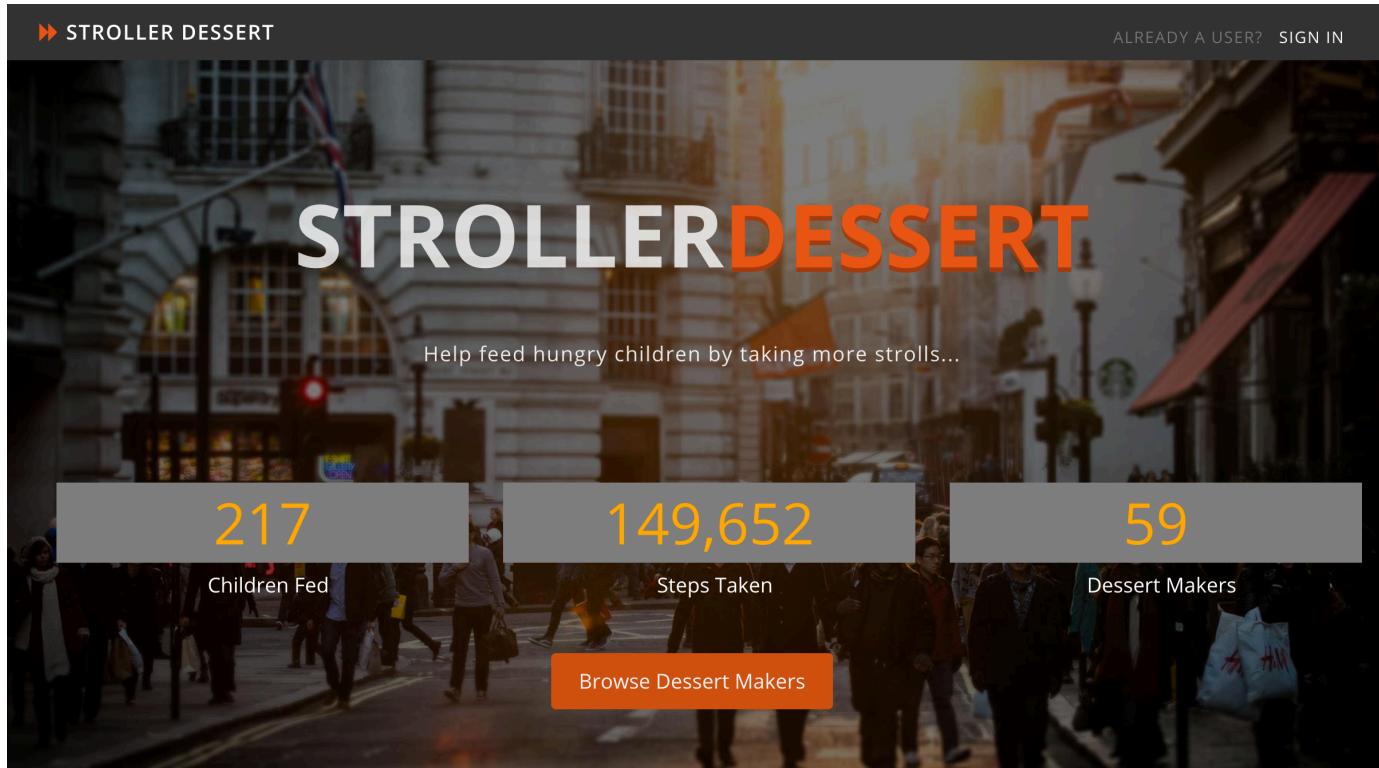
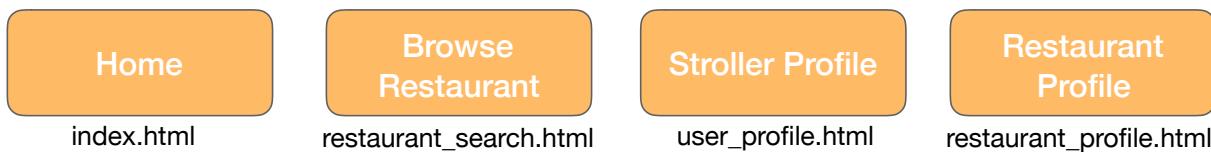


