GOATS

Development of a Cloud-Based Enterprise Resource Planning System for Gompei's Goat Cheese

Christopher Guerrette, Natalie Mohn

Development of a Cloud-Based Enterprise Resource Planning System for Gompei's Goat Cheese

A Major Qualifying Project Submitted to the Faculty of WORCESTER POLYTECHNIC INSTITUTE in partial fulfillment of the requirements for the Degree of Bachelor of Science

By:

Christopher Guerrette
BS / MS Computer Science

Natalie Mohn
BS Management Information Systems, MS Innovation with User Experience

Date: April 28, 2022

Submitted to:

Gompei's Goat Cheese, Unofficial Sponsor

James E. Ryan, MIS Advisor

Therese Smith, CS Advisor

Abstract

Gompei’s Goat Cheese (GGC) is a non-profit, student-run business at WPI that supports student scholarships and teaches students how to apply entrepreneurial skills. The goal of our project was to begin development of a cloud-based enterprise resource planning system for GGC so that they may improve and grow operations. From interviews with key stakeholders, we developed a set of system requirements that fit the specific needs of GGC. The foundation we created has been designed to support GGC with benefits such as improved transparency, centralized order and invoice tracking, and more effective communication with their partner farm.

Acknowledgements

First, we would like to thank Gompei's Goat Cheese, our unofficial 'official' sponsor. As alumni of GGC, we are invested in the future of GGC. We are proud of how much this company has grown and how much they have endured in the last four years. We wish GGC the best and know we will see them thrive in the coming years.

Next, we would like to thank the interviewees from the Gompei's Goat Cheese team and Westfield Farm for taking time out of their busy schedules to let us interview them. The feedback they provided was instrumental to the success of this project.

We are also incredibly indebted to the previous MQP groups that started GGC in 2013 and 2014. Thank you to Jeremy Berman, Danielle Payne, Joseph Botelho and Rodrigo Calles; these dedicated students created the foundation for this thriving student-run business and for the future of student-run entrepreneurial endeavors at WPI. Without them, none of this would have been possible. We are also excited to congratulate the founders and the current GGC team; GGC is approaching its ten year anniversary on April 18, 2023.

Finally, we would like to thank our advisors, Professor Jim Ryan and Professor Therese Smith, for supporting us through this experimental MQP. We are grateful for your guidance and for your positivity throughout this past year.

Executive Summary

Small businesses are the backbone of the nation's economy. According to the Small Business Administration (2021), small businesses account for 99.9% of all U.S. employer firms. Not only are small businesses everywhere, but they are also important contributors to their local communities. However, as important and beneficial as these small businesses may be, not all of them will be able to succeed.

In data collected by the Small Business Administration, about a third of small businesses fail within the first two years of their existence (Small, 2021).

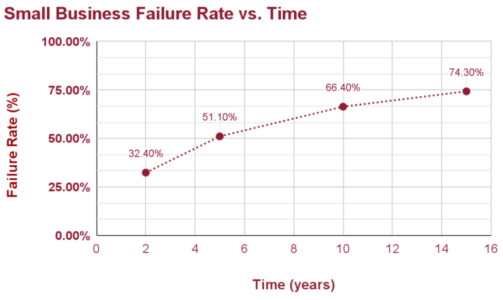


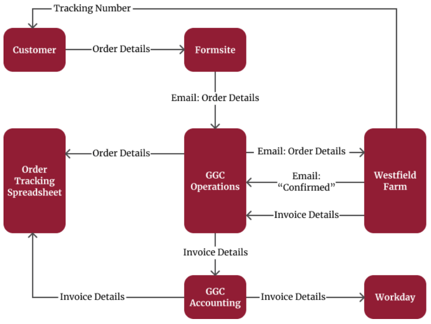
Figure i. Failure Rates of Small Businesses Over Time (Data from (Small, 2021)).

So, how does a small business stay afloat? Robert Normand attributes a portion of the success of small businesses to Operational Support Systems. One example of an Operational Support System is an Enterprise Resource Planning System (ERP).

Gompei's Goat Cheese (GGC) is a non-profit, student-run business at WPI. They sell award winning goat cheese produced by Westfield Farm in Hubbardston, MA, and donate all proceeds to the WPI Global Scholarship Program, which supports students studying abroad.

Figure ii. The GGC Team - A Term 2021.

The existing operations structure of GGC created harmful inconsistencies between GGC's records of orders and invoices and the farm's records, which impacted the fulfillment of orders and contributed to limiting the growth of GGC. It became clear that GGC was in need of a new order and invoice management system, along with improved communication with the farm.



Pain Points to Note:

- No shared order reference number between GGC and Westfield Farm
- Communication with the farm only occurs through the operations team
- Customer places order with GGC but gets a tracking number from Westfield Farm
- Emails sent to the farm have been lost (no email reply with "confirmed")

Figure iii. Data Flow Diagram (Level 0) of GGC's Current Operations System.

Project Goal and Objectives

The goal of our project was to begin development of a cloud-based enterprise resource planning system for GGC so that they may improve and grow operations.

Our team accomplished this goal by following a methodology that implements the fundamental four phase Systems Development Life Cycle (SLDC). Due to the condensed timeline of this project, we decided to use the system prototyping methodology. This methodology performs the analysis, design, and implementation phases concurrently in order to quickly develop a simplified version of the proposed system.

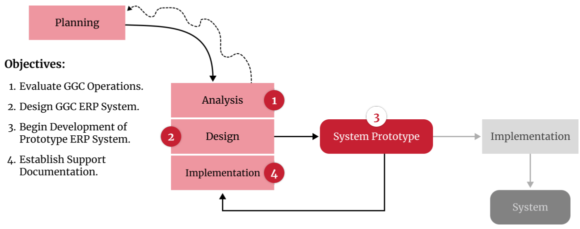

Planning
Objectives:
1. Evaluate GGC Operations.
2. Design GGC ERP System.
3. Begin Development of Prototype ERP System.
4. Establish Support Documentation.
Analysis
1
2
Design
Implementation
4
3
System Prototype
Implementation
System

Figure iv. System Prototyping Methodology and Project Objectives.

1. **Evaluate GGC Operations.** We interviewed GGC and Westfield Farm members to gather feedback on the current operations system and ideas and needs for the new system. Next, we compiled interview feedback and created a set of system requirements to follow when designing the new system.

2. **Design GGC ERP System.** To design the new system for GGC, we compiled critical system documentation such as System Use-Cases, Data Flow Diagrams, an Entity Relationship Diagram, a Data Dictionary, a User Flow Diagram, and System Mockups.

3. **Begin Development of Prototype ERP System.** We created the foundation for the prototype within AWS's Amplify Studio, using DynamoDB to develop the back-end and React.js to produce the front-end.

4. **Establish Support Documentation.** Along with this report, we created additional documentation for GGC, Westfield Farm, and future student groups so that they can continue development and implementation of this new system.

Key Interview Findings

- Interaction with the current operations system ranges from a few times a semester to nearly every day.
- Team members prefer to use their computer, rather than their phone, when using the current operations system.
- The current system is easy to use, and everyone has access to it.
- Westfield Farm and GGC use different order numbering systems.
- Inputting information into the current system takes too long, and there's a lot of redundancy in the work that the operations team does.
- Westfield Farm handles Gompei's Goat Cheese order's differently than its normal orders.

System Requirements

The following requirements have been summarized to include the most crucial requirements of the new system.

Business Requirements

- Store all order, invoice, product, and customer information
- View number of active orders and invoices and where they are in the system
- Westfield Farm and GGC both use the system

User Requirements

- Input and modify order, invoice information
- Search for orders and invoices via reference number, invoice number, or customer last name

Functional Requirements

- Store, modify, and recall information
- Query for an order based on reference number, invoice number, or customer last name
- Generate an invoice when an order is added

Nonfunctional Requirements

- Desktop application
- Multi-user capability
- Secure user login and data storage
- User friendly
- Incorporate GGC's brand identity

System Requirements

- Develop database using Amplify Studio within AWS
- Develop front-end using React.js

System Design

List of Use-Cases

- Add User, Remove User
- Log In, Log Out
- Record New Order, Delete Order
- Update Order Information
- Update Invoice Information
- View Active Orders
- View Completed Orders
- View Order Information
- View Customers
- Search Orders

For each use case, we noted the participating actors, entry conditions, exit criteria and flow of events. The full use-case documentation can be found in Appendix D.

Data Flow Diagrams

One data flow diagram (DFD) we created was another level zero data flow diagram (DFD). This DFD helps show how the new system would reduce redundancy by centralizing order and other data.

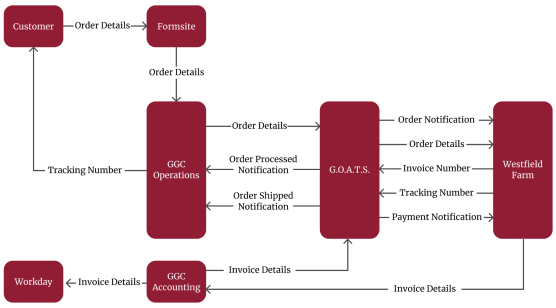


Figure v. Data Flow Diagram (Level 0) AKA Context Diagram.

The next diagram created was a level one DFD. This takes the level zero DFD and breaks it up into entities, processes, and data stores. These reflect the data flow of critical functionalities of the system, such as adding a new order (Process 1.0) and updating order details (Process 1.1).

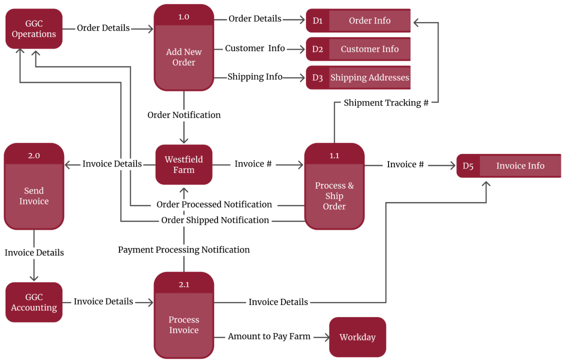


Figure vi. Data Flow Diagram (Level 1).

Logical Entity Relationship Diagram

Although we ended up using a NoSQL database for the initial prototype, we also wanted to document the logical entity relationship diagram (ERD).

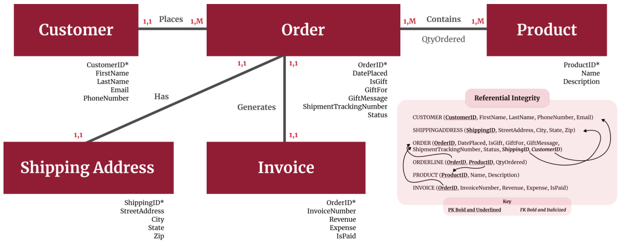
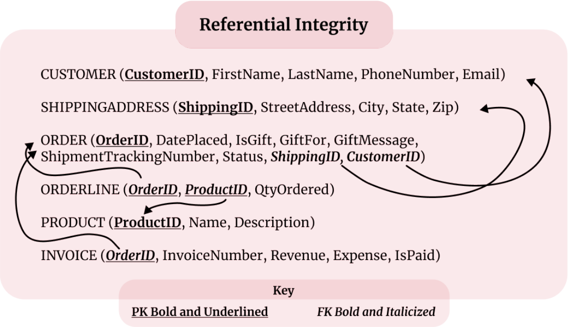


Figure vii. Entity Relationship Diagram.



Referential Integrity

CUSTOMER (**CustomerID**, FirstName, LastName, PhoneNumber, Email)

SHIPPINGADDRESS (**ShippingID**, StreetAddress, City, State, Zip)

ORDER (**OrderID**, DatePlaced, IsGift, GiftFor, GiftMessage, ShipmentTrackingNumber, Status, ***ShippingID***, ***CustomerID***)

ORDERLINE (***OrderID***, ***ProductID***, QtyOrdered)

PRODUCT (**ProductID**, Name, Description)

INVOICE (***OrderID***, InvoiceNumber, Revenue, Expense, IsPaid)

Key

PK Bold and Underlined ***FK Bold and Italicized***

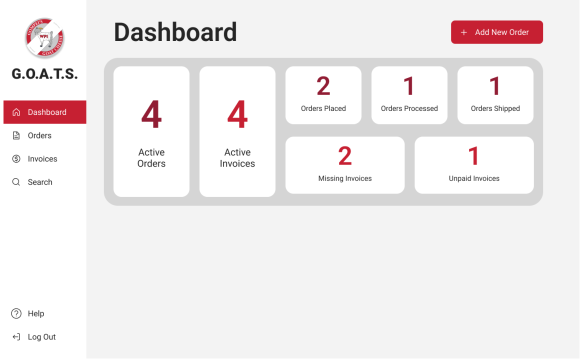
Figure viii. Referential Integrity (Magnified).

Along with the ERD, we created a data dictionary that reflects the tables that would be expected in a relational database. However, this also defines all of the details for the attributes in each table reflected in the current database schema. This documentation can be found in the System Design & Development chapter.

System Mockups & Hi-fidelity Prototype

Before developing system mockups, we created a user flow diagram (UFD) to help figure out the flow of the system before diving into designing the UI. The UFD can be found in the System Design & Development chapter.

The following mockup screenshots were designed and prototyped in Figma to become the hi-fidelity prototype of the new system. More mockups can be found in Appendix E.



Dashboard

4 Active Orders

4 Active Invoices

2 Orders Placed

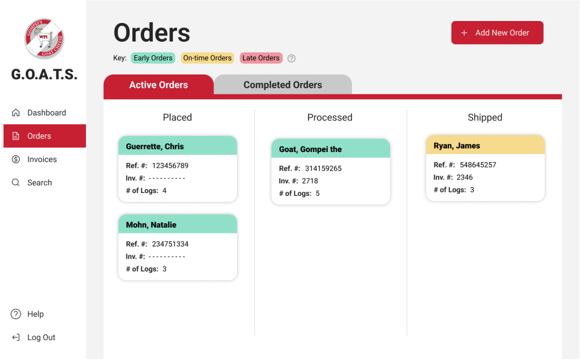
1 Orders Processed

1 Orders Shipped

2 Missing Invoices

1 Unpaid Invoices

Figure ix. Dashboard Mockup.



Orders

Placed

Guerrette, Chris
Ref. #: 123456789
Inv. #: ----------
# of Logs: 4

Mohn, Natalie
Ref. #: 234751334
Inv. #: ----------
# of Logs: 3

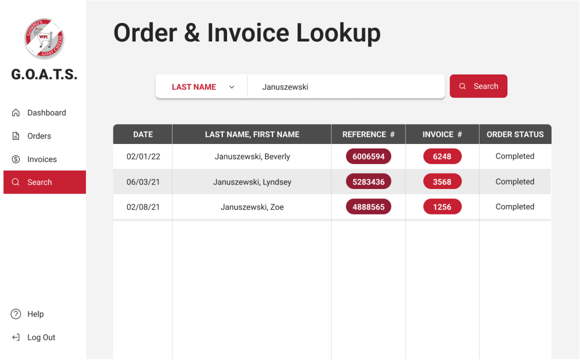
Processed

Goat, Gompei the
Ref. #: 314159265
Inv. #: 2718
# of Logs: 5

Shipped

Ryan, James
Ref. #: 548645257
Inv. #: 2346
# of Logs: 3

Figure x. Active Orders Mockup.



Order & Invoice Lookup

LAST NAME: Januszewski

DATE	LAST NAME, FIRST NAME	REFERENCE #	INVOICE #	ORDER STATUS
02/01/22	Januszewski, Beverly	6006594	6248	Completed
06/03/21	Januszewski, Lyndsey	5283436	3568	Completed
02/08/21	Januszewski, Zoe	4888565	1256	Completed

Figure xi. Order and Invoice Lookup Mockup.

Figure xii. Logo: Gompei's Operations and Accounting Tracking System.

System Development

To begin development of the new system, we used Amplify Studio within Amazon Web Services. We used Amplify Studio to host project resources like the backend database, API, and front-end code.

We created our database schema using GraphQL which automatically generated a DynamoDB table for each defined model. Along with that, it created a fully functioning GraphQL API, complete with auto generated resolvers for basic interaction functionality with the database. DynamoDB is a key value store (NoSQL) database which allows us to read and write data quickly.

Finally we began front end development using React.js to make a single-page application, which will help make transitions faster and give the site more of a native app feel.

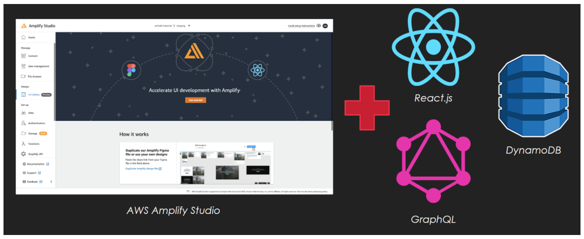


Figure xiii. System Development Tools.

Next Steps

Due to the nature of system prototyping, the "prototype is a "quick and dirty" version of the system and provides minimal features" (Dennis, 2014). The next step in the future for this project is to conduct user tests. "Following reaction and comments from the users, the developers reanalyze, redesign, and reimplement a second prototype that corrects deficiencies and adds more features. This cycle continues until the analysts, users, and sponsors agree that the prototype provides enough functionality to be installed and used in the organization (Dennis, 2014).

Other future steps include:

- Implement additional functionality
- Automate order input
- Create Westfield Farm Perspective
- Include accessibility features

Final Thoughts

Within the last four years, Gompei's Goat Cheese has become a pillar of the WPI community. From fundraisers to raffles to even trivia nights with The WPI Business School, GGC has become more involved in campus events every day. We want to see this small business beat the odds and make it past its 15 year anniversary. To do this, we hope that this project will be the foundation and inspiration for future projects to be completed with Gompei's Goat Cheese.

