**project management plan**

**for the**

**Uber Eats Food Delivery Service**

**10/12/2023**



Team Names:

Team Member 1: Mahmoud Hussein 221523

Team Member 2: Maximos Ashraf 226650

Team Member 3: Mostapha Abdulaziz 227824

Team Member 4: Sameh Ayman 218767

**Uber Eats Food Delivery Service, 23CSCI08I**

Table of Contents

Section Page

[SECTION 1. OVERVIEW 1](#_Toc257388665)

[1.1 Project Summary 1](#_Toc257388666)

[1.1.1 Purpose, Scope, and Objectives 1](#_Toc257388667)

[1.1.2 Assumptions and Constraints 4](#_Toc257388668)

[1.1.3 Project Deliverables 5](#_Toc257388669)

[SECTION 4. PROJECT ORGANIZATION 6](#_Toc257388676)

[4.3 Project Roles and Responsibilities 6](#_Toc257388680)

[SECTION 5. MANAGEment PROCESS 7](#_Toc257388681)

[5.1.1 Estimation 7](#_Toc257388683)

[5.2.1 Work Activities 8](#_Toc257388688)

[5.2.2 Schedule Allocation 8](#_Toc257388689)

[5.2.3 Resource Allocation 9](#_Toc257388690)

[5.4 Risk Management 12](#_Toc257388699)

[SECTION 6. TECHNICAL PROCESS 13](#_Toc257388701)

[6.2 Methods, Tools and Techniques 13](#_Toc257388703)

# SECTION 1. OVERVIEW

## 1.1 Project Summary

The food delivery service project aims to create a digital platform that connects customers to restaurants through an intermediary delivery driver that is willing to make the trip to deliver the food wherever the customer is at. It is an efficient and seamless solution for ordering and food delivery that offers a convenient and user-friendly experience for customers by providing them access to an entire selection of restaurants, customize and manage orders to their preferences, set their location, match with a delivery driver, track deliveries in real-time, and make secure payments.

### 1.1.1 Purpose, Scope, and Objectives

Purpose:

The purpose of this project is to develop a platform that revolutionizes the way customers interact with the food industry by addressing issues such as limited delivery driver range, restaurants not having delivery option or simply a faster delivery time. Whatever the problem may be, this platform aims to simplify, connect and expand the network between customers and restaurants in their area by creating an unparalleled, convenient, and efficient experience for ordering and receiving meals. By prioritizing user-centric design, fostering valuable restaurant partnerships, implementing automation for efficient order management, and ensuring real-time delivery tracking, the project strives to simplify the entire food delivery process by making the process of finding, customizing, and ordering food as effortless as possible. The end-goal purpose of this business is to normalize and ease the process of ordering food to a customer’s door with a comprehensive and user-friendly platform, while supporting local businesses, local delivery drivers and driving an overall net positive innovation in the food delivery sector.

Scope:

The scope of the food delivery system is to provide a variety of services for the system’s main actors: customers, delivery drivers and restaurant owners.

For the customer, they should be able to access a user-friendly interface to easily conduct the registration process where they provide essential personal information such as name, email, location and secure payment details. The customer retains the capability the ability to change their profile preferences whenever they need to, keeping their information up-to-date, modifying contact details, and adding or switching payment methods.

When it’s all been said and done and registration is complete, the customer is redirected to the home page where they have access to a variety of restaurants, menu options and latest offers categorized by price, rating, type and distance. The customer should have customization options to allow customers to exclude specific food items or include extra features, tailoring their orders to their own personal preferences.

A shopping cart feature allows customers to make multiple orders from either the same restaurants or various other restaurants simultaneously by facilitating order reviews, adjustments and providing necessary details before placing the orders. It also grants the option to save favorite or frequently ordered items for quick and easy access next time.

After all the orders have been confirmed, the customer is then prompted with multiple payment options including credit/debit cards, PayPal, and other accepted methods that suit personal preferences. The order is sent to the restaurant and the customer then proceeds to the next step of getting connected with a driver and given the total delivery time.

After the meal is done being cooked and is picked up by the delivery driver, the customer has access to real-time tracking of the entire journey to monitor their orders' status from the moment they are accepted by the restaurant to the driver's arrival at their doorstep. Lastly, after the food is delivered and paid for, the customer has the ability to provide feedback by rating both the specific restaurant and the delivery driver offering an avenue for expressing satisfaction and improvement suggestions.

For the driver, a similar process to the customer is followed where they have access to a user-friendly interface and are able to easily conduct the registration process which includes providing essential personal information such as name, email, location and secure payment details. The driver retains the capability to change their profile preferences whenever they need to, keeping their information up-to-date, modifying contact details, and adding or switching payment methods.

When the account has been set up, the driver is matched with a customer to a specific restaurant based on the incoming order where they have the ability to view the distance, total delivery time, and total price and is then prompted to either accept or reject the delivery.

After accepting, the driver shall go to the restaurant, provide proper identifications, pick up the meal and drive to the customer’s location. Upon arrival at the set destination, the driver shall get paid according to the set payment preference of their account, confirm payment and be given the ability to receive new orders. Lastly, the driver has the ability to rate the customer and report any issues faced along the way.

For the restaurant owner, the process is once again the same as the both the driver and customer in that they need to register with their own personal information in order to create an account. Afterwards, the restaurant owner fills in necessary information about their restaurant such as the name, contact information, cuisine type, main menu, delivering times, and operating hours. Once the restaurant's account is set up, owners have the capability to manage and customize their profile preferences such as contact details, and upload new dishes. They can add or modify menu items, set prices, and provide discounts for their most loyal customers. Restaurants receive real-time notifications and details of incoming customer orders and the delivery driver that has come to pick it up. Once ready, the restaurant package meals and confirm with both the delivery driver and customer. The restaurant owner gets paid by credit card, and the driver picks up the meal to deliver to the customer.

Objectives:

1. Establish and maintain partnerships with a wide range of restaurants.
2. Achieve steady revenue growth with net profit of $100,000 in the first year of operation.
3. Increase net profit by 10% each year to attain a net profit of $500,000 five years forward.
4. Become the go-to food delivery platform in our target markets.
5. Capture and control a 30% of the food delivery market share after 10 years of operation.
6. Foster a strong customer loyalty and acquire 250,000 of active app users within the first six months after launch.
7. Innovate and stay ahead of industry trends to expand into all the continents in 4 years.
8. Establish a strong and recognizable brand identity that resonates with customers, creating a household name that can migrate into future businesses.

### 1.1.2 Assumptions and Constraints

Assumptions:

1. Customers have access to a reliable internet connection.
2. Customers must be familiar with basic smartphone or web app usage.
3. Customers must enter accurate delivery addresses and contact information.
4. Customers may have dietary restrictions or preferences that need to be accommodated.
5. Customers are responsible for ensuring the security of their account credentials.
6. The development team assumes access to the necessary hardware and software tools.
7. The team assumes access to an existing restaurant database or the ability to onboard restaurants.
8. Assumption that third-party payment gateways and services are reliable and well-documented.
9. Developers assume adherence to coding standards and best practices during implementation.

Constraints:

1. Customers must provide accurate and up-to-date personal information.
2. Customers must follow payment guidelines and avoid fraudulent activities.
3. Customers must have a unique username in order to create an account.
4. Customers must adhere to restaurant-specific policies and rules, such as minimum order amounts.
5. The development team must follow industry best practices for security and data protection.
6. The team is limited by the permissions granted by the licenses of third-party services.
7. The development process must adhere to agreed-upon project timelines and milestones.
8. The system must comply with local and international regulations and laws governing online food delivery services.

### 1.1.3 Project Deliverables

Analysis phase: Starting date: 1/10/2023

1. Feasibility study. Expected date: 1/11/2023
2. Market research report: Expected date: 4/11/2023
3. Business requirements document: Expected date: 7/11/2023
4. Stakeholder analysis: Expected date: 10/11/2023
5. Legal assessment: Expected date: 17/11/2023

· Design Phase: Starting date: 29/11/2023

1. Context diagram: Expected date: 3/12/2023
2. Activity diagram: Expected date: 8/12/2023
3. Sequence diagram: Expected date: 11/12/2023
4. Class diagram: Expected date: 12/12/2023
5. Wireframes design: Expected date: 14/12/2023
6. GUI design: Expected date: 17/12/2023
7. Database Schema Design: Expected date: 21/12/2023

· Implementation Phase: Starting date: 1/1/2023

1. Development of mobile app: Expected date: 10/1/2024
2. Development of website: Expected date: 12/1/2024
3. Database setup: Expected date: 16/1/2024
4. Payment integration: Expected date: 19/1/2024

· Testing Phase: Starting date: 25/1/2023

1. Unit testing: Expected date: 2/2/2024
2. Integration testing: Expected date: 5/2/2024
3. User acceptance testing: Expected date: 26/2/2024
4. Security testing: Expected date: 27/2/2024
5. Load testing: Expected date: 29/2/2024

# SECTION 4. PROJECT ORGANIZATION

## 4.3 Project Roles and Responsibilities

Analysis Phase Manager: Sameh

Responsibilities:

* Define the scope of the food delivery app project in terms of geographical coverage.
* Lead discussions with stakeholders to gather specific requirements.
* Lead the feasibility study of the project’s overall potential.
* Identify market competitors and unique selling points.
* Manage communication with restaurant partners to establish initial agreements.

Design Phase Manager: Maximos

Responsibilities:

* Create an architecture diagram that outlines the main system components.
* Lead the design team to create user-friendly wireframes and mockups.
* Define the database schema to efficiently store menu items, restaurant details, user profiles, and order data.
* Develop customer flow diagrams that map out the journey from customer registration to order completion.

Implementation Phase Manager: Mahmoud

Responsibilities:

* Lead the development team to write code for various app features.
* Manage the development of menu management and order processing systems.
* Integrate secure payment third-party services and implement transaction processing with multiple payment options.
* Lead the development of both mobile and web app versions.
* Implement features that enable real-time order tracking and notifications for customers and drivers.

Testing Phase Manager: Mostapha

Responsibilities:

* Develop a comprehensive test plan covering the entire app.
* Lead a testing team to execute unit testing.
* Lead integration testing to validate that different app modules work together seamlessly.
* Manage user acceptance testing efforts.
* Conduct load and performance testing.
* Lead security testing efforts to identify and address vulnerabilities.

# SECTION 5. Management PROCESS

### 5.1.1 Estimation

**Size Estimation:**

**Scenario**: Placing order (Sameh)

The customer decides to place an order, when the app is opened, he or she will be prompted to enter the login credentials to sign in into the app. Afterwards, they’ll have access to a wide variety of food items and restaurants to choose from as desired. The customer then chooses the food items of their choice and adds special instructions to specify their personal preferences. Once all is set, the customer will then proceed to checkout where he or she will be asked to enter their payment information. Lastly, the customer will set the location upon where the order shall be delivered afterwards.

