QuickShop
A faster way to shop

Kunal Malhotra

Sam Britt

Motivation

- User shopping behavior:

  - Adds items to an empty list
  - Categorizes them
  - Starts shopping !!!!

- Existing shopping list apps mostly categorize items into predefined categories, display available coupons, saving cards, etc.

- Not much work done in optimizing the user's path through the grocery store

- Categories close together in the list can be in different parts of the store which makes the user go back and forth between aisles of the store.

- **What is needed? - A grocery list which dynamically sorts itself by category based on the location of items in the store.**

Functionality of existing shopping list apps

- Help users make a grocery list categorized under existing categories
- Offer free-form item input or have a large database of items for the users to choose from.
- Add frequently purchased items to new lists and keep track of items which have been checked out.
- Store specific coupon recommendation.
- Sync lists with friends / spouse.

Technologies used

- **Java** - Programming language
- **Eclipse IDE** - Development engine
- **Android SDK** - Application development
- **SQLite Database** - To record store layout information

QuickShop

- Android app to sort a shopping list by category to make the user pick up items in one pass without returning to a section
- Store layout stored in a SQLite Database
- Dynamic sorting of categories
- Helps saving user's time and effort
- List becomes a virtual guide helping picking up items in the right order.
- Great help in an unfamiliar store

Database Design

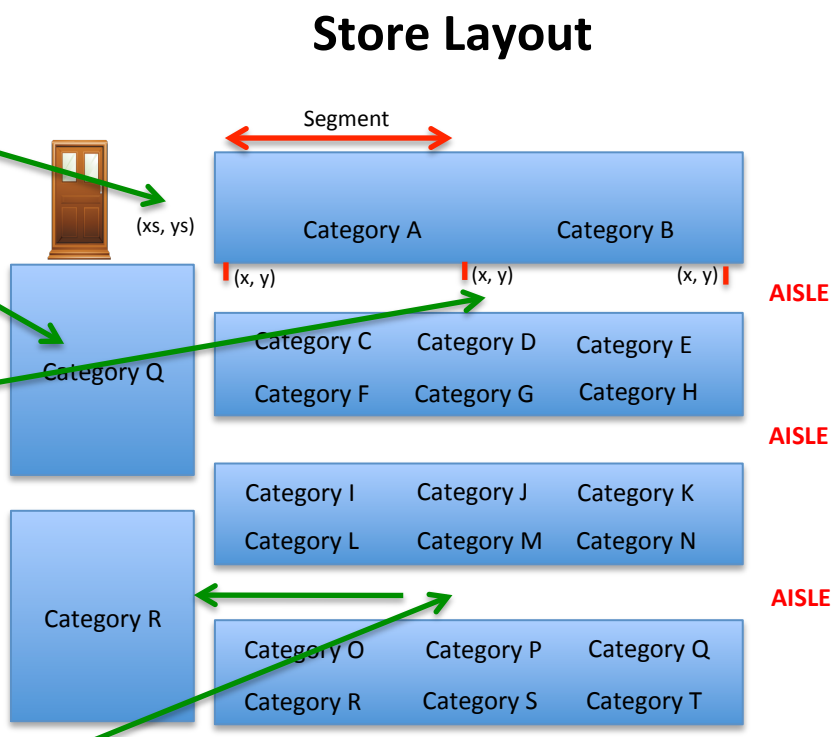
- Store names and location coordinates (Coordinates of the entrance door)

- System specified categories

- Coordinates of each category for every store in the database

- Each category spans one or more segments

- Items can be along the aisle as well as perpendicular to the aisle (Coordinates spanning multiple aisles)



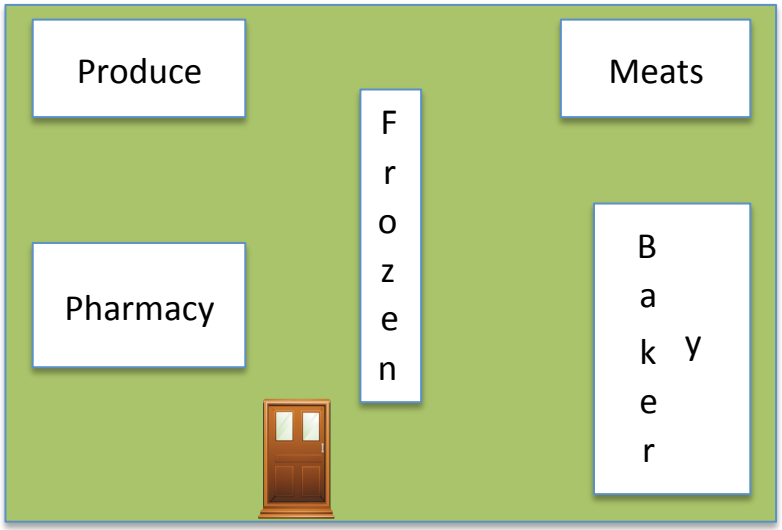
- The design is comprehensive since it plays a primary role in sorting the user's grocery list in a way that can optimize the time he takes to finish shopping.

- Topological component of the store was critical in designing the database

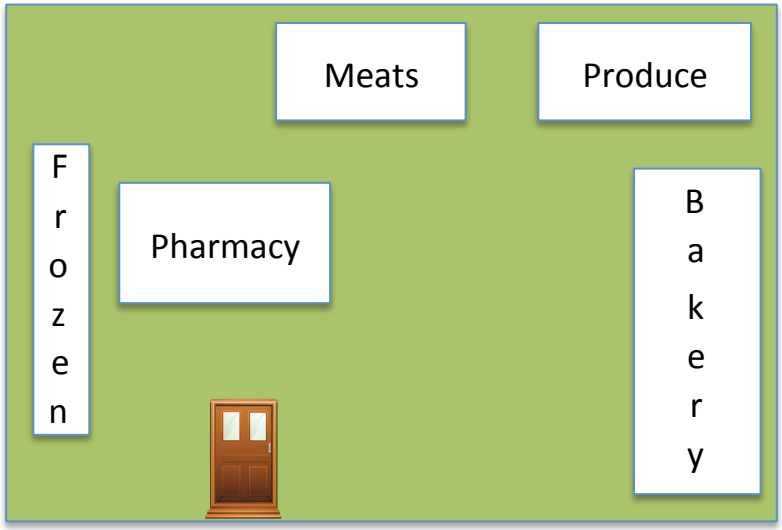
- Store floor plans of major chains are hard to get

