Sprint 1 Planning Document

The Flabbergasters

Michael Andrade

Aaron Fine

Abe Gunther

Matthew Lister

Sprint Overview

During this sprint we are going to utilize extreme programming and incremental prototyping to help mitigate the risks / challenges for this exercise. The goal, by the end of the sprint, is to have a horizontal prototype running on a physical server at the robotwaitress.tk domain space. We intend to have a multi-page navigable website unified by a global CSS. To better prepare for future increments all dummy data will be read into the html forms as JavaScript variables paving the way for the server to provide that data later.

Scrum master: Matthew Lister

Meeting Schedule (@330pm unless otherwise noted)

Feb 12: finish planning document

Feb 14: no meeting

Feb 16: no meeting

Feb 20: virtual machines up and running

Feb 21: first code review

Feb 23: second code review

Feb 26: final CSS pattern set

Feb 28: Practice demo / final configurations

Mar 2: Demo presentation

Risks and Challenges

Writing code is going to be an overwhelming challenge for this sprint. Our HTML, CSS, and JavaScript skill level varies from inexperienced to novice. To address this challenge, we are going to implement team programming (i.e. extreme programming, where two programmers work on one item at a time.). Furthermore, in order to increase consistency in both solution mechanisms and visual style we plan to hold multiple code reviews. Hopefully, extreme programming will result in a more uniform code-base for our team to standardize.

During this phase, building a working design environment for our programmers will also be a challenge. Each team will need a virtual machine running Django for testing purposes. Also building the physical server with matching specifications to virtual machines could be difficult. We believe that putting a multi-gigabyte .VHD file on GitHub would not be practical. However the configuration files for the Django server must be accessible to the team to modify and test, hence the need to have virtual machine.

Building incremental prototypes through extreme programming presents a challenge for our team in particular. The Fabbergasters have multiple commitments outside of school and scheduling shared time may be difficult. Lastly, this sprint will be delayed by the CS 3450 midterm exam, but hopefully the team will get a chance to both study and code together.

============================================== very ruff below ======================

to do list:

build server

install os

install python and Django

build Django hello world

start git with necessary Django files

build VMs with same config

as above but pull the git

map domain name to matt’s Comcast port 80;

five views

1. menu
2. admin \*needs forms that take fields as args for database manip functions
3. cook
4. inventory
5. review

5x build view

2x 5x code review

create master CSS

5x make data variables in JavaScript to fill html forms

USER stories: looks like this will address most user stories but with almost no functionality

=================================================================================

Backlog (this is everything from product backlog)

