PIZZA ORDERING SYSTEM PROJECT REPORT

SWE 3313 Fall Project 2018

NOVEMBER 6, 2018

KENNESAW STATE UNIVERSITY

Pizza Ordering System Project Report

SWE 3313 Fall 2018

Group Leader

Meghan Dowell

Group Members

Bayleigh Correll

William Diamanduros

Kyle Duncan

Lane Dyer

TABLE OF CONTENTS

Introduction

The SWE 3313 Final Project was introduced to the group at the beginning of the Fall 2018 semester as a semester long challenging project assigned in order to place the group members in a situation similar to that of a real software development project and allowed the group members to experience each step of software project development in depth. Through the entirety of the project, group members collaborated and carried out individual tasks, providing them an idea of what it is like to work on a software development team. Though the combined efforts of the group members, any difficulties or problems were overcome and a finished product that met all requirements was produced and submitted by the due date. Each group member was able to confidently submit their completed part of the project and finish the semester with newfound knowledge and experience of what it is like to work on a software development team to complete a project assigned by an outside source.

2. PROJECT PLAN

SWE 3313 Fall 2018 Project

2.1 Project Scope Statement

Project Name:	SWE 3313 Pizza Ordering System
Team Members:	Bayleigh Correll, Will Diamanduros, Meghan Dowell, Kyle Duncan, Lane Dyer
Team Leader:	Meghan Dowell
Due Date:	November 16, 2018
Date Last Modified:	October 16, 2018

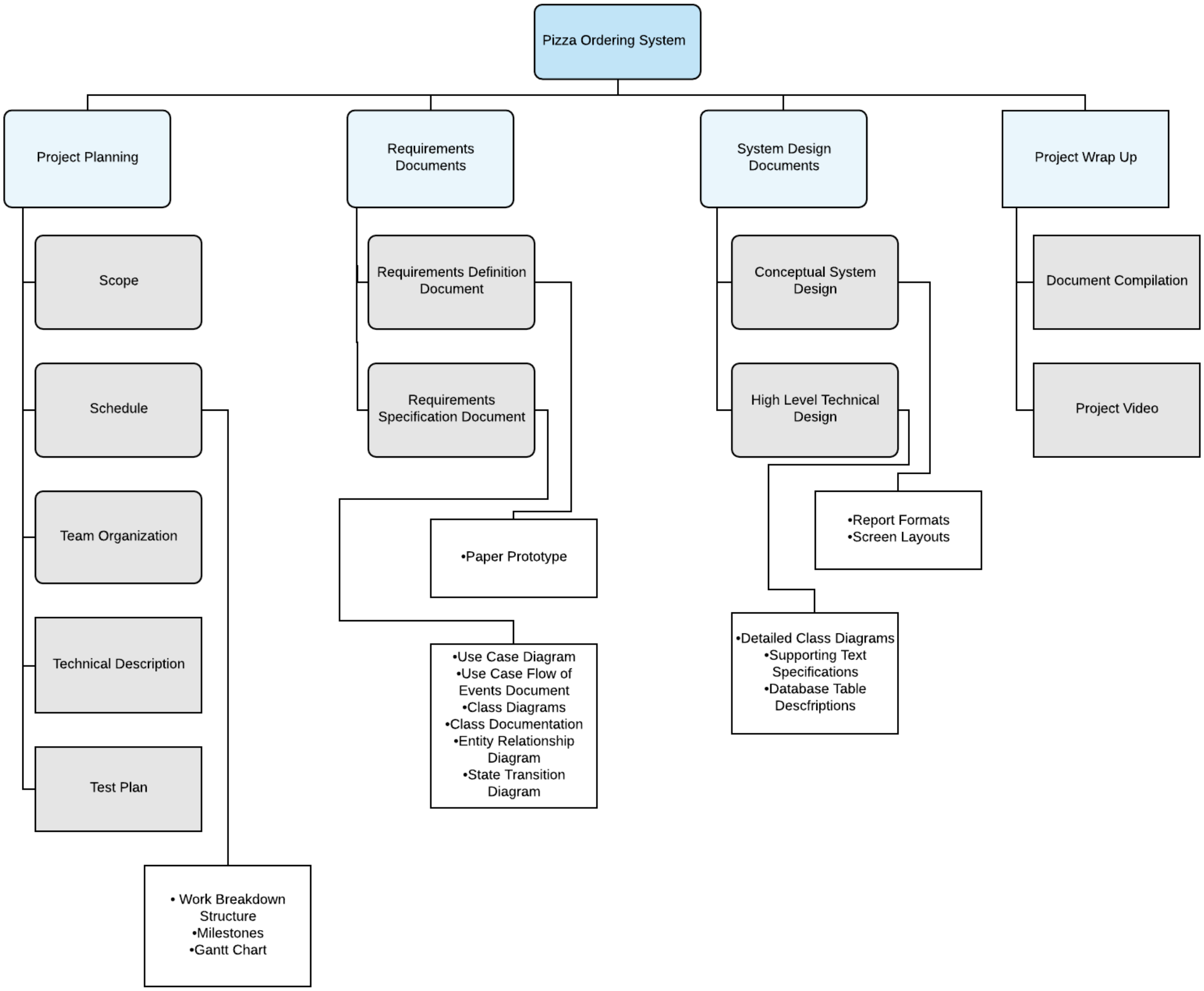
Project Description:	Create an ordering system that allows customers to order customized pizza for take-out or delivery, record customer information, order receipt, and payment options.
Project Requirements:	• Allow customers to choose pick-up or delivery • Allow customers to order pizza with various constraints such as size, toppings, crust, etc. • Create record for new customers via their phone number • Allow and process payments via check, cash, or credit card • Save established customers information • Print receipt with order, customer information, amount due, dining method, and a space for signature for card users • Implement a GUI including the above
Project Deliverables	• Project Report: Project Planning Documents, Requirements Documents, System Design Documents • Polished Prototype • Project Video
Strategy	For this project, the group will split the tasks equally among group members and partake in meetings at least twice a week for project discussion and/or help other team members with any problems they come across
Acceptance Criteria	• Meet all requirements • Provide all deliverables • Finish and submit project on time
Constraints	• Non-Uniform programing language experience among group members • Differing schedules among group members may interfere with planning meetings • Project must be completed and turned in by November 16, 2018

2.2 Project Schedule/Milestones

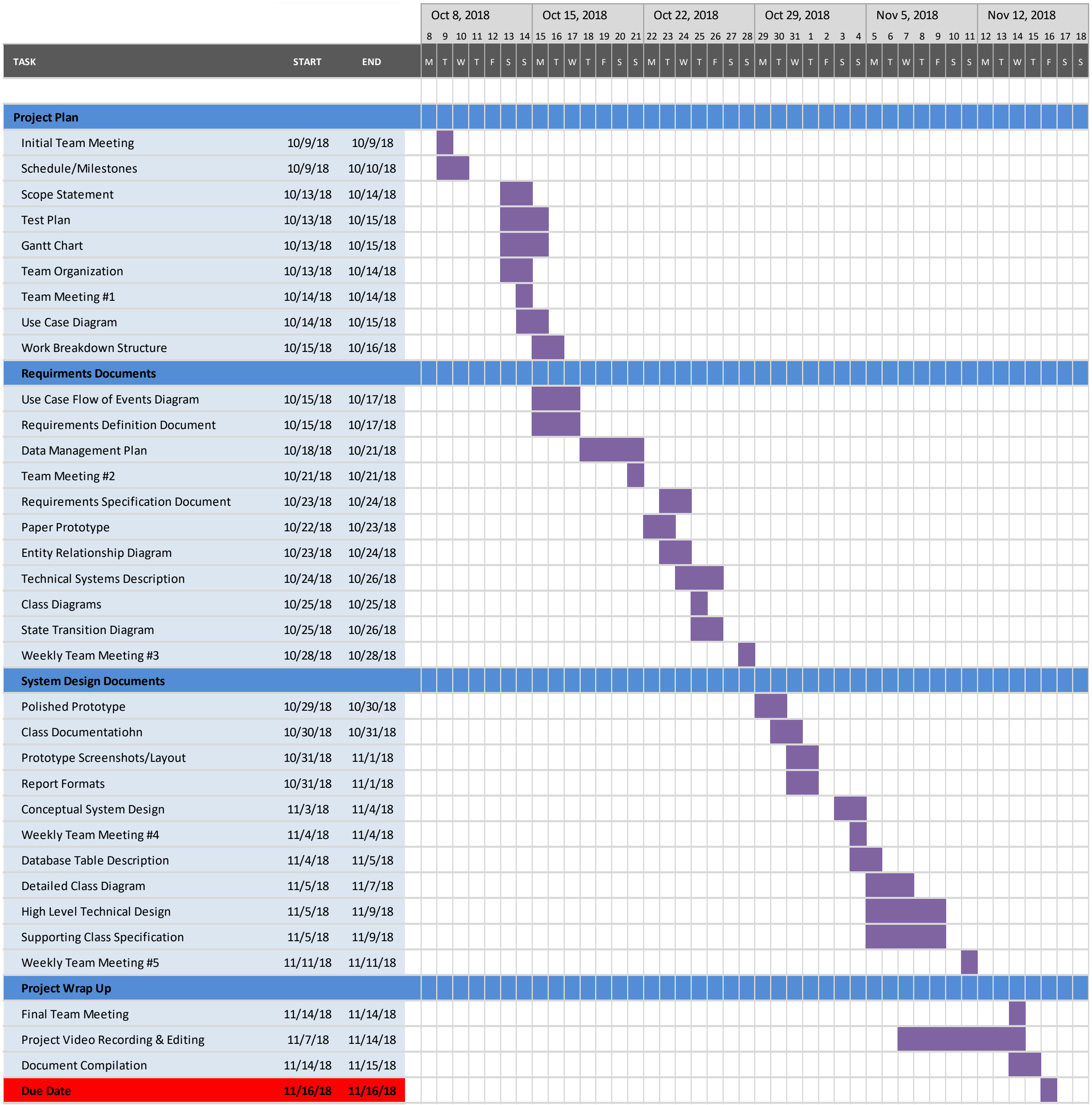
Milestone	Deliverables	Status	Start Date	End Date
Project Plan	•Project Scope Statement •Project Schedule •Work Breakdown Structure •Milestones •Gantt Chart •Team Organization and Resumes •Technical Systems Description •Data Management Plan •Test Plan	Complete	10/9/18	10/25/18
Requirements Documents	•Requirements Definition Document •Paper Prototype •Requirements Specification Document •Use Case Diagram •Use Case Flow of Events Diagram •Entity Relationship Diagram •State Transition Diagram	Complete	10/15/18	10/26/18
System Design Documents	•Conceptual System Design •Report Formats •Screen Layouts •High Level Technical Design •Detailed Class Diagrams •Class Documentation	Complete	10/28/18	11/9/18

	•Supporting Text Specifications •Database table descriptions •Polished Prototype			
Project Wrap Up	•Weekly Status Reports •Meeting Schedule and Attendance • Project video • Report Compilation	Complete	11/11/18	11/15/11

2.2.1 Work Breakdown Structure



2.2.2 Gantt Chart



TASK	START	END
Project Plan		
Initial Team Meeting	10/9/18	10/9/18
Schedule/Milestones	10/9/18	10/10/18
Scope Statement	10/13/18	10/14/18
Test Plan	10/13/18	10/15/18
Gantt Chart	10/13/18	10/15/18
Team Organization	10/13/18	10/14/18
Team Meeting #1	10/14/18	10/14/18
Use Case Diagram	10/14/18	10/15/18
Work Breakdown Structure	10/15/18	10/16/18
Requirments Documents		
Use Case Flow of Events Diagram	10/15/18	10/17/18
Requirements Definition Document	10/15/18	10/17/18
Data Management Plan	10/18/18	10/21/18
Team Meeting #2	10/21/18	10/21/18
Requirements Specification Document	10/23/18	10/24/18
Paper Prototype	10/22/18	10/23/18
Entity Relationship Diagram	10/23/18	10/24/18
Technical Systems Description	10/24/18	10/26/18
Class Diagrams	10/25/18	10/25/18
State Transition Diagram	10/25/18	10/26/18
Weekly Team Meeting #3	10/28/18	10/28/18
System Design Documents		
Polished Prototype	10/29/18	10/30/18
Class Documentatiohn	10/30/18	10/31/18
Prototype Screenshots/Layout	10/31/18	11/1/18
Report Formats	10/31/18	11/1/18
Conceptual System Design	11/3/18	11/4/18
Weekly Team Meeting #4	11/4/18	11/4/18
Database Table Description	11/4/18	11/5/18
Detailed Class Diagram	11/5/18	11/7/18
High Level Technical Design	11/5/18	11/9/18
Supporting Class Specification	11/5/18	11/9/18
Weekly Team Meeting #5	11/11/18	11/11/18
Project Wrap Up		
Final Team Meeting	11/14/18	11/14/18
Project Video Recording & Editing	11/7/18	11/14/18
Document Compilation	11/14/18	11/15/18
Due Date	**11/16/18**	**11/16/18**

2.3 Team Organization

Group Member	Tasks
Bayleigh Correll	• Scope Statement • Work Breakdown Structure • Team Organization • Milestones • Gantt Chart • Weekly Status Reports
Will Diamanduros	• Test Plan • Use Case Diagram • Use Case Flow of Events Document • Report Formats • Software Requirements Specification Document • Conceptual System Design
Meghan Dowell	• Group Leader • Requirements Definition Document • Paper Prototype • Screen Layouts/Shots • Entity Relationship Diagram • State Transition Diagram • Video Editing
Kyle Duncan	• Class Diagrams • Class Documentation • Supporting Text Specification
Lane Dyer	• Technical System Description • Data Management Plan • Database Table Descriptions

BC

BAYLEIGH CORRELL

4795 Shallow Ridge Road | Kennesaw, Georgia404-551-9346 | bcorrel2@students.kennesaw.edu

OBJECTIVE

Improve software engineering skills including planning/scheduling, requirement specification and documentation, development, testing, etc.

SKILLS

- Typing
- Organization
- Computer Software (Microsoft Office, Excel, Photoshop, etc.)
- Computer Hardware
- JAVA Coding
- Basic Database Programming
- Basic C++ Coding

ADDITIONAL CONACT INFORMATION

OTHER EMAIL

- Baycorrell@gmail.com

HOME PHONE

- 404-551-9364

EXPERIENCE

CAMP COUNSELOR • CLUB SCI KIDS • MAY 2016 – AUGUST 2016

• Use computers, and other equipment and materials to aid in presentations
• Supervise campers in class, hallways, school yard, and gym
• Enforce administration rules and policies
• Prepare lesson materials and outlines
• Organize experiment materials
• Type, file, and duplicate materials
• Maintain computers and assist students with hardware and software in the classroom
• Create Camp Sci Kids 'Beginner Programing' Camp Lesson Plan

WORK STUDY STUDENT ASSISTANT • KENNESAW STATE UNIVERSITY • AUGUST 2017 – PRESENT

• Utilize computer skills and software to create documents, schedules, and spread sheets
• Type and file various documents
• Adapt to carry out special tasks
• Relay information to students and faculty in person ad over phone
• Troubleshoot

EDUCATION

DIPLOMA • MAY 2015 • KELL HIGHSCHOOL

• Graduated with 4.0 and Honors.
• Member of Thespian Troop 7459
• Varsity Lacrosse Head Manager

MAJOR: COMPUTER SCIENCE • MINOR: CRIMINAL JUSTICE • EXPECTED GRADUATION: MAY 2019 • KENNESAW STATE UNIVERSITY

• Credits in CS 1301, CS 1302, CS 3410, CS 3501, CS 3304, CSE 3801, TCOM 2010, MATH 2202, MATH 1190, MATH 1112, MATH 2345, MATH 3322, CRJU 1101, CRJU 3332, CRJU 4305
• Current GPA – 3.51

William (Will) James Davis Diamanduros

Contact Information-

Cell (preferred)-
843-737-1054
Home-
912-489-1853

Email-
wdiamanduros@gmail.com

Address-
306 Sterling Court
Statesboro, GA
30458

Profile

I am an aspiring Software Engineering major with knowledge in Java. Received high honors during high school and awarded academic certificates.

Experience

Volunteer Experience-

- Keep Bulloch Beautiful (2005-2013)
- RIF (Reading is Fundamental) (2006-2012)
- Assisted my high school teacher with a Statesboro history project (2012)
- Assisted in the collection of 400+ clothing for Safe Haven as part of a high school project (2013)

Education

High School-

Charter Conservatory of Liberal Arts and Technology, Statesboro, GA
Graduated in 2013, with High Honors (3.76 GPA)

College-

Kennesaw State University- Marietta Campus, Marietta, GA
Currently pursuing - Software Engineering Major

Skills

- Knowledgeable in Java.

References

Corliss Reese (academic) - Director of Charter Conservatory
Phone- 912-764-5888 Email-
coreese@chartercat.org

Sheila Nielsen (academic) - Former High School Teacher from Charter Conservatory
Phone- 912-842-5060 Email-
sheilanielsen@mac.com

Benji Lewis (academic) - Student Services Director of Charter Conservatory
Phone- 912-531-6285 Email-
blewis@chartercat.org

Meghan Dowell

(706) 224-4692
MeghanLynn131@gmail.com

OBJECTIVE

To seek challenging opportunities where I can use and enhance my skills for organizational success and personal growth while continuing my education.

EDUCATION

Kennesaw State University **Marietta, GA**
B.S. in Computer Game Development Expected Graduation: December 2020
GPA: 4.0

Savannah College of Art and Design **Savannah, GA**
Interactive Design and Game Development September 2015 - October 2017
GPA: 3.88

WORK HISTORY

Chili's (Athens, GA) *August 2016 to March 2017*

- Answered the phone to take orders, reservations, and questions
- Seated customers
- Maintained appearance of dining room and restrooms

Walmart (Commerce, GA) *May 2015 to July 2015*

- Helped customers with product and pricing issues
- Scanned and bagged purchases

Arby's (Jefferson, GA) *April 2014 to January 2015*

- Worked counter and drive-through window
- Maintained appearance of the lobby and restrooms

SCHOOL INVOLVEMENT

Beta Club *Member, 2013-2015*

- Performed community service and assisted teachers

East Jackson Student Leadership Organization *Member, 2014-2015*

- Ran website, attended leadership training events, and tutored students

COMMUNITY INVOLVEMENT

Franklin Life Pregnancy Resource Center *2018*

- Assisted in the *Golf for Life* event

Southside Church *2015*

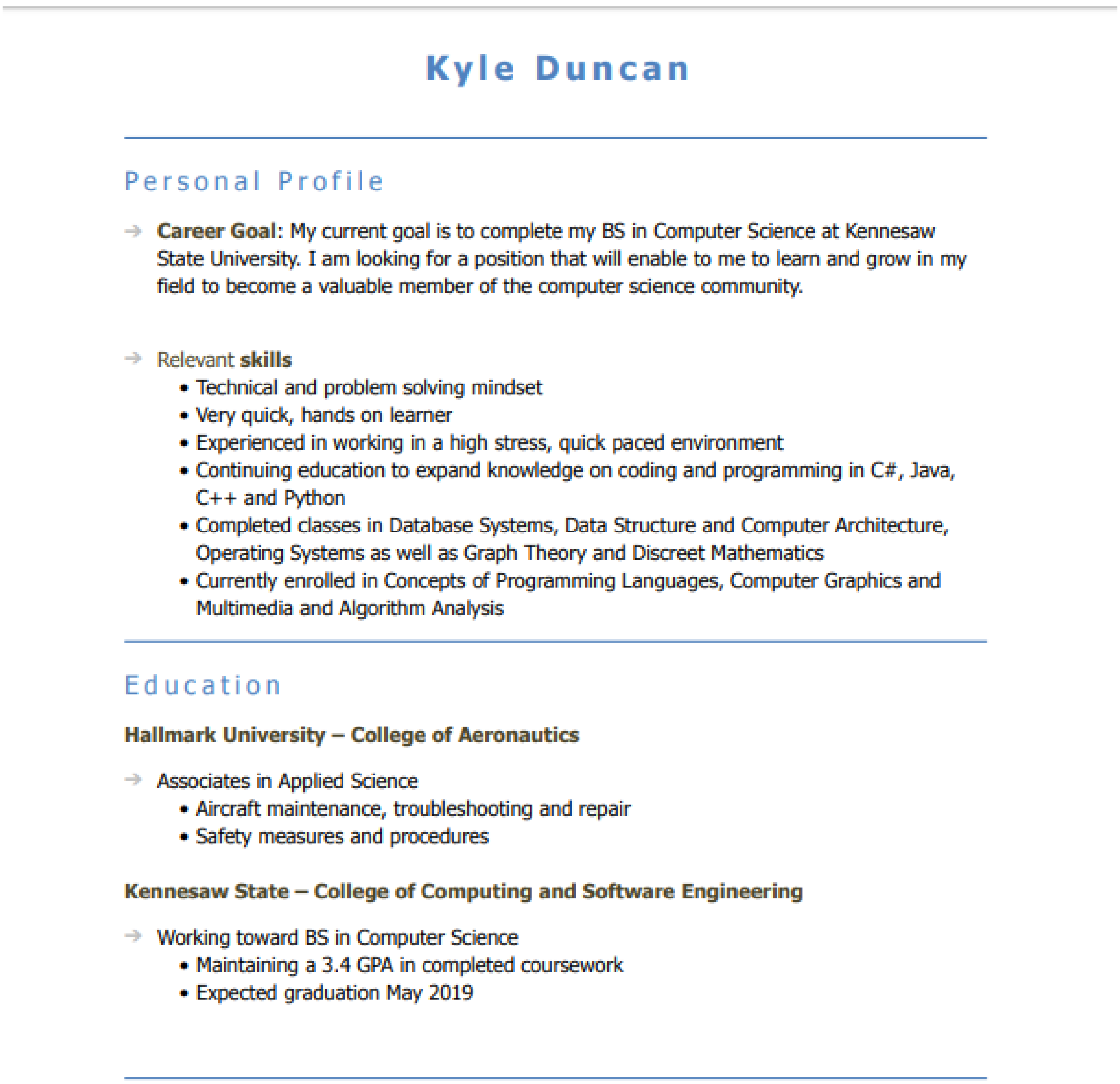
- Removed furniture and cleaned rooms when the church moved buildings

Food Bank of Northeast Georgia *2015*

- Cleaned the building and organized food
- Set up and removed furniture and sold shirts during a fundraising event

SKILLS

- Goal-oriented and adaptable
- Exceptional time management skills
- Experience with C#
- Experience using Microsoft Word, Publisher, PowerPoint, Photoshop, and Maya
- Some experience with C++, MEL, Bash, and Python



Kyle Duncan

Personal Profile

→ **Career Goal**: My current goal is to complete my BS in Computer Science at Kennesaw State University. I am looking for a position that will enable to me to learn and grow in my field to become a valuable member of the computer science community.