# Stroller Dessert by Hassan Badru

Jan 2017



## HTML PAGES



## REQUIREMENTS

### Stroll taker

- Check how many steps they've taken so far today ✓
- See dessert makers participating and how much dessert they've delivered ✓
- Track progress toward a goal ✓

### Dessert Maker

- Measure how many children they've treated to free dessert ✓

- Measure how many steps they've encourage strollers to take

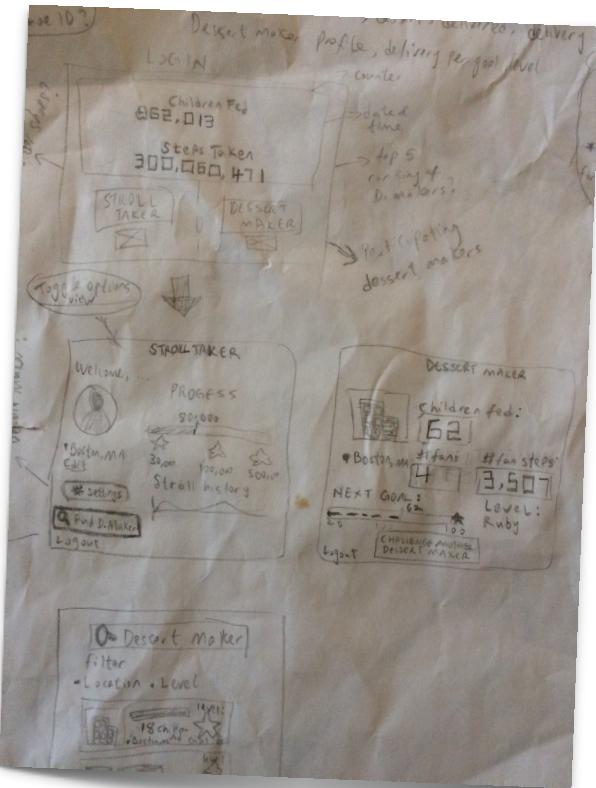
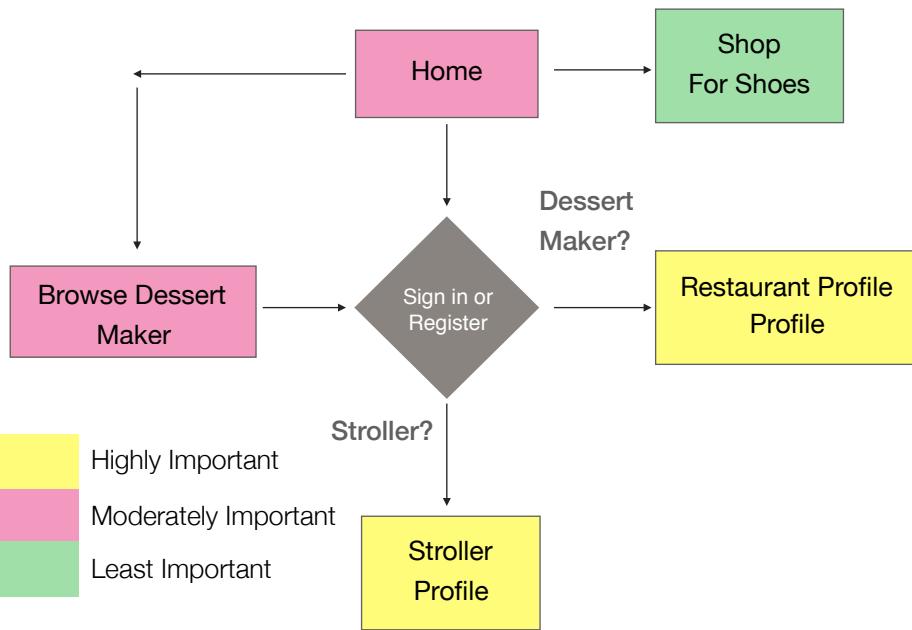


- Know when and where to deliver dessert



## MOCKUP

### FLOWCHART



### ASSUMPTIONS/NOTES

- There are currently 3 Dessert Makers on the platform, one per location
- Dessert Makers only make deliveries if they meet their monthly goal
- Although password authentication isn't required, users can login to Stroller Profile or Dessert Maker Portal
- "Dessert Maker" and "Restaurant" are used interchangeably on this project
- Sign In/Sign up for both strollers and restaurants are modal boxes
- Sign in/Sign up from: Home and Browse Restaurants

