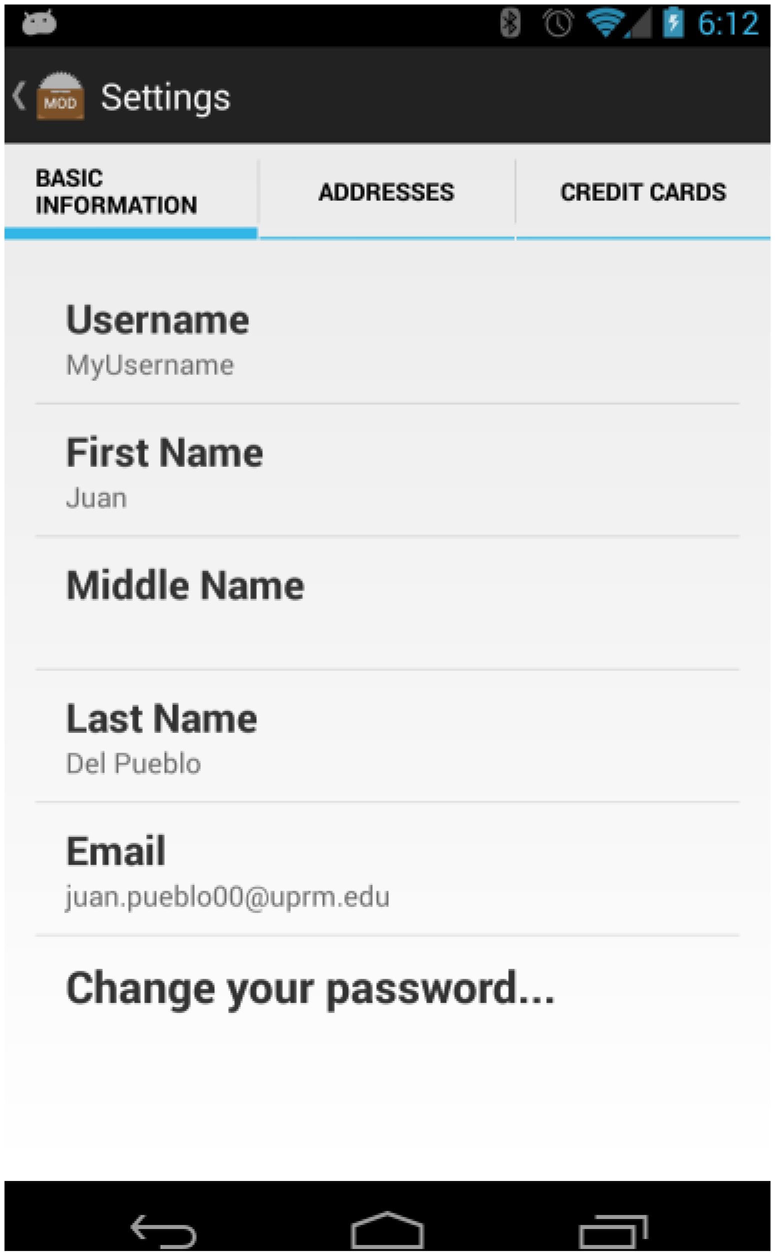
Setup Account

You can create your account in just a few steps:

Username:	
Email:	
Password:	
Confirm Password:	
First Name:	
Last Name:	

Register

New user register form and fields.All fields are required.



Settings

Username
MyUsername

First Name
Juan

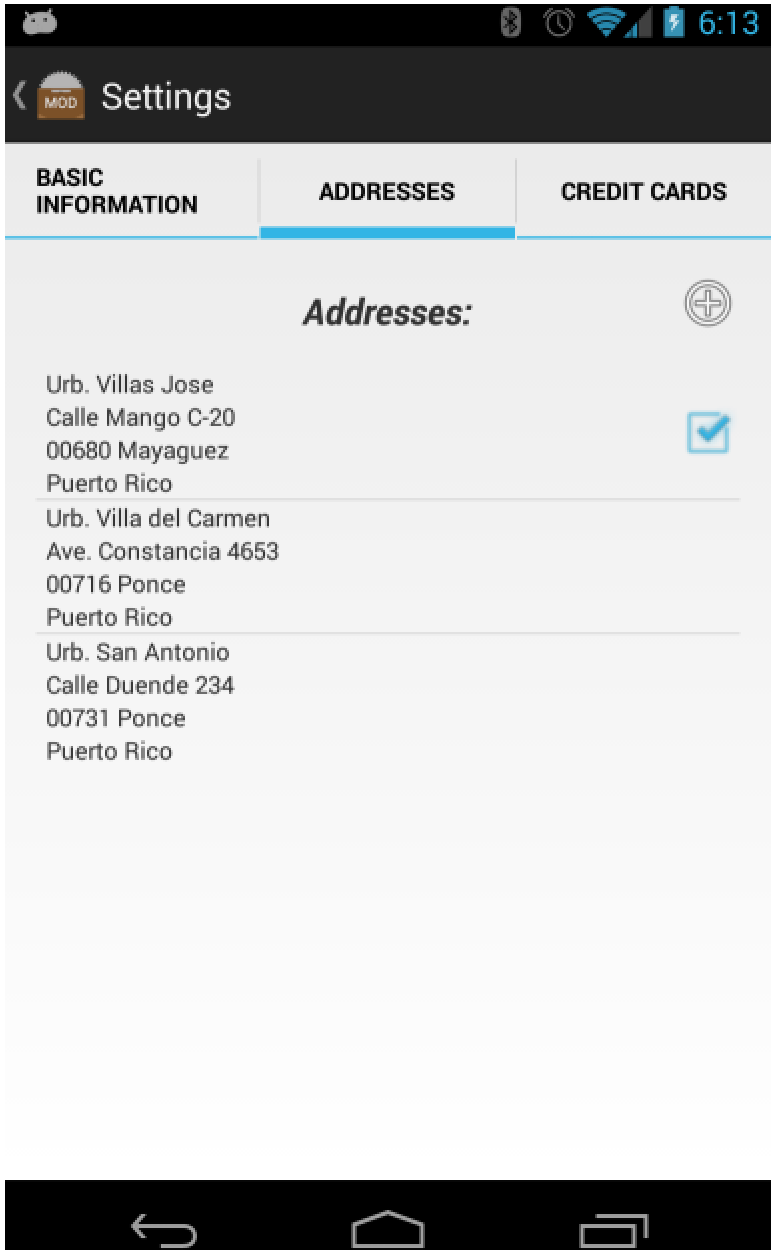
Middle Name

Last Name
Del Pueblo

Email
juan.pueblo00@uprm.edu

Change your password...

User profile view. Each element can be edit by clicking on it, that will open a dialog box that will allow the user to change the profile item. For the password, the user will be prompted twice for the password to avoid a misspelled new password.



