

LiveMenu

by

Luke Greeley

Hunter Holland

Tristen St. John

Submitted to

the Faculty of the School of Information Technology

in Partial Fulfillment of the Requirements for

the Degree of Bachelor of Science

in Information Technology/Cybersecurity

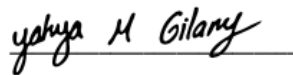
© Copyright 2022 Greeley, Holland, St. John

The author grants to the School of Information Technology permission
to reproduce and distribute copies of this document in whole or in part.



Luke Greeley

4/24/2022



Yahya Gilany

4/24/2022



Hunter Holland

4/24/2022



Tristen St. John

4/24/2022

University of Cincinnati

College of Education, Criminal Justice, and Human Services

April 2022

Table of Contents

Abstract	3
Senior Design Final Report	4
Project Summary:	4
Problem Statement:.....	4
Solution:.....	4
Project Source:.....	5
Project Objectives/Goals:.....	6
Project Scope:.....	7
Quick Project Timeline:	9
Technologies Used:.....	11
Technical Diagram:.....	11
User Personas:.....	13
Use Cases:	15
Use Case Diagram:.....	18
Testing Plan:.....	19
Test Logs and Procedures.....	21
Testing Review	21
Change Management Plan:.....	21
Budget:	22
Conclusion:.....	24
Figures:	25

Abstract

Through the rise of the COVID-19 pandemic in 2020, we also saw the rise of QR codes and other forms of touchless solutions. These solutions included menus, payments, ads, etc. We noticed that restaurant menus from displayed from these QR codes presented a plethora of issues. They typically displayed PDF versions of the company's physical menu. The problem with PDF menus is that they're not easily updated, leaving inaccurate menus. They're also not made for viewing on a mobile device. Users must pinch to zoom, scroll around, and often open multiple browser tabs. LiveMenu aims to solve these issues.

LiveMenu is a software solution for restaurants to display mobile-friendly interfaces to their customers, offering a user-friendly experience. These menus remain accurate thanks to the restaurant-facing mobile app, where management can easily toggle on and off items, as they're available/offered. If a menu item is no longer available, management simply turns it off and it is no longer viewable on the customer-facing website.

The result of this solution is even better than we expected. Both the customers' and the businesses' experiences are totally transformed through our service. Customers no longer have trouble navigating restaurant menus and no longer are confused about what is accurate on the menu. Businesses no longer have trouble updating their menus, and no longer must explain menu inaccuracies to their customers. Overall, this provides a better experience for everyone in the restaurant industry, whether their role is customer or staff.

Senior Design Final Report 2021-2022

Project Name: LiveMenu

Project Summary:

LiveMenu is a software solution to the service industry's mobile menu issues. This service will allow businesses to instantly edit their menus (food categories, different locations, etc.) and provide a clean user interface for their customers to interact with the menus.

Problem Statement:

QR code menus have become very popular over the past year with the rise of COVID-19 and the decrease in appeal for physical menus. In 2020, it was projected that, "an estimated 11 million households will scan a QR Code... in the USA alone, up from 9.6 million, according to Statista," (Sumner).

As many smaller businesses have opted for options that simply display a pdf version of the menu on a webpage, a variety of issues are presented:

- Most consumers in restaurants are visiting the site on their phones. Viewing PDF files on a mobile phone can be difficult to read and navigate, presenting a negative user experience for the business's customers.
- These menus are also difficult to update. you would first have to update the PDF, then upload to a hosting service. In the brewery space, specifically, menus can change often, sometimes daily.
- Mobile Menus could also lead to mobile ordering, "According to the National Restaurant Association, three in five U.S. consumers order delivery or takeout at least once a week," (Resendes).
- According to Statista, roughly 90-100% of people aged 64 or younger own a smartphone, and these rates have been inclining steadily over the past eight years (Figure 1).

Solution:

This issue could be solved by a service that:

- Provides a multi-platform application for the business to instantly update their menu, add and delete multiple locations and menu items/categories.
 - This will be offered in mobile and web for variety of uses.
 - Business owner/management will be able to:
 - Edit active menu that displays to customers
 - Add/Remove/Edit menu items
 - Add/Remove/Edit menu categories
 - Add/Remove/Edit locations
- Provides a mobile-friendly web interface to display the menu

- Customers scan QR code stickers to launch the web page and can navigate the menu with more ease than PDFs.
- There will be a segment bar at the top of the menu to select different categories (Food, Beers, Cocktails, etc.)
- Once a segment is selected, the customer can scroll through the menu, reading the name, description, price, etc. of each menu item.
- QR Codes will be location-specific (Main location, secondary location, mobile van/food truck, etc.), populating different menus.
- Eliminates the need for pdf menus on a business's website.
 - PDFs are not mobile friendly, as the user needs to zoom and scroll in all directions.

Project Source:

I originally thought of this idea from working at a local brewery. Throughout the pandemic, I noticed other breweries, bars, restaurants, etc. transition to the mobile device menus. My company did not for some reason. As I noticed the shortcomings with most mobile menu systems, it began to look like a hassle (especially in the beer industry, where menus can change frequently, sometimes even daily).