Gompei's Goat Cheese may be the first student-run, non-profit business at WPI, but we know it will not be the last. Our hope is that this project may inspire others to pursue their own student-run business endeavors so that they can apply their knowledge they are gaining from WPI classes in real time while also "be[ing] a part of something meaningful that impacts [their] community" (Gompei's, 2022).

Table of Contents

List of Figures

Introduction

Small businesses are the backbone of the nation's economy. Without them, the freedom of choice amongst consumers would be vastly diminished. According to the Small Business Administration (2021), small businesses account for 99.9% of all U.S. employer firms. Not only are small businesses everywhere, but they are also important contributors to their local communities. According to a study by Sims (2009), tourists deliberately consume food or drink that they consider to be "local" to experience the culture of the area. When people travel to a new place, they are looking for new experiences, especially with food. Tourism helps the food industry thrive, which in turn boosts the local economy. However, as important and beneficial as these small businesses may be, not all of them will be able to succeed.

It is challenging to create and maintain a successful small business. In data collected by the Small Business Administration, about a third of small businesses failed within the first two years of their existence (Small, 2021). So, how does a small business stay afloat? Robert Normand attributes a portion of the success of small businesses to Operational Support Systems. The goal of Operational Support Systems is to help a team stay organized and work more efficiently (Normand, 2019).

One example of an Operational Support System is an Enterprise Resource Planning System, or ERP for short. Businesses who adopt ERP systems gain the benefits of: improved business insight, enhanced collaboration, improved efficiency, consistent infrastructure, higher user-adoption rates, and reduced risk (Oracle, 2022).

Gompei's Goat Cheese (GGC) is a non-profit, student-run business at WPI. They sell award winning goat cheese produced by Westfield Farm in Hubbardston, MA, and donate all proceeds to the WPI Global Scholarship Program, which supports students studying abroad. The existing operations structure of GGC created harmful inconsistencies between GGC's records of orders and invoices and the farm's records, which impacted the fulfillment of orders and contributed to limiting the growth of GGC. It became clear that GGC was in need of a new order and invoice management system, along with improved communication with the farm.

The goal of our project was to begin development of a cloud-based enterprise resource planning system for GGC so that they may improve and grow operations.

Our team accomplished this goal through four objectives:

1. Evaluate GGC Operations.
2. Design GGC ERP System.
3. Begin Development of Prototype ERP System.
4. Establish Support Documentation.

Background

We begin this chapter by introducing small businesses and explaining their importance in local communities. We then analyze the successes and failures of small businesses. Next, we discuss enterprise resources planning systems and the role they play in helping small businesses. Lastly, we introduce Gompei's Goat Cheese by giving a brief overview of their business and explaining the need for an improved operations and accounting management system.

Importance of Small Businesses

Small businesses are the backbone of the nation's economy. Without them, the freedom of choice amongst consumers would be vastly diminished. According to the Small Business Administration (2021), a small business is defined as an independent business having less than 500 employees. In the United States, there are 32,540,953 small businesses, which account for 99.9% of all U.S. employer firms (Small, 2021). Not only are small businesses everywhere, but they are also important contributors to their local communities. Small business products tend to be locally made and specific to their community. According to Better Business Bureau (2019), and depicted in figure 1, "if you spend $100 at a local business, roughly $68 stays within your local economy." That's 58% higher than non-local businesses, where about $43 stay within the local economy.

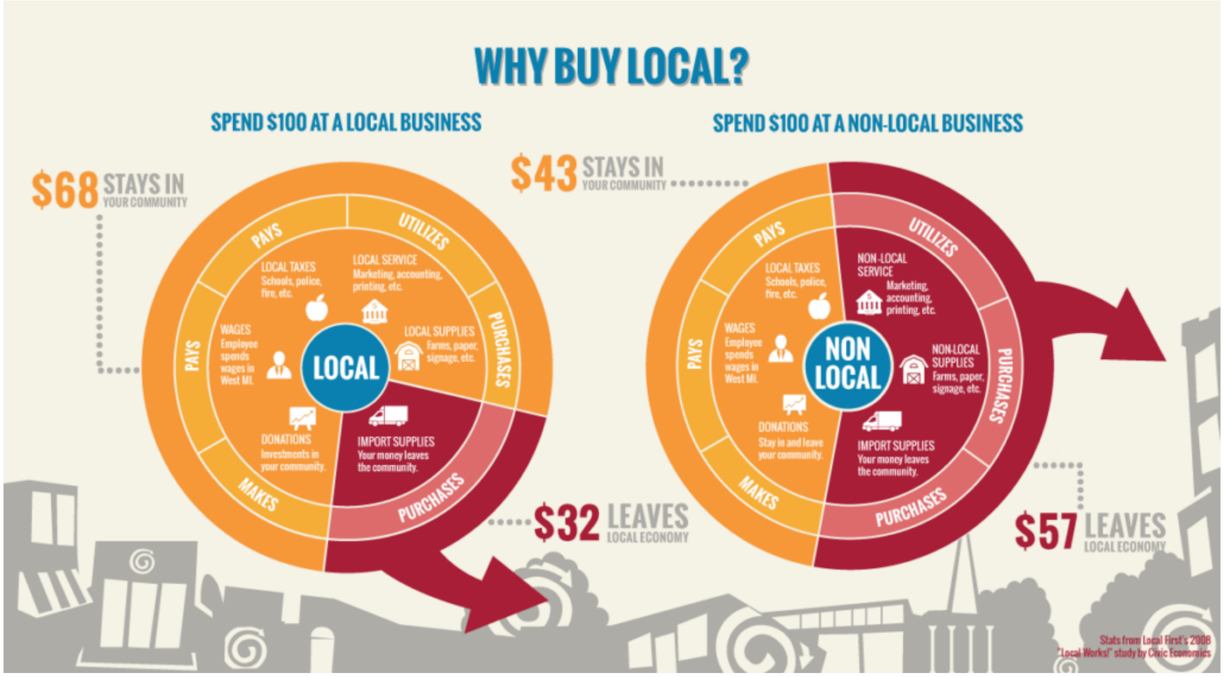

WHY BUY LOCAL?
SPEND $100 AT A LOCAL BUSINESS
$68 STAYS IN YOUR COMMUNITY
PAYS
UTILIZES
PAYS
PURCHASES
MAKES
PURCHASES
LOCAL TAXES Schools, police, fire, etc.
LOCAL SERVICE Marketing, accounting, printing, etc.
WAGES Employee spends wages in West MI.
LOCAL
LOCAL SUPPLIES Farms, paper, signage, etc.
DONATIONS Investments in your community.
IMPORT SUPPLIES Your money leaves the community.
$32 LEAVES LOCAL ECONOMY
SPEND $100 AT A NON-LOCAL BUSINESS
$43 STAYS IN YOUR COMMUNITY
PAYS
UTILIZES
PAYS
PURCHASES
MAKES
PURCHASES
LOCAL TAXES Schools, police, fire, etc.
NON-LOCAL SERVICE Marketing, accounting, printing, etc.
WAGES Employee spends wages in West MI.
NON LOCAL
NON-LOCAL SUPPLIES Farms, paper, signage, etc.
DONATIONS Stay in and leave your community.
IMPORT SUPPLIES Your money leaves the community.
$57 LEAVES LOCAL ECONOMY
Stats from Local First's 2008 "Local Works!" study by Civic Economics

Figure 1. Why Buy Local? (Bureau, 2019).

Small businesses in the food industry contribute to a sustainable tourism experience in their local communities. According to a study by Sims (2009), "Over 60% of the tourists interviewed said that they had deliberately chosen to consume foods or drinks that they considered 'local' while on holiday which suggests that, rather than just looking for something 'different', tourists are seeking products that they feel will give them an insight into the nature of a place and its people." When people travel to a new place, they are looking for new experiences, especially with food. Tourism helps the food industry thrive, which in turn boosts the local economy. However, as important and beneficial as these small businesses may be, not all of them will be able to succeed.

Successes and Failures of Small Businesses

According to the Small Business Administration and the data they collected between 1994 and 2019, around 32.4% small businesses failed to survive longer than two years. By year five, 51.1% of these businesses had failed, and by year ten, 66.4% had failed. By the fifteen year mark, only around 25% of those original businesses were still operating (Small, 2021). This trend can be seen in figure 2. It is evident that keeping a small business afloat can be challenging.

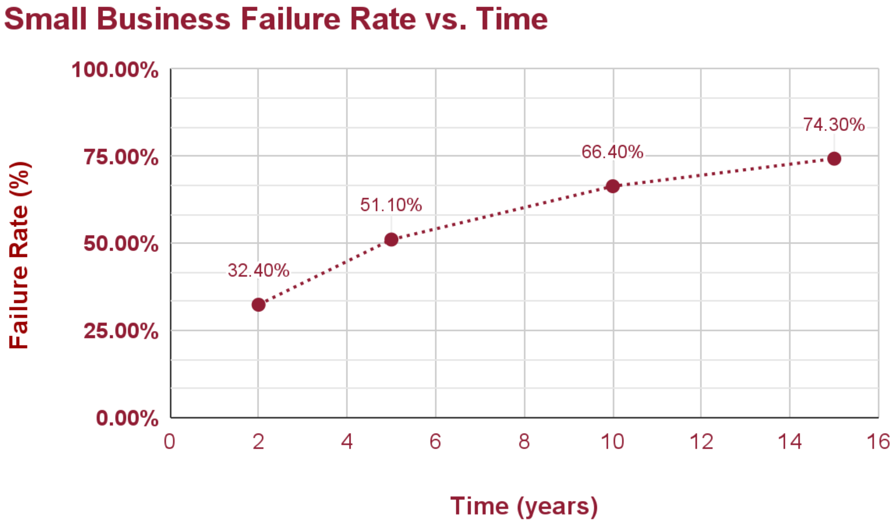


Figure 2. Failure Rates of Small Businesses Over Time (Data from (Small, 2021)).

The COVID-19 Pandemic negatively affected many industries during 2020 and 2021, but none as much as Accommodations & Food industry. In 2020, around 80% of these small businesses were negatively impacted by COVID-19, which was 1.6 times higher than the national average at the time. Since then, the percentage of small businesses affected in the Accommodations &

Food industry dropped to 55%, but that is still over 2 times the national average for the same time period (Small, 2021).

Given the uphill slope new small businesses have to climb, what are ways they can succeed and outlive the competition? In the article “4 Reasons Why Small Businesses Succeed”, Robert Normand mentions that the owners of successful businesses have developed Operational Support Systems. The goal of these systems is to help a team stay organized and work more efficiently. This is done by automating certain tasks involving basic tracking needs and allowing the team more time to focus on innovating (Normand, 2019). An example of a system that fills this role is an Enterprise Resource Planning System.

Effectiveness of Enterprise Resource Planning Systems

One way for businesses to help improve their chances of succeeding is by using an Enterprise Resource Planning System, or ERP for short. According to Oracle (2022), an Enterprise Resource Planning System is “a type of software that organizations use to manage day-to-day business activities such as accounting, procurement, project management, risk management and compliance, and supply chain operations. Businesses who adopt ERP systems gain the benefits of: improved business insight, enhanced collaboration, improved efficiency, consistent infrastructure, higher user-adoption rates, and reduced risk (Oracle, 2022). Depending on the requirements of a system, developers may choose to use a cloud-based ERP instead of one that is stored locally. The benefit to a cloud-based ERP system is that “organizations access the software over the Internet, so all that’s needed is connection and a browser” (Fisher, 2022).

An ERP can have different components, depending “on [the business’] industry and specific business needs” (Fisher, 2022). Figure 3 depicts many of the possible components that an ERP can include.

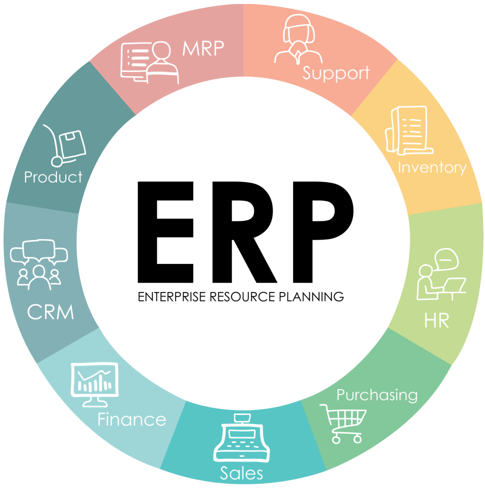
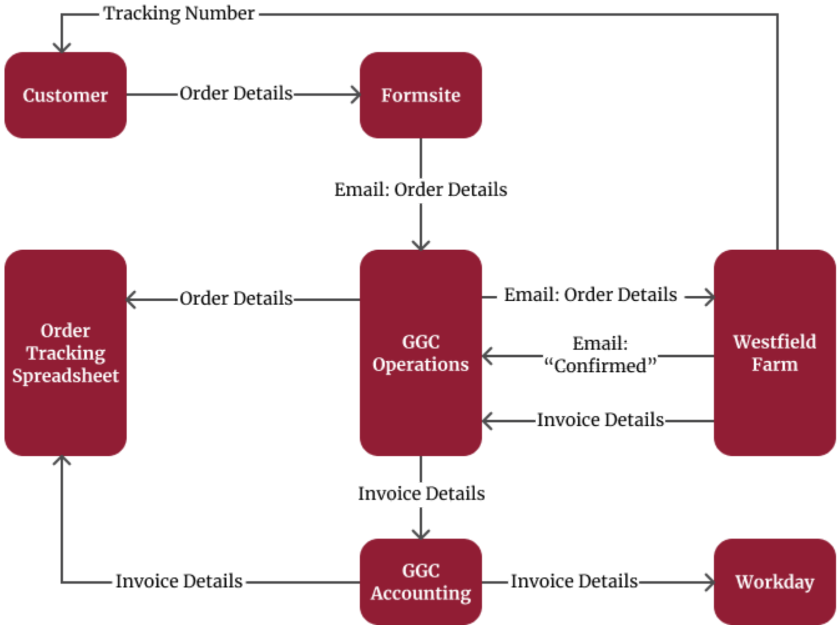


Figure 3. Components of an ERP System (Adapted from (Fisher, 2022)).

The implementation of an Enterprise Resource Planning System is a critical factor in determining its success. According to Ghosh (2012), 60% of ERP projects fail. Typical reasons for the failure of an ERP system include: lower returns than expected, inability of the ERP system to meet the predetermined functional requirements, missing development and deployment ideas, incorrect working of the system, and not meeting expectations.

Gompei’s Goat Cheese

Gompei’s Goat Cheese (GGC) is a non-profit, student-run business at Worcester Polytechnic Institute. GGC’s mission is “to support WPI global scholarships, learn entrepreneurial skills, and interact with the WPI and Worcester community by selling award-winning goat cheese so that we can be a part of something meaningful that impacts our community” (Gompei’s, 2022). GGC sells local goat cheese produced by Westfield Farm in Hubbardston, MA, and donates all proceeds to the WPI Global Scholarship Program, which supports students studying abroad. GGC handles orders and promotion of their goat cheese, but does not interact with the production nor shipping of the cheese to the customer.



Pain Points to Note:

- No shared order reference number between GGC and Westfield Farm
- Communication with the farm only occurs through the operations team
- Customer places order with GGC but gets a tracking number from Westfield Farm
- Emails sent to the farm have been lost (no email reply with “confirmed”)

Figure 4. Data Flow Diagram (Level 0) of GGC’s Current Operations System.

Currently, GGC uses Google Sheets and communicates by email to manage operations. Figure 4 (above) shows the data flow between GGC and Westfield Farm. The team collects order details and payment, and then forwards order information to the farm. The farm will confirm orders through email. Once the farm ships the order to the customer’s home, they automatically generate and send an invoice to GGC to request payment for the order.

This existing operations structure created harmful inconsistencies between GGC's records of orders and invoices and the farm's records, which impacted the fulfillment of orders and contributed to limiting the growth of GGC. It became clear that GGC was in need of a new order and invoice management system, along with improved communication with the farm.

Methodology

The goal of our project was to begin development of a cloud-based enterprise resource planning system for GGC so that they may improve and grow operations. We achieved our goal by completing the following objectives:

1. Evaluate current GGC Operations by gathering feedback on the current operations system and ideas and needs for the new system.
2. Design an ERP system to meet GGC's needs.
3. Begin development of a prototype ERP system.
4. Establish support documentation for GGC, Westfield Farm, and future student groups so that they can continue development and implementation of this new system.

To properly develop this ERP system, we followed a methodology that implements the fundamental four phase Systems Development Life Cycle (SLDC). Due to the condensed timeline of this project, we decided to use the system prototyping methodology. "System prototyping performs the analysis, design, and implementation phases concurrently in order to quickly develop a simplified version of the proposed system" (Dennis, 2014). Figure 5 below depicts this methodology and shows how our objectives fit into each phase.

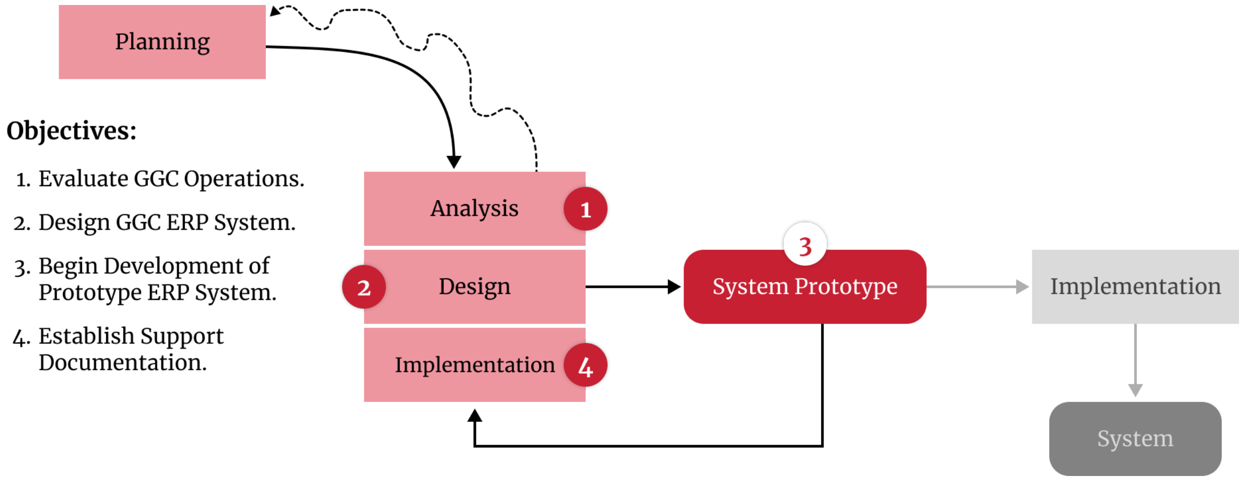

Planning
Objectives:
1. Evaluate GGC Operations.
2. Design GGC ERP System.
3. Begin Development of Prototype ERP System.
4. Establish Support Documentation.
Analysis
1
2
Design
Implementation
4
3
System Prototype
Implementation
System

Figure 5. System Prototyping Methodology and Project Objectives.