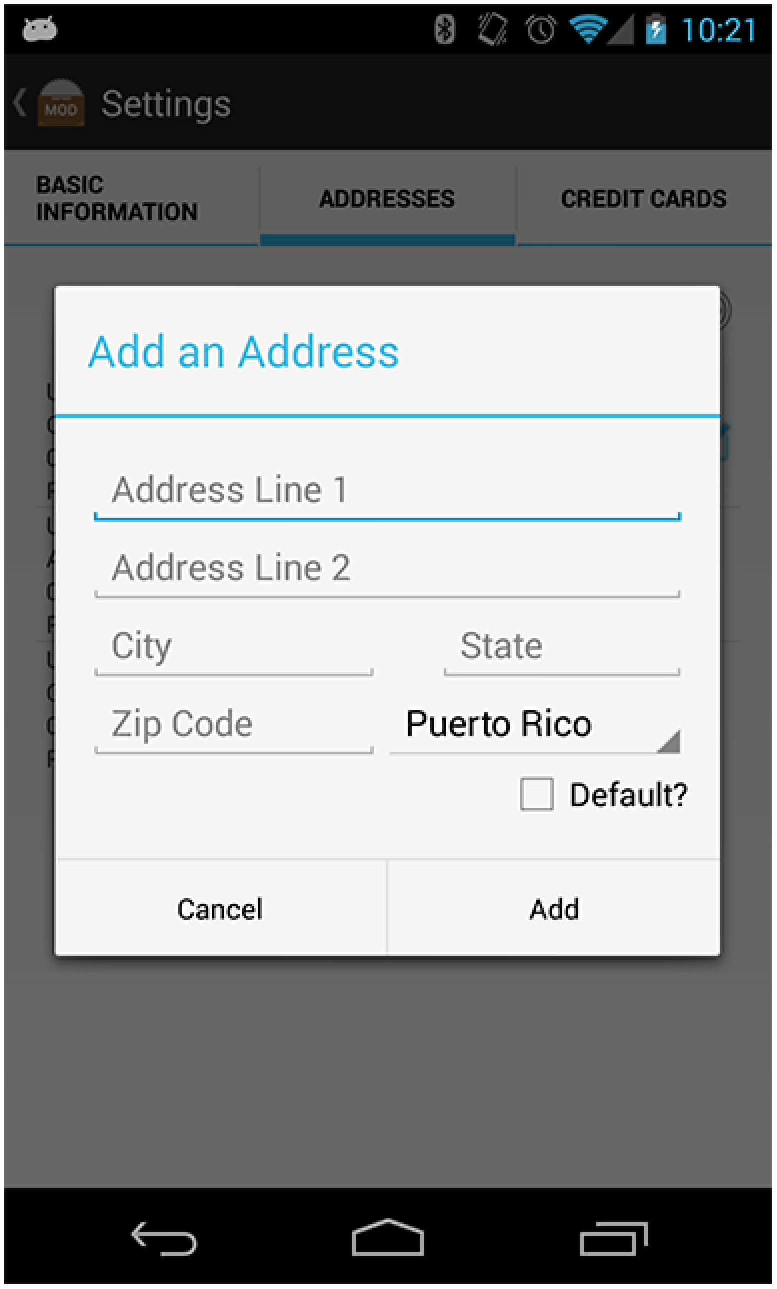
Addresses:

Urb. Villas Jose
Calle Mango C-20
00680 Mayaguez
Puerto Rico ✓

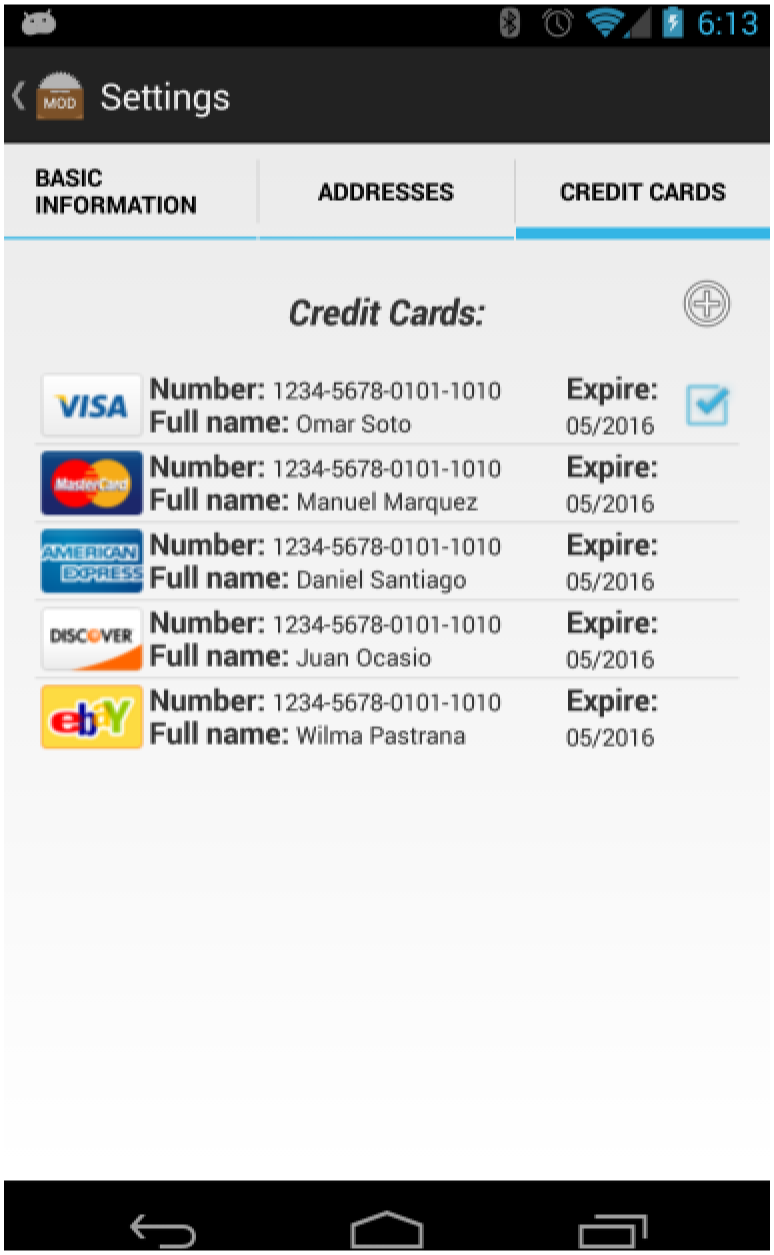
Urb. Villa del Carmen
Ave. Constancia 4653
00716 Ponce
Puerto Rico

Urb. San Antonio
Calle Duende 234
00731 Ponce
Puerto Rico

User addresses view. More addresses can be added using the plus button at the top right corner, that will open a Dialog asking for the information. Also, clicking on the list will show a dialog that allows the user to edit the clicked Address.



