Beans

Brief

Eating well and living mindfully leads away from supermarkets with their long supply chains, squeezing of farmers and chasing of shareholder profits toward small groups of people cooperating to buy from trusted farmers in their area. To source organic food locally and reliably though is quite complex especially on the scale of 10 - 30 members of a typical food group. What is needed is an app which takes away much of the pain of organising custom orders of regular local ethical food.

A sophisticated and intuitive app which enables the many facets of a food group to work smoothly could also then be a major help in others wishing to adopt the local food group approach. This is not a lucrative business opportunity: there is not more than $50 per week per group for this kind of app. However for those who want the freshest most nutritious food grown in a way that builds soil and gives local farmers a fair return this app would soon become essential.

Russell & Gabrielle Austerberry have been involved in such a group for 5 years, using a local online web app. While ensuring $3000+ of custom orders per fortnight within a group of 40 people would be impossible without it, there are many places where friction could be reduced. 'Beans' is a re-think and extension of existing software.

The development of 'Beans' also is a means for Russell to learn and showcase skills in web development with a strong focus in Vue JS.

Beans

Easy administration for Food Buying Groups of 15+ members

Goals - project

1. An easy-to-administer management tool for food group organisers
2. An easy-to-use ordering and reminder tool for food group members
3. A fast, mobile, easy to use app that works well even with flaky internet
4. An app which assists admin & members as follows:
   1. Sophisticated import & processing of supplier availability files
   2. Intuitive ordering and shopping cart experience
   3. Managing orders to suppliers
   4. Generated printouts for delivery, packing and pickup
   5. Timely notifications re ordering, pickup, packing rosters, etc
   6. In-app tracking of invoices & payments for both suppliers and members
   7. API/s to allow integration with 3rd party invoicing, chat and ordering software
   8. Flexible array of real-time reporting features

Goals – development skills

Demonstrate web development proficiency across the following areas.

To view commits and code in Github, let me know your username so I can invite you as collaborator on private repository.

Progress rated by stars, as at May 2020:
- ● ○ ○   preliminary research
- ● ● ○   basic implementation
- ● ● ● thorough implementation

1. Planning (this document)
    1. concept articulation & development   ● ● ○
    2. user stories, sketches & wireframes   ● ● ○
    3. data modelling   ● ● ●
2. Setup
    1. Vue CLI   ● ● ●
    2. version control (Git)   ● ● ○
    3. Test coverage   ● ○ ○
    4. Deploy to web   ● ○ ○
3. Vue
    1. folder structure   ● ● ○
    2. naming convention   ● ● ○
    3. modular & elegant code   ● ● ○
    4. well commented code   ● ● ●
    5. appropriate data structures   ● ● ○
    6. logical routing system   ● ● ○
    7. navigation system   ● ● ○
    8. scalable state management   ● ● ○
    9. fast & reactive app   ● ○ ○
    10.   mix of libraries, JS, HTML & CSS   ● ● ○
4. Persistent storage
    1. in-app first (via PouchDB)   ● ○ ○
    2. synced to cloud (via CloudDB)   ● ○ ○

Beans

Easy administration for Food Buying Groups of 15+ members

Features

1. Admin
   1. calendar based order cycle management
   2. Role based management of packing teams & others
   3. Import and translate supplier lists to standard 'beans' format
   4. Fast multi-level sorting, culling, modifying & replacement of items
   5. Flexible pricing structures
   6. Print-ready exports of various lists (delivery checklist, member orders, packing sheets)
   7. API for interaction with other ordering systems
2. Finance
   1. Auto invoicing with tracking for member & supplier balances
   2. Real time reports: sales|surcharge|member-history per order cycle| quarter|year
   3. API for integration with accounting package (Xero etc)
3. Members
   1. Easy online ordering
   2. Reminders & notifications (to order, buy extras, news, call for packers)
   3. 'Demo mode' when outside ordering window
   4. Customisable recurring orders
   5. invoice, payment & balance history on demand
4. General
   1. authentication & role-based permissions for access to ordering/admin/ finance functionality
   2. onboarding and setup wizards

Beans

Easy administration for Food Buying Groups of 15+ members

Phases

Progress as at May 2020 marked by 1 to 3 stars.

Preliminaries: ● ● ●
  Planning
  Architecture
  Tech stack
  Version Control
  Initial setup of project
  Basic folder structure
  Navigation model
  Routing model
Scaffolding:
  Look and feel   ● ● ○
  State management pattern   ● ● ○
  Persistent storage patterns   ○ ○ ○
  Component development: imports   ● ○ ○
Development:
  Admin   ○ ○ ○
      calendar
      calendar-based order cycle
      main admin interface
  Authentication
      authentication system with roles & permissions
      notification system – based on calendar, roles, permissions
  Ordering
      shopping cart for members
      order → invoice tracking + member balances
      ordering management via suppliers
  Accounting
      track all transactions
      export to csv
  APIs
      notification API for integration with Slack or similar
      availability API for suppliers to list items
      API for integration with Xero etc
Production: ○ ○ ○
  Testing & Deployment
      set up testing framework
      set up hosted solution
      deploy
  Maintenance

Beans

Easy administration for Food Buying Groups of 15+ members