

Kennesaw State University
College of Computing and Software Engineering

COFFEE & CODE

Project Plan

SWE 3313

Professor:
Jeff Adkisson

Group 6:
Elliot Larez
Garrett Heffner
Michael Butler
Sahan Reddy

Table of Contents

1. Scope	3
2. Schedule	4
3. Team Organization	7
4. Technical Description	12
5. Data Management Plan	15
6. Test Plan	17

1. Scope

Taking orders

- Show options on the screen
- Customize orders details
- Save selection in the program

Identifying customers:

- Add/Find customers in the customer list using names and phone numbers
- Manage anonymous customers with no names and phone numbers

Processing credit cards:

- Accept credit card payments in the program
- Verify credit card information and preventing leaks of information

Using reward system

- Accept reward points as an alternative to the credit card payment
- Add reward points for customers listed in the customer list

Loading configurations

- Read from the program data (JSON file) the menu and product prices
- Set the user configurations from the last run

Recording program data

- Upload the customer list and orders taken to the program data (JSON file)

Reporting sales

- Update sale report (CSV file,) including order details and customer data.

2. Schedule

Project Plan

Task ID	Name	Members	Start	Finish	Deadline	Time (man-hours)
1	Project Plan	Elliot, Garret, Michael, Sahan	Week 1	Week 2	Oct 9, 2022, 8:59 AM	10
1.1	Scope	Elliot	Week 1			2
1.2	Schedule					2
1.3	Team Organization	Michael, Sahan				1
1.4	Technical Description		Week 2	4		
1.5	Data Management Plan	Garret		1		
1.6	Test Plan			2		

Requirements

Task ID	Name	Members	Start	Finish	Deadline	Time (man-hours)
2	Requirements	Elliot, Garret, Michael, Sahan	Week 2	Week 3	Oct 18, 2022, 5:59 PM	15
2.1	Case Diagram	Elliot, Garret	Week 2			5
2.2	Case Flow					5
2.3	Documentation	Michael, Sahan	Week 3			10
2.4	Entity Relationship Diagram					5