Internally, the system will access its database to process the order, update the restaurant's sales records, and charge the customer's account. A message confirming the order will be sent from the app to the customer.

The order details will be recorded in the database. The restaurant will receive the order, notify the driver and the driver will deliver the food to the customer. Once the order is delivered, the transaction will be added to both the restaurant's sales records, the customer's order history and driver’s own commission.

**Function points size estimation:**

External Input types (EI): 8

1. Username
2. Password
3. Selected food items
4. Special food preferences
5. Full name on card
6. Card number
7. Expiration date
8. Delivery address

External Output Types (EO): 3

1. Order confirmation message to the user
2. Order request message to the driver
3. Order request message to the restaurant

External Inquiry types (EQ): 3

1. Retrieve order status
2. Retrieve driver details
3. Retrieve payment transaction details

Logical Internal File types (LIF): 5

1. Customer table
2. Restaurant table
3. Driver table
4. Order table
5. Transaction table

External interface file types (UI): 3

1. Customer's bank information
2. Restaurant’s bank information
3. Driver’s bank information

**Identify data types, record types and complexity:**

**EI:**

* **8 Datatypes**: customer username, customer password, selected food items, special food preferences, customer full name on card, customer card number, customer expiration date, customer delivery address
* **3 Record Types**: customer table, restaurant table, order table
* (High Complexity)

**EO:**

* **4 Datatype**: order confirmation message to the user, payment confirmation message, order request message to the driver, order request message to the restaurant
* **4 Record Types**: customer table, restaurant table, driver table, transaction table
* (Average Complexity)

**EQ:**

* **8 Datatypes**: customer username, customer password, selected food items, special food preferences, customer full name on card, customer card number, customer expiration date, customer delivery address
* **5 Record Types**: customer table, restaurant table, driver table, order table, transaction table
* (High Complexity)

**LIF:**

* **20 Datatypes**: customer name, customer mobile number, selected food items, special food preferences, customer full name on card, customer card number, customer expiration date, customer delivery address, driver status, driver ID, driver name, order ID, customer ID, restaurant ID, delivery status, order date, order time, payment details, transaction date, transaction ID
* **5 Record Types**: customer table, restaurant table, driver table, order table, transaction table
* (Average Complexity)

**UI:**

* **9 Datatypes**: customer full name on card, customer card number, customer expiration date, restaurant full name on card, restaurant card number, restaurant expiration date, driver full name on card, driver card number, driver expiration date
* **4 Record Types**: customer table, restaurant table, driver table, transaction table
* (Low Complexity)

**Size Estimation**: (8\*6) + (3\*5) + (3\*6) + (5\*10) + (3\*5) = 146FP

146FP \* 46= 6716 LOC = 6.72 KLOC

**Scenario:** Saving Meal Combinations (Sameh)

The customer has the ability to save meal combos of multiple restaurants either from previous food deliveries or simply creating it and saving it for a later time. He or she gets to select various items from different restaurant menus to build a unique meal combination custom-made to their preferences. After completing their meal creation, they give the combination a personalized name, description, fitting tags, dietary label, and get to choose to save it for future orders and add to favorites. Lastly, users of the app get to share their combinations with other users or family and friends. If combinations are public, other customers and users of the app can also review and rate saved meal combinations based on their own anecdotal experience.

Internally, the system will update the database to save the details of these unique combinations, reviews and ratings for saved meal combinations, any user activity that updates meal combinations. Lastly, a confirmation message will be sent to both the customer and anyone who saved their meal combo each time said customer adds, updates or deletes their meal combinations.

**Function points size estimation:**

External Input types (EI): 8

1. Menu items from various restaurants
2. Individual item preferences
3. Quantity of each item
4. Custom name for the meal combination
5. Personal reviews
6. Personal ratings
7. Option to set the combination as public or private
8. Add to favorites

External Output Types (EO): 2

1. Confirmation message to the user that the meal combo has been saved
2. Confirmation message to other users that the meal combo has been edited

External Inquiry types (EQ): 3

1. Retrieve saved meal combinations
2. Retrieve meal combos based on ratings
3. Retrieve meal combos with certain cuisines

Logical Internal File types (LIF): 3

1. Saved meal combinations
2. Customer table
3. Restaurant table

External interface file types (UI): 0

**Identify data types, record types and complexity:**

**EI**:

* **8 Data Types**: selected menu items, individual item preferences, quantity, custom names for meal combinations, personal reviews, personal ratings, option to set combinations as public or private, favorite list
* **3 Record Types**: saved meal combinations, customer table, restaurant table
* (High Complexity)

**EO**:

* **2 Data Types**: confirmation message to the customer, confirmation message to other users that the meal combo has been edited
* **1 Record Types**: customer table
* (Low Complexity)

**EQ**:

* **6 Data Types**: retrieved saved meal combinations, customer's favorite items, order status, cuisine type, meal combo rating, date created
* **3 Record Types**: saved meal combinations, customer table, restaurant table
* (Average Complexity)

**LIF**:

* **20 Data Types**: selected menu items, individual item preferences, quantity, custom names for meal combinations, food item ID, restaurant ID, personal reviews, personal ratings, option to set combinations as public or private, overall community rating, community reviews, favorite list, meal combo date created, meal combo time created, meal combo last modified, meal combo tags, share count, meal combo description, meal combo price, meal combo dietary label
* **3 Record Types**: saved meal combinations, customer table, restaurant table
* (Average Complexity)

**EIF**:

* **0 Data Types**:
* **0 Record Types**:
* (Low Complexity)

**Size Estimation**: (8\*6) + (2\*4) + (3\*4) + (3\*10) + (0\*5) = 98FP

98FP \* 46= 4508 LOC = 4.51 KLOC

**Scenario: Upload Restaurant Menu (Mostapha)**

The restaurant owner logs into the system by providing the essential credentials password, username, and account type. Then in the restaurant’s owner dashboard the restaurant owner can manage a various side of their restaurant’s info, then he navigates to restaurant information section where he can update and provide information about his restaurant, in this section the owner fills up the required details such as phone numbers of the restaurant and hotline, cuisine type, restaurant’s website and social media pages. The owner goes to the ‘menu’ section where he can upload the menu for his restaurant and can provide the items for different sections in the menu such as appetizers, main courses, and beverages according to his restaurant type. To set the menu items the process goes as follows the owner types the dish name, description about that dish and the price then click on ‘add item’ button he also can add notes on every item such as customizations. Also the owner has the option to upload an image for each item in the menu by either selecting this pic from their local device or providing a URL, once he’s done with entering the menu details he saves the changes by clicking on ‘publish’ to make the updates live on the system, Then a POP-UP message will appear on the screen; Finally he Exit the dashboard by clicking on the ‘logout hyperlink’ and then the updates he made will be available on the system for the customers on the food delivery system.

**Function Points Size Estimation: -**

External Input Types (EI): 13

1. Username
2. Password
3. Account Type
4. Phone Number
5. Restaurant’s Hotline
6. Cuisine Type
7. Restaurant’s Website
8. Restaurant’s social media
9. Dish Name
10. Dish Description
11. Dish Price
12. Image
13. URL

External Output Types (EO): 1

1. a POP-UP message confirming the changes in the menu.

External Inquiry Types (EQ): 4

1. Retrieve the account details.
2. Retrieve restaurant’s details.
3. Retrieve saved details about the menu.
4. Retrieve menu picture from either computer’s disk or through URL.

Logical Internal File Types (LIF): 5

1. User table
2. Restaurant’s table
3. Menu table
4. Dish table
5. Validation table

External Inquiry Types (UI): 0

**Datatypes, Record Types and Complexity Identification:**

**EI:**

* 13 Datatypes: Username, Password, Account Type, Phone Number, Restaurant’s Hotline, Cuisine Type, Restaurant’s Website, Restaurant’s social media, Dish Name, Dish Description, Dish Price, Image, URL.
* 4 Record Types: User table, Restaurant’s table, Menu table, Dish table.
* High Complexity

**EO**:

* 1 Datatype: The POP-UP message confirming the changes in the menu.
* 2 Record Types: Menu table, Dish table.
* Low Complexity

**EQ:**

* 13 Datatypes: Username, Password, Define the account Type, Phone Number, Restaurant’s Hotline, Cuisine Type, Restaurant’s Website, Restaurant’s social media, Dish Name, Dish Description, Dish Price, Image, URL.
* 4 Record Types: User table, Restaurant’s table, Menu table, Dish table.
* High Complexity

**LIF**:

* 14 Datatypes: Username, Password, Account Type, Phone Number, Restaurant’s Hotline, Cuisine Type, Restaurant’s Website, Restaurant’s social media, Dish Name, Dish Description, Dish Price, Image, URL, Information validation.

5 Record Types: User table, Restaurant’s table, Menu table, Dish table, Validation table.

* Low Complexity

**UI:**

* 0 Datatypes
* 0 Record Types
* Low Complexity

**Size Estimation**: (13\*6) + (1\*4) + (4\*6) + (5\*7) + (0\*5) = 141FP

141FP \* 46 = 6486 LOC = 6.486 KLOC

**Scenario: Receive orders. (Mostapha)**

The restaurant owner logs into the system by providing the essential credentials password, username, and account type. Then in the restaurant’s owner dashboard he opens tab ‘order handling ‘by ‘clicking on it’ to see the process of incoming orders. Or he will be directed to this section directly in case he clicked on the notification of an order, this notification can be either sound alert or pop-up message or real-time message on the dashboard. After clicking the notification he’s directed to order management section which displays the details of this order such as customer name, phone number, address, location, order instructions, order items and the price of the order, The owner then review the order and make sure it can be done. Next the coordination with customer and delivery man for order delivery; The owner has the access to a list of the available delivery men that can deliver the order he select one and the system will send a notification for this delivery man with order information client name , phone number, location, address, delivery instructions and the total price of the order; After the delivery man receives the message he clicks on it or opens the system after completing the logging in process as a delivery man and navigates to received orders and click on it and confirm it by clicking on ‘confirm button’ or refuse it, So if he confirmed the order a message is sent to the owner that the order is confirmed by the delivery man and the customer receives a message with the estimated delivery time and delivery man information. After, the customer receives the order the delivery man update the status of the order by ‘marking the check box’ beside the order in orders progression list, Then the owner receives a message ensuring that the order is delivered, and the customer receives a message to rate the order.

**Function Points Size Estimation: -**

External Input Types (EI): 6

1. Username

1. Password
2. Account Type
3. Delivery man Selection
4. Order Confirmation
5. Updating Order Status

External Output Types (EO): 8

1. A pop-up message or real-time message on the dashboard for the received orders.
2. Message title.
3. Message body.
4. Notification for the delivery man with order information.
5. A message sent to the owner that the order is confirmed by the delivery man.
6. A Message sent to the customer with the estimated delivery time and delivery man information.
7. A message sent to the owner ensuring that the order is delivered.
8. A Message sent to the customer receives to rate the order.