- Manual collection of data - imported into our database

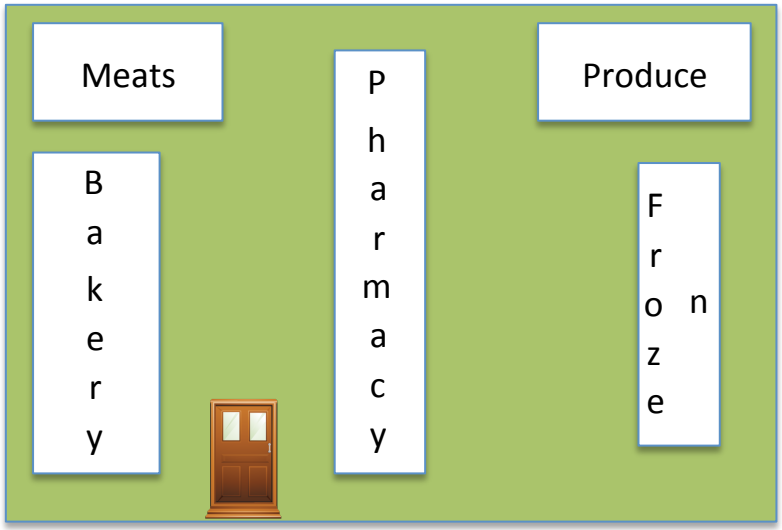
Various Publix Stores across Atlanta



Publix - Holcomb Bridge Road

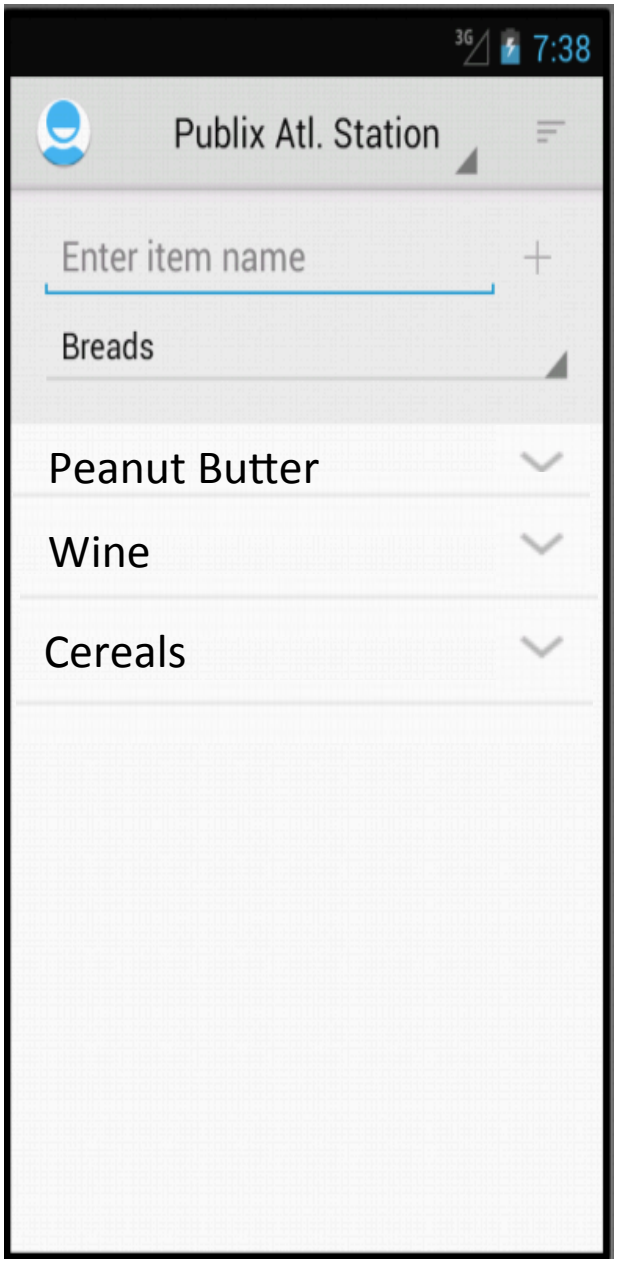


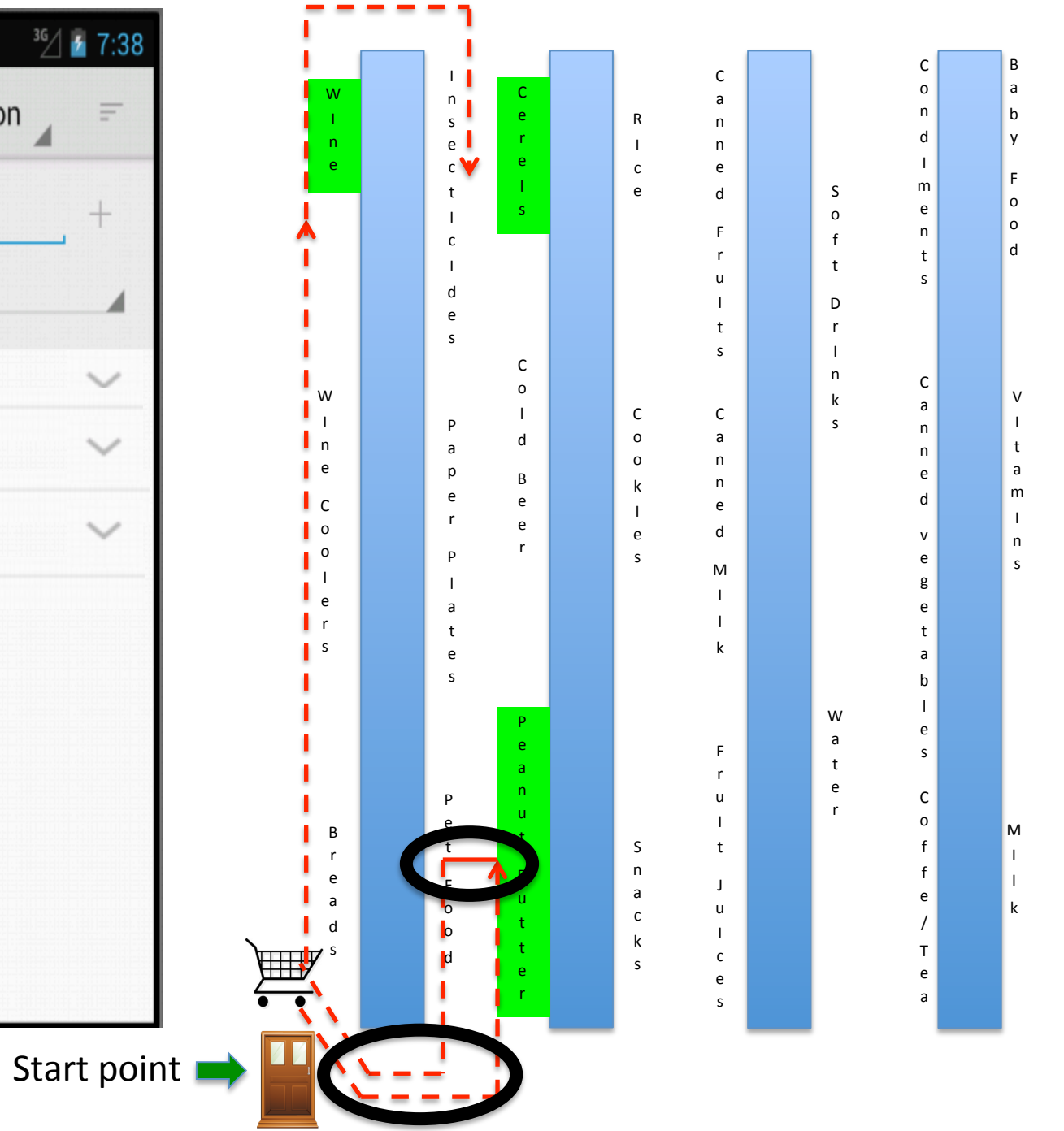
Publix - Atlantic Station



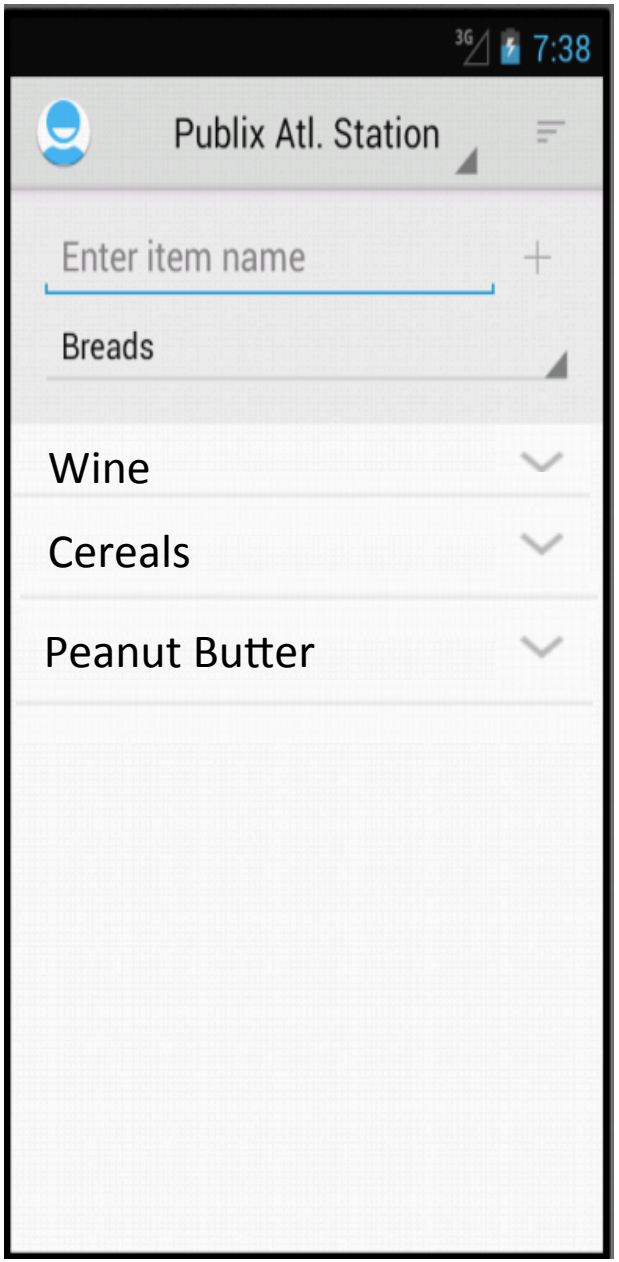
Publix - Shallowford Road

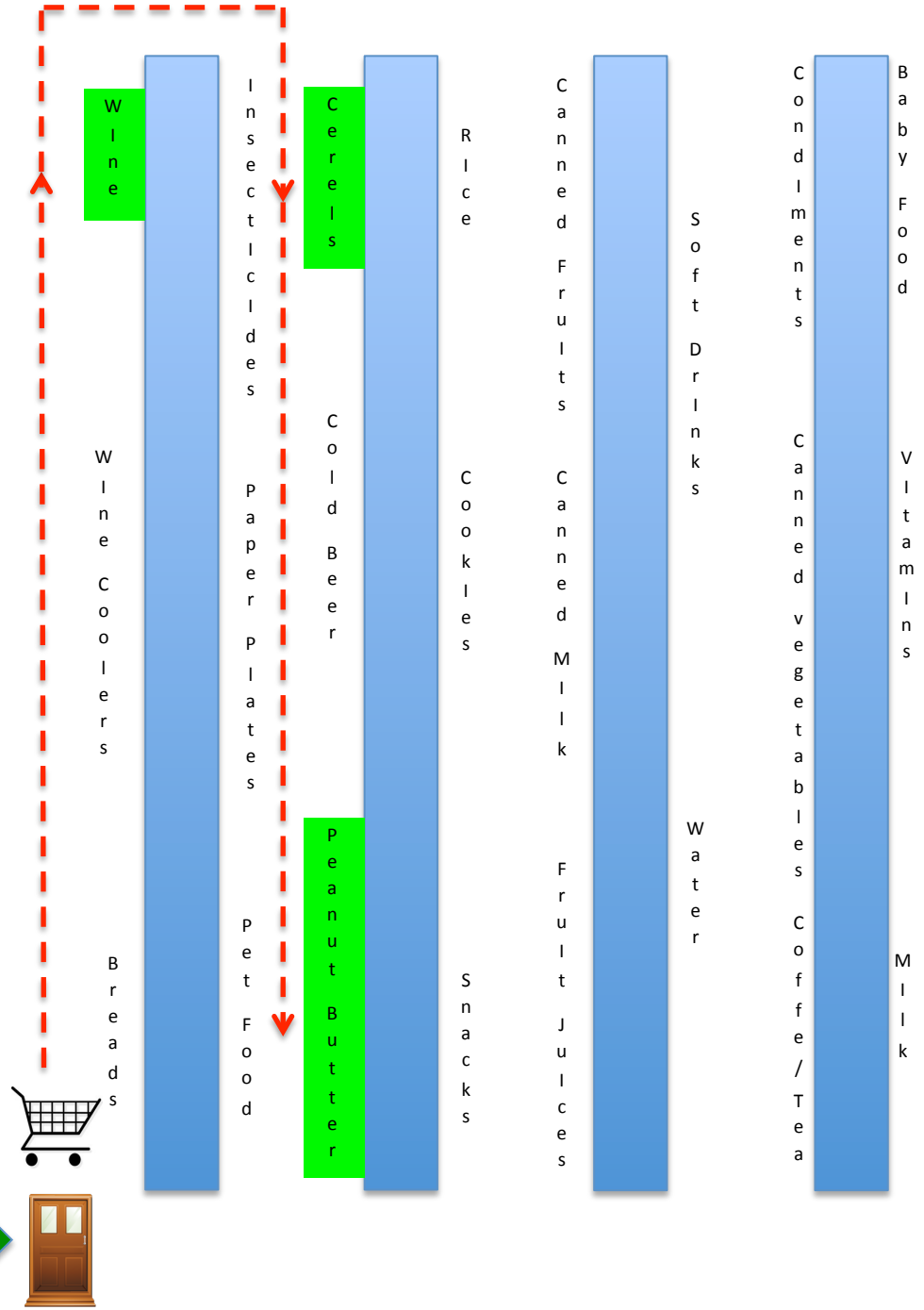
DEMO


7:38
Publix Atl. Station
Enter item name
Breads
Peanut Butter
Wine
Cereals

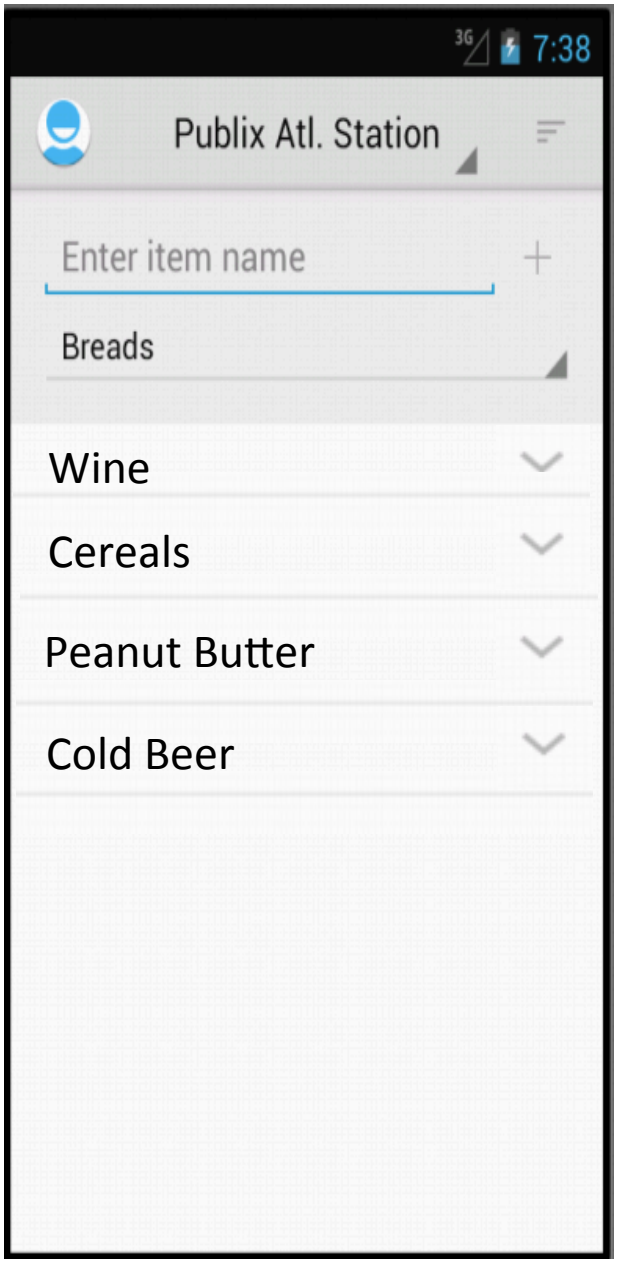

Wine
Insecticides
Cereals
Rice
Wine Coolers