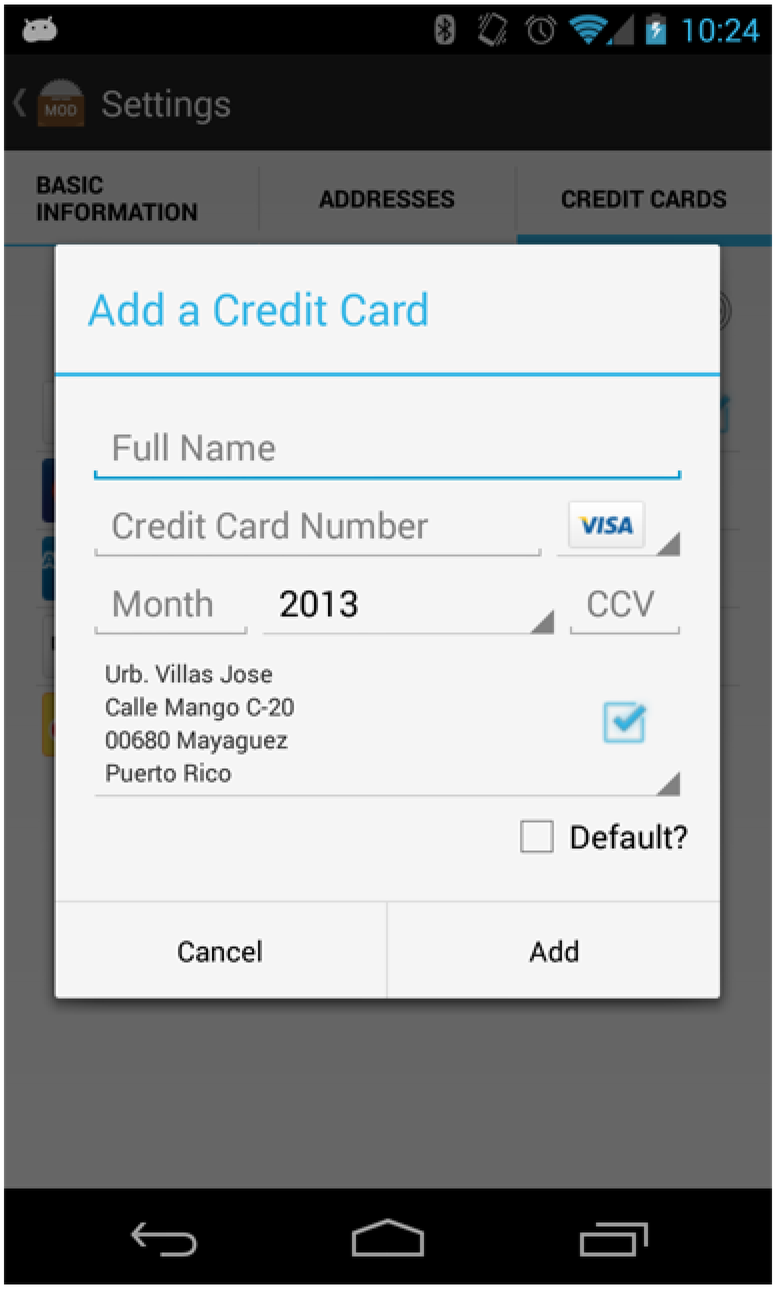
This is the dialog that is used to create or update an Address.



Credit Cards:

VISA	**Number:** 1234-5678-0101-1010 **Full name:** Omar Soto	**Expire:** 05/2016 ✓
MasterCard	**Number:** 1234-5678-0101-1010 **Full name:** Manuel Marquez	**Expire:** 05/2016
AMERICAN EXPRESS	**Number:** 1234-5678-0101-1010 **Full name:** Daniel Santiago	**Expire:** 05/2016
DISCOVER	**Number:** 1234-5678-0101-1010 **Full name:** Juan Ocasio	**Expire:** 05/2016
ebay	**Number:** 1234-5678-0101-1010 **Full name:** Wilma Pastrana	**Expire:** 05/2016

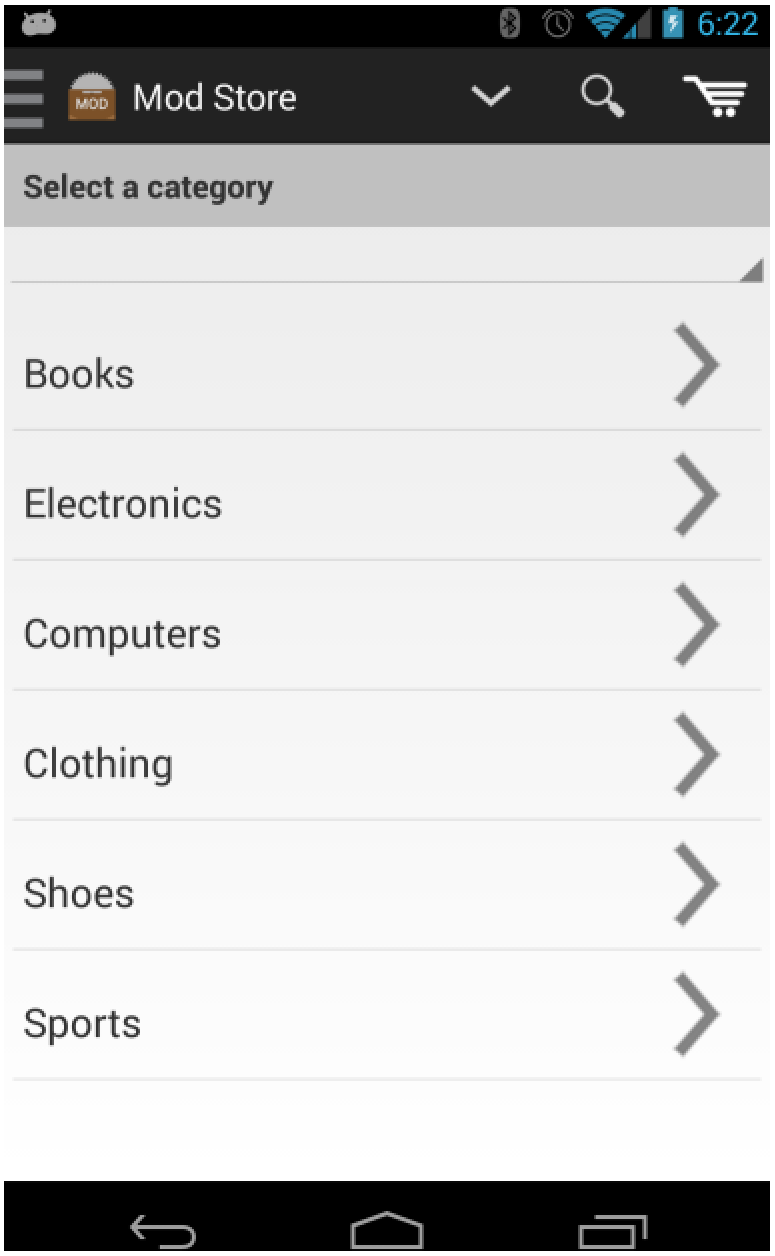
User credit card view. More credit cards can be added using the plus button at the top right corner, that will open a Dialog asking for the information. Also, clicking on the list will show a dialog that allows the user to edit the clicked Credit Card.



Urb. Villas Jose
Calle Mango C-20
00680 Mayaguez
Puerto Rico

This is the dialog that is used to create or update a Credit Card. The addresses of the user are shown because that is important in order to verify that the Credit Card is real.

