Sunshine Waiter

COMP9900 Project Proposal

Group Name: Sunshine-Waiter

Submission Date: Sunday, 8 March 2020

Zitong Li	z5189688	z5189688@ad.unsw.edu.au	Scrum Master
Yaoye Lu	z5188093	y.y.lu@student.unsw.edu.au	Developer
Yingjie Zheng	z5213329	yingjie.zheng1@student.unsw.edu.au	Developer
Jiangyuan Li	z5182721	jiangyuan.li@student.unsw.edu.au	Developer

We choose the project 6, Waiter, as our project this term.

Background

Many restaurants or cafes now have an increasingly demanding for an automated ordering system that allows customers to place orders online. There are plenty of advantages to providing such service:

1. Saving time, both for the restaurants and customers.
2. Can take care of multiple customers simultaneously, which may need extra waiters/ waitresses if not using such a system.
3. Having a database that stores all the activities data (logs) so that it can provide some insights on the business condition for the restaurant owner.

Existing systems and their drawbacks

There are several restaurant management systems used on the market, but cannot exactly fulfil the needs of this project. The market lacks such a waiter service system that can be generally implemented by any kind of restaurants/cafes.

1. Yelp

Yelp is a business directory service and crowd-sourced review forum where users can leave their comments on restaurants they visited. It is a great place to find a popular restaurant.

Drawbacks: However, Yelp doesn't provide an automated order service in a restaurant. Customer can't view the menu or place order online.

2. Mymacca's

A mobile app created by Mcdonald's for the customers so that they can view menu and place orders online without queuing.

Drawbacks: The app is designed specifically for Mcdonald's customers. Other restaurant managers cannot use this system to develop their own business.

3. MenuDrive

This website provides a service that can transfer your traditional menu into an interactive online menu by just sending them the pictures, so no HTML or programming knowledge is required. Once the system is implemented into your restaurant, it can handle multiple order types like a pickup, delivery and dine-in.

Drawbacks: Every customer who wants to order something has to sign in with his/her e-mail or other social accounts. So on the side of customers, their personal information might be used for marketing promotions like advertising and customers may also find it cumbersome because sometimes they just want a fast meal ("Why do I have to sign in just for a hot dog?").

And the visual style of its online menu is not modern enough, which might not be attractive to some users.

4. Toast

Toast offers a popular point of sale system for restaurants that also includes an online ordering system. This option is most suited to the restaurants that already use Toast POS. The whole system is meant to work together to help restaurants manage all of their operations in one system, to even include online order reports and delivery systems. Pricing starts at $140 per month.

Drawbacks: High pricing. Same potential problems as MenuDrive.

Purpose of the system and features

When people go out for a meal, sometimes they need to wait for a long time to place order in peak time. Too long waiting time may make them feel frustrated and unhappy. In order to control the waiting time, the restaurant has to hire more waiters , which will increase the financial burden.

Therefore, our product "waiter" is proposed to address these issues as well as the drawbacks mentioned above in the existing competitors. we are aiming to create a useful and easy-to-use ordering system that benefits both customer and resturant.

For customers, our system provides functions of placing orders online and requesting assistance from a waiter without registration..

For restaurant, our product provides different pages according to different roles including waiter, kitchen staff and manager. The detail is as following:

1. The page for waiter allows to get request from customer and mark as completed once it is handled. Besides, it allows them to deal with the prepared order from kitchen and serve it to customer.

2. The page for kitchen provides kitchen staff a time-sorted list of customer orders and allow them to mark an order item as prepared once it is ready for serving.

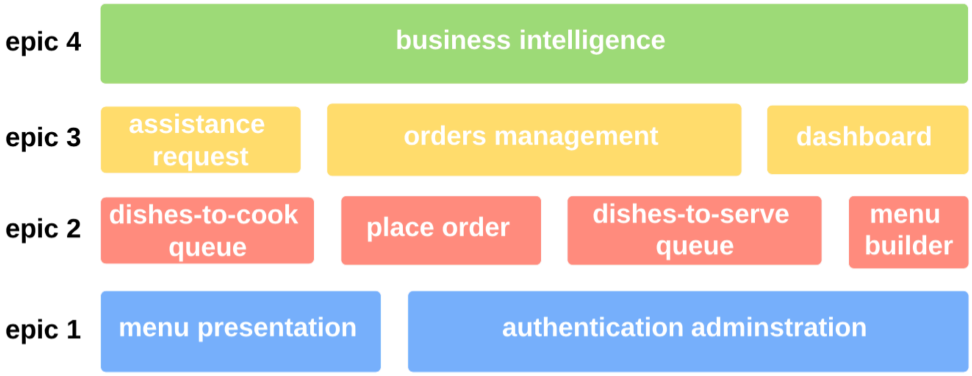
3. The page for manager allow to create, update, delete menu. With an easy menu tool, the manager can quickly add items to site completing with high resolution images, description, pricing, category and more. And also, the manager can update the sequence of menu items or categories.