→ Relevant **skills**

- Technical and problem solving mindset
- Very quick, hands on learner
- Experienced in working in a high stress, quick paced environment
- Continuing education to expand knowledge on coding and programming in C#, Java, C++ and Python
- Completed classes in Database Systems, Data Structure and Computer Architecture, Operating Systems as well as Graph Theory and Discreet Mathematics
- Currently enrolled in Concepts of Programming Languages, Computer Graphics and Multimedia and Algorithm Analysis

Education

Hallmark University – College of Aeronautics

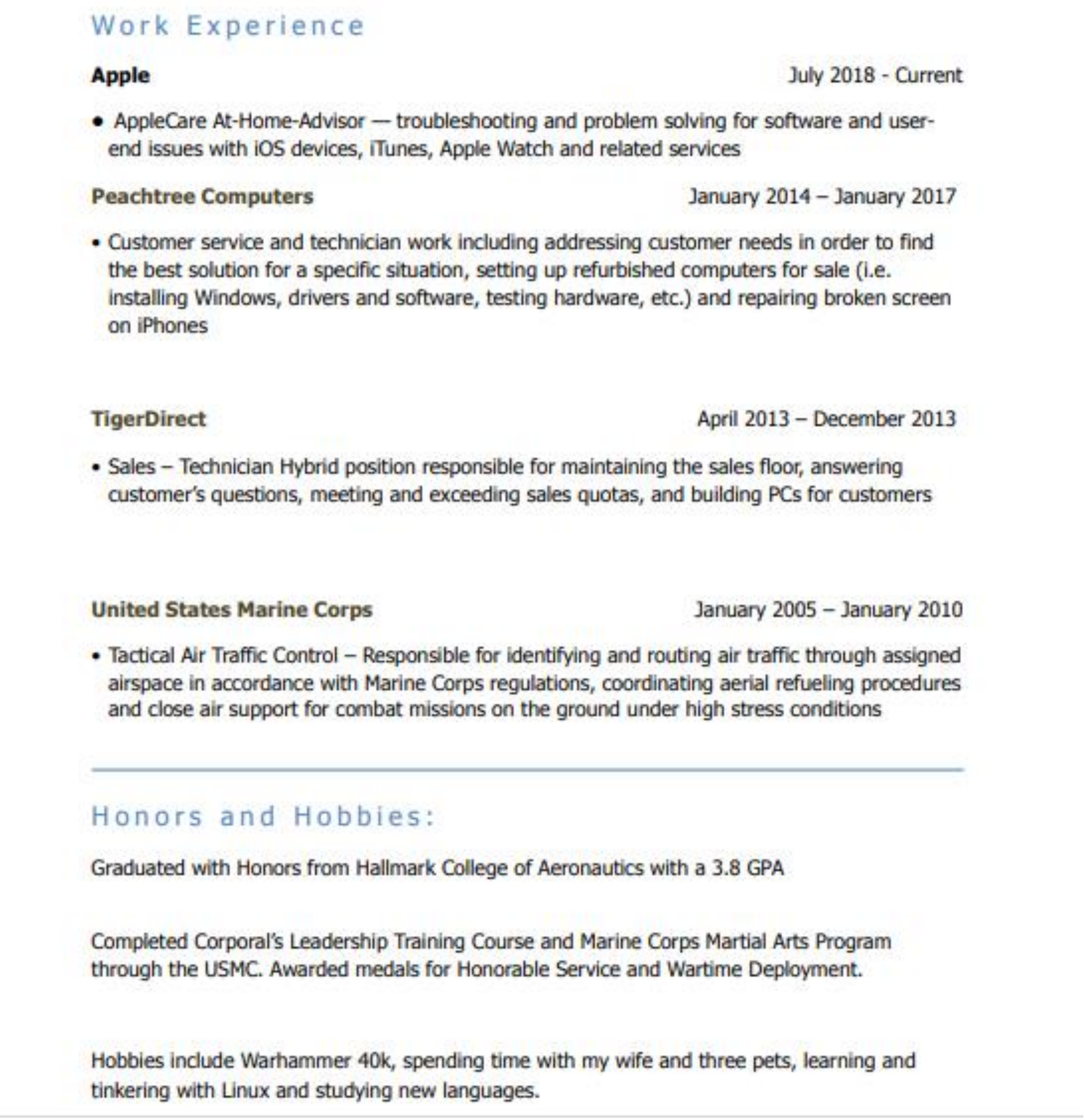
→ Associates in Applied Science

- Aircraft maintenance, troubleshooting and repair
- Safety measures and procedures

Kennesaw State – College of Computing and Software Engineering

→ Working toward BS in Computer Science

- Maintaining a 3.4 GPA in completed coursework
- Expected graduation May 2019



Work Experience

Apple July 2018 - Current

- AppleCare At-Home-Advisor — troubleshooting and problem solving for software and user-end issues with iOS devices, iTunes, Apple Watch and related services

Peachtree Computers January 2014 – January 2017

- Customer service and technician work including addressing customer needs in order to find the best solution for a specific situation, setting up refurbished computers for sale (i.e. installing Windows, drivers and software, testing hardware, etc.) and repairing broken screen on iPhones

TigerDirect April 2013 – December 2013

- Sales – Technician Hybrid position responsible for maintaining the sales floor, answering customer's questions, meeting and exceeding sales quotas, and building PCs for customers

United States Marine Corps January 2005 – January 2010

- Tactical Air Traffic Control – Responsible for identifying and routing air traffic through assigned airspace in accordance with Marine Corps regulations, coordinating aerial refueling procedures and close air support for combat missions on the ground under high stress conditions

Honors and Hobbies:

Graduated with Honors from Hallmark College of Aeronautics with a 3.8 GPA

Completed Corporal's Leadership Training Course and Marine Corps Martial Arts Program through the USMC. Awarded medals for Honorable Service and Wartime Deployment.

Hobbies include Warhammer 40k, spending time with my wife and three pets, learning and tinkering with Linux and studying new languages.

RD

ROBERT LANE DYER

OBJECTIVE

I am in college for game design. I plan on being a game developer in the future.

SKILLS

I've done programming for 3 years. I've also done graphic design for a year.

EXPERIENCE

SALES ASSOCIATE • BEST BUY • 09/2018 – CURRENT DATE
I contact customers and would help them find the product they needed.

INVENTORY SPECIALISTS • BEST BUY • 06/2017 – 09/2018
I kept track of inventory for the store, completed online orders and completed shipping and receiving.

EDUCATION

COMPUTER GAME DESIGN AND DEVELOPMENT • N/A • KENNESAW STATE UNIVERSITY

VOLUNTEER EXPERIENCE OR LEADERSHIP

I was a stream team member and community manager for a media company called EasternMediaGG.

2.4 Technical Systems Description

The team has been tasked with creating an ordering system that allows customers to order customized pizza for take-out, record customer information, order receipt, and payment options.

Description of the Pizza Ordering System

The system will consist of four parts in order for the system to work properly. The GUI for the system will have a section to record customer information, to create customized pizza and to select payment options. The last part for the system to work is the database that will be running underneath the GUI.

Recording Customer Information

For the system to start recording the customer information, the customer must provide a phone number. The customer will be able to put their name and address after they have provided a phone number. The customer can provide extra locations to help locate their address.

Creating Customized Pizza

Once the customer information has been provided, the customer can begin creating their pizza. This section will provide different sizes of pizza, crust style and toppings. The customer can also select a beverage from here as well.

Selecting Payment Options

This section the customer can select what card they would like to use to pay for their pizza. The customer can also provide a tip here as well. From this section the order can be confirmed and the receipt will be made.

The Database

The database will hold all recorded customer information, order history and receipts. This will ensure faster ordering for customers who are in the system. This will also allow the restaurant to keep track of revenue.

2.5 Data Management Plan

Project Name:	SWE 3313 Pizza Ordering System
Team Members:	Bayleigh Correll, Will Diamanduros, Meghan Dowell, Kyle Duncan, Lane Dyer
Team Leader:	Meghan Dowell
Due Date:	November 16, 2018
Date Last Modified:	October 21, 2018

Types of Data	• Basic Customer Information (Phone number, name and address) • Charge account (Visa/ Mastercard) • GPS information to locate address • Type of payment and amount • Delivery or pickup information • Types of toppings • Crust options • Size of pizza • Beverages • Price of Pizza • Price of Beverages
How the Data will be obtained and stored	• New customer information will be obtained via customer phone number then will be placed into the master record. • Customer information will be placed into a database to allow ease of access for returning customers. • A database for all various pizza options and beverages.
Policies	• Customer information will only be used for ordering purposes. • Customer information will not be shared publicly. • Customer information can only be accessed via their phone number. • The menu will be publicly available to everyone

2.6 Test Plan

October 13, 2018

Goal and Exit Criteria

- Goal
    - Members of the group must properly test the Pizza Order and Delivery System. All requirements of the system must be identified as bug free and in working order. System must also be tested for performance and run at a fast and reasonable speed.
- Exit Criteria
    - Goals and testing scenarios must be met and identified as bug free.
    - Product must meet the type of quality standards set by our client.

Items to Test

- Test new customer login process.
- Test system's payment acceptance process.
- Test viewing transactions process.
- Test the printing of receipts process.
- Test the system's "order taking" process.

Test Processes

- Assigning group members to test each part of the requirements.
- Identifying bugs and fixing them.

Resources

- 5 people consisting of 2 experienced in Computer Science, 1 experienced in Software Engineering and 2 experienced in Game Development.
- Developing and testing using Unity with C#

Schedule

- Begin testing on 11/10/18
- End testing on 11/15/18

Test Cases

- Test Scenario 1 – Setting up a new record/profile for a new customer
    - Check results when entering all required information correctly.
    - Check results when entering one or more information requirements incorrectly.
    - Check results when leaving one or more information requirements blank.
- Test Scenario 2 – Processing forms of payments.
    - Check results when processing all forms of payments.
    - Check results when a form of payment is missing.
    - Check results on if system is keeping track of payments.
- Test Scenario 3 - Access transactions database
    - Check results when user has access to the transactions database.
    - Check results when user is unable to access database.
    - Check results when user has access to the transactions database and that address and delivery information is shown correctly.
- Test Scenario 4 - Printing receipt
    - Check results when system prints a receipt.
    - Check results when system prints a receipt and that it has a place to sign and also correctly shows forms of payment, all user information, list of items ordered, if it was delivery or pickup and amount due.
    - Check results when system is unable to print a receipt.

- Test Scenario 5 – Viewing menu
    - Check results to see if all information on the menu is showing correctly.

- Check results for when a piece of information on the menu is not showing correctly.
- Check results when a user is viewing and picking pizza size, topping, crust and beverage options.
- Check results when an option on the menu is not chosen.

3. REQUIREMENTS DOCUMENTS

3.1 Requirements Definition Document

Project:

Pizza order and delivery system for a Pizza Shop

Aim:

The purpose of this document is to document and track the requirements of the pizza order and delivery system in order to successfully define the functional and nonfunctional requirements of the system. The system must contain the restaurant menu and must allow the user to input a new customer, take an order, calculate a bill, and process payments.

Functional Requirements:

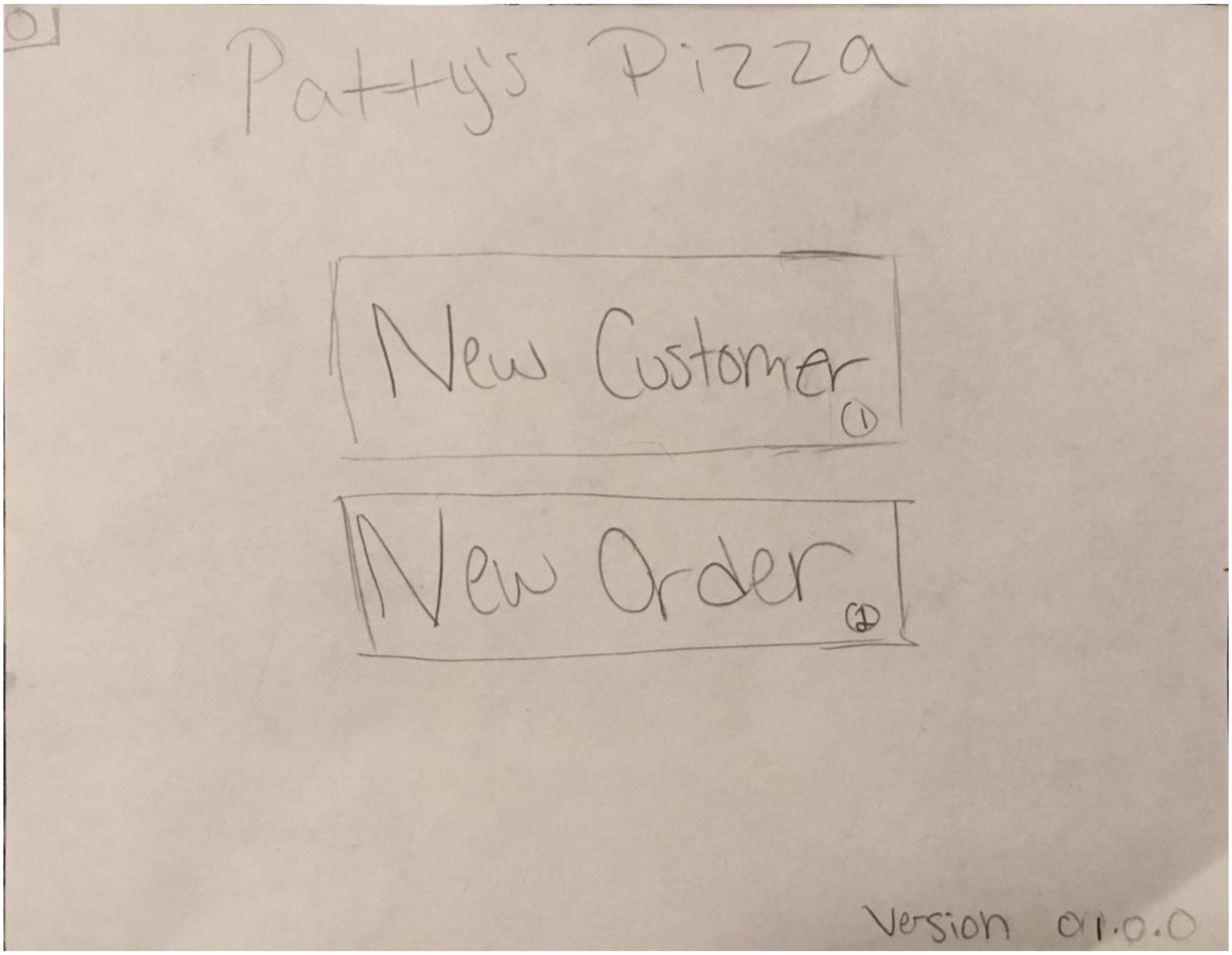
1. User must be able to create a record for a new customer
2. Customer information must include
    a. Customer telephone number
    b. Customer name
    c. Customer address
    d. Type of charge account
    e. Information on how to locate customer address, such as subdivision name or closest major intersection
3. User must be able to find a customer by telephone number
4. Customer address and delivery information must be displayed for all customer transactions
5. System must contain the complete restaurant menu
6. User must be able to have access to the menu items through a graphical user interface
7. Menu must contain
    a. Pizza sizes
    b. Pizza toppings
    c. Pizza crust options
    d. Beverage options

8. System must calculate and display the order total
9. System must process payments
10. Payments must be processed in the form of
    a. Checks
    b. Cash
    c. Credit cards
11. System must retain information on type of payment and amount charged
12. System must print a receipt when a transaction is complete
13. The receipt must contain
    a. a place for customer signature if payment type used is credit card
    b. all customer information
    c. a list of items ordered
    d. information on if the order is for delivery or pickup
    e. total amount due

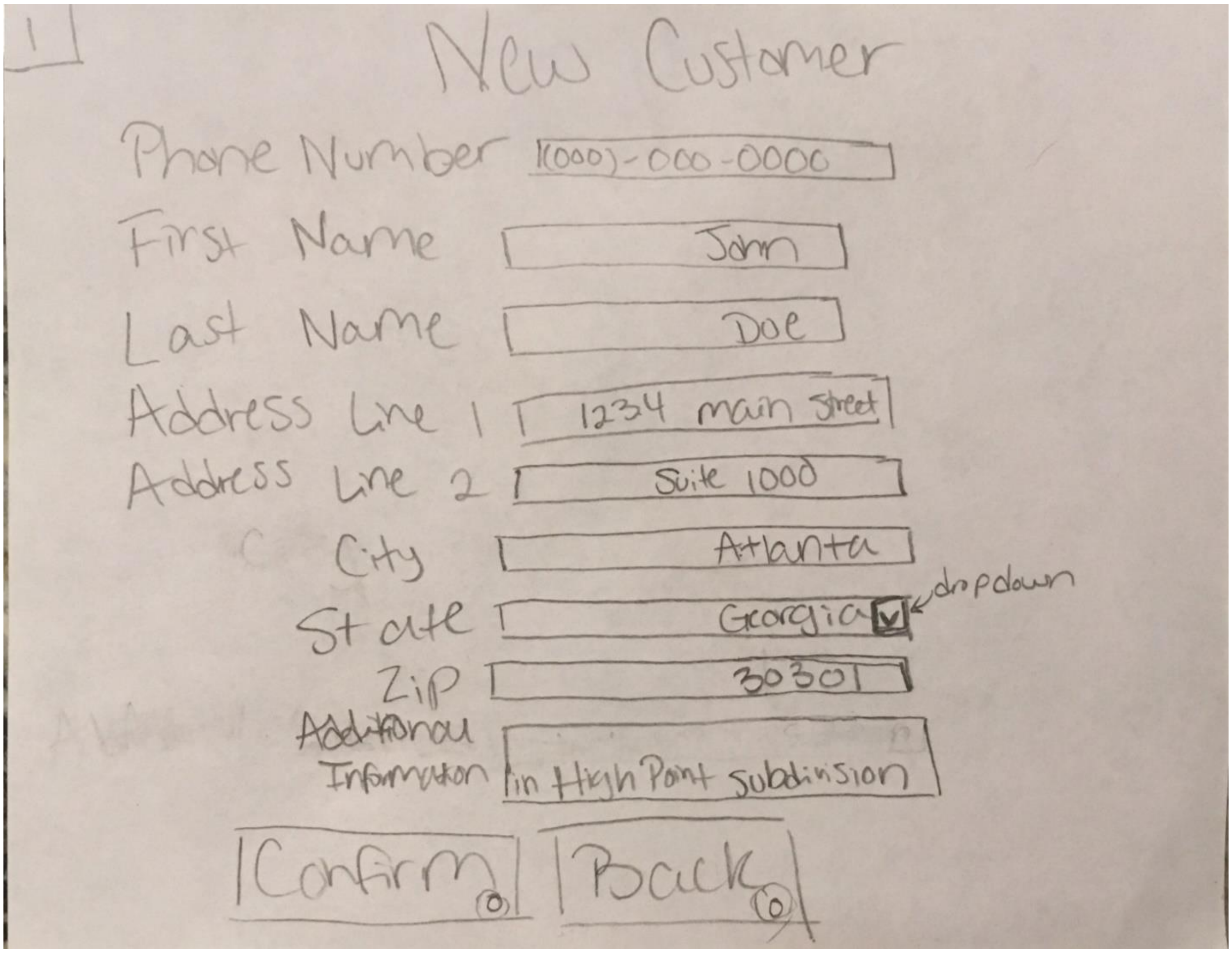
Nonfunctional Requirements:

1. Acceptable system response time is less than 5 seconds
2. The system shall display appropriate messages to the user upon user interaction
3. The interface must be user-friendly for quick and easy order input
4. Program instructions should be clear and easy to understand
5. System must be maintainable, able to be updated efficiently in future versions

3.1.1 Paper Prototype



Screen 0 – main screen – if “New Customer” is pressed, go to screen 1. If “New Order” is pressed, go to screen 2



