Deliverable 1

Angelica Longo

Project #1: MealStats

Fall 2020 – COP 4331C

Contents of this Document

Concept of Operations

The Current System

MealStats is a modern web application for both restaurant workers and customers. All customers have the option to create an account, search dish categories, select different meal options, place orders, and make their payment all online. On the same application, workers will be able to update the status of each order that is placed, customers may easily view the status of their order anytime and can pick up their order once finished. Tax is automatically added to each order, and no delivery fee is needed.

The Proposed System: Motivation

Unlike other current restaurant systems, this system will provide a simple design with minimal features. This will result in a much faster and easier to use interface for both restaurant workers and customers. The system will be accessible to anyone in need of straightforward online ordering to make things as effortless as possible.

The Proposed System: Users and Modes of Operation

The users for this new system will be divided between workers and customers, were both types of users must make an account, but have different accessibility features.

Costumers will have the ability to make an account using their email and a password which will be saved into the system. While signed into their MealStats account, customers can search dish categories from the restaurant, select different meal options, place their orders, and make their payment. Once the customer has made a payment for the order, they will have the ability to view the status of their order in real-time, and see any updates made by restaurant works using the system.

All restaurant workers will have accounts built into the system so that they can view any receiving orders from customers and may update the status of any placed order. These status options can be viewed by the customer at any time, and the customer may pick up their order once their status has been updated to “Order Done”. When a customer has picked up their order from the restaurant, the worker will no longer have the ability to change the status of that order again.

The Proposed System: Operational Scenarios

Operational scenarios for customers include, registration using an email and password, searching through predefined dish categories (such as main dishes, appetizers, desserts, drinks), creating an order (by selecting dishes, drinks, desserts, etc), making an online payment for the order, and viewing any status updates on the order before pickup.

Operational scenarios for workers include, encrypted accounts to view any receiving orders from customers, and update options to mark on orders (such as “Received Order”, “Preparing Order”, “Order Done”, or “Picked Up”).

Faults that both restaurant workers and customers might experience are loss of internet connection or system failure. The web application system is required to work over an internet connection, and if either of these faults happens, refreshing the web browser and signing in again can be a solution.

A possible error that both restaurant workers and customers might experience would be logging into the MealStats system incorrectly. The restaurant system will ask for input during login, where an email and password is required. If any of the two specified fields are entered incorrectly, the application will state to try again and will not proceed further to the system until a recognized email and password is matched from the database. Once logged into MealStats, all inputs from users will be from dropdown menus so it is unlikely that other input errors will occur.

The Proposed System: Operational Features

MealStats must have features such as: login page, home page for customers to browse dish categories, payment selection for customers to submit their order, a special page that workers get redirected to once they sign in that shows all the orders that have been made, and an automatic tax fee that get charged to every order.

A possible feature that would be possible to have, is a “Sign in as Guest” feature, so that customers who do not want to register in the MealStats database, can simply place an order using their first and last name.

The Proposed System: Analysis

The MealStats system will be developed in the Java programming language using the Eclipse IDE for web developers, and the Apache Tomcat Java servlet container. This is due to the familiarity and previous experience working with this framework. The developing software is intended to be made available on any web browser across all platforms.

Creating the encrypted database aspect of system’s development will take up the most time in regard to the learning period, as well as understanding how to make the web application system available on all public browsers and not just the local testing machine. Since the MealStats system is meant to be a web application, the system is mobile and will require internet access (thus cannot be used in location where network access is blocked or limited).

The system making process that I am using builds the application from scratch, while other software may already have built-in tools and libraries. However, since I am not familiar with other web-developing programs, and since I am strongest in Java, I will stick to this selection of tools while learning during the process rather than use time to explore other alternatives.

Project Management Plan

Project Overview

MealStats is a simplistic web application for both restaurant workers and customers. All customers may create an account, search dish categories, select different meal options, place orders, and make their payment all online. On the same application, workers will be able to update the status of each order that is placed, customers may easily view the real-time status of their order and can pick up their order once finished. Tax is automatically added to each order, and no delivery fee is needed.

Applicable Standards

Coding standards used in the developing system include Javadoc format documentation, class names are nouns and interface names are adjectives, use of tabs for indentation, camel-case writing practice, brief commenting, exception handling, and other typical standards included in the published citation for Java Coding Conventions (https://www.oracle.com/technetwork/java/codeconventions-150003.pdf)

Regarding the artifact size metric standard, the project's software is planned to have 1000 lines of code, 10 classes, 5 input sources, and a single API.