2. Browse Categories



Select a category

Books

Electronics

Computers

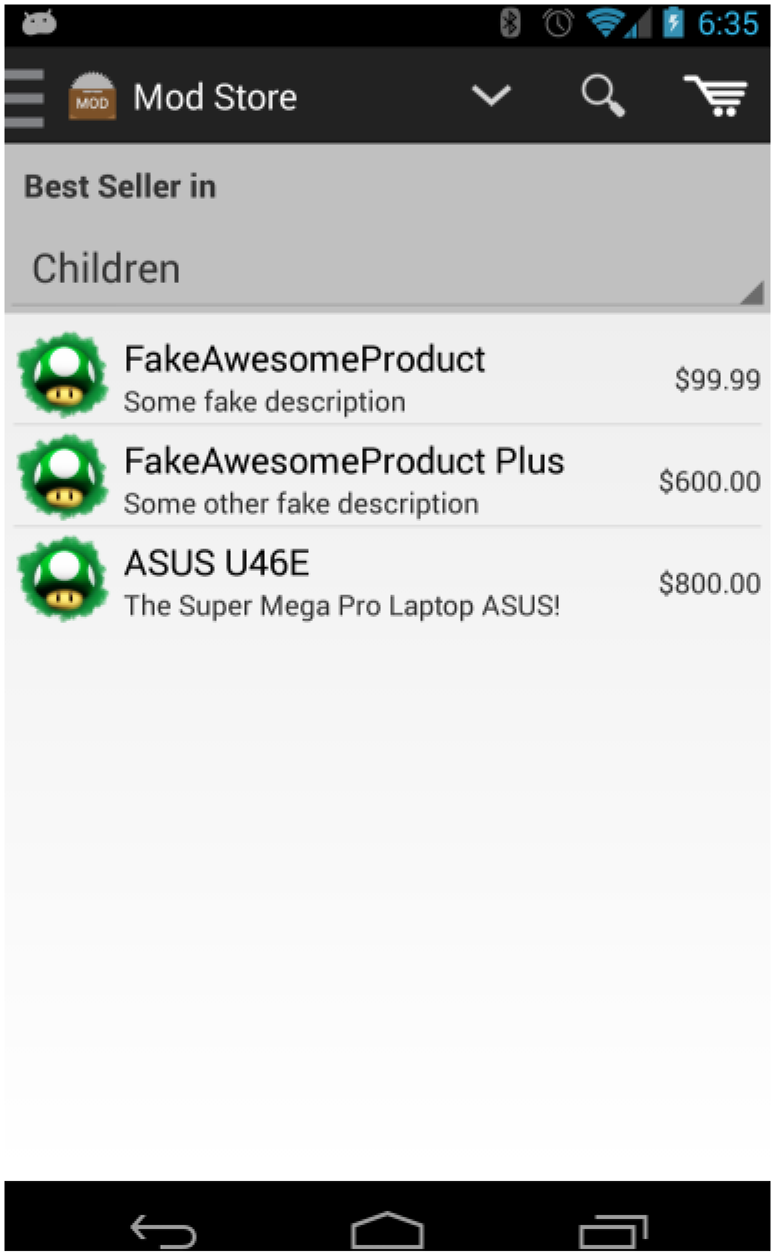
Clothing

Shoes

Sports

Top list of categories, as the user click each category the list is populated with subcategories until the list of products is shown.

3. Browse Products



Best Seller in

Children

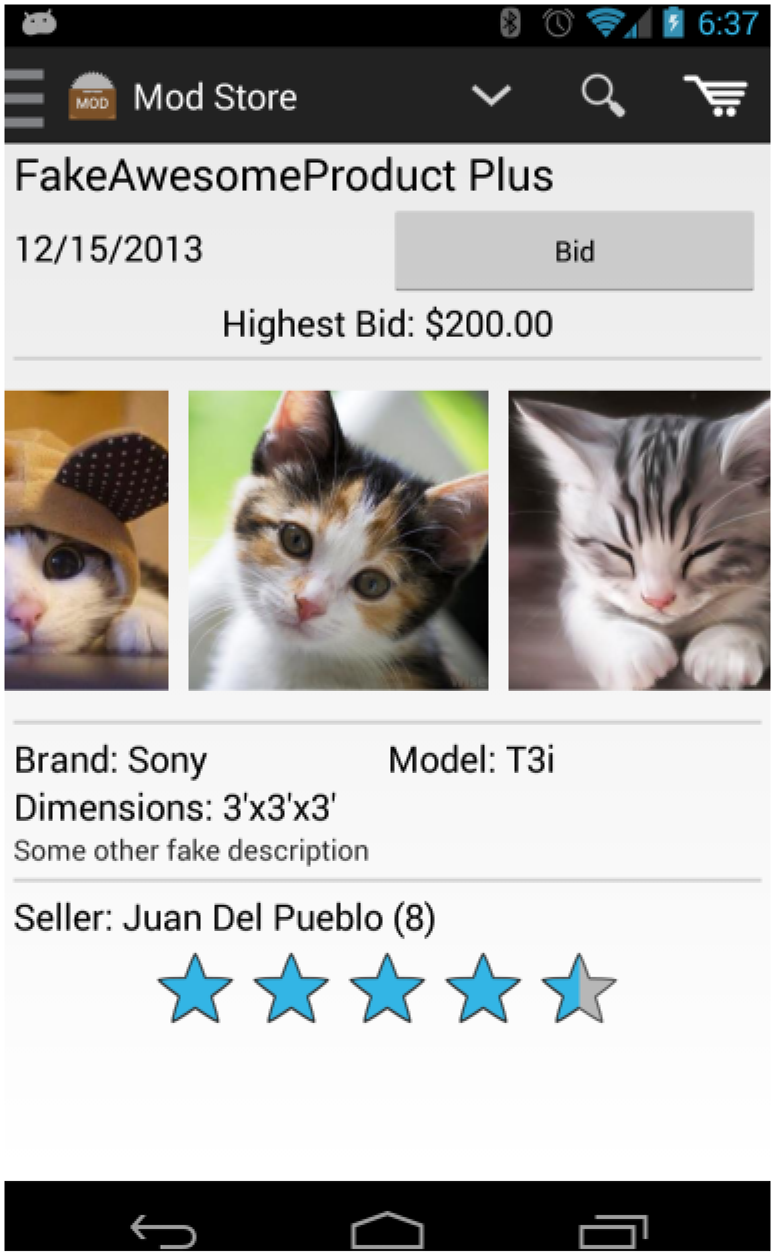
FakeAwesomeProduct
Some fake description
$99.99