Phone Number (000)-000-0000

First Name John

Last Name Doe

Address Line 1 1234 main street

Address Line 2 Suite 1000

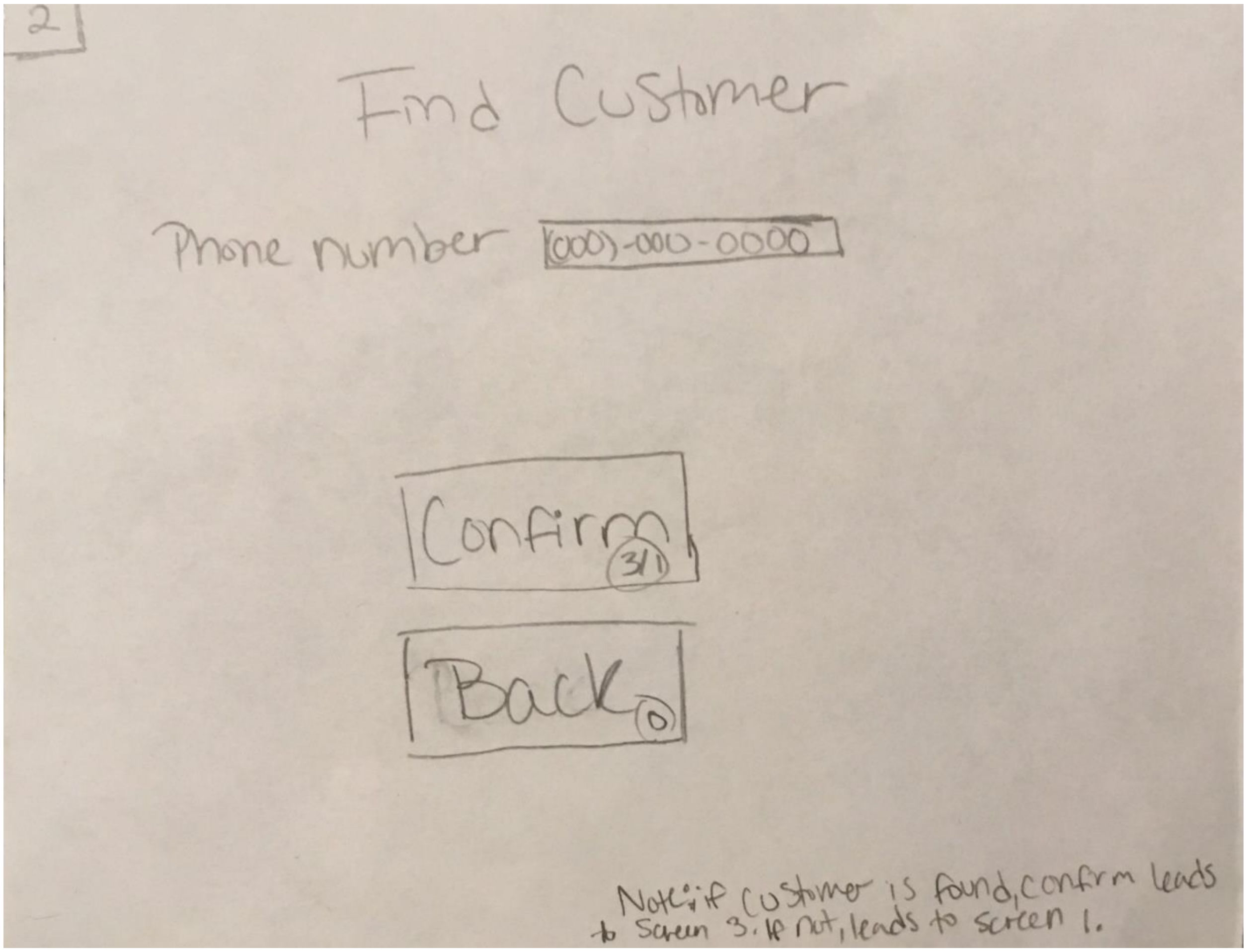
City Atlanta

State Georgia

Zip 30301

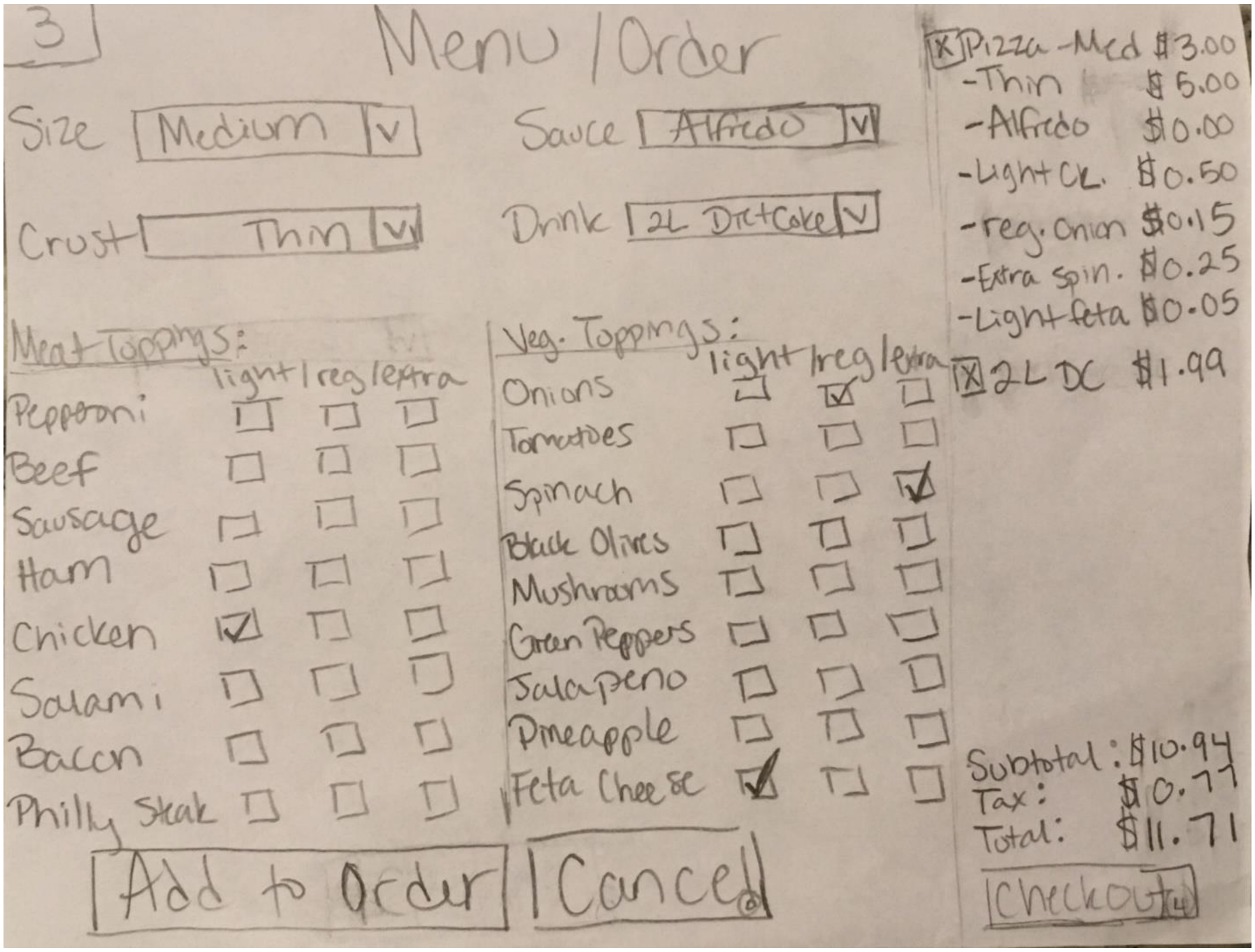
Additionau Information in High Point subdivision

Screen 1- new customer screen – if “Confirm” is pressed, customer data is saved, go to screen 0. If “Back” is pressed, customer data is not saved, go to screen 0



Note: if customer is found, confirm leads to Screen 3. If not, leads to screen 1.

Screen 2 – find customer screen – If “Confirm” is pressed, program finds the customer by inputted phone number. If customer is found, go to screen 3. If customer is not found, go to screen 1. If “Back” is pressed, go to screen 0



Size Medium
Sauce Alfredo
Crust Thin
Drink 2L DietCoke

Meat Toppings:

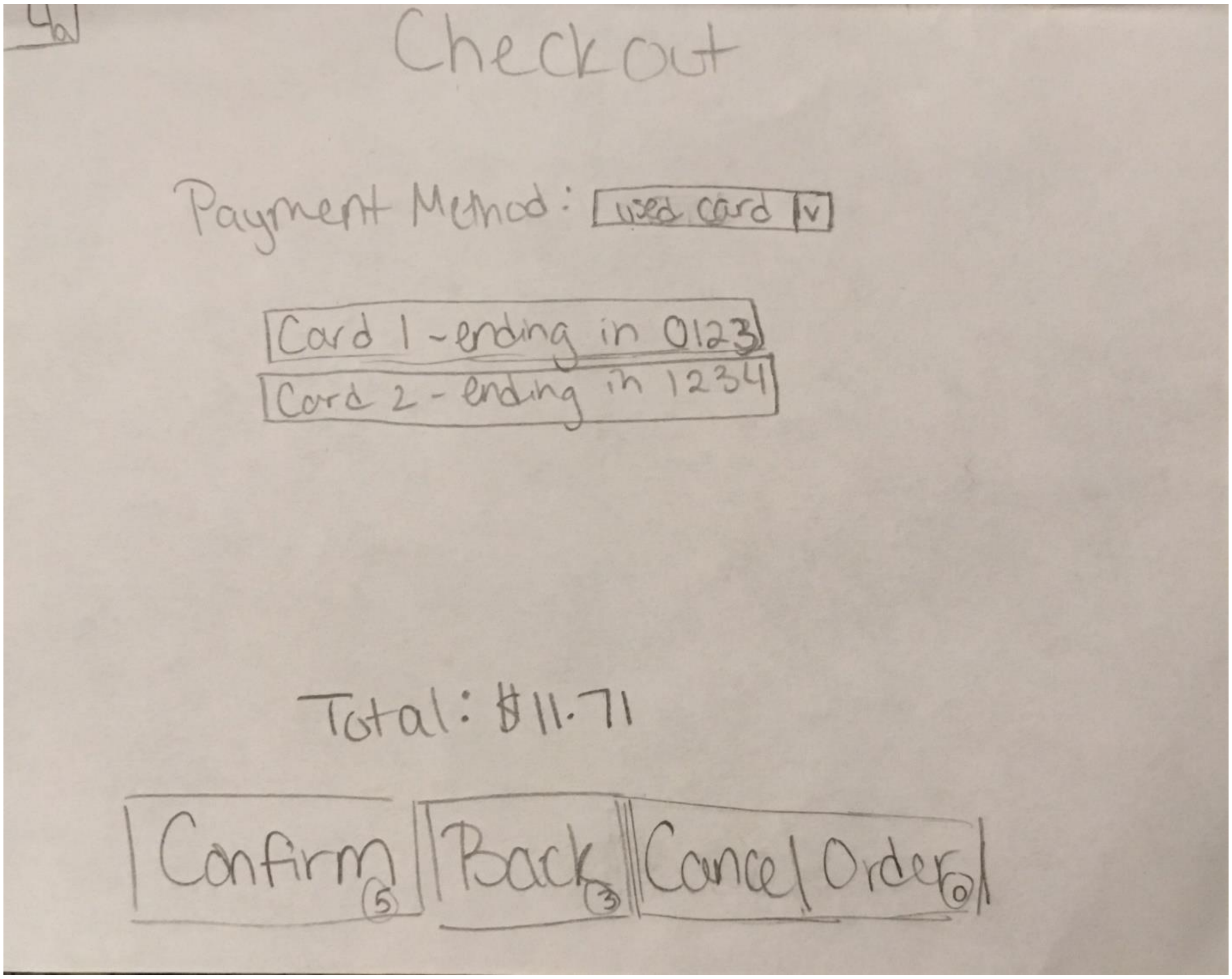
	light	reg	extra
Pepperoni	□	□	□
Beef	□	□	□
Sausage	□	□	□
Ham	□	□	□
Chicken	☑	□	□
Salami	□	□	□
Bacon	□	□	□
Philly Steak	□	□	□

Veg. Toppings:

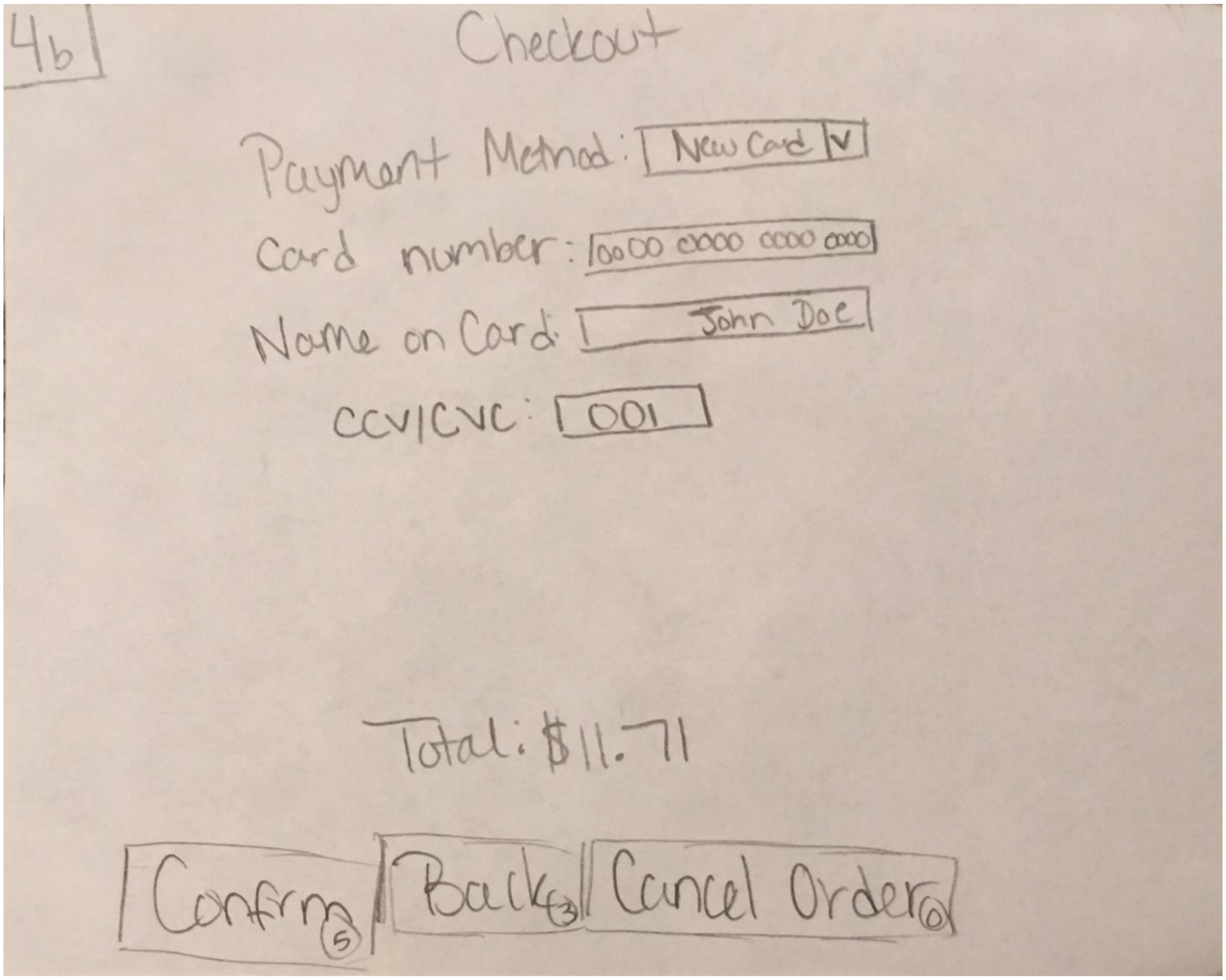
	light	reg	extra
Onions	□	☑	□
Tomatoes	□	□	□
Spinach	□	□	☑
Black Olives	□	□	□
Mushrooms	□	□	□
Green Peppers	□	□	□
Jalapeno	□	□	□
Pineapple	□	□	□
Feta Cheese	☑	□	□

☒ Pizza - Med	$3.00
- Thin	$5.00
- Alfredo	$0.00
- Light Ck.	$0.50
- reg. Onion	$0.15
- Extra spin.	$0.25
- Light feta	$0.05
☒ 2L DC	$1.99
Subtotal:	$10.94
Tax:	$0.77
Total:	$11.71

Screen 3 – order screen – drop-down menus for pizza size, sauce, crust, and beverage. Check boxes for toppings. If “Add to Order” is pressed, currently selected options are added to order on right side of screen with prices and added to total cost. Then all selections are reset to unchecked boxes and “make selection” for drop-down menus If “Cancel” is pressed, go to screen 0. If “Checkout” is pressed, order is completed, go to screen 4.



Screen 4a – payment screen – shows if "Used Card" is selected from the drop-down menu. User selects "Card 1" or "Card 2" and presses "Confirm" to complete the transaction, go to screen 5 (receipt). If "Back" is pressed, go to screen 3. If "Cancel Order" is pressed, order is removed, go to screen 0.



4b

Checkout

Payment Method: New Card

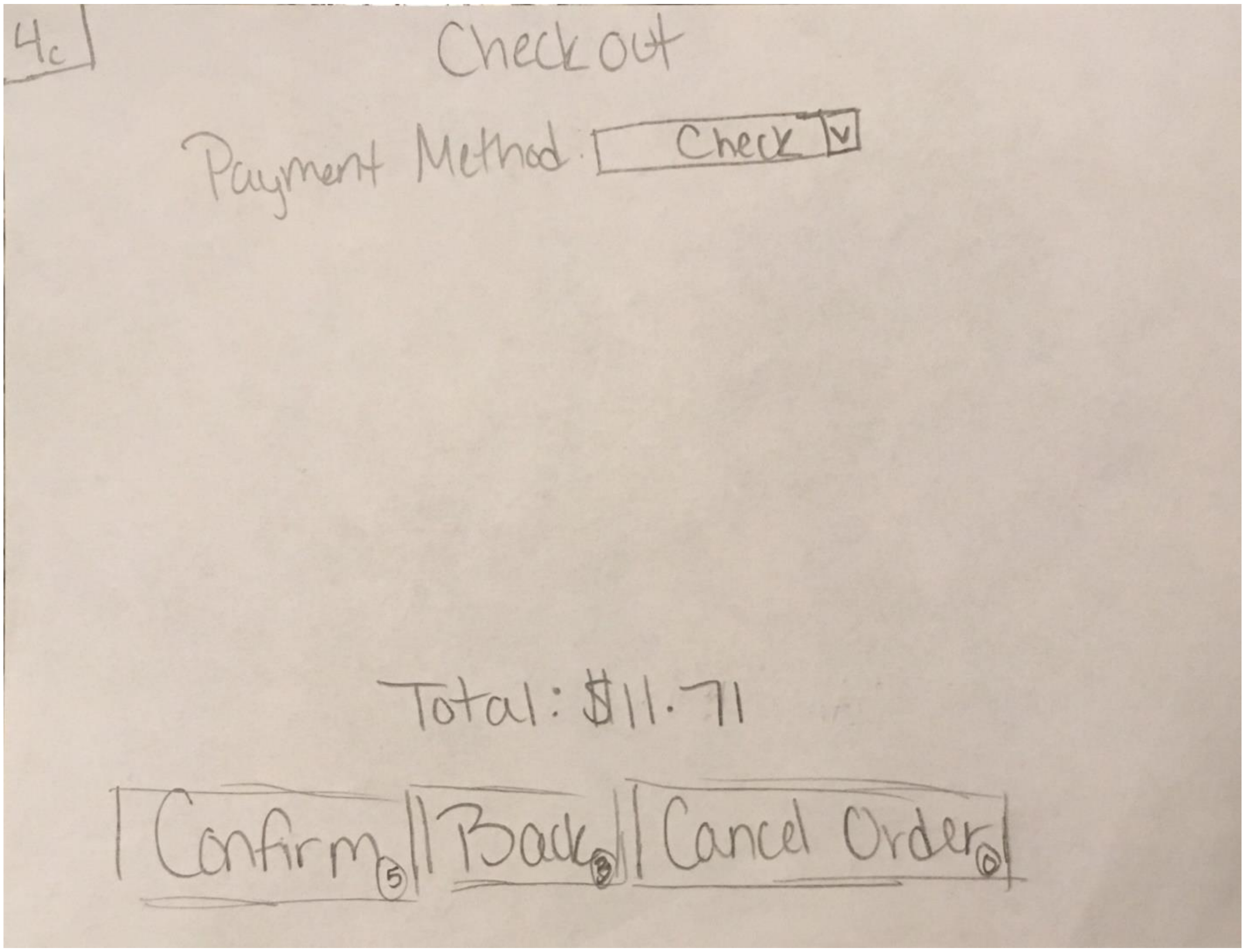
Card number: 0000 0000 0000 0000

Name on Card: John Doe

CCV/CVC: 001

Total: $11.71

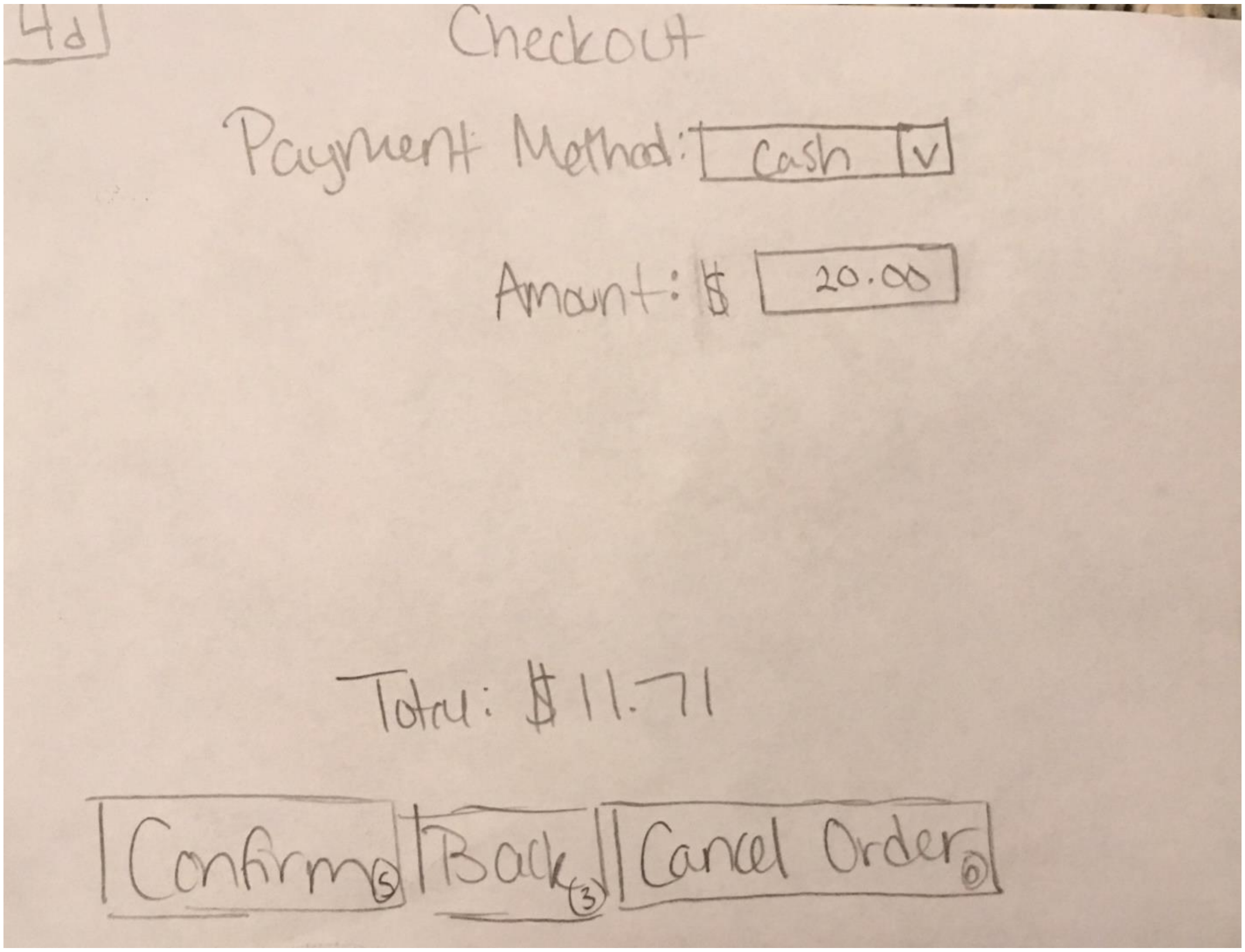
Screen 4b – payment screen – shows if “New Card” is selected from the drop-down menu. User inputs card information and presses “Confirm” to complete the transaction, card information is saved for the customer, go to screen 5 (receipt). If “Back” is pressed, go to screen 3. If “Cancel Order” is selected, go to screen 0.