- Luke Greeley

Hunter was the mutual connection in the group, having prior known both Tristen and Luke. Tristen and Hunter were partners, and Luke asked Hunter to join.

Project Objectives/Goals:

We hope this application will provide an easier experience for both restaurant management and consumers by implementing:

- Active Menu Web App
 - Provide more space for menu items than the typical menu can provide
 - Provide a contactless menu system
 - Customers scan QR Code to open menu
 - Reduce time and confusion for consumer to navigate the menu
 - Mobile-friendly web app is easier to navigate than the issues that arise with mobile PDFs, like:
 - Multiple tabs being opened
 - Scrolling/zooming
 - Only displaying certain sections of the menu
 - This endpoint returns the current menu to display on the customers' web app
 - Includes:
 - Categories (Food, Beer, Cocktails, etc.)
 - Sub-categories (Under food: Appetizers, Sandwiches, Sides, etc.)
 - Menu Items:
 - Header (Item Name)
 - Price
 - Sub header (style, like different styles of beer)
 - Sub-category 1 (Short info piece, like ABV)
 - Sub-category 2 (Short info piece, like IBU)
 - Description (paragraph describing the menu item)
- Admin Application
 - Cross Platform (mobile & web)
 - Provides ability to reorder and change menu items of active menu
 - Ability to add/delete menu items permanently
 - Ability to add/delete locations
 - Ability to add/delete menu categories
 - Removes hassle of editing/uploading PDF menu
 - Reducing time to edit the menu
 - Plan to make a scalable project so we can add more restaurants in the future

Project Scope:

Our team will develop a functional application that enables restaurant businesses to update/edit their menus easily and quickly. This service will also provide a better user experience for the business' customers by providing a scannable code that will show a clean, modern user interface on their mobile device.

This project will consist of a backend REST API service, and two client-side UI projects (one for the business and one to display the menu to the customers). We will also have to implement other services like database, authentication, hosting, etc.

Business Client Overview:

Cross-Platform Application

- Functions Include:
 - Display Full Menu
 - Both active and non-active items for a particular location
 - Display Active Menu
 - This is what the customer sees in customer client for a particular location
 - Editable in business client
 - Add Menu Item
 - Allows business client to add a menu item with the following details:
 - header
 - subHeader
 - price
 - subCategory1
 - subCategory2
 - description
 - category
 - Remove Menu Item
 - Allows business client to permanently remove a menu item from the list
 - Edit Menu Item
 - Allows business client to update a current menu item's details
 - Activate Menu Item
 - Allows business client to add menu item from full menu to active menu
 - Toggles "ON" in business client
 - De-Activate Menu Item
 - Allows business client to remove menu item from active menu
 - Toggles "OFF" in business client
 - Add Category
 - Allows business client to add categories (Food, Drinks, Dessert, etc.)
 - Holds Menu Items
 - Delete Category

- Allows business client to permanently delete a category and its menu items
- Add Location
 - Allows business client to add locations for trucks/vans or other storefronts
- Delete Location
 - Allows business client to permanently delete a location and its menu
- Change Location
 - Allows business client to choose a location to edit the menu for
 - Allows for different QR codes to display different active menus in the customer client
- Style Guide
 - Endpoint returns all css styles for front end for a particular business
 - Aids in scalability

Customer Client Overview:

Mobile-Friendly Web Application

- Provides a segment at the top of the page to choose category (Food, Drinks, Desserts, etc.)
- Provides a search bar with typeahead functionality to allow customers to search for their favorite menu items by name
- Provides a clean UI to display the active menu set by the business client for a particular location
- Sample URL with response:
 - northernrow.livemenu.com/1
 - northernrow: business
 - livemenu.com: our hosted web app
 - /1: location
 - Response gives back the active menu for location 1 of Northern Row

Quick Project Timeline:

A rough estimate of your project timeline in a tabular format with the following fields:

Task #	Task Name	Duration	Start Date	End Date
1.	Team Contract	3 weeks	8/30/21	9/27/21
2.	Create Projects & have hard-coded customer menu -customer client	2 weeks	9/27/21	10/4/21
3.	Display active customer menu dynamically -customer client	2 weeks	10/4/21	10/18/21
4.	Create Admin page showing full menu -business client	2 weeks	10/18/21	11/1/21
5.	Ability to toggle on/off menu items -business client	2 weeks	11/1/21	11/15/21
6.	Ability to change/add/delete locations & show respective menu -business client	2 weeks	11/15/21	11/29/21
7.	Ability to add/remove menu items -business client	2 weeks	11/29/21	12/6/21
8.	Ability to edit menu items -business client	2 weeks	1/10/21	1/24/21
9.	Ability to change styling dynamically Allows for scaling -business & customer client	2 weeks	1/24/21	2/7/21
10.	Authentication & Adding Menu Items/Location in bulk -business client	2 weeks	2/7/21	2/21/21