Paper Plates
Cold Beer
Cookies
Breads
Pet Food
Peanut Butter
Snacks
Canned Fruits
Soft Drinks
Canned Milk
Water
Fruit Juices
Condiments
Baby Food
Canned vegetables
Vitamins
Coffee/Tea
Milk
Start point

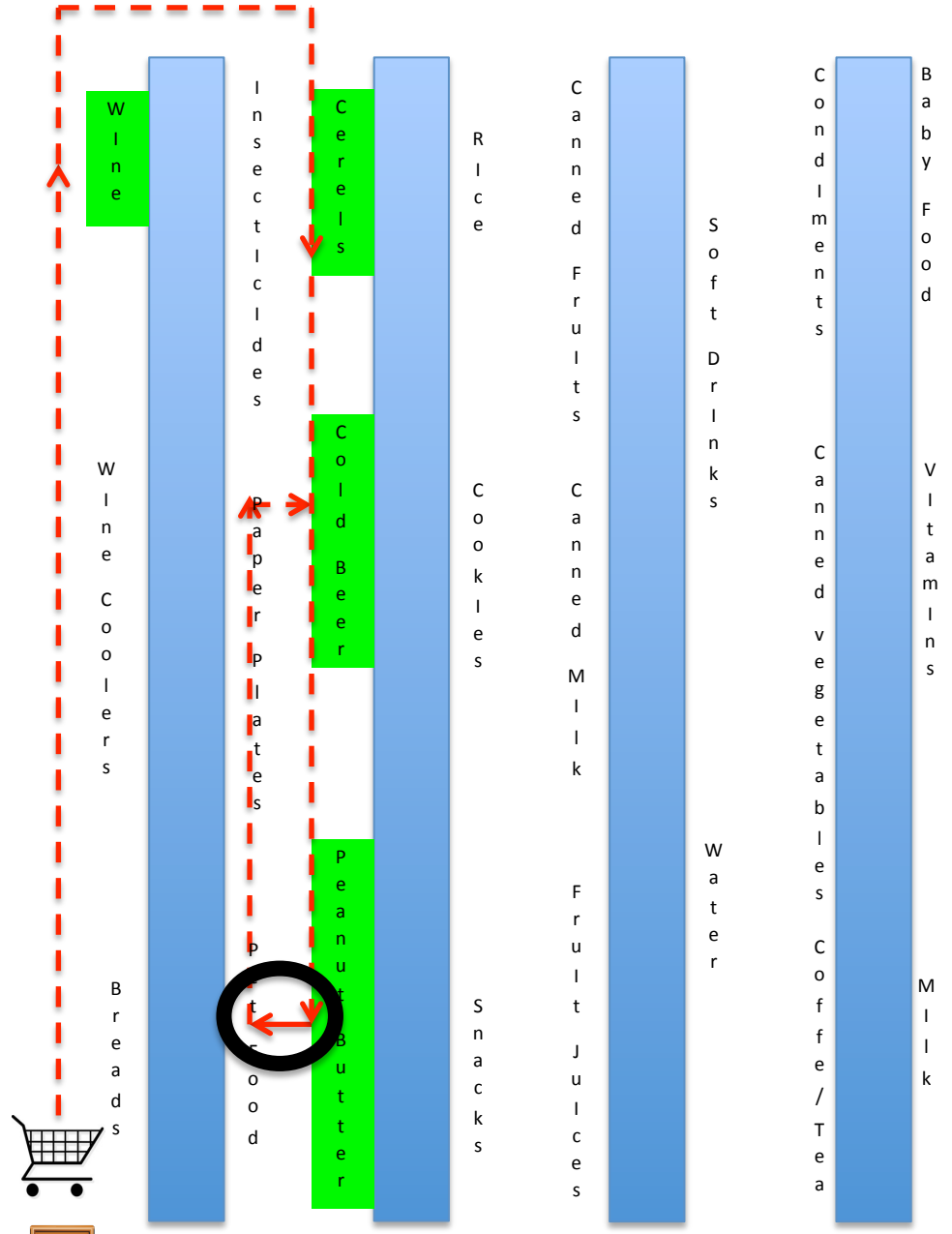
WALL


7:38
Publix Atl. Station
Enter item name
Breads
Wine
Cereals
Peanut Butter


Wine
Insecticides
Cereals
Rice
Wine Coolers
Paper Plates
Cold Beer
Cookies
Canned Fruits
Soft Drinks
Condiments
Baby Food
Breads
Pet Food
Peanut Butter
Snacks
Canned Milk
Water
Canned vegetables
Vitamins
Fruit Juices
Coffe/Tea
Milk
Start point