External Inquiry Types (EQ): 5

1. Extract account details of the owner.
2. Retrieve restaurant’s details.
3. Retrieve the orders on the dashboard.
4. Retrieve customer information from the order.
5. Extract the delivery men list from the database.

Logical Internal File Types (LIF): 6

1. User table
2. Restaurant’s table
3. Orders table
4. Deliveryman table
5. Customer table
6. Order\_status table

External Inquiry Types (UI): 1

1. Global Positioning System.

**Datatypes, Record Types and Complexity Identification:**

**EI:**

* 7 Datatypes Owner’s username, Password, Account Type, Delivery man Selection, Order Confirmation, Updating Order Status, deliveryman username.
* 3 Record Types: Orders table, Deliveryman table, Order\_status table.
* High Complexity

**EO:**

* 4 Datatype: pop-up message of received orders, Message title, Message body, Notification of order information.
* 1 Record Types: Orders table.
* Low Complexity

**EQ:**

* 17 Datatypes: Owner’s Username, Owners’ Password, Owner’s account Type, Customer’s Phone Number, Customer’s location, Customer’s address, Customer’s Name, Restaurant’s ID, Restaurant’s name, Order’s Id, Order’s Status, Deliveryman name, Deliveryman id, Deliveryman Username, Deliveryman Password, Deliveryman account type, Deliveryman number.
* 4 Record Types: Restaurant’s table, Orders table, Deliveryman table, Customer table, Order\_status table.
* High Complexity

**LIF:**

* 18 Datatypes: User ID, Owner’s Username, Owners’ Password, Owner’s account Type, Customer’s Phone Number, Customer’s location, Customer’s address, Customer’s Name, Restaurant’s ID, Restaurant’s name, Order’s Id, Order’s Status, Deliveryman name, Deliveryman id, Deliveryman Username, Deliveryman Password, Deliveryman account type, Deliveryman number.
* 5 Record Types: User Table, Restaurant’s table, Orders table, Deliveryman table, Customer table, Order\_status table.
* Low Complexity

**UI:**

* 1 Datatypes Customer Location Latitude, Longitude.

1 Record Types Google’s maps Database.

* Low Complexity

**Size Estimation**: (6\*6) + (8\*4) + (5\*6) + (6\*7) + (1\*5) = 145FP

145FP \* 46 = 6670 LOC = 6.67 KLOC

**Scenario: Pay online. (Maximos)**

When a user wants to pay for a food order online, he can navigate the app to the restaurant's page where he wants. All the food types in the restaurant that are available will be displayed for the user to choose from. Once the user adds his order he can click buy now. He will be forwarded to a page where they’ll have to sign in to confirm their identity by username, password, cardholder, full name, card number, expiration date, and payment method. The customer will then be asked to choose a delivery address. Then the system will access its database to update the seller’s and buyer’s accounts and to save the transaction in the database. A message confirming the transaction will be sent from the app to the users, Also the user will be able to track. his transaction will be recorded in the database and added to both of the users’ accounts.

. Function Points Size Estimation: -

- COUNT OF COMPONENTS

External Input types (EI): 8

1. Username

2. Password

3. Selected order

4. Full name on Card

5. Card number

6. Expiration Date

7. Payment method

8. Delivery address

External Output Types (EO): 2

* Message confirming that the transaction is confirmed
* Open the ability to track the order

External inquiry types (EQ): 3

1. Retrieve account information

2. Retrieve the user’s bank account information

3. Retrieve order information

Logical internal file types (LIF):4

1. Customer Table

2. Transaction Table

3. Restaurant Table

4. Order Table

External interface file types (EIF):1

1. Customer’s Bank database system

IDENTIFY DATATYPES, RECORD TYPES AND COMPLEXITY.

**EI:**

• 8 Datatypes: Username, password, full name on card, card number, expiration date, delivery address, payment method, order details.

• 3 Record Types: Customer Table, Order Table, Transaction table

High Complexity

**EO:**

• 2 Datatype: Confirmation Message of Transaction, Open the ability to track the order

• 2 Record Types: Customer Table, Transaction table

Low Complexity

**EQ:**

• 9 Datatypes: Username, Password, Cardholder Full-Name, Card number, Expiration date, payment method, Account Type, Transaction amount, order details

• 3 Record Types: Customer Table, Transaction Table, Order Table

Average Complexity

**LIF:**

• 10 Datatypes: Username, Password, Address, Full name, Card number, expiration date, order ID, order price, order information, restaurant information

• 4 Record Types: Customer Table, Transaction Table, Order Table, Restaurant Table.

Low Complexity

**UI:**

• 4 Datatypes: Name on Card, Card number, Expiration Date, transaction amount

• 2 Record types: Customer Table, Transaction Table

Low Complexity

**Size Estimation**: (8\*6) + (2\*4) + (3\*4) + (4\*7) + (1\*5) = 101FP

101FP \* 46 = 4646 LOC = 4.646 KLOC

**Scenario: Track order. (Maximos)**

The track order scenario offers customers a seamless and convenient way to monitor the progress of their food orders in real-time. They will be forwarded to a page where they’ll have to sign in to confirm their identity by username and password, and they can use the order ID in our mobile app or website to track their delivery. As the order moves through the various stages of processing, from the restaurant's kitchen to the delivery driver's location, the customer receives timely updates. They can see when the order is being prepared when it's out for delivery, and even the estimated time of arrival. This not only provides transparency and assurance to the customer but also helps them plan their meal and ensure that they are ready to receive their delicious food when it arrives at their doorstep. Our track order feature is designed to enhance the overall user experience, making food delivery as convenient and efficient as possible.

. Function Points Size Estimation: -

- COUNT OF COMPONENTS

External Input types (EI): 3

1. Username

2. Password

3. order ID

External Output Types (EO): 6

1. Order current location

2. Time of arrival

3. Driver’s full name

4. Driver's license

5. Driver ID

6. Driver's car type

External inquiry types (EQ): 2

1. Retrieve account information

2. Retrieve delivery details

Logical internal file types (LIF):4

1. Customer Table

2. Driver table

3. Duration-location table

4. Order Table

External interface file types (EIF):1

1. map database system

IDENTIFY DATATYPES, RECORD TYPES AND COMPLEXITY.

**EI:**

• 3 Datatypes: Username, password, order ID

• 1 Record Types: Customer Table

Low Complexity

**EO:**

• 6 Datatype: Drive’s full name, driver's license, driver ID, driver's car type, Order current location, Time of arrival

• 2 Record Types: Duration-location table, Driver table

Average Complexity

**EQ:**

• 10 Datatypes: Username, Password, order ID, Full name, order details, Driver full name, driver's license, driver ID, Order current location, Time of arrival

• 4 Record Types: Driver Table, Duration-location table, Order Table, Customer Table

High Complexity

**LIF:**

• 12 Datatypes: Username, Password, Address, Full name, order ID, order price, order information, Driver full name, driver's license, driver ID, Order current location, Time of arrival

• 4 Record Types: Customer Table, Driver Table, Duration-location table, Order Table

Low Complexity

**UI:**

• 2 Datatypes: Order current location, Time of arrival

• 1 Record types: Duration-location table

Low Complexity

**Size Estimation**: (3\*3) + (6\*5) + (2\*6) + (4\*7) + (1\*5) = 84FP

84FP \* 46 = 3864 LOC = 3.864 KLOC

**Scenario: View incoming Order. (Mahmoud)**

The View Incoming Order feature aids drivers to efficiently monitor the status and details of incoming orders. Upon a customer's order placement, drivers receive notifications containing order IDs, directing them to the "View Incoming Order" page. After logging in and entering the order ID, drivers access comprehensive information such as list of items, total price, and any special instructions. Drivers can update orders progress through the stages like confirmation, preparation, and delivery, they have a real-time tracking of their location on a map for the customer. This functionality allows drivers to make a delivery planning, to ensure they are well prepared for each order arrival. The drivers have the option to confirm or reject the incoming orders they receive to suit their needs, which helps contributing to a seamless and efficient delivery process for all parties, After that the driver gets the order location, id and details to pick them up.

. Function Points Size Estimation: -

- COUNT OF COMPONENTS

External Input types (EI): 3

1. Username

2. Password

3. Order Confirmation or Rejection

External Output Types (EO): 4

1.Order Details Display

2.Order Location

3.Customer Phone

4. Order ID

External inquiry types (EQ): 2

1. Order Status Inquiry

2. Location Inquiry

Logical internal file types (LIF):4

1.Customer Table

2.Driver table

3.Order Table

4.Order Driver

External interface file types (EIF):1

1.Map database system

IDENTIFY DATATYPES, RECORD TYPES AND COMPLEXITY.

**EI:**

• 3 Datatypes: Username, password, order status

• 2 Record Types: Driver Table, Order Driver Table

Low Complexity

**EO:**

• 5 Datatype: Order Price, Order Details, Order Details, Order Status, Customer Phone

• 3 Record Types: Order table, Driver table, Customer table

Low Complexity

**EQ:**

• 7 Datatypes: Username, Password, order ID, order details, Driver full name, driver ID, Order location

• 3 Record Types: Driver Table, Order Driver table, Order Table

High Complexity

**LIF:**

• 10 Datatypes: Username, Password, Order ID, Order price, Order details, driver's license, driver ID, Order location, Customer ID, Customer Phone

• 4 Record Types: Driver Table, Order Driver table, Order Table, Customer Table

Low Complexity

**UI:**

• 1 Datatypes: Order location

• 1 Record types: Order table

Low Complexity

**Size Estimation**: (3\*3) + (4\*4) + (2\*6) + (4\*7) + (1\*5) = 70FP

70FP \* 46 = 3220 LOC = 3.22KLOC

**Scenario: Receive pickup destination and delivery. (Mahmoud)**

The Receive Pickup, Destination, and Delivery functionality is meant to help the drivers in efficiently manage and fulfill pickup and delivery requests. Upon receiving a real-time notification through the mobile app, drivers can either accept or decline the pickup request. The driver is provided essential details such as the pickup location and customer instructions provided. Upon the driver accepting the system offers detailed information about the pickup location and destination, along with real-time navigation assistance. Drivers confirm the pickup upon arrival and proceed to the destination, where the app displays delivery details and additional instructions. Real-time navigation ensures a smooth journey, and drivers confirm the delivery upon reaching the destination.

. Function Points Size Estimation: -

- COUNT OF COMPONENTS

External Input types (EI): 4

1. Username

2. Password

3.Order ID

4.Order Status

External Output Types (EO): 6

1.Order Details

2.Order Location

3.Customer Phone

4.Order ID

5.Order Destination

6.Customer Address

External inquiry types (EQ): 4

1. Order Status Inquiry

2. Location Tracking Inquiry

3.Arrival Time

4.Customer Data

Logical internal file types (LIF): 4

1.Customer Table

2.Driver table

3.Order Table

4.Order Driver Location

External interface file types (EIF): 0

IDENTIFY DATATYPES, RECORD TYPES AND COMPLEXITY.