Payment Method: Check

Total: $11.71

Screen 4c – payment screen – shows if "Check" is selected from the drop-down menu. User presses "Confirm" and puts check in drawer to complete the transaction, go to screen 5 (receipt). If "Back" is pressed, go to screen 3. If "Cancel Order" is pressed, go to screen 0.



4d

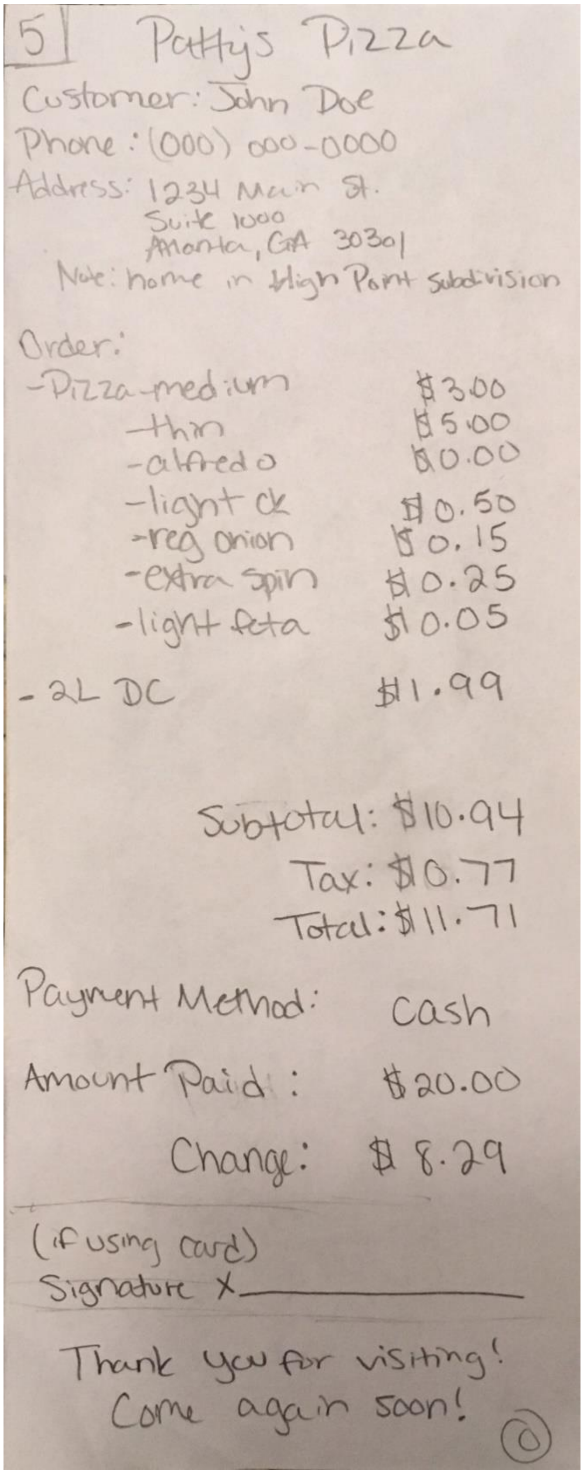
Checkout

Payment Method: Cash v

Amount: $ 20.00

Total: $11.71

Screen 4d – payment screen – shows if “Cash” is selected from the drop-down menu. User inputs cash amount paid and presses “Confirm” to complete the transaction, receipt will show change due, go to screen 5 (receipt). If “Back” is pressed, go to screen 3. If “Cancel Order” is pressed, go to screen 0.



5

Pattys Pizza

Customer: John Doe

Phone: (000) 000-0000

Address: 1234 Main St.
Suite 1000
Atlanta, GA 30301

Note: home in High Point subdivision

Order:

-Pizza-medium	$3.00
-thin	$5.00
-alfredo	$0.00
-light ck	$0.50
-reg onion	$0.15
-extra spin	$0.25
-light feta	$0.05
- 2L DC	$1.99

Subtotal: $10.94

Tax: $0.77

Total: $11.71

Payment Method: cash

Amount Paid: $20.00

Change: $8.29

(if using card)
Signature X________________

Thank you for visiting!
Come again soon!

0

Screen 5 – receipt – shows customer information, order information and prices, total cost, payment method, amount paid, change due, and signature spot if customer used card. System then defaults back to screen 0.

3.2 Software Requirements Specifications

October 23, 2018

System Overview

- This system is a pizza order and delivery system for a new pizza shop. The system is intended to allow the user to view the pizza shop's menu and pick size, toppings and crust flavor of a pizza. The system should also allow the user to create a new record for new customers if needed as well as print a receipt with all necessary information, accept all forms of payments and allow the user to access a customer transaction database.

Technical Requirement

- Functional Requirements
    - Allows user to create a new record for new customers requiring the new customer's phone number, name, address and form of payment.
    - System must be able to accept all forms of payments (cash, check, and credit card).
    - Allows user to view a customer transaction database showing the customer's address and delivery information.
    - Allows user to print a receipt with a place to sign and showing all customer's information, list of items ordered, delivery preference and amount due.
    - Allows user to view a menu where the customer picks the pizza's size, toppings and crust flavor. The customer can also pick a beverage of their choosing.
- Nonfunctional Requirements
    - Response time must be less than 5 seconds.
    - Instructions are clear, easy to understand and user friendly.
    - System displays everything properly without errors.
    - System is able to be patched and updated if need be.
- User-interface Specifications
    - Must be clear, simple and easy to understand.
    - Must be easy to use with minimal instructions required.

- Must be simple enough to allow user to complete order processes as quickly as possible.

Interaction Scenarios

1. When customer is ready to place an order, user must pick “new customer” or “new order.”
    1. If new customer is picked, user must enter customers information including phone number, name, address and form of payment.
    2. If new order is picked then user must enter customer’s phone number to proceed.
2. When the customer’s phone number is entered, user is now able to view a menu where the customer must now pick their pizza’s size, toppings and crust flavor as well as a beverage.
3. Once order is complete, user must now click checkout.
4. Once clicked customers must now pick their preferred payment and delivery options.
5. Once chosen the user now has an option to print a receipt that has a place to sign, all customer’s information, list of items ordered, delivery preference and amount due.
6. If the customer wants to view transactions the user has an option to do so.

Validation

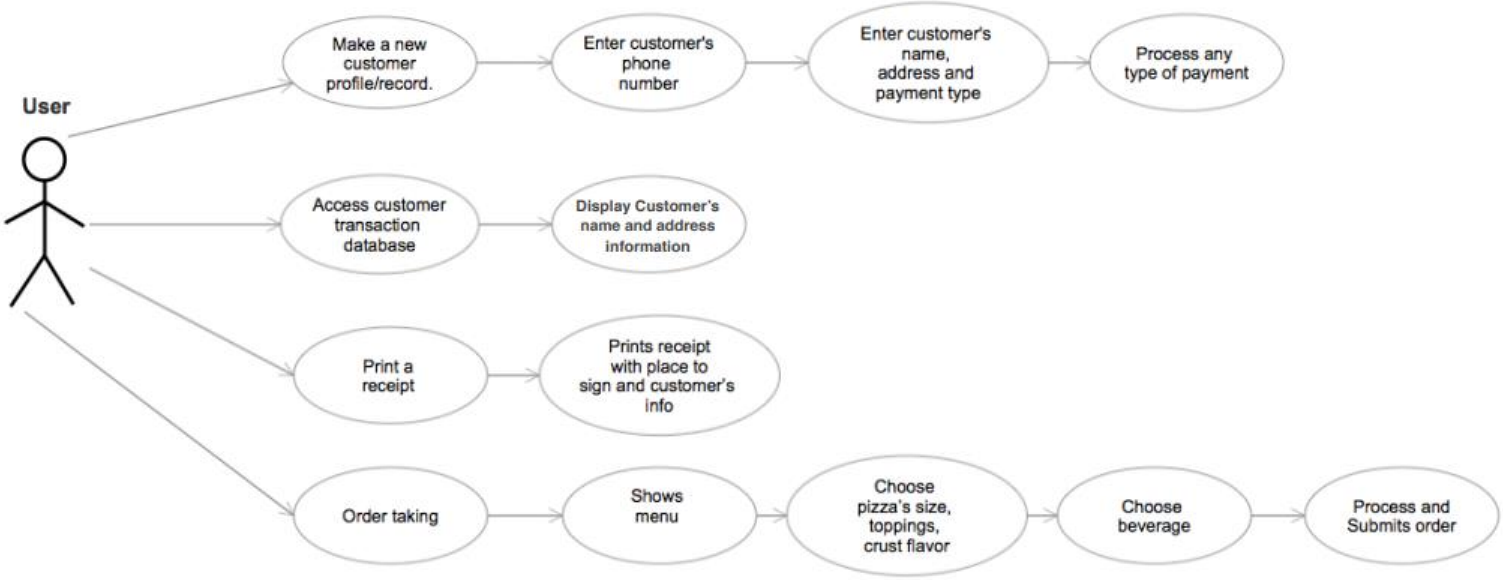
- Based on the interaction scenarios, software has been validated to follow the customer’s requirement as written below.
    1. You will set up a record for each new customer. This information will be keyed by customer phone number. Put information into the master record such as name, address, phone and type of charge account (Visa/ MasterCard). You also need to record pertinent information for locating the address (i.e., subdivision name, closest major intersection, etc.)
    2. Your system needs to process payments in the form of checks, cash or credit cards. You will need to keep track of the type of payment made and the amount.
    3. Users will access the customer database for all customer transactions. The information should come up for already established customers showing their address and delivery information.
    4. Your system should be able to print out a receipt of the order with a place to sign the form by the customer if it is on a credit card. This should have customer information, a list of items ordered, if it is for delivery or pickup, and amount due.
    5. Your system should contain the complete restaurant menu for order-taking purposes. The user needs to have GUI access to the various menu items and be able to quickly enter the desired orders. This menu needs to have various sizes of pizzas, the typical

types of toppings, various crust options, and beverages. The customer should be able to order things like a medium, thin crust pizza, with pepperoni and extra cheese.

Requirement Consideration

- This system is made for an employee to use when a customer is ready to order a pizza.
- The system is made for an environment that is fast paced and under constant use.
- Limitations include only being able to preform the customer's requirements nothing more.
-

3.2.1 Use Case Diagram


User
Make a new customer profile/record.
Enter customer's phone number
Enter customer's name, address and payment type
Process any type of payment
Access customer transaction database
Display Customer's name and address information
Print a receipt
Prints receipt with place to sign and customer's info
Order taking
Shows menu
Choose pizza's size, toppings, crust flavor
Choose beverage
Process and Submits order

3.2.2 Use Case Flow of Events

Prepared for: Pizza Order and Delivery System

October 17, 2018

Main Flow

This system begins when an existing customer is ready to start the order taking process for the pizza. It starts by allowing the user to view the menu. From there, the customer is able to pick the pizza's size, the topping and the crust flavor. The customer is also able to pick a beverage as well. There will also be an option to choose delivery or pickup. Once finished, the user must hit submit and the order will be processed.

Sub Flows

- ### S-1: Creating a New User

  - The system displays an option to create a new customer profile. Once the user starts the process they must first enter the customer's phone number as their key. Once completed they must ask for the customer's name, address and type of payment. All forms of payment are accepted and will be processed and tracked.

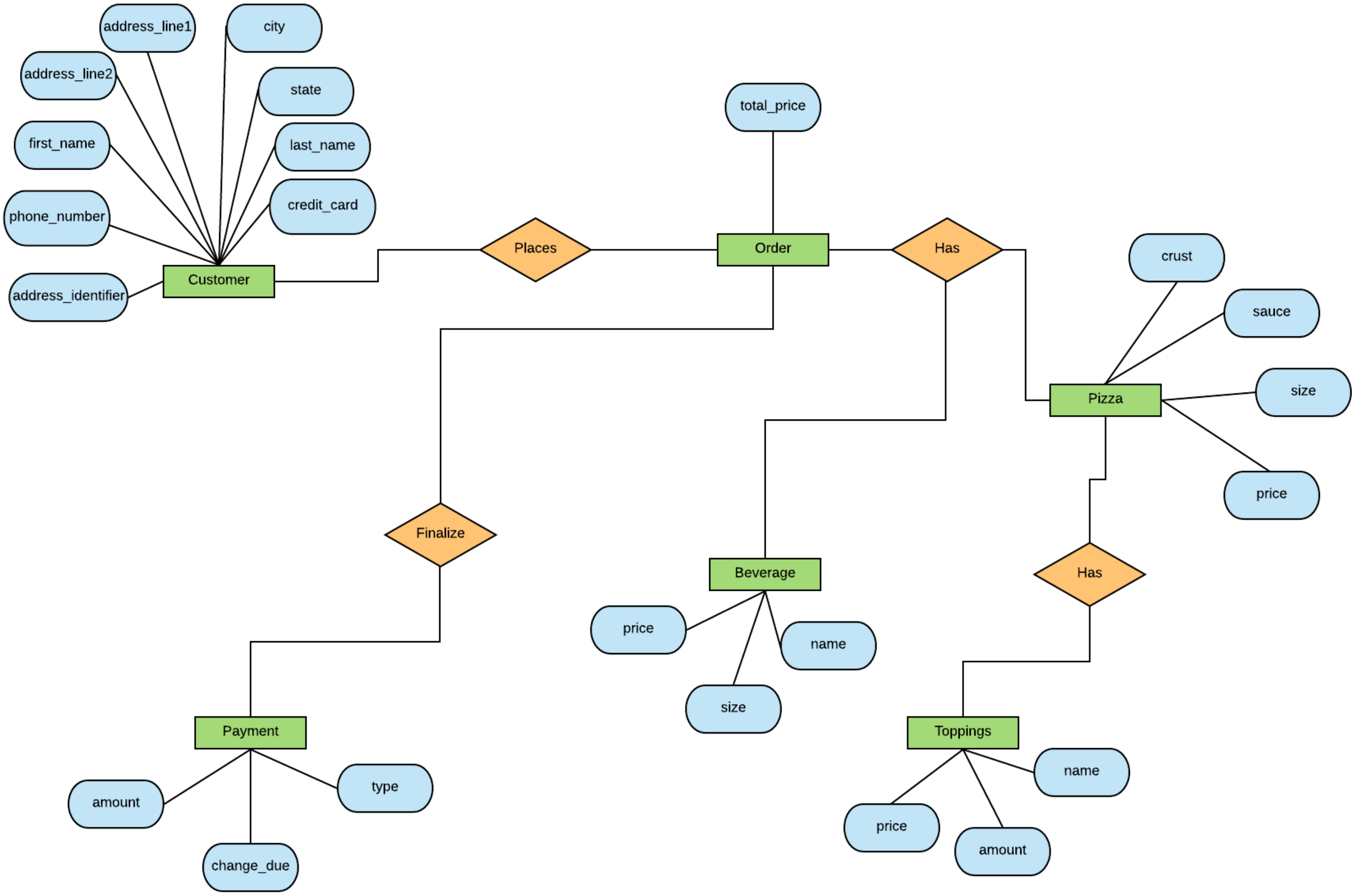
- ### S-2: Accessing Customer Transaction Database

  - The system displays a way for the user to access the customer transaction database. The database will display all pervious customer transactions as well as showing each address and delivery information that was enter with each transaction.

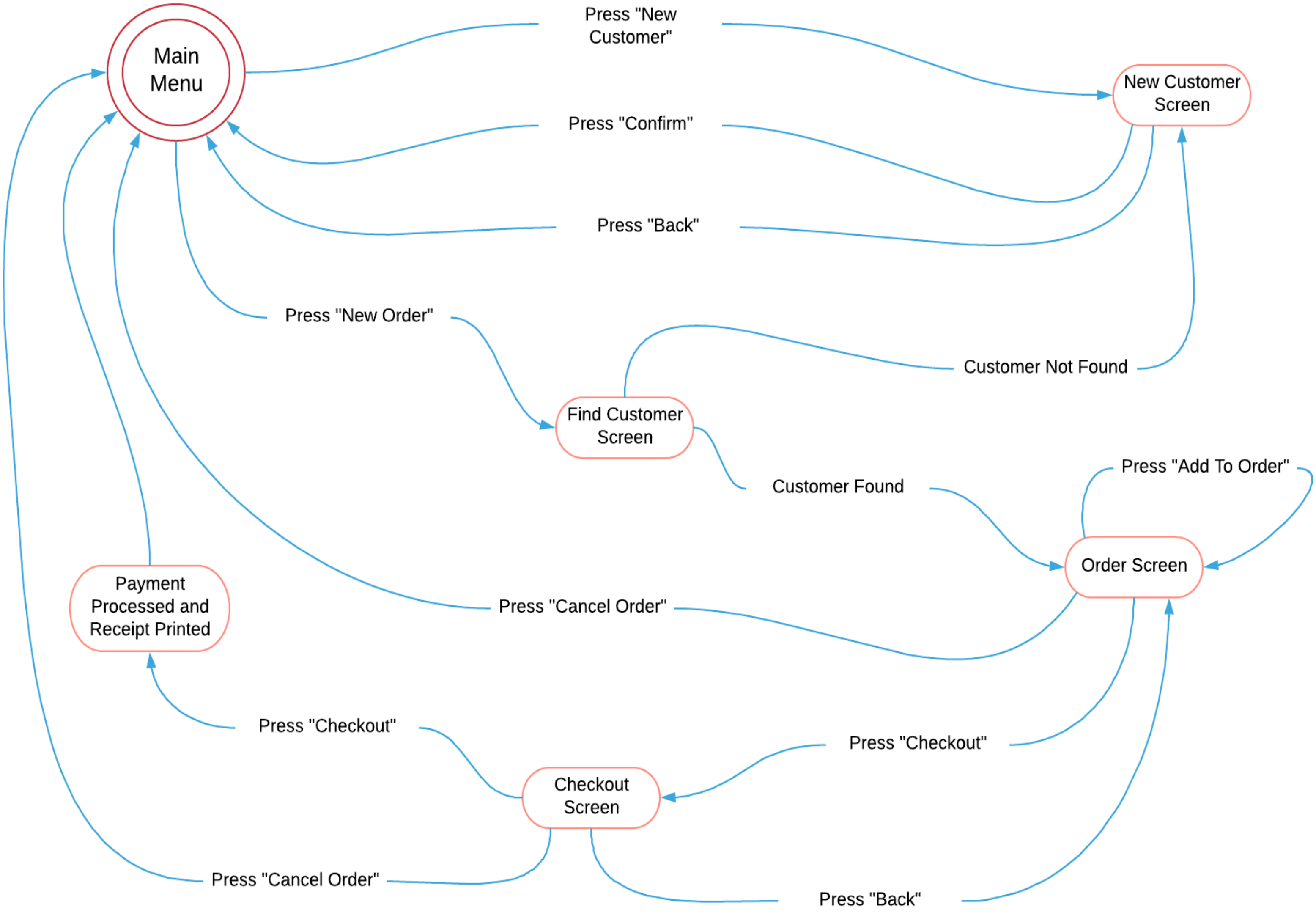
- ### S-3: Printing Receipt

  - If the user needs to print a receipt the system will display an option to do so. The receipt will have a place to sign as well as the customer's information, a list of items ordered, if it was delivery or pickup, and amount due.