UI Design

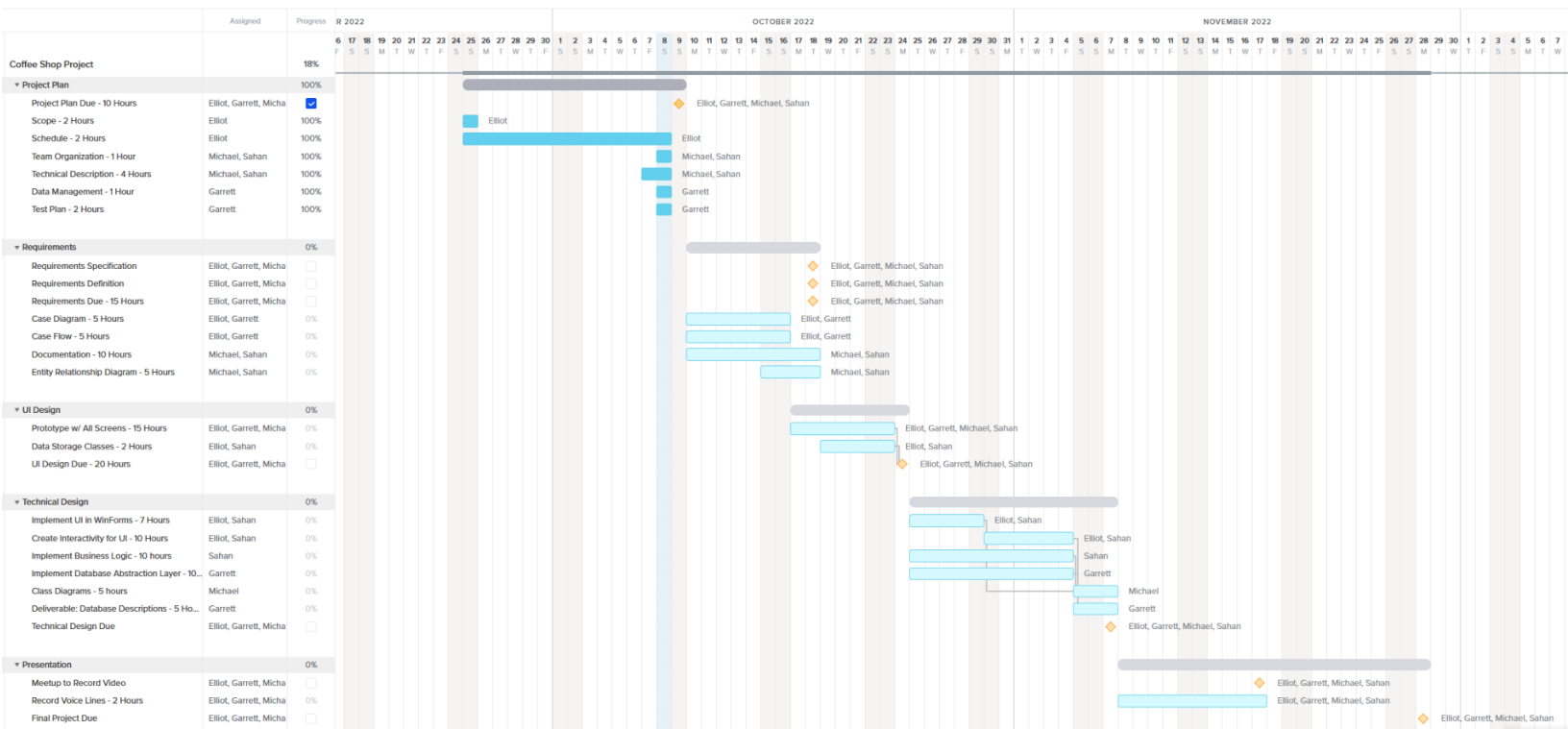
Task ID	Name	Members	Start	Finish	Deadline	Time (man-hours)
3	UI Design	Elliot, Garret Michael, Sahan	Week 3		Oct 24, 2022, 11:59 PM	20
3.1	Marvel prototype with every screen	Elliot, Garret Michael, Sahan	Week 3	Week 4		15
3.2	Data Storages Classes	Elliot, Sahan	Week 3	Week 4		2

Technical Design

Task ID	Name	Members	Start	Finish	Deadline	Time (man-hours)
4	Technical Design	Elliot, Garret Michael, Sahan	Week 4	Week 5	Nov 7, 2022, 11:59 PM	10
4.1	Class Diagram	Michael	Week 4			5
4.2	Database Descriptions	Garret				5
4.3	Implement UI in WinForms	Elliot Sahan	Week 5			7
4.4	Create Interactivity for UI					10
4.5	Implement Business Logic	Sahan				10

Final Testing and Presentation

Task ID	Name	Members	Start	Finish	Deadline	Time (man-hours)
5	Final Testing	Elliot, Garret Michael, Sahan	Week 6		Nov 28, 2022, 4:59 PM	10
6	Presentation	Elliot, Garret, Michael, Sahan	Week 7			10
6.1	Video Demo	Elliot, Michael	Week 7			5



[Gantt Chart Image Link](#)

3. Team Organization

Michael Butler - Team Organizer, UI Designer, Document Writer

Garrett Heffner - Database Designer, Secondary Programmer

Elliot Larez - Backend Coder,

Sahan Reddy - Technical Lead, Programmer

Jeff Adkisson - Customer

MICHAEL BUTLER

678-358-7901 | michaelbutler2002@icloud.com | mbutle98@students.kennesaw.edu

Work Experience

Supplemental Instruction Leader for Calculus 1 at Kennesaw State (Fall Semester 2022 - present)

Six Flags Over Georgia: Park Services (Spring & Summer 2018) and Retail Team Member (Summer 2021)