**EI:**

• 4 Datatypes: Username, password, order ID, order status

• 2 Record Types: Driver Table, Order Table

Low Complexity

**EO:**

• 7 Datatype: Order Price, Order Details, Order Status, Customer ID, Customer Phone, Customer Adress, Order Location

• 4 Record Types: Order table, Driver table, Customer table, Order Driver Location table

Average Complexity

**EQ:**

• 12 Datatypes: Username, Password, Order ID, Order details, Driver full name, Driver ID, Order location, Time of arrival, Customer full name, Customer ID, Customer Address, Customer Phone

• 4 Record Types: Driver Table, Order Driver Location table, Order Table, Customer Table

High Complexity

**LIF:**

• 12 Datatypes: Username, Password, Order ID, Order details, Driver full name, Driver ID, Order location, Time of arrival, Customer full name, Customer ID, Customer Address, Customer Phone

• 4 Record Types: Driver Table, Order Driver Location table, Order Table, Customer Table

Low Complexity

**UI:**

• 4 Datatypes: Order details, Order status, Customer Phone, Customer Address

• 2 Record types: Order table, Customer table

Low Complexity

**Size Estimation**: (4\*3) + (6\*5) + (4\*6) + (4\*7) + (0\*5) = 94FP

94FP \* 46 = 4324 LOC = 4.32 KLOC

**Effort estimation:**

**COCOMO81:**

The system is of a complex nature, given its size and scope in connecting various actors such as customers, restaurants, and delivery drivers. This complex network of users, restaurants, and delivery drivers, requires a higher level of coordination and interaction as well as more effort put into the establishment and maintenance of partnerships with the wide range of restaurant partners. This along with the integration of real-time tracking and dynamic order management, indicates a level of complexity that is typically associated with embedded system projects. Lastly, the system faces multiple constraints from geographic limitations and internet connectivity to the availability of both the restaurants and drivers to ensure the overall functioning and success of the platform. However, because the total system size of 40.34 KLOC falls within the higher end of the organic system range, it can be assumed that this is an *organic system*.

**Total size** = 40434 = 40.43 KLOC

**Effort (PM)** = 2.4(*40.43*)1.05 = 116.74 PM

**COCOMO II:**

**COCOMO II Exponent driver ratings**

|  |  |  |
| --- | --- | --- |
| Driver | Rating | Comment |
| PREC | Nominal  (3.72) | The project is somewhat similar to previous ones such as Uber Eats and Talabat but introduces a few unique features, as such, while there is some precedent, the uniqueness of certain aspects increases the overall rating and complexity of the system. |
| FLEX | High  (2.03) | The system architecture that involves a lot of working parts from the restaurant to the delivery drivers and geographic limitations, the constraints make the system rigid in meeting requirements. |
| RESL | Low  (5.65) | The project has well-defined and unchanging requirements, minimizing the overall ambiguity and uncertainty. That comes with developing it. |
| TEAM | High  (2.19) | The team members are close-knitted and work in the same office. Although a large team, working on previous projects together helps in forming the right connections among them for the new assignments. |
| PMAT | Low  (6.24) | The project follows established processes, and there is a high level of organization and structure in project management. |

**COCOMO II Post architecture effort multipliers:**

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Code | Rating | Comments |
| Product | RELY | High  (1.10) | High software reliability is crucial to ensure that the ordering process, payment transactions, and real-time tracking are consistently accurate and dependable as to not disrupt user experience. |
| DATA | V. High  (1.28) | A huge database is needed in order to manage the huge influx of data coming from the restaurant, customer and driver. |
| DOCU | Nominal  (1.00) | Due to the nature of the system, aligning to a clear and well-structured documentation does not pose an exceptional challenge. |
| CPLX | V. High  (1.34) | System complexity is high due to the involvement of multiple stakeholders, intricate business rules, diverse menus, restaurant owners, real-time tracking, delivery drivers, customers and more. |
| REUSE | High  (1.07) | Due to the creation of various components and functionalities for the system, a lot of it can be efficiently implemented across different modules and components. |
| Computer | TIME | Extra High  (1.63) | An extra high execution time constraint is warranted due to the huge scope of the project that involves various actors and complex functionalities across the board. |
| STOR | Nominal  (1.00) | Due to the clearly defined constraints in the database, the system should efficiently manage data storage requirements without facing significant challenges in meeting storage constraints. |
| PVOL | Low  (0.87) | Most of the volatility will be post creation, but the system is of a predictable nature and thus the development environment is to, resulting in a more stable system. |
| Personnel | ACAP | Nominal  (1.00) | It is expected that the analysts chosen have the necessary experience that align with the standard requirements for gathering and analyzing system specifications. |
| AEXP | High  (0.88) | High application experience is essential as the software deals with complex business processes related to food ordering and delivery and better user experience. |
| PCAP | High  (0.88) | A team with high programmer capabilities is critical to the effective implementation of the intricate functionalities of the system. |
| PEXP | Low  (1.09) | It is expected that the development team may need to familiarize themselves with certain aspects of the unfamiliar environment needed in the system creation. |
| LEXP | Nominal  (1.00) | It is expected that the team meets the standard requirements and experience in the needed programming language for the development of the system. |
| PCON | High  (0.90) | Due to the size of the system, a high personnel continuity is crucial for maintaining uniformity during development. |
| Project | TOOL | High  (0.90) | The development team needs a high degree of understanding of the required software development tools in order to improve overall efficiency of the development environment. |
| SITE | V. High  (0.86) | A very high degree of seamless collaboration skills is needed to maintain consistent level of understanding across various locations. |
| SCED | High  (1.00) | Due to the size of the overall project, it is of the utmost importance that the deadlines are met in a timely manner as to maintain a competitive edge in the market. |

**General Equation**: Effort (PM) = A (Size)sf \* (Product of exponent multipliers)

A = 2.94

B = 0.91

Sum of exponent ratings = 19.83

Sf = 0.91 + 0.01 \* 19.83 = 1.1083

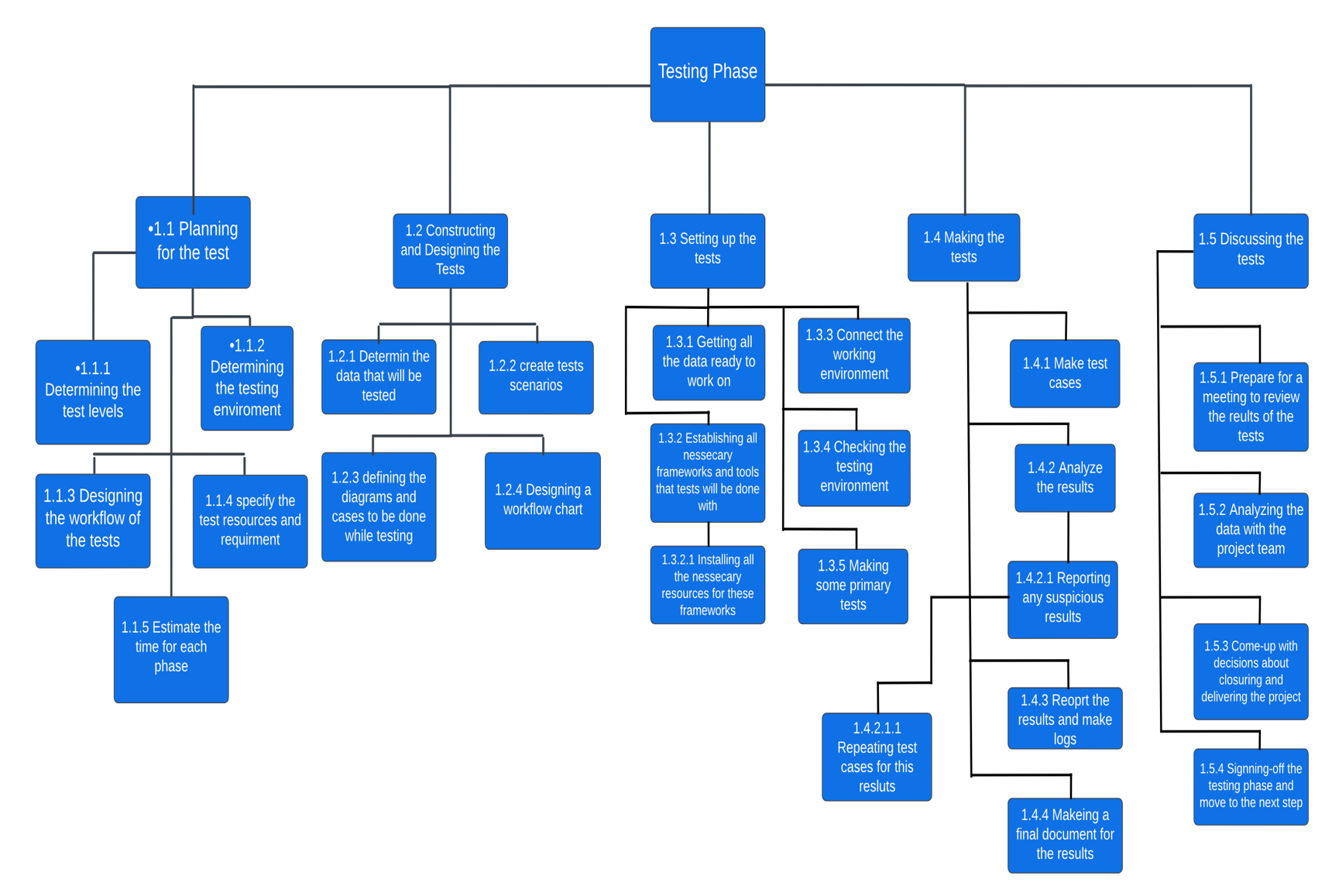
Product of exponent multipliers = 1.54435146

