Tyler Pearson, Jasmine Gutierrez, Mansi Chaubey
CS 440, Group 13
Professor John Bell
1 May 2021

Project Summary Final Release

As of May 1, 2021, Group 13 has completed the whole of the design document and finalized all details within. As follows will be the summary of the document and all relevant information related to sections three and beyond of the Eat With Us project.

Design

System Design and Software Architectures

Environment: Standard web designer client for page, Linux for server environment
Programming Language(s): HTML5/CSS, Javascript, SQL/MongoDB distributions
APIs: Left to the discretion of the development team

User Interface and Object Design

The user interface should be able to successfully:

- Perform account functions.
- Take work orders.
- Customize a user's home page.
- Send/receive messages.
- Modify user information.

Testing Plans

Features to be Tested and Their Pass/Fail Criteria

The items selected to be tested throughout the development process for Eat With Us include the account functions, including:

- Account creation and recovery functions
- Work order creation, including creating a work order for one service ordered once, one service ordered multiple times, many different services added to a work order once each, and many different services added to the work order multiple times
- Homepage customization and UI stability
- Security functions
- Work order handling
- User information handling and management
- Processing speed

The pass/fail criteria for each of these cases is simple: the functions perform within accepted bounds without issue and without problem, and that all security requirements are met with the testing as well.

Testing Materials and Schedules

The testing materials for the product are simple enough: there must be a device that can connect to the internet with a web browser, a database that the product must be able to access and interface with, and the other necessary adjacent systems required for the product.

In terms of scheduling, the first tests out the door will be with account functions, then the customization features, and then all features related to work orders and other services will be tested together. These tests are to be run multiple times throughout development, and run concurrently with all security testing.

Project Issues

Open Issues

The main two issues that the product will have to look out for are data protection and privacy laws that may come into play later in the product's life cycle and the status of "gig" workers as employees if delivery services are to be included into the product. The development team is expected to be able to account for these issues as they become apparent and make the product adaptable to them

Off-the-Shelf Solutions

There are multiple options for off the shelf products and solutions for developing the product and all its necessary functions. For server design a standard Linux server environment will be enough to handle the product's delivery and hosting, while services and products like MongoDB and the various SQL distributions will be sufficient for the database implementation. For building the product proper, standard web development libraries and languages will be used, such as Javascript. Though it is not necessary for the development team to do so, there are products like Google Firebase that can handle account creation/authentication/recovery.

Risks and Costs

It is expected that, assuming that this product is developed by a team of college students, that the product will cost an estimated $1,850 to develop, with each team member contributing one and a half hours a day on development. The main risks involved in this are going to be feature creep and inaccurate measurements and metrics.

Project Retrospective

Short bursts of work for writing the development document did not work as intended, and it produced more worry and stress for the whole group. Though this document was written during the CoViD-19 pandemic, communication was not as much of an issue as it could have been with the group using messaging services like Discord to communicate and "meet".

The other major issue that the team had encountered was scheduling the group meeting, as the group made them more spontaneous rather than planned out as it should have been. For future reference, it is better to have the meetings be scheduled for a specific time on a specific day each week to avoid having to do last minute planning.