Out of Scope:

During the brainstorming, we think of many other features, including payment and marketing analysis. Due to the relatively complicated implementation and time limitation, we decide to let customers go to the front counter to pay instead of paying online. Besides, the marketing analysis includes data collection and analysis, which is relatively time-consuming. Therefore, this part is out of our scope.

Epics

In this project, we have four main epics including basic, core , management and business analysis part. The structure picture is as following:



1. Basics

As basic part of our application, we will realise two main functions: administration and authentication for the restaurant side, menu presentation.

1. Menu presentation: show a customer a menu so that he/she can start an order

2. Authentication: the restaurant manager can sign up/in the system by using a social account or e-mail

2. Core

After defining the main structure, we will set up 4 main functions: dishes-to-cook for kitchen, place order for customer, dishes-to-serve for waiter, menu builder for manager.

1. Dishes-to-cook: For kitchen page, the kitchen staff can view a time-sorted list of customer orders that have come in and mark an order item as prepared.

2. Placing order: For the customer page, the customer can select their table numbers, browse through the menu, add items from the menu to their order.

3. Dishes-to-serve: For waiter page, the waiter can receive a notification when an order item is prepared and ready to be served, mark order items as complete once they have been served to the customer.

4. Menu builder: For manager page, the manager can add new categories to menu, add new menu items with titles, descriptions, ingredients, category, and cost to the menu, remove existing menu items from the menu, update the description, ingredient, category or cost to the menu.

3. Management

After finishing the main body of the system, this epic will add three functions: assistance request from customer, orders management and dashboard for manager.

1. Assistance request:

    – For customer page, the customer can request assistance from a waiter.

    – For waiter page, the waiter can receive a notification when a customer requests assistance, which should include information about the table number requesting assistance and mark requests for assistance as completed once they have dealt with.

2. Order management: For the manager, he/she can modify an order if the order is mis-placed by customers

3. Dashboard: For the manager, he/she can view the basic business condition of his/her restaurant, including the sales records and the profit.

4. Analysis

This epic will analyse the order data to arrange work schedule more effectively and reasonably and have a reasonable recruitment plan. For example, the system will give the manager advise that which dish sells best during the last season or which dish is becoming more and more popular.

Software architecture

Overview of this application

This high level of this application is quite straightforward. We use nginx to route the requests from browser to React server or Express server, and use axios to make the http request and handle the CRUD operations. MongoDB is used as the database to store all the data due to its convenience and non-relational property.

Workflow of Development and Deployment

- The development phase

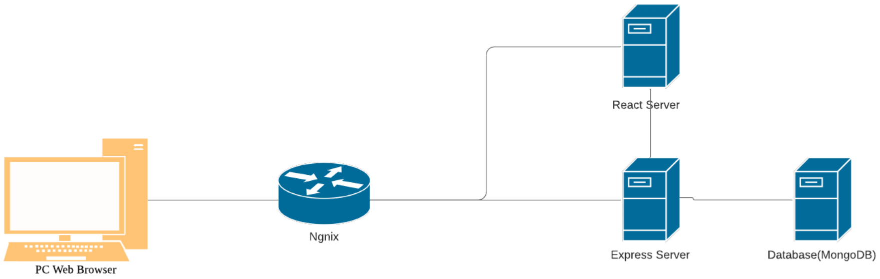
We will follow the industrial workflow using github to host all our codes and use pull request to push our work to the master to avoid our master being contaminated.