Objective 1: Evaluate GGC Operations

The first objective was to evaluate current GGC operations by gathering feedback from GGC and Westfield Farm on the current operations system as well as exploring ideas that the two groups had for the new system. The second part of this objective was to create clear documentation of the requirements for the new system. To complete this objective, we answered the following research questions:

1. What do GGC members like about the current operations process?
2. What do GGC members think needs improvement in the operations process?
3. What would GGC members like to see in a new operations system?
4. What does the partner farm like about the current process?
5. What does the partner farm think needs improvement in the operations process?
6. What would the partner farm like to see in a new operations system?

To answer these questions, we used the most common Requirement Elicitation Technique: the interview (Dennis, 2014). We conducted individual, semi-structured, informal interviews. We wanted to conduct these interviews individually so that each person could express their personal experiences with the system without being influenced by others. We interviewed 6 GGC members virtually over Zoom: two retired Chief Operations Officers (COO), the current COO, the current Chief Accounting Officer (CAO), the CAO in training, a retired Chief Executive Officer (CEO), and the current CEO. We also interviewed the President of Westfield Farm in person at the farm in Hubbardston, MA. The interview consent forms, study protocols, and guiding questions can be found in Appendices A, B and C.

The second part of this objective was to categorize and prioritize the feedback about the current operations system and ideas for the new system identified in the interviews. To do this, we pulled direct quotes from interviewees and grouped them into five different categories:

1. How often do stakeholders interact with the system?
2. What device do stakeholders use the system on?
3. What do stakeholders like about the current system?
4. What do stakeholders think needs improvement/ do not like about the current system?
5. What would GGC like to see in a new system?

We then created a set of conclusions based on the responses and used them to help develop the requirements for GGC's ERP System. Along with this, we created a set of requirements for the new system to aid us in the next objective.

Objective 2: Design ERP System

The second objective was to design an ERP system that can better fit the needs of Gompei's Goat Cheese. To complete this objective, we will develop documentation to design the new system. This documentation will consist of the following deliverables:

1. System Use-Cases
2. Data Flow Diagram (Levels 0, 1)
3. Entity Relationship Diagram (Logical)
4. Data Dictionary
5. User Flow Diagrams

6. System Mockups & Hi-fidelity Prototype

Objective 3: Begin Development of Prototype ERP System

The third objective was to begin development based on the design documentation developed in objective 2 to create a foundation for the prototype ERP system. The following steps were followed to complete this objective:

1. Create database and API backend
2. Develop user interface using React.js

We chose Amazon Web Services to host the database and code for the project because AWS is reliable and would allow ease of access for future teams to add and make changes to the ERP system. Within AWS we used their Amplify Studio tool to hold all of the project's contents. We did this because everything would be all in one place and it would be easier to add new users to the project to allow them to edit and make changes.

We created a key value store (NoSQL) database using DynamoDB. Key value stores are useful in systems where you want to read and write data quickly. We chose to use a key value store such as DynamoDB because we originally planned to automate and use real-time order data. Although the scope of our project changed, we wanted to allow future projects the flexibility that NoSQL databases provide. Another reason why we chose to use DynamoDB was because it was the only type of database in AWS, at the time, that could interface directly into Amplify Studio.

To connect the database with the front-end site, we used the GraphQL API. We did this because when building a GraphQL schema, Amplify studio automatically creates a DynamoDB table for any object with the *@model* directive tag. This made it very easy to create the database that we described earlier. To go along with that, it creates a fully functioning GraphQL API complete with auto generated resolvers for basic interaction functionality with the database.

To develop the front end, we used React.js because it is a powerful, modern Javascript library for building user interfaces. React.js also has the advantage of being used to create single-page applications, which helps make transitions faster and gives the site a native app feel.

Objective 4: Establish Support Documentation

The fifth and final objective was to establish support documentation so that Gompei's Goat Cheese, Westfield Farm, and future student groups can continue working on this prototype ERP system and integrate it into GGC's daily operations. This objective, while listed last, will be developed concurrently with objective 3. To complete this objective, we will answer the following research questions:

1. What methods of giving support would benefit GGC employees the most?
2. What support do regular users need?
3. What support do maintenance users need?

We researched best practices and worked with the team to create written documentation on the system's design and use. We provided the Gompei's Goat Cheese Team full access to all the code and resources created during the development of this project. Finally, we created a contact pipeline with ourselves and the team to help aid in the future development of the site. The Support Documentation can be found in Appendix F.

Requirement Elicitation Findings

In this chapter we share key findings from our interviews with GGC team members and the President of Westfield Farm. Based on these findings and our own experience with the system as GGC associates, we provide a set of system requirements.

Interview Findings

Interaction with the current GGC operations system ranges from a few times a semester to nearly every day.

The Gompei's Goat Cheese Chief Operations Officer and Chief Accounting Officer interact with the current system the most frequently. The COO in 2021 shared that they usually interacted with the system "*nearly every day, or nearly every day that we got orders*." They are the ones who input the orders into the order tracking system and then the CAO will record when invoices are received from Westfield Farm. Other team members, like those in marketing and sales, usually don't interact with the system at all. The new CEO, who used to be in sales, said that they interacted "*extremely infrequently, like maybe once or twice throughout this entire semester that's already past*" with the current system.

Team members prefer to use their computer, rather than their phone, when using the current operations system.

Every team member that we interviewed shared that they much prefer to interact with the system while using their computer, rather than other devices. Specifically, the COO from 2019-2021 emphasized that, "*Just based on how I was sending, like the orders received from formsite to [the partner farm], I did not ever want to touch that on my phone*."

The current system is easy to use, and everyone has access to it.

One of the most frequent positive feedback that we received about the current system is that it is very simple to use, and that everyone has access to it. When speaking with the CEO from 2020 - 2021, they mentioned that, *"it's a spreadsheet, like it's pretty intuitive and straightforward."* and the COO in 2021 shared that, *"Just being able to have that information between the different departments so people being able to view all aspects of it, like having that shared area with finance so that I can see what finance is doing and how they're progressing through invoices and things like that and then they can see how we're doing in receiving invoices and then therefore, sending them over to finance and things like that."*

Westfield Farm and GGC use different order numbering systems.

When interviewing with members of the accounting team, it was brought to our attention that GGC's internal order tracking number is different from the one used by Westfield Farm. Typically, this is something that can be expected, as they are separate systems after all, but it creates a difficult task for the finance/accounting team who has to reconcile orders given Westfield Farm's invoice number with orders in GGC's current system. This process is mostly done using the name and address associated with the order and invoice.

Inputting information into the current system takes too long, and there's a lot of redundancy in the work that the operations team does.

When interviewing with the past COO's, they shared that a lot of the work they do tends to take a large amount of time, and can be repetitive. The COO in 2021 said that, *"There's no reason you should have to go into your email and copy and paste each individual line. Like yes that's super simple and it's not a hard thing to do, but it's tedious and monotonous and it's inefficient."* If the operations team was able to worry less about the specific details of each order, it would allow them to focus on other, more innovative, work to help the team.

Westfield Farm handles Gompei's Goat Cheese order's differently than its normal orders.

When interviewing with the President of Westfield Farm, we learned that the farm has to treat their order's differently than normal ones. To start, when an order is placed on the farm's website, it goes directly into the farm's system and an invoice is automatically created. When GGC sends an order to the farm via email, the farm has to manually enter the order details into an invoice, meaning that it is more prone to errors. These errors consist of address discrepancies and name misspellings.

System Requirements

Business Requirements

- Store all order, invoice, product, and customer information
- View number of active orders and invoices and where they are in the system
    - e.g. how many orders are placed, processed, or shipped
- Westfield Farm and GGC collaborate and use the system (both are able to access)

User Requirements

- Input and modify order information
    - Order information includes customer information
- Input and modify invoice information

- Search for orders and invoices via reference number, invoice number, or customer last name
- Add new products
- Copy paste order information easily
- Manage user permissions (add / remove collaborators)

Functional Requirements

- Store, modify, and recall order/invoice information
- Query for an order based on reference number, invoice number, or customer last name
- Generate an invoice when an order is added
- Create or reference a customer when an order is added

Nonfunctional Requirements

- Desktop application
- Multi-user capability (multiple users can access at the same time)
- Multi-user capability (multiple users exist and can both access the system)
- Users have different roles (GGC, Partner Farm)
- Secure user login and data storage
- User friendly (follow Jakob Nielsen's principles)
- Incorporate GGC's brand identity
- Load data / Refresh UI quickly
- Provide help on every page (help to use the system)
- Provide accessibility options per user (i.e. high contrast mode, increased text size)

System Requirements

- Develop database using AWS, Amplify Studio
    - Reliable, easy to share with future developers
- Automate order input
- Show real time order data
- Develop front-end using React.js
    - Single-page application builder, makes transitions faster

System Design & Development

This chapter dives into the system design and development phase of our project.

List of Use-Cases

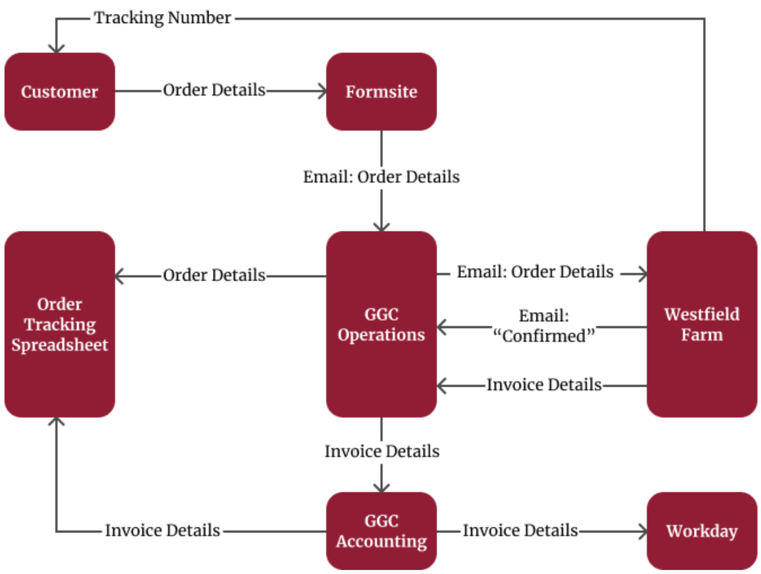
- Add User
- Remove User
- Log In
- Log Out
- Record New Order
- Delete Order
- Update Order Information
- Update Invoice Information
- View Active Orders
- View Completed Orders
- View Order Information
- View Invoice
- View Customers
- Search Orders
- Copy Text Information

For each use case, we noted the participating actors, entry conditions, exit criteria and flow of events. The full use-case documentation can be found in Appendix D.

These use-cases are what was planned for the system to implement in this prototype; the future version of this system will most likely contain many more use-cases.

Data Flow Diagrams

The first data flow diagram we created is located in the background chapter (Figure 4. Data Flow Diagram (Level 0) of GGC's Current Operations System). It is repeated below in figure 6 to make it easier to read and understand the importance of this diagram. This diagram helped us identify certain pain points like the redundancy of order data.



Pain Points to Note:

- No shared order reference number between GGC and Westfield Farm
- Communication with the farm only occurs through the operations team
- Customer places order with GGC but gets a tracking number from Westfield Farm
- Emails sent to the farm have been lost (no email reply with "confirmed")

Figure 6. Data Flow Diagram (Level 0) of GGC's As-Is Operations System (repeat).

The next data flow diagram we created was another level zero data flow diagram (DFD). This DFD, depicted in figure 7, helps show how the new system (G.O.A.T.S.) would reduce redundancy by centralizing order and other data. One place where we did not reduce redundancy was in the invoice details. This is due to one of our findings from the interview that we conducted with Westfield Farm. The farm's system allows many invoices to be sent in a batch to a GGC email. This part of their system was noted as important to the farm, so we decided that it would be best to not change that functionality and simply change the email to the GGC accounting team to streamline the input of invoice details on GGC's end.

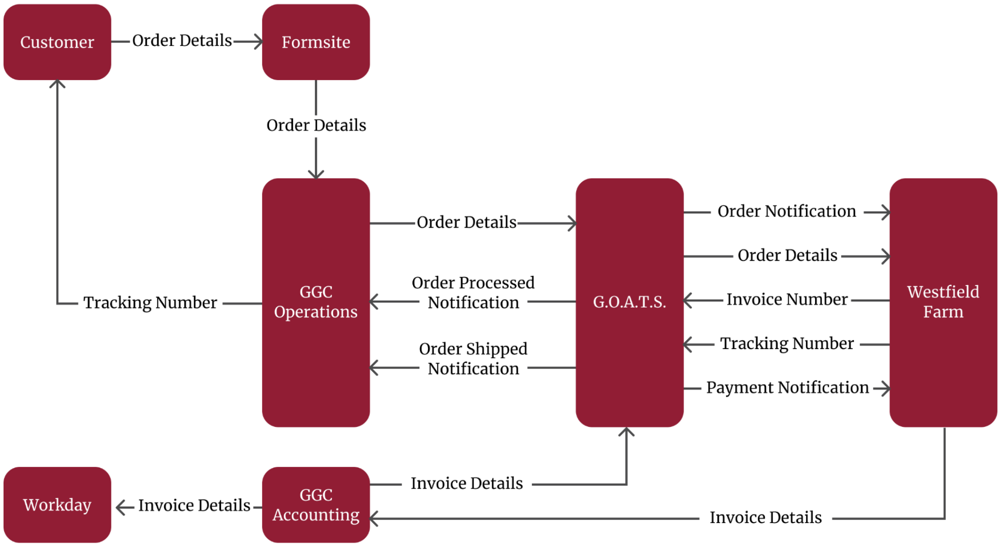


Figure 7. Data Flow Diagram (Level 0) AKA Context Diagram of To-Be System.

This next diagram, figure 8, is a level one DFD. This takes the level zero DFD and breaks it up into entities, processes, and data stores. These reflect the data flow of critical functionalities of the system, such as adding a new order (Process 1.0) and updating order details (Process 1.1).

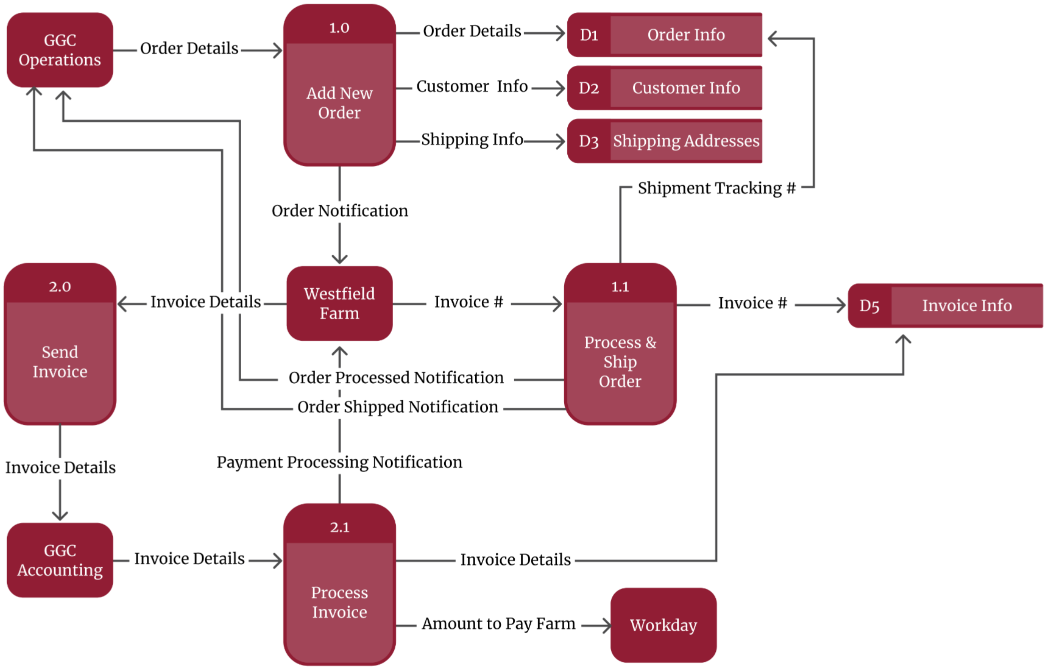


Figure 8. Data Flow Diagram (Level 1) of To-Be System.

Logical Entity Relationship Diagram

Although we ended up using a NoSQL database for the initial prototype, we also wanted to document the logical entity relationship diagram (ERD). A logical ERD is a visual representation of a relational database. In the future of this project, a team may decide that a relational database may be more suited to our needs.

