PROJECT PLAN

TEAM ESPRESSO DELIVERY

(DION GREEN, TREVOR GREEN, DOVE RAY, ADELE ROUSSEAU)

PROFESSOR ADKISSON

SWE 3133, GROUP BBB

OCTOBER 9, 2022

TABLE OF CONTENTS

High Level Project Scope 3

Brief Project Description 3

Complete Screen Relationship Process: 3

Visual Interface Details: 7

Schedule 8

Work Breakdown Structure: 8

GANTT Chart: 9

Deliverables: 9

Team Organization 10

Roles: 10

Resumes: 10

Technical Description 14

Data Management Plan 15

Test Plan 17

Overview: 17

Testing Roles: 17

Testing Duration: 17

HIGH LEVEL PROJECT SCOPE

Brief Project Description

ℹ A point of sales system for a coffee chain:

1) *Ability for staff to login*
    a) *Different levels of staff have access to certain abilities*
2) *Take orders*
    a) *Option to order and customize one or more drinks*
3) *A system for tracking Members*
    a) *Option to remain anonymous*
    b) *Create new Members*
    c) *Add or redeem reward points*
4) *Verify validity of the method of payment*
5) *Generate receipt for current order*
6) *Generate a sales report of all orders*

Complete Screen Relationship Process:

ℹ 1) Login Screen
    a) *Username and Password Fields*
        i) *Verified by checking a JSON file of all staff for a match of both fields*
        ii) *Staff are categorized into two roles:*
            (1) *Manager*
                (a) *Can access Management Screen*
            (2) *Employee*
2) Main Screen
    a) *Member Button*
        i) *Transfers to Member List Screen*

b) *Guest Button*

i) *Transfers to Order Screen*

ii) *Customer Information is recorded as “Anon Guest”*

(1) *Guests cannot receive or redeem any reward points*

c) *Management Button*

i) *Transfers to Management Screen*

3) Member List Screen

a) *Load a list of all current Members*

i) *Newtonsoft JSON file*

ii) *Contains:*

(1) *Last name*

(2) *First name*

(3) *Phone number*

(4) *Rewards Points Balance*

iii) *“Anon Guest” option at the top of the list*

b) *Each customer has an order drink button adjacent to their entry*

c) *New Member Button*

i) *Transfers to Create Member Screen*

d) *Home Button*

i) *Transfers to Main Screen*

4) Create Member Screen

a) *Cancel Button*

i) *Transfers to Main Screen*

b) *If not canceled:*

i) *Prompts user to enter first name, last name, and phone number*

ii) *New member is stored into the JSON file*

iii) *Transfers to Order Screen*

5) Order Screen

a) Menu loaded and displayed

i) Newtonsoft JSON file

ii) Each item can come in 3 different sizes which affects the price

iii) Each item can be customizable which affects the price

(1) Decorator pattern

iv) One customer can order multiple drinks

b) Cart

i) Shows all items currently in the order

ii) Shows subtotal, tax, and total for the order

iii) Checkout Button

(1) Transfers to Payment Screen

c) Cancel Button

i) Transfers to Main Screen

6) Payment Screen

a) Two methods of payment:

i) Credit Card

(1) Validate Credit Card number and expiration date

(a) CreditCardValidator

(2) If the customer is a Member, add rewards points to their account

(a) Every $1 spent = 10 rewards points (rate is configurable)

(b) Fractional rewards points are rounded down

ii) Rewards Points

(1) Validate total rewards points in Customer account can cover the complete cost of the order

(a) Remove rewards points from Customer account after being used

(b) Customer cannot earn rewards points for an order that was paid for using rewards points

b) When payment is completed:

i) Record order details to the database

(1) Newtonsoft JSON file

ii) Transfer to Receipt Screen

7) Receipt Screen

a) List the following elements:

