DESIGN DOCUMENT

Jakub Rybicki

Contents

Requirements

This project will feature an ‘shopping list’ app, a database, and a web service. The application’s main use will be to determine how many items you have at home, and alert you when you are running low on some things. This application will be distributed through the Google Play Store, and will allow anyone to download it. Once downloaded, the application will require you to make a new ‘household’, join an existing household, or login through your username and password. The creator of the household will be an admin, allowing them to see their invite token, make new categories, and place items into those categories.

This app will sync automatically when any changes have been made, and will automatically update on every device that is linked with your household. Items in your ‘list’ will automatically be sorted by their quantity (lowest at top), and the app will attempt to learn what you use most often, and show you most used items at the top of the screen, regardless of their stock.

Must have

- New user is able to create a ‘household’, making him an admin for that household.
- Admin is able to invite others via a ‘token’.
- Users can add new items to the item list.
- Admin can add and modify categories.
- Users can change quantity of items in their house only.
- Hashing passwords

Should have

- Admin resets password for user – SMS to user or email if available.
- Items are sorted based on quantity, set on an individual item basis.
- Users can ‘login’ if they already have an account and are a part of a family.

Could have

- Ability to pick icons for categories.
- Self-learning algorithm for brining commonly used items to the top of the items view.
- Slide left on an item to decrease quantity, and slide right to increase quantity

Won’t have

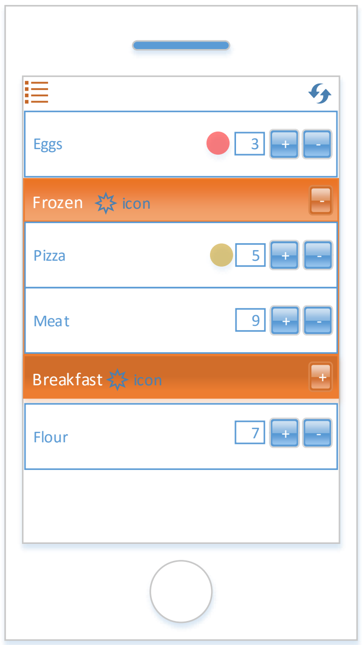
- Ability for admin to upload own icons for categories