11.	Build for iOS, Android, and Web/ensure project is working across all platforms	2 weeks	2/21/21	3/7/21
12.	Final issues resolved and cleanup	2 weeks	3/7/21	3/21/21
13.	Cleanup/Presentation Preparation	2 weeks	3/21/21	4/5/21
14.	Final Project Wrap-up & Expo Prep	1 week	4/5/21	4/12/21

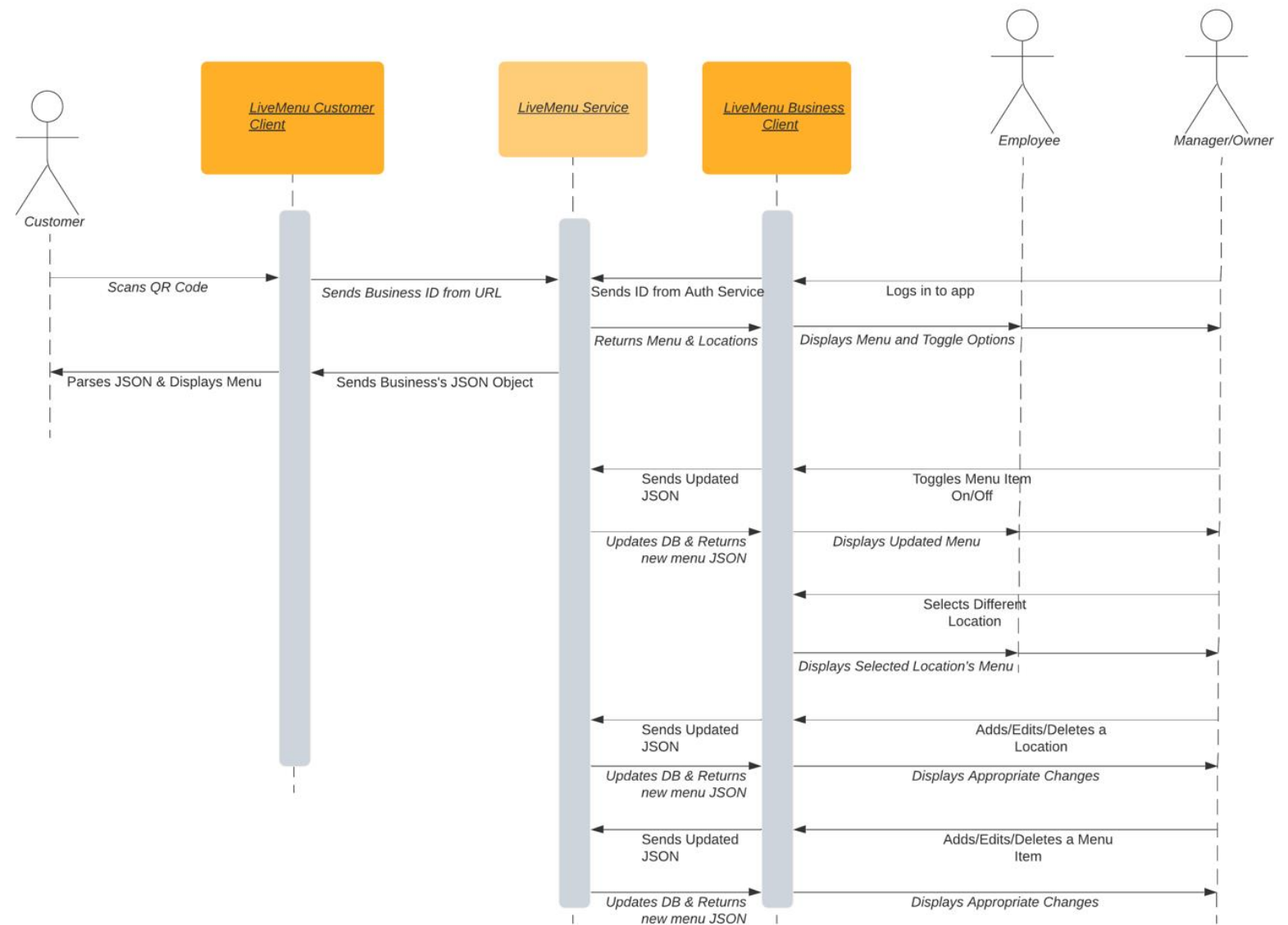
Technologies Used:

We decided to use a tech stack which may not be very conventional but works for our group. For the frontend, we are using Ionic which is built upon Angular. Ionic is a great framework because it allows us to use one tool to build the user interface. By using Ionic, we can create an iOS, Android, and web application with the same code. Ionic streamlines development and makes sure that all our bases are covered for our users and their devices. For our backend, we are using FastAPI. This API framework is very new but is simple to use and understand. Our team has familiarity with Python, so using Python for our backend was a no-brainer. FastAPI, as its name would imply, is very fast. Speed is a huge priority for our team because the last thing we want is to have users disgruntled and hungry. For our database, we are using MongoDB. Our team had not used a NoSQL database solution before, but we were looking for something that was extremely fast and flexible. MongoDB checks both of those boxes for us which is crucial to our project. When it comes to Continuous Integration and Continuous Deployment, our team is leveraging the power of GitHub actions. Our project repo is in GitHub, and there is a plethora of actions set up to accommodate our Azure hosting. Azure is our cloud solution of choice because our team is very familiar with Azure. It is also affordable because it lets us use all their cloud technology for 12 months or \$200, whichever comes first. In Azure, we will run our application as an Azure App Service. To secure our application, we are using Azure Authentication with Azure Active Directory—otherwise known as Azure AD. When hosting an Azure App Service, Azure makes it easy to secure the web address from the outside world using Azure AD.