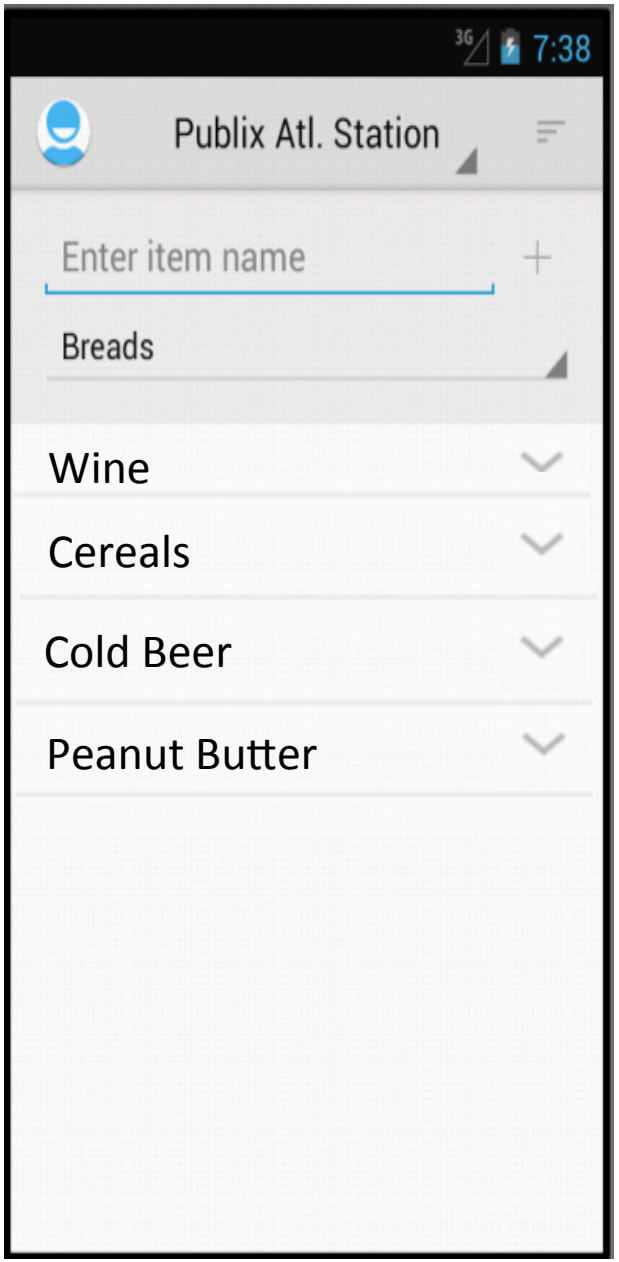
WALL

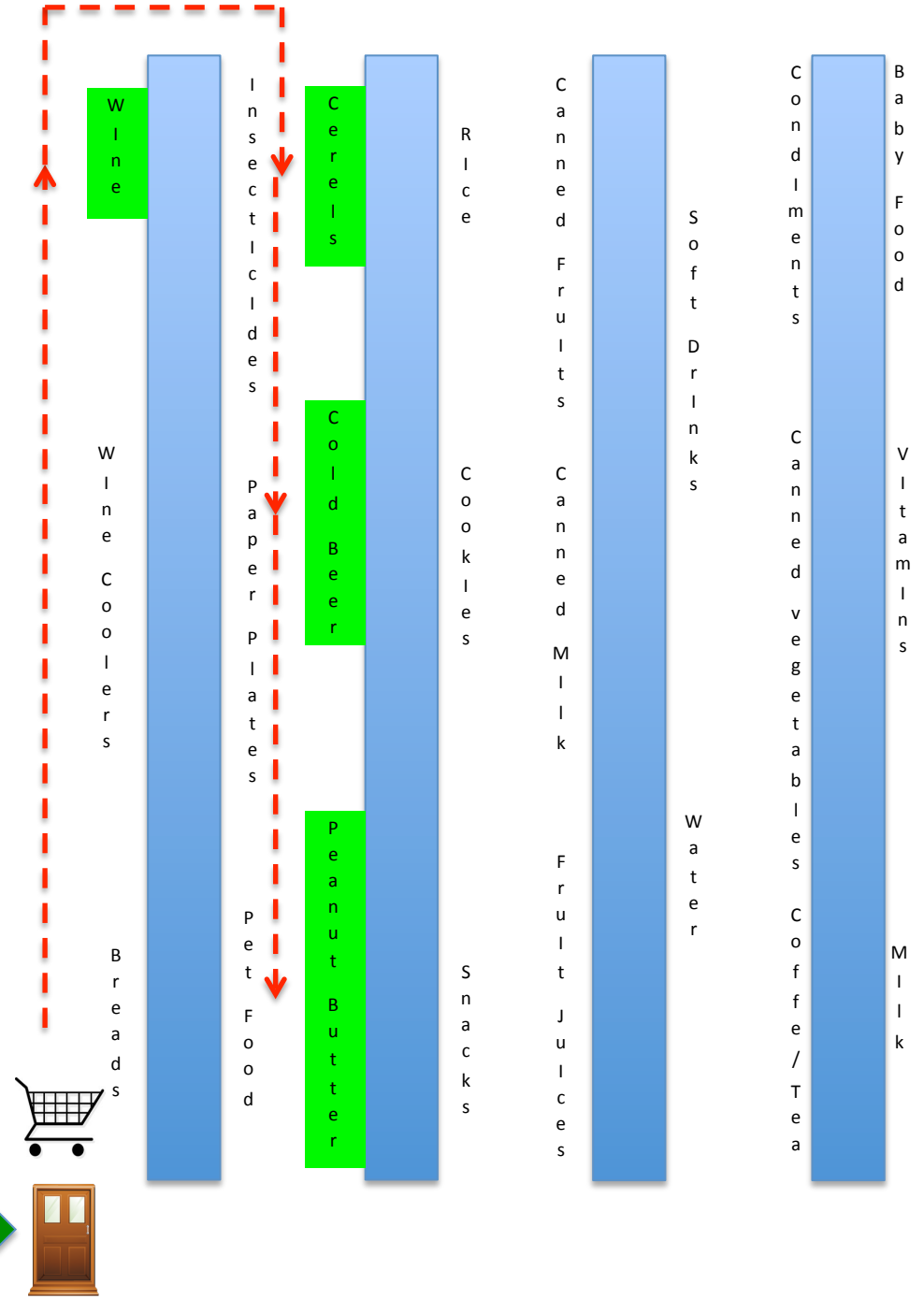




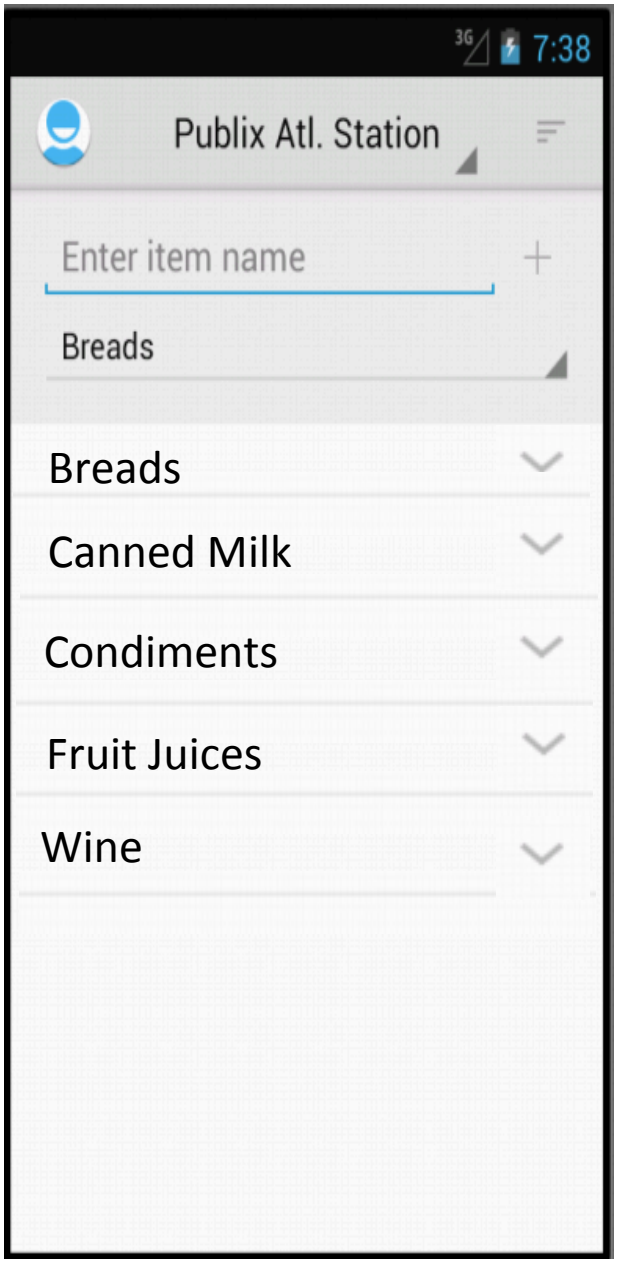
Start point

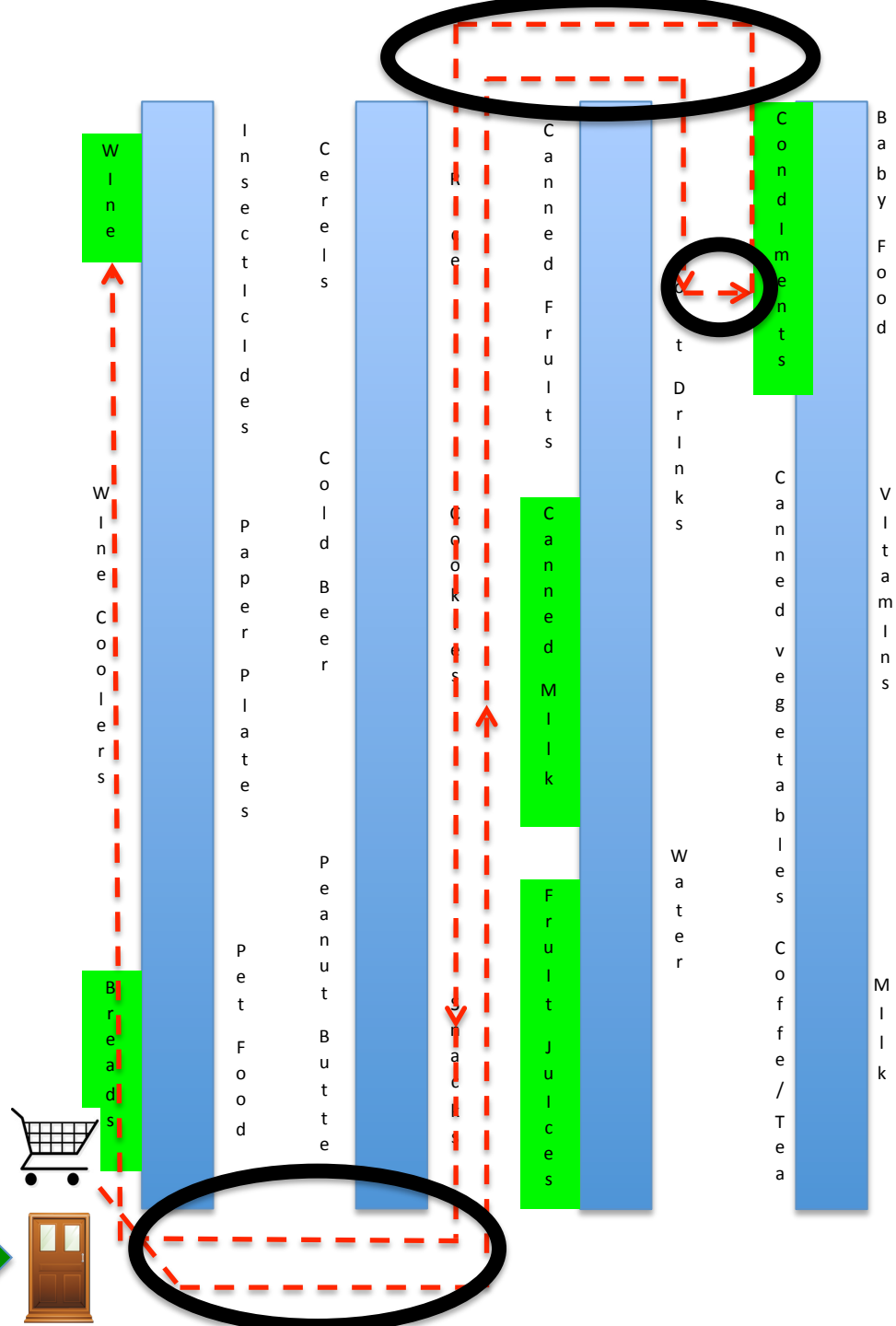
WALL


7:38
Publix Atl. Station
Enter item name
Breads
Wine
Cereals
Cold Beer
Peanut Butter

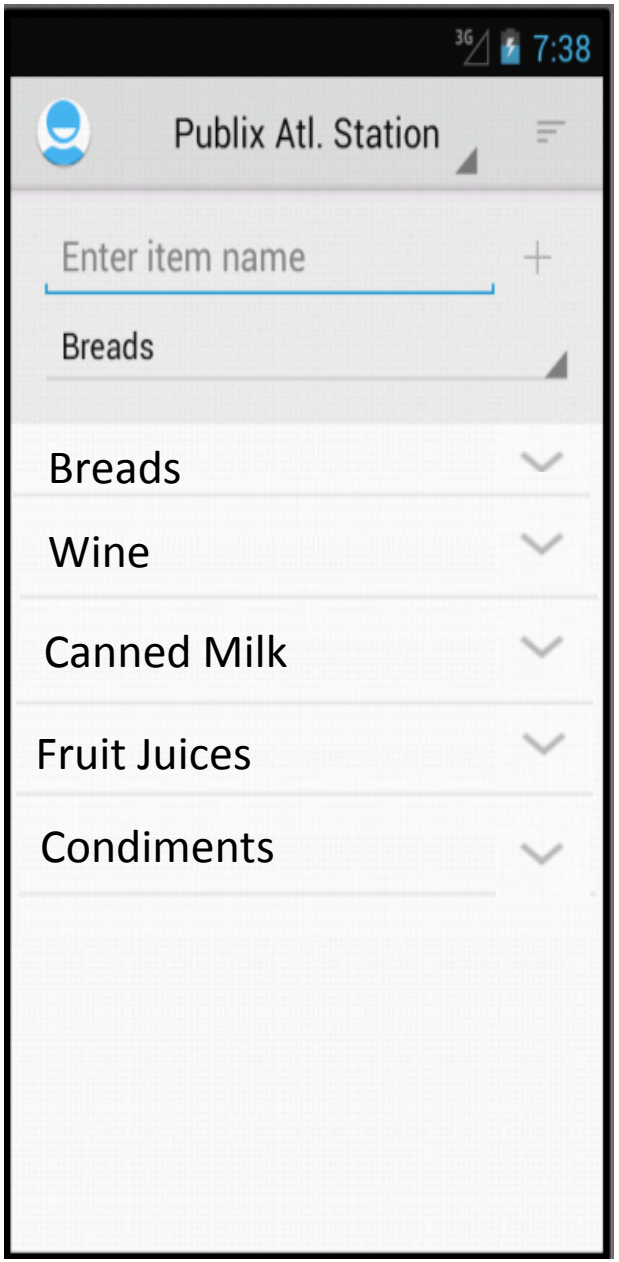

Wine
Insecticides
Cerels
Rice