## HOME

## Libraries/References

CSS	JS	IMG
• <a href="#">bootstrap.css</a>	• <a href="#">jquery.min.js</a>	
• <a href="#">font-awesome.css</a>	• <a href="#">bootstrap.js</a>	
• <a href="#">style.css</a>		

## Features

- Children Fed: *Total number of desserts delivered by all restaurants*
- Steps Taken: *Total number of steps taken by all strollers*
- Dessert Makers: *Total number of dessert makers participating in the program*
- Sign up/Sign in to Stroller Account
- Sign up/Sign in to Restaurant Account
- Search for Restaurants (Dessert Makers)
- **Shop for shoes**

## BROWSE RESTAURANTS

## Libraries/References

CSS	JS
• <a href="#">bootstrap.css</a>	• <a href="#">jquery.min.js</a>
• <a href="#">font-awesome.css</a>	• <a href="#">bootstrap.js</a>
• <a href="#">style.css</a>	

## Features

- Restaurant Search
- Sign up/Sign in to Stroller Account
- Sign up/Sign in to Restaurant Account

## STROLLER PROFILE

## Libraries/References

<u>CSS</u>	<u>JS</u>
• <a href="#">bootstrap.css</a>	• <a href="#">jquery-3.1.0.min.js</a>
• <a href="#">font-awesome.css</a>	• <a href="#">bootstrap.js</a>
• <a href="#">style.css</a>	• <a href="#">jquery.1.11.1.js</a>
• <a href="#">profile.css</a>	• <a href="#">change_avatar.js</a>
	• <a href="#">Chart.min.js</a>
	• <a href="#">knockout-3.4.1.js</a>

## Features

- View total steps
- Next stroller level & steps left
- Profile information: *Name, Preferred Dessert restaurant & Location*
- Edit profile information
- Progress: *Steps today* and *total stroll miles covered*
- View Stroll History: *All, 12 months* and *6 months*
- Settings: *Change email/password, (add shoe ID) and set daily goal*
- Logout