Technical Diagram:

Below is our technical diagram. This diagram lays out the different actions associated with the work that goes on behind the scenes. From the customer perspective, there is only one action. The customer will simply scan a QR code, which will pass the business's ID to our service. Our service will then query the database and return the business's menu in a mobile-friendly interface for viewing on the customer's device. The business's experience has a few more functions.

When management logs in, they will also see the current menu, but this also shows menu items that are not active. Staff has the option to toggle these menu items on and off, which will send an update to our database, affecting the customers' menu in real time. Staff members also have the option to view and edit other locations the business might operate, like other storefronts, mobile trucks/vans, and popup events. Our service updates the database with a whole new menu for the business, with a new menu ID. This allows for different QR codes at these locations, so the customers pass the corresponding ID to our service when they scan the code. Staff can also add/edit/delete menu items, which updates the database to keep menu items accurate.



User Personas:

Table 1 User Persona Table

User Persona: Restaurant Customer	
	Customer
	Janet
	54
	Female
Behavior	Enters restaurant and sits down, wants to view the menu.
Pain	Gets confused when navigating a restaurant's PDF menu system on her mobile phone. Pinching and zooming, multiple tabs, etc. are all annoyances to her.
Needs & Goals	Janet wants an easier way of navigating the mobile menu

User Persona: Restaurant Customer	
	Customer
	Susan
	24
	Female
Behavior	Susan is a frequent customer at her local brewery, she knows what she wants when she visits
Pain	Often times she orders her favorite beer to be told it's not currently offered, even though it's on the menu.
Needs & Goals	Needs to see an accurate menu to see if her favorite beer is currently on tap. She wants to be able to quickly search the name of it and view it on the menu.

User Persona: Restaurant Owner/Manager	
	Manager
	Jon
	35
	Male
Behavior	Manages restaurant's customer experience and staff behavior
Pain	Updating the menu daily and uploading the new PDF is time consuming and annoying for Jon. If a keg blows, for example, there is no easy way of removing it from the menu, leading to a false menu for the customers.
Needs & Goals	Jon wants an easy menu management system that allows for editing the menu on the fly, and he wants an easy-to-use interface for both his customers and staff

User Persona: Restaurant Staff	
	Bartender
	Kyle
	26
	Male
Behavior	Works as part of restaurant/bar staff. Is responsible for interacting with customers and upkeep of the bar. Kyle will have access to the application, but with less permissions than the owner/manager.
Pain	Kyle wants to be able to easily update the menu if something changes throughout the night. He gets multiple questions from customers about menu items that aren't currently offered and is asked how to navigate the current pdf menu on mobile devices.
Needs & Goals	Kyle wants to be able to update the menu on the fly and reduce the amount of time spent helping customers navigate the menu and explaining what the bar is not currently serving.

Use Cases:

Table 2 Use Case Table

Use Case ID	001
Use Case Name	Customer views menu
End Objective	Provide a visually pleasant, easy-to-navigate, mobile friendly menu for the customer
User/Actor	Restaurant Customer
Trigger	Customer enters restaurant and is seated
Frequency of Use	Every customer upon every visit, daily
Preconditions	Customer has a smart phone Customer wants to view menu
Basic Flow	Customer is seated Customer scans QR code on table with their smartphone Customer is brought to a mobile friendly web app that shows an accurate menu of the restaurant/bar and can easily navigate the web page, and even search the menu
Alternate Flow	Customer can also view the restaurant's menu remotely, by visiting the website directly to bring up the menu.
Postconditions	Customer can continue to refer to the menu or close the browser tab when they're finished viewing the menu

Use Case ID	002
Use Case Name	Pull an item off the menu
End Objective	Instantly remove the menu item from the menu, no longer showing to the customer. This provides a real-time, accurate menu for the restaurant's patrons.
User/Actor	Restaurant Staff/Manager
Trigger	A keg blows, the kitchen runs out of chicken, etc. The restaurant runs out of stock of a particular item, or item is simply out of season or removed for other reasons
Frequency of Use	Daily
Preconditions	There is an active menu with an item that is no longer available.
Basic Flow	Restaurant staff logs into mobile app, finds desired menu item, and slides the toggle to the "off" position. After refresh, or any new instance of the customer-facing web app will no longer display the menu item.
Alternate Flow	Staff could also log in to the web app on any browser and follow the same steps as above.