FakeAwesomeProduct Plus
Some other fake description
$600.00

ASUS U46E
The Super Mega Pro Laptop ASUS!
$800.00

List of products

4. View Details of Product



FakeAwesomeProduct Plus

12/15/2013

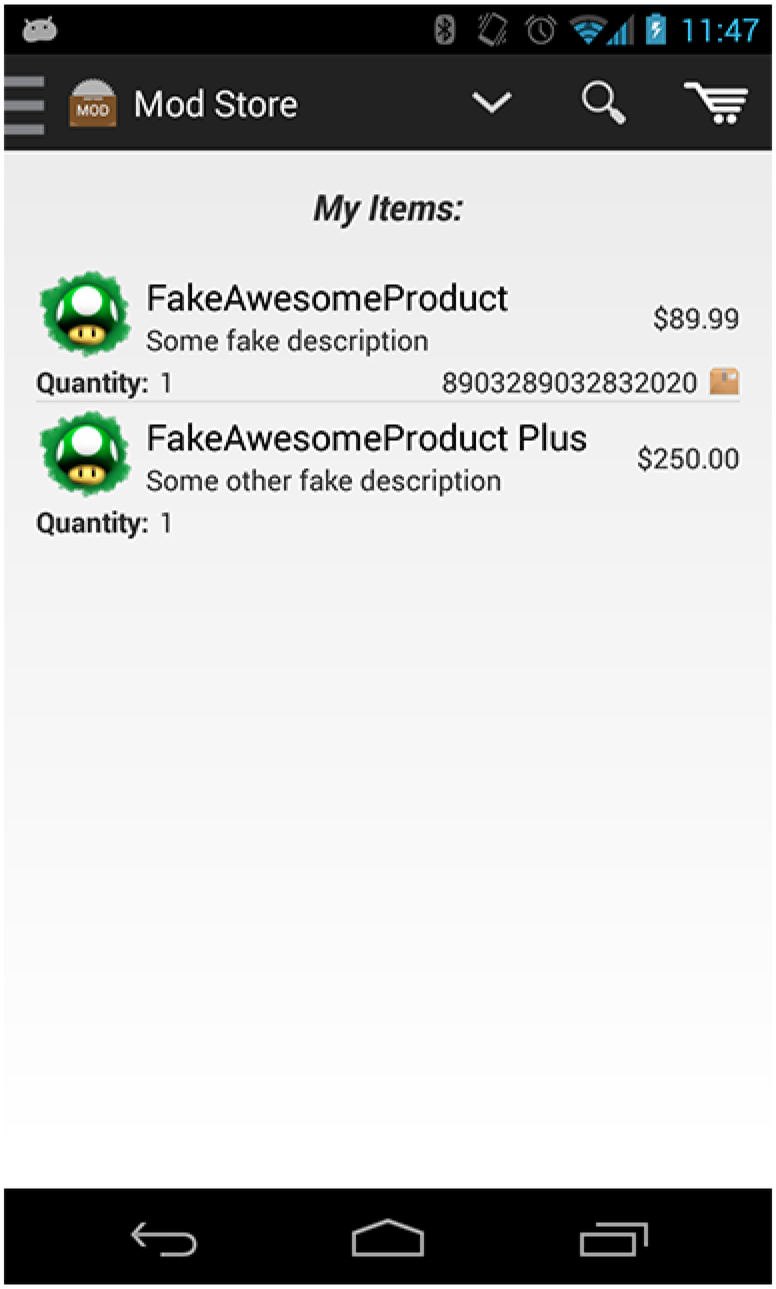
Highest Bid: $200.00

Brand: Sony
Model: T3i
Dimensions: 3'x3'x3'
Some other fake description

Seller: Juan Del Pueblo (8)

Detailed information of product.

5. List of purchased/won items



My Items:

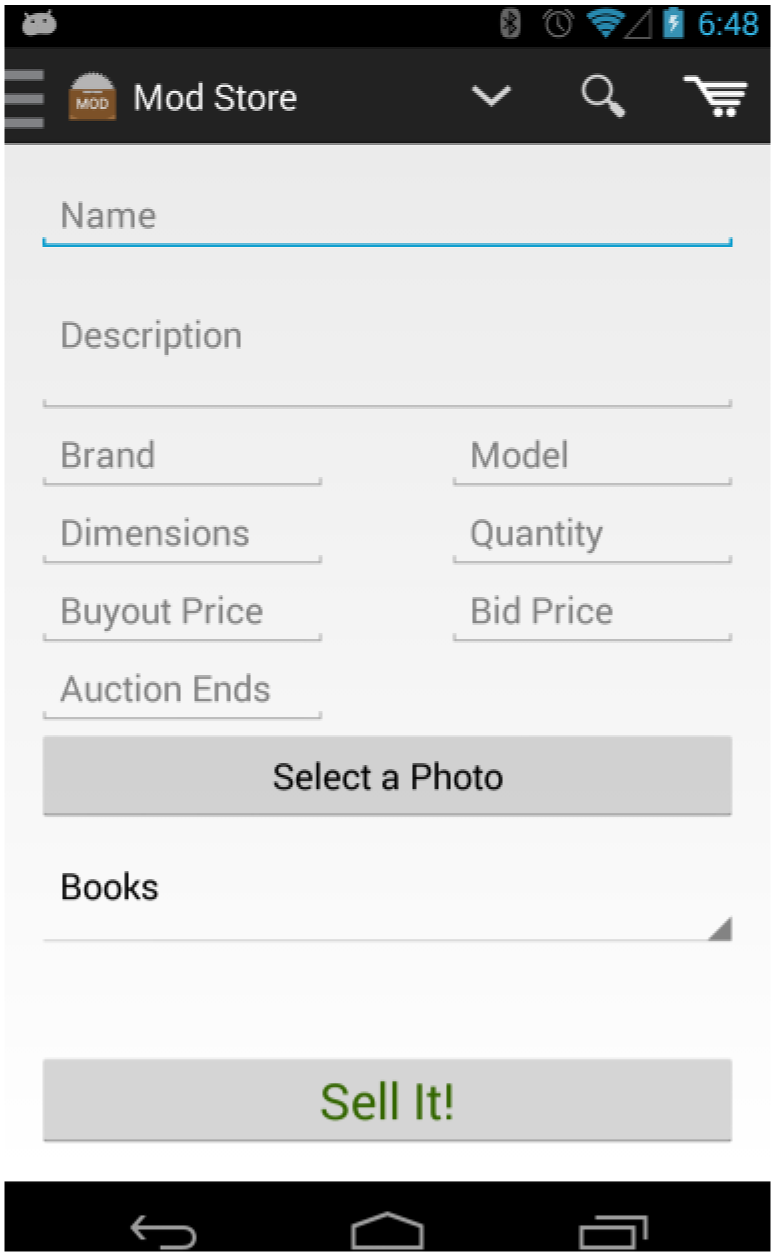
FakeAwesomeProduct
Some fake description
$89.99
Quantity: 1
8903289032832020