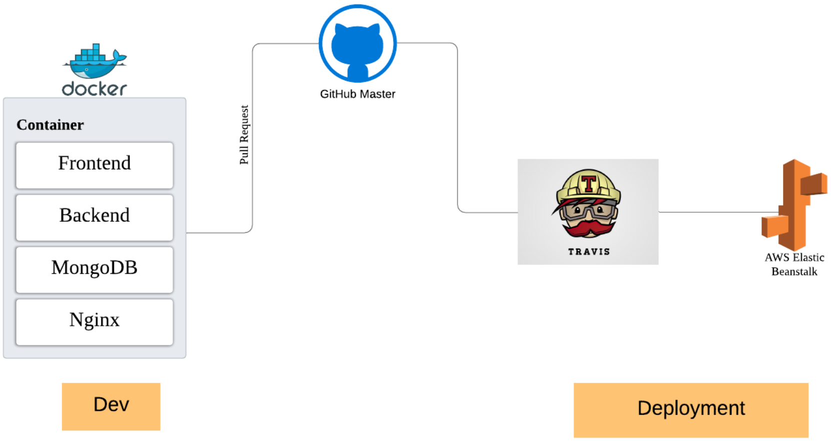
All coding work will be run in the docker image to ease the running of the developing environment.

- The deployment phase

This application will be hosted on AWS Elastic Beanstalk. We use travis CI to build projects and integrate service between github and AWS EB.

Sunshine Waiter





Service flows

The following picture illustrates the services.

Flow of an order getting fulfilled:

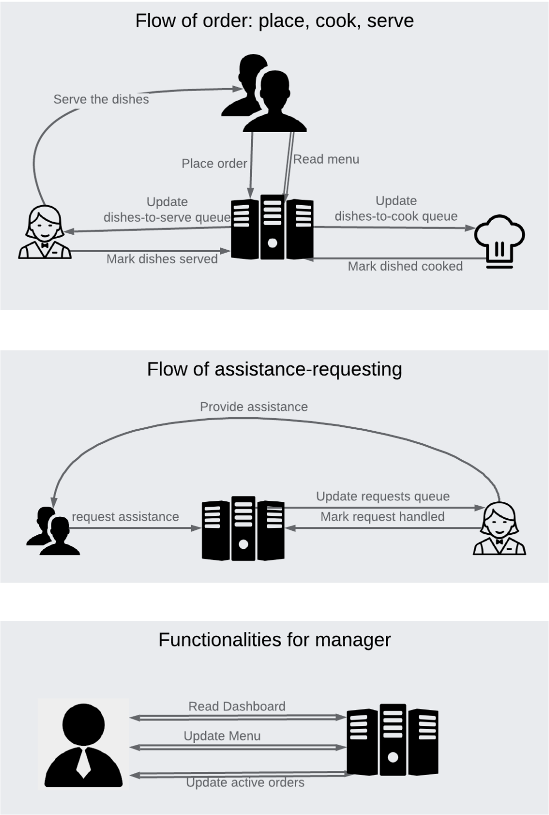
1. customer places order

2. server receives order

3. Server maintain the dishes-to-cook queue and push update to cook page

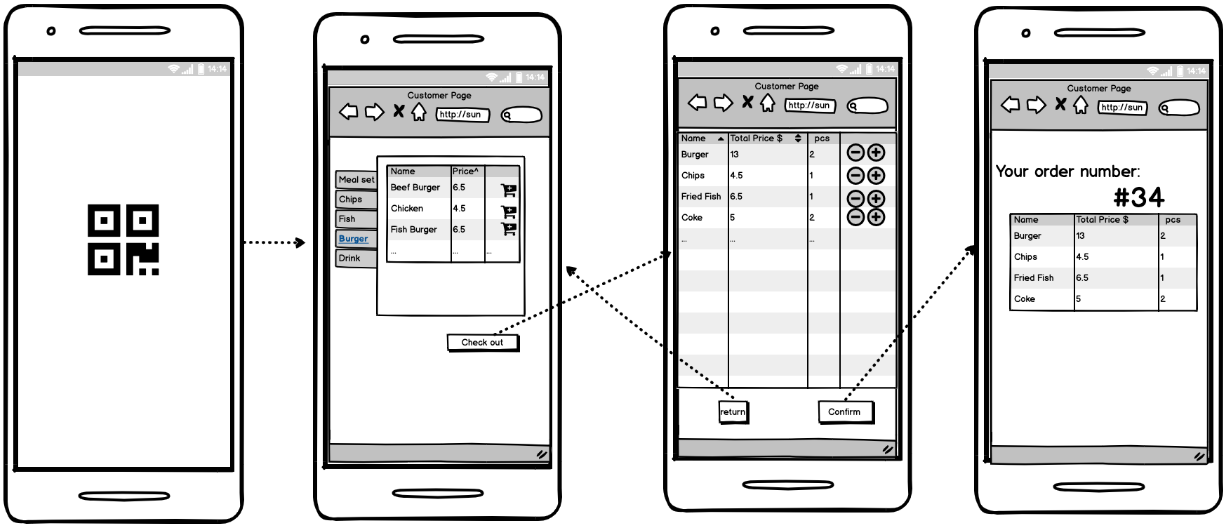
The dishes-to-cook is a time-sorted queue that cook can view and mark a dish as cooked

Sunshine-Waiter
restaurant order management system



1. cooks
   - cook dishes listed on dishes-to-cook queue
   - deliver them to ready-to-serve area
   - mark them as cooked on dishes-to-cook queue
2. Server receives dishes status update (getting cooked)
3. Server maintain the dishes-to-serve queue and push update to waiter page.