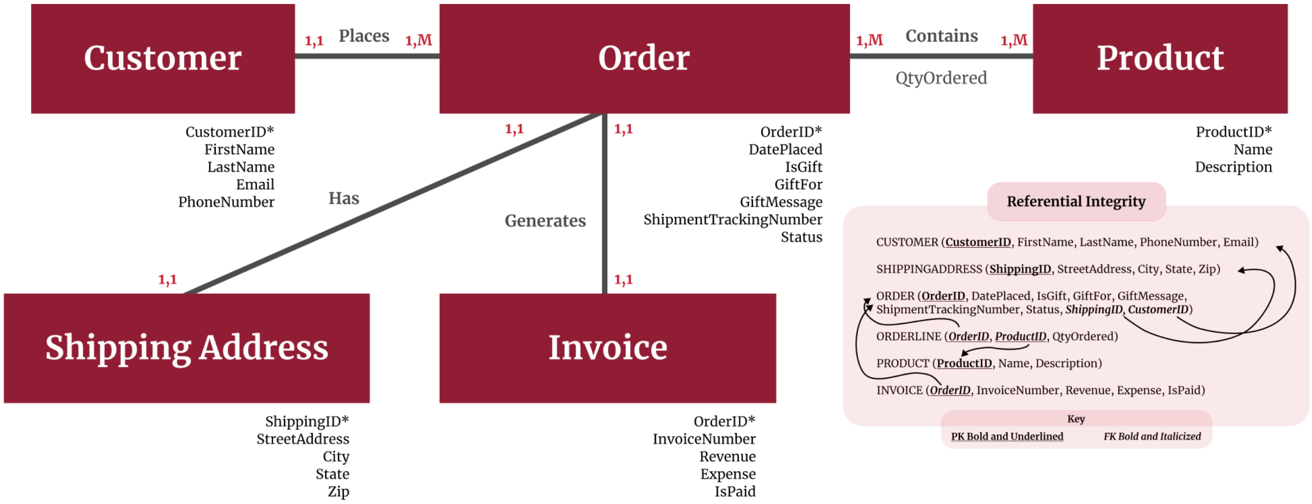
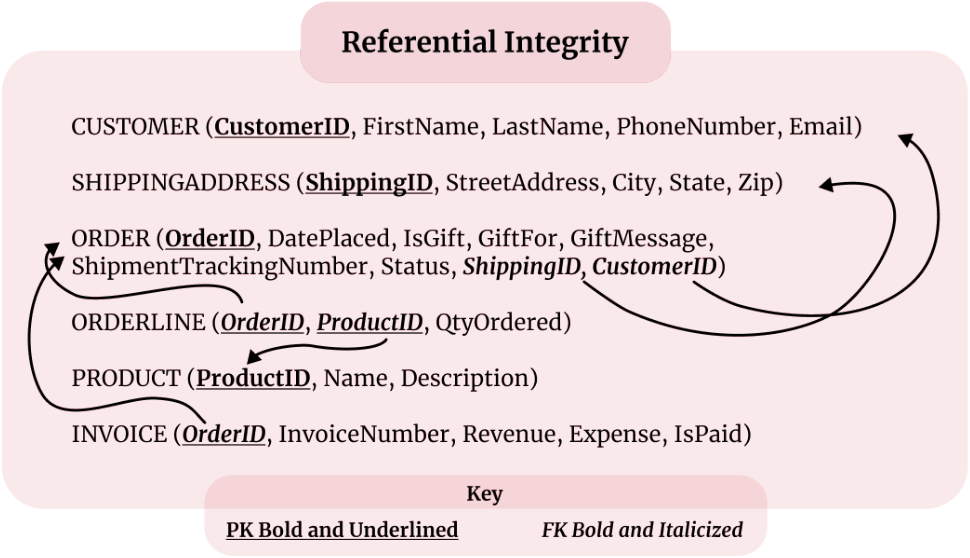


Figure 9. Entity Relationship Diagram.



Referential Integrity

CUSTOMER (**CustomerID**, FirstName, LastName, PhoneNumber, Email)

SHIPPINGADDRESS (**ShippingID**, StreetAddress, City, State, Zip)

ORDER (**OrderID**, DatePlaced, IsGift, GiftFor, GiftMessage, ShipmentTrackingNumber, Status, ***ShippingID***, ***CustomerID***)

ORDERLINE (***OrderID***, ***ProductID***, QtyOrdered)

PRODUCT (**ProductID**, Name, Description)

INVOICE (***OrderID***, InvoiceNumber, Revenue, Expense, IsPaid)

Key

PK Bold and Underlined ***FK Bold and Italicized***

Figure 10. Referential Integrity (Magnified).

Data Dictionary

This data dictionary reflects the tables that would be expected in a relational database. However, this also defines all of the details for the attributes in each table reflected in the current database schema.

CUSTOMER

Attribute	**Type**	**Constraint**	**Description**
CustomerID	ID	Primary Key	Automatically generated ID
FirstName	String		Customer's first name
LastName	String		Customer's last name
PhoneNumber	String		Customer's primary phone number
Email	String		Customer's primary email

SHIPPING ADDRESS

Attribute	**Type**	**Constraint**	**Description**
ShippingID	ID	Primary Key	Automatically generated ID
StreetAddress	String	Not Null	
City	String	Not Null	
State	String	Not Null	
Zip	String	Not Null	

ORDER

Attribute	**Type**	**Constraint**	**Description**
OrderID	ID	Primary Key	Reference Number (generated by GGC Formsite)
DatePlaced	AWS Date	Not Null	Date the order was

			placed on.
IsGift	Boolean		True if the order is a gift.
GiftFor	String		Name of the recipient of the gift (Full name / title to address the order)
GiftMessage	String		Optional gift message that customers can choose to send with their order.
ShipmentTrackingNumber	String		When shipping the order, the farm receives a shipment tracking number from Stamps.com. This attribute tracks that number. When filled out, the order status changes to shipped.
Status	OrderStatus		Current order status (Placed, Processed, Shipped)
ShippingID		Foreign Key, References SHIPPING ADDRESS	Associated shipping address (where to send the order)
CustomerID		Foreign Key, References CUSTOMER	ID of the customer who placed an order.

ORDER LINE

Attribute	**Type**	**Constraint**	**Description**
OrderID	ID	Primary Key, References ORDER	Associated Order's ID
ProductID	ID	Foreign Key, References	Associated ID of one product in an order.

		PRODUCT	
QtyOrdered	Int		Quantity ordered of the associated product.

PRODUCT

Attribute	**Type**	**Constraint**	**Description**
ProductID	ID	Primary Key	Westfield Farm SKU Value or automatically generated sequential ID (if not cheese)
Name	String		Name of the cheese flavor or item.
Description	String		Description of the item.

INVOICE

Attribute	**Type**	**Constraint**	**Description**
OrderID	ID	Primary Key, References ORDER	Associated Order ID
InvoiceNumber	String		Westfield Farm invoice number. When filled out, order status changes to “Processed.”
Revenue	Float		Amount customer paid for the order.
Expense	Float		Amount GGC owes the farm.
IsPaid	Boolean		True if the order payment has been sent by GGC Accounting.

User Flow Diagrams

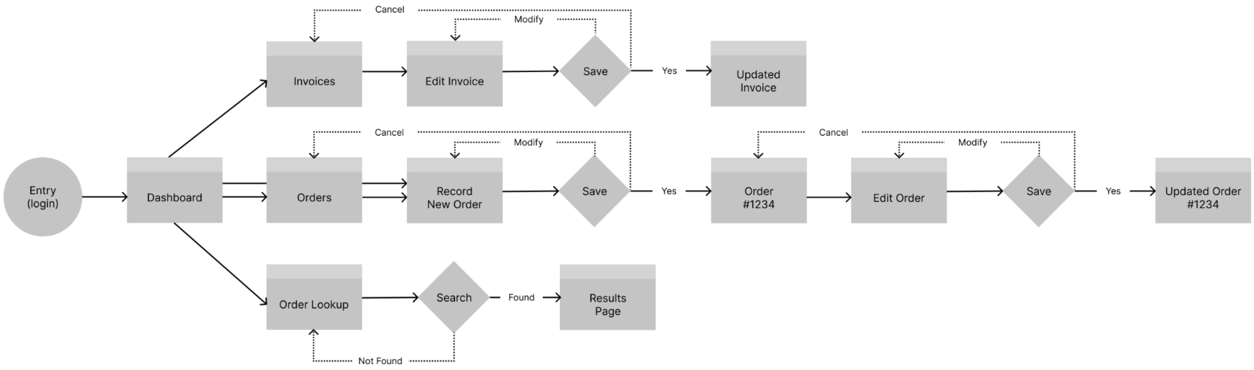


Figure 11. User Flow Diagram - Gompei's Goat Cheese.

System Mockups & Hi-fidelity Prototype

Below in figures 12-16 are screenshots of the system mockups, which were prototyped in Figma to become the hi-fidelity prototype of the new system. More mockups can be found in Appendix E. More information on Figma and how to view and interact with the mockup prototype can be found in Appendix F.

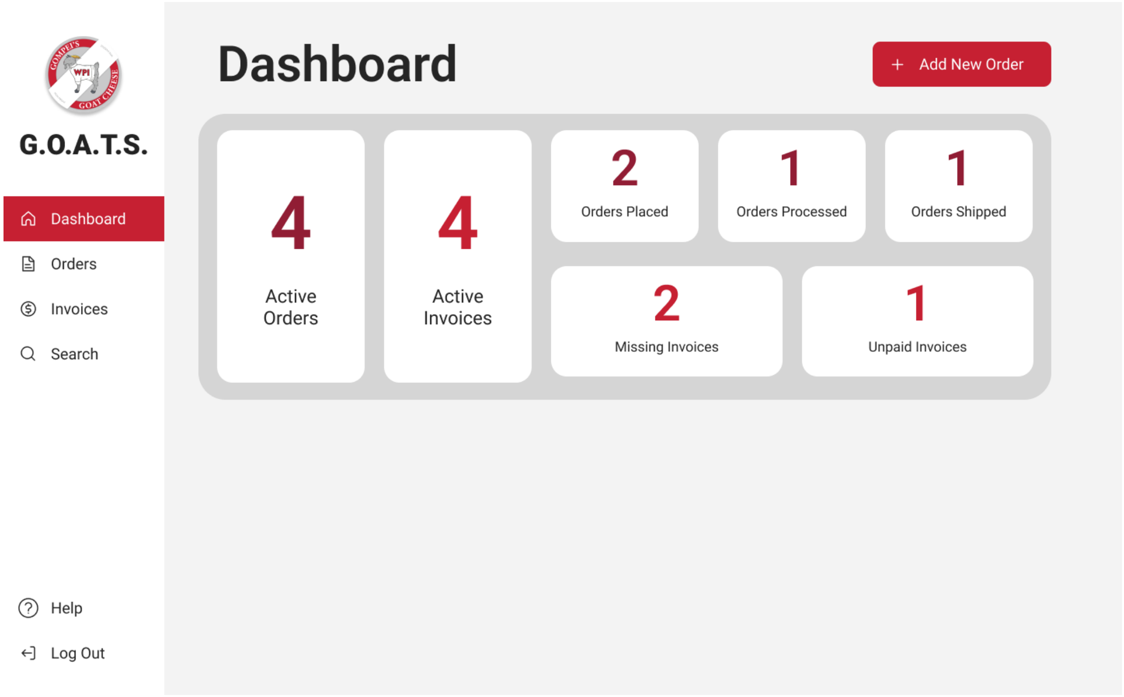
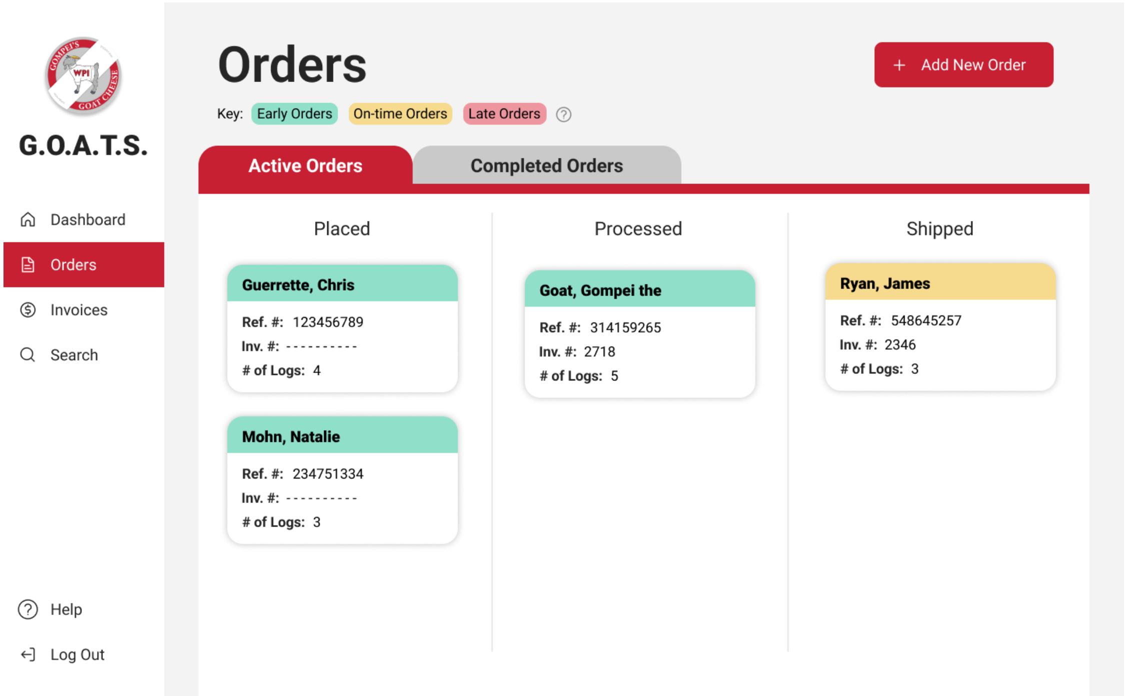


Figure 12. Dashboard Mockup.



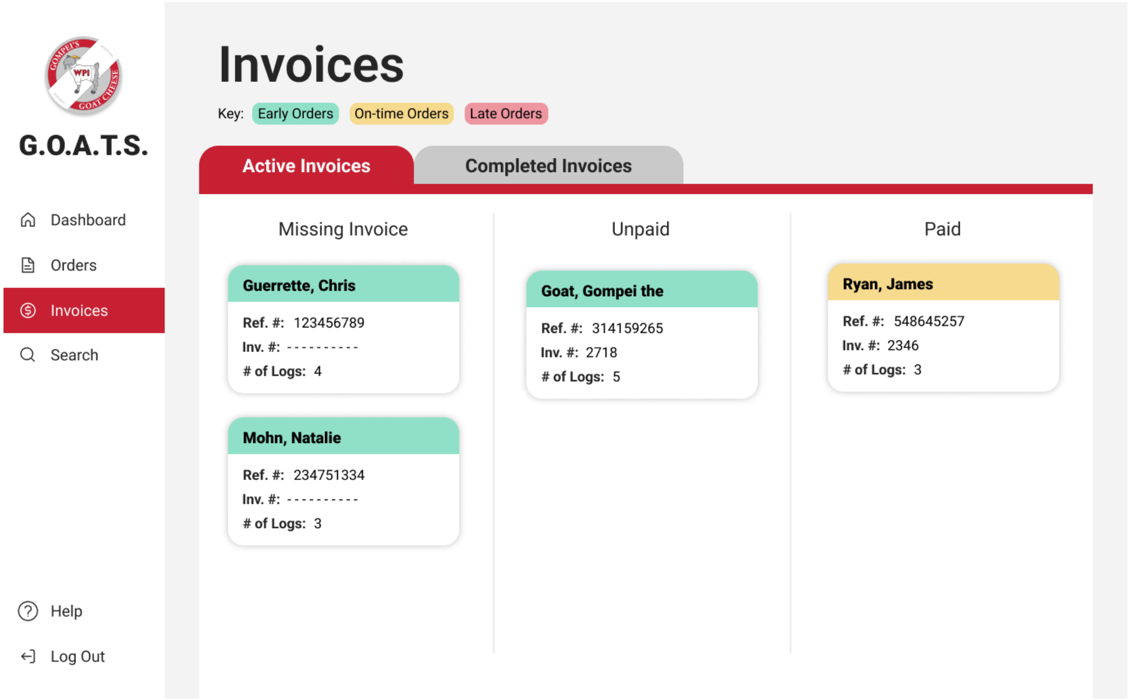
Guerrette, Chris
Ref. #: 123456789
Inv. #: ----------
# of Logs: 4

Mohn, Natalie
Ref. #: 234751334
Inv. #: ----------
# of Logs: 3

Goat, Gompei the
Ref. #: 314159265
Inv. #: 2718
# of Logs: 5

Ryan, James
Ref. #: 548645257
Inv. #: 2346
# of Logs: 3

Figure 13. Active Orders Mockup.