Postconditions	Menu item still appears in business-facing application, just simply in the “off” state. Item can be added to the menu again by sliding the toggle to the “on” position.
----------------	---

Use Case ID	003
Use Case Name	Restaurant adds/deletes menu item
End Objective	Easily add or delete menu items from a restaurant’s menu
User/Actor	Restaurant Manager
Trigger	A new menu item is offered by the restaurant.
Frequency of Use	Whenever a menu item is new or is no longer offered by the restaurant.
Preconditions	Restaurant is already using our system and offers a new menu item or wants to delete an old one.
Basic Flow	Restaurant offers a new menu item. Manager logs into app. Manager clicks “add” button. Manager enters details of menu item. Manager clicks “save”
Alternate Flow	Manager can do this on any device, provided correct permissions.
Postconditions	Menu item is added to business-facing application and can be toggled “on” at any time to display in customer-facing web app.

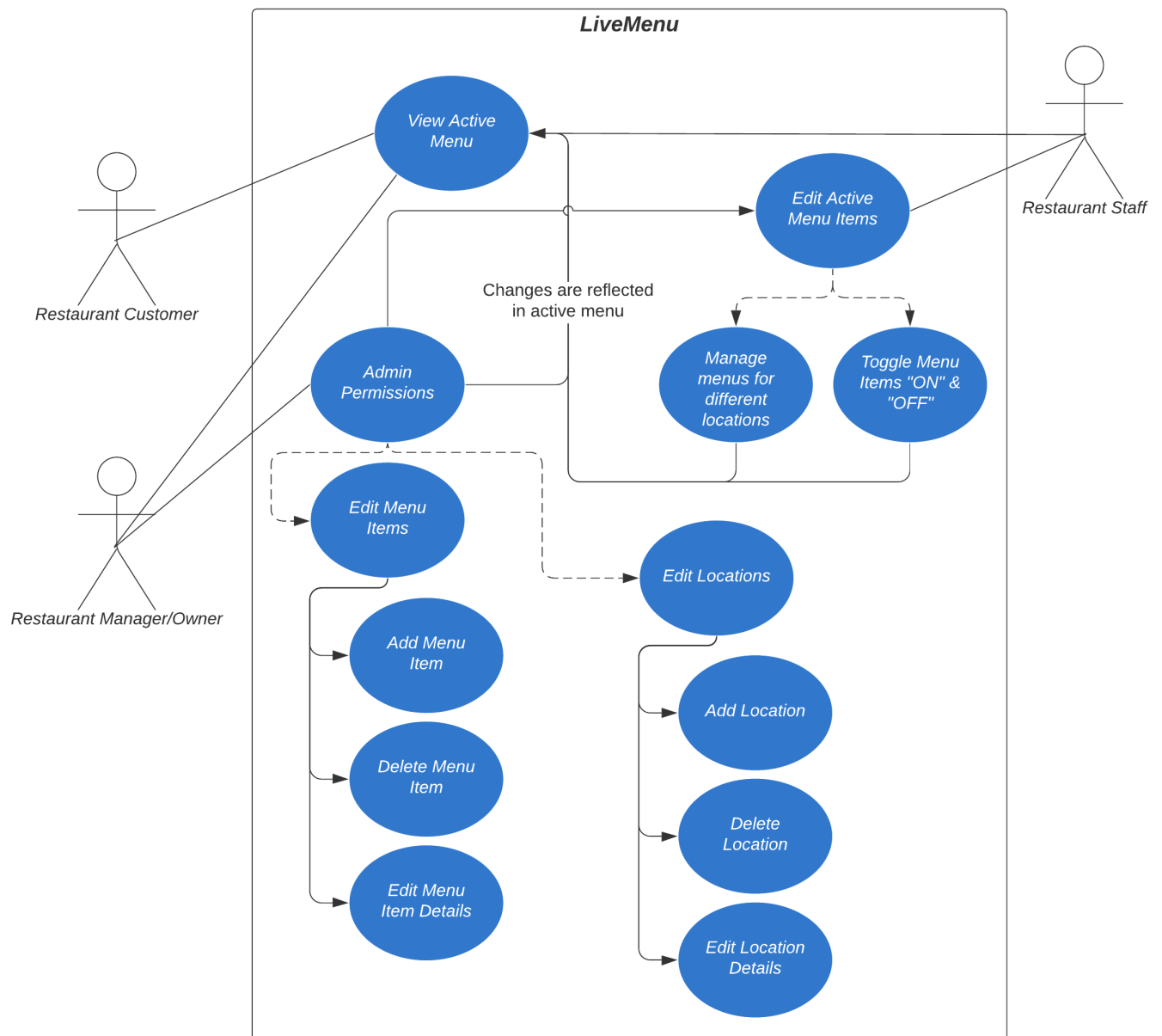
Use Case ID	004
Use Case Name	Restaurant changes details of a menu item
End Objective	Provide an easy way to instantly change the description, price, name, or other details of a menu item
User/Actor	Restaurant Manager
Trigger	Menu item details are changed
Frequency of Use	Whenever an item’s details change, possibly daily.
Preconditions	The menu item exists in current menu User has write access to menu items
Basic Flow	Example: A new batch of a particular beer has a higher abv than previous batches. Manager logs into app. Manager finds desired menu item. Manager clicks on desired menu item and is brought to a details page. Manager clicks “Edit” and can edit the details of said menu item.