Deliverables

Artifact	Due Dates
Individual Weekly Progress Reports	Weekly (Fridays) submission throughout the semester through Webcourses
Concept of Operations	9/18/2020
Software Project Management Plan (SPMP)	9/18/2020
Software Requirements Specification (SRS)	9/18/2020
Test Plan	10/30/2020
High-Level Design	10/30/2020
Detailed Design	10/30/2020
Test Results	11/30/2020
Source, Executable, Build Instructions	11/30/2020

Software Life Cycle Process

The Agile Method will be implemented during the duration of the project to ensure emphasis on flexibility in producing software quickly. This process will help to concentrate on responding to change rather than on creating a plan and then following it.

Tools and Computing Environment

MealStats will be developed in the Eclipse IDE for web developers within a Windows 64-bit operating system using the Java programming language. The Apache Tomcat Java servlet container tool will be utilized inside Eclipse for its ability to render web pages that use Java Server page coding.

Configuration Management

All visual control and configuration management will be personally handled by myself on GitHub. Personal backups of the system might be used to maintain private organization, but no other CM tool is currently being considered.

Quality Assurance

Weekly modifications will be done to any necessary portions of the project to ensure productivity and regular advancement. These procedures will personally be completed and acknowledged within each deliverable assignment. This includes citations on reports, documentation in the code itself, testing, and applying improvements to finished features.

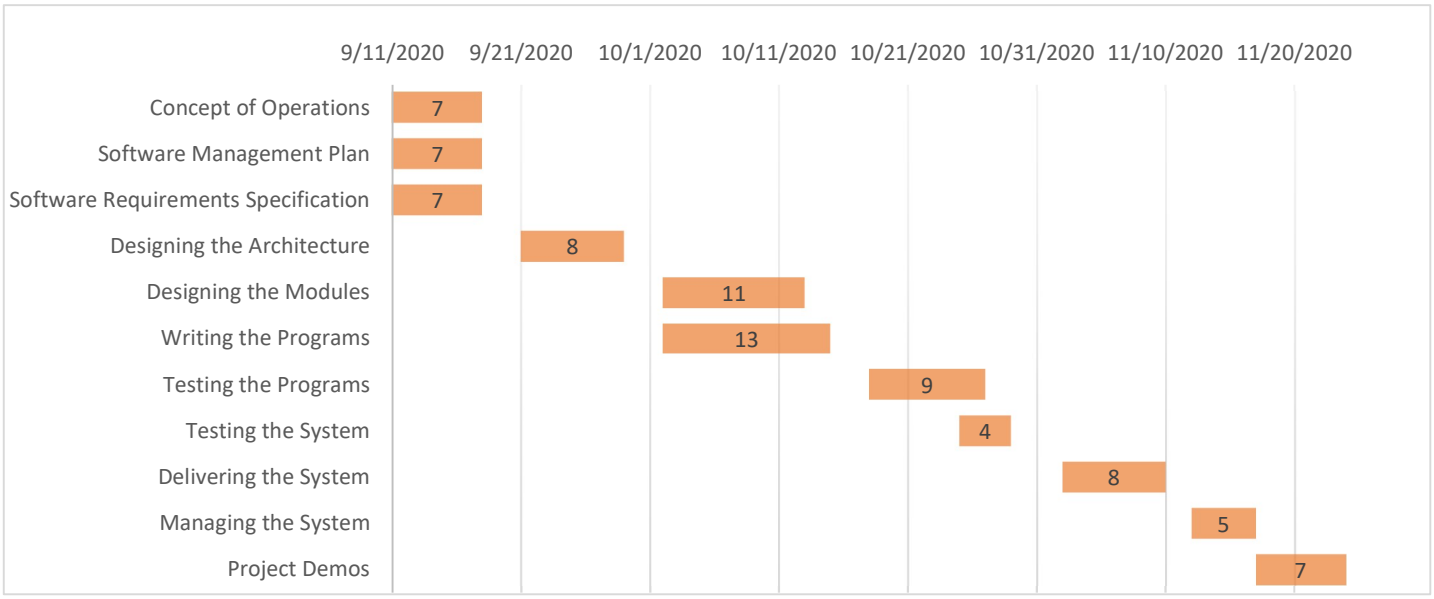
Risk Management

Potential generic risks that may affect the project schedule or resources consist of procrastination, possible illness, or misinterpretation of assigned requirements. These risks can mostly be handled and resolved by anticipating these risks and dealing with major requirements beforehand to prevent falling behind on schedule. Potential specific risks that may affect the quality or performance of the software being developed include failures within the database, IDE, or GitHub. Crashes that happen inside the IDE or GitHub can be resolved by having a back-up save, and crashes within the database can alert users of potential loss of data, but may not resolve the issue of the loss of data itself.