## **RESTAURANT PROFILE**

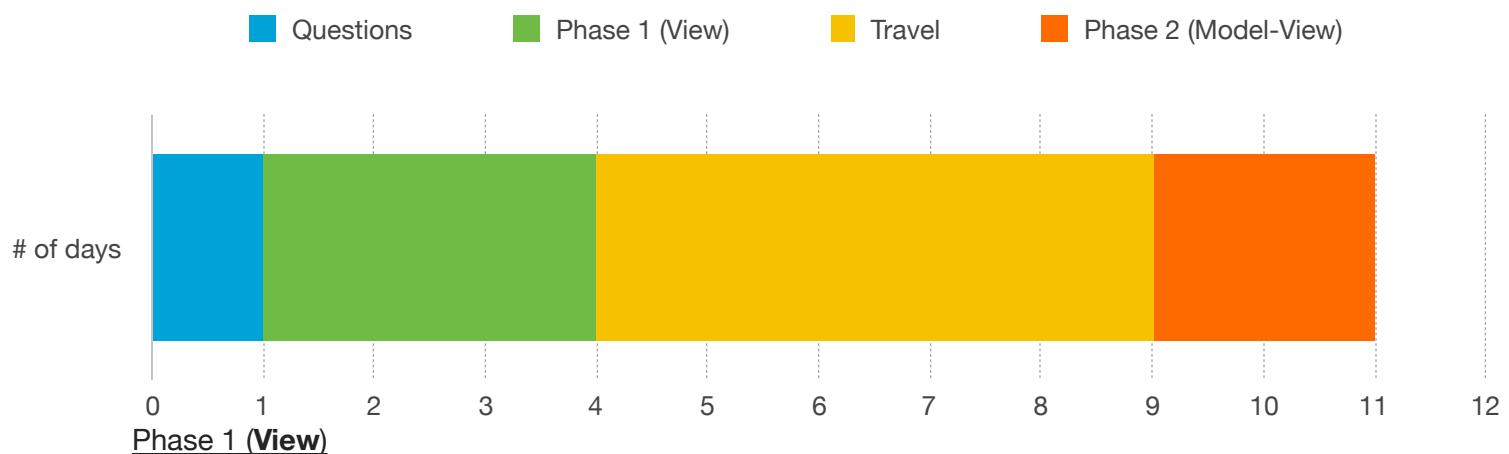
## Libraries/References

<u>CSS</u>	<u>JS</u>
• <a href="#">bootstrap.css</a>	• <a href="#">change_avatar.js</a>
• <a href="#">font-awesome.css</a>	• <a href="#">jquery.1.11.1.js</a>
• <a href="#">light-bootstrap-dashboard.css</a>	• <a href="#">bootstrap.js</a>
• <a href="#">style.css</a>	• <a href="#">jquery.svgDoughnutChart_rest.js</a>
• <a href="#">profile.css</a>	• <a href="#">knockout-3.4.1.js</a>

## Features

- Children Fed: Total desserts delivered
- Dessert delivery locations
- Number of fans
- Fan Steps: *Total Steps taken by fans and average steps per fan*

## TIMELINE



- Using HTML, CSS & JS

## Phase 2 (Model-View)

- Using KnockoutJS for data binding and keeping track of dependencies

## MODEL

Stroller (User) attributes: Bolded attributes are currently mutable by user\*

- |                               |  |                                     |
|-------------------------------|--|-------------------------------------|
| • shoeID                      | • <b>location</b>                                  | • history: All, 12 months, 6 months |
| • email                       | • <b>avatar</b>                                    |                                     |
| • password                    | • Totals Steps and Miles                           |                                     |
| • <b>name</b>                 | • progress: <i>stepsToday</i> and <i>dailyGoal</i> |                                     |
| • <b>Preferred restaurant</b> |  |                                     |

### Stroller Levels

Indigo	0 - 60,000 Steps
Red	60,001 - 308,000 Steps
Orange	308,000 - 1,240,000 Steps
Green	1,240,001 - 3,104,000 Steps
Blue	3,104,001 - 6,212,000 Steps
Silver	6,212,001 - 18,638,000 Steps
Gold	18,638,001+ Steps

Dessert Maker (Restaurant) attributes: Bolded attributes are currently mutable by user\*

- |  |   |   |
|--|---|---|
| <ul style="list-style-type: none"> <li>• <b>username</b></li> <li>• email</li> <li>• password</li> <li>• <b>restaurant name</b></li> <li>• number of fans</li> </ul> | <ul style="list-style-type: none"> <li>• <b>location</b></li> <li>• <b>avatar</b></li> <li>• number of steps by fans</li> <li>• Desserts Delivered</li> <li>• delivery history (date and location)</li> </ul> | <ul style="list-style-type: none"> <li>• <b>Monthly Goal</b></li> </ul> |
|--|---|---|

Home (Achievements) attributes:

- **Children Fed:** Sum of Desserts Delivered by all restaurants/Dessert makers
- **Steps Taken:** Sum of all steps taking by every *Stroller*
- **Dessert Makers:** Total Number of Dessert Makers

## USER EXPERIENCE

### HOME

- A counter is used to keep track of overall progress and gives first time visitors an idea of Stroller Desserts' impact
- “Browse Dessert Makers” button allows visitors get straight into it searching for restaurants
- Sign in or register as a stroller (regular user) or as a dessert maker using the “Dessert Maker Portal” button
- Shop for Stroll Shoes

### BROWSE RESTAURANTS

- Visitors can search for Dessert Makers using any keyword such as name, location, username or any statistics
- Option to sign in or sign up as a regular user
- Option to sign in or sign up as a dessert maker

The screenshot shows a search interface for "DESSERT MAKERS". At the top, there's a header with a "STROLLER DESSERT" logo and a "SIGN IN" link. Below the header, a sub-header reads "DESSERT MAKERS". A subtext explains the service: "Using our proprietary pair of Strolling Sneakers, we track the number of steps that people take and then we send free dessert to children in need. More strolls, more dessert; it's a classic win-win." A "Dessert Makers Portal" button is visible. Below the subtext, there's a search bar labeled "Find A Dessert Maker" with a placeholder "Enter any keyword". Underneath the search bar are two cards, each showing a thumbnail of a dessert maker, their name, the number of children fed, the number of fans, and their location.