i) Drinks ordered

(1) includes size and customization

ii) Subtotal, tax, and total

iii) Payment method

(1) Last 4 digits of credit card or

(2) Rewards points redeemed

(a) Show the Member’s remaining rewards points balance

b) Home Button

i) Transfers to Main Screen

8) Management Screen

a) Button to create a sales report

i) Generates a CSV file of all orders

(1) CSVHelper

ii) Opens file in Excel

b) Button to view and modify Customer List

c) Button to view and modify Staff roster

d) Settings Button

i) Transfers to Settings Screen

e) Home Button

i) Transfers to Main Screen

9) Settings Screen

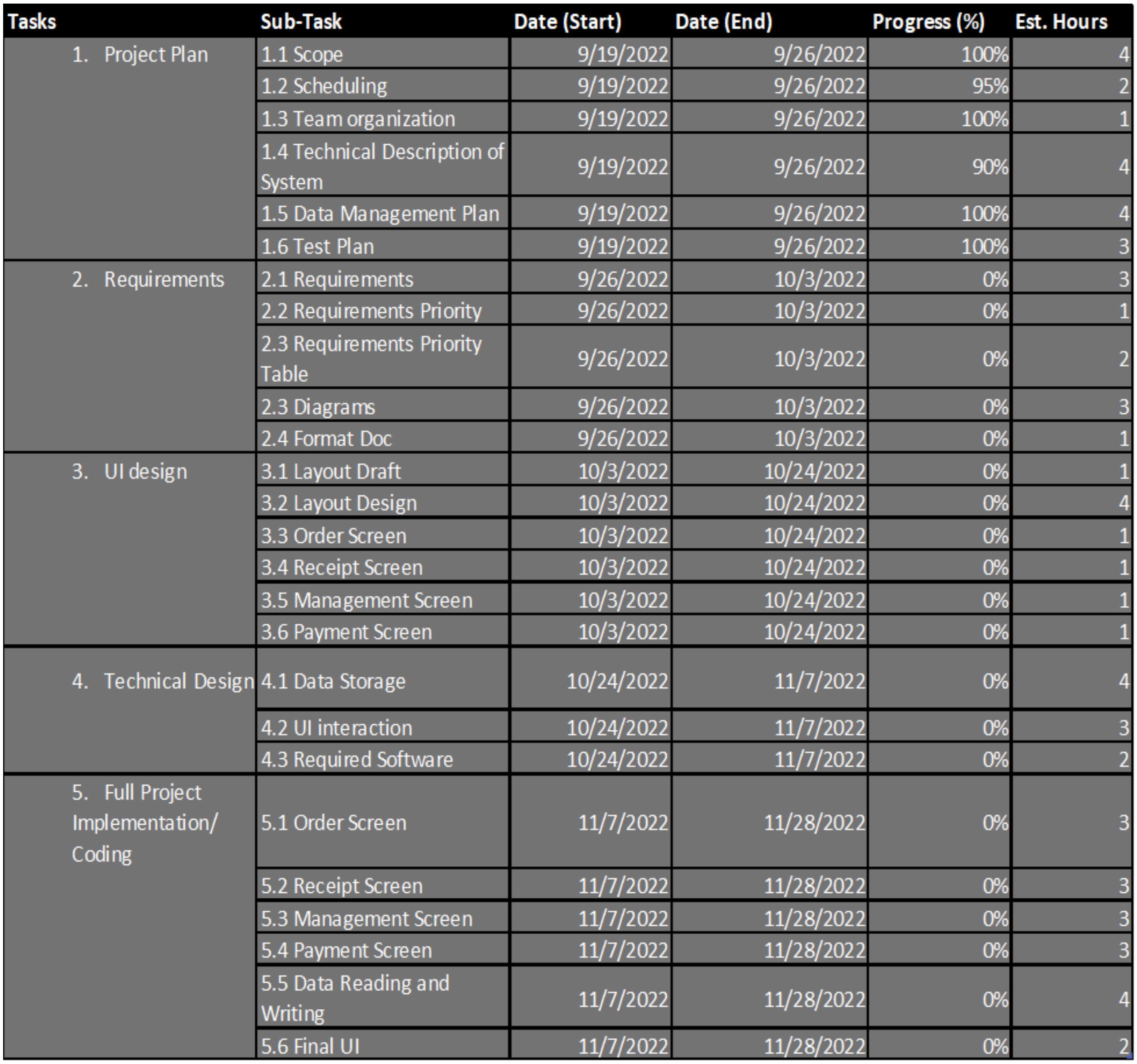
a) Ability to modify the menu and rewards points exchange rate