FakeAwesomeProduct Plus
Some other fake description
$250.00
Quantity: 1

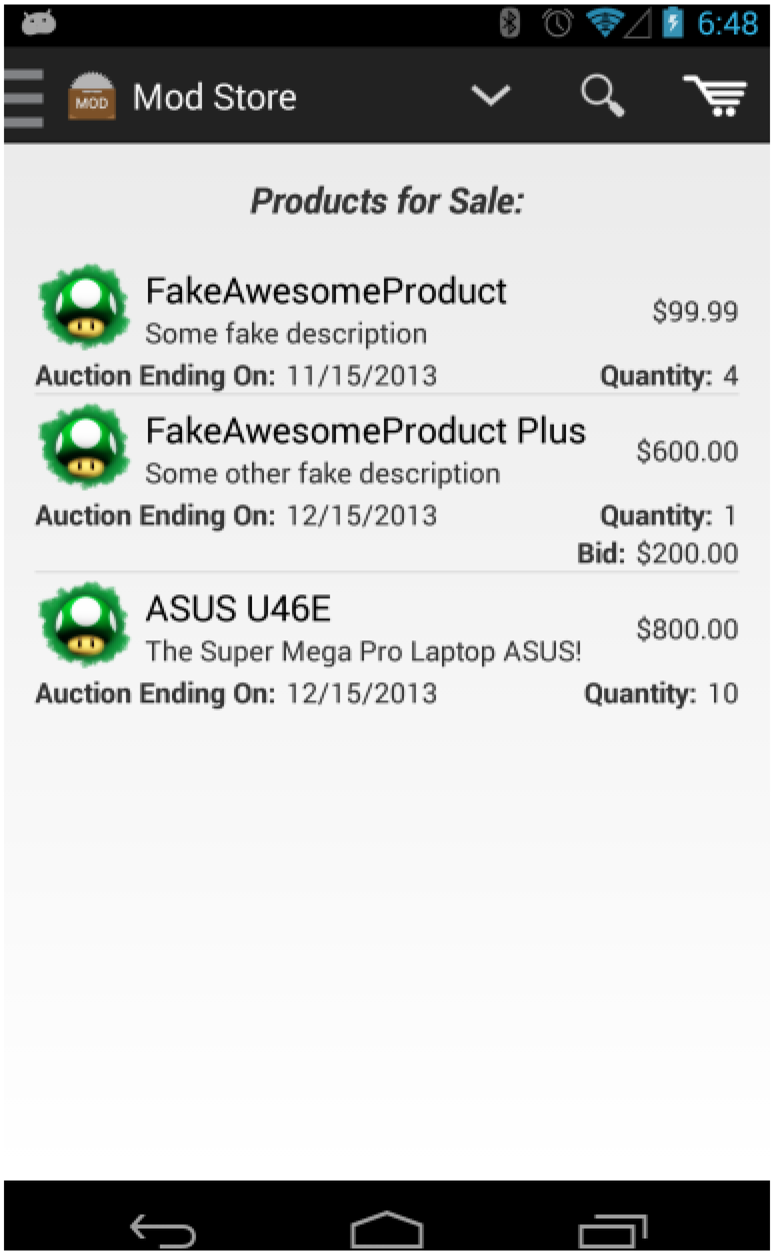
Users list of purchased and won items

6. List of User Selling Items

a. Sell an item



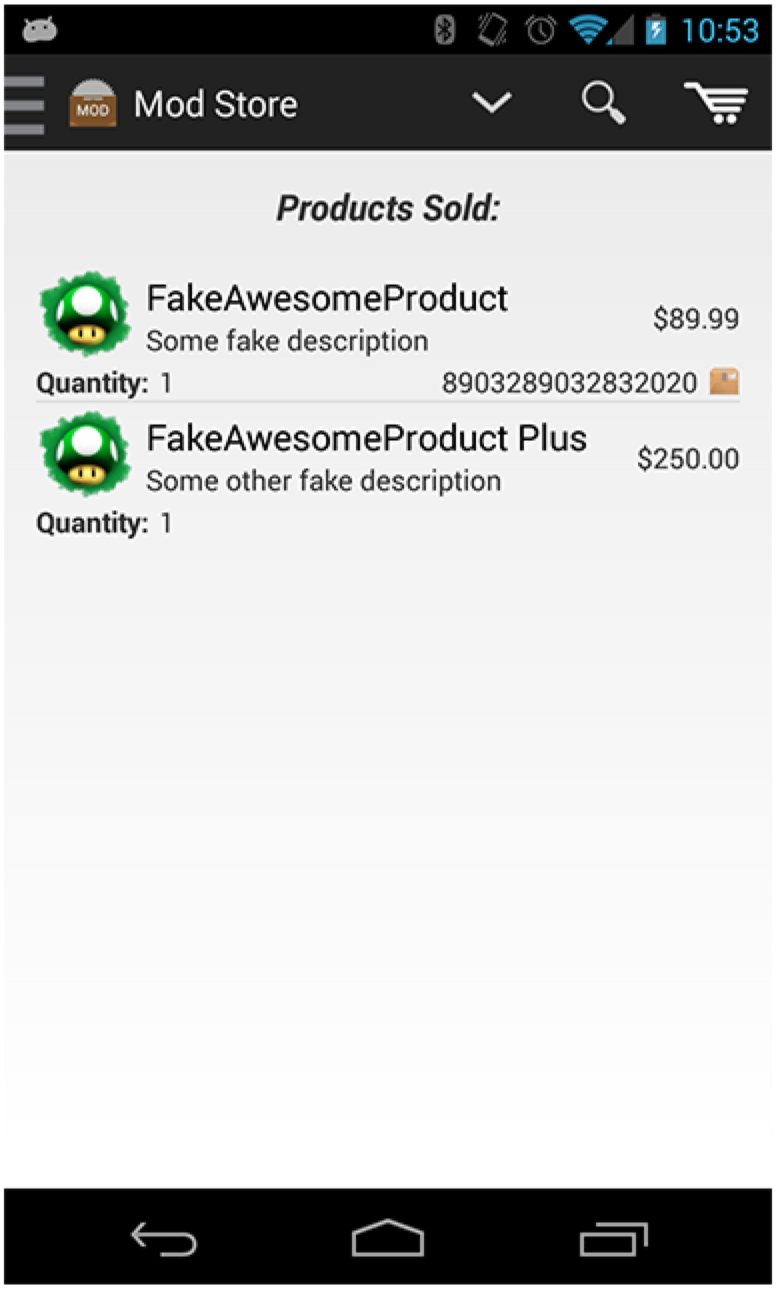
Add item for sell form view. In this form the user will add new Products to sell. The category combo box has all the subcategories under his main category, allowing the user to choose specifically where his item is going to be classified. Also when the user selects a photo it will ask for various ways supported by the Android OS to upload a picture.



Products for Sale:

FakeAwesomeProduct	Some fake description	$99.99
Auction Ending On: 11/15/2013		Quantity: 4
FakeAwesomeProduct Plus	Some other fake description	$600.00
Auction Ending On: 12/15/2013		Quantity: 1
		Bid: $200.00
ASUS U46E	The Super Mega Pro Laptop ASUS!	$800.00
Auction Ending On: 12/15/2013		Quantity: 10

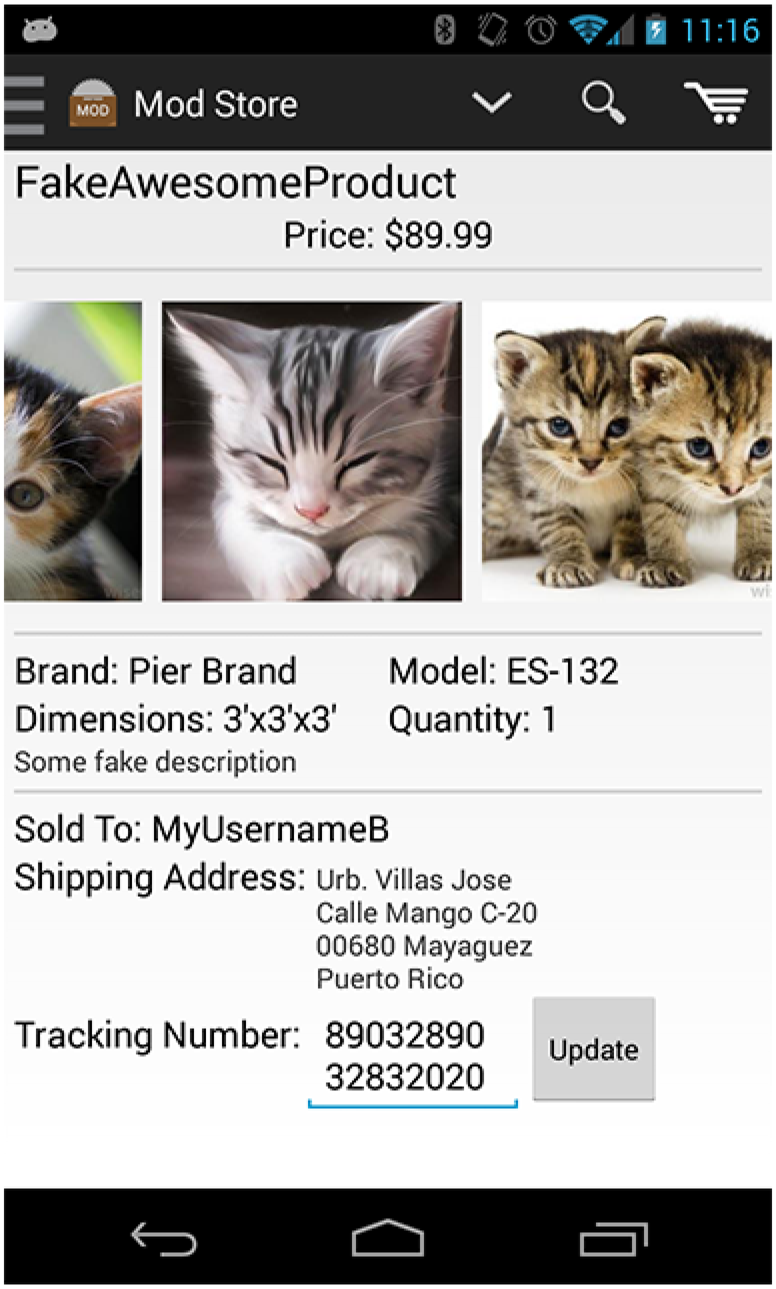
User list of products being sold.