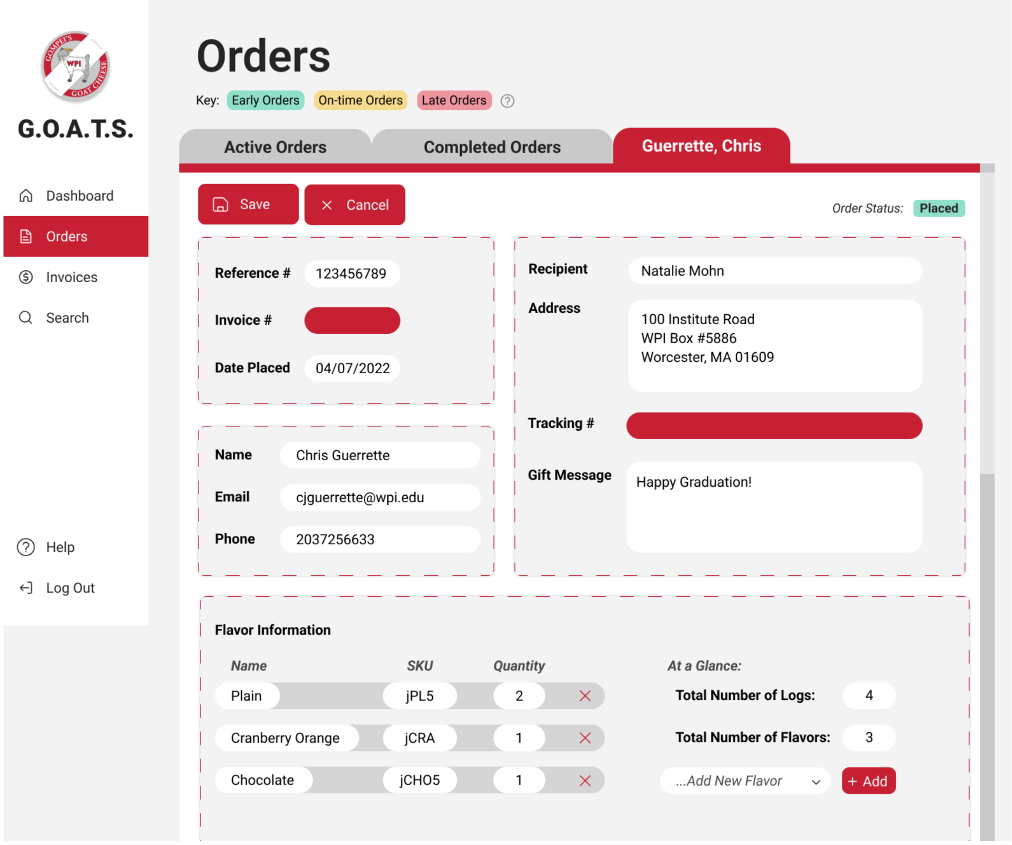
Guerrette, Chris
Ref. #: 123456789
Inv. #: ----------
# of Logs: 4

Mohn, Natalie
Ref. #: 234751334
Inv. #: ----------
# of Logs: 3

Goat, Gompei the
Ref. #: 314159265
Inv. #: 2718
# of Logs: 5

Ryan, James
Ref. #: 548645257
Inv. #: 2346
# of Logs: 3

Figure 14. Active Invoices Mockup.



Orders

Order Status: Placed

Reference #	123456789
Invoice #	
Date Placed	04/07/2022

Name	Chris Guerrette
Email	cjguerrette@wpi.edu
Phone	2037256633

Recipient	Natalie Mohn
Address	100 Institute Road WPI Box #5886 Worcester, MA 01609
Tracking #	
Gift Message	Happy Graduation!

Flavor Information

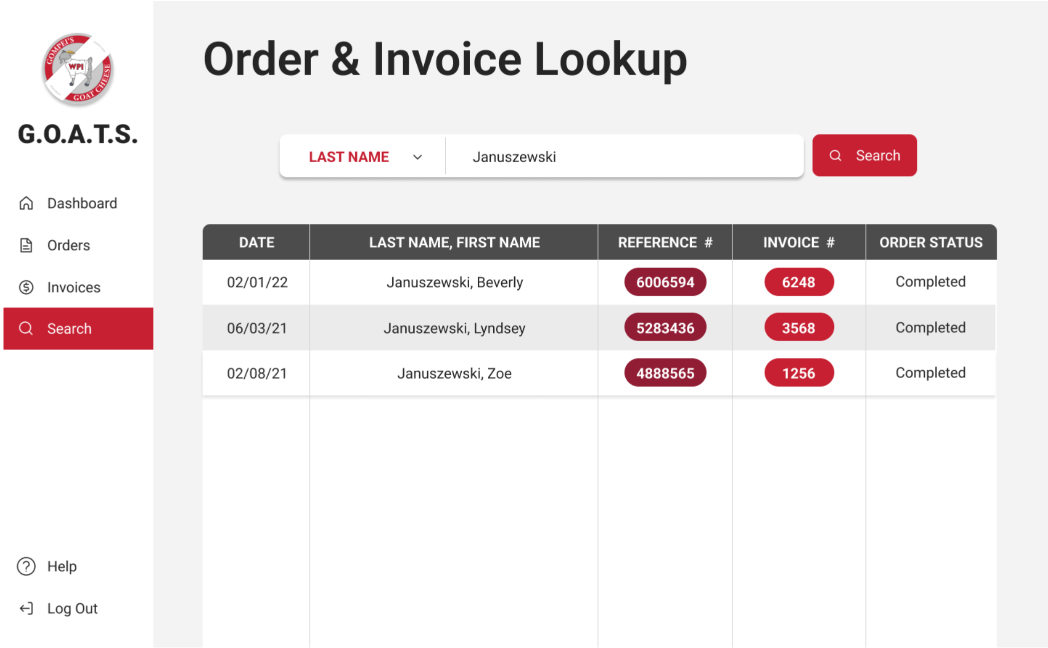
Name	*SKU*	*Quantity*	
Plain	jPL5	2	×
Cranberry Orange	jCRA	1	×
Chocolate	jCHO5	1	×

At a Glance:

Total Number of Logs:	4
Total Number of Flavors:	3

...Add New Flavor

Figure 15. Order Example Mockup.



Order & Invoice Lookup

Januszewski

DATE	LAST NAME, FIRST NAME	REFERENCE #	INVOICE #	ORDER STATUS
02/01/22	Januszewski, Beverly	6006594	6248	Completed
06/03/21	Januszewski, Lyndsey	5283436	3568	Completed
02/08/21	Januszewski, Zoe	4888565	1256	Completed

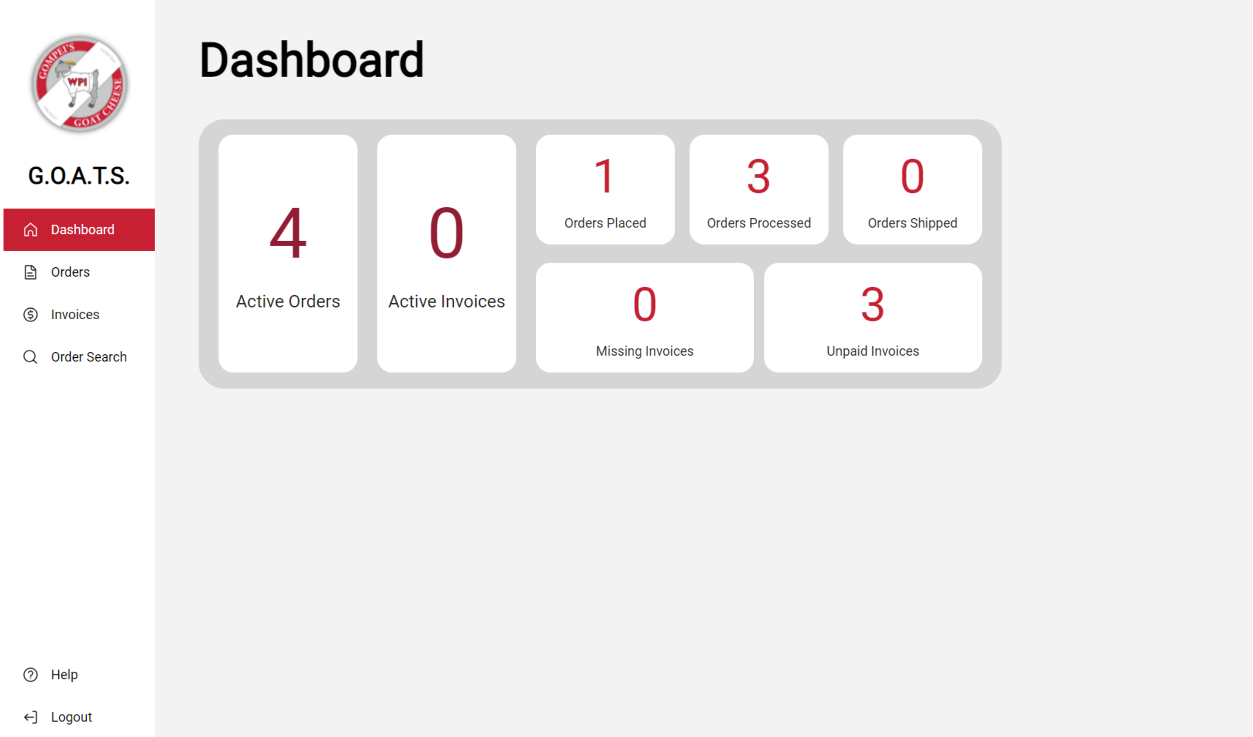
Figure 16. Order and Invoice Lookup Mockup.

Database Schema

Moving into implementation: below is the script for the database schema used in the functional prototype.

```
enum Role {
  GGC
  FARM
}

enum OrderStatus {
  PLACED
  PROCESSED
  SHIPPED_NO_INVOICE
  SHIPPED_UNPAID
  COMPLETED
  ARCHIVED
}

type Users @model @auth(rules: [{allow: public}]) {
  id: ID! @primaryKey
  email: String!
  role: Role!
}

type Product @model @auth(rules: [{allow: public}]) {
  id: ID! @primaryKey
  name: String!
  productDescription: String!
}

type OrderLine @model @auth(rules: [{allow: public}]) {
  id: ID! @primaryKey
  orderID: @index(name: "byOrderReferenceNumber", queryField:
"orderLineByOrderReferenceNumber")
  product: Product @hasOne
  qtyOrdered: Int!
}

type Invoice @model @auth(rules: [{allow: public}]) {
  id: ID! @primaryKey
  invoiceNumber: String!
  revenue: Float!
  expense: Float!
  isPaid: Boolean!
}

type ShippingAddress @model @auth(rules: [{allow: public}]) {
  id: ID! @primaryKey
  streetAddress: String!
  city: String!
  state: String!
  zipCode: String!
}

type Customer @model @auth(rules: [{allow: public}]) {
  id: ID! @primaryKey
  firstName: String
  lastName: String
  phoneNumber: String
  email: String! @index(name: "byEmail", queryField: "customerByEmail")
}

type Order @model @auth(rules: [{allow: public}]) {
  id: ID! @primaryKey
  datePlaced: AWSDate!
  isGift: Boolean!
  giftFor: String
  giftMessage: String
  shipmentTrackingNumber: String
  status: OrderStatus! @index(name: "byOrderStatus", queryField:
"orderByOrderStatus")
  shippingAddress: ShippingAddress @hasOne
  customer: Customer @hasOne
}

input AMPLIFY {
  globalAuthRule: AuthRule = {allow: public}
}
```

Front-End Development



Dashboard

4 Active Orders

0 Active Invoices

1 Orders Placed

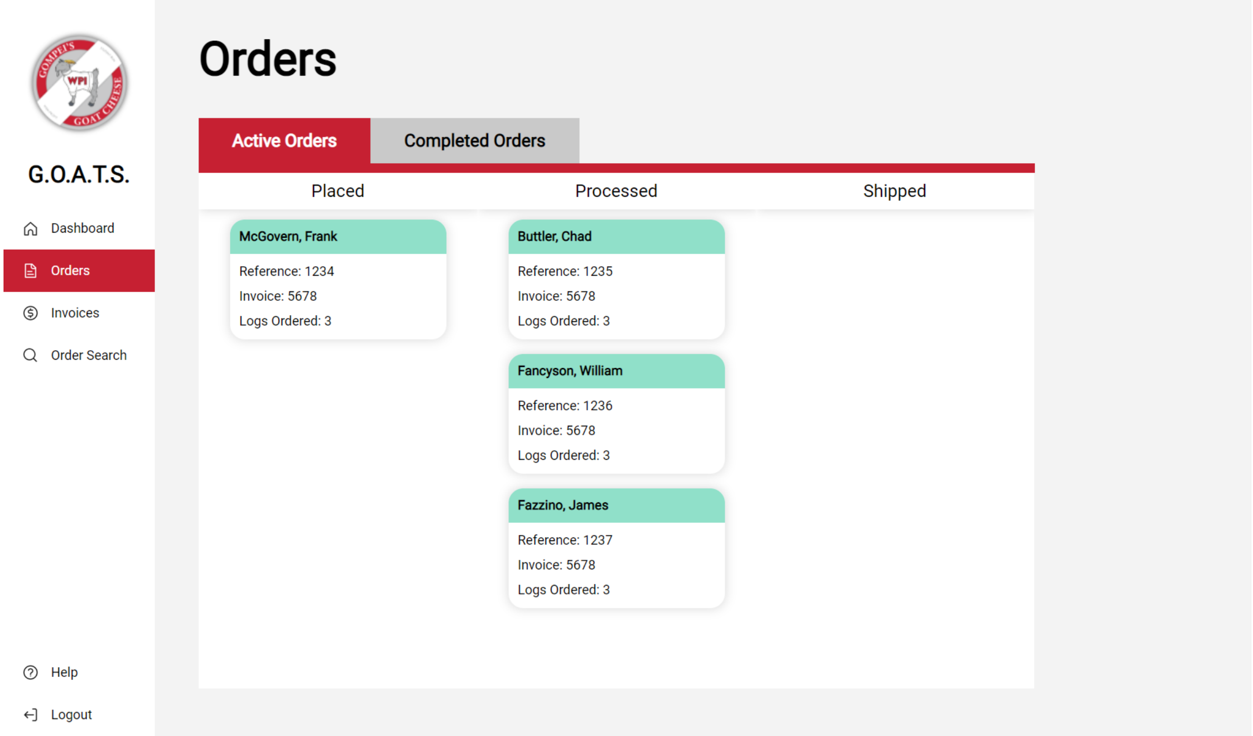
3 Orders Processed

0 Orders Shipped

0 Missing Invoices

3 Unpaid Invoices

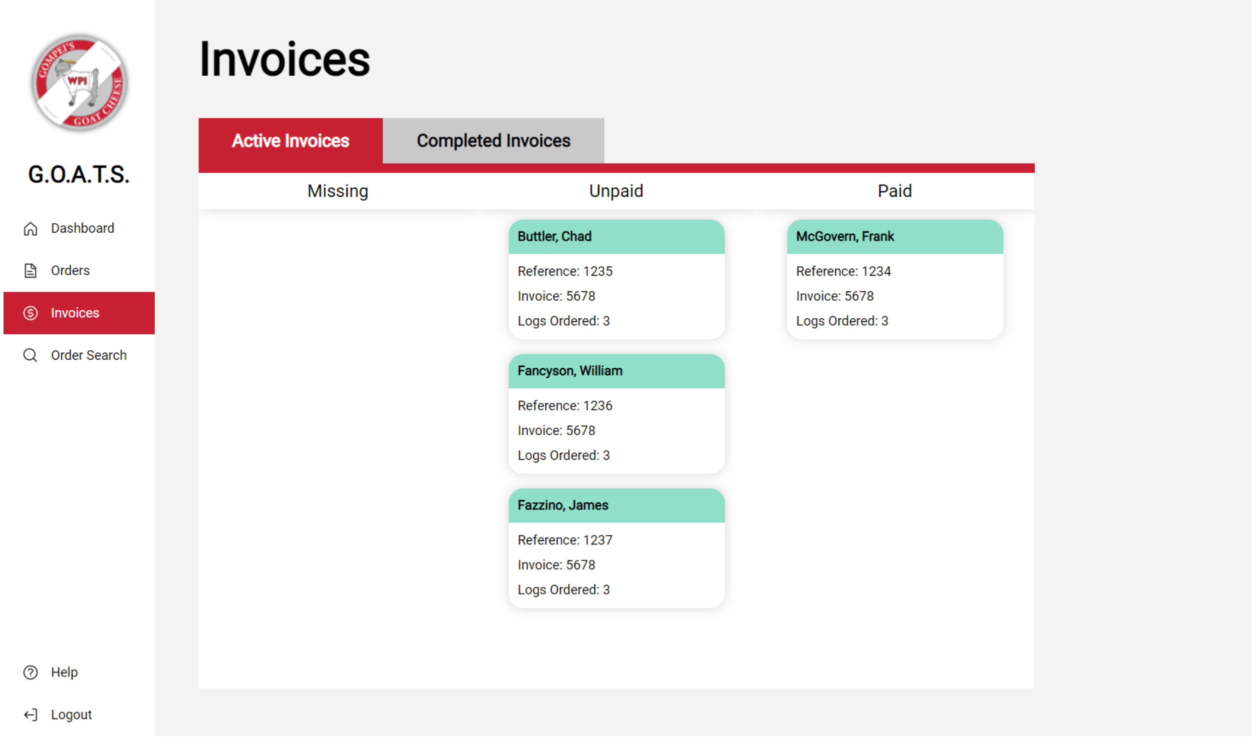
Figure 17. Developed System Dashboard Screenshot



Orders

Placed	Processed	Shipped
McGovern, Frank Reference: 1234 Invoice: 5678 Logs Ordered: 3	**Buttler, Chad** Reference: 1235 Invoice: 5678 Logs Ordered: 3	
	Fancyson, William Reference: 1236 Invoice: 5678 Logs Ordered: 3	
	Fazzino, James Reference: 1237 Invoice: 5678 Logs Ordered: 3	