1. As a customer, I would like to be able create an account
2. As a customer, I would like to be able to log into or out of my account
3. As a customer, I would like to sort the menu in various ways
4. As a customer, I would like to post a rating and select a predefined statement for a particular dish, leave a custom review, or leave neither a statement or a review.
5. As a customer, I would like to pay a bill
6. As a customer, I would like to order with or without giving a name (would be given a table or location instead)
7. As a customer, I would like to add items to my order
8. As a customer, I would like to request service from a waiter / waitress
9. As a customer, I want to order something after I’ve paid my bill
10. As a customer, I want to be able to change my table / location at any time
11. As a customer, I want to be able to order something for another table / location
12. As a customer, I want to use a phone app compatible with android or ios with the same functionality as the website (As time allows -- Stretch goal)
13. As a customer, I want to request a order-ID
14. As a customer, I want to be able to view the menu without logging in
15. As a cook, I need to be able to select an open order
16. As a cook, I need to view a selected order
17. As a cook, I need to sort open orders by at least one criteria
18. As a cook, I need to indicate that an entire order has finished cooking
19. As a cook, I need to signal that an order is ready for pickup
20. As a cook, I may need to indicate that part of an order has finished cooking
21. As a cook, I need to signal the waiter / waitress that there is a problem or clarification needed with the order
22. As a cook, I need to be able to update inventory counts
23. As a cook, I need to be able to place no-cost meal orders (subject to daily limits)
24. As a cook, I need the arrival of new orders to be clearly signaled
25. As a cook, I need to be able to select sub-components of an overall order
26. As a cook, I need to be able to logout
27. As a cook, I need default prep orders available at the start of the workday (cut up 10 lbs of onions, make 20 gallons of soup, etc.) (If time allows - - Stretch Goal)
28. As a cook, I need default cleanup orders to come up at the end of the shift (clean grill, mop floor, etc.)(If time allows - - Stretch Goal)
29. As a cook, I need the system to interface with external buttons (hit a large button for next order, complete order, etc.)(If time allows - - Stretch Goal)
30. As a cook, I need the next open order to be automatically selected when I complete the current order
31. As a cook, I need to see how long it has been since an order was placed
32. As a cook, I need to see how much more time I have to complete the order to be within my target time (if the admin should be able to say that most order should be completed within 10 minutes of ordering then the cook should see that they only have 5 minutes left to complete the order)(If time allows - - Stretch Goal)
33. As a cook, I need to see how much overage time an order has taken beyond the target time (if the target time is 10 minutes and it took 11 min 20 seconds to complete the system should show -1 min 20 seconds for target time remaining)(If time allows - - Stretch Goal)
34. As a cook, I need to see how much time remaining until my next break or end of shift(If time allows - - Stretch Goal)
35. As a waiter / waitress I need to have the duration of login work as a timecard (If time allows - - Stretch Goal)
36. As a waiter / waitress I need to be able to issue a new customer order-ID
37. As a waiter / waitress I need to place an order for a customer
38. As a waiter / waitress I need to identify which table / location an order relates to
39. As a waiter / waitress I need to be able to pull up an ingredient list for menu items (If time allows - - Stretch Goal)
40. As a waiter / waitress I need to be able to place zero cost orders (subject to approval)
41. As a waiter / waitress I need to get notified when certain orders are available for pickup from the kitchen (i.e. I don’t care about orders not in my section)
42. As a waiter / waitress I need to be able to select or be assigned to a section of locations / tables
43. As a waiter / waitress I need to know what locations / tables have service requests
44. As a waiter / waitress, I would like to switch a bill from one customer to another
45. As a waiter / waitress, I would like to split or combine different bills
46. As a waiter / waitress / cook, I would like to view items that are marked not available
47. As a waiter / waitress / customer, I would like to view the daily special menu items(If time allows - - Stretch Goal)
48. As a manager, I would like to designate particular menu items as a daily special(If time allows - - Stretch Goal)
49. As an employee, I would like to log in and out
50. As a manager, I would like to manage other user accounts
51. As a manager, I would like to delete, create, and manage menu items
52. As a manager, I would like to delete, moderate, and approve user feedback before display to other users
53. As a manager, I would like to be able to manage inventory
54. As a manager, I need moderate payment requirements for some order types. The orders can be set such that the customer must pay before an order is placed, that the customer must pay after they eat but before they leave, or for net 30 (or other terms).(If time allows - - Stretch Goal)
55. As a manager, I need the system to interface with my accounting system (inventory, payroll, etc.)(If time allows - - Stretch Goal)
56. As a manager, I need the system to interface with third party order and delivery systems like ChowHound, PostMates, etc.(If time allows - - Stretch Goal)
57. As a manager, I need to be able to override details about an open order (menu items, price, location, etc.)
58. As a manager, I should be able to pull up a history of any order.
59. As a manager, I need a daily summary report on all activity at the restaurant
60. As a manager, I need an inventory status report
61. As a manager, I need a projected inventory report based on current inventory levels and historical data (I.E. do I have enough inventory for tomorrow?) (If time allows - - Stretch Goal)
62. As a manager, I need to be able to mark a menu item as currently unavailable
63. As a manager, I want to be able to see customer feedback in near real-time.
64. As a manager, I want customer feedback translated into sentiment. (If time allows - - Stretch Goal)
65. Upon first time login, users will be able to login and subscribe to receive notifications of their choice within 60 seconds of the application loading.
66. Application shall be available on the Google Play Store.(If time allows - - Stretch Goal)
67. The app must allow for at least 4+ simultaneous users that are pulling information from the database while others are creating and submitting events to the database at any given time.
68. The app must follow best practices to prevent sql injection, cross site scripting attacks or other common hacking techniques that can harm an application.
69. This project shall utilize a two tier Client/Server architecture. I.e. will not use proxy server
70. The project will have a logical design of hub and spokes.
71. All communications between the clients and server will be encrypted.
72. Aside from the guest/customer view, all user views will be password protected.
73. All views shall use a session cookie (life to not exceed 24 hours)
74. The server will validate all user requests.
75. Unique transaction IDs are generated for every transaction.
76. Party name on orders can be blank
77. Server will provide data to mobile client as JSON (If time allows - - Stretch Goal)
78. Mobile app will handle formatting and presentation of server data. (If time allows - - Stretch Goal)
79. Server will provide formatting through html/css to clients using a web browser.
80. Order and inventory data will be preserved for a user specified amount of time.
81. Changes to orders must be maintained for auditability.
82. Old records must be purged from the system after a user specified amount of time.
83. It should be configurable to allow orders through or not depending on inventory levels. I.E. if the inventory shows you are out of soup then it would depend on the configuration if a customer could order a bowl of soup.
84. The system will support access cards or other tokens for login instead of just username and password (If time allows - - Stretch Goal)
85. The source code will undergo static analysis
86. Within 30 seconds from time of order, new orders will show up in the cook view