

Team 1 Proposal

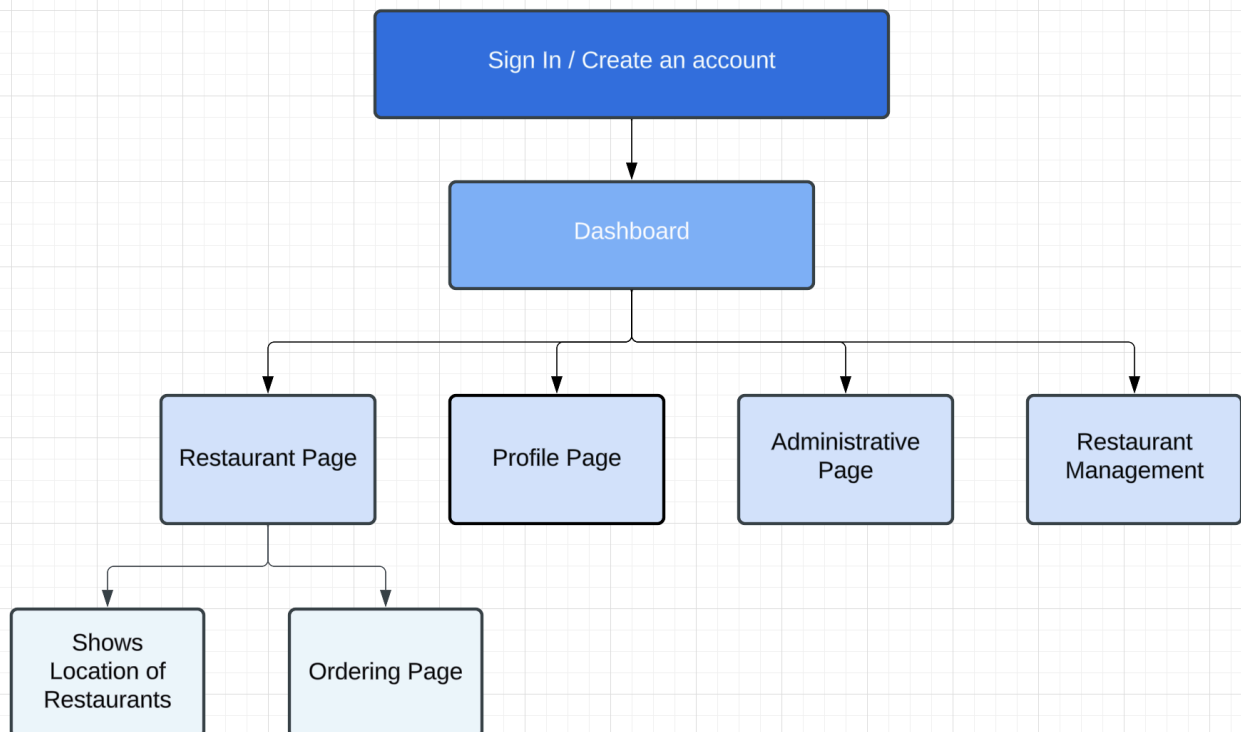
By: Peter Simone, Elizabeth Rice, Angela Rudel, Alex Xie, Colin Pace,
Justin Isaac

Introduction / Idea

Chew Chew Train

Drawing inspiration from similar applications that allow you to remotely place orders, we wanted to try and make one that would (theoretically, at least) work with RPI students, and specifically, to support use of Flex dollars. In a sense, the concept is related to the system that once existed at RPI to order food from the Union during COVID. This project, however, is meant to be a broader generalization of the system. Currently, you can order remotely at *some* locations, but oftentimes the process is different for each restaurant/location.

Site Map



Here is a summary of the above map:

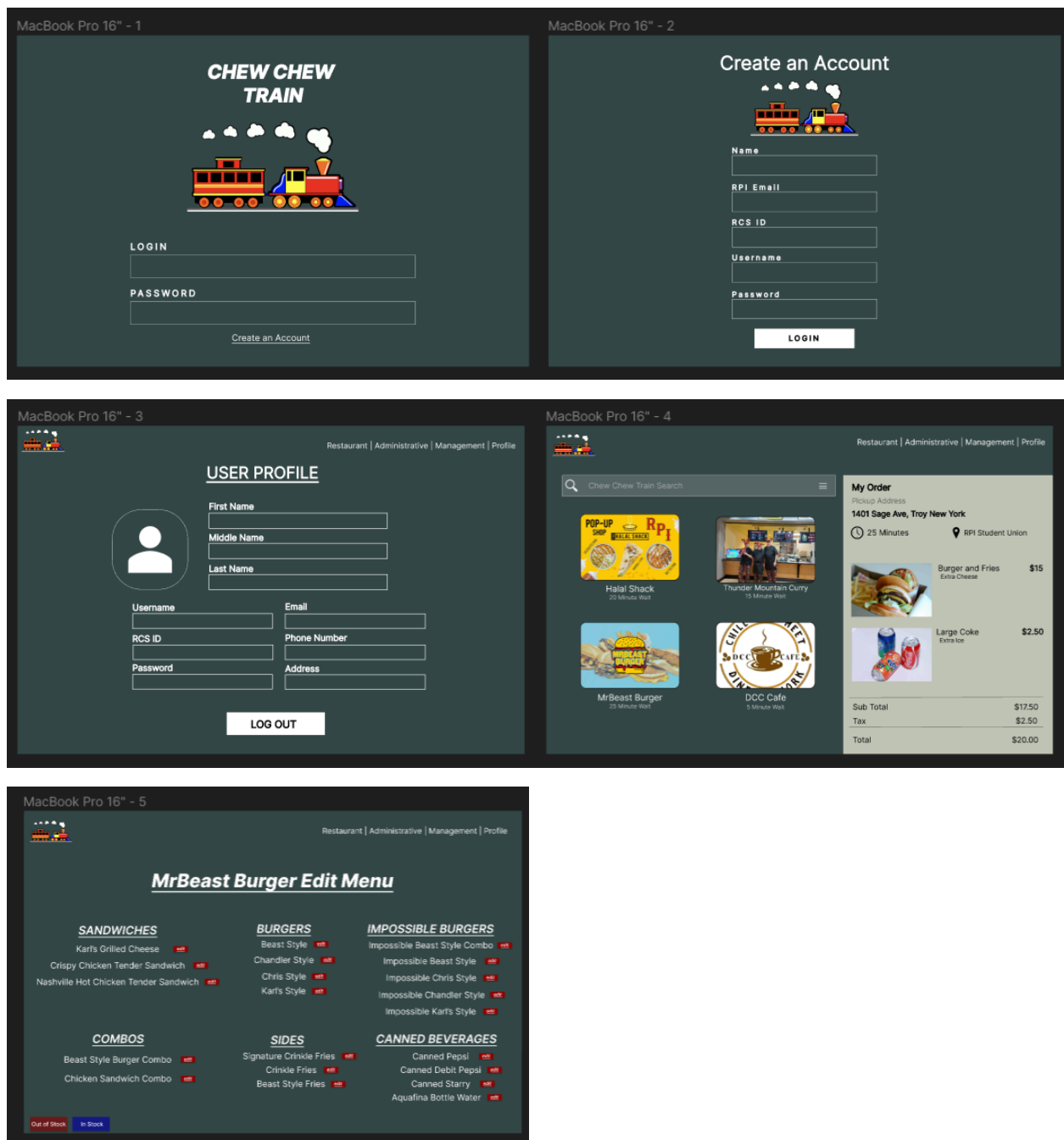
- Landing page
 - Doubles as the restaurant list page; shows all locations
- Restaurant view page
 - Doubles as the ordering page, potentially (may be separated later if needed)
- Sign in / register page(s)
- (Private) profile page
- Restaurant management page
 - Used to manage menu items, employees, orders, and so on
- Administration page
 - Used by site admins to add/remove restaurants and manage permissions

Information Architecture

- API structure
 - Generally will follow most of REST API standards; various endpoints for dealing with orders, restaurants, administration
- Database structure
 - Using MongoDB, data is structured in a format similar to JSON documents
 - We can store restaurants as JSON objects, users as JSON objects, and so on. It should be fairly flexible and extensible as the project progresses
 - The most complex part will be the ordering system, as data will need to be updated and fetched in real-time. Still, each order will likely be represented as a single JSON object

Initial Mockup

Below is our current mockup of some critical pages on the website, with a consistent theme:



Stakeholder Analysis

- RPI Students
 - Saves time & boosts productivity, by reducing time spent waiting in lines
 - Example: Imagine being in a class that ends at 12PM, wanting to get food from the Union, and being unable to do so until class ends, *and* walking all the way to the Union. With places such as Panera, you may do this with your own money, but not necessarily with Flex—and there's no connected system.
- RPI visitors
 - Would be able to more efficiently figure out the times certain locations are open (such as in the Union), and be able to order from them effectively.
- Union employees
 - Would be able to process more orders, and satisfy more customers at busy hours (since they order ahead of time), because there's a time window to be able to schedule everything.
- Union restaurants
 - Increased sales from more orders
 - Increased visibility

Functional Requirements

- Ability for customers to order food remotely.
- Ability for employees to track and manage orders.
- Administrative features to update restaurants, available items for purchase, etc.
- User accounts and registration
- Ability to pay for orders using Flex or Paypal
- Restaurant registration and verification

Non-functional Requirements

- For simplification, we will use fake currency (which seems real on the site)
- Scheduling system for orders that will be able to track data in realtime, automatically remove old information, and so on.
- User accounts—authentication with RCS ID if possible?
- Administration: need multiple levels

- Base user (customer): can only place orders, and other basic interaction
- Restaurant employee: can track and manage orders, on top of above
- Restaurant owner: can manage overarching information about restaurant
- Site admin: can oversee/create/remove restaurants, as well as users

Schedule

- Project proposal presentation & short video: *January 23rd*
- Midterm presentation: *March 1st*
 - In February, we need to create a workable frontend, and semi-functioning backend (at least, depending on which we do first).
 - Some mock data (e.g. for restaurants) should be present on the site for this.
- Final presentations begin: *April 16th*
 - Throughout March and early April, we will finalize our backend operations and polish our frontend, to whatever extent necessary.
 - Final restaurant data will be added to the site's database.
 - Security testing and fixing should also be performed before this presentation.