**Effort (PM)** = 2.94 (40.43)1.1083 \* (1.54435146) = 274.0332PM

### 5.2.1 Work Activities

**Analysis Phase WBS (Sameh)**

**Testing Phase WBS (Mostapha)**



**Design Phase WBS (Maximos)**

**Implementation Phase WBS (Mahmoud)**

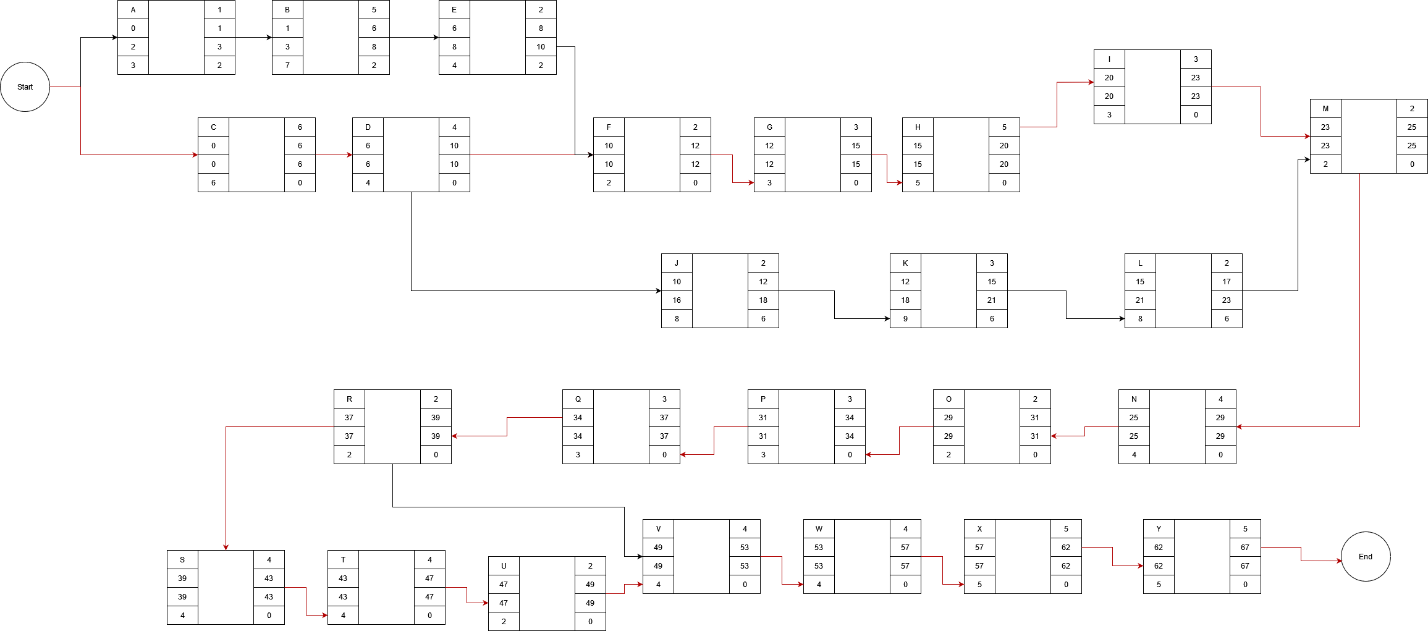
### 5.2.2 Schedule Allocation

**Precedence Table: (Group)**

|  |  |  |  |
| --- | --- | --- | --- |
| Activity | Description | Predecessor | Duration (Weeks) |
| A1 | Identify Key Stakeholders | -- | 1 |
| B1 | Document Stakeholder Expectations | A1 | 5 |
| C1 | Conduct Surveys And Questionnaires | -- | 6 |
| D1 | Gather User Requirements | C1 | 4 |
| E1 | Gather Business Requirements | B1 | 2 |
| F1 | Conduct Document Review | D1, E1 | 2 |
| G1 | Assess Technology Requirements | F1 | 3 |
| H1 | Evaluate Scalability | G1 | 5 |
| I1 | Conduct Operational Feasibility | H1 | 3 |
| J1 | Discuss Economic Feasibility | D1 | 2 |
| K1 | Discuss Legal Feasibility | J1 | 3 |
| L1 | Conduct Time Feasibility | K1 | 2 |
| M1 | Specify Deliverables | I1, L1 | 2 |
| N1 | State Scope Boundaries | M1 | 4 |
| O1 | Verify Scope | N1 | 2 |
| P1 | Specify Target Regions | O1 | 3 |
| Q1 | Identify Delivery Zones | P1 | 3 |
| R1 | Study Competitors | Q1 | 2 |
| S1 | Discuss Unique Selling Points (Usps) | R1 | 4 |
| T1 | Gather Relevant Market Data | S1 | 4 |
| U1 | Analyze Market Trends | T1 | 2 |
| V1 | Summarize Key Findings | U1, R1 | 4 |
| W1 | Sign Memorandums Of Understanding (Mous) | V1 | 4 |
| X1 | Conduct Agreement Negotiations | W1 | 5 |
| Y1 | Create Service Level Agreements (Slas) | X1 | 5 |
| A2 | Design a legal system | --- | 3 |
| B2 | Design ERD | --- | 4 |
| C2 | Design a financial regularity system | A2 | 3 |
| D2 | Design schema | B2 | 2 |
| E2 | Design sequence diagram | B2 | 1 |
| F2 | Design itineraries maps | C2 | 2 |
| G2 | Define tables | D2 | 2 |
| H2 | Define relationships | G2 | 1 |
| I2 | Design class diagram | H2 | 3 |
| J2 | Create advertising pages | I2, F2 | 5 |
| K2 | Design wireframes | J2 | 2 |
| L2 | Design GUI | E2, K2 | 3 |
| A3 | Hire new members |  | 1 |
| B3 | Train the current team | A3 | 3 |
| C3 | Evaluate the entire team | B3 | 1 |
| D3 | Determine the device specs needed |  | 2.5 |
| E3 | Buy enough devices | D3 | 0.5 |
| F3 | Test the devices | E3 | 1 |
| G3 | Evaluate the methodology delivery time |  | 2 |
| H3 | Request feedback from the stakeholders | G3 | 1 |
| I3 | Re-adjust the methodology | H3 | 1 |
| J3 | Divide the roles among the team | C3, F3, I3 | 2 |
| K3 | Divide the work phase into tasks | J3 | 1 |
| L3 | Divide tasks among the roles | K3 | 3 |
| M3 | Start the back-end implementation cycle | L3 | 24 |
| N3 | Start the Front-end Implementation | L3 | 24 |
| O3 | Evaluate the team's progress | N3, M3 | 4 |
| P3 | Re-adjust the plans if needed | O3 | 2 |
| A4 | Determine test environment | --- | 3 |
| B4 | Design the workflow | A4 | 2 |
| C4 | Specify the resources | B4 | 1 |
| D4 | Determine the data to test | C4 | 2 |
| E4 | Create testing scenarios | D4 | 1 |
| F4 | Define the diagrams and test cases | E4 | 1 |
| G4 | Design a workflow chart | F4 | 2 |
| H4 | Set up the tests | G4 | 6 |
| I4 | Connecting the testing environment | O4 | 1 |
| J4 | Make primary tests | I4 | 1 |
| K4 | Making the tests | J4 | 8 |
| L4 | Make logs and reports | M4,J4 | 3 |
| M4 | Analyze the results | K4, H4 | 1 |
| N4 | Make Final documentation | L4, M4 | 3 |
| O4 | Install necessary apps | C4 | 2 |