Alternate Flow	Manager can do this on any device, provided correct rights.
Postconditions	Menu item changes are instantly reflected upon refresh or new instance of customer-facing web app.

Use Case ID	005
Use Case Name	Business acquires new location/mobile van
End Objective	Add/remove business locations for multi-location management
User/Actor	Restaurant Manager
Trigger	Restaurant opens a new location or mobile van/truck, which needs a menu.
Frequency of Use	Upon acquisition or deletion of restaurant location.
Preconditions	Restaurant is already using our system and acquires a new location
Basic Flow	Example: Bar buys a van to use as a mobile bar at external events. Manager logs into app. Manager finds the “locations” button. Manager clicks the “edit” button, where options to delete other locations, and “add” button will appear. Manager clicks “add” Manager can now add location name, categories, and menu items to the menu for that location.
Alternate Flow	Manager can do this on any device, provided correct rights. Manager cannot delete location if there is only one location in the list.
Postconditions	New location’s customer-facing menu is now active at a different endpoint, provided by different QR codes to display at said location.

Use Case Diagram:

Below is our use case diagram. This diagram shows the different actions users can take and the results they will see. Customers only have the option to view the active menu. The active menu is the list of menu items and categories that the restaurant staff has toggled "On." The restaurant staff has a few more options, depending on their role. Basic staff members can edit which menu items are active for any location within the business. Their actions will affect the menu displayed to the customer. Restaurant management can do all this, plus the ability to edit menu items and locations. This means they can add and delete menu items or locations, as well as change their details, like descriptions, prices, etc.



Testing Plan:

Overview

In this section, you will find an overview of our plans for testing, testing scope, objectives, logs/procedures, review, as well as our change management plan. We plan to implement both unit testing and user testing. Unit testing both the frontend and the backend will ensure our project works as expected at the code level. User testing will ensure our application is user-friendly and easily navigated/used.

Methodology

Our testing throughout the project lifecycle includes unit testing, along with continuous integration (CI/CD) through GitHub Actions. In the current GitHub Actions pipeline, a docker container must be built and deployed to Azure. If there are any failures in the pipeline, nothing will be deployed to Azure. While deploying to Azure, we will push the docker image to Azure Container Registry. After the image has been pushed, this image will be used by Azure Container Instances, and be made available publicly. Throughout development, user testing will be done by the team internally to ensure the project is working as expected, with external user testing taking place once development is done.

Unit testing is a huge part of the development lifecycle, and we want to make sure that our tests are being run during the deployment process to guarantee that our application is working as expected. It is very important to make sure that we have code coverage, so that there is no guesswork as to what is working or will continue working when we are deploying new features.

User testing should include giving people access to our customer facing client, as well as our business facing client. We want to ensure things such as styling, navigation, and user input work as expected. Users will be able to view a completed menu, as well as switch to the business facing client to edit said menu and return to the customer client to view their changes. We also want to ensure users have a seamless experience across platforms, as these projects will all be used as one complete service. Below you can find a list of features to be tested by our users.

Scope

Unit Testing Scope:

1. Test the database connection.
 - a. We must make sure that our API has a connection to the database in order to make changes to menu items.
2. Test the connection to the API.
 - a. When deploying, we should make sure that our back end is available and reachable for our front end. Without a connection from the UI to the API, users will be unable to make changes to the database.
3. Test expected response codes.

- a. We will send test responses to the API to make sure we are getting back the correct response code. Having this check will allow us to make sure users working with our platform have enough information when our API does not accept their response body.

User Testing Features:

1. Users can log in and appropriate merchant's menu will be shown.
 - a. We want our project to be scalable, meaning any business can sign up for our service and build mobile menus.
2. Menu is displayed when QR code is scanned.
 - a. This means the QR code includes correct endpoints and merchant id's to be passed. If a QR code is scanned and no menu is available, we want to show an error screen (Page not found).
3. Menu is easily navigated, and all components work (search function, scrolling, styling, etc.)
 - a. Ensuring things like search function works allows our users to use the service to its full capabilities. We wouldn't want to put out a service that feels slow or glitches out when users try to click on different buttons or user input is needed.
 - b. Business client functions work (toggle on/off, switching locations, etc.)
 - c. Menus are matching - business client's items toggled "ON" match the menu being displayed in the customer client, and those that are "OFF" are not.
 - d. Our service is accessible to those who may be visibly or physically impaired. We want to ensure our service is usable by everyone who has access to the internet and is an enjoyable experience.

Objectives

1. All methods must have effective unit testing procedures.
 - a. This ensures all endpoints are accounted for and are tested to make sure they are working properly.
2. All use cases must account for all user roles.
 - a. Some permissions in our project will allow for different actions that can be taken. We must ensure only admin users can perform admin actions, and other roles cannot.
3. All bugs need to be resolved before the IT Expo.