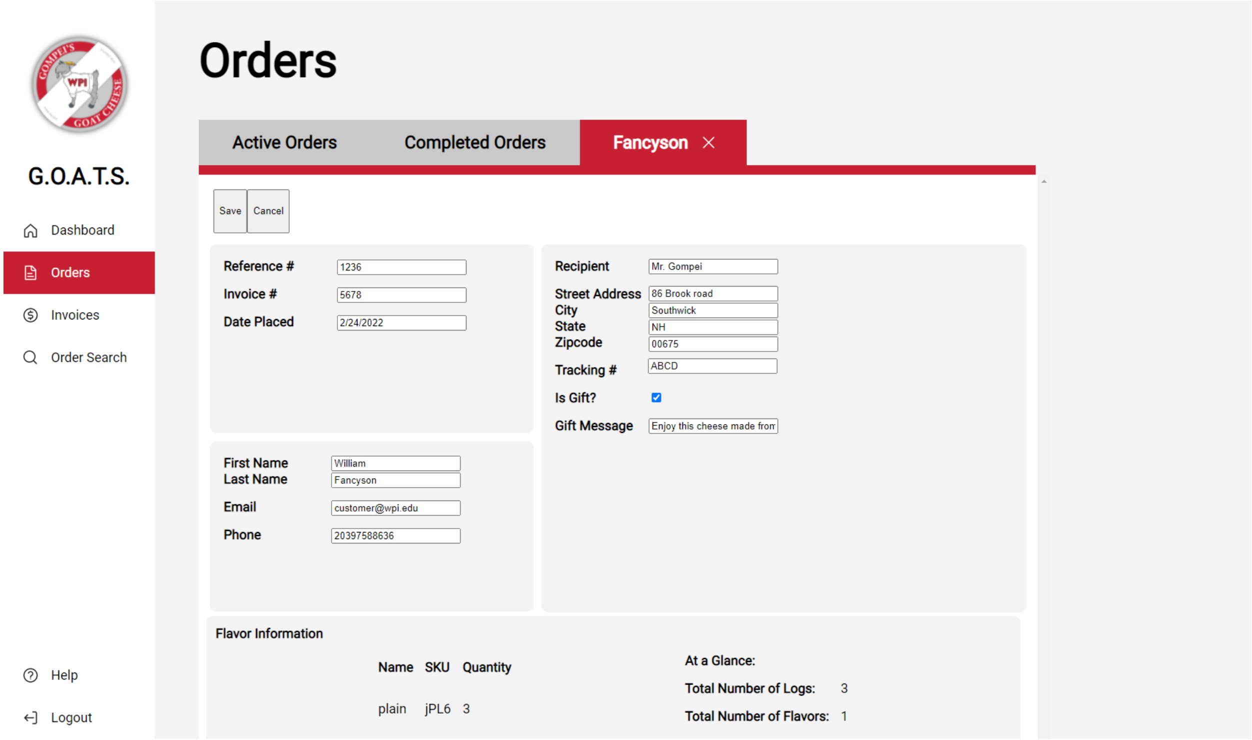
Figure 18. Developed System Active Orders Screenshot



Invoices

Missing	Unpaid	Paid
	Buttler, Chad Reference: 1235 Invoice: 5678 Logs Ordered: 3	McGovern, Frank Reference: 1234 Invoice: 5678 Logs Ordered: 3
	Fancyson, William Reference: 1236 Invoice: 5678 Logs Ordered: 3	
	Fazzino, James Reference: 1237 Invoice: 5678 Logs Ordered: 3	

Figure 19. Developed System Active Invoices Screenshot



Orders

Reference # 1236

Invoice # 5678

Date Placed 2/24/2022

First Name William

Last Name Fancyson

Email customer@wpi.edu

Phone 20397588636

Recipient Mr. Gompei

Street Address 86 Brook road

City Southwick

State NH

Zipcode 00675

Tracking # ABCD

Is Gift? ☑

Gift Message Enjoy this cheese made from

Flavor Information

Name	SKU	Quantity
plain	jPL6	3

At a Glance:

Total Number of Logs: 3

Total Number of Flavors: 1

Figure 20. Developed System Order Example Screenshot

Final Thoughts & Future Work

This paper outlines the start of a project that will continue into the future and go through multiple iterations before reaching its ultimate end-state. We believe that the completed system will provide Gompei's Goat Cheese with benefits such as improved transparency, centralized order and invoice tracking, and more effective communication with their partner farm. As a project team we completed research with stakeholders from Gompei's Goat Cheese and their partner farm, created system requirements to drive functionality and capabilities, designed mockups, started building the front end of the site, and created documentation to help anyone continuing the project in the future.

In true Gompei's Goat Cheese fashion, we came up with a clever name for the system:

Figure 21. Gompei's Operations and Accounting Tracking System Logo.

Next Steps

Due to the nature of system prototyping, the “prototype is a “quick and dirty” version of the system and provides minimal features” (Dennis, 2014). The next step in the future for this project is to conduct user tests. “Following reaction and comments from the users, the developers reanalyze, redesign, and reimplement a second prototype that corrects deficiencies and adds more features. This cycle continues until the analysts, users, and sponsors agree that the prototype provides enough functionality to be installed and used in the organization (Dennis, 2014).

General Development

The front-end development of the site has only just begun. The list below outlines what functionality was completed, what functionality is currently in progress, and what needs to be implemented. So far, the system only uses local data for testing purposes and must be connected with the database using the GraphQL API.

Functionality Completed:

- View Completed Orders
- View Active Orders

Functionality in Progress:

- View Order Information
- Update Order Information

Functionality to Implement:

- Add User
- Remove User
- Log In
- Log Out
- Record New Order
- Delete Order
- Update Invoice Information
- View Invoice
- View Customers
- Search Orders
- Copy Text Information

Reducing Redundancy

One of the goals that the project hoped to accomplish was to find a way to reduce redundancy. In the implementation of the new system, the operations team enters order information by hand, and then the farm receives an email, notifying them of a new order. Compared to GGC's current system, this reduces the number of times the operations team interacts with order data –they no longer have to send the same order information to the farm. In the future, however, the new system has great automation potential. One idea would be for the system to process a file that contains the order data that the operations team receives from Formsite. This would further reduce the operations team's workload.

Westfield Farm Perspective

The Westfield Farm side of the system has not been developed, so this will be another part in the future of this project. While it will be very similar, we want to make sure that a farm user does not have all of the same functionality that a GGC user would have. The farm only needs to be able to update the order information with details like their invoice number and the shipment tracking number. The farm will also only need to see the shared order and invoice details, and not internal GGC information like customer data and finances.

Accessibility

Another element to consider during future work is accessibility. While we did our best when keeping to good design practices, there are many things that we were not able to include in the system to make it accessible. The system will need to be developed further to include potential features such as a high contrast mode and the ability to magnify or change text size. Another task to consider completing to make the system more accessible is to make sure screens in the system will be properly read by a speech to text extension.

Final Thoughts

Within the last four years, Gompei's Goat Cheese has become a pillar of the WPI community. From fundraisers to raffles to even trivia nights with The WPI Business School, GGC has become more involved in campus events every day. We want to see this small business beat the odds and make it past its 15 year anniversary. To do this, we hope that this project will be the foundation and inspiration for future projects to be completed with Gompei's Goat Cheese.

Gompei's Goat Cheese may be the first student-run, non-profit business at WPI, but we know it will not be the last. Our hope is that this project may inspire others to pursue their own student-run business endeavors so that they can apply their knowledge they are gaining from WPI classes in real time while also "be[ing] a part of something meaningful that impacts [their] community" (Gompei's, 2022).

References

Better Business Bureau (2019). *10 Ways Small Businesses Benefit Their Local Communities*. Medium.com. https://medium.com/@BBBNWP/10-ways-small-businesses-benefit-their-local-communities-7273380c90a9

Dennis, A., Wixom, B. H., & Roth, R. M. (2014). *Systems Analysis and Design* (6th ed.). Wiley.

Fisher, K. (2022, February 8). *What is Cloud ERP and How Does It Work?* Oracle NetSuite. https://www.netsuite.com/portal/resource/articles/erp/cloud-erp.shtml

Ghosh, R. (2012). *A comprehensive study on ERP failures stressing on reluctance to change as a cause of failure*. Journal of Marketing and Management, 3(1), 123-134. http://ezproxy.wpi.edu/login?url=https://www.proquest.com/scholarly-journals/comprehensive-study-on-erp-failures-stressing/docview/1736298229/se-2?accountid=29120

Gompei's Goat Cheese. (2022). *Gompei's Goat Cheese*. WPI. https://www.wpi.edu/alumni/benefits/merchandise/gompei-goat-cheese

Normand, R. (2019, July 16). *4 Reasons Why Small Businesses Succeed.* Zenbusiness. https://www.zenbusiness.com/blog/reasons-startups-succeed/#:~:text=Positive%2C%20Committed%2C%20Persistent%20and%20Patient,organization%20and%20make%20it%20efficient.

Oracle (2022). *What is an ERP?* Oracle.com. https://www.oracle.com/erp/what-is-erp/

Sims, R. (2009). *Food, place and authenticity: local food and the sustainable tourism experience*. Taylor Francis Online. https://www.tandfonline.com/doi/full/10.1080/09669580802359293

Small Business Administration. (2021) *Frequently Asked Questions*. https://cdn.advocacy.sba.gov/wp-content/uploads/2021/12/06095731/Small-Business-FAQ-Revised-December-2021.pdf

Appendices

Appendix A: Interview Consent

Informed Consent Agreement

For Participation in a Research Study: Interview

Project Team: Natalie Mohn, Chris Guerrette

Contact Information: ggcerp@wpi.edu

Title of Research Study
Development of Cloud-Based Enterprise Resource Planning System for Gompei's Goat Cheese

Introduction
You are being asked to participate in a study researching the operations of Gompei's Goat Cheese. Before you agree, however, you must be fully informed about the purpose of the study, the procedures to be followed, and any benefits, risks or discomfort that you may experience as a result of your participation. This form presents information about the study so that you may make a fully informed decision regarding your participation.

Study Purpose
The purpose of this study is to learn more about the current operations of Gompei's Goat Cheese, identify the areas where it performs well, and where there is room for improvement. This information will be used to inform the creation of a cloud-based enterprise resource planning system.

Study Procedure
You will participate in a 20-30 minute interview with the members of the research team in person or over zoom if necessary (up to the discretion of the interviewee - preference of in person). The interview will be with the two team members, one leading the interview by questioning and the other will be listening, taking notes, and asking follow up questions if needed. Interview time and dates will be of your convenience. If the interview will be over zoom, a link will be sent by email once an interview time is established. This interview is voluntary and you can choose not to answer any questions you do not wish to answer, and can end the interview at any time.

Risks to Study Participants
There are no foreseeable risks or discomfort to you.

Benefits to Research Participants and Others

This study will inform the development of a system that will greatly benefit the operations of Gompei's Goat Cheese. This system will improve the internal operations of GGC and communication with its supplier, Westfield Farm.

Confidentiality

No participants will be identified by name in this study, only by title in relation to GGC if we have your permission. Your comments will not be shared with anyone other than our project team. Constructive criticism is encouraged and will not negatively affect your relationship with GGC or Westfield Farm.

Voluntary Nature of Study

Your refusal to participate will not result in any penalty to you. If you feel uncomfortable once you begin the study, you may stop participating at any time.

By signing below, you acknowledge that you have been informed about and consent to be a participant in the study described above. Make sure that your questions are answered to your satisfaction before signing. You are entitled to retain a copy of this consent agreement.

____ ____________________

Study Participant Signature Date

____

Study Participant Name (Please print)

____

Signature of Person who explained this study

Appendix B: Interview Study Protocol

Introduction

Hi, my name is [Chris Guerrette/Natalie Mohn]. I am a student at Worcester Polytechnic Institute, and I will be walking you through the interview today. My colleague, [Chris/Natalie], is also here and will be taking notes and asking follow up questions during our session.

Study Purpose

The purpose of this study is to learn more about the current operations of Gompei's Goat Cheese, identify the areas where it performs well, and where there is room for improvement. This information will be used to inform the creation of a cloud-based enterprise resource planning system. The hope is that this system will streamline the operations process, all the way from the customers ordering cheese to the invoice payment.

Consent

Before we start, we would like to go through important parts of the interview information document (Interview Consent) that was sent to you via email earlier.

This interview will be voice recorded and transcribed for the purpose of analysis only. We would like to have the option to use your responses in our final report. This would mean that we would identify you by your title in relation to Gompei's Goat Cheese. Are you comfortable with this?

(Participant gives oral consent)

Do you have any questions about the interview today?

(Participant asks questions if they have any)

Are you comfortable with the study procedures and ready to move forward?

(Participant gives oral consent)

Interview Time

Our interview will last about 30 minutes. There are no right or wrong answers. We are interested in your ideas and opinions. At any time if you feel uncomfortable and wish to stop the interview, please let us know. Are you ready to start?

(Participant gives oral consent)

Great. Let's get started.

[Interviewer starts recording]

Possible Question to Guide Conversation

[See Appendix C]

End of Interview

Thank you. This marks the end of the interview. Do you have any questions for us?

Thank you very much for participating. Have a great day!

[Interviewer stops recording]

Appendix C: Interview Questions

Questions for Gompei's Goat Cheese Current and Past Employees

1. What is (was) your position in Gompei's Goat Cheese?

2. How do (did) you interact with the operations of GGC?
3. How long have you been using the current order tracking system at GGC?
    a. How often do you have to interact with it?
4. Can you describe your knowledge of the current process of how an order of GGC gets to the customer?
    a. Approximately how long does an order take to get sent to Westfield Farms from the time of the order?
5. What do you think is easy to use about the current system?
    a. Is the current system easy to use overall?
6. What do you like most about the current system?
7. What do you think needs improvement in the current system?
8. What specific aspects of the current system are difficult or complicated to use?
9. Is the current system difficult or easy to use overall?
10. What device do you prefer to use when interacting with the system? (Mobile or Desktop)
11. In general how confident do you feel about using new technology?
12. What would you like to see in a new operations system?
    a. Thinking about specific aspects like order tracking, inventory, analytics, operations status - is there anything that you would like to see improved in those categories?
13. Is there anything that we have not covered that you would like to discuss?
14. Do you have any questions about our project?

Questions for Westfield Farm Employee(s)

1. What is your position at Westfield Farms? What is your role?
2. How long have you been working with Gompei's Goat Cheese?
3. Can you describe your knowledge of the current process of how an order of GGC gets to the customer?
    a. What kind of system do you currently use to track goat cheese orders?
4. What do you think is easy to use about the current system?
    a. Is the current system easy to use overall?
5. What do you like most about the current system?
6. What do you think needs improvement in the current system?
7. What specific aspects of the current system are difficult or complicated to use?
8. Is the current system difficult or easy to use overall?
9. In general how confident do you feel about using new technology?
10. What would you like to see in a new operations system with GGC?
11. Is there anything that we have not covered that you would like to discuss?

Appendix D: System Use-Cases

Add User	
Participating Actors	GGC User
Entry Conditions	● User is a GGC User. ● Email is a real email.
Exit Criteria	The email is verified for system login. User has a role (GGC or Westfield Farm)
Flow of Events	1. GGC User requests to add a user. 2. GGC User types in new User’s email and submits request. 3. System adds email to a list of verified login emails, refreshes display.

Remove User	
Participating Actors	GGC User
Entry Conditions	● User is selected.
Exit Criteria	The email is no longer verified for system login.
Flow of Events	1. GGC User requests to remove a verified email. 2. System prompts GGC User to confirm removal. 3. GGC User confirms removal. 4. System removed email from the list of verified email logins, refreshes display.

Log In	
Participating Actors	User
Entry Conditions	User’s email is in the system.
Exit Criteria	User is logged in.
Flow of Events	1. User requests to log in. 2. System verifies user email and displays GGC’s dashboard.

Log Out	
Participating Actors	User
Entry Conditions	User is logged in.

Exit Criteria	User is logged out.
Flow of Events	1. User requests to log out. 2. System logs out User and shows the login screen.

Record New Order	
Participating Actors	GGC User
Entry Conditions	All required order information has been put in.
Exit Criteria	Order has been recorded and an email is sent about the new order.
Flow of Events	1. GGC requests to add an order. 2. System adds order to the system and refreshes the display. 3. System sends an email to GGC and the Partner Farm that a new order has been added.

Delete Order	
Participating Actors	GGC User
Entry Conditions	An order is selected.
Exit Criteria	The selected order and associated invoice is deleted.
Flow of Events	1. GGC requests to delete an order. 2. System prompts GGC to confirm that they want to delete the order. 3. GGC responds to prompt on screen. 4. System deletes the order if GGC confirms the prompt and refreshes the screen.

Update Order Information	
Participating Actors	User
Entry Conditions	Order Information is being viewed and new information has been put in.
Exit Criteria	Order Information is updated.
Flow of Events	1. User requests to update order information. 2. The system updates the order information and refreshes the screen.

Update Invoice Information

Participating Actors	User
Entry Conditions	Invoice Information is being viewed and new information has been put in.
Exit Criteria	Order Information is updated.
Flow of Events	3. User requests to update order information. 4. The system updates the order information and refreshes the screen.

View Active Orders	
Participating Actors	User
Entry Conditions	None
Exit Criteria	The current orders are displayed.
Flow of Events	1. User requests to view the current active orders. 2. The system displays the current active orders.

View Completed Orders	
Participating Actors	User
Entry Conditions	None
Exit Criteria	All the orders are displayed.
Flow of Events	1. User requests to view completed orders. 2. The system displays the completed orders.

View Order Information	
Participating Actors	User
Entry Conditions	Order is selected.
Exit Criteria	The order information is displayed.
Flow of Events	1. User requests that the order information be displayed. 2. The system displays the order information.

View Invoice	
Participating Actors	User

Entry Conditions	Order is selected and has an invoice.
Exit Criteria	Invoice PDF is displayed.
Flow of Events	1. Userrequests to view invoice. 2. System displays Invoice PDF in a new window.