**CPM Networks:**

**Analysis Phase CPM Network (Sameh) Starting A1:**

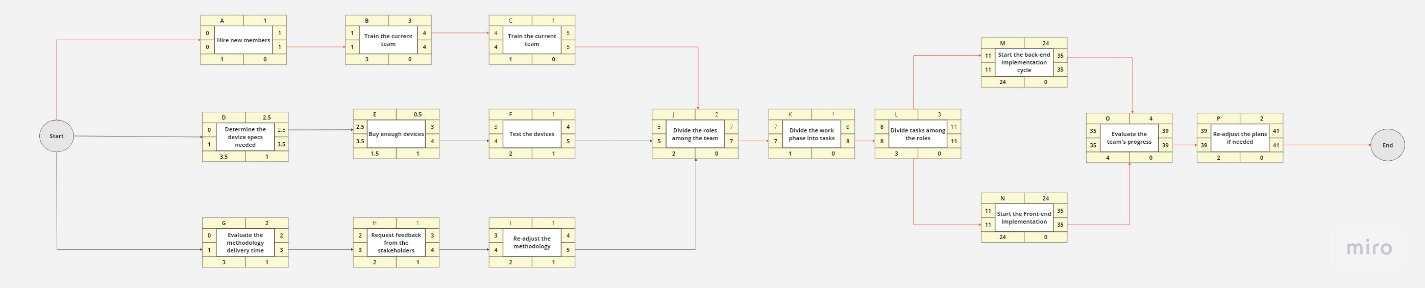
****

**Design Phase CPM Network (Maximos) Starting A2:**

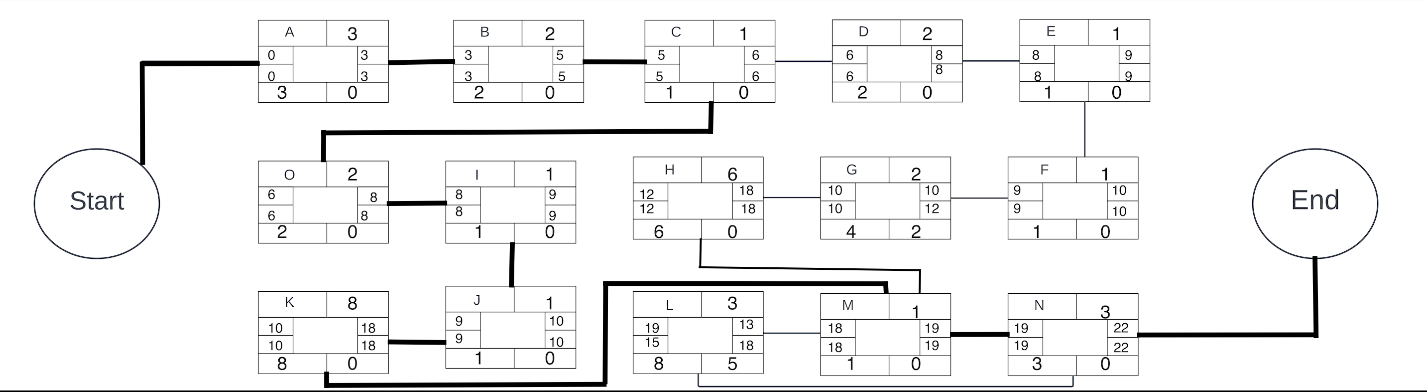
A screenshot of a computer game

Description automatically generated

**Implementation Phase CPM Network (Mahmoud) Starting A3:**

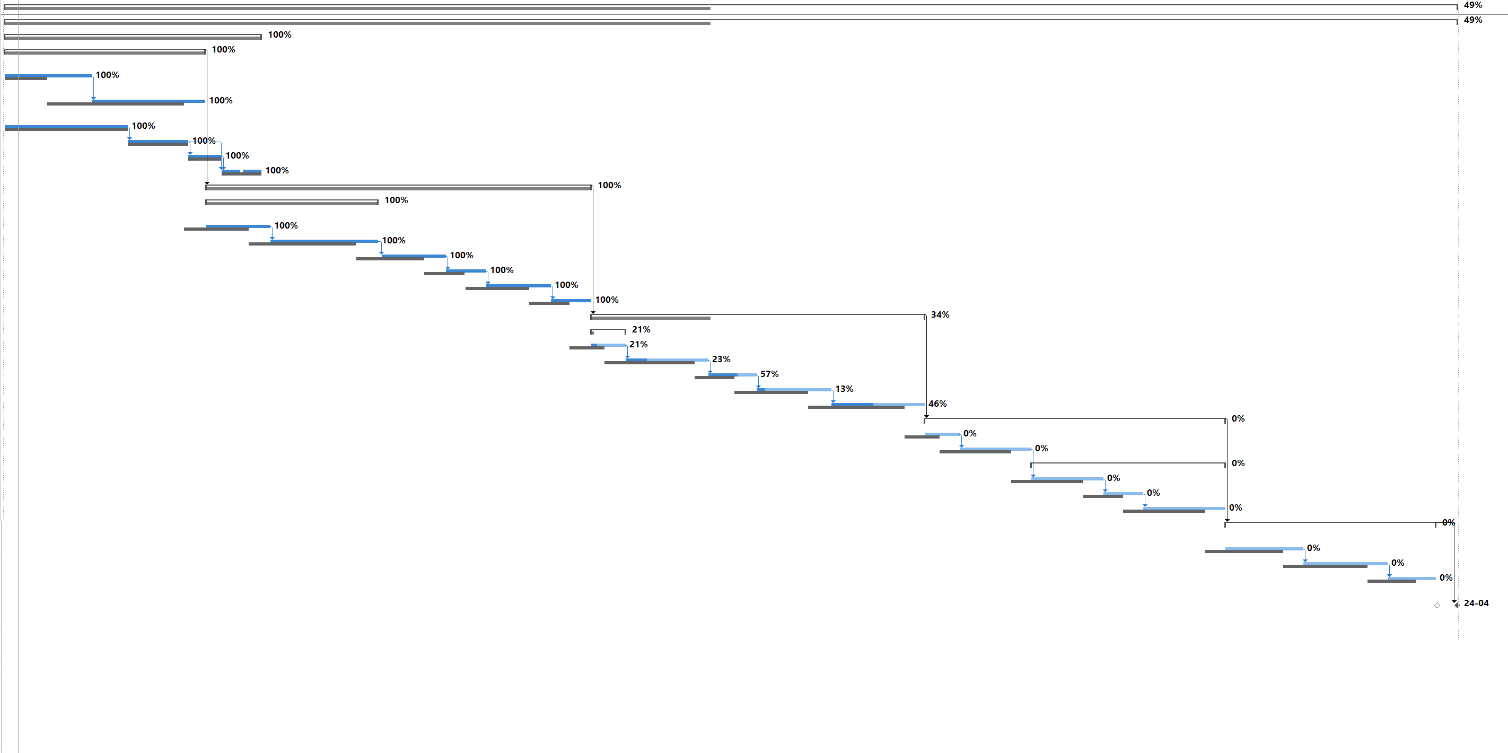
****

**Testing Phase CPM Network (Mostapha) Starting A4:**

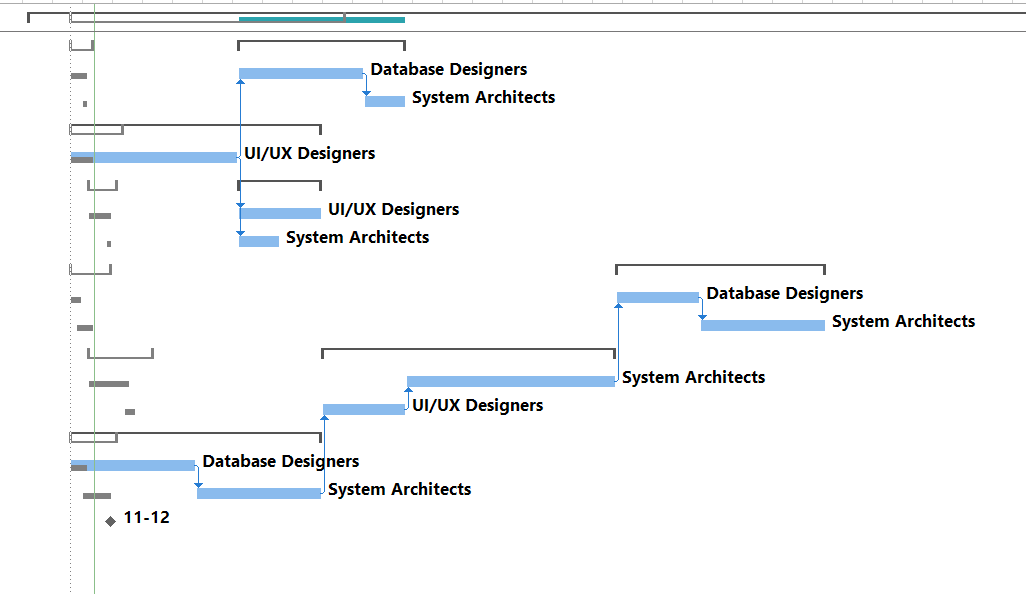
****

**Gantt Charts:**

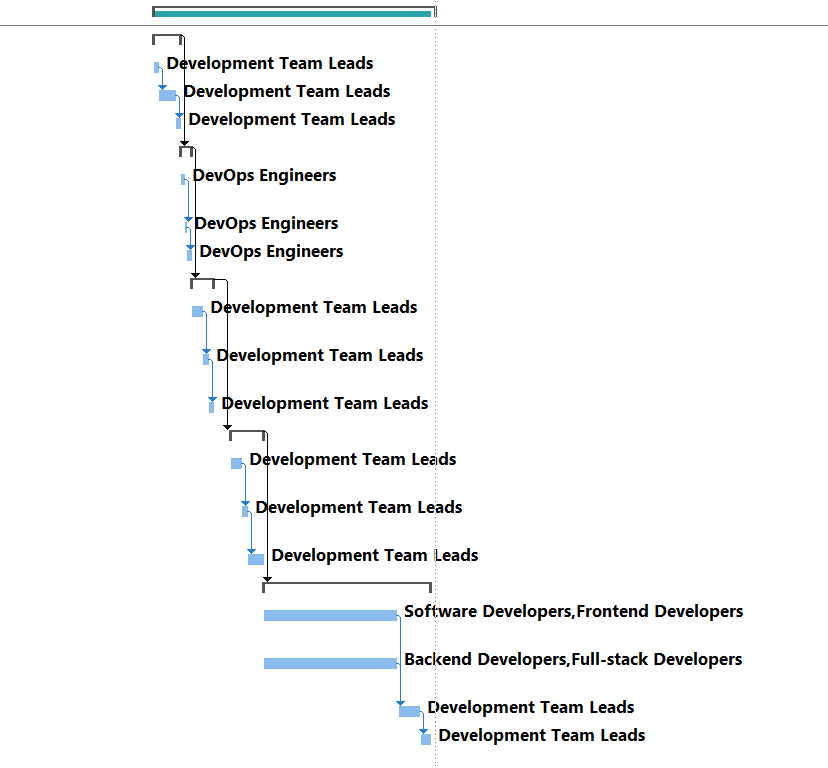
**Analysis Phase Gantt Chart (Sameh)**



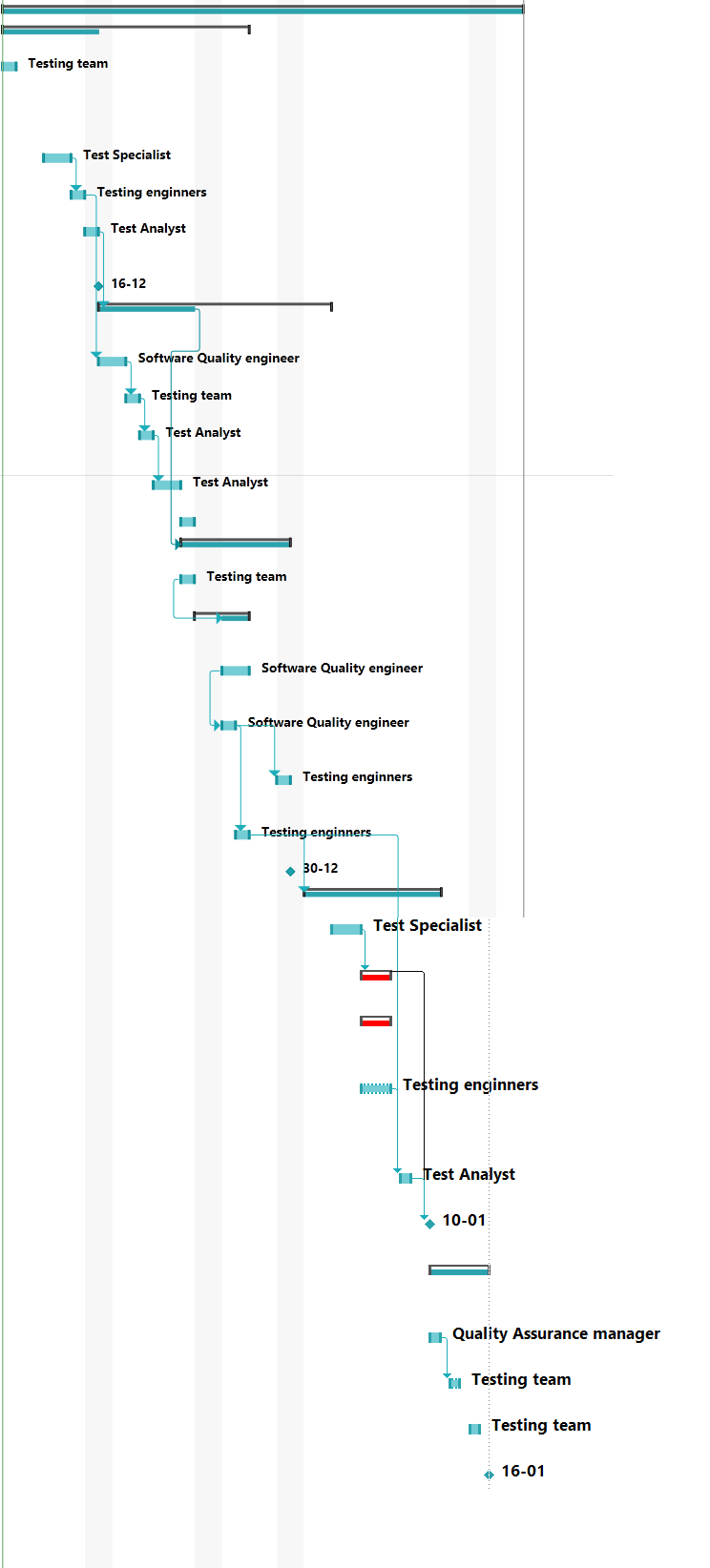
**Design Phase Gantt Chart (Maximos)**



**Implementation Phase Gantt Chart (Mahmoud)**

****

**Testing Phase Gantt Chart (Mostapha)**



### 5.2.3 Resource Allocation

### 1. Analysis Phase:

**Human Resources:**

* Business Analysts
* Systems Analysts
* Project Managers
* Stakeholders
* Customers

**Materials:**

* Project documentation templates
* Business requirement documents
* Stakeholder feedback forms
* Customer feedback forms

**Tools:**

* Collaboration tools such as Zoom, Miro and Discord
* Requirements gathering and analysis software such as ProjectManager

**Equipment:**

* 4 16GB RAM, quad-core processors HP laptops.