Education

Pebblebrook High School – GPA: 4.1- Class of 2021

Kennesaw State University as an Honor Student - Class of 2025

Working towards Bachelor's Degree in Computer Game Design & Development

GPA: 3.96, made President's List in Spring 2022

Major Courses

- Fundamentals of Game Design
- Intro to Software Engineering
- Programming and Problem Solving
- Discrete Structures

Technical Skills

- Mainly code in Java
- Working on C#
- Had experience working on GUI's (JavaScript [Code.org], JavaFX)

Skills

- Time Management
- Good communication
- Flexible
- Adapting to change

Projects

- Took on a leadership role in past group projects
- Built two games: one was JavaScript ([Trivia All Star](#)) and the other was on Java ([Dice Battle](#))

Experiences

Beta Club President at Pebblebrook High School

CCCEPA Actor at Pebblebrook High School

Tutor in Math and Chemistry at Pebblebrook High School

[Trivia All Star](#)

[Dice Battle](#)

Garrett Heffner**Computer Science Undergraduate****School Email:**

gheffner@students.kennesaw.edu

Education

Bachelor of Science in Computer Science

Currently Enrolled at Kennesaw State University

Skills:

Strong understanding of Java

Working knowledge of JavaScript, SQL

Took an Nvidia AI and Deep Learning course

Soft Skills:

Problem Solving

Creativity

Worth Ethic

Major Courses:

Programming and Problem Solving

Computer organization and Architecture

Intro to Database Systems

Data Structures

Projects:

Successfully lead team in project creating a sequential circuit for counting trashketball baskets

Elliot J. Larez

Acworth, GA - elarez@students.kennesaw.edu - (470) 707-7841

Education

Kennesaw State University-Marietta, GA

Expected Graduation: December 2024

Bachelor of Science in Computer Science

- Honors College Student
- Sophomore. Cumulative GPA: 4.00/4.00
- President's List in Fall 2021 and Spring 2022

Work Experience

Research Assistant (Visualization, Simulation, and 3D printing)

(Fall 2022 - Present)

KSU Cole College of Business

- Run simulations and graph databases in an Immersive Visualization Environment
- Collect data and construct a dashboard in Tableau to represent statistics
- Monitor the status of the university 3d printers and assist students in the process

Tutor/Lab Assistant

(Spring 2022)

KSU College of Computing and Software Engineering

- Explain to students programming concepts in individual appointments
- Manage an active plan to report student attendees and the material covered
- Assist a graduate teaching assistant in answering students' questions in a programming lab
- Proctoring the programming lab during midterms exams

Student Assistant

(Fall 2021)

KSU On-Campus Chick-fil-A

- Provide an exceptional experience to customers in a busy restaurant.
- Operate register machines with a constant flow of transactions.
- Complete cleaning routines through the facilities with team members.

Skills

Communication: Proficient in English and Spanish, eloquent speaker, and engaging team member

Computer Programming: Object-Oriented Programming in C++, Java, Python, and C#.

Problem Solving: Experience finding creative solutions from a logical-mathematical point of view

Additional Experience

Distinguished Sustainable Development Goals Ambassador

(Fall 2021 - Spring 2022)

KSU Division of Global Affairs

- **Leadership Award Recipient**
- Study the United Nations' Sustainable Development Goals
- Volunteer 20 hours in activities that match the Sustainable Agenda
- Develop an action plan to fit the goals in the ambassadors' community

University Brand Ambassador

(Fall 2021 - Present)

KSU Admissions

- Promote the on-campus life and events on social media
- Answer questions related to KSU experiences and services
- Share online the college life of a Computing Science student

Sahan Reddy

Email: sahancreddy@gmail.com
Phone: (404) 901-3005

LinkedIn: <https://www.linkedin.com/in/sahanreddy/>
GitLab: <https://gitlab.com/kfajdsl/>

Education

B.S. Computer Science, Kennesaw State University · June 2021 - Present
GPA: 3.6/4.0

Work Experience

Area-I (an Anduril Company) · Autonomy Intern Summer 2022

C++, Python, CMake, Docker, Nix, Boost.Test

- Optimized the performance of a model predictive control algorithm by 30%, allowing a longer horizon at the same speeds.
- Integrated our autonomy stack with the SCRIMAGE simulator, allowing a faster development iteration cycles than with our high-fidelity simulator.
- Created a moving object position and velocity prediction algorithm using kinematic models.
- Developed the automated tests and visualization tooling for all of the above.

