جامعة أم القرى
كلية الحاسبات

SOFTWARE DEVELOPMENT
TOOLS PROJECT REPORT

Flower shop website

Team Members:

jumana abdularazaq alrehili 443009827

Lamar bandar felemban 444003576

Rola saleh alkhusayfi 4440090107

Prepared for: Eman Alharbi

Table of Contents

Introduction:

This report outlines the development of a flower shop software designed to simplify the process of purchasing bouquets for customers. Leveraging Unix shell scripts and software development tools, the team created an application that streamlines the flower selection and .purchasing workflow

To facilitate teamwork and collaboration, the project utilized GitHub, which enabled version control, change tracking, and effective .communication among the development team

The primary goal of the flower shop software is to provide customers with an intuitive interface that allows them to easily browse, select, and purchase the perfect bouquets for their needs. By automating and enhancing the customer experience, the software aims to improve the .overall efficiency of the flower shop's operations

This report will provide a detailed overview of the development process, the key features and functionalities of the software, and the strategic decisions made throughout the project.

The purpose of the report:

The purpose of this report is to showcase the development of a flower shop software that simplifies the customer purchasing experience. The goal is to ,provide an efficient and user-friendly platform for customers to easily brows select, and purchase the perfect flower bouquets.

This report will detail the development process, including the use of Unix shell scripts and software tools, as well as highlight the collaborative efforts of the team in creating this program.

- ## Inside Software (Functions and Operations):

The services offered by our flower store software encompass a range of functionalities to meet the diverse needs of customers and flower store operators. These services include:

1. **Welcome Screen:** Upon accessing the flower shop software, the customer is greeted with a welcome screen
2. **Information Input:** The customer is presented with a form where they can enter their name, contact number, and delivery location. This form include these fields:
    - **Name**: The customer provides their full name.
    - **Contact Number:** The customer enters their phone number for communication purposes.
    - **Location**: The customer inputs their delivery location
3. **Flower Selection:** Displays available flower types with their prices. Allows the user to select the type and quantity of flowers they want to include in the bouquet. It continues to ask if more flowers are needed until the user declines.
4. **Total Price Calculation:** Calculates the total price of the selected flowers.
5. **Customization Options:** Customers have the flexibility to customize their bouquets by selecting wrapping colors, and ribbon colors, ensuring that their purchase meets their unique preferences and requirements.
6. **Delivery or Pickup Selection:** Asks whether the user wants delivery or pickup and prompts for the respective date.
7. **Payment Processing**: Asks for the payment method. If the user chooses cash on delivery/pickup or PayPal, it concludes the order. If the user chooses card payment, it asks for card details, validates them, and prompts if the user wants to save the card information.
8. **Order Summary:** Displays a summary of the order, including order number, customer name, phone number, selected location, delivery/pickup details, selected flowers with quantities, wrapping details, and total price.
9. **Confirmation Message**: Depending on the delivery or pickup choice and payment method, it provides a confirmation message indicating either the delivery date or pickup date.

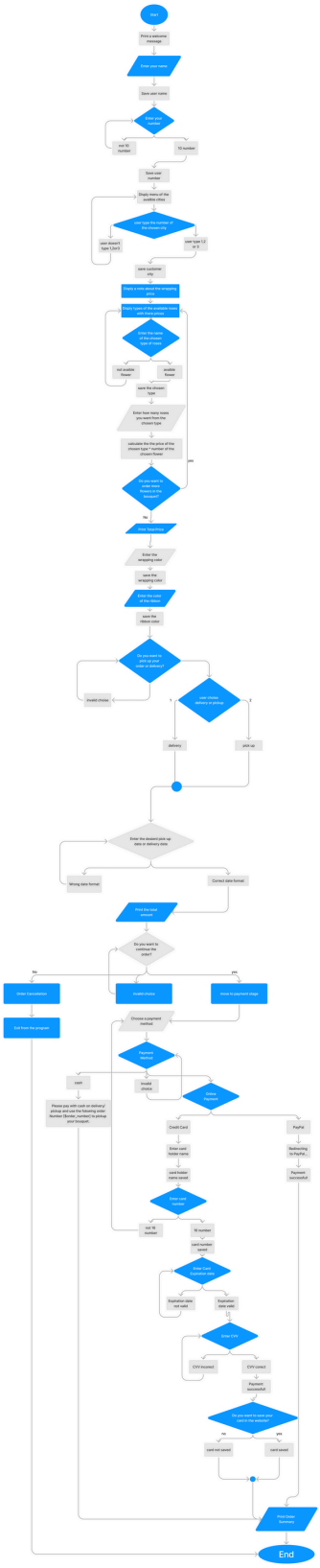
- # Inside Software (Flowchart and senarios):