4. Waiters
   - serve dishes listed on dishes-to-serve queue
   - deliver them to corresponding customer
   - mark these dishes as served
5. Server keep tracking status of all dishes. When all the dishes in one order get served, this order would be marked as served.



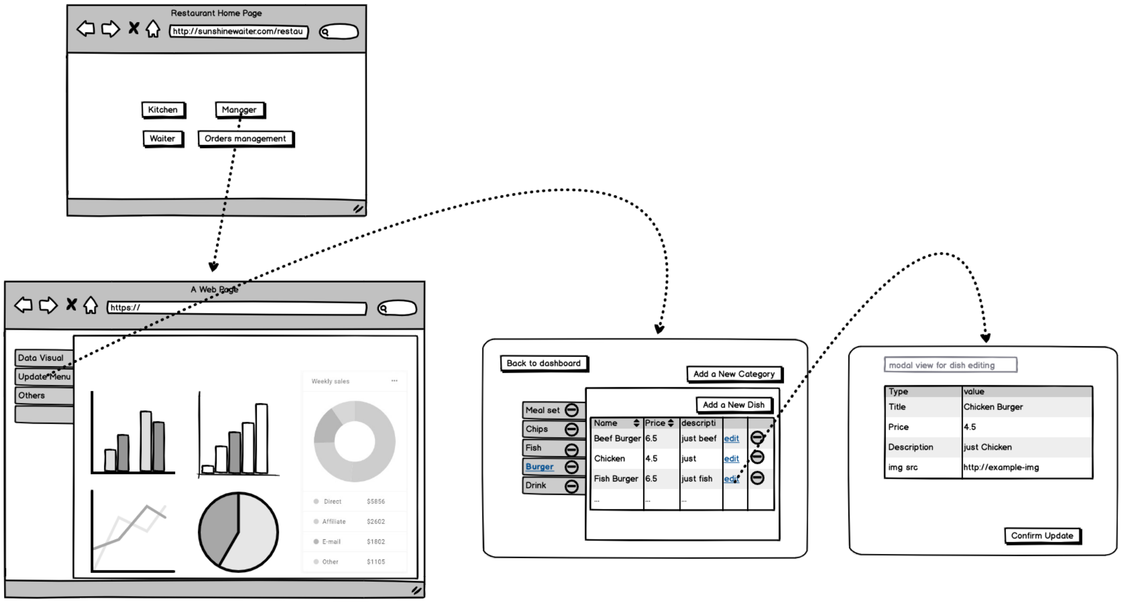
Name	Price^	
Beef Burger	6.5	
Chicken	4.5	
Fish Burger	6.5	
...	...	...

Name	Total Price $	pcs	
Burger	13	2	⊖⊕
Chips	4.5	1	⊖⊕
Fried Fish	6.5	1	⊖⊕
Coke	5	2	⊖⊕
...	...	...	

Your order number:

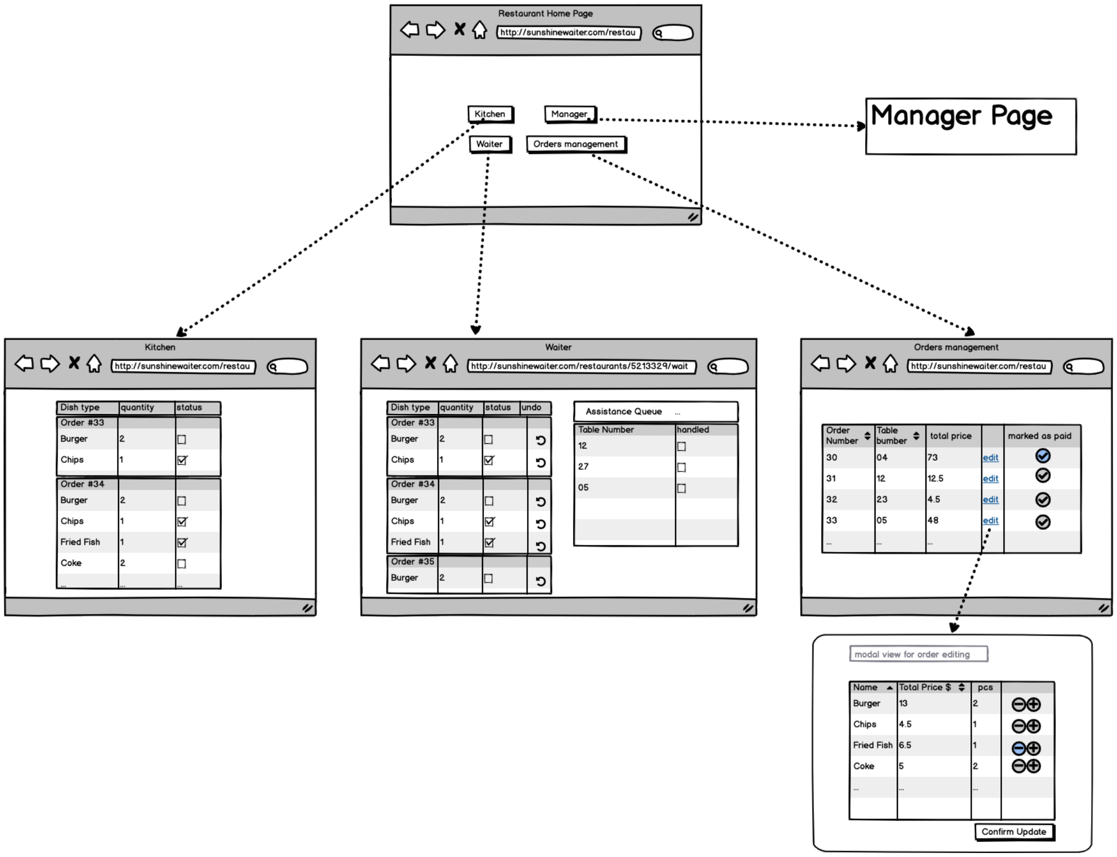
#34

Name	Total Price $	pcs
Burger	13	2
Chips	4.5	1
Fried Fish	6.5	1
Coke	5	2



Name	Price	descripti		
Beef Burger	6.5	just beef	edit	⊖
Chicken	4.5	just	edit	⊖
Fish Burger	6.5	just fish	edit	⊖
...	...	...		

Type	value
Title	Chicken Burger
Price	4.5
Description	just Chicken
img src	http://example-img


Restaurant Home Page
http://sunshinewaiter.com/restau
Kitchen
Manager
Waiter
Orders management
Manager Page
Kitchen
http://sunshinewaiter.com/restau
Waiter
http://sunshinewaiter.com/restaurants/5213329/wait
Assistance Queue ...
Orders management
http://sunshinewaiter.com/restau
modal view for order editing
Confirm Update

Dish type	quantity	status
Order #33		
Burger	2	□
Chips	1	☑
Order #34		
Burger	2	□
Chips	1	☑
Fried Fish	1	☑
Coke	2	□
...	...	...

Dish type	quantity	status	undo
Order #33			
Burger	2	□	↺
Chips	1	☑	↺
Order #34			
Burger	2	□	↺
Chips	1	☑	↺
Fried Fish	1	☑	↺
Order #35			
Burger	2	□	↺

Table Number	handled
12	□
27	□
05	□

Order Number	Table bumber	total price		marked as paid
30	04	73	edit	✓
31	12	12.5	edit	✓
32	23	4.5	edit	✓
33	05	48	edit	✓
...	...	...		

Name	Total Price $	pcs	
Burger	13	2	⊖⊕
Chips	4.5	1	⊖⊕
Fried Fish	6.5	1	⊖⊕
Coke	5	2	⊖⊕
...	...	...	

Technical stack

We decide to utilise these frameworks as tools:

- Frontend: React + Antd + Semantic UI
- Backend: Node.js + Express + MongoDB
- DevOps: AWS EB + Docker + Nginx

Sprints schedule

Sprints	Time	Activities
1	Week 3 - Week 4 (3 March - 15 March)	Complete Epic 1, including menu presentation and User authentication
2	Week 5 - Week 6 (16 March - 29 March)	Complete Epic 2, including the core functions of ordering and a menu builder
3	Week 7 - Week 8 (30 March - 12 April)	Complete Epic 3, including handling customers' requests and a dashboard page for manager
4	Week 9 - Week 10 (13 April - 26 April)	Complete Epic 4 and test the whole system. Create a demo and write the final report.