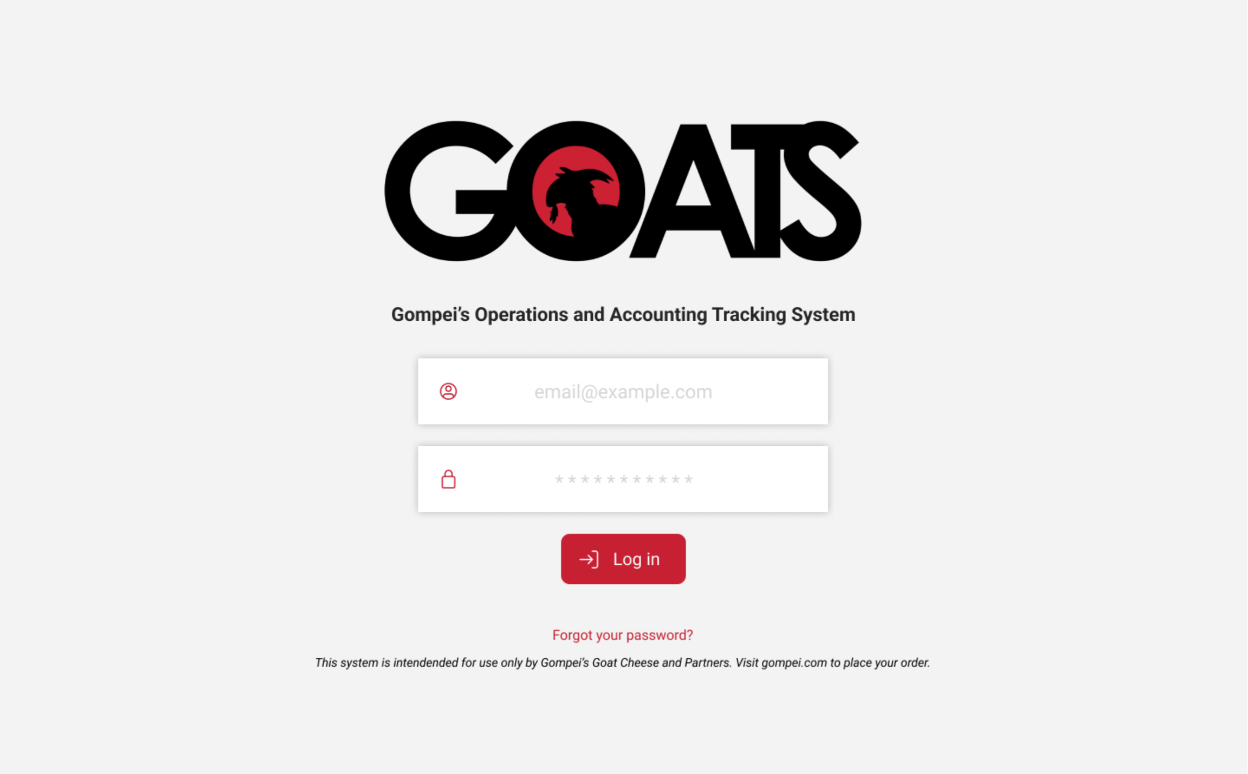
Search Orders	
Participating Actors	User
Entry Conditions	Search parameters are specified.
Exit Criteria	Orders matching search parameters are displayed.
Flow of Events	1. User requests to search for orders matching a set of parameters. 2. The system displays all orders matching the search parameters.

View Customers	
Participating Actors	GGC User
Entry Conditions	None
Exit Criteria	List of customers is displayed
Flow of Events	1. GGC requests to view the list of customers 2. The system displays all orders matching the search parameters.

Copy Text Information	
Participating Actors	User
Entry Conditions	None
Exit Criteria	Text is copied to clipboard
Flow of Events	1. User requests to copy nearby text. 2. The system copies that text to the users clipboard.

Appendix E: System Mockups

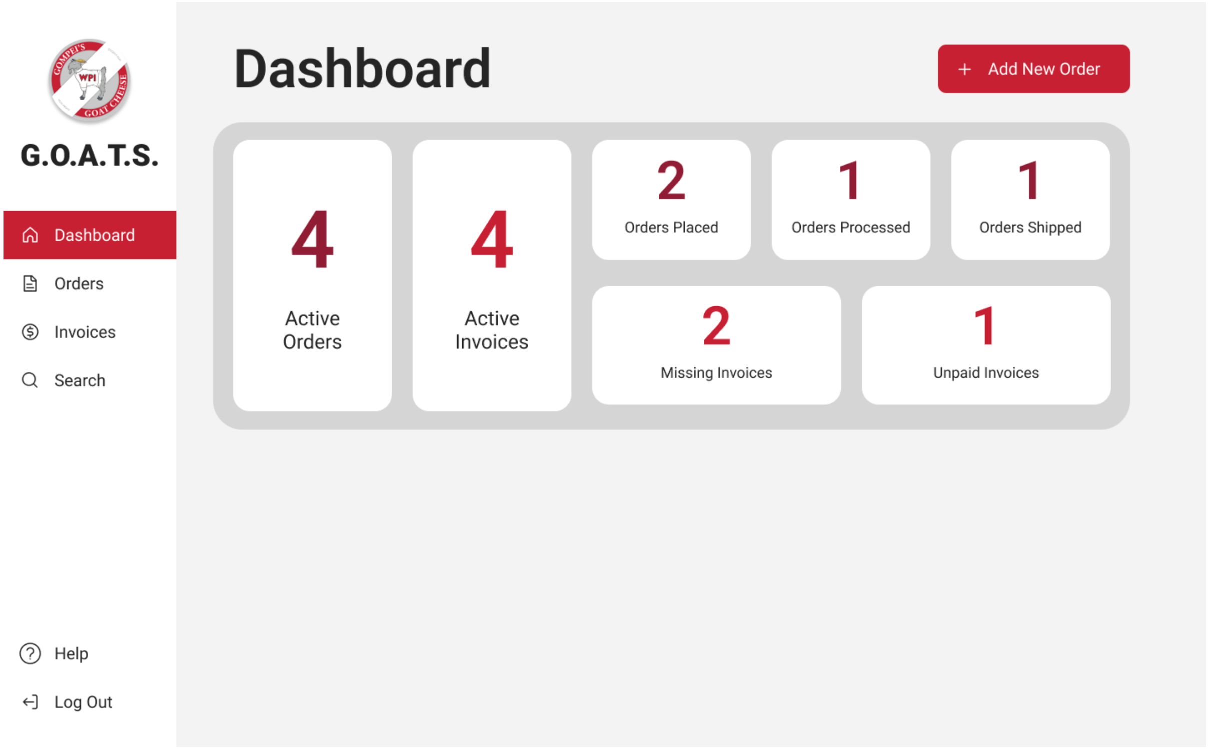
Login



Gompei's Operations and Accounting Tracking System

This system is intendended for use only by Gompei's Goat Cheese and Partners. Visit gompei.com to place your order.

Dashboard



Dashboard

4 Active Orders

4 Active Invoices

2 Orders Placed

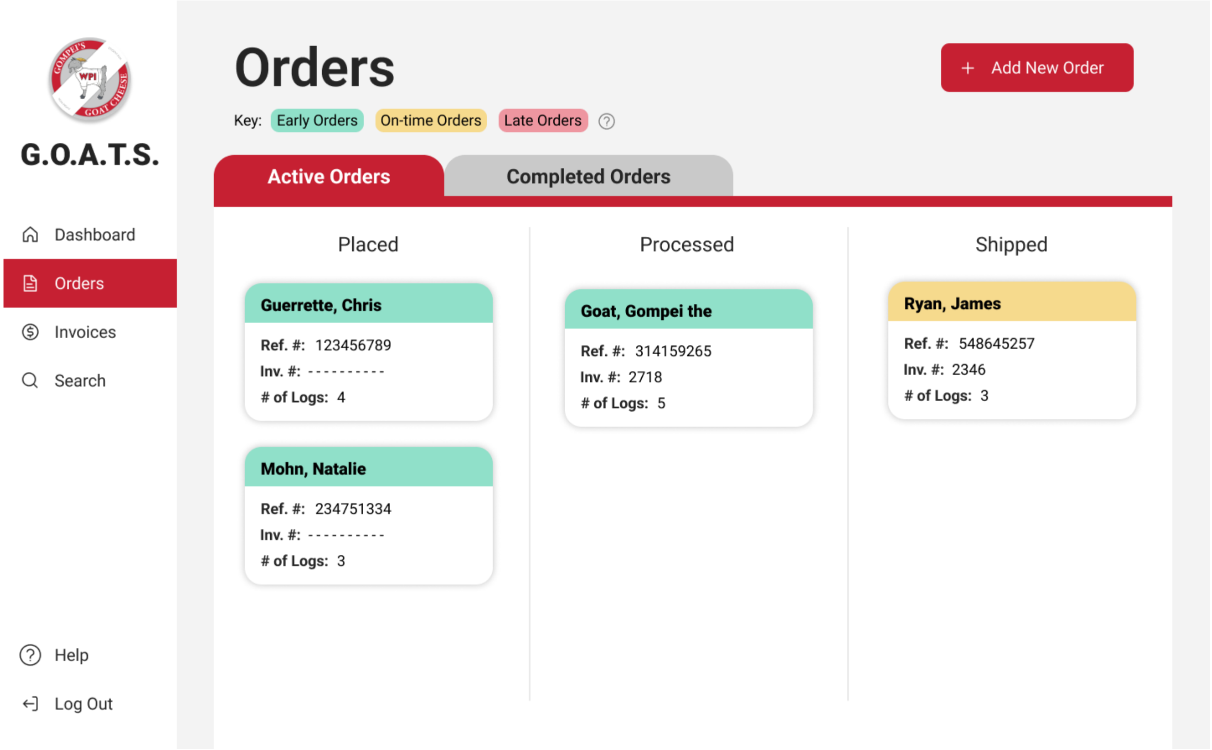
1 Orders Processed

1 Orders Shipped

2 Missing Invoices

1 Unpaid Invoices

Active Orders



Orders

Key: Early Orders On-time Orders Late Orders

Placed

Guerrette, Chris
Ref. #: 123456789
Inv. #: ----------
# of Logs: 4

Mohn, Natalie
Ref. #: 234751334
Inv. #: ----------
# of Logs: 3

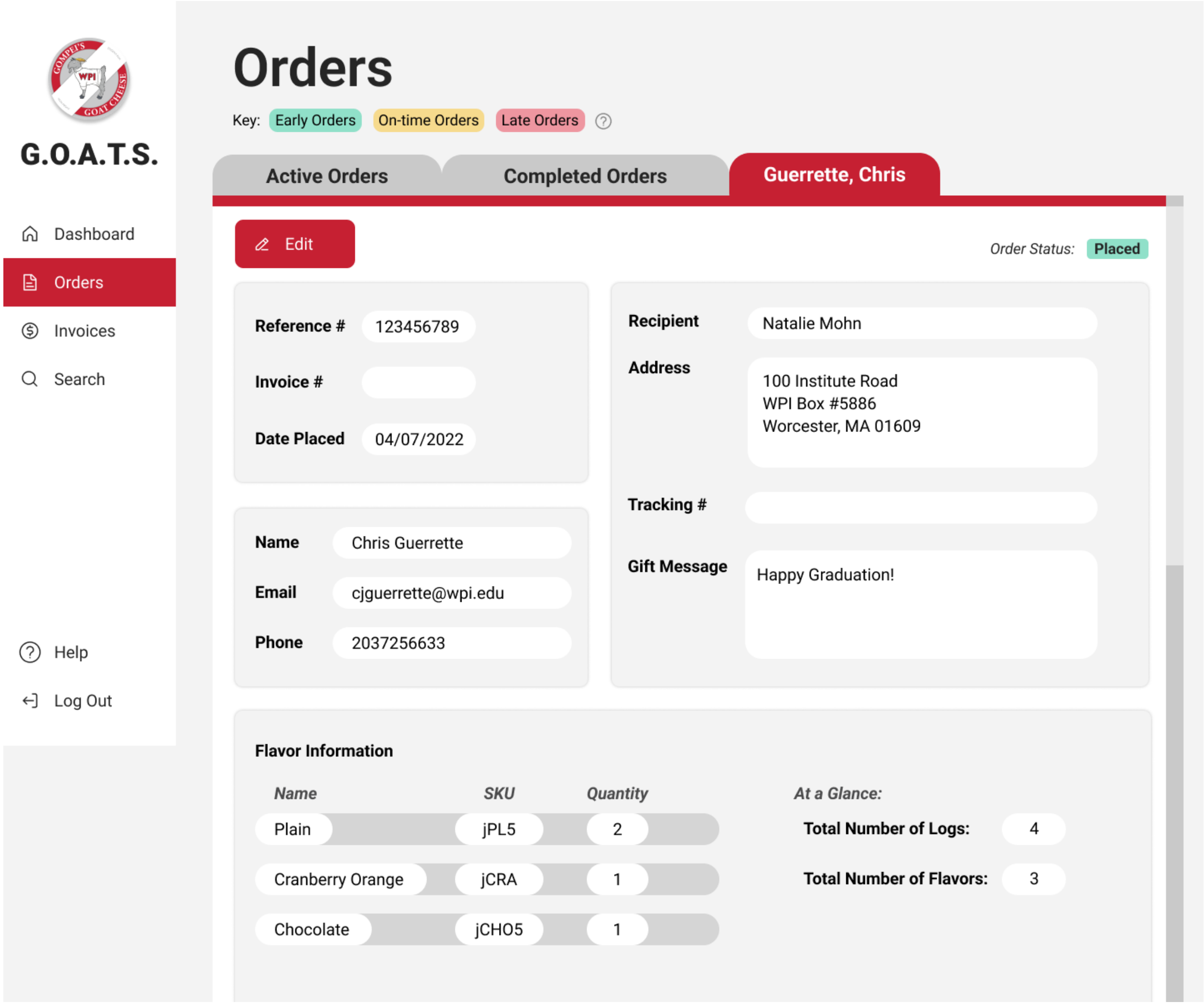
Processed

Goat, Gompei the
Ref. #: 314159265
Inv. #: 2718
# of Logs: 5

Shipped

Ryan, James
Ref. #: 548645257
Inv. #: 2346
# of Logs: 3

View Order



Orders

Key: Early Orders On-time Orders Late Orders

Order Status: **Placed**

Reference #	123456789
Invoice #	
Date Placed	04/07/2022

Name	Chris Guerrette
Email	cjguerrette@wpi.edu
Phone	2037256633

Recipient	Natalie Mohn
Address	100 Institute Road WPI Box #5886 Worcester, MA 01609
Tracking #	
Gift Message	Happy Graduation!

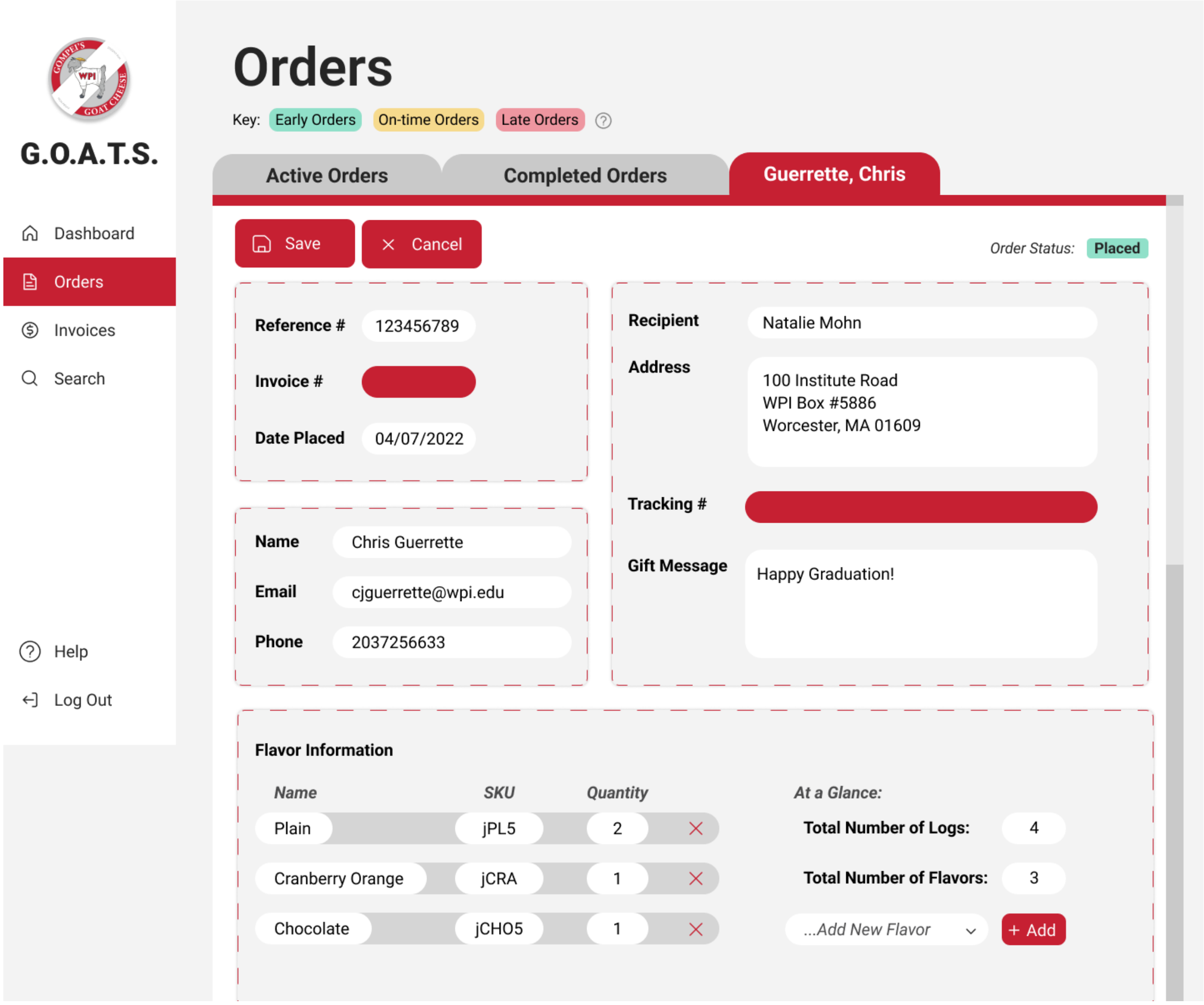
Flavor Information

Name	*SKU*	*Quantity*
Plain	jPL5	2
Cranberry Orange	jCRA	1
Chocolate	jCHO5	1

At a Glance:

Total Number of Logs:	4
Total Number of Flavors:	3

Edit Order



G.O.A.T.S.