Wine Coolers
Paper Plates
Cold Beer
Cookies
Breads
Pet Food
Peanut Butter
Snacks
Canned Fruits
Soft Drinks
Canned Milk
Water
Fruit Juices
Condiments
Baby Food
Canned vegetables
Vitamins
Coffe/Tea
Milk
Start point

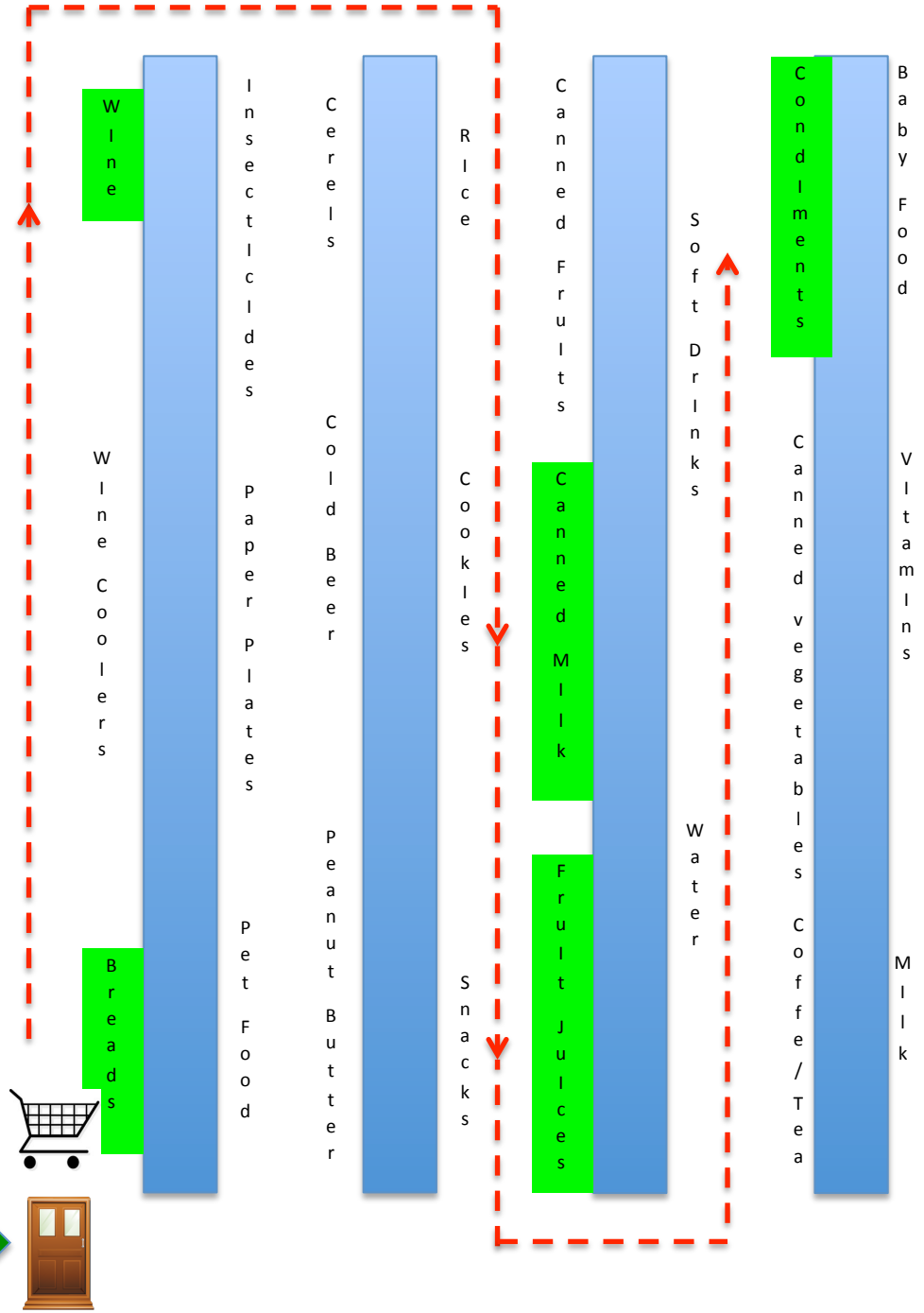
W A L L


7:38
Publix Atl. Station
Enter item name
Breads
Breads
Canned Milk
Condiments
Fruit Juices
Wine