**Special Skills/Experience:**

* Strong communication skills
* 13+ years of experience in domain relevant to the project

**Total Cost:**

* Salaries and benefits for analysts and project managers: About $350,000
* Licensing costs for specialized tools: About $30,000
* Laptop costs. About $20,000

### 2. Design Phase:

**Human Resources:**

* System Architects
* UI/UX Designers
* Database Designers
* Technical Leads

**Materials:**

* Design specifications
* Relevant templates
* Design review documents

**Tools:**

* Design and modeling tools such as Adobe XD, Canva and MS Visio

**Equipment:**

* 12 Wacom Cintiq Pro 24 drawing tablets
* 8 High-end, core i7 computers with a windows display

**Special Skills/Experience:**

* 20+ years of experience in design tools and methodologies
* Knowledge of system architecture and database design
* Knowledge of UI/UX design on Figma

**Total Cost:**

* Salaries and benefits for design and technical teams: About $600,000
* Licensing costs for design tools: About $20,000
* Costs associated with design reviews and prototyping materials: About $20,000
* Computers, displays and Wacom tablets costs. About $100,000

### 3. Implementation Phase:

**Human Resources:**

* Software Developers
* Frontend Developers.
* Backend Developers.
* Full-stack Developers.
* DevOps Engineers
* Development Team Leads

**Materials:**

* Project documentation
* Prototypes
* Third-party components

**Tools:**

* VS Code IDE
* Github
* Microsoft SQL Server Manager
* AWS

**Equipment:**

* 25 16GB RAM, quad-core processors Alienware laptops.

**Special Skills/Experience:**

* 10+ years of experience in programming languages and frameworks require for the project
* Experience working on a project in a similar domain

**Total Cost:**

* Salaries and benefits for development teams: About $800,000
* Licensing costs for development tools and frameworks: About $30,000
* Laptop costs. About $120,000

### 4. Testing Phase:

**Human Resources:**

* Testing engineers
* Test Analyst
* Test Specialist
* Quality Assurance manager
* Testing team
* Software Quality engineer

**Materials:**

* Test cases
* Test data
* Defect tracking reports
* Testing reports

**Tools:**

* Automated testing tools such as Selenium
* Test management tools such as TestRail

**Equipment:**

* 5 16GB RAM, quad-core processors Alienware laptops.

**Special Skills/Experience:**

* 15+ years in testing and debugging skills
* Knowledge of Selenium.

**Total Cost:**

* Salaries and benefits for testing teams: About $100,000
* Licensing costs for testing tools: About $10,000
* Laptop costs. About $30,000

## 5.4 Risk Management

* **Risk Identification:**

**Generic:**

1. **Unexpected Increase in Project Size (Sameh)**

There is a risk that during that the overall size of the project is subject to increases beyond initial estimates which could lead to heightened chances of errors, communication difficulties, extended development times, resource strains, integration complexities, increased maintenance, and elevated project failure risk.

1. **Delivery Drivers Getting Injured on The Job (Sameh)**

There is a risk that certain areas are prone to increase of the likelihood of tragic accidents involving delivery drivers transporting food from restaurants to customers. Such incidents can lead to disruptions in service, impacting delivery timelines, create notorious image to the branding and decrease overall customer satisfaction. Moreover, the negative publicity associated with accidents can harm the reputation of the delivery platform, leading to decreased user trust and potential legal implications.

1. **Staff Dissatisfaction Throughout the Project (Maximos)**

Staff dissatisfaction can significantly impact software projects by diminishing motivation, reducing productivity, and hindering effective communication. Unhappy team members may be less inclined to collaborate, slowing down tasks and decision-making processes. This can result in project delays and compromises in software quality.

1. **Cybersecurity and Data Privacy (Mostapha)**

Food delivery systems collect and store customer data, including personal and financial information. The risks that the food delivery system can face in this area are data breaches, unauthorized access to customer data, and potential misuse of sensitive information.

1. **Operational Challenges (Mostapha)**

Food delivery systems rely on efficient logistics and operations to ensure timely and accurate deliveries. Risks in this area include timely delivery is crucial in the food delivery industry. Challenges such as traffic congestion, unforeseen circumstances (accidents, road closures), and inefficient route planning can lead to delays in delivering orders to customers., order mix-ups, inventory management issues, and technical glitches in the ordering or tracking systems.

1. **Food Safety and Quality (Mostapha)**

Maintaining food safety and quality during the delivery process is crucial. Risks can be contamination during transportation, mishandling of food by delivery personnel, and inadequate temperature control, which can lead to foodborne illnesses and customer dissatisfaction.

**Specific:**

**Analysis Phase:**

1. **Changing Requirements After Requirements Gathering: (Sameh)**

There is a risk that during the analysis phase key stakeholders may have a change of heart during the requirements gathering phase citing needs for certain features or higher expectations, causing potential misalignment with the initially documented requirements and causing greater misunderstandings and tensions among project team members and stakeholders, which in turn leads an increased risk of longer working hours, more delays, and higher costs.

1. **Insufficient Number of Partnering Restaurant: (Sameh)**

Since the application’s main spiel is variability in food items and options, there is a serious risk of lack of suitable third-party restaurants that fulfill all standards and requirements willing to partner with the app, which in turn would cause a limit on the overall variety of foods and services that the software solution can provide and impact the overall attractiveness and competitiveness of the software solution, potentially affecting the adoption and success of the system.

**Design Phase:**

1. **Staff Dissatisfaction Impact on Software Projects: (Maximos)**

Staff dissatisfaction can significantly hinder software projects by diminishing motivation, reducing productivity, and impeding effective communication. Unhappy team members may be less inclined to collaborate, slowing down task delegation and decision-making processes, ultimately extending project timelines.

1. **User Interface Design Complexity: (Maximos)**

In the context of the food delivery app, the complexity of the user interface design has the potential to introduce challenges that affect user satisfaction. An overly intricate design can lead to difficulties in user navigation, causing frustration and dissatisfaction among users. The impact is not only limited to user experience but also extends to the operational aspect of the application.

**Implementation Phase:**

1. **Weak Security Measures: (Mahmoud)**

A downside that might occur during the process of creating our system, is the possibility of a venerability or exploit existing in our system that may cause a lot of issues, the issues can include minor issues like minor bugs, or serious privacy breaches for our customers and employees.

1. **Team incompetence during implementation: (Mahmoud)**

An issue that might come up during the implementation process is a lack of team competence, where our team might commit a lot of mistakes, which might slip by the dev leaders and dev Ops, tracking these mistakes and fixing them would cause a lot of delays and a lot of effort remotivate and retrain our team.

**Testing Phase:**

1. **Performance bottlenecks of the system risks. (Mostapha)**

This risk can cause performance bottlenecks, leading to slow response times, system crashes, or degraded user experience. Performance bottlenecks can occur due to various factors, including inefficient code, database queries, network latency, or inadequate server resources. These bottlenecks can negatively impact the system's ability to handle user requests efficiently, resulting in customer dissatisfaction, loss of business, and reputational damage.

1. **A bug occurrence while performing the testing cases. (Mostapha)**

Performance, and usability are evaluated. Bugs can manifest as software defects, errors, or unexpected behavior that deviates from the intended functionality. If significant bugs go undetected during testing, they can have various negative consequences, including system malfunctions, customer dissatisfaction, and data corruption.

* **Risk Analysis and Prioritization (Highest to Lowest):**

1. **Weak Security Measures: (Mahmoud)**

The potential cost damage of $10,000,000 would be due to the amount of staff hired to track the flaw or venerability in the system, extended working hours for our developers, any legal fees that might occur during the case of a breach of data. The probability that this risk will occur is 10%.

**RE** = $10m x 0.1 = $1,000,000

1. **Insufficient Number of Partnering Restaurant Risk Analysis: (Sameh)**

There is a potential cost damage of $1,750,000 that would be caused by a stagnating and slow growth that does not match predicted numbers, in addition to hindering the overall branding of the system considering potential marketing efforts to mitigate the impact and the potential need to onboard alternative restaurants, which may come at a higher cost. The probability that this hazard will occur is 10%. Impact Level: Significant. Probability Level: Low.

Probability of loss: 0.10 BEFORE Resolution

Loss: $1,750,000

**RE** = $1.75m x 0.1 = $175,000

1. **Changing Requirements After Requirements Gathering Risk Analysis: (Sameh)**

There is a potential cost damage of $1,500,000 that include additional expenses for rework, extended working hours, and potential delays. The increased risk of misunderstandings and tensions among project team members and stakeholders could further amplify the challenges, and the probability that this hazard will occur is 5%. Impact Level: Moderate. Probability Level: Low.

Probability of loss: 0.05 BEFORE Resolution

Loss: $1,500,000

**RE** = $1.5m x 0.05 = $75,000

1. **Team incompetence during implementation: (Mahmoud)**

There is a potential cost damage of $500,000, due to development mistakes, that may slip by our dev Ops and lead developers will cost more overtime for our entire team as a whole, and may mess up our schedule and cause back for our testing team. The probability of this risk is 15%.

**RE** = $500k x 0.15 = $75,000

1. **Unexpected Increase in Project Size (Sameh)**

There is a potential cost damage of $1,250,000 that would be caused by increase in the overall size of the project beyond initial estimates which could lead to increased errors, delays, disorganization, extended development times, resource strains, integration complexities, increased maintenance, and elevated project failure risk. The probability that this hazard will occur is 5%. Impact Level: Moderate. Probability Level: Low.

Probability of loss: 0.05 BEFORE Resolution

Loss: $1,250,000

**RE** = $1.25m x 0.05 = $62,500

1. **A bug occurrence while performing the testing cases. (Mostapha)**

The impacts that may happen because of this risk can be security vulnerabilities, system malfunctions, data corruption, customer dissatisfaction, delayed launch or release, increased development costs which can cause a damage with around $30,000 the probability level for this risk is considered low (9%).

**RE** = $30,000x 0.9 = $27,000

1. **Delivery Drivers Getting Injured on The Job (Sameh)**

There is a potential cost damage of 750,000 resulting from covering for the insurance caused by tragic accidents involving delivery drivers transporting food from restaurants to customers, as well as, the negative publicity associated with accidents can harm the reputation of the delivery platform, leading to decreased user trust and potential legal implications. The probability that this hazard will occur is 3%. Impact Level: Low. Probability Level: Low.