Area-I specializes in developing autonomous fixed-wing aircraft.

Related Experience

KSU Electric Vehicle Team · Software Lead Aug 2021 - Present

C++, Python, ROS 2, PCL, OpenCV, Docker, FreeRTOS

- Helped implement our occupancy grid based reactive navigation algorithm (paper submitted to IEEE HONET 2022).
- Redesigned the ground segmentation phase of our LiDAR based perception pipeline, improving cone detection range by 50%.
- Led the development of the embedded software used for controlling all the low level systems on our go-kart.
- Dockerized our ROS 2 development and build environments, increasing reproducibility, simplifying CI, and reducing the barrier of entry for contributing to the project.

We build, design, and race autonomous vehicles. We won 1st place in both the 2022 and 2021 evGrandPrix.

StudentsForTomorrow · Co-Founder Apr 2020 - Jul 2021

TypeScript, Firebase (Functions, Auth, Firestore, Realtime DB), React, Twilio

- Designed and implemented the backend for our automatic text banking service.
- Wrote the cloud functions powering our college voter registration tool, which got hundreds of thousands of hits.

StudentsForTomorrow is a 501(c)(4) nonprofit with the goal of putting younger people into office. During the 2020 presidential election, we registered 65,000 newly eligible college students to vote.

Skills

Languages: C++, Python, Java, Bash, TypeScript, C, Elixir

Frameworks: ROS 2, FreeRTOS, Express.js, React.js, Phoenix

Tools: Git, Docker, Docker Compose, CMake, Nix/NixOS, PostgreSQL

4. Technical Description

Basic Customer Requirements

The customer wants the software to run on a PC where WinForms is used to display the graphical user interface and JSON, which holds the drink menu and customizations. The information that is in the customer data includes first & last name, phone number, and reward points. The application should be able to run on an average PC as long as it has enough storage space to hold the application itself as well as the JSON and CSV files that go along with it. Due to the application running on WinForms, the application will be coded in C#, and the operating system it will run on is Windows. WinForms can only run on Windows unless a Macintosh or Linux has a Windows virtual machine installed on them.

GUI Design

We will use the built-in GUI designer in Visual Studio to implement the buttons, text, and menus. We will also appropriately design the screens to where they are easy to read and look presentable to any user. This means adding color and design to engage the users. The UI design should allow the users to navigate back and forth between screens, enter any information needed, and show the sales report from an Excel document. To achieve this, we need to implement multiple buttons that can advance to a particular screen, like the back button going to a previous screen or clicking a button that advances to a screen where they can place an order. We will also need to have a button that is coded to where it will open an Excel document to show the sales report. We will also need to add textboxes so the user can input the customer's information.

Screens: Main Menu, Customer List, & Add Customer

The application will consist of six screens, where the Main Screen is the default screen. From there, a customer who does not wish to be a member (anonymous) can place an order. So, all the user has to do is hit Order Drink. If a customer is a member, the user hits the Customer List and then finds the customer by last name, first name, and phone number. The screen will always show the anonymous customer at the top of the list. After the customer is found, the user can hit Order Drink and take it from there. If a customer wishes to be a member of the coffee shop, the user should hit Add Customer and enter the valid credentials about the customer, such as full name and phone number. After the customer has been added, the screen should go to the Order Drink Screen. There should be a cancel button if a customer who wants to be a member changes their mind. The screen should switch to the Main screen if the button is pressed.