Table of Work Packages, Time Estimates, and Assignments

The software being developed can be broken down into a hierarchy of work packages for simplicity. A week may be needed to understand how to build an encrypted database, and another week to figure out how to make the web application system available on all public browsers and not just the local testing machine. Tasks such as making the webpages should take a day to achieve as long as they are kept simple. Each additional feature to the webpages can be expected to take a day and testing any advancements can take 2-3 hours.

GANTT Chart



Technical Progress Metrics

The planning phase of the project is projected to consist of 6 main requirements (create encrypted database of user accounts, users can create orders, create predefined dish categories, submit order button, workers can update status, and include automatic tax on every order). The entire software is expected to take up 20KB of memory, with a simple complexity of code.

Plan for tracking, control, and reporting of progress

Data that needs to be collected will be done so weekly through Webcourses inside of a project progress report. This report will include time spent in each activity, completed tasks, tasks in-progress, tasks planned for the following week, and individual issues and problems. In addition, the technical content of every entry will be examined to the date each week, as well as a reassessment of the potential project risks. If necessary, updates and adjustment will be made to align with estimated quality assurance results.

Software Requirements Specification

Introduction: Software to be Produced

MealStats is a simplistic web application for both restaurant workers and customers to view and update the real-time status of meal orders. Refer to Project Overview in the Project Management Plan section of this document for more information.

Introduction: Applicable Standards

View Applicable Standards in the Project Management Plan section of this document for more information on the coding standards cited for the project.

Definitions, Acronyms, and Abbreviations

None. This document is self-explanatory, and no acronyms or abbreviations will be used.

Product Overview: Assumption

Developers will assume that the system being developed will be able to function entirely on any other system/device that supports a web browser and will be able to operate across all platforms. The technological environment will also be assumed to perform in locations where internet access is not blocked or limited.