Visual Interface Details:

1) Created using C# WinForms
    a) Secondary programs like Clip Studio Paint and Figma will also be used for icons and other smaller details in the program
2) Formatted to US currency, rounded to two decimal places
3) Visual elements to be designed include:
    a) *Buttons*
    b) *Entry fields for:*
        i) *Login screen*
        ii) *New member information*
        iii) *Credit card verification*
    c) *Customer list format*
    d) *Menu list format*
    e) *Cart format*
    f) *Receipt format*

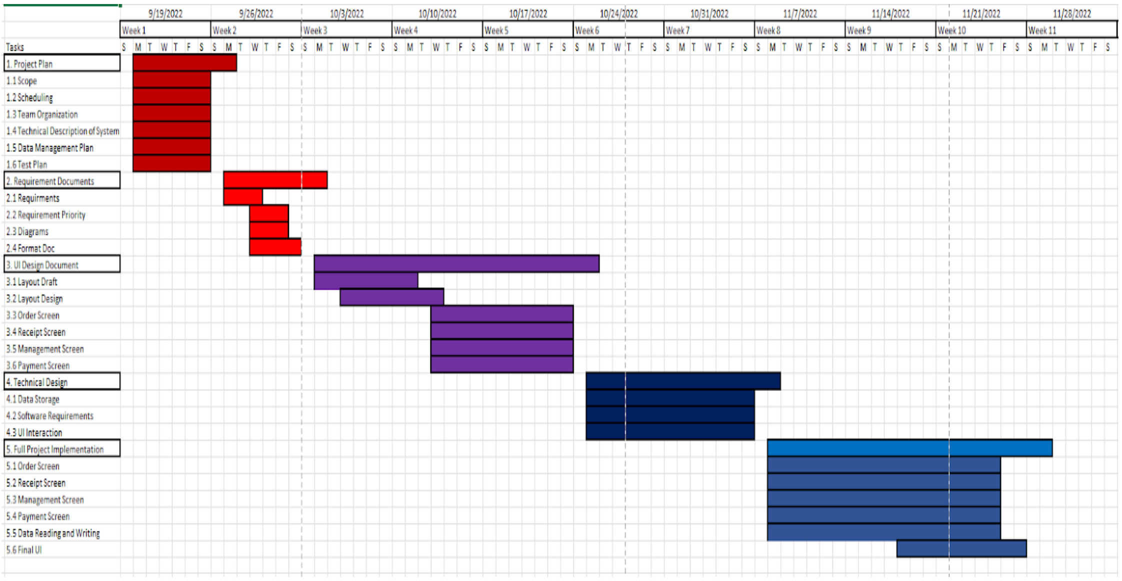
SCHEDULE

Work Breakdown Structure:



Tasks	Sub-Task	Date (Start)	Date (End)	Progress (%)	Est. Hours
1. Project Plan	1.1 Scope	9/19/2022	9/26/2022	100%	4
	1.2 Scheduling	9/19/2022	9/26/2022	95%	2
	1.3 Team organization	9/19/2022	9/26/2022	100%	1
	1.4 Technical Description of System	9/19/2022	9/26/2022	90%	4
	1.5 Data Management Plan	9/19/2022	9/26/2022	100%	4
	1.6 Test Plan	9/19/2022	9/26/2022	100%	3
2. Requirements	2.1 Requirements	9/26/2022	10/3/2022	0%	3
	2.2 Requirements Priority	9/26/2022	10/3/2022	0%	1
	2.3 Requirements Priority Table	9/26/2022	10/3/2022	0%	2
	2.3 Diagrams	9/26/2022	10/3/2022	0%	3
	2.4 Format Doc	9/26/2022	10/3/2022	0%	1
3. UI design	3.1 Layout Draft	10/3/2022	10/24/2022	0%	1
	3.2 Layout Design	10/3/2022	10/24/2022	0%	4
	3.3 Order Screen	10/3/2022	10/24/2022	0%	1
	3.4 Receipt Screen	10/3/2022	10/24/2022	0%	1
	3.5 Management Screen	10/3/2022	10/24/2022	0%	1
	3.6 Payment Screen	10/3/2022	10/24/2022	0%	1
4. Technical Design	4.1 Data Storage	10/24/2022	11/7/2022	0%	4
	4.2 UI interaction	10/24/2022	11/7/2022	0%	3
	4.3 Required Software	10/24/2022	11/7/2022	0%	2
5. Full Project Implementation/ Coding	5.1 Order Screen	11/7/2022	11/28/2022	0%	3
	5.2 Receipt Screen	11/7/2022	11/28/2022	0%	3
	5.3 Management Screen	11/7/2022	11/28/2022	0%	3
	5.4 Payment Screen	11/7/2022	11/28/2022	0%	3
	5.5 Data Reading and Writing	11/7/2022	11/28/2022	0%	4
	5.6 Final UI	11/7/2022	11/28/2022	0%	2