3.2.3 Entity Relationship Diagram



3.2.4 State Transition Diagram



4. SYSTEM DESIGN DOCUMENTS

4.1 Conceptual System Design

November 3, 2018

The Problem

- The pizza order and delivery system needs to be started from scratch, needs to be finished in four months, and that its five requirements must be met for the finished product. The first requirement is that it must be able to create a new customer recorded for the system. The system must take the name, address, phone number and payment method of the new customer to create a new record. Secondly the system needs to be able to accept all types of payment methods. The third requirement is that the system must have access to a customer database for all the customer's transactions and show their address and delivery information. The fourth one is that the system needs to be able to print a receipt with a place to sign, and has the customer's information, a list of items ordered, if it is for delivery or pickup, and the amount due. Finally the last requirement is that the system must have a menu where the customer can pick the size of the pizza, types of toppings, crust options, and beverages.

System Objectives

- First objective is to finish all required paperwork that is first needed to begin working on the system by the beginning of the month it is started on.

- Second objective is that we must create a prototype for the system towards the end of the month stated in objective one.

- Third objective is to begin working on the system at the beginning of the second month and finish the system by the beginning/midpoint of the third month.

- Fourth objective is that we must test the system and fix any errors and bugs at the midpoint of the third month.

- Fifth objective is to have everything done by the midpoint of the fourth and final month.

Constraints

- External constraints are that we must follow the requirements set by the customer.

- Internal constraints are that the members making the system have different backgrounds when it comes to programming languages, and that some members are knowledgeable in programs used to create the system, while some are not.

Internal Records

- **Project Plan Documents-**
  - Scope
  - Schedule (including work breakdown structure, milestones and/or deliverables, and a Gantt chart)
  - Team organization (including resumes of the team members)
  - Technical description of the system
  - Data management plan
  - Test plan
- **Other required Documents-**
  - Requirements definition document
  - Requirements specification document
  - Paper prototype (drawn by hand or via a drawing tool like Visio or Word)
  - Use case diagram (created by a drawing tool or Visio)
  - Use case flow of events document (text document, created in Word)
  - Class diagrams (created in Visio)
  - Class documentation (created in Word)
  - Entity relationship diagram(s) (created by a drawing tool or DB tool)
  - At least one of each of the following (for documenting classes): EITHER decision table(s) or state transition diagram(s)
- **System Design Documents-**
  - Conceptual system design
  - Report formats (created in Word)
  - Screen layouts/ shots (screen captures from your prototype application embedded in a text document guiding the user through the application, with the requirements and use cases being addressed by each screen clearly identified on the bottom of the page – underneath the screenshot)
  - Technical design: (high level)

    - Detailed class diagrams with all methods, attributes and relationships identified
    - Supporting text specification (created in Word)
    - Database table descriptions

Operating Personnel

- Operating Leader-
    - Meghan Dowell
- Operating Personnel-
    - Bayleigh Correll
    - Will Diamanduros
    - Kyle Duncan
    - Lane Dyer

Documentation of the Conceptual Design

- See the Report Format document for input, output and error scenarios.

4.1.1 Report Formats

October 31, 2018

Input, Output and Error Scenarios

- **Scenario 1** = Begins on the "New Customer" or "New Order" page
  - Input 1 = Clicks "New Customer" option.
    - Output 1 = Displays new customer page.
      - Input 1.1 = User enters costumer's phone number, name and address and then clicks "Confirm."
        - Output 1.1 = User is taken to scenario 2.
      - Input 1.2 = User clicks "Back."
        - Output 1.2 = User is taken back to the beginning of scenario 1.
      - Input 1.3 = User clicks "Confirm" without entering all of the costumer's info.
        - Output 1.3 = User gets error message saying the required information is missing.
  - Input 2 = Clicks "New Order" option.
    - Output 2 = Displays find customer page.
      - Input 2.1 = User enters customer's phone number and clicks "Confirm."
        - Output 2.1 = User is taken to scenario 2.
      - Input 2.2 = User clicks "Back."
        - Output 2.2 = User is taken back to the beginning of scenario 1.
      - Input 2.3 = User clicks "Confirm" without entering the costumer's phone number.
        - Output 2.3 = User gets error message saying the required information is missing.

- **Scenario 2** = Begins on verifying customer's info page.

- Input 1 = User clicks “Confirm.”
  - Output 1 = User is taken to scenario 3.
- Input 2 = User clicks “Back.”
  - Output 1 = User is taken back to the beginning of scenario 1.

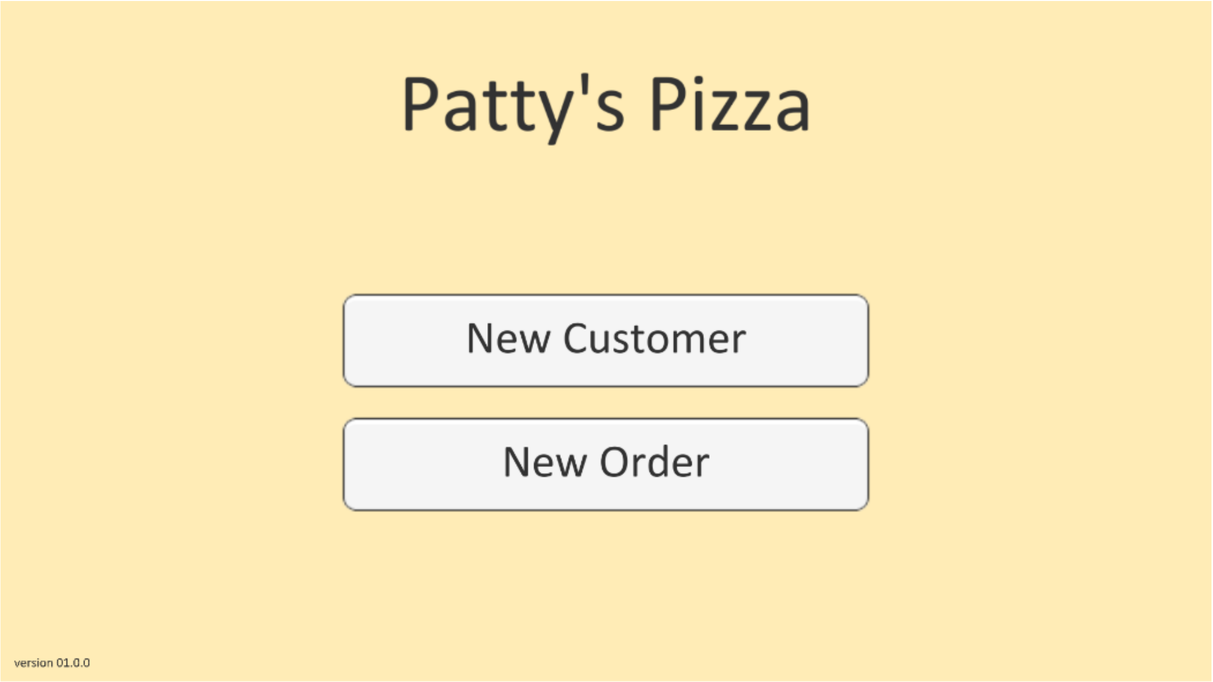
- **Scenario 3** = Begins on menu page.
  - Input 1 = User clicks customer’s chosen pizza size, topping, sauce, crust flavor and beverage and clicks “Add to Order.”
    - Output 1 = All pizza options chosen appear in the checkout section on the right side of the screen.
      - Input 1.1 = User clicks “Checkout.”
        - Output 1.1 = User is taken to scenario 4.
  - Input 2 = User clicks “Back.”
    - Output 2 = User is taken back to the beginning of scenario 1.
  - Input 3 = User clicks “Add to Order” or “Checkout” without entering any order options.
    - Output 3 = User gets error message saying the required information is missing.
- **Scenario 4** = Begins on checkout page.
  - Input 1 = User picks “ New Card” payment method.
    - Output 1 = Displays new card checkout page.
      - Input 1.1 = User enters customer’s card number, name on card and CCV/CVC code and clicks “Confirm.”
        - Output 1.1= User is taken to scenario 5.
      - Input 1.2 = User clicks “Back.”
        - Output 1.2 = User is taken back to the beginning of scenario 4.
      - Input 1.3 = User clicks “Cancel Order.”
        - Output 1.3 = User is taken back to scenario 1.
      - Input 1.4 = User clicks “Confirm” without entering required info.

            - Output 1.4 = User gets error message saying the required information is missing.

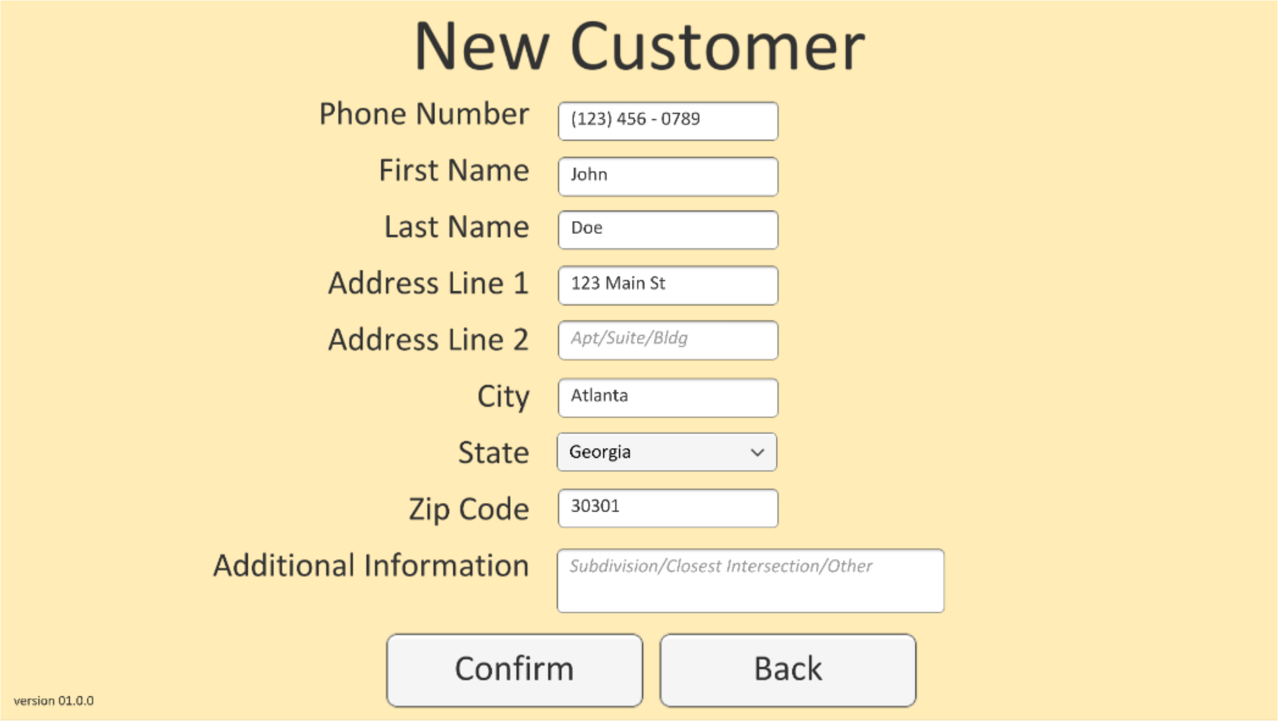
- Input 2 = User picks “ Used Card” payment method.
    - Output 2 = Displays used card checkout page.
        - Input 2.1 = User picks customer’s used card of choice and clicks “Confirm.”
            - Output 2.1= User is taken to scenario 5.
        - Input 2.2 = User clicks “Back.”
            - Output 2.2 = User is taken back to the beginning of scenario 4.
        - Input 2.3 = User clicks “Cancel Order.”
            - Output 2.3 = User is taken back to scenario 1.
        - Input 2.4 = User clicks “Confirm” without pick used card option.
            - Output 2.4 = User gets error message saying to pick a used card.
- Input 3 = User picks “ Cash” payment method.
    - Output 3 = Displays cash checkout page.
        - Input 3.1 = User enters amount of cash given to user and clicks “Confirm.”
            - Output 3.1= User is taken to scenario 5.
        - Input 3.2 = User clicks “Back.”
            - Output 3.2 = User is taken back to the beginning of scenario 4.
        - Input 3.3 = User clicks “Cancel Order.”
            - Output 3.3 = User is taken back to scenario 1.
        - Input 3.4 = User clicks “Confirm” without entering the amount of cash.
            - Output 3.4 = User gets error message saying the required information is missing.
- Input 4 = User picks “ Check” payment method.
    - Output 4 = Displays check checkout page.
        - Input 4.1 = User is given check and clicks “Confirm.”

            - Output 4.1= User is taken to scenario 5.
        - Input 4.2 = User clicks “Back.”
            - Output 4.2 = User is taken back to the beginning of scenario 4.
        - Input 4.3 = User clicks “Cancel Order.”
            - Output 4.3 = User is taken back to scenario 1.
    - Input 5 = User clicks “Back.”
        - Output 5 = User is taken back to scenario 3.
    - Input 6 = User clicks “Cancel Order.”
        - Output 6 = User is taken back to scenario 1.
    - Input 7 = User clicks “Confirm” without picking payment method.
        - Output 7 = User gets error message saying to pick payment method.
- **Scenario 5** = Printing customer’s receipt.
    - Input 1 = User clicks “Confirm” and doesn’t get error message in scenario 4.
        - Output 1 = System prints receipt that has all customer’s info, order info and payment method.

4.1.2 Screen Layouts



Screen 0 – main screen – if “New Customer” is pressed, go to screen 1. If “New Order” is pressed, go to screen 2a.



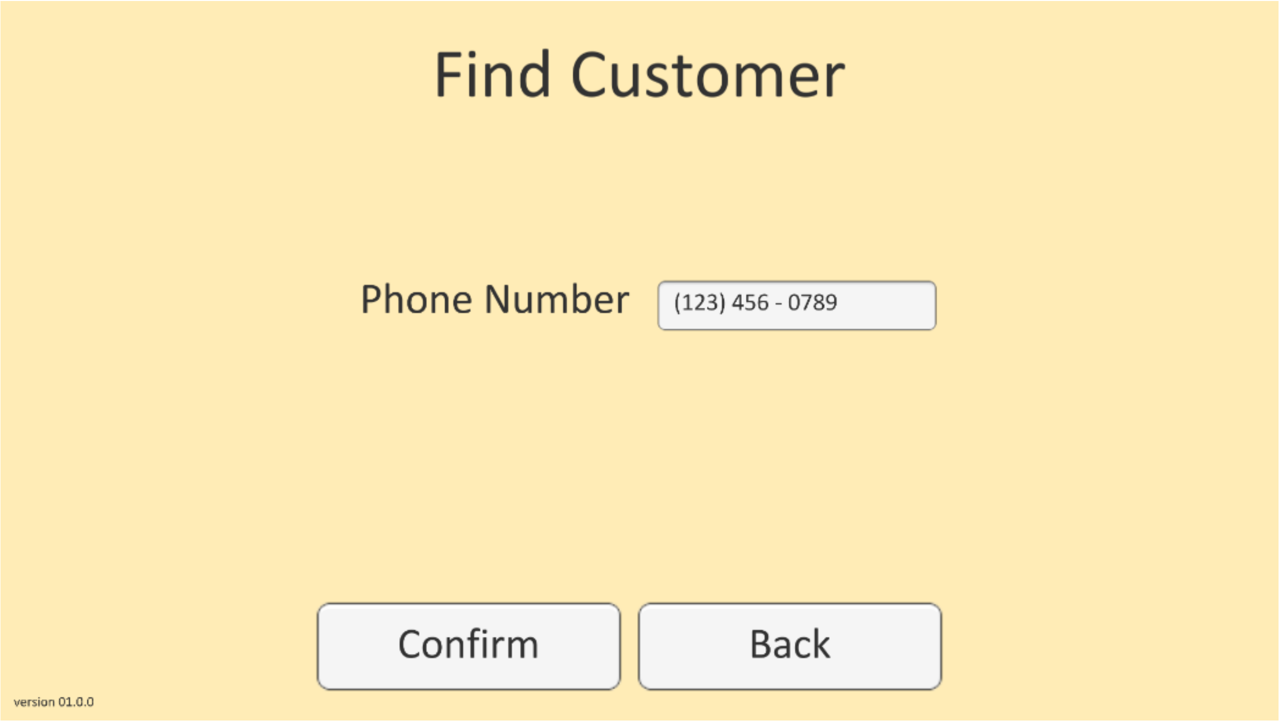
New Customer

Phone Number	(123) 456 - 0789
First Name	John
Last Name	Doe
Address Line 1	123 Main St
Address Line 2	Apt/Suite/Bldg
City	Atlanta
State	Georgia
Zip Code	30301
Additional Information	Subdivision/Closest Intersection/Other

Screen 1 – new customer screen – if "Confirm" is pressed, customer data is saved, go to screen 0. If "Back" is pressed, customer data is not saved, go to screen 0.

Requirements addressed: user must be able to create record for new customer, customer information must include telephone number, name, address, and additional location information

Use-Cases addressed: Make a new customer profile/record, enter customer's phone number, enter customer's name and address



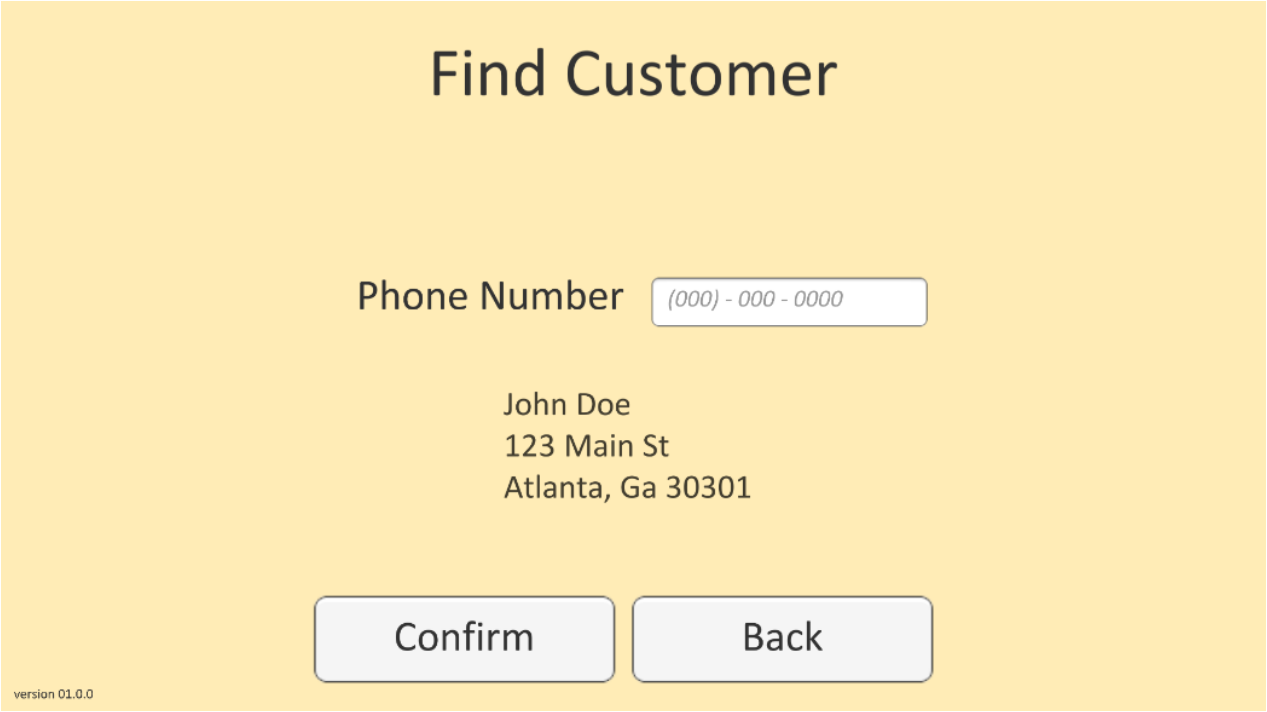
Find Customer

Phone Number (123) 456 - 0789

Screen 2a – find customer screen – if “Confirm” is pressed, program searches for the customer by inputted phone number. If customer is found, go to screen 2b. If customer is not found, go to screen 1. If “Back” is pressed, go to screen 0.