Thumbnail	Name	Children Fed	Number of Fans	Location
	25	108	729,100	Los Angeles, CA
	11	17	37,760	New York, NY

### STROLLER PROFILE

- Show and edit basic profile information: Avatar, Name, Preferred Dessert Maker (Restaurant) and Location
- Shows total number of steps Stroller has completed and progress towards the next level
- Includes a legend to help stroller understand the amount of steps needed to reach the next level
- Shows the number of steps taken so far for *today* vs *daily goal* set by the strollers within settings.
- Shows total number of miles completed (assuming 2,000 steps = 1 mile)
- Shows stroll history that can be

The screenshot shows a user profile for "HASSAN BADRU" from "The Cake Factory" in "New York, NY". The profile includes a circular avatar, the user's name, the restaurant they prefer, and the city they are in. It also displays the user's current step count and the distance needed to reach the next level. Below the profile, there are two sections: "Progress" and "Stroll History". The "Progress" section shows a donut chart with the percentage of steps completed today. The "Stroll History" section lists the user's activity over different time frames: All, 12 Months, and 6 Months. It includes a table with columns for Date, Steps, and Miles.

Date	Steps	Miles
Yesterday	1,073	0.54
May 6, 2016	4,338	2.17
Now 12:00 PM	4,228	2.17

filtered based on: All strolls, with past 6 months, within past 1 year

## DESSERT MAKER (RESTAURANT) PROFILE

- Show and edit basic profile information: Avatar, Name and Location
- Shows dessert delivery history
- Shows total number of steps Stroller has completed and progress towards the next level
- Includes a legend to help stroller understand the amount of steps needed to reach the next level

- Shows the number of steps taken so far for that day vs the daily goal set by the strollers within settings.
- Shows total number of miles completed (assuming 2,000 steps = 1 mile)
- Shows stroll history that can be filtered based on: All strolls, with past 6 months, within past 1 year

```
{
  "data": [
    /*=====
     * USER 1 *
     =====*/
    {
      "shoeID": "001",
      "email": "adam@sd.com",
      "password": "123",
      "name": "Adam Goldstein",
      "restaurant": "Cupcake Deli",
      "location": "Cambridge, MA",
      "avatar": "img/photo1.jpg",
      "steps": 100000, /* 100,000 */
      "progress": {"stepsToday": 3500,
                  "dailyGoal": 4000
                },
      "hist": {"all": [
        {"Date": "Yesterday", "Steps": 1073, "Miles": 0.54},
        {"Date": "May 6, 2016", "Steps": 4338, "Miles": 2.17},
        {"Date": "Yesterday", "Steps": 4600, "Miles": 2.30}
      ]},
      "twelve": [
        {"Date": "Yesterday", "Steps": 1073, "Miles": 0.54}
      ]
    }
  ]
}
```

## **MODEL VIEW VIEW-MODEL ARCHITECTURE:**

Framework used: KnockOutJS

Binding: Declarative “Two-Way” Binding

```

<script>

// This is the *viewmodel*
var viewModel = {
    fullName : ko.observable("Hassan Badru"),
    location: ko.observable("New York, NY"),
    optionValues: ["The Cake Factory", "Cupcake Deli", "Croissant Express"],
    selectedOptionValue: ko.observable("The Cake Factory"),
    dGoalOpt: ["1000", "2000", "3000", "4000", "5000"],
    dGoal: ko.observable("2000"),
    dSteps: "1213"
};

this.dLeft = ko.computed(function(){
    return parseInt(this.dGoal) - parseInt(this.dSteps);
});

ko.applyBindings(viewModel);
</script>

```

different instances of objects (user and restaurant).

The objects would be stored as a JSON file. It would consist the various instances of **Stroller** and **Dessert Maker**.

Values like **% Goal Completed** are computed using values from the **user** object such **stepsToday** and **dailyGoal**.

Using the knockoutJS observables, I was able to enable each user set a Daily Goal. The % completed is then computed by dividing the total steps for that day by the daily goal. This fraction is then plotting on the Donut (Pie) Chart but I had issues with my Programs environment when it tried executing the object property *modelview.dGoals()*. This wasn't experienced when I ran my script through console and the changing string values were converted to integers.

## CONCLUSION & FUTURE WORK

Given more time. I would use ReactJS, the plan is to render the HTML elements using JSX. This would make it possible for the web app to become more dynamic and more quickly manipulate the DOM Structure. Each mutable element would be assigned to a variable which would be rendered based on the