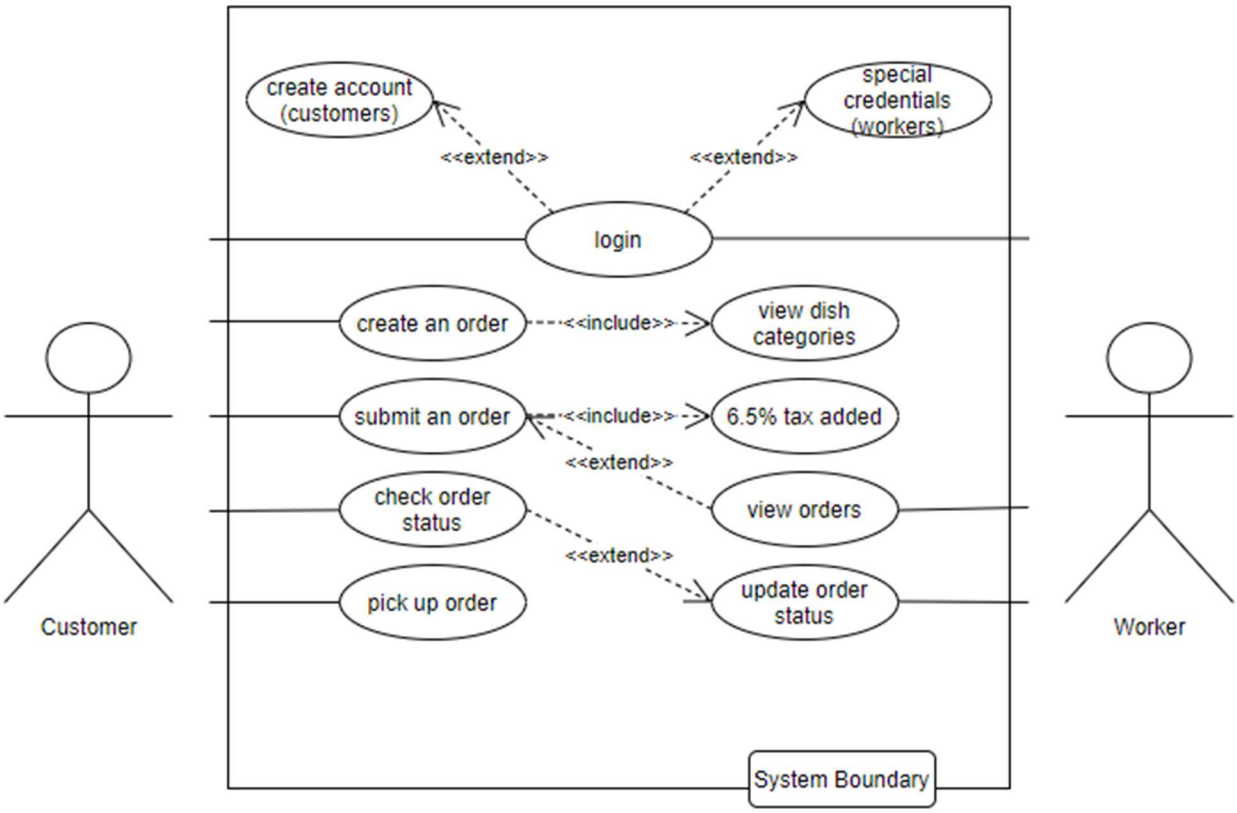
Product Overview: Stakeholders

Those who have interest in the software being built include restaurant workers, customers, and the system developer. Restaurant workers will take interest in the web app's ability to view customer's requested orders, as well as the "status updating" functionality to meal orders since this allows for production to be more efficient. The customers using MealStats will find significance in the ability to order their meals online and view the real-time status of their order until its ready to be picked up. The importance of this system to the developer is ensuring a product that is simple to understand, easy use, and makes the productivity of meal ordering increase.

Product Overview: Event Table

Event Name	External Stimuli	External Responses	Internal data and state
Login (for customers)	Enter email and password, click “login”	Redirects to restaurant homepage	Request gets processed and generates authentication token for user
View items on website	Click on drop down menu for dish categories.	Drop down menu includes inventory of predefined main dishes, appetizers, desserts, and drinks	N/A
Create order	Click on item(s) from the menu drop down list	Items selected will appear on the side of the webpage labeled as, “Your Order”	Prices of all items in the customer’s order will be added together
Submit order	Click “Make Payment” to complete the order	Redirects to a page that says, “Your order has completed” with a table at the bottom showing the order’s status	An automatic 6.5% tax is added to the order
Login (for workers)	Enter *special* ID and password, click “login”	Redirects to receiving orders page	Request gets processed and generates authentication token for user
Update order status	Click on dropdown menu from any of the receiving orders	Drop down menu includes status options of, “Received Order”, “Preparing Order”, “Order Done”, and “Picked Up”.	Updated status will also be viewable on the customer’s end

Product Overview: Use Case Diagram



Product Overview: Use Case Descriptions

- Customers can:
  - Login by registering accounts in the system
  - Create an order while viewing different dish categories
  - Submit their order (which will include an automatic 6.5% tax fee)
  - Check the status of their order(s) updated by worker
  - Pick up their order from restaurant
- Workers can:
  - Login with special credentials built into system
  - View orders submitted by customers
  - Update the status of customer's orders

Specific Requirements

No: FUNCT-1
Statement: Shall prompt user to login or create an account
Source: Client
Dependency: None
Conflicts: None
Supporting Materials: None
Evaluation Method: Once the customer logs in, their credentials are stored in the system's database and they should be redirected to the restaurant homepage. Once the worker logs in, they will be redirected to the "received orders" page.
Revision History: Myself, stated at beginning of project requirements, not yet edited
No: FUNCT-2
Statement: Shall provide customers with a webpage of dish categories
Source: Client
Dependency: None
Conflicts: None
Supporting Materials: None
Evaluation Method: Once customers can view this webpage, they should be able to interact with features displayed on page and being creating an order
Revision History: Myself, stated at beginning of project requirements, not yet edited
No: FUNCT-3
Statement: Shall provide workers with a list of submitted orders from customers
Source: Client
Dependency: None
Conflicts: None
Supporting Materials: None
Evaluation Method: Once workers are able to view this webpage, they should be able to interact with features displayed on page and being updating received orders
Revision History: Myself, stated at beginning of project requirements, not yet edited