Requirements addressed: user must be able to find customer by telephone number

Use-Cases addressed: enter customer’s phone number



Find Customer

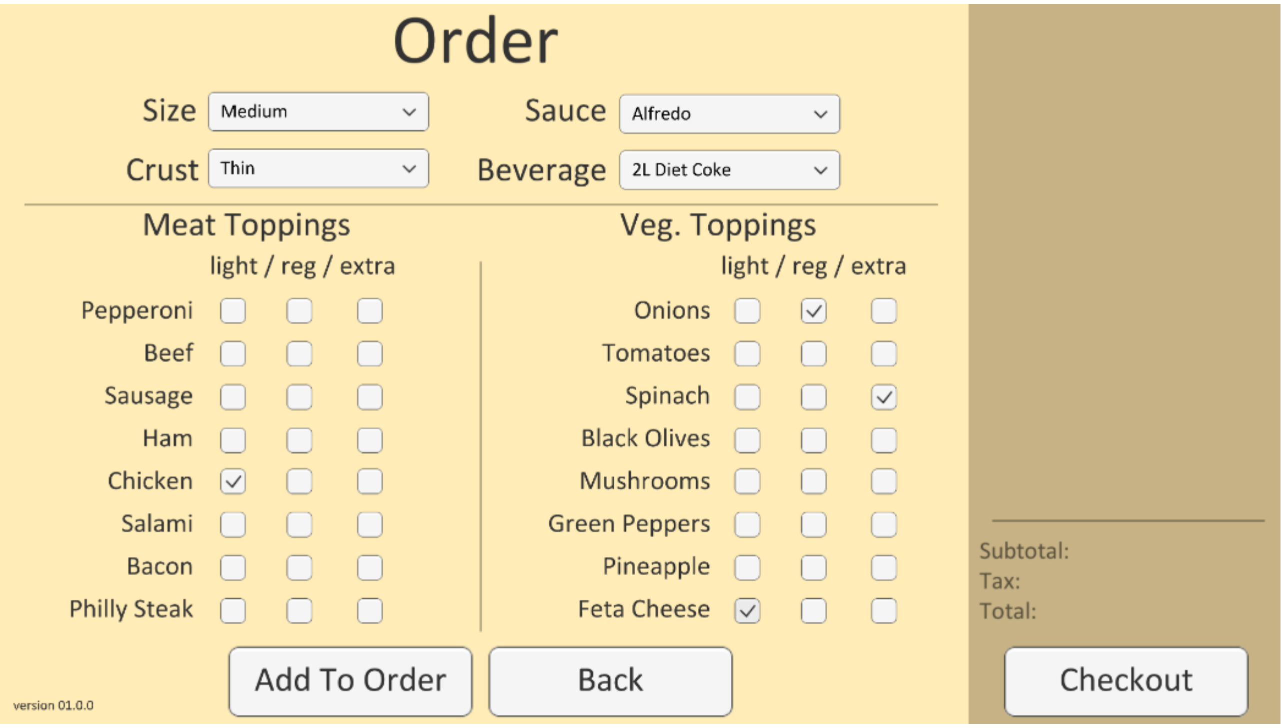
Phone Number

John Doe
123 Main St
Atlanta, Ga 30301

Screen 2b – find customer screen – if "Confirm" is pressed, customer is saved for this order, go to screen 3a. If "Back" is pressed, go to screen 0.

Requirements addressed: user must be able to find customer by telephone number, customer address and delivery information must be displayed for all customer transactions

Use-Cases addressed: enter customer's phone number



Order

Size: Medium

Sauce: Alfredo

Crust: Thin

Beverage: 2L Diet Coke

Meat Toppings

	light	reg	extra
Pepperoni	☐	☐	☐
Beef	☐	☐	☐
Sausage	☐	☐	☐
Ham	☐	☐	☐
Chicken	☑	☐	☐
Salami	☐	☐	☐
Bacon	☐	☐	☐
Philly Steak	☐	☐	☐

Veg. Toppings

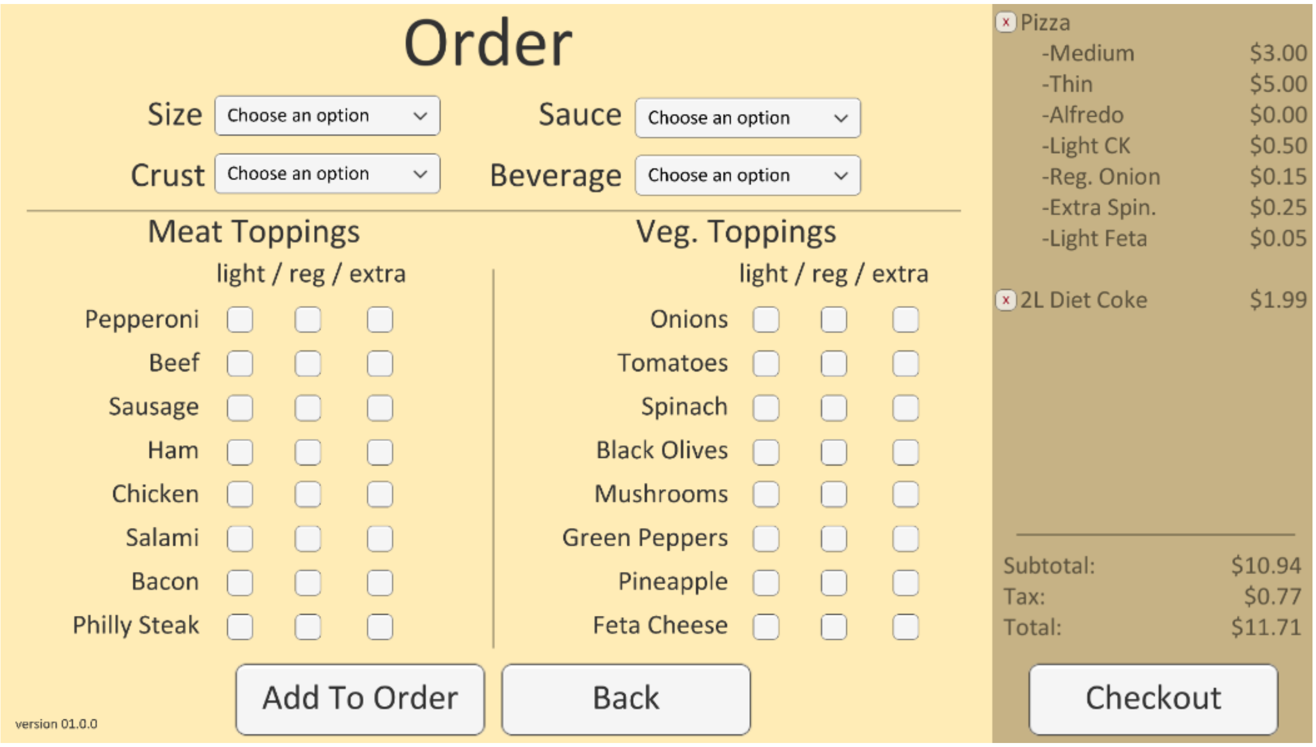
	light	reg	extra
Onions	☐	☑	☐
Tomatoes	☐	☐	☐
Spinach	☐	☐	☑
Black Olives	☐	☐	☐
Mushrooms	☐	☐	☐
Green Peppers	☐	☐	☐
Pineapple	☐	☐	☐
Feta Cheese	☑	☐	☐

Subtotal:
Tax:
Total:

Screen 3a – order screen – drop-down menus for pizza size, sauce, crust, and beverage. Check boxes for toppings. If “Add to Order” is pressed, currently selected options are added to order, go to screen 3b. If “Back” is pressed, order is deleted, go to screen 0. If “Checkout” is pressed while there are no order items, system prompts user to add items to order, stay on this screen. If “Checkout” is pressed while there are order items, order is completed, go to screen 4.

Requirements addressed: system must contain the complete restaurant menu, user must be able to have access to the menu items through a graphical user interface, menu must contain pizza sizes, toppings, crust options, and beverage options, system must calculate and display the order total

Use-Cases addressed: order taking, shows menu, choose pizza size, toppings, crust flavor, choose beverage

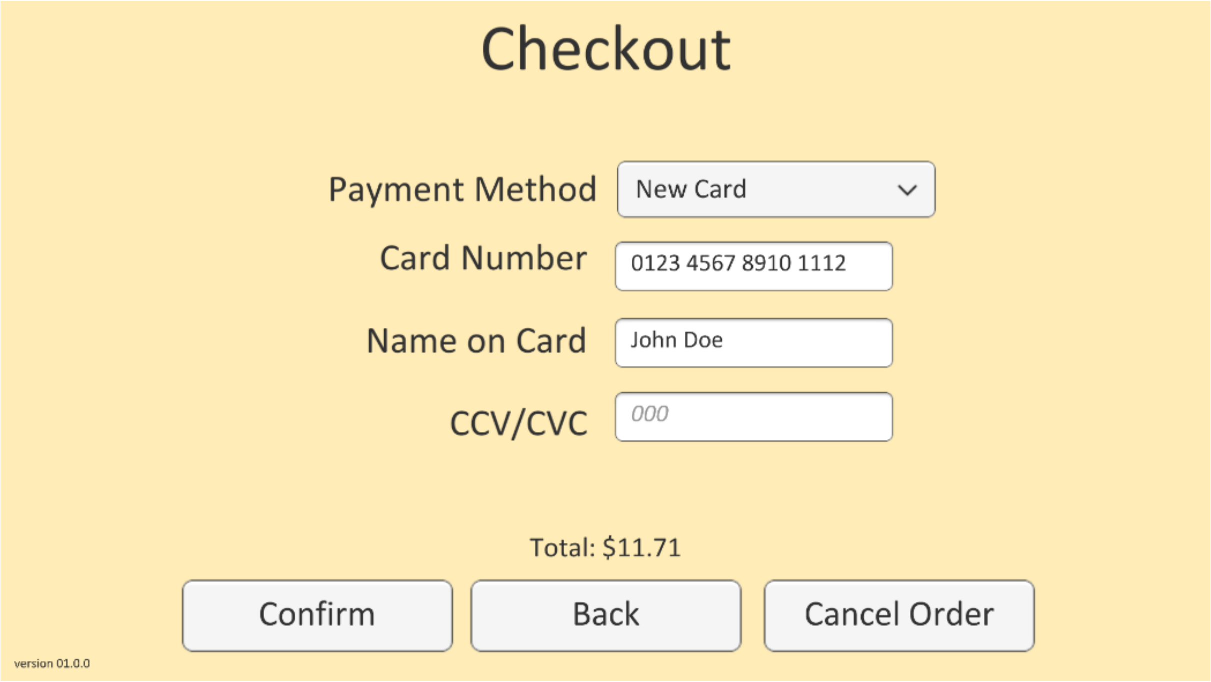


x Pizza	
-Medium	$3.00
-Thin	$5.00
-Alfredo	$0.00
-Light CK	$0.50
-Reg. Onion	$0.15
-Extra Spin.	$0.25
-Light Feta	$0.05
x 2L Diet Coke	$1.99
Subtotal:	$10.94
Tax:	$0.77
Total:	$11.71

Screen 3b – order screen after item is added – drop-down menus for pizza size, sauce, crust, and beverage. Check boxes for toppings. If "x" box beside order item is pressed, item is removed from order. If "Add to Order" is pressed, currently selected options are added to order on right side of screen with prices and added to total cost, stay on this screen. If "Back" is pressed, order is deleted, go to screen 0. If "Checkout" is pressed, order is completed, go to screen 4.

Requirements addressed: system must contain the complete restaurant menu, user must be able to have access to the menu items through a graphical user interface, menu must contain pizza sizes, toppings, crust options, and beverage options, system must calculate and display the order total

Use-Cases addressed: order taking, shows menu, choose pizza size, toppings, crust flavor, choose beverage



Checkout

Payment Method: New Card

Card Number: 0123 4567 8910 1112

Name on Card: John Doe

CCV/CVC: 000

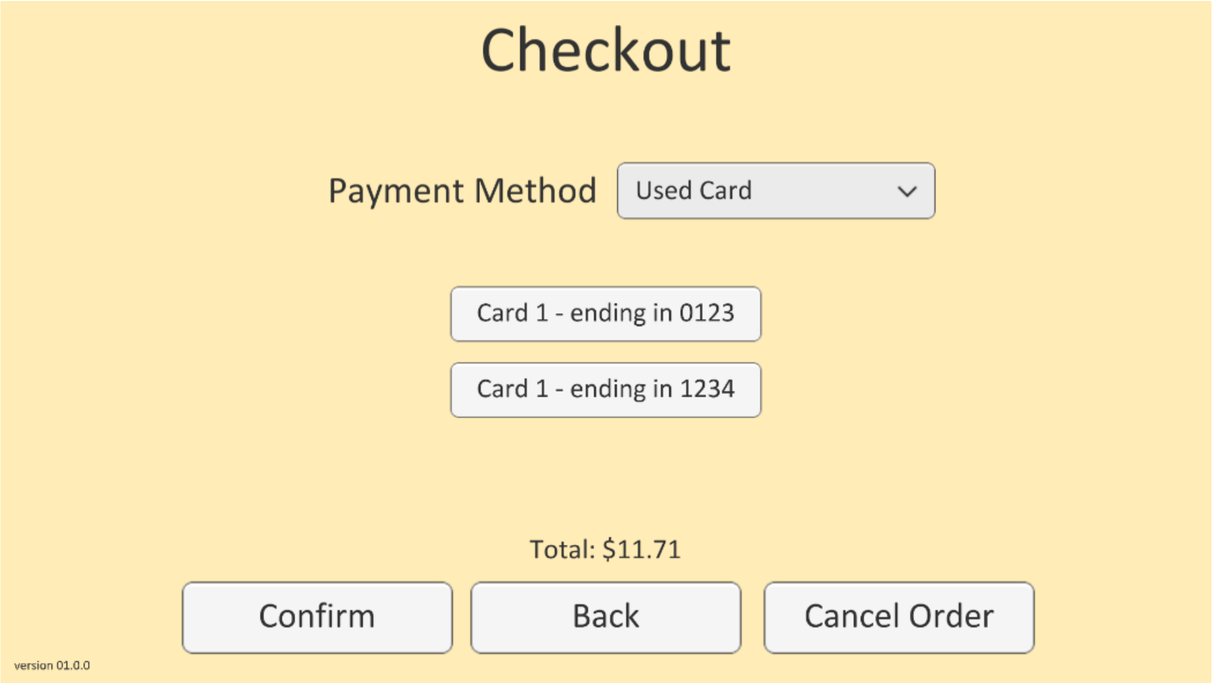
Total: $11.71

version 01.0.0

Screen 4a – payment screen – shows if “new card” is selected from the drop-down menu. User inputs card information and presses “confirm” to save transaction information, charge the amount to the card, send order to kitchen, and complete the transaction, go to screen 5a (receipt). If “Back” is pressed, go to screen 3b. If “Cancel Order” is pressed, order is removed, go to screen 0.

Requirements addressed: system must calculate and display the order total, system must process payments, payments must be processed in the form of credit cards, system must retain information on type of payment and amount charged

Use-Cases addressed: process any type of payment, process and submit order



Checkout

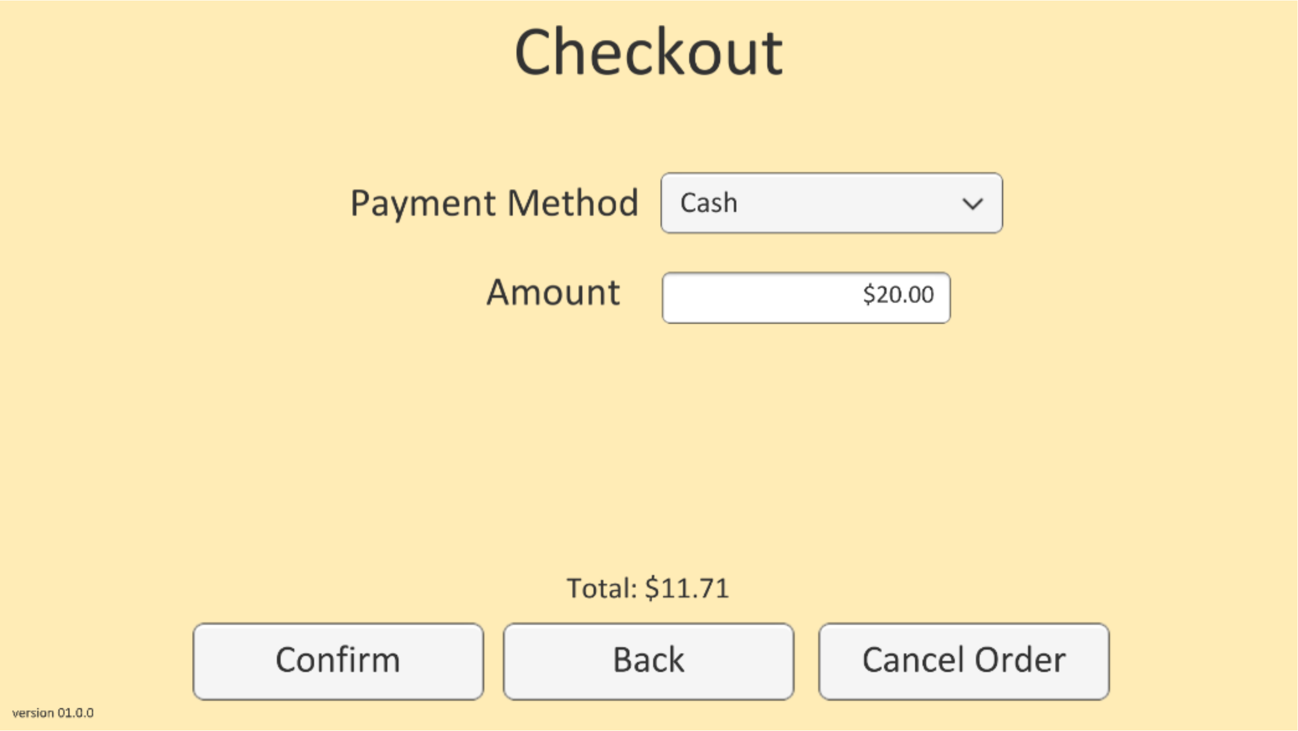
Payment Method: Used Card

Total: $11.71

Screen 4b – payment screen – shows if “used card” is selected from the drop-down menu. User selects card option such as “card 1” or “card 2” and presses “confirm” to charge the amount to the card, order is sent to kitchen, transaction is complete, go to screen 5a (receipt). If “Back” is pressed, go to screen 3b. If “Cancel Order” is pressed, order is removed, go to screen 0.

Requirements addressed: system must calculate and display the order total, system must process payments, payments must be processed in the form of credit cards, system must retain information on type of payment and amount charged

Use-Cases addressed: process any type of payment, process and submit order



Checkout

Payment Method: Cash

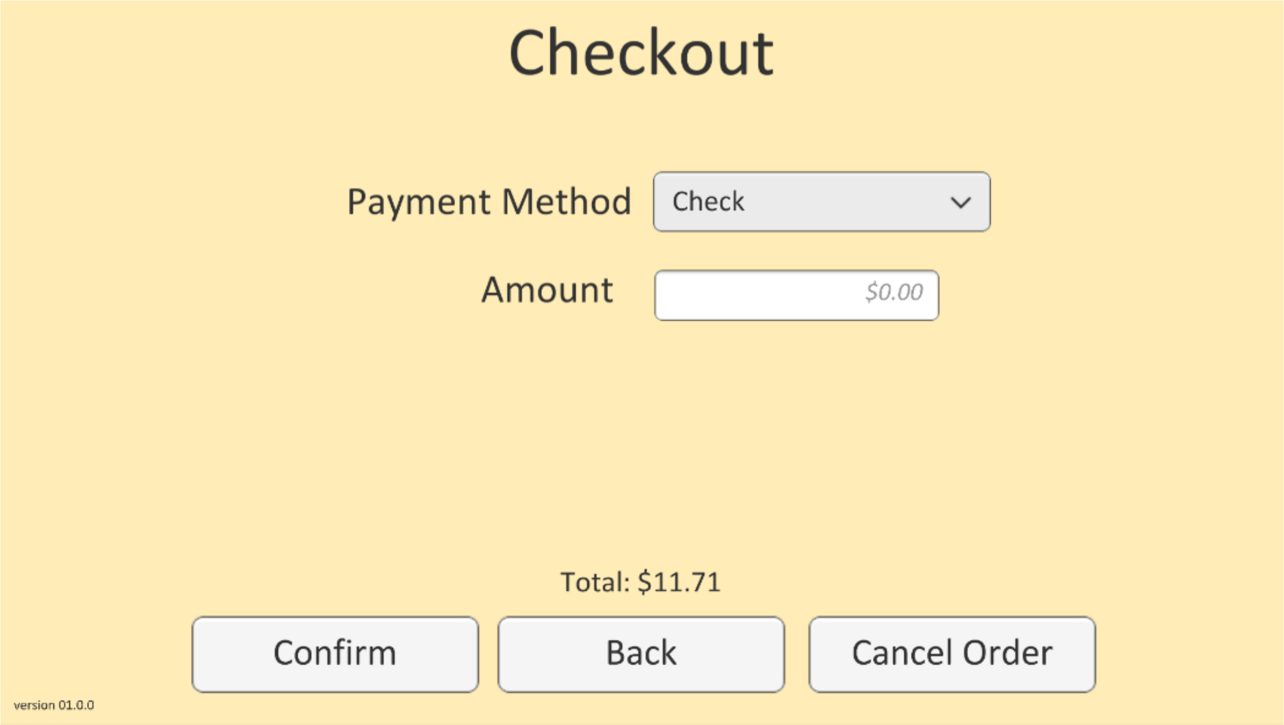
Amount: $20.00

Total: $11.71

Screen 4c – payment screen – shows if “cash” is selected from the drop-down menu. User inputs cash amount received and presses “confirm” to complete the transaction, order is sent to kitchen, receipt will show change due, go to screen 5b (receipt). If “Back” is pressed, go to screen 3b. If “Cancel Order” is pressed, order is removed, go to screen 0.