- a. While there may be incomplete sections of the project at the Expo, we want to ensure there are no bugs that cause the app to crash during presentation. This will allow for a smooth presentation to show our project and its potential.

Test Logs and Procedures

Item #	Test Case #	User	Role	Expected Output	Actual Output	Pass/Fail	Reason for Failure/Success	Date
1.	1	Customer	Customer	Scans QR code and sees menu	Cans QR code and sees menu	Pass	Hosted through Azure, customer can see menu	3/2/2022
2.	2	Employee	Employee	Log in and see editable menu	Log in and see editable menu	Pass	Hosted through TestFlight, app can be downloaded and used	3/10/2022
3.	3	Employee	Employee	Toggle menu item on/off	Can toggle menu item on/off	Pass	Hosted through TestFlight, mobile app allows for editing menu	3/10/2022

Testing Review

User testing has begun through our hosted site and our beta application through Apple's TestFlight. Testers can download our application [here](#) on iOS devices to test the functionality of our service. A sample business's menu can be viewed [here](#) (Northern Row Brewery and Distillery is used for this example). We learned a lot about the hosting and deployment process through setup for user testing. Allowing people to test our application has been beneficial in learning the expected behavior of the application.

Change Management Plan:

- Any team member can request a change at any time.
 - Team will then discuss the request together and move forward.
 - Team will identify pros/cons of the change and assign a priority to it.
 - Priority will be classified into three levels of severity and changes will be labeled as such:
 - Low
 - Medium
 - High

- We will communicate these changes with our stakeholders at scheduled meetings and schedule unplanned meetings if we feel it needs to be discussed more urgently.
- When implementing any new changes, a member of our team must open a pull request.
 - This request should include comments documenting the changes at a high level.
 - This request should be approved by at least one team member other than the member who submitted the changes.
 - Upon approval of these changes, our team will merge with master.

Budget:

Our estimated budget and return on investment are shown in Table 3. We decided to charge per request that is sent through the customer client (\$0.05 per request). In the model, we're estimating an average of 200 requests per day, for a business that is open 360 days out of the year. We're also assuming we have 5 businesses using our service in year 1.

Development Time and laptops used were the primary costs in this project, as we are using free services, such as hosting and authentication.

Table 3 Project Budget

Estimate of Benefits											
1 year's estimated revenue (use the following formula: $((\text{rate} * \text{customers}) * (\text{days open}) * (\# \text{ of clients}))$ using the variables shown below):	\$ 18,000.00										
<table> <tr> <th colspan="2">Revenue Formula</th></tr> <tr> <td>Rate per Request</td><td>\$ 0.05</td></tr> <tr> <td>Customers/Day (avg.)</td><td>200</td></tr> <tr> <td>Days Open</td><td>360</td></tr> <tr> <td># of clients</td><td>5</td></tr> </table>		Revenue Formula		Rate per Request	\$ 0.05	Customers/Day (avg.)	200	Days Open	360	# of clients	5
Revenue Formula											
Rate per Request	\$ 0.05										
Customers/Day (avg.)	200										
Days Open	360										
# of clients	5										
Select other benefits the project may bring a customer or user:											
Improved customer satisfaction											
Increased system availability											
Productivity or process improvement											
Estimated Cost Rough Order of Magnitude:											

	Rate Per/Hr	Work Effort (Hours)	1 X Costs
Labor - Database/Security	20	480	\$ 9,600.00
Labor - Developer 1	20	480	\$ 9,600.00
Labor - Developer 2	20	480	\$ 9,600.00
Macbook 13"			\$ 1,299.00
Macbook 15"			\$ 1,499.00
Macbook 15"			\$ 1,499.00
Misc.			
TOTAL			\$ 33,097.00

Ongoing Annual

	Rate Per/Hr	Work Effort (Hours)	1 X Support Cost
	20	52	\$ 1,040.00
	20	52	\$ 1,040.00
	20	52	\$ 1,040.00
TOTAL			\$ 3,120.00

5-Year ROI Analysis

Description	5- Year Expected	Conservative (1.5)
Total Costs	\$ 48,697.00	\$ 73,045.50
Total Benefit	\$ 90,000.00	\$45,000
Total Costs/Benefit Differential	\$ 41,303.00	\$ (28,045.50)

Conclusion:

Lessons Learned

Lessons we've learned throughout this process include:

Communication is key. Work cannot get done in an effective way without effective and constant communication. We've found that we're far more productive during weeks where we keep in contact with each other.

Learning is part of the challenge. We've all learned so much about each other's strength and comfort zones of development and project infrastructure, and we've been able to teach each other a what we know about the respective languages/technologies.

Skills/Abilities

As stated above, we've been able to teach each other our skills and abilities and further enhance them as well. Luke knew the front end going into this (Angular & Ionic). Hunter was comfortable with the backend (Python & FAST API). Tristen was comfortable with MongoDB and security measures we need to ensure are taking place. Overall, we've all helped each other stay mindful of all these different aspects of the project, as some may be missing without each other's knowledge. From a development standpoint, we've enhanced our skills by building this project from scratch. In the workplace, we're typically working in a repository that's been previously established and maintained, meaning we have other people's code to work from and refer to. Here, we're building from scratch and need to do our own research in order to problem solve.