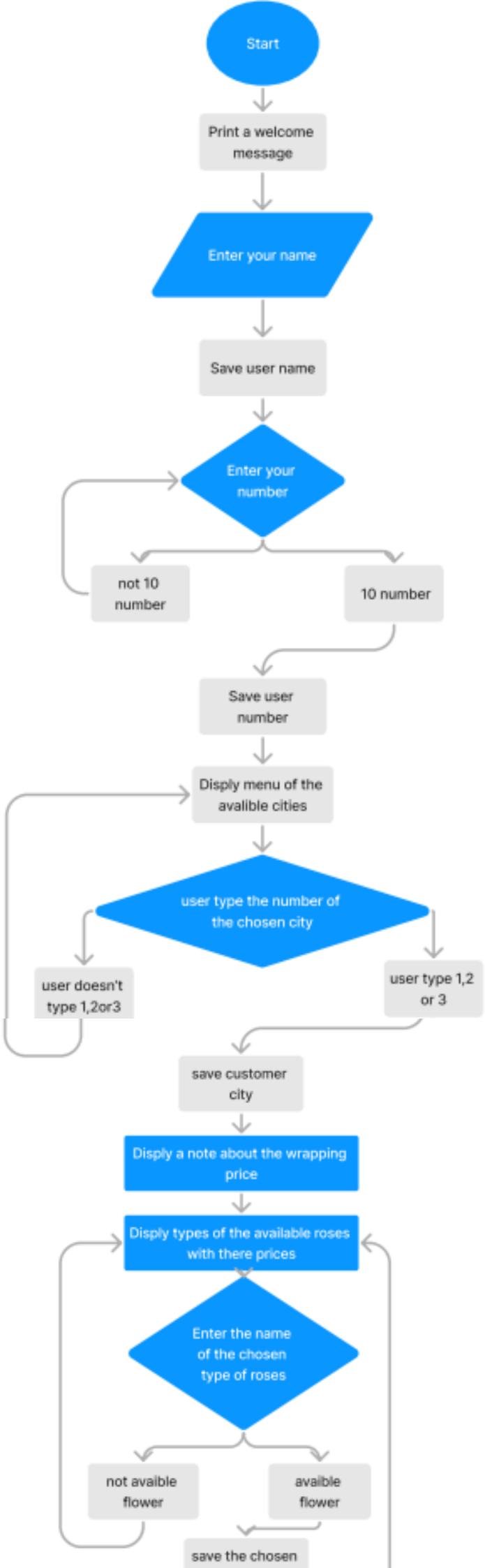
Software Flowchart

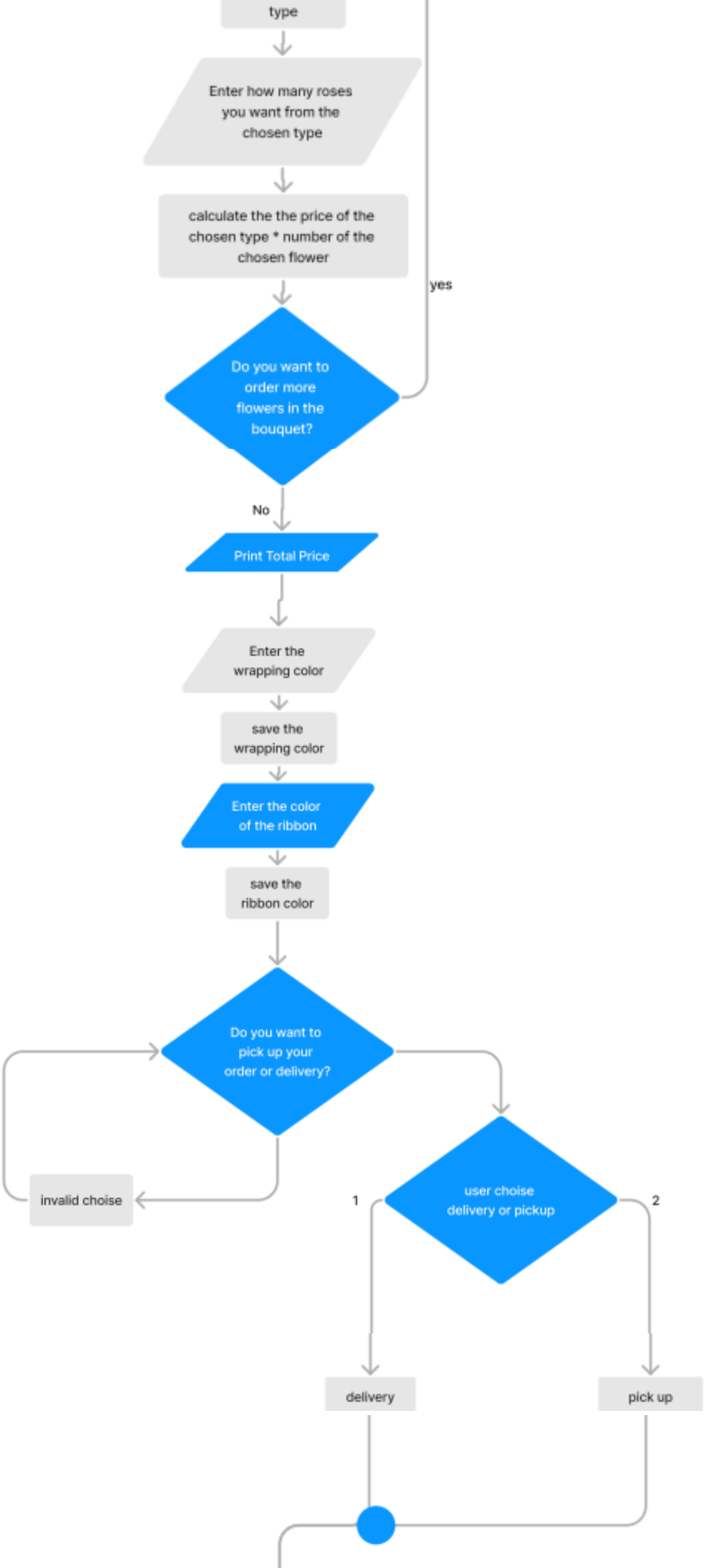
The software flowchart provides a comprehensive visual representation of the various scenarios and pathways within the flower shop software. It elucidates the sequence of actions and decision points encountered by users throughout their interaction with the application.