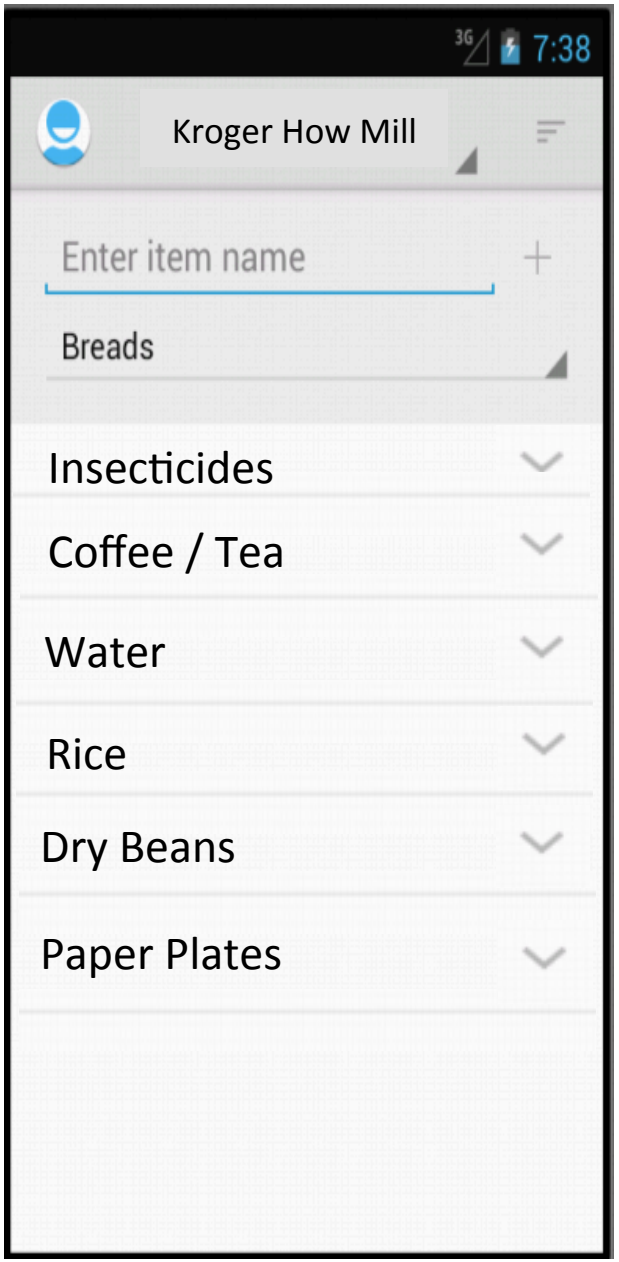

Wine
Insecticides
Cereals
Canned Fruits
Condiments
Baby Food
Wine Coolers
Paper Plates
Cold Beer
Cookies
Drinks
Vitamins
Breads
Pet Food
Peanut Butter
Snacks
Water
Canned Milk
Fruit Juices
Canned vegetables
Coffe/Tea
Milk
Start point

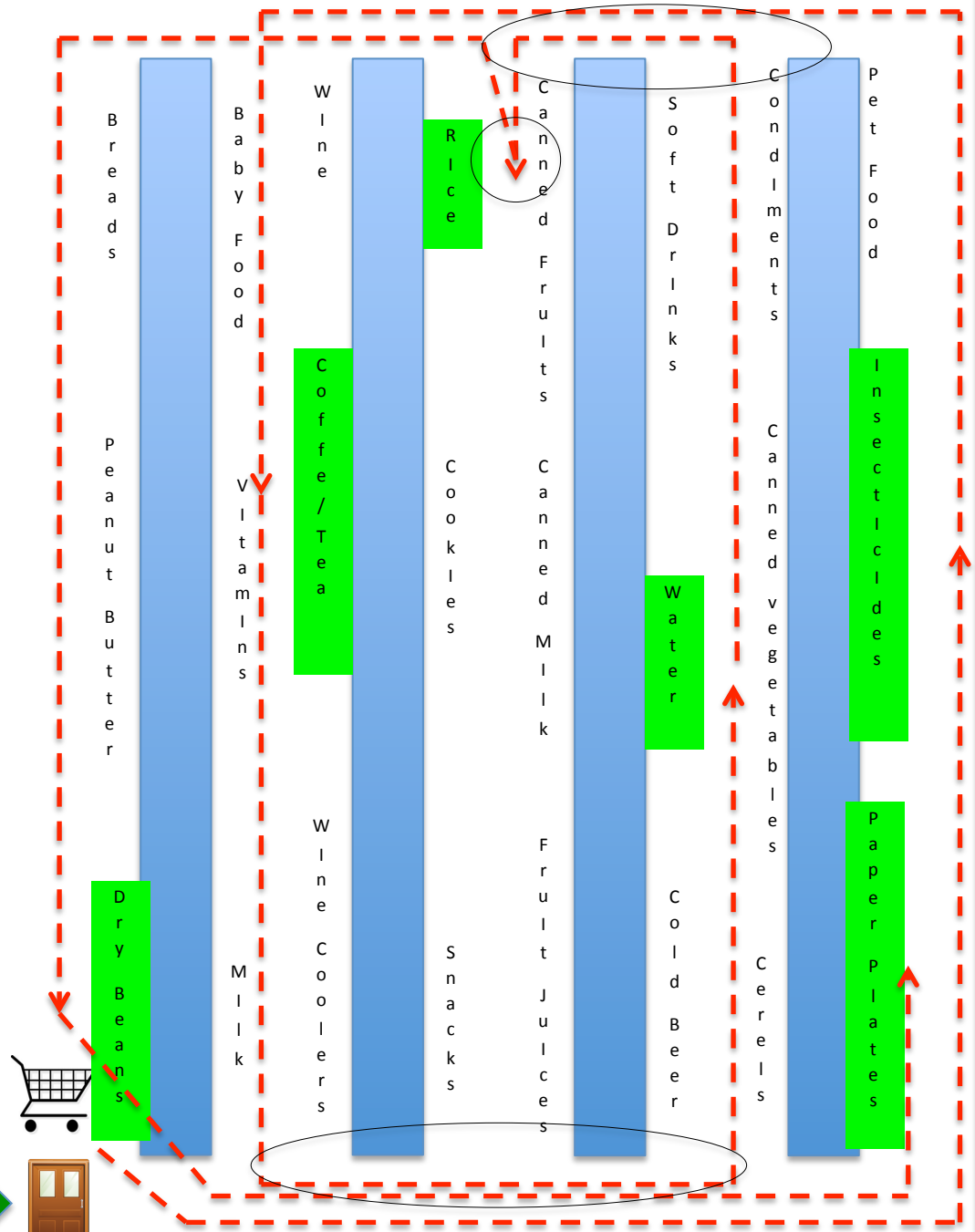
WALL


7:38
Publix Atl. Station
Enter item name
Breads
Breads
Wine
Canned Milk
Fruit Juices
Condiments

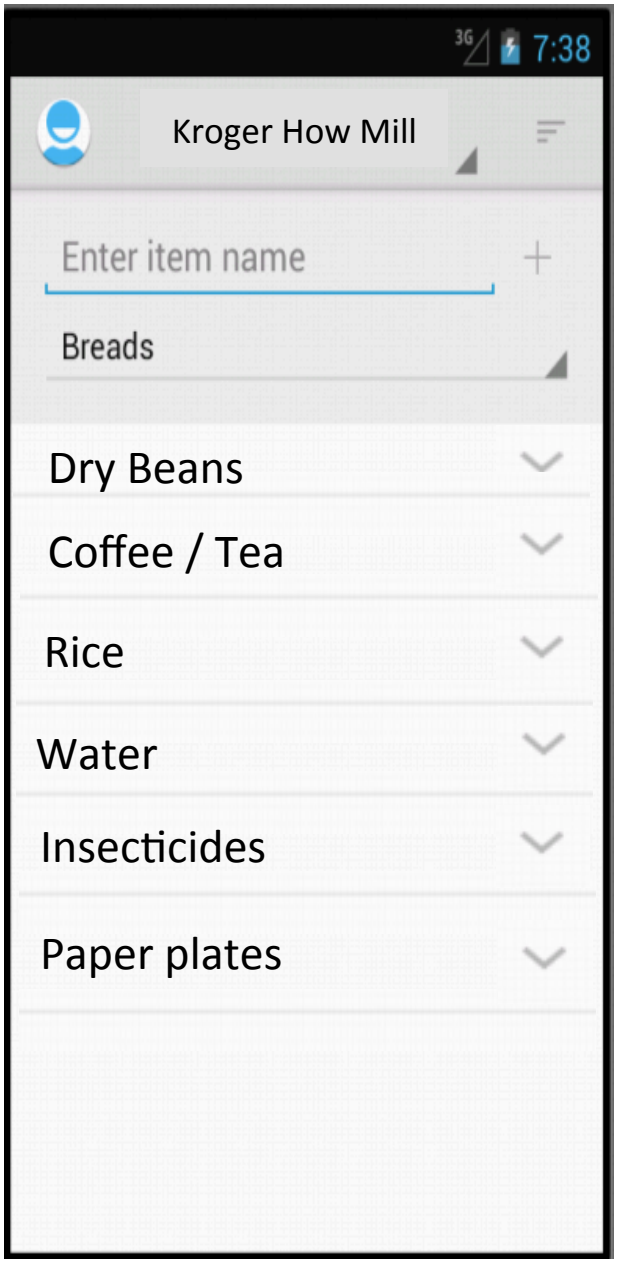

Wine
Breads
Wine Coolers
Insecticides
Paper Plates
Pet Food
Cerels
Cold Beer
Peanut Butter
Rice
Cookies
Snacks
Canned Fruits
Canned Milk
Fruit Juices
Soft Drinks
Water
Condiments
Canned vegetables
Coffe/Tea
Baby Food
Vitamins
Milk
Start point