No: PE-1
Statement: Shall run on all modern/smart devices which support internet access and contain a built-in web browser
Source: Client
Dependency: None
Conflicts: None
Supporting Materials: None
Evaluation Method: Most modern/smart devices support internet access and a built-in web browser. Will not be possible to use system otherwise
Revision History: Myself, stated at beginning of project requirements, not yet edited
No: UHF-1
Statement: Shall support users to are restaurant workers
Source: Client
Dependency: None
Conflicts: None
Supporting Materials: None
Evaluation Method: Restaurant workers will have accounts built into the MealStats system without having to register an email and password into the database
Revision History: Myself, stated at beginning of project requirements, not yet edited
No: UHF-2
Statement: Shall support users to are restaurant customers
Source: Client
Dependency: None
Conflicts: None
Supporting Materials: FUNCT-2
Evaluation Method: Restaurant customers will be able to create accounts, submit orders, and view the status of their orders
Revision History: Myself, stated at beginning of project requirements, not yet edited

No: DOC-1
Statement: Shall make use the of GitHub for visual control and configuration management
Source: Client
Dependency: None
Conflicts: None
Supporting Materials: None
Evaluation Method: Weekly revisions and reports are required to track progress
Revision History: Myself, stated at beginning of project requirements, not yet edited
No: DATA-1
Statement: Shall automatically add a 6.5% tax fee to every order submitted
Source: Client
Dependency: None
Conflicts: None
Supporting Materials: None
Evaluation Method: Every time a customer selects a dish, it gets added to the side of the webpage labeled as “Your Order”, and the prices of each item gets added up. The tax fee should then be calculated as 6.5% of the order’s total
Revision History: Myself, stated at beginning of project requirements, not yet edited
No: RES-1
Statement: Shall operate on any web browser
Source: Client
Dependency: None
Conflicts: None
Supporting Materials: None
Evaluation Method: The system being developed is a web application, and thus should be able to run on any web browser as a minimum requirement
Revision History: Myself, stated at beginning of project requirements, not yet edited

No: SECRTY-1
Statement: Shall maintain a backup of all the customer's and worker's credentials
Source: Client
Dependency: None
Conflicts: None
Supporting Materials: None
Evaluation Method: The database in the system will store customer's email and password, as well as the special identification and password for workers that is built into the system
Revision History: Myself, stated at beginning of project requirements, not yet edited
No: QA-1
Statement: Shall be personally documented which includes citations on reports, documentation in the code itself, testing, and applying improvements to finished features
Source: Client
Dependency: None
Conflicts: None
Supporting Materials: None
Evaluation Method: Weekly documentation will be done to ensure productivity and regular advancement
Revision History: Myself, stated at beginning of project requirements, not yet edited

Specific Requirements: Functional Requirements

The software must perform actions such as a login functionality, redirect customers and workers to two different webpages, and update the responses seen on the customer's page based on the input given by the worker.

Specific Requirements: Interface Requirements

Data being controlled in the system begin with identifying the user by accessing their credentials (given by an email and password). Other user given input is done via drop down menus to gather what the customer wants to order, or a different status the worker can change orders to. Each time the worker updates the status of an order, this input should immediately be visual on the customer's end so that the status data is close to real-time accuracy.

Specific Requirements: Physical Environment Requirements

Developing software is expected to run on any device that contains a web browser with supporting internet connection. Since the MealStats system is meant to be a web application, the system is mobile and will require internet access (thus cannot be used in location where network access is blocked or limited).

Specific Requirements: User and Human Factors Requirements

The system is projected to be used for restaurant customers and workers. Both types of users should be able to operate/navigate a web browser since the application itself is a web application. The software is intended to have a minimalistic design and have easy to use features that are aimed to make online ordering as effortless as possible. Special accommodations will be considered such as a font size that is larger enough for though who may be visually impaired, and less use of colors for those who may be colorblind.

Specific Requirements: Documentation Requirements

The documentation required for this system is personally handled by myself on GitHub to control configuration management.

Specific Requirements: Data Requirements

System is required to calculate 6.5% tax fee of the order's added total, and the degree of precision of this calculation will be rounded to two decimal places.

Specific Requirements: Resource Requirements

Besides being operational on a web browser, the resources for this software do not require skilled personnel, funding, special hardware, or specific tools.

Specific Requirements: Security Requirements

Customers that want to access the system for must do so by registering an email and password, which will then be stored on the system's database. Accounts for the workers will already be built into the system so registration with email is not required.

Specific Requirements: Quality Assurance Requirements

Weekly modifications will be done to any necessary portions of the project to ensure productivity and regular advancement. These procedures will personally be completed and acknowledged within each deliverable assignment. This includes citations on reports, documentation in the code itself, testing, and applying improvements to finished features.

Section 4: Supporting Material

None.