GANTT Chart:



Deliverables:

- *Project Plan*
- *Requirements*
- *UI design*
- *Technical Design*
- *Coffee Company Application*

TEAM ORGANIZATION

Roles:

Dion Green – Team Organizer, Lead WinForms Developer

Trevor Green – Lead Programmer, Unit Tester

Dove Ray – Visual Component Designer, Programmer

Adele Rousseau – Record Keeper, Programmer

Resumes:

Dion Green

dgree176@students.kennesaw.edu

Education

Bachelor of Science in Computer Game Design and Development

Technical Skills

- ➢ *HTML & CSS*
- ➢ *C#*
- ➢ *JavaScript*
- ➢ *Learning Java React*

Trevor Green

tgree140@students.kennesaw.edu

Education

ℹ *Seeking a Masters in Computer Game Design and Development*

Technical Skills

ℹ
- ➢ *C# (3 years)*
- ➢ *Unity Game Engine (3 years)*
- ➢ *Python (4 months)*

Previous Projects

ℹ
- ➢ *Arcade-style 3d shooter released on Itch.io in December 2021 under the name Liminal Shooter.*

Dove Ray

eray21@students.kennesaw.edu

Education

Bachelor of Science in Computer Game Design and Development

Computer Science Minor, Interactive Design Minor

Relevant Completed Courses:

- *Programming & Problem Solving I*
- *Programming & Problem Solving II*
- *Database Systems*
- *Data Structures*
- *Discrete Mathematics*
- *User Interface Design I*

Technical Skills

- *Object Oriented Programming (4 years)*
    - *C# (3 years)*
    - *Java (1 year)*
- *Unity Game Engine (2 years)*
- *SQL (6 months)*

Adele Rousseau

arousse3@students.kennesaw.edu

Education

Bachelor of Science in Computer Game Design and Development

Relevant Completed Courses:

- *Programming & Problem Solving I*
- *Programming & Problem Solving II*
- *Discrete Mathematics*

Technical Skills

- *C# (3 years)*
- *Unity Game Engine (2 years)*
- *Java (2 months)*

TECHNICAL DESCRIPTION

The system will be designed for computers and will not be designed for phones or any other type of device. It should be able to run on any computer because the program will not have excessive amounts of data that needs to be stored. In total, there will be four JSON files: menu list, customer list, staff list, and order history. We will use CSVHelper to convert the order history JSON file to CSV. This will allow the order statement to be viewed through Excel, which can help with readability for the company managers.

We will be using WinForms in Visual Studio to create the User Interface elements that will be used, like buttons and labels, within the system for the users to interact with to pick their drinks and input their data into. Any additional icons will be made using Clip Studio Paint. The code itself will also be done in C# in Visual Studio as well. GitHub will be used as a place to assemble the code.

The program will allow customers to either pick if they want to order anonymously. Anonymous customers are not allowed to earn or redeem rewards points. The price of said drink will be determined when the customer is done. This total is affected by customizations the customer might want when ordering. We plan to handle these customizations using Decorator patterns.

There are some potential problems that arise in the planning stages of the program, one of the biggest ones being customer specifications. This can be fixed by remaining in contact with the customer to ensure all requirements are met to the satisfaction of the customer.