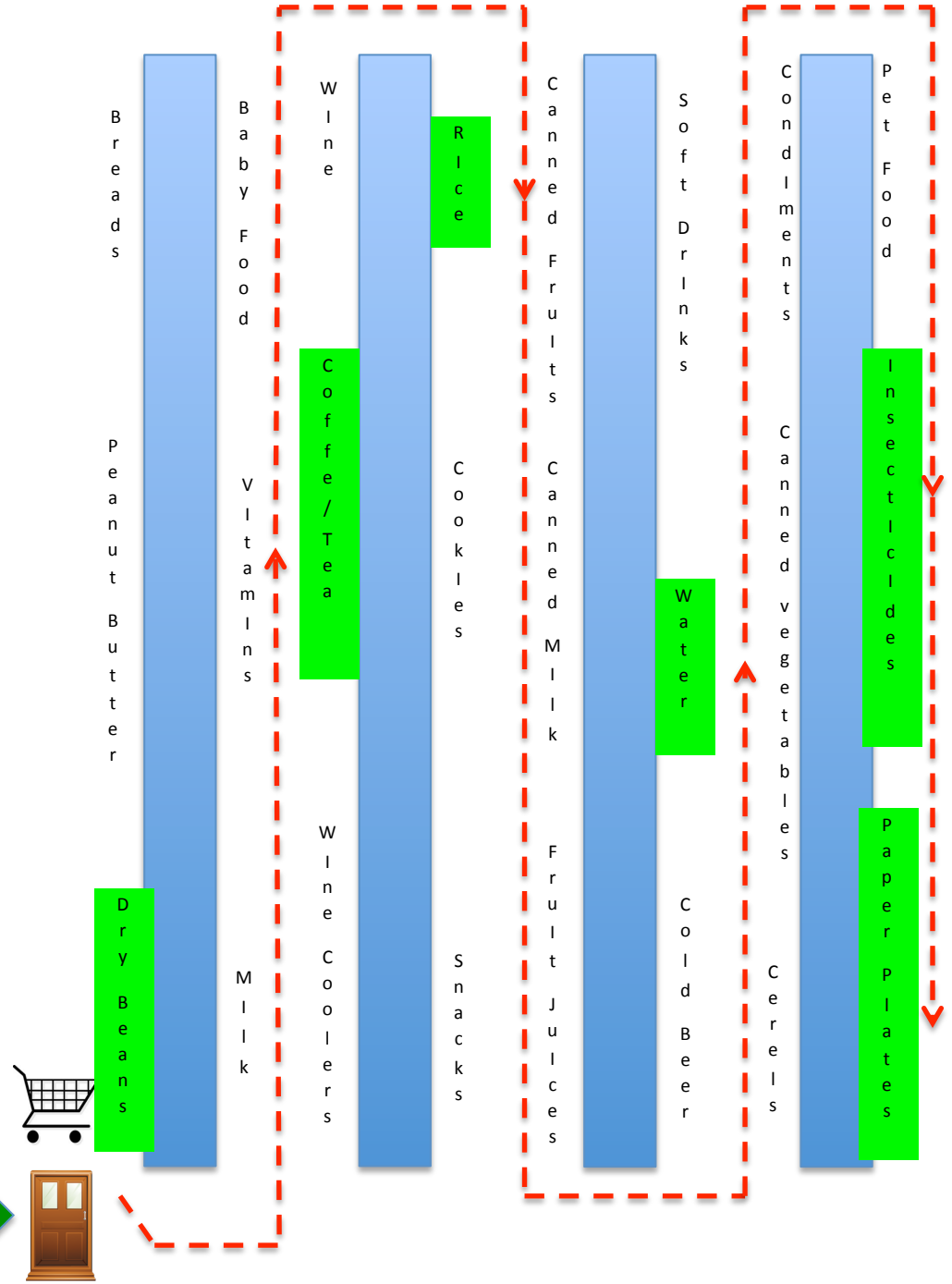
WALL


7:38
Kroger How Mill
Enter item name
Breads
Insecticides
Coffee / Tea
Water
Rice
Dry Beans
Paper Plates


Breads
Peanut Butter
Dry Beans
Baby Food
Vitamins
Milk
Wine
Coffe / Tea
Wine Coolers
Rice
Cookies
Snacks
Canned Fruits
Canned Milk
Fruit Juices
Soft Drinks
Water
Cold Beer
Condiments
Canned vegetables
Cerels
Pet Food
Insecticides
Paper Plates
W
A
L
L

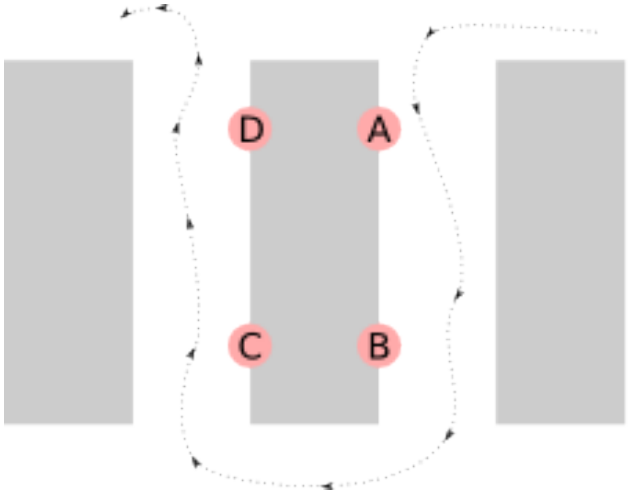
Start point

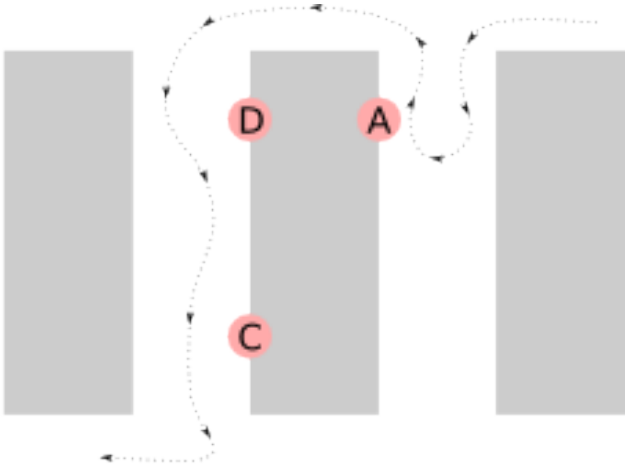

7:38
Kroger How Mill
Enter item name
Breads
Dry Beans
Coffee / Tea
Rice
Water
Insecticides
Paper plates


Breads
Baby Food
Wine
Rice
Canned Fruits
Soft Drinks
Condiments
Pet Food
Peanut Butter
Vitamins
Coffe/Tea
Cookies
Canned Milk
Water
Canned vegetables
Insecticides
Dry Beans
Milk
Wine Coolers
Snacks
Fruit Juices
Cold Beer
Cereals
Paper Plates
Start point

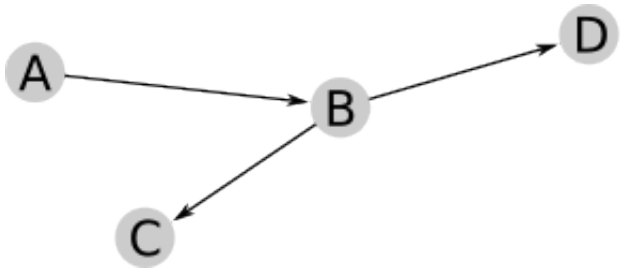
WALL

Issues in finding shortest path

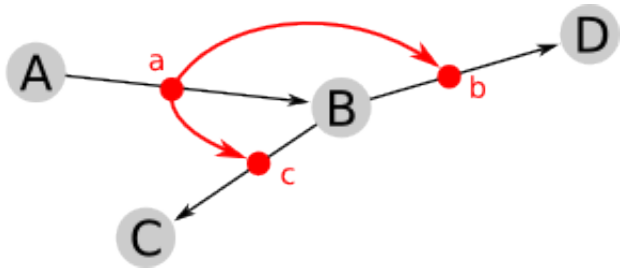




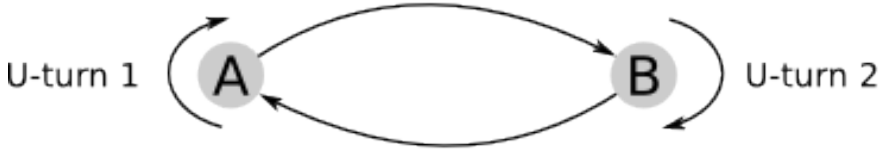
Modeling Turns

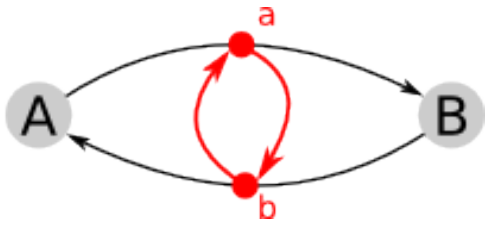


Primal Graph



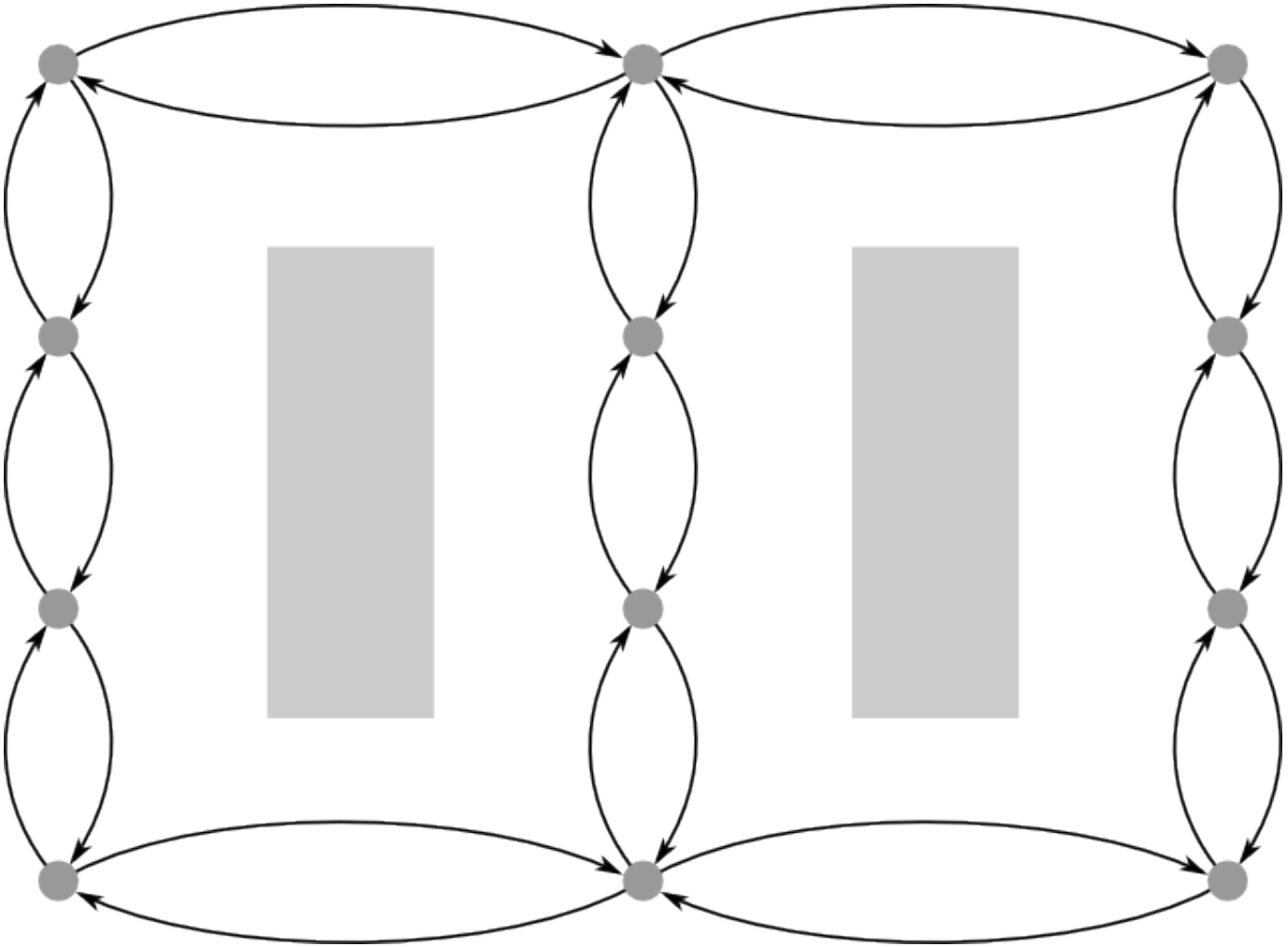
Dual Graph:
Nodes: Primal edges
Edges: Turns in primal graph