- Dashboard
- Orders
- Invoices
- Search
- Help
- Log Out

Orders

Key: Early Orders On-time Orders Late Orders

Active Orders | Completed Orders | Guerrette, Chris

Save | Cancel

Order Status: **Placed**

Reference #	123456789
Invoice #	
Date Placed	04/07/2022

Name	Chris Guerrette
Email	cjguerrette@wpi.edu
Phone	2037256633

Recipient	Natalie Mohn
Address	100 Institute Road WPI Box #5886 Worcester, MA 01609
Tracking #	
Gift Message	Happy Graduation!

Flavor Information

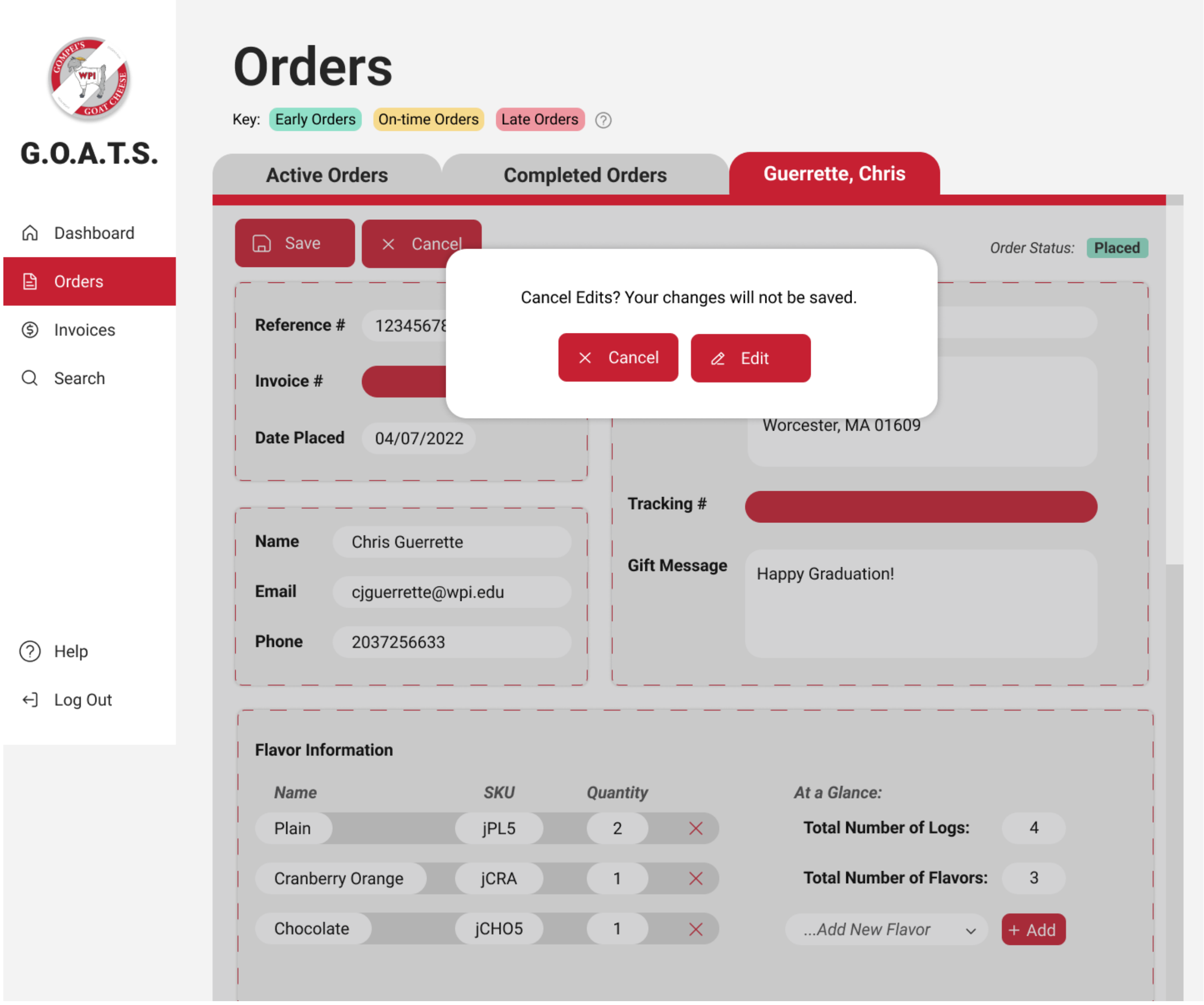
Name	*SKU*	*Quantity*	
Plain	jPL5	2	×
Cranberry Orange	jCRA	1	×
Chocolate	jCHO5	1	×

At a Glance:

Total Number of Logs:	4
Total Number of Flavors:	3

...Add New Flavor | + Add

Edit Order (Cancel)



Orders

Key: Early Orders On-time Orders Late Orders

Order Status: Placed

Cancel Edits? Your changes will not be saved.

Reference #	12345678
Invoice #	
Date Placed	04/07/2022

Worcester, MA 01609

Name	Chris Guerrette
Email	cjguerrette@wpi.edu
Phone	2037256633
Tracking #	
Gift Message	Happy Graduation!

Flavor Information

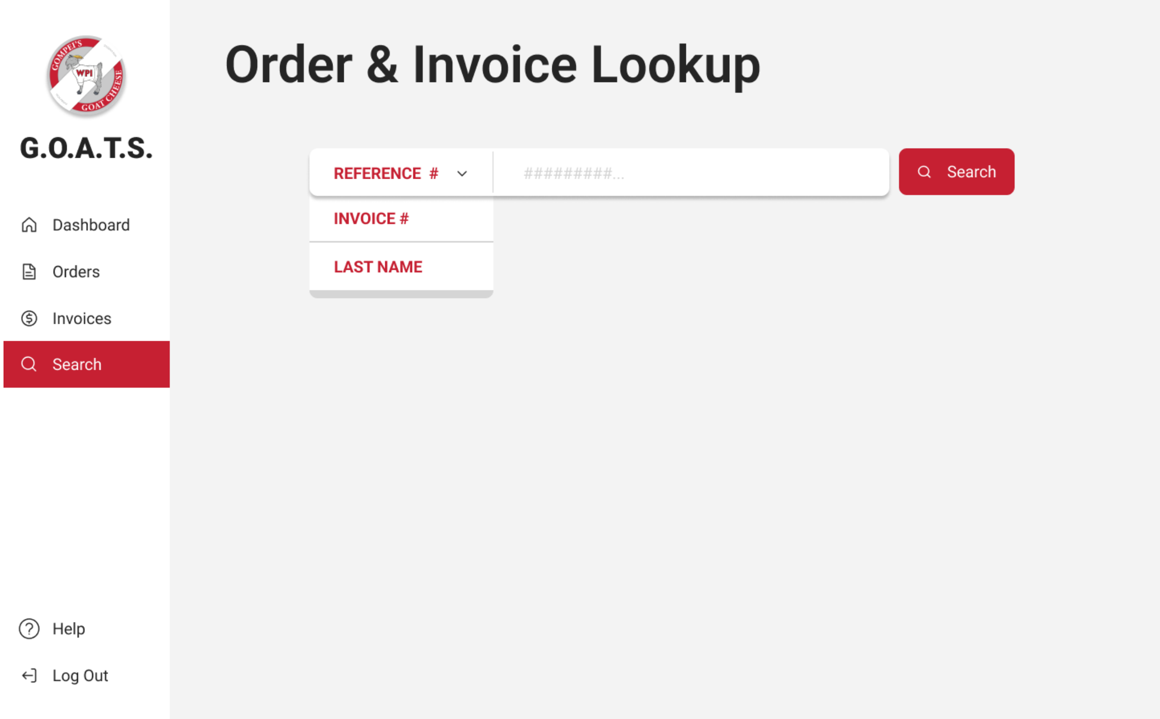
Name	*SKU*	*Quantity*
Plain	jPL5	2
Cranberry Orange	jCRA	1
Chocolate	jCHO5	1

At a Glance:

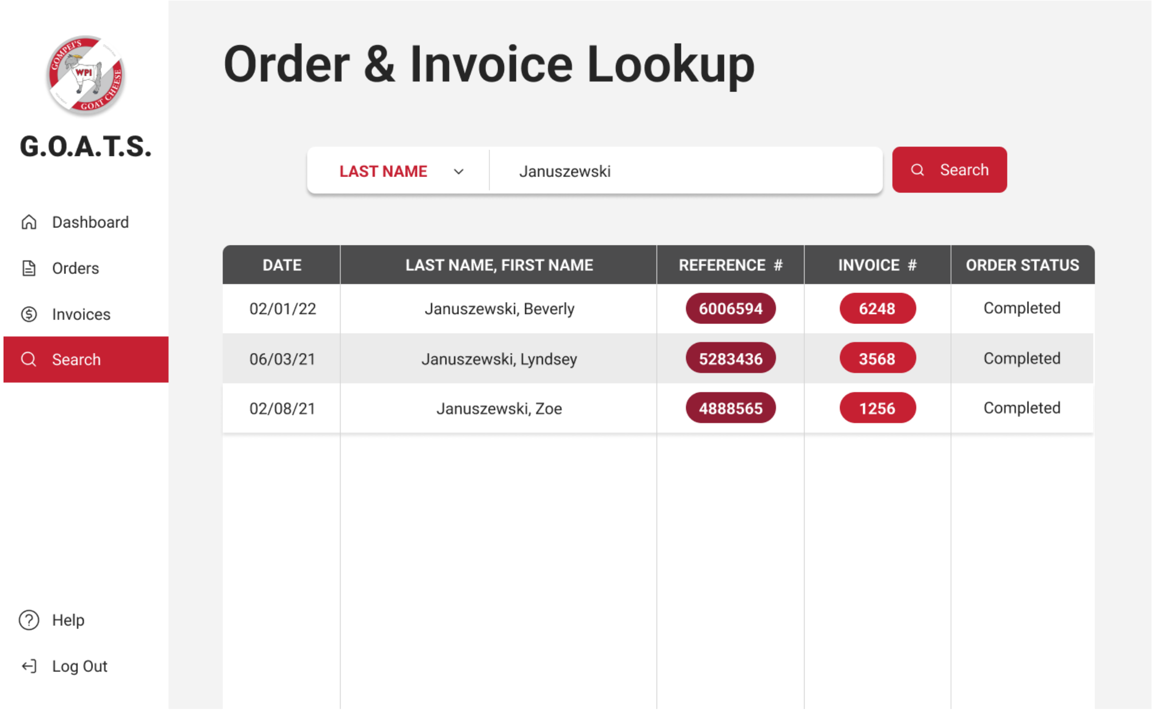
Total Number of Logs:	4
Total Number of Flavors:	3

...Add New Flavor

Order Lookup



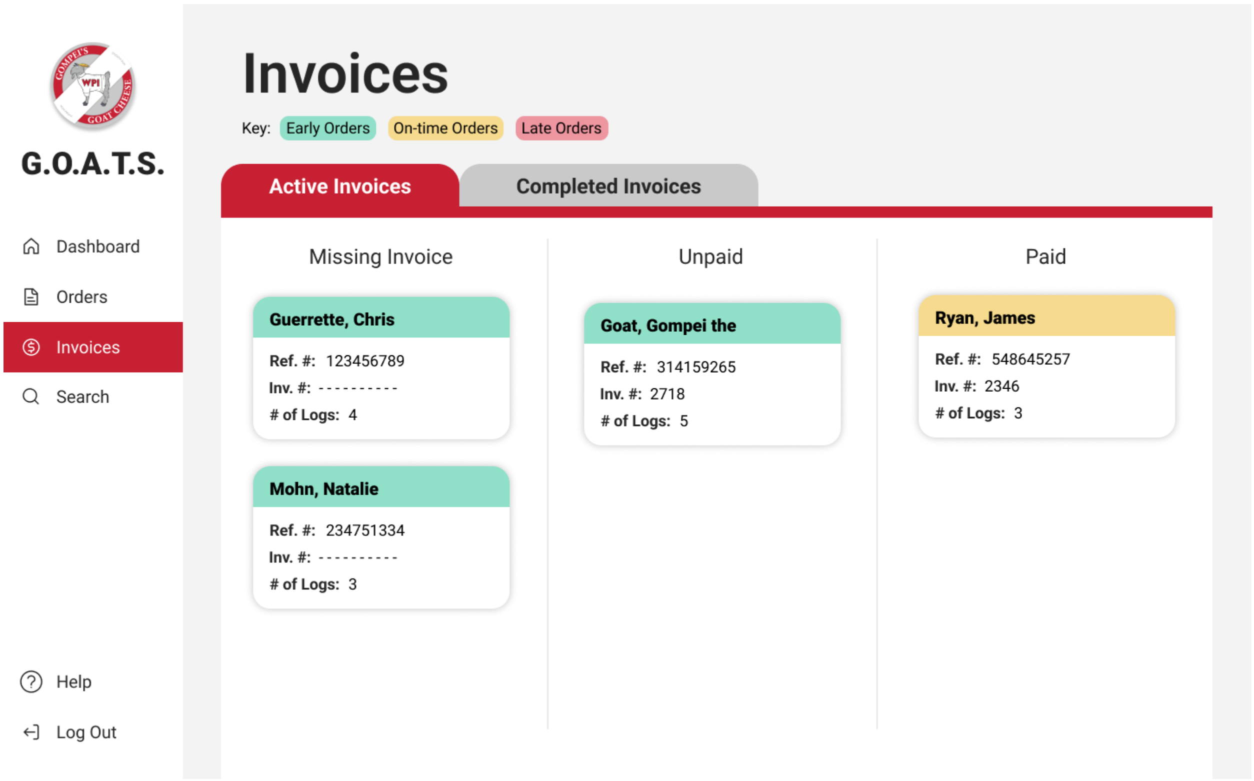
Order Lookup - Last Name Searched



LAST NAME: Januszewski

DATE	LAST NAME, FIRST NAME	REFERENCE #	INVOICE #	ORDER STATUS
02/01/22	Januszewski, Beverly	6006594	6248	Completed
06/03/21	Januszewski, Lyndsey	5283436	3568	Completed
02/08/21	Januszewski, Zoe	4888565	1256	Completed

Active Invoices



Invoices

Missing Invoice	Unpaid	Paid
Guerrette, Chris **Ref. #:** 123456789 **Inv. #:** ---------- **# of Logs:** 4	**Goat, Gompei the** **Ref. #:** 314159265 **Inv. #:** 2718 **# of Logs:** 5	**Ryan, James** **Ref. #:** 548645257 **Inv. #:** 2346 **# of Logs:** 3
Mohn, Natalie **Ref. #:** 234751334 **Inv. #:** ---------- **# of Logs:** 3		

Appendix F: Support Documentation

Gompei's Goat Cheese ERP System Support Documentation:

Where can everything be found?

We created a project within Amazon Web Services' Amplify Studio to host everything and be the main point of entry for the project resources. We also used Figma to create the mockups for the site.

How to access information?

[redacted]

This section was removed for security reasons.

Once you login with the ggc accounts for each site, you can then easily share it with your personal emails.

Getting project code

Once you are signed into Amplify Studio, you should see a link in the top right corner that says *Local Setup Instructions*. Follow the steps there to pull the code from Amplify Studio. You may also need to install Amplify CLI.

What has been done?

Paper Documentation

In our paper, there are a lot of helpful diagrams and other documentation, including: System Use-Cases, Data Flow Diagrams, an Entity Relationship Diagram, a Data Dictionary, a User Flow Diagram, and System Mockups.

Figma Prototype

Currently, there is a basic hi-fi prototype created in Figma. If you “Present” the design in Figma using the top right play button, you can play around with what was created. Figma is a powerful tool to visualize what a system could look like, with the benefit that you do not need to code anything. It is a vector based program that functions similarly to Adobe Illustrator and XD. Check out the resources at the end of this document for some recommended tutorials for Figma.

Database Schema

The database that this project uses is DynamoDB which is a NoSQL database. This was chosen because it was the only database type that AWS would have integrated directly with the Amplify Studio project. For the future, it may make more sense to use Amazon RDS, and if that is the case then the API would need to be restructured as well. If you decide to migrate the project to an RDS, check out the Entity Relationship Diagram and Data Dictionary that is in the final report.

To continue with using the current database and API, only a few extra lines of code needs to be added to the React.js files, and you should be able to communicate directly with the database.

Started front-end development

The front end of the site is built as a React.js project, which means that it is a single-page application. React is great because it allows for an easier time for UI development and makes the app more flexible for change. All the React code is under the *src* folder in the project directory and we then created a *components* folder to hold the code for each different view that we have in the ERP. This is not required but it helps to keep things a bit more organized. In the *components* folder is also an *img* folder. This contains image assets for the site. The *img* folder is not directly under the *src* folder because we had an issue looking a directory up to find images, and the files in the *components* folder use the image assets. The outermost react component is in App.js.

App.js has some local “dummy data” in place to use for testing. This is an example of how the data could look but it will be different when actually querying the database.

When designing the component structure, we wanted to have a component for each view in the system. Within that view, the UI will need to change depending on what is happening. For instance, the Order view can either be looking at active orders, completed orders, or viewing information about a specific order. To keep track of the state of our Components, we created an enum. This enum would contain values that correspond to the different views needed for a component. On rendering the component, it would check to see what the view enum is and then draw the corresponding sub components. We’re not sure if there is a more proper way to do conditional renders like this but it seems to work well and is organized.

Currently all the CSS is in index.css.

What are some resources to continue working on it?

There are lots of great resources out there but here’s a list of some of the things that we used.

Figma:

Figma Tutorial: Prototyping
Figma Tutorial: Interactive Components
Figma Tutorial: Variants
Also explore the Figma YouTube channel for more.

React:
React.js Crash Course
Crash Course Github
Intro to React Tutorial
React Functional Components
React Hooks
Conditional Rendering
Conditional Rendering with Enum
Forms
Using React Hooks with Forms

AWS:
DynamoDB Beginer Guide
Amplify Basic App Tutorial
In Depth Amplify Studio App Tutorial
Graph QL API