Requirements addressed: system must calculate and display the order total, system must process payments, payments must be processed in the form of cash, system must retain information on type of payment and amount charged

Use-Cases addressed: process any type of payment, process and submit order



Checkout

Payment Method: Check

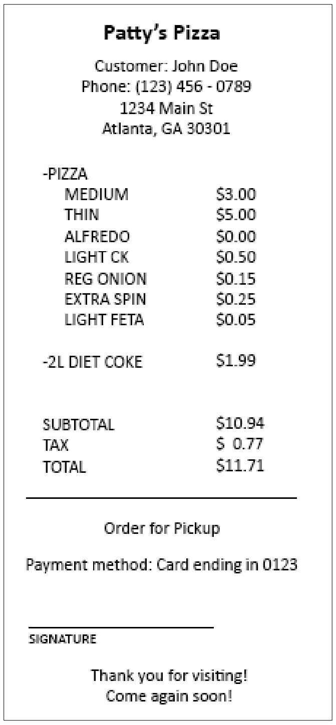
Amount: $0.00

Total: $11.71

Screen 4d – payment screen – shows if "check" is selected from the drop-down menu. User inputs amount on check, presses "confirm," and runs check through check scanner to complete the transaction, order is sent to kitchen, go to screen 5c (receipt). If "Back" is pressed, go to screen 3b. If "Cancel Order" is pressed, order is removed, go to screen 0.

Requirements addressed: system must calculate and display the order total, system must process payments, payments must be processed in the form of checks, system must retain information on type of payment and amount charged

Use-Cases addressed: process any type of payment, process and submit order



Patty's Pizza

Customer: John Doe
Phone: (123) 456 - 0789
1234 Main St
Atlanta, GA 30301

-PIZZA	
MEDIUM	$3.00
THIN	$5.00
ALFREDO	$0.00
LIGHT CK	$0.50
REG ONION	$0.15
EXTRA SPIN	$0.25
LIGHT FETA	$0.05
-2L DIET COKE	$1.99
SUBTOTAL	$10.94
TAX	$ 0.77
TOTAL	$11.71

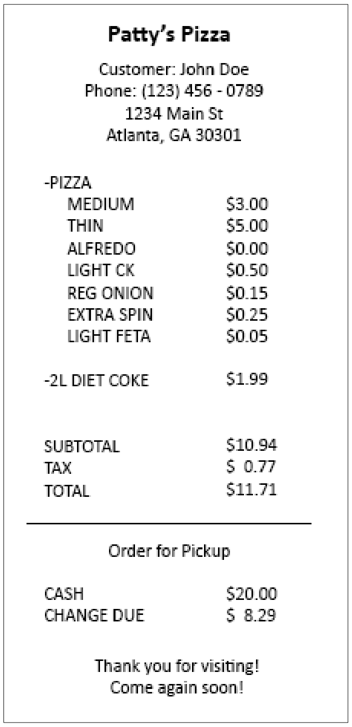
Order for Pickup

Payment method: Card ending in 0123

SIGNATURE

Thank you for visiting!
Come again soon!

5a



Patty's Pizza

Customer: John Doe
Phone: (123) 456 - 0789
1234 Main St
Atlanta, GA 30301

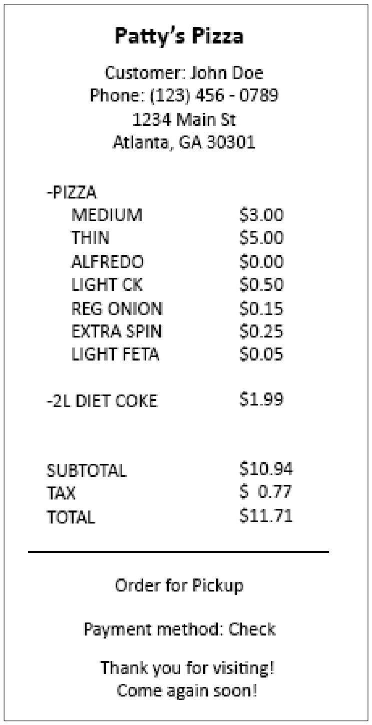
-PIZZA	
MEDIUM	$3.00
THIN	$5.00
ALFREDO	$0.00
LIGHT CK	$0.50
REG ONION	$0.15
EXTRA SPIN	$0.25
LIGHT FETA	$0.05
-2L DIET COKE	$1.99
SUBTOTAL	$10.94
TAX	$ 0.77
TOTAL	$11.71

Order for Pickup

CASH	$20.00
CHANGE DUE	$ 8.29

Thank you for visiting!
Come again soon!

5b



Patty's Pizza

Customer: John Doe
Phone: (123) 456 - 0789
1234 Main St
Atlanta, GA 30301

-PIZZA	
MEDIUM	$3.00
THIN	$5.00
ALFREDO	$0.00
LIGHT CK	$0.50
REG ONION	$0.15
EXTRA SPIN	$0.25
LIGHT FETA	$0.05
-2L DIET COKE	$1.99
SUBTOTAL	$10.94
TAX	$ 0.77
TOTAL	$11.71

Order for Pickup

Payment method: Check

Thank you for visiting!
Come again soon!

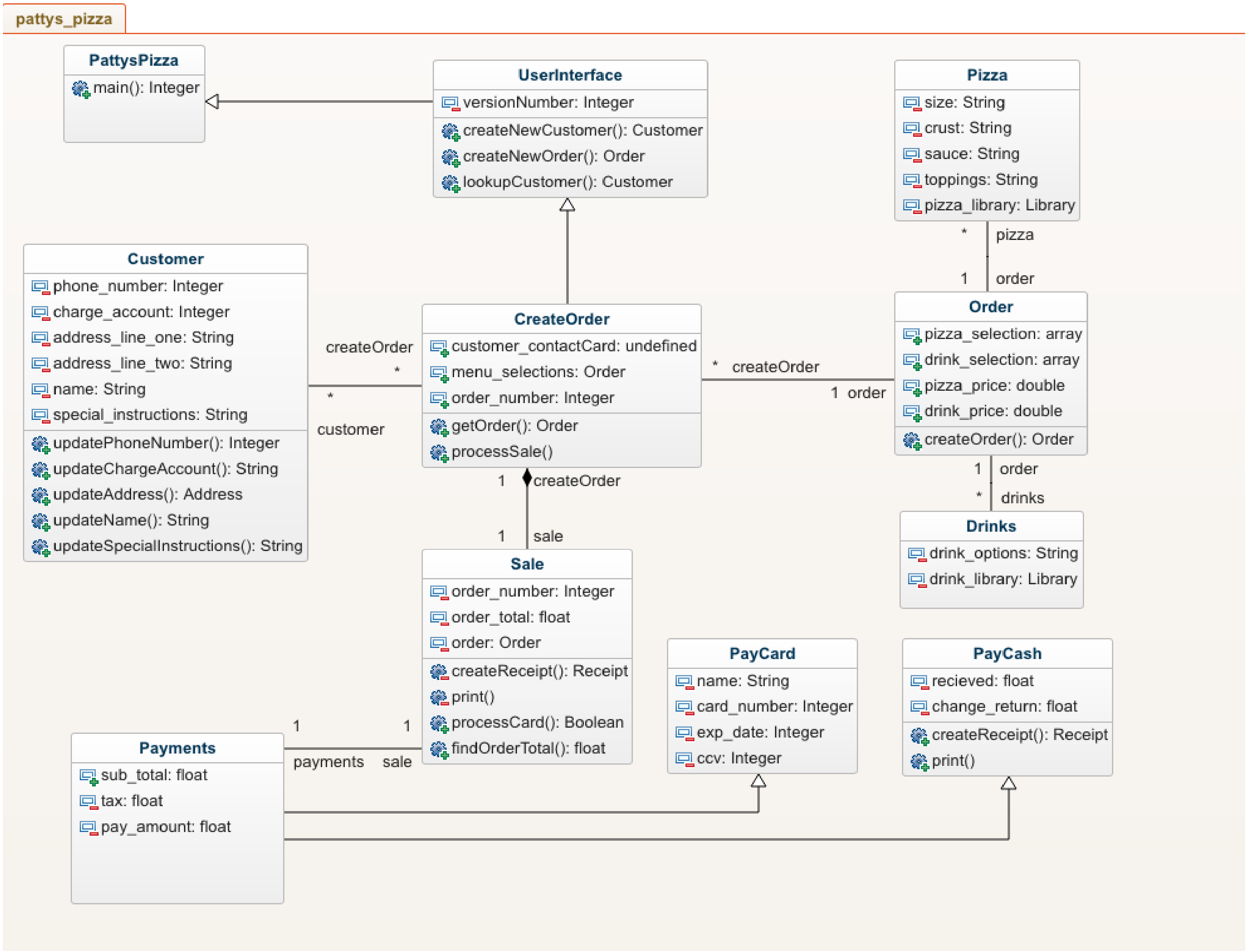
5c

Screens 5a, 5b, 5c – receipts - shows customer information, order information and prices, total cost, payment method, amount paid and change due if customer uses cash, and signature spot if customer uses card. System then defaults back to screen 0.

Requirements addressed: system must print a receipt when transaction is complete, the receipt must contain a place for customer signature if payment type used is credit card, customer information, a list of items ordered, information on if order is for delivery or pickup, and total amount due

Use-Cases addressed: print a receipt, prints a receipt with place to sign and customer's information

4.2 Class Diagrams



4.3 Class Documentation and Supporting Text Specification

The Patty's Pizza application will have twelve classes in all. Each class will be named in a way that facilitates ease of readability and understanding for both the developers and future maintainers. The classes included in this application are as follows:

- PattysPizza (contains main())
- UserInterface
- CreateOrder
- Order
- Pizza
- Drinks
- Customer
- Sale
- Payments
- PayCard
- PayCash
- PayCheck

PattysPizza

PattysPizza contains the main() method which has a return type of integer by default (C++). main() creates a new instance of UserInterface, then calls the createNewOrder() function. There are no preconditions or postconditions, assuming the application starts normally.

UserInterface

UserInterface has a single private attribute, versionNumber. This attribute is not used in the operation of the class and is included for information and debugging purposes. The version number of the application is displayed on the bottom left hand corner of the user interface. There are no other attributes for this class.

When the createNewCustomer() function is called inside UserInterface, a new instance of the user interface object is created, presenting two options - New Customer and New Order. If the customer is not currently registered in the data base, the New Customer option must be selected, otherwise the user will select New Order. When New Customer is selected, a new customer database entry is created with the createNewCustomer() method. When New Order is selected, the lookupCustomer() method is called and returns the Customer object associated with the phone number provided by the customer. Once the Customer object has been established in UserInterface, a new CreateOrder object is instantiated with the Customer object and the createNewOrder() method is called with Customer as a parameter.

There are no preconditions for this class. The postcondition requires the existence of a customer object in the customer database with the appropriate attributes.

Customer

The Customer class functions solely as a template for creating customer objects and their entries in the customer database. The attributes for each customer are phone_number, charge_account, address_line_one, address_line_two, name and special_instructions. The customer class also includes methods to update each attribute as well as a method to add a new charge account (credit or debit card).

CreateOrder

The CreateOrder class contains the methods getOrder() and processSale(). This class is accessed with a call to createNewOrder() from UserInterface. createNewOrder() immediately assigns an order_number and makes a call to the getOrder() method in the Order class. When control is returned to CreateOrder, the processSale() method is called from the Sale class with order_number and the Order object created by the Order class.

The only precondition for createNewOrder is that a Customer object must be supplied as an identifier for the order. Postconditions are that an order_number (integer) must have been assigned and menu_selections (array) cannot be null.

Order

The Order class only has one method, getOrder(). getOrder() is called from the CreateOrder class and has four attributes: pizza_selection (array), drink_selection (array), pizza_price (double) and drink_price (double). getOrder() uses the user interface to provide options for building pizza and drink orders using the Pizza and Drinks classes. These attributes are assigned when the getOrder() method is called and Pizza and Drinks objects are instantiated. The instantiation of these objects is optional but required to move forward with the ordering process. If no objects are created (i.e. no items are ordered), the user will be prompted to either put in an order or cancel the order and start over.

Preconditions for Order are the same as the postconditions for CreateOrder. There are no post conditions, as anything that would have been a postcondition here is handled in CreateOrder before the processSale() method is called from that class.

Pizza

The Pizza class contains only a default constructor which sets all attributes to null and the attributes themselves, size, crust, sauce and toppings. The Pizza class also contains a library, called pizza_library, that lists the price associated with each pizza option available on the menu. When each item is selected through the user interface, the items prices is identified through the library. Then the option name and price are both stored respectively in the first two dimensions of a four-dimensional array (menu_selections) in CreateOrder using the same indices. The third and fourth dimensions are used to store similar information for the customers drink order.

There are no preconditions or postconditions for this class as it is only instantiated from the Order class. The Order class has its own pre- and postconditions.

Drinks

The Drinks class is identical in function to the Pizza class, though not in content. The only attribute for the Drinks class is drink_options. Just like the Pizza class, Drinks contains a library defining the price for every drink option available on the menu. When drink options are selected through the UI, the name of the drink and the associated price are stored in the third and fourth dimensions of the four-dimensional array (menu_selections) in CreateOrder.

There are no pre- or postconditions for Drinks as it is only instantiated from the Order class.

Sale

The sale class contains the following attributes: order_number, sub_total, menu_selections, customer, and payment_method. This class is instantiated in CreateOrder with customer, menu_selections and order_number as parameters for the constructor. When the processSale() method is called, it iterates through the menu_selections array to calculate the price. Then the function waits for the user interface selection for payment method. The user interface will give the options to use a saved card (from the Customer object), add a new card (makes a call to the addChargeAccount() method from the Customer class), pay with cash or pay with a check. The selection is saved and passed as a parameter along with sub_total, customer and order_number to the createReceipt() method from the Payments class.

The precondition for Sale is that no attributes can be null. There are no postconditions.

Payments

The Payments class's attributes are: sub_total, tax, order_total, payment_method, customer and order_number. It has one method called createReceipt() that is called from the Sale class. createReceipt() uses the local tax rate to calculate the order_total and uses the payment_method selection to determine which class instantiate next. If credit is selected, the PayCard method is instantiated and its takePayment() method is called with customer and order_total as parameters. If the PayCash method is selected, the PayCash class is instantiated and its takePayment() method is called with order_total as a parameter. Finally, if PayCheck is selected, the PayCheck class is instantiated and its takePayment() method is called with order_total as a parameter.

The Payments class precondition is that all attributes must be validly populated. The postcondition is that a payment method must be selected.

PayCard

The PayCard class has the following attributes: card_number, exp_date, ccv, customer and amount_due. Its takePayment method takes order_total as a parameter and runs the card for that amount. The card information is pulled from the customer object's properties. Once the payment has gone through, the print() method is called from inside takePayment(), the receipt is printing and the transaction is complete.

PayCard's precondition is that no attribute can be passed in null. There are no postconditions, however, if the card payment is not successful, the user will be notified and control will be passed to the createReceipt() method to wait for a new payment_method selection.

PayCash

The PayCash class has the attributes: received and change_return. The user interface will prompt the user to input the amount, in dollars and cents, received from the customer and display the change_return amount to be given back to the customer. Once payment is successful, the print() method is called and the transaction is complete.

PayCash's precondition is that no attribute can be passed in null. There are no postconditions. If the payment is not successful, control will be passed back to the createReceipt() method to wait for a new payment_method selection.

PayCheck

The PayCheck class has the attributes bank, account_number and routing_number. The user interface will prompt the user to input the amount, in dollars and cents, written on the check. The check scanner will then pull the routing number and account number from the check and run it through the bank. Once payment goes through, the print() method is called and the transaction is complete.

PayCheck's precondition is that no attribute can be passed in null. There are no postconditions. If the payment is not successful, control will be passed back to the createReceipt() method to wait for a new payment_method selection.

4.4 Database Table Descriptions

	pattys_pizza	Date	11/5/18
Team	Lane Dyer, Bayleigh Correll, Will Diamanduros, Meghan Dowell, Kyle Duncan		

Customer	Data type	Description
phone_number	int	This is the customer's phone number that will be used to look up the customer in future purchases.
charge_account	int	The charge account is the customer's card that is saved to their account with the store.
address_line_one	string	Customer's address that the pizza is delivered to.
address_line_two	string	If customer's address is an apartment number.
name	string	Customer's name.
special_intructions	string	Any instructions the customer has given for the driver to locate the house or approaching the house.
updatePhoneNumber()	int	This will update the customer's phone number
updateChargeAccount()	string	This will update the customer's card.
updateAddress()	string	Updates customer's address
updateName()	string	Updates customer's name
updateSpecialInstructions()	string	Allows to add any extra instructions.
addChargeAccount()	string	Allows customer to add a second card.

PattysPizza	Data Type	Description
main()	int	This initiates the UI

UserInterface	Data Type	Description
versionNumber	int	Shows the system is up-to-date
createNewCustomer()	Customer	Takes the employee to a menu to create a customer account
lookupCustomer()	Customer	Employee will lookup customer by phone number

CreateOrder	Data Type	Description
customer	Customer	Customer's account will appear here to create a easy order.
menu_selections	array	Customer's selections will appear here
order_number	int	Customer's order number

createNewOrder()	Order	Takes employee to the menu to create a new order for the customer.

Order	Data Type	Description
pizza_selection	array	The pizza that the customer has selected.
drink_selection	array	The beverage the customer has selected
pizza_price	double	The price of the pizza
drink_price	double	The price of the beverage
getOrder()	CreateOrder	Sends employee to the order summary page

Pizza	Data Type	Description
size	string	Large, medium, or small pizzas
crust	string	Regular or thin crust
sauce	string	Normal amount of sauce or light sauce
toppings	string	A variety of topping will be stored here
pizza_library	library	The created pizza will be stored here.

Drinks	Data Type	Description
drink_options	array	A variety of beverages will be stored here
drink_library	library	The drink choice will be stored here

Sale	Data Type	Description
order_number	int	Customer's order number
menu_selections	array	The customer's selection for pizza and drinks will be stored here
customer	Customer	Customer's information will be displayed here
payment_method	string	Card, cash or check
sub_total	double	Total before taxes
findOrderTotal()	float	Calculates order total
processSale	Sale	Completes the sale

Payments	Data Type	Description
sub_total	float	Total before taxes
tax	float	Amount added to the sub total
order_total	float	Total after taxes
payment_method	float	Card, cash or check
customer	Customer	Customer information will be displayed here
order_number	int	Customer's order number
createReceipt()	Payments	A receipt will be created and given to the customer

PayCard	Data Type	Description
card_number	int	Card account number

exp_date	int	The expiration date of the card being used
ccv	int	The security number on the card
customer	Customer	Customer information will be displayed here
print()	PayCard	Displays the payment method
takePayment()	PayCard	Processes the payment

PayCash	Data Type	Description
received	float	Amount the customer gave.
change_return	float	Amount employee returned to customer
print()	PayCash	Displays the payment method
takePayment()	PayCash	Processes the payment

PayCheck	Data Type	Description
bank	string	Bank name
account_number	int	Customer's bank account number
routing_number	int	Customer's routing number to the bank
print()	PayCheck	Displays the payment method
takePayment	PayCheck	Processes the payment

5. APPENDIX

5.1 Weekly Status Reports

Week One Status Report (10/8/2018 – 10/14/2018)

Assignment	Contributor	Status	Start Date	Completion Date
Schedule/Milestones	Bayleigh C.	Complete – 100%	10/9/18	10/10/18
Scope Statement	Bayleigh C.	Complete – 100%	10/13/18	10/14/18
Test Plan	Will D.	In Progress – 70%	10/13/18	N/A
Gantt Chart	Bayleigh C.	In Progress – 75%	10/13/18	N/A
Team Organization	Bayleigh C.	Complete – 100%	10/13/18	10/14/18
Use Case Diagram	Will D.	In Progress – 50%	10/14/18	N/A

Week Two Status Report (10/15/2018 – 10/21/2018)

Assignment	Contributor	Status	Start Date	Completion Date
Test Plan	Will D.	Complete – 100%	10/13/18	10/15/18
Gannt Chart	Bayleigh C.	Complete – 100%	10/13/18	10/15/18
Use Case Diagram	Will D.	Complete – 100%	10/14/18	10/15/18
Work Breakdown Structure	Bayleigh C.	Complete – 100%	10/15/18	10/16/18
Requirements Definition Document	Meghan D.	Complete – 100%	10/15/18	10/17/18
Use Case Flow of Events Document	William D.	Complete – 100%	10/16/18	10/17/18
Data Management Plan	Lane D.	Complete – 100$	10/18/18	10/21/21

Week Three Status Report (10/22/2018 – 10/28/2018)

Assignment	Contributor	Status	Start Date	Completion Date
Paper Prototype	Meghan D.	Complete – 100%	10/22/18	10/23/18
Entity Relationship Diagram	Meghan D.	Complete – 100%	10/23/18	10/24/18
System requirements specification document	Will D.	Complete – 100%	10/23/18	10/24/18
Technical Systems Description	Lane D.	Complete – 100%	10/24/18	10/26/18
Class Diagrams	Kyle D.	Complete -100%	10/25/18	10/25/18
State Transition Diagram	Meghan D.	Complete – 100%	10/25/18	10/26/18

Week Four Status Report (10/29/18 – 11/4/18)

Assignment	Contributor	Status	Start Date	Completion Date
Polished Prototype	Meghan D.	Complete – 100%	10/29/18	10/30/18
Class Documentation	Kyle D.	Complete – 100%	10/30/18	10/31/18
Prototype Screenshots/Layouts	Meghan D.	Complete – 100%	10/30/18	10/30/18
Report Formats	Will D.	Complete – 100 %	10/31/18	11/1/18
Conceptual System Design	Will D.	Complete – 100%	11/3/18	11/4/18

Week Five Status Report (11/5//18 – 11/11/18)

Assignment	Contributor	Status	Start Date	Completion Date
Database Table Descriptions	Lane D.	Complete 100%	11/5/18	11/6/18
Supporting Class Specification	Kyle D.	Complete - 100%	11/5/18	11/5/18
Project Video Recording/Editing	All	In Progress – 80%	11/7/18	N/A