Web service prototype

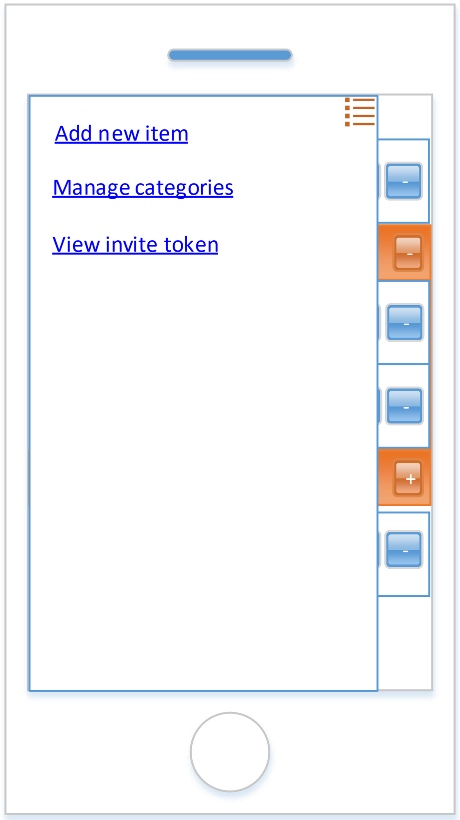
Job #	Web Service: rootURL +	Method	Description	Passes data	returns
1	/delete/item/:id	DELETE	deletes a specific item	item_ID	-
2	/delete/category/:id	DELETE	deletes a specific category	category_ID	-
3	/delete/house/:id	DELETE	Deletes a specific house	house_ID	-
4	/delete/house/:id/all	DELETE	deletes a house and all information relating to it	house_ID	-
5	/add/house	POST	add a new house	-	-
6	/add/category	POST	add a new category	-	-
7	/add/item	POST	add a new item	-	-
8	/add/user	POST	add a new user	-	-
9	/get/item/:id	GET	get a speicfic item	item_ID	JSON - one item
10	/get/items	GET	get all items	-	JSON - all items
11	/get/item/:id/category	GET	get the category of an item	item_ID	String
12	/get/item/:id/qty	GET	get qty of an item	item_ID	integer - items qty
13	/get/item/:id/name	GET	get the item's name	item_ID	String - items's name
14	/get/house	GET	get all houses	-	JSON - all houses
15	/get/house/:id	GET	get a specific house	house_ID	JSON - one house
16	/get/house/:id/categories	GET	gets all categories from a house	house_ID	JSON - all categories that belor
17	/get/house/:id/users	GET	get all users from a house	house_ID	JSON - all users that belong to
18	/get/house/:id/items	GET	get all items from a house	house_ID	JSON - all items that belong to
19	/get/house/:id/needed	GET	get all 'needed' items from a specific house (qty<warning_qty)	house_ID	JSON - all items that are neede
20	/get/user/:id	GET	get all information relating to a user	user_id	JSON - specific user informatio
21	/get/user/:id/house	GET	get the house ID that the user belongs to	user_id	String - house ID
22	/get/category/:id	GET	get all information relating to a category	category_ID	JSON - specific category inform
23	/get/category/:id/name	GET	get the specific categories name	category_ID	String - category name
24	/set/house/:id	PUT	update a house	house_ID	-
25	/set/house/:id/tokenID	PUT	update a specific houses token	house_ID	-
26	/set/item/:id	PUT	update an item	item_ID	-
27	/set/item/:id/name	PUT	update an items name	item_ID	-
28	/set/item/:id/qty	PUT	update an items quantity	item_ID	-
29	/set/item/:id/warningqty	PUT	update an items warning Qty	item_ID	-
30	/set/item/:id/priority	PUT	update an items priority	item_ID	-
31	/set/category/:id	PUT	Update a category	category_ID	-
32	/set/category/:id/name	PUT	update a categories name	category_ID	-
33	/set/user/:id/username	PUT	update a users username	user_ID	-
34	/set/user/:id/password	PUT	update a users password	user_ID	-
35	/set/user/:id/email	PUT	update a users email	user_ID	-
36	/set/user/:id/phone	PUT	update a users phone	user_ID	-

Design diagrams

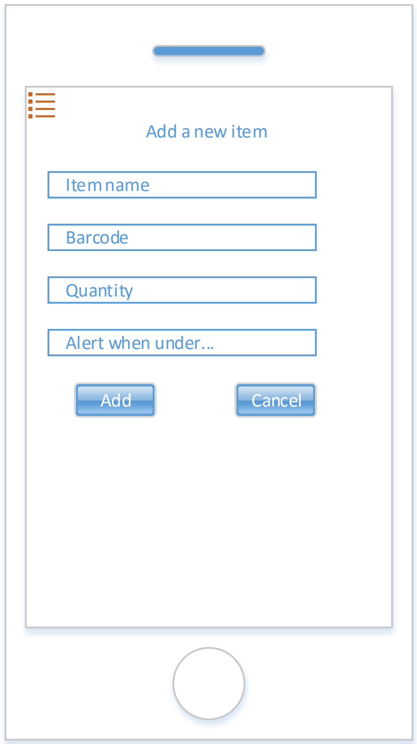
Main starting screen:



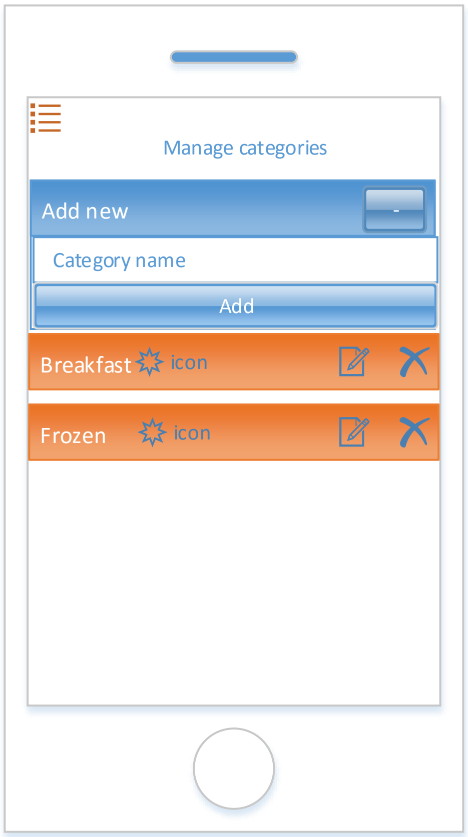
Menu/tools screen



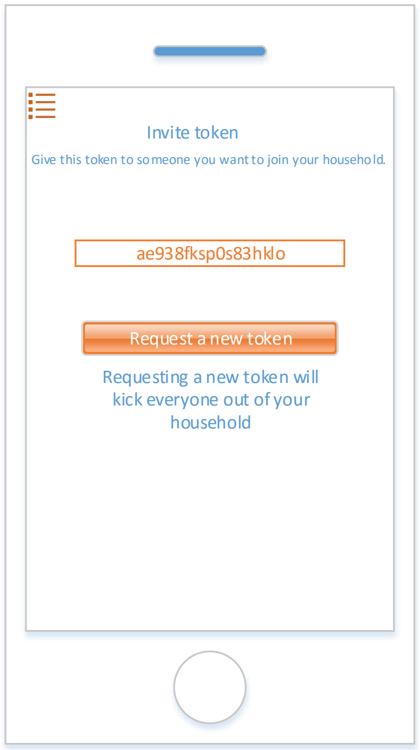
Adding a new item:



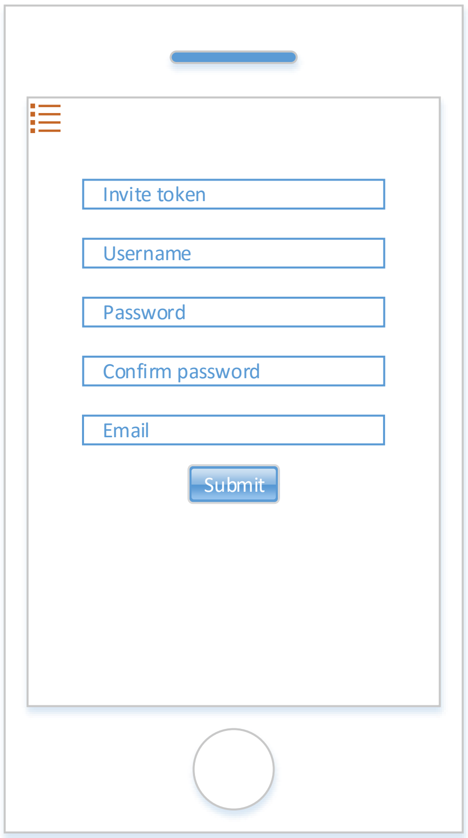
Managing categories:



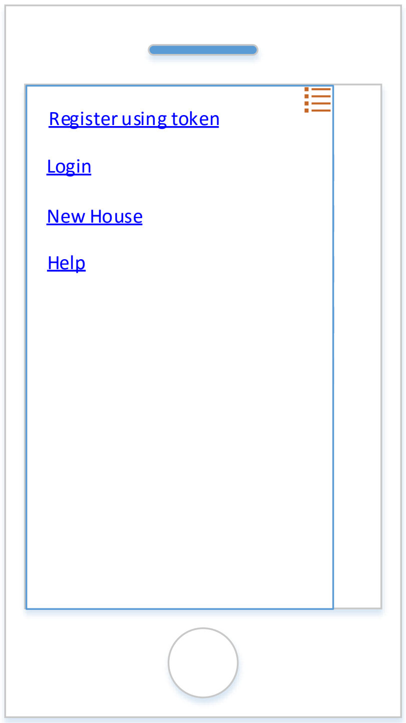
Viewing/requesting a new invite token:


Invite token
Give this token to someone you want to join your household.
ae938fksp0s83hklo
Request a new token
Requesting a new token will kick everyone out of your household