There will be nine screens that can be accessed, the login screen, main screen, member list screen, create member screen, order screen, payment screen, receipt screen, management screen, and settings screen.

DATA MANAGEMENT PLAN

Upon launch, the program will display a log in screen. A legitimate log in is needed to proceed further.

If the program is logged in by a manager/location owner:

- *The user will have an option to view all employee accounts and add new ones.*
- *The user can view all registered customer accounts including account email, phone number, payment info, and reward points, along with the ability to manually add or remove points from specific customers and change customer account information*
- *The user can view a complete order history including quarterly profits and percentage of orders made by registered customers. Shows each individual order including full receipt and customer information (shows as anonymous if not registered).*
- *The user can enter a settings menu where the base price of items can be changed, items can be added or removed from the menu, and reward points per dollar spent can be changed.*
- *The manager user can access the same menus as the employee.*

If the program is logged into by an employee:

- *The user can input a new order, adding and removing menu items to said order. The system will differentiate customization items from standard items and only allow customization items to be added to valid items.*
- *The user can finalize an order, giving the option for the user to input customer credentials before prompting the user for the payment method to be used.*
    - *If the order is completed with a credit card, rewards points are added if the customer signed in with a customer account.*
    - *If the order is completed with rewards points, remove the correct amount of points from the customer account.*
- *If a customer wants to sign in but doesn't yet have an account, the user can input the customer's information allowing the program to automatically create a customer account to add to the system.*

The data this program will store consists of:

- *Menu items and price*
- *Manager Account info*
- *Employees Account info*
- *Customer Account info*
- *Order history*
- *Reward points settings*

Data will be stored in a small number of organized Json files.

Menu data consists of: (Read/Write)

- *Item name*
- *Item type (drink/customization/nondrink)*
- *Item base price*

Customer Account info consists of: (Read/Write)

- *Customer Name*
- *Phone number*
- *Reward points*
- *Payment Card number*

Employee/Manager Account info consists of: (Read/Write)

- *Username*
- *Password*

Order History consists of: (read only)

- *Items' names*
- *Total price*
- *Customer account (if applicable)*
- *Payment method used*

Credit Card info will be tested using the Luhn algorithm to ensure card validity.

Vigorous and flexible case management will be used with all data handling to ensure system stability.

TEST PLAN

Overview:

Testing the Coffee Company application will require several tests. These tests will target different requirements of the application (functional and non-functional). Test will look at the User Experience/ UI design, Reading and Writing data, Order Screen, Payment Screen, Receipt Screen, Management Screen. All requirements have sub-aspects that will be accounted when testing. Testing will overall focus on stress testing and ensuring that any error that may arise from base level use will be addressed and corrected. Before testing starts the project, the team will decide an appropriate strategy for testing. As the project develops these strategies are subject to change, to fit with the current flow of the project. Overall tests will be something geared towards white box testing due to the nature of this project but how we implement white box testing is subject to change.

Testing Roles:

During testing the group will be split into several roles to make the process of testing features more organized. Roles include Lead Tester, Testers, and Fixers. Lead Tester will keep track of all aspects of the application that needs to be tested and ensuring that there is movement in the process of testing. Testers will rotate the features they are testing, to ensure that all features are properly tested to mitigate the risk of faulty code being implemented in the next testing phase, when it could've been caught by a second eye first. Testers will also be responsible for writing down issues and their potential cause and provide a concise report for the Fixers. Fixers' main role is to deal with issues that arise from testing, while not interfering with current testing. Due to having a small team, several roles are subject to be assigned to a team member.

Testing Duration:

While the overall project follows a waterfall style model, during the coding phase the project will take on a more agile approach, where there are several integrations and testing of the application. This implies that as the project is moving forward aspects from the windows form and aspects from the coding will be merged, tested, analyzed, and fixed. A schedule that is flexible will be made for testing, since major tasks might be completed early or late. While testing is important, the application does have a very present time constraint. Therefore, functional requirements will be prioritized when any testing occurs to ensure no matter what the application meets the requirements of the client.