Screens: Order Drinks, Receipts, and Management

There should be two panes on the Order Drink Screen: drink creator and drinks added to the order plus the total cost. The drink creator should allow the user to customize the drink to the customer's liking. Once a drink has been added, the payment button to the Payment Screen should activate. There should also be a button to cancel the order and go to the main screen. On the Payment Screen, the customer should be able to pay with either reward points or a credit card if they have enough reward points or pay with a valid credit card (the card is 16 digits and not expired). If a credit card is used, the customer, considering that they are a rewards member, earns ten reward points for every \$1 spent, which will be stored. If the customer wishes to cancel, the user hits the cancel button and will go back to the main screen. After paying, the screen will switch to the Receipt screen, which will display the subtotal, tax, total, payment method,

remaining reward points, items purchased, and last four digits of the credit card if it was used. From there, the user can go back to the main menu. If the user goes to the management screen, they can pull up a CSV file that displays the sales report. The sales report should include what has been sold, how many of the items have been sold, the reward members, and the anonymous customers.

GitHub & Potential Risks

The group will use GitHub as means of sharing code with other members of the group. Since there is code to get us started on the project, we will fork the code and work from there. There are potential risks that could stop us. One of them is lack of communication which doesn't just mean among the team. Lack of customer communication will cause the program to fail the requirements that the customer is expecting. To prevent this, we must communicate anything we are unsure about or clarify a requirement. We all need to keep a flexible schedule to make the project and our lives run smoothly; otherwise, we will be rushing and mostly miss essential requirements that need to be implemented. Another risk could be that someone is unable to do or not doing their part. This causes the pace of the project to run slower. If someone is unable to do their part due to personal issues in their lives or they are sick, they need to communicate with the group so that the group can find a workaround. If someone decides they are not doing their part, the group can go to the customer and explain what is happening.

5. Data Management Plan

Customer List

Our application will keep track of each customer in the rewards program in a JSON file. Customers who give a full name and phone number and join the rewards plan are recorded with fields CustomerID (String GUID), FirstName (String), LastName (String), PhoneNumber (String), and RewardsPoints(int). Anonymous customers are represented by the first entry in the file for customers who do not wish to give their name and phone number. Rewards points are always rounded down to the nearest whole number and are updated after each purchase a customer makes.

CustomerID	String (GUID)
FirstName	String
LastName	String
PhoneNumber	String
RewardsPoints	Int

Sales Data

Our application will create and store data entries for each sale registered in the system for both rewards and anonymous customers. New entries are automatically added after each order is completed with CustomerID (String), Date (DateTime), Tax(decimal), Subtotal(decimal), Total (decimal), PaymentType (String (Credit or Rewards)), and Order (String (Name, comma separated customizations, and total before tax)).

CustomerID	String (GUID)
Date	DateTime
Tax	Decimal
Subtotal	Decimal
Total	Decimal
PaymentType	String
Order	String

Configuration Data

Configuration data is held in an AppSettings JSON file that holds information about the tax rate, rewards earning rate.

Sales Report

Managers in the system can generate a sales data report as a CSV file. On the manager screen, the user will be prompted to enter a valid password where they can then create a report of all data from the Customer List or Sales Data.

6. Test Plan

Manual Testing

The program will be tested manually. This will include making sure every button pressed goes to the appropriate screen. For example, we do not want the Order Drink button to cause the program to switch to the Management screen. We will also check that the Excel document opens when prompted from the Management screen, checking to make sure a customer that is a rewards member is found. The payment button should not activate until one order has been processed, so we also need to check that. We will also check to make sure that all the calculations are correct. This means checking that the reward points increase by 10 for every dollar the customer spends on their credit card and making sure that the total cost is added up correctly.

For the anonymous customer, we need to ensure that they are at the top of the customer list with the appropriate credentials, such as the first and last name is "Anonymous". We need to ensure that if we hit Order Drink right from the Main screen, it automatically declares the customer anonymous. If the customer pays with reward points, they should not receive reward points. These are the kind of things that we will be checking to ensure the program runs smoothly and meets the customer's requirements.

Black & White Box

We will primarily use white box testing because the testers will be two of us. Since we know the program's internal workings, it will effectively test to ensure it does what it's supposed to. We may even try out black box testing by getting some students not from the class to test out the program. This will help to check that the program is easy to follow while also checking to see that the program works correctly.