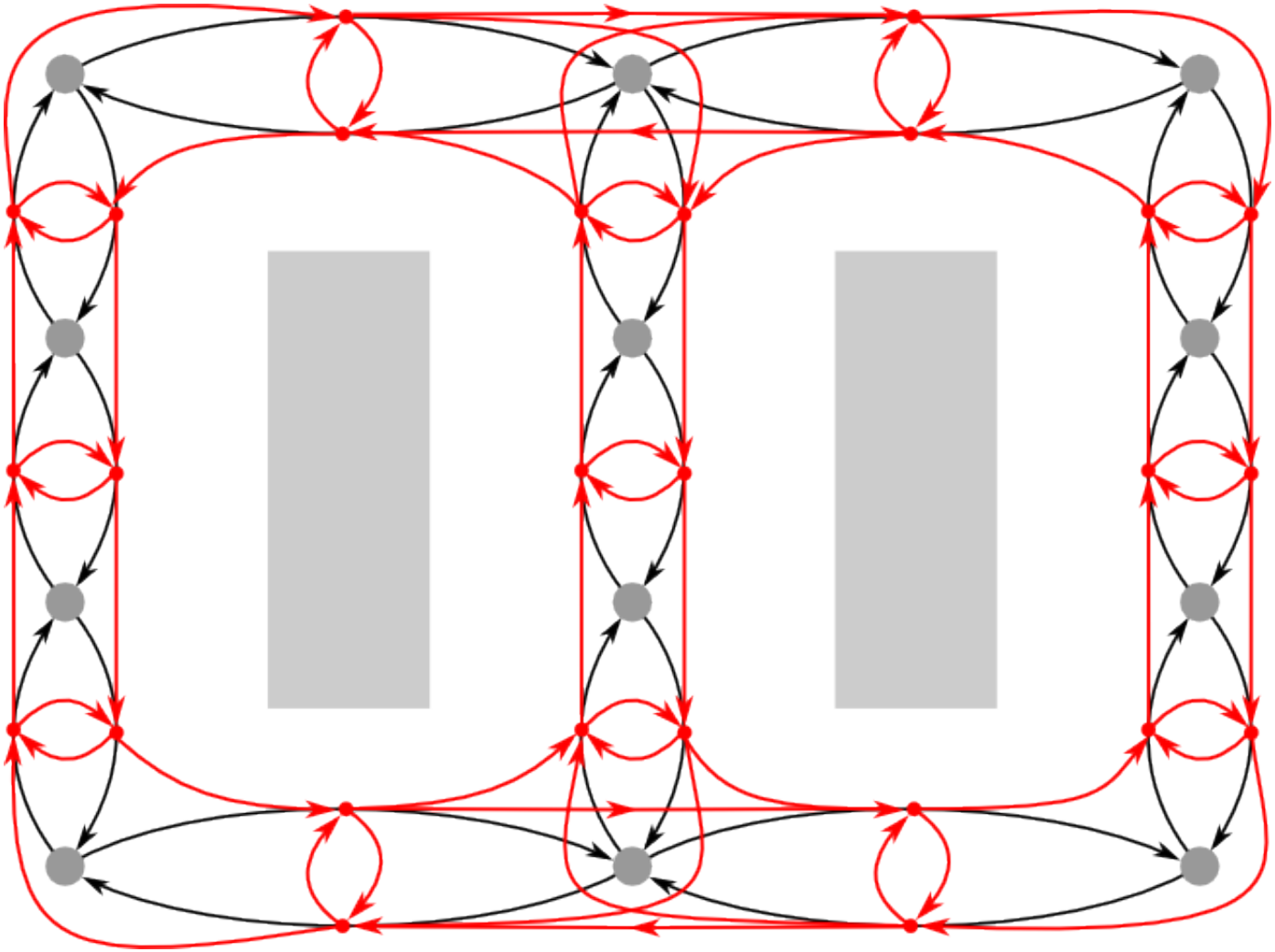


U-turns modeled naturally

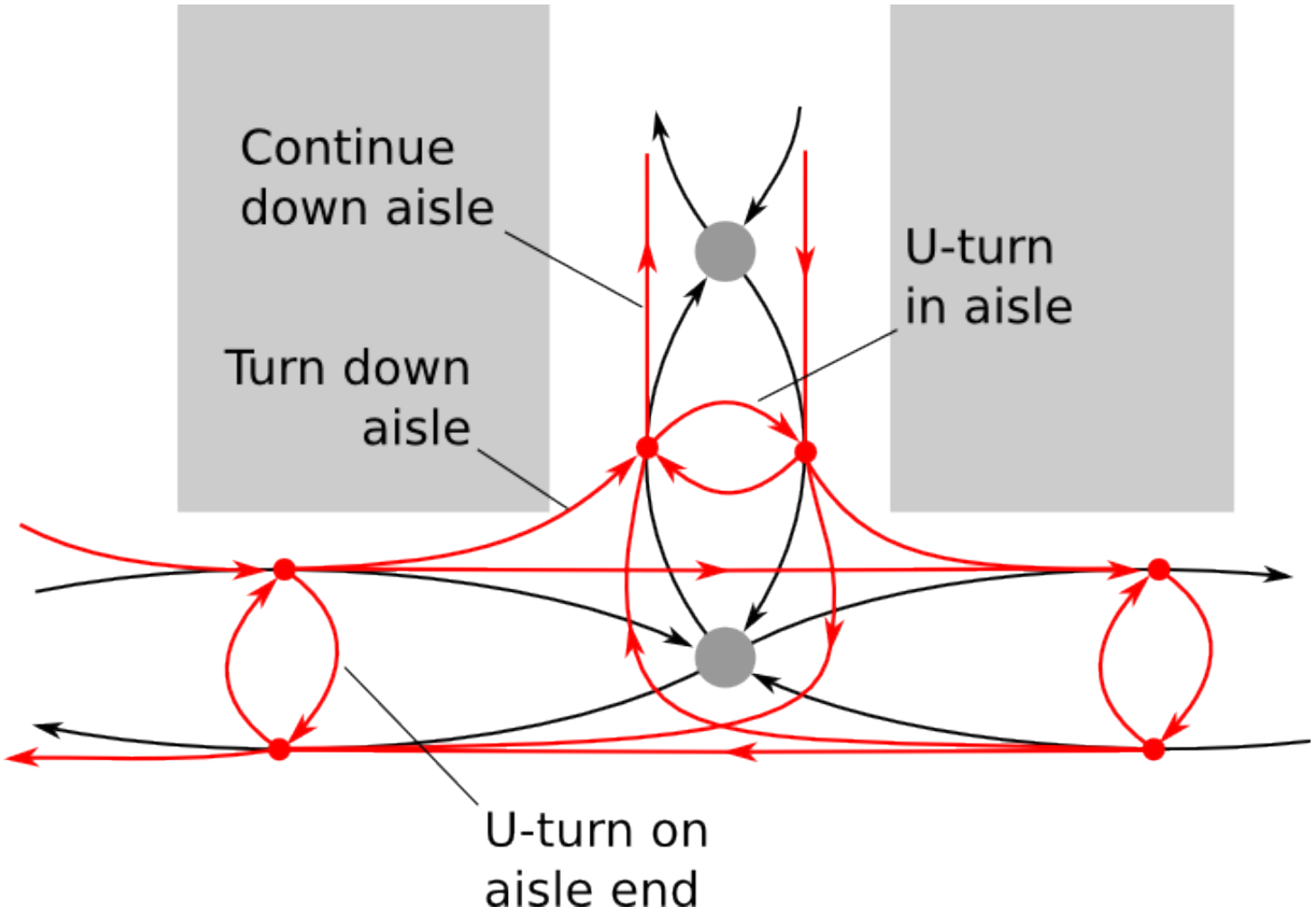
Layout of the store



Layout of the store



Turns in detail



Algorithm

- Start with fully characterized Dual graph
- Use all-pairs shortest path finder (Dijkstra's algorithm on every node)
- Each category associated with
  - a set of **primal edges**
  - a set of **dual nodes**
- Consider distance from a dual node to a category as:

  dist (*u*, category) = min { dist(*u, v*) }

- Greedy algorithm to walk from category to category