Products Sold:

FakeAwesomeProduct Some fake description		$89.99
Quantity: 1	8903289032832020	
FakeAwesomeProduct Plus Some other fake description		$250.00
Quantity: 1		

List of products that have been successfully sold. In this list the user can see If he has set a tracking number for each product.



FakeAwesomeProduct
Price: $89.99

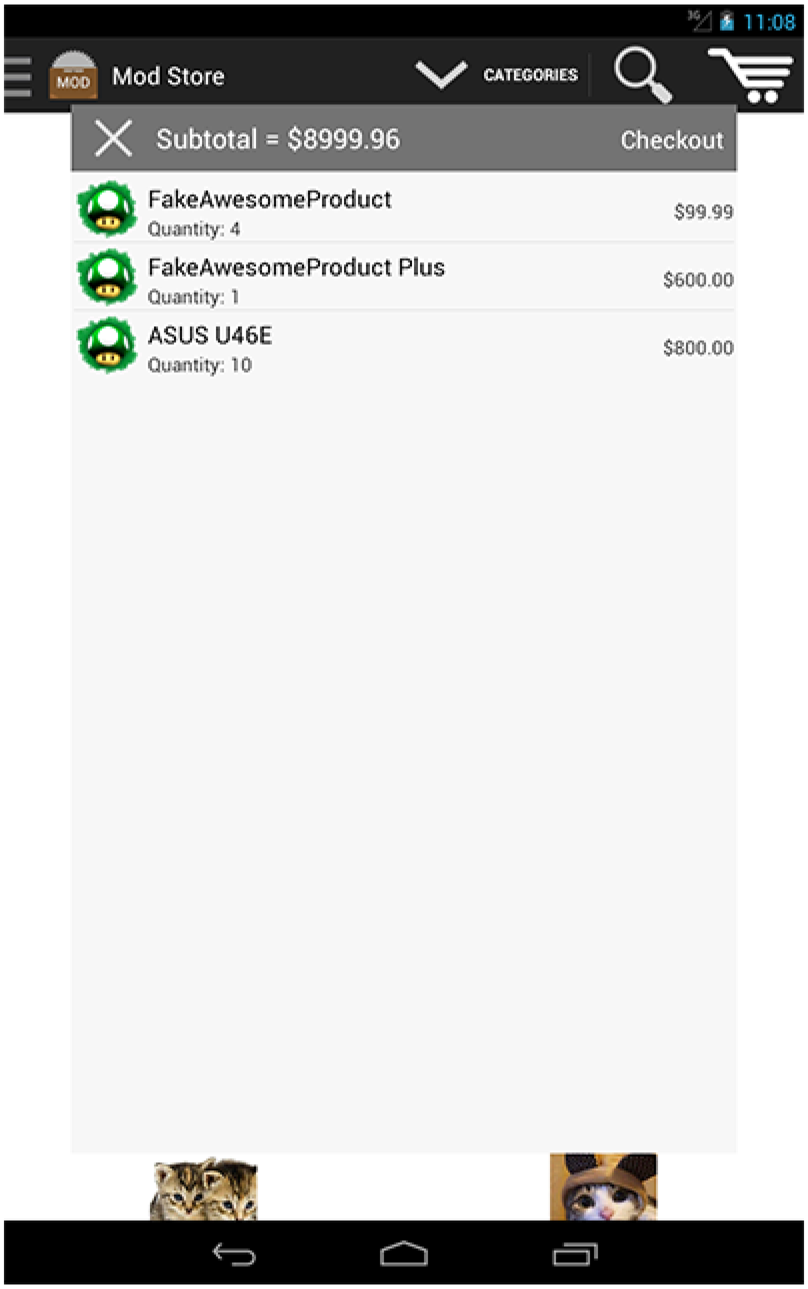
Brand: Pier Brand
Model: ES-132
Dimensions: 3'x3'x3'
Quantity: 1
Some fake description

Sold To: MyUsernameB
Shipping Address: Urb. Villas Jose
Calle Mango C-20
00680 Mayaguez
Puerto Rico

Tracking Number: 89032890 32832020

Detailed view of a sold item. The user can see the buyer and update the tracking information.

7. Cart



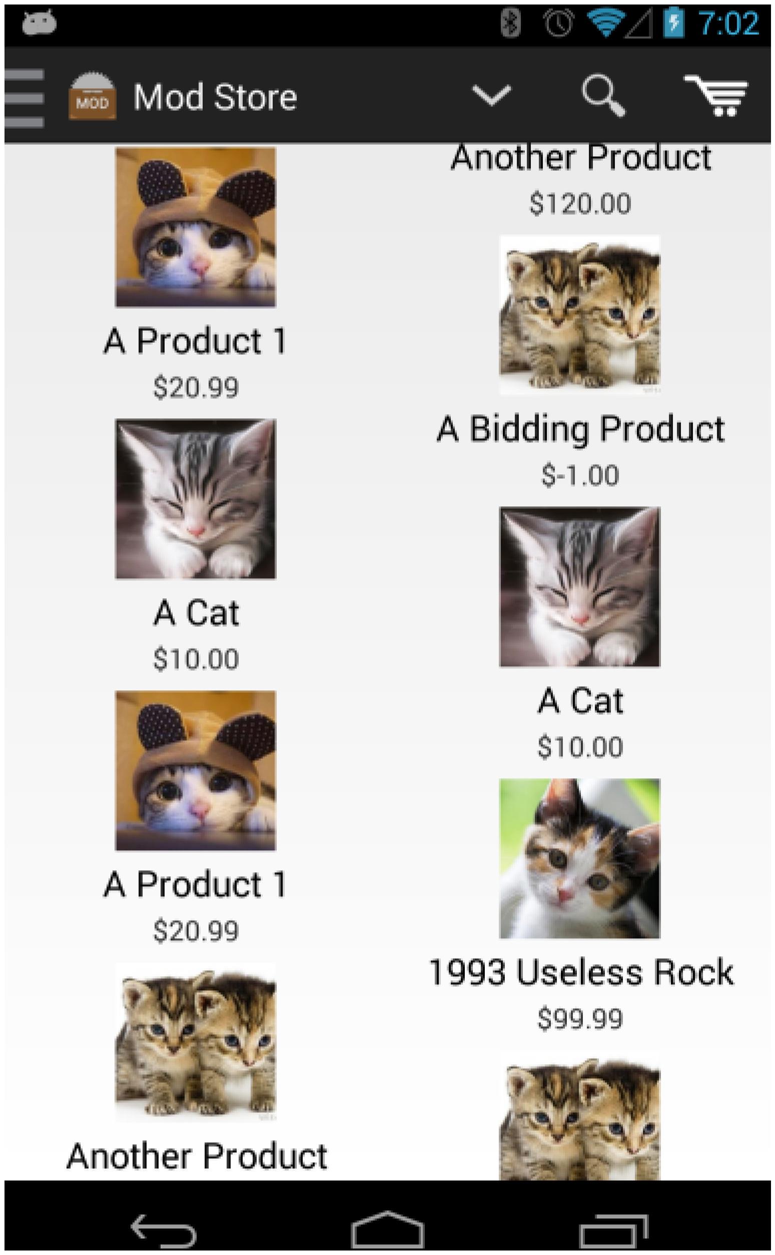
Subtotal = $8999.96 Checkout

FakeAwesomeProduct Quantity: 4	$99.99
FakeAwesomeProduct Plus Quantity: 1	$600.00
ASUS U46E Quantity: 10	$800.00