Signatures

Luke Greeley

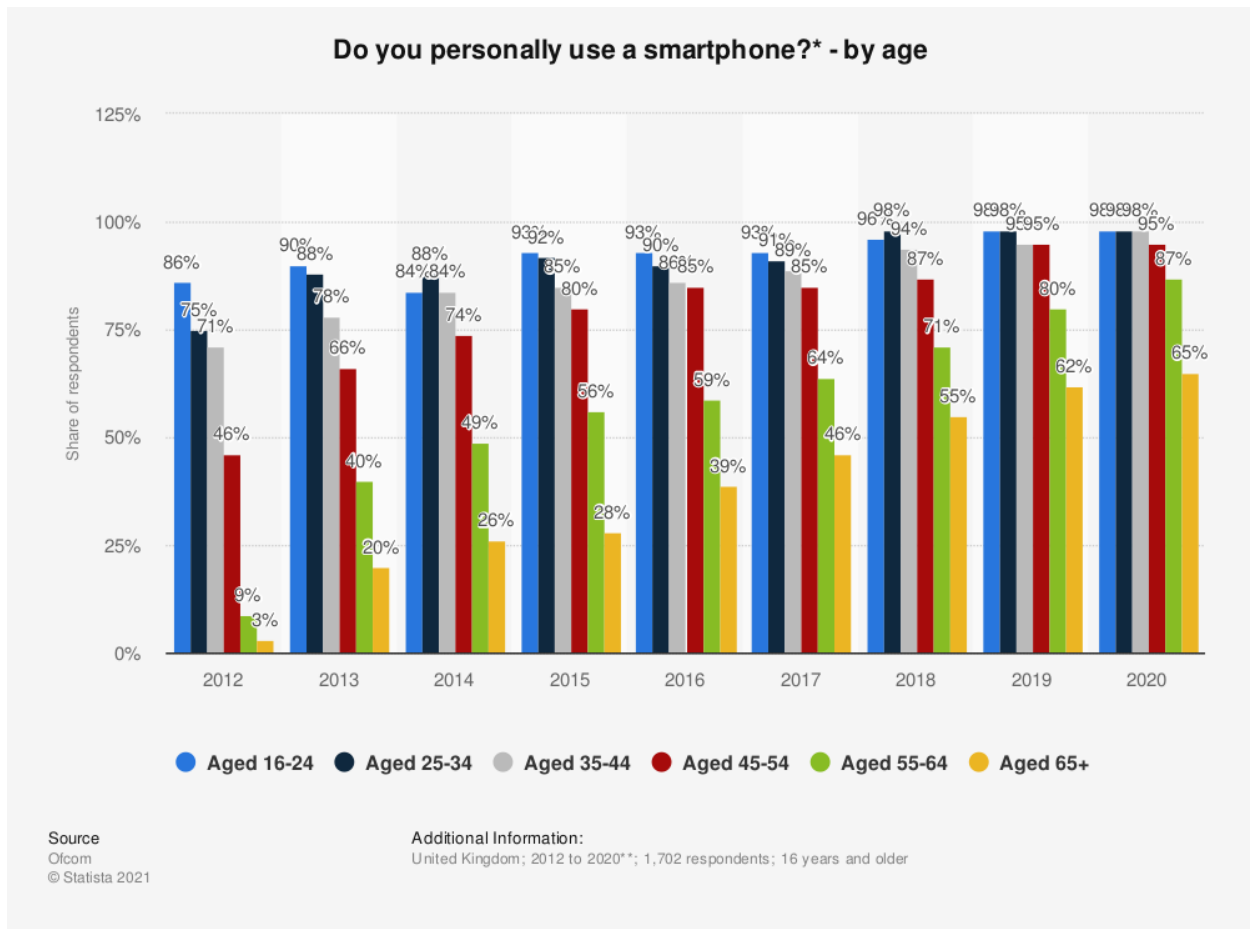
Yahya Gilany

Hunter Holland

Tristen St. John

Figures:

Figure 1



References

- Published by S. O'Dea, & 15, J. (2021, June 15). Smartphone usage by AGE UK 2012-2020. Statista. Retrieved September 10, 2021, from <https://www.statista.com/statistics/300402/smartphone-usage-in-the-uk-by-age/>.
- Resendes, S. (2021, May 12). 2020 restaurant INDUSTRY Trends: Shaping the future of food. Restaurant Insider. Retrieved September 8, 2021, from <https://upserve.com/restaurant-insider/restaurant-industry-trends/>.
- Sumner, T. (2021, May 19). The Post-Covid rise of QR codes in US retail and the retailers that use them - Perch - retail Marketing LIFT-AND-LEARN digital Signage, kiosks, and in-store analytics software. Perch. Retrieved September 7, 2021, from <https://www.perchinteractive.com/retail-marketing-blog/2020/8/1/qr-codes-retail-covid-contactless>.