Probability of loss: 0.03 BEFORE Resolution

Loss: $750,000

**RE** = $750,000 x 0.03 = $22,500

1. **Performance bottlenecks of the system risks. (Mostapha)**

The impacts that may happen because of this risk can be decreased customer satisfaction, loss of business egative reviews and reputation damage, increased customer churn, operational disruptions,

financial losses which can cause a damage with around $60,000 the probability level for this risk is considered low (16%).

Since RE = potential cost damage x probability level

**RE** = $60,000 x 0.16 = $9,600

1. **Inadequate Cross-Platform Compatibility: (Maximos)**

Let us consider the design may not sufficiently consider cross-platform compatibility, leading to inconsistencies and usability issues across different devices and operating systems. Users accessing the app on various platforms may experience functionality or display issues, resulting in a fragmented user experience and potential dissatisfaction. The probability of losing is 30%. The cost of losing is estimated at $30,000

Probability of loss: 0.3 BEFORE Resolution

Loss: $30,000

RE= $30,000\* 0.3 = 9000

1. **Staff Dissatisfaction Throughout the Project (Maximos)**

Staff dissatisfaction can cause big problems for software projects. People who are not happy are not as motivated and don't work as well, which means they do less and might leave their jobs. When communication is bad and people are not excited about their work, things like passing tasks to others and making decisions become slow and less effective, making the project take longer. The team might not follow strict rules to make sure the software is good, and this can make the software not as good as it could be. Staff dissatisfaction through proactive measures is crucial for maintaining a positive work environment and ensuring the project progresses smoothly and on schedule. This risk will cause damage of $100,000, and the probability that this hazard will occur is 7%

Probability of loss: 0.7 BEFORE Resolution

Loss: $100,000

RE = $100,000x 0.07 = $7,000

1. **User Interface Design Complexity: (Maximos)**

Let us consider the design of the user interface for the food delivery app may become overly complex, leading to difficulties in user navigation and increased chances of user dissatisfaction. The probability of losing the time is 10%. The cost of losing such time is measured in terms of the cost of recovering it. This is estimated at $10,000

Probability of loss: 0.1 BEFORE Resolution

Loss: $10,000

RE= $10,000\* 0.1 = 1000

* **Risk Planning:**

1. **Changing Requirements After Requirements Gathering Contingency Plan: (Sameh)**

To mitigate the risk of key stakeholders changing requirements during or after the analysis phase, it is necessary to establish open communication with stakeholders along with rigorous documentation processes and conduct regular review meetings to stay up-to-dates with requirements and involve the stakeholders in the process of data gathering and inform of them of successes and hurdles. Lastly, it is a must have them signing an agreement on paper that can be formally and legally bound to all partners involved to minimize late-stage changes, and ensure the project team is trained to handle evolving needs efficiently. Such meetings could cost upwards of $25,000 and reduce the probability of any of damage down to 0.1%. Impact Level: Low. Probability Level: Low.

Probability of loss: 0.001 AFTER Resolution

Loss: $1,500,000

Exposure 0.001 x $1,500,000= $1500

Cost of Risk Reduction: $25,000

RRL = ($75,000 - $1,500)/ $25,000 =2.94

1. **Insufficient Number of Partnering Restaurant Contingency Plan: (Sameh)**

To mitigate this risk, it is a necessity to launch a large and extensive marketing campaign to attract public attention and restaurants alike. This could also be coupled with the option to send restaurants offers that highlight the numerous benefits of collaboration from reaching a broader customer base to customized promotional opportunities, exclusive perks, and joint marketing initiatives. Such a campaign and incentive programs could cost upwards of $150,000 but could reduce the risk to at least 1%. Impact Level: Low. Probability Level: Low.

Probability of loss: 0.01 AFTER Resolution

Loss: $1,750,000

Exposure 0.01 x $1,750,000= $17,500

Cost of Risk Reduction: $350,000

RRL = ($175,000 - $17,500)/ $150,000 =1.05

1. **User Interface Design Complexity: (Maximos)**

Now we provide a method of reducing this problem as we can Conduct user testing and feedback sessions during the design phase to ensure that the interface is user-friendly this reduces the risk to 1%. However, the cost of introducing the loss reduction is $800

Probability of loss: 0.01 AFTER Resolution

Loss: $10,000

Exposure to data loss: $10,000\* 0.01 = 100

RRL = (1000-100)/800=1.125

1. **Inadequate Cross-Platform Compatibility: (Maximos)**

Implement responsive design principles, conduct testing on multiple devices and operating systems, and ensure that the user interface adapts seamlessly to different screen sizes and resolutions. this reduced the risk to 4%. However, the cost of introducing the loss reduction is $1000

Probability of loss: 0.04 AFTER Resolution

Loss: $30,000

Exposure to data loss: $30,000\* 0.04 = 1200

RRL = (9000-1200)/1000=7.8

1. **Performance bottlenecks of the system risks (Mostapha)**

There is a risk that the food delivery system may experience performance bottlenecks, resulting in slow response times, system crashes, or degraded user experience. This could lead to customer dissatisfaction, loss of business, and reputational damage. Based on historical data and system complexity analysis, the probability of performance bottlenecks occurring is estimated to be 22%. This would cause financial losses due to decreased customer orders and potential refunds: $60,000. To avoid this risk the testing team decided to Optimize the system's code, database queries, and algorithms to improve efficiency and reduce response times. Consider techniques such as code refactoring, query optimization, caching, and asynchronous processing to streamline operations and minimize bottlenecks. However, this would cost about $5,000 and would decrease the probability of this risk by 15%.

Here's a breakdown of the risk exposure and cost analysis before and after implementing the risk reduction measure:

BEFORE Resolution:

Probability of bottlenecks occurring: 0.22

Loss exposure: $60,000

Expected loss: 0.22 x $60,000 = $13,200

AFTER Resolution:

Probability of bottlenecks occurring: 0.07

Loss exposure: $60,000

Expected loss: 0.07 x $60,000 = $4,200

Cost of risk reduction (code reviews): $5,000

Risk Reduction Leverage (RRL)= (Risk Exposure Before - Risk Exposure After) / Cost of Risk Reduction

RRL = (13,200 – 4,200) / 5000 = 1.8 So it worth doing it.

1. **A bug occurrence while performing the testing cases (Mostapha)**

While we’re constructing the food delivery system, there would be a risk associated with a critical bug going undetected during the testing phase. The probability that such a bug to occur is around 19%. If this bug hasn’t been detected, this could lead to severe consequences, such as financial losses, data corruption, or system crashes which we'll cost around $30,000.

To avoid this risk, we decide to introduce an additional testing technique which called code review. Which requires a senior developer. This technique has proven effective results in improving code quality, identifying bugs. By implementing this risk reduction measure, we estimate that the probability of the critical bug if undetected will decrease to 6%. However, conducting this code reviews would cost an additional cost, such as the effort and time of the developer, which amounts $3,000.

Here's a breakdown of the risk exposure and cost analysis before and after implementing the risk reduction measure:

BEFORE Resolution:

Probability of critical bug if undetected: 0.19

Loss exposure: $30,000

Expected loss: 0.19 x $30,000 = $5,700

AFTER Resolution:

Probability of critical bug if undetected: 0.06

Loss exposure: $30,000

Expected loss: 0.06 x $30,000 = $1,800

Cost of risk reduction (code reviews): $3,000

Risk Reduction Leverage (RRL)= (Risk Exposure Before - Risk Exposure After) / Cost of Risk Reduction

RRL = (5700 – 1800) / 3000 = 1.3 So it worth doing it.

1. **Weak Security Measures: (Mahmoud)**

To deal with this major issue that would lead to major losses, a team of cybersecurity specialists is hired to vet our progress on regular bases. The team is meant to make sure there are no major flaws or exploits by using the most advanced and safe security measures in our code throughout the implementation process. This would massively cut the probability of any major exploits occurring down to 0.1%, but it would cost us $500,000 to hire a professional team of cybersecurity experts.

Probability of loss: 0.001 AFTER Resolution

Loss: $10,000,000

Exposure 0.001 x $10,000,000= $10,000

Cost of Risk Reduction: $500,000

RRL = ($1,000,000 - $10,00)/ $500,000 = 1.98

1. **Team incompetence during implementation: (Mahmoud)**

To Keep our motivated, we entice them with extra bonuses if they meet their goals, so everyone is motivated to work and make sure there are minimum errors. The bonuses will cost us around $50,000 to satisfy all of our development team. And drop down the incompetence to 1%

Probability of loss: 0.01 AFTER Resolution

Loss: $500,000

Exposure 0.01 x $500,000 = $5,000

Cost of Risk Reduction: $50,000

RRL = ($75,000 - $5,000)/ $50,000 = 1.4

# SECTION 6. TECHNICAL PROCESS

## 6.2 Methods, Tools and Techniques

Methods:

* SP Method: PRINCE2

PRINCE2 is a methodology that fits the rapid and continuously evolving state of the food delivery industry that is at the whims of changing customer preferences and market conditions; PRINCE2's adaptability and emphasis on risk management provide a strong foundation for navigating these challenges. Furthermore, PRINCE2's ensure that everyone understands their duties and contributes effectively to the project's success. Its scalability, flexibility and robust project management approach can help develop and scale the food delivery app further down the line while maintaining the aspects of accountability and responsibility throughout.

* SD Method: Scrum (Agile methodology)

Scrum encourages regular stakeholder involvement, which makes it possible to quickly adjust to shifting consumer preferences and market dynamics. Scrum helps with continuous improvement, quick feedback, and the capacity to prioritize and solve changing customer requirements, all of which are crucial in the fast-paced, fiercely competitive food delivery market. Furthermore, Scrum's focus on openness, frequent review sessions, and early risk detection can assist in guaranteeing an effective, customer-focused, and successful development process for this kind of project.

Tools:

* VS Code IDE: Used for coding and development tasks.
* Adobe XD: Used for designing the app's user interface and user experience.
* Canva: Used for creating graphic assets, such as promotional materials.
* GitHub: Used for version control and collaborative code management.
* Microsoft SQL Server Manager: Manage and maintain the system’s relational database.
* AWS: Provides cloud hosting and infrastructure for the app's deployment and scalability.

Techniques:

1. HTML, CSS, JavaScript, Java: These programming languages are used for the development of the system’s website. HTML, CSS and JavaScript will be utilized in creating the front-end portion of the website, while Java will be used to manage the back-end side of the website.
2. Flutter: Flutter will aid in building the mobile-centric app of the system by developing a cross-platform mobile applications with a single codebase, ensuring consistent user experiences on both Android and iOS devices and guaranteed ease of use.