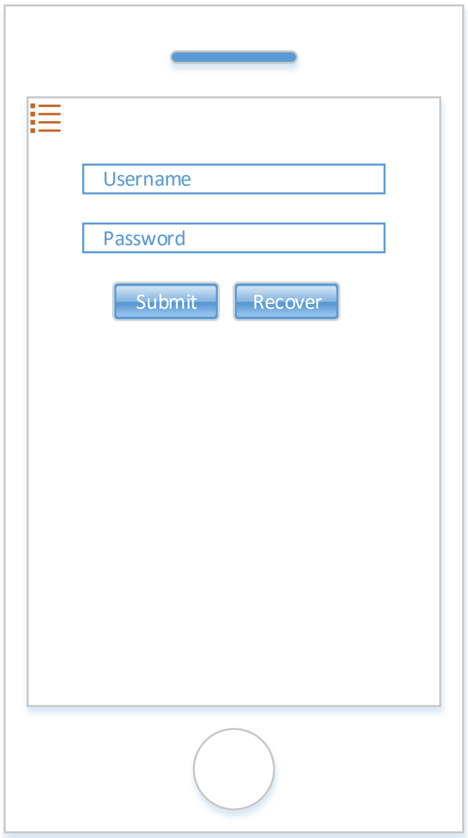
New user registration page:



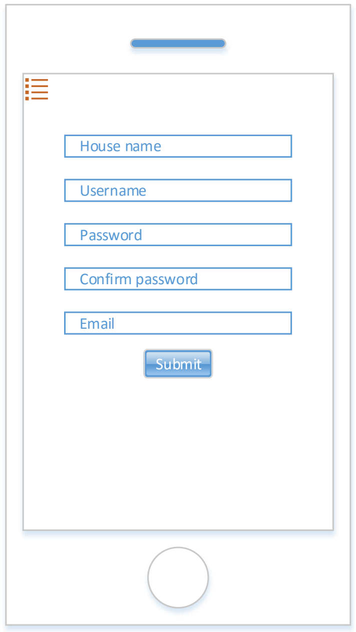
Settings menu for unregistered users



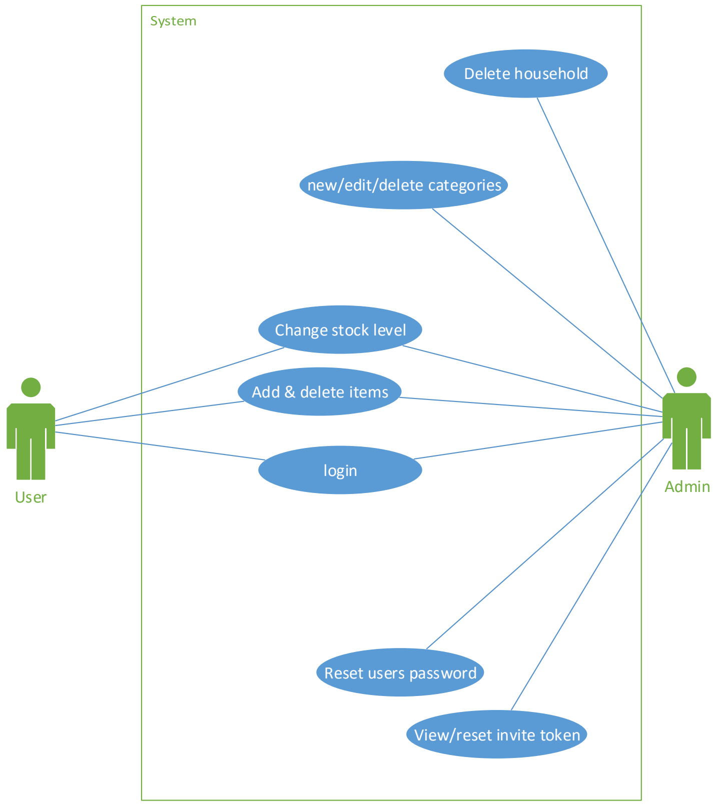
Login



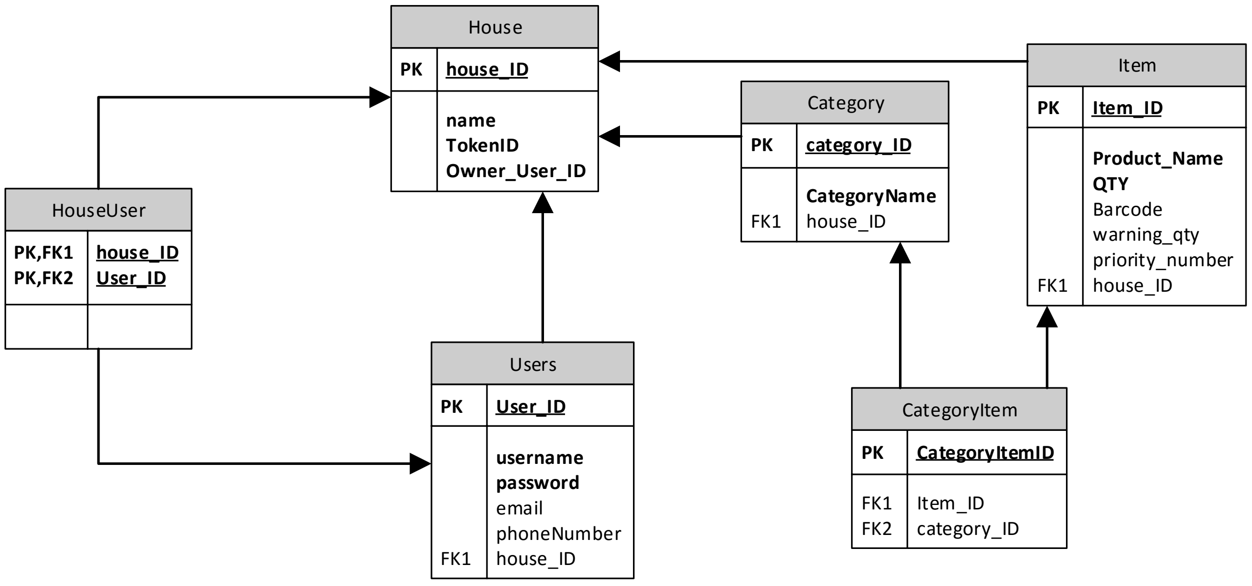
New house



Use case diagram



Database Design



Database script

Database Script.SQL

```
-- JAKUB RYBICKI
CREATE DATABASE JakubRybickiDB;
USE JakubRybickiDB;
CREATE TABLE House(
house_ID int  NOT NULL auto_increment
, name VARCHAR(30)  NOT NULL
, tokenID VARCHAR(10)  NOT NULL
, owner_User_ID CHAR(10)  NOT NULL
,PRIMARY KEY (house_ID)
);


CREATE TABLE Users (
user_ID int  NOT NULL  AUTO_INCREMENT
, username VARCHAR(25)  NOT NULL
, password VARCHAR(40)  NOT NULL
, email VARCHAR(30)
, phoneNumber INTEGER
, house_ID int
,PRIMARY KEY (user_ID)
,FOREIGN KEY (house_ID) REFERENCES House(house_ID)
);

CREATE TABLE HouseUser (
house_ID int  NOT NULL
, user_id int  NOT NULL
,FOREIGN KEY (house_ID) REFERENCES House(house_ID)
);


CREATE TABLE Category (
category_ID int  NOT NULL
, CategoryName VARCHAR(25)  NOT NULL
, house_ID int
,PRIMARY KEY (category_ID)
,FOREIGN KEY (house_ID) REFERENCES House(house_ID)
);

CREATE TABLE Item (
Item_ID int  NOT NULL AUTO_INCREMENT
, Product_Name VARCHAR(25)  NOT NULL
, QTY INTEGER  NOT NULL
, Barcode INTEGER
, warning_qty INTEGER
, priority_number int
, house_ID int not null
,PRIMARY KEY (item_ID)
,FOREIGN KEY (house_ID) REFERENCES House(house_ID)
);


CREATE TABLE CategoryItem (
CategoryItemID int   NOT NULL  AUTO_INCREMENT
, Item_ID int Not NULL
, category_ID int not NULL
,PRIMARY KEY (CategoryItemID)
,FOREIGN KEY (Item_ID) REFERENCES Item(Item_ID)
);
```