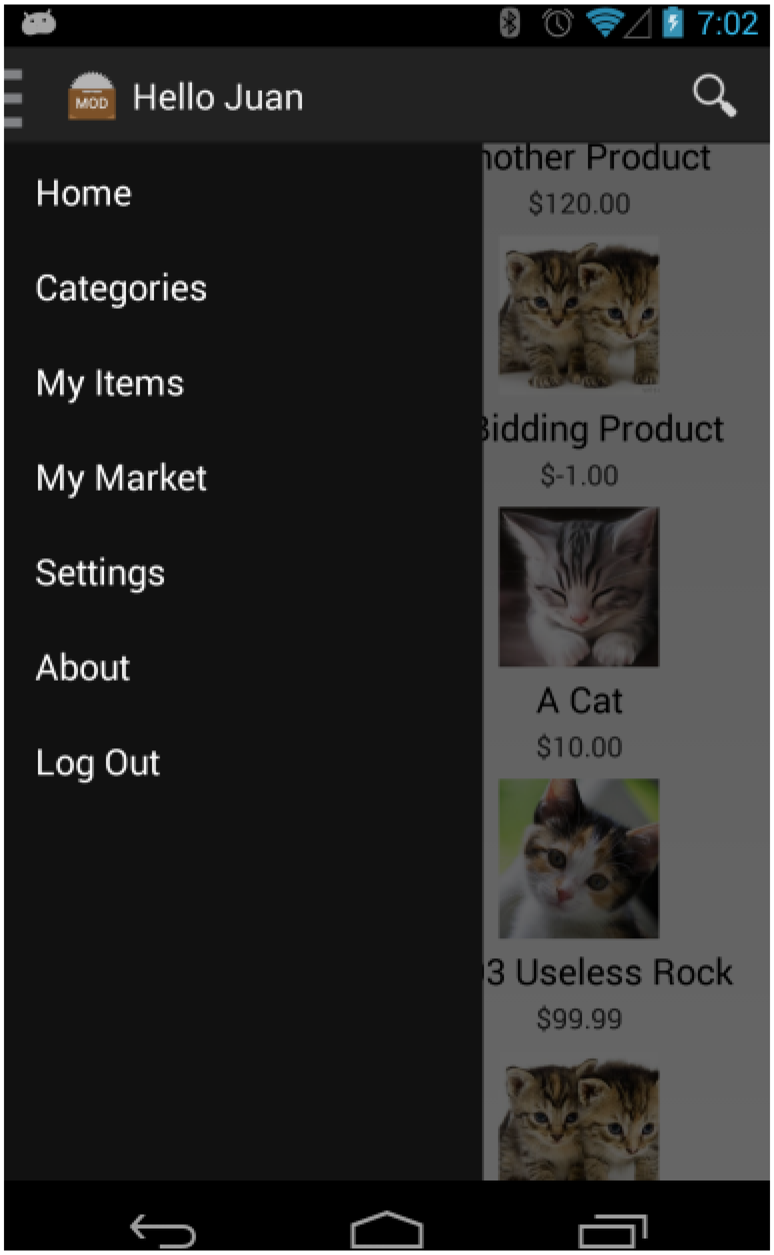
Cart view, user can see subtotal of cart and can proceed to checkout.

8. Main View

a. Drawer



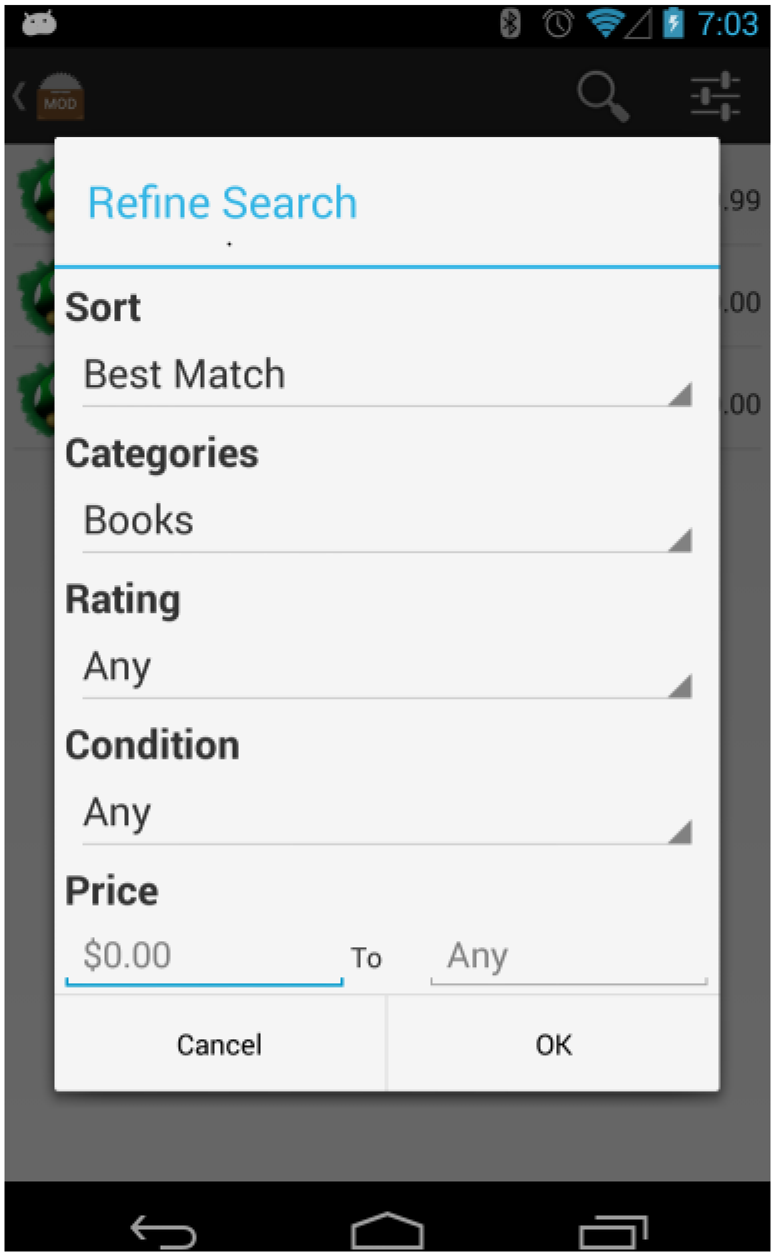
Main view, showcasing featured products.



Drawer, a menu to various parts of the app. The drawer menu is available on almost all the views in the app, to open it, either click the top left logo or swipe from left to right to drag it out.

9. Search

a. Filters



Refine Search

Sort: Best Match

Categories: Books

Rating: Any

Condition: Any

Price: $0.00 To Any

Search view filters