Week Six Status Report (11/12/18 – 11/18/18)

Assignment	Contributor	Status	Start Date	Completion Date
Project Video Recording/Editing	All	Complete – 100%	11/7/18	11/12/18
Report Finalization and submission	All	Complete – 100 %	11/14/18	11/14/18
Due Date:				11/16/18

5.2 Time Sheets

Week One Time Sheet – Bayleigh Correll		
Day	Summary	Total Hours
Monday (10/8/18)	N/A	N/A
Tuesday (10/9/18)	Attended initial team meeting (one hour) and began working on the schedule/milestones task for one hour	2 hours
Wednesday (10/10/18)	Finished, edited, and revised schedule and be/milestones task for an hour	1 hour
Thursday (10/11/18)	N/A	N/A
Friday (10/12/18)	N/A	N/A
Saturday (10/13/18)	Finished scope statement and began setting up the Gantt chart for one hour and started and worked on the team organization page for an 45 minutes	1 hour, 45 minutes
Sunday (10/14/18)	Attended week 1 team meeting (one hour) and continued to work on the Gantt chart and team organization pages for an hour and 15 minutes	2 hours, 15 minutes
Total Hours Worked This week		7 hours

Week Two Time Sheet – Bayleigh Correll		
Day	Summary	Total Hours
Monday (10/15/18)	Finished, revised, and edited Gantt chart and team organization pages for an hour	1 hour
Tuesday (10/16/18)	Began, finished, edited, and revised the work breakdown structure for the project for one hour	1 hour
Wednesday (10/17/18)	N/A	N/A
Thursday (10/18/18)	N/A	N/A
Friday (10/19/18)	Began compilation of completed pages and formatting of project report for 45 minutes	45 minutes
Saturday (10/20/18)	N/A	N/A
Sunday (10/21/18)	Attended weekly team meeting (one hour, 15 minutes) and	1 hour, 15 minutes

	updated weekly time sheet and weekly status report for 15 minutes	
Total Hours Worked This week		4 hours, 15 minutes

Week Three Time Sheet – Bayleigh Correll		
Day	Summary	Total Hours
Monday (10/22/18)	N/A	N/A
Tuesday (10/23/18)	N/A	N/A
Wednesday (10/24/18)	Compiled newly completed documents and formatted them into the project report for half an hour	30 minutes
Thursday (10/25/18)	N/A	N/A
Friday (10/26/18)	N/A	N/A
Saturday (10/27/18)	N/A	N/A
Sunday (10/28/18)	Attended weekly meeting (one hour), updated weekly time sheet and weekly status report for 30 minutes	1 hour, 30 minutes
Total Hours Worked This week		2 hours

Week Four Time Sheet – Bayleigh Correll		
Day	Summary	Total Hours
Monday (10/29/18)	N/A	N/A
Tuesday (10/30/18)	N/A	N/A
Wednesday (10/31/18)	N/A	N/A
Thursday (11/1/18)	Compiled newy completed documents and formatted them into the project report 15 minutes	15 minutes
Friday (11/2/18)	N/A	
Saturday (11/3/18)	N/A	N/A
Sunday (11/4/18)	Attended weekly team meeting (one hour), updated weekly time sheets and weekly status report for 20 minutes	1 hour, 20 minutes
Total Hours Worked This week		1 hour, 35 minutes

Week Five Time Sheet – Bayleigh Correll		
Day	Summary	Total Hours
Monday (11/5/18)	N/A	N/A
Tuesday (11/6/18)	N/A	N//A
Wednesday (11/7/18)	Wrote and edited ‘script’ for video recording for assigned tasks for an hour	1 hour
Thursday (11/8/18)	N/A	N/A
Friday (11/9/18)	N/A	N/A
Saturday (11/10/18)	Compiled finished documents and formatted them into the group project report for 15 minutes	15 minutes
Sunday (11/11/18)	Attended weekly team meeting (one hour), updated weekly time sheets and weekly status report for 15 minutes	1 hour, 15 minutes
Total Hours Worked This week		2 hours

Week Six Time Sheet – Bayleigh Correll		
Day	Summary	Total Hours
Monday (11/12/18)	Wrote introduction for project report for 15 minutes	15 minutes
Tuesday (11/13/18)	N/A	N/A
Wednesday (11/14/18)	Attended final team meeting (one hour and 15 minutes), finished final revision and formatting for the project for 30 minutes, updated and finalized weekly time sheets and status reports for an hour	2 hours, 45 minutes
Thursday (11/15/18)	N/A	
Friday (11/16/18))	N/A (Due Date)	N/A
Saturday (11/17/18)	N/A	N/A
Sunday (111/18/18)	N/A	N/A
Total Hours Worked This week		3 hours

Week One Time Sheet – William Diamanduros		
Day	Summary	Total Hours
Monday (10/8/18)	N/A	N/A
Tuesday (10/9/18)	Attended initial team meeting (one hour)	1 hour
Wednesday (10/10/18)	N/A	N/A

Thursday (10/11/18)	N/A	N/A
Friday (10/12/18)	N/A	N/A
Saturday (10/13/18)	Worked on test plan for two hours	2 hours
Sunday (10/14/18)	Attended weekly meeting (one hour), updated weekly time sheet for 15 minutes, finished and edited test plan for one hour and worked on finished use-case diagram for one hour and 15 minutes	3 hour, 30 minutes
Total Hours Worked This week		6 hours, 15 minutes

Week Two Time Sheet – William Diamanduros		
Day	Summary	Total Hours
Monday (10/15/18)	N/A	N/A
Tuesday (10/16/18)	Started, finished, and revised flow of events document for one hour and 15 minutes	1 hour, 15 minutes
Wednesday (10/17/18)	N/A	N/A
Thursday (10/18/18)	N/A	N/A
Friday (10/19/18)	N/A	N/A
Saturday (10/20/18)	N/A	N/A
Sunday (10/21/18)	Attended weekly meeting (one hour, 15 minutes), updated weekly time sheet for 15 minutes	1 hour, 30 minutes
Total Hours Worked This week		2 hours, 45 minutes

Week Three Time Sheet – William Diamanduros		
Day	Summary	Total Hours
Monday (10/22/18)	N/A	N/A
Tuesday (10/23/18)	Worked on software requirements specification document for 2 hours	2 hours
Wednesday (10/24/18)	Finished and revised software specification document for one hour	1 hour
Thursday (10/25/18)	N/A	N/A
Friday (10/26/18)	N/A	N/A
Saturday (10/27/18)	N/A	N/A

Sunday (10/28/18)	Attended weekly meeting (one hour), updated weekly time sheet for 15 minutes	1 hour, 15 minutes
Total Hours Worked This week		4 hours, 15 minutes

Week Four Time Sheet – William Diamanduros		
Day	Summary	Total Hours
Monday (10/29/18)	N/A	N/A
Tuesday (10/30/18)	N/A	N/S
Wednesday (10/31/18)	Worked on report format for four hours	4 hours
Thursday (11/1/18)	Finished and revised report format for one hour	1 hour
Friday (11/2/18)	N/A	N/A
Saturday (11/3/18)	Worked on conceptual system design for 2 hours	2 hours
Sunday (11/4/18)	Attended weekly meeting (one hour), updated weekly time sheet for 15 minutes	1 hour, 15 minutes
Total Hours Worked This week		8 hours, 15 minutes

Week Five Time Sheet – William Diamanduros		
Day	Summary	Total Hours
Monday (11/5/18)	N/A	N/A
Tuesday (11/6/18)	N/A	N/A
Wednesday (11/7/18)	Recorded Video Portion of the project for one hour	1 hour
Thursday (11/8/18)	N/A	N/A
Friday (11/9/18)	N/A	N/A
Saturday (11/10/18)	N/A	N/A
Sunday (11/11/18)	Attended weekly meeting (one hour), updated weekly time sheet for 15 minutes	1 hour, 15 minutes
Total Hours Worked This week		2 hours, 15 minutes

Week Six Time Sheet – William Diamanduros		
Day	Summary	Total Hours
Monday (11/12/18)	N/A	N/A
Tuesday (11/13/18)	N/A	N/A
Wednesday (11/14/18)	Attended final team meeting (one hour and 15 minutes),	1 hour, 45 minutes

	finished final revision and formatting for the project for 30 minutes	
Thursday (11/15/18)	N/A	N/A
Friday (11/16/18))	N/A (Due date)	N/A
Saturday (11/17/18)	N/A	N/A
Sunday (111/18/18)	N/A	N/A
Total Hours Worked This week		1 hour, 45 minutes

Week One Time Sheet – Meghan Dowell		
Day	Summary	Total Hours
Monday (10/8/18)	N/A	N/A
Tuesday (10/9/18)	Attended initial team meeting (one hour)	1 hour
Wednesday (10/10/18)	N/A	N/A
Thursday (10/11/18)	N/A	N/A
Friday (10/12/18)	N/A	N/A
Saturday (10/13/18)	N/A	N/A
Sunday (10/14/18)	Attended weekly meeting (one hour)	1 hour
Total Hours Worked This week		2 hours

Week Two Time Sheet – Meghan Dowell		
Day	Summary	Total Hours
Monday (10/15/18)	Started, finished, and edited requirements definition document for two hours	2 hours
Tuesday (10/16/18)	N/A	N/A
Wednesday (10/17/18)	N/A	N/A
Thursday (10/18/18)	N/A	N/A
Friday (10/19/18)	N/A	N/A
Saturday (10/20/18)	N/A	N/A
Sunday (10/21/18)	Attended weekly meeting (one hour, 15 minutes)	1 hour, 15 minutes
Total Hours Worked This week		3 hours, 15 minutes

Week Three Time Sheet – Meghan Dowell		
Day	Summary	Total Hours
Monday (10/22/18)	Began paper prototype and worked on it for an hour and fifteen minutes	1 hour, 15 minutes

Tuesday (10/23/18)	Finished and revised paper prototype for an hour. Began, finished, and revised entity relationship diagram for an hour and fifteen minutes. Began, finished, and edited state transition diagram for an hour and fifteen minutes	3 hours, 30 minutes
Wednesday (10/24/18)	N/A	N/A
Thursday (10/25/18)	N/A	N/A
Friday (10/26/18)	N/A	N/A
Saturday (10/27/18)	N/A	N/A
Sunday (10/28/18)	Attended weekly meeting (one hour)	1 hour
Total Hours Worked This week		5 hours, 45 minutes

Week Four Time Sheet – Meghan Dowell		
Day	Summary	Total Hours
Monday (10/29/18)	Began and worked on polished prototype for two hours and fifteen minutes	2 hours, 15 minutes
Tuesday (10/30/18)	Finished and revised polished prototype for two hours and fifteen minutes. Began, finished, and revised screen layouts for two hours and fifteen minutes	4 hours, 30 minutes
Wednesday (10/31/18)	N/A	N/A
Thursday (11/1/18)	N/A	N/A
Friday (11/2/18)	N/A	N/A
Saturday (11/3/18)	N/A	N/A
Sunday (11/4/18)	Attended weekly meeting (one hour)	1 hour
Total Hours Worked This week		6 hours, 45 minutes

Week Five Time Sheet – Meghan Dowell		
Day	Summary	Total Hours
Monday (11/5/18)	N/A	N/A
Tuesday (11/6/18)	N/A	N/A
Wednesday (11/7/18)	Recorded video portions of project for one hour	1 hour
Thursday (11/8/18)	Edited all video portions of the project for two hours	2 hours

Friday (11/9/18)	N/A	N/A
Saturday (11/10/18)	N/A	N/A
Sunday (11/11/18)	Attended weekly meeting (one hour, fifteen minutes)	1 hour, 15 minutes
Total Hours Worked This week		4 hours, 15 minutes

Week Six Time Sheet – Meghan Dowell		
Day	Summary	Total Hours
Monday (11/12/18)	Recorded remaining video portions of project for an hour and a half. Finished editing remaining project videos for two hours	3 hours, 15 minutes
Tuesday (11/13/18)	N/A	N/A
Wednesday (11/14/18)	Attended final team meeting (one hour and 15 minutes),	1 hour, 15 minutes
Thursday (11/15/18)	N/A	N/A
Friday (11/16/18))	N/A	N/A
Saturday (11/17/18)	N/A	N/A
Sunday (111/18/18)	N/A	N/A
Total Hours Worked This week		4 hours, 45 minutes

Week One Time Sheet – Kyle Duncan		
Day	Summary	Total Hours
Monday (10/8/18)	Attended initial team meeting (one hour)	1 hour
Tuesday (10/9/18)	N/A	N/A
Wednesday (10/10/18)	N/A	N/A
Thursday (10/11/18)	N/A	N/A
Friday (10/12/18)	N/A	N/A
Saturday (10/13/18)	N/A	N/A
Sunday (10/14/18)	Attended weekly meeting (one hour)	1 hour
Total Hours Worked This week		2 hours

Week Two Time Sheet – Kyle Duncan		
Day	Summary	Total Hours
Monday (10/15/18)	N/A	N/A
Tuesday (10/16/18)	N/A	N/A
Wednesday (10/17/18)	N/A	N/A
Thursday (10/18/18)	N/A	N/A

Friday (10/19/18)	N/A	N/A
Saturday (10/20/18)	N/A	N/A
Sunday (10/21/18)	Attended weekly meeting (one hour, fifteen minutes)	1 hour, 15 minutes
Total Hours Worked This week		

Week Three Time Sheet – Kyle Duncan		
Day	Summary	Total Hours
Monday (10/22/18)	N/A	N/A
Tuesday (10/23/18)	N/A	N/A
Wednesday (10/24/18)	N/A	N/A
Thursday (10/25/18)	Began, finished, and revised class diagrams for four hours, fifty minutes	4 hours, 50 minutes
Friday (10/26/18)	N/A	N/A
Saturday (10/27/18)	N/A	N/A
Sunday (10/28/18)	Attended weekly meeting (one hour)	1 hour
Total Hours Worked This week		5 hours, 50 minutes

Week Four Time Sheet – Kyle Duncan		
Day	Summary	Total Hours
Monday (10/29/18)	N/A	N/A
Tuesday (10/30/18)	Began working on class documentation for three hours and fifteen minutes	3 hours, 15 minutes
Wednesday (10/31/18)	Finished and revised class documentation for three hours and thirty minutes	3 hours, 30 minutes
Thursday (11/1/18)	N/A	N/A
Friday (11/2/18)	N/A	N/A
Saturday (11/3/18)	N/A	N/A
Sunday (11/4/18)	Attended weekly meeting (one hour)	1 hour
Total Hours Worked This week		7 hours, 45 minutes

Week Five Time Sheet – Kyle Duncan		
Day	Summary	Total Hours
Monday (11/5/18)	Began supporting class specification and worked on it for two hours	2 hours

Tuesday (11/6/18)	Finished and revised supporting class specification for an hour and a half	1 hour
Wednesday (11/7/18)	N/A	N/A
Thursday (11/8/18)	N/A	N/A
Friday (11/9/18)	Recorded project video portions for an hour	1 hour
Saturday (11/10/18)	N/A	N/A
Sunday (11/11/18)	Attended weekly meeting (one hour, fifteen minutes)	1 hour, 15 minutes
Total Hours Worked This week		5 hours, 15 minutes

Week Six Time Sheet – Kyle Duncan		
Day	Summary	Total Hours
Monday (11/12/18)	N/A	N/A
Tuesday (11/13/18)	N/A	N/A
Wednesday (11/14/18)	Attended final team meeting (one hour and 15 minutes),	1 hour, 15 minutes
Thursday (11/15/18)	N/A	N/A
Friday (11/16/18))	N/A	N/A
Saturday (11/17/18)	N/A	N/A
Sunday (111/18/18)	N/A	N/A
Total Hours Worked This week		1 hour, 15 minutes

Week One Time Sheet – Lane Dyer		
Day	Summary	Total Hours
Monday (10/8/18)	Attended initial team meeting (one hour)	1 hour
Tuesday (10/9/18)	N/A	N/A
Wednesday (10/10/18)	N/A	N/A
Thursday (10/11/18)	N/A	N/A
Friday (10/12/18)	N/A	N/A
Saturday (10/13/18)	N/A	N/A
Sunday (10/14/18)	Attended week 1 team meeting (one hour)	1 hour
Total Hours Worked This week		2 hour

Week Two Time Sheet – Lane Dyer		
Day	Summary	Total Hours
Monday (10/15/18)	N/A	N/A
Tuesday (10/16/18)	N/A	N/A

Wednesday (10/17/18)	N/A	N/A
Thursday (10/18/18)	N/A	N/A
Friday (10/19/18)	N/A	N/A
Saturday (10/20/18)	N/A	N/A
Sunday (10/21/18)	Attended weekly team meeting (one hour, 15 minutes) and began working on the data management plan for 2 hours and 30 minutes	3 hours, 45 minutes
Total Hours Worked This week		3 hours 45 minutes

Week Three Time Sheet – Lane Dyer		
Day	Summary	Total Hours
Monday (10/22/18)	Finished and revised the data management plan for two hours	2 hours
Tuesday (10/23/18)	N/A	N/A
Wednesday (10/24/18)	N/A	N/A
Thursday (10/25/18)	Began working on the technical system descriptions for two hours	2 hours
Friday (10/26/18)	the technical system descriptions for an hour and a half	1 hour, 30 minutes
Saturday (10/27/18)	N/A	N/A
Sunday (10/28/18)	Attended weekly meeting (one hour)	1 hour
Total Hours Worked This week		6 hours, 30 minutes

Week Four Time Sheet – Lane Dyer		
Day	Summary	Total Hours
Monday (10/29/18)	N/A	N/A
Tuesday (10/30/18)	N/A	N/A
Wednesday (10/31/18)	N/A	N/A
Thursday (11/1/18)	N/A	N/A
Friday (11/2/18)	N/A	N/A
Saturday (11/3/18)	N/A	N/A
Sunday (11/4/18)	Attended weekly meeting (one hour)	1 hour
Total Hours Worked This week		1 hour

Week Five Time Sheet – Lane Dyer		

Day	Summary	Total Hours
Monday (11/5/18)	Began working on database table descriptions for two hours	2 hours
Tuesday (11/6/18)	Finished and revised database table descriptions for an hour and a half	1 hour, 30 minutes
Wednesday (11/7/18)	N/A	N/A
Thursday (11/8/18)	Recorded project video portions for an hour	1 hour
Friday (11/9/18)	N/A	N/A
Saturday (11/10/18)	N/A	N/A
Sunday (11/11/18)	Attended Weekly meeting for one hour and 15 minutes	1 hour, 15 minutes
Total Hours Worked This week		5 hours, 45 minutes

Week Six Time Sheet – Lane Dyer		
Day	Summary	Total Hours
Monday (11/12/18)	N/A	N/A
Tuesday (11/13/18)	N/A	N/A
Wednesday (11/14/18)	Attended final team meeting (one hour and 15 minutes),	1 hour, 15 minutes
Thursday (11/15/18)	N/A	N/A
Friday (11/16/18))	N/A	N/A
Saturday (11/17/18)	N/A	N/A
Sunday (111/18/18)	N/A	N/A
Total Hours Worked This week		1 hour, 15 minutes

5.3 Meeting Schedule and Attendance

Meeting	Date	Start Time	End Time	Summary and Attendance
Initial Team Meeting	10/9/18	12:00 PM	1:00 PM	During this meeting, each group member introduced themselves and summarized their strengths and weaknesses relating to the project (such as known programing languages, software knowledge, etc.) A very rough outline of the project schedule was drawn up and the project was dissected into the five parts: project planning, requirements documents, system design documents, system creation, and project wrap up. Tasks within the Project Planning phase were discussed and assigned. All group members were present.
Team Meeting #1	10/14/18	12:00 PM	1:00 PM	Completed project planning assignments (the scope statement, Gannt chart, and team organization documents) were reviewed by the entire group, and then revised if needed. Resumes were gathered and requirements for the next set of project planning assignments were discussed. The next part of the project, requirements documents, was discussed, and assignments for this part were assigned to group members. Current progress was reviewed, and changes were made to the schedule accordingly. All group members were present.
Team Meeting #2	10/21/18	12:15 PM	1:30 PM	Completed assignments were reviewed and revised if needed by the group. The requirements documents phase was then discussed. Requirements specification documents required for the project were discussed and assigned to group members, and classes and variables to use for diagrams/documents were established. Questions about the coding portion to ask the client/professor were recorded and given to Meghan to ask in class. All group members were present.
Team Meeting #3	10/28/18	12:00 PM	1:00 PM	Following the answer that no actual coding was required, the group removed the 'system creation' phase from the project schedule, as well as changes to any documents referring to this phase. Finished assignments were reviewed and revised and discussion about the polished prototype took place. The next set of the requirements specification documents phase tasks were discussed and assigned. All group members were present.

Team Meeting #4	11/4/18	12:00 PM	1:00 PM	The polished prototype was reviewed and edited, and from this, requirements for the class diagrams and database tables were laid out and the tasks were assigned. The rest of the 'system design documents' phase assignments were discussed and assigned as well (High level technical design and class specification). All group members were present.
Team Meeting #5	11/11/18	12:00 PM	1:15 PM	Project video guidelines were laid out and it was decided that each group member would be responsible for recording their own videos over their respective portions of the assignment. Any final concerns were addressed as well. Weekly status reports for each member were turned in and set to be compiled and added to the project report by the final meeting. All group members were present.
Final Team Meeting	11/14/18	5:00 PM	6:15 PM	Weekly status reports updated. The project report was reviewed, with each section of the project being carefully gone over by all group members. Final revisions were made. Video portions were compiled and edited. The final version of the project report and prototype were made ready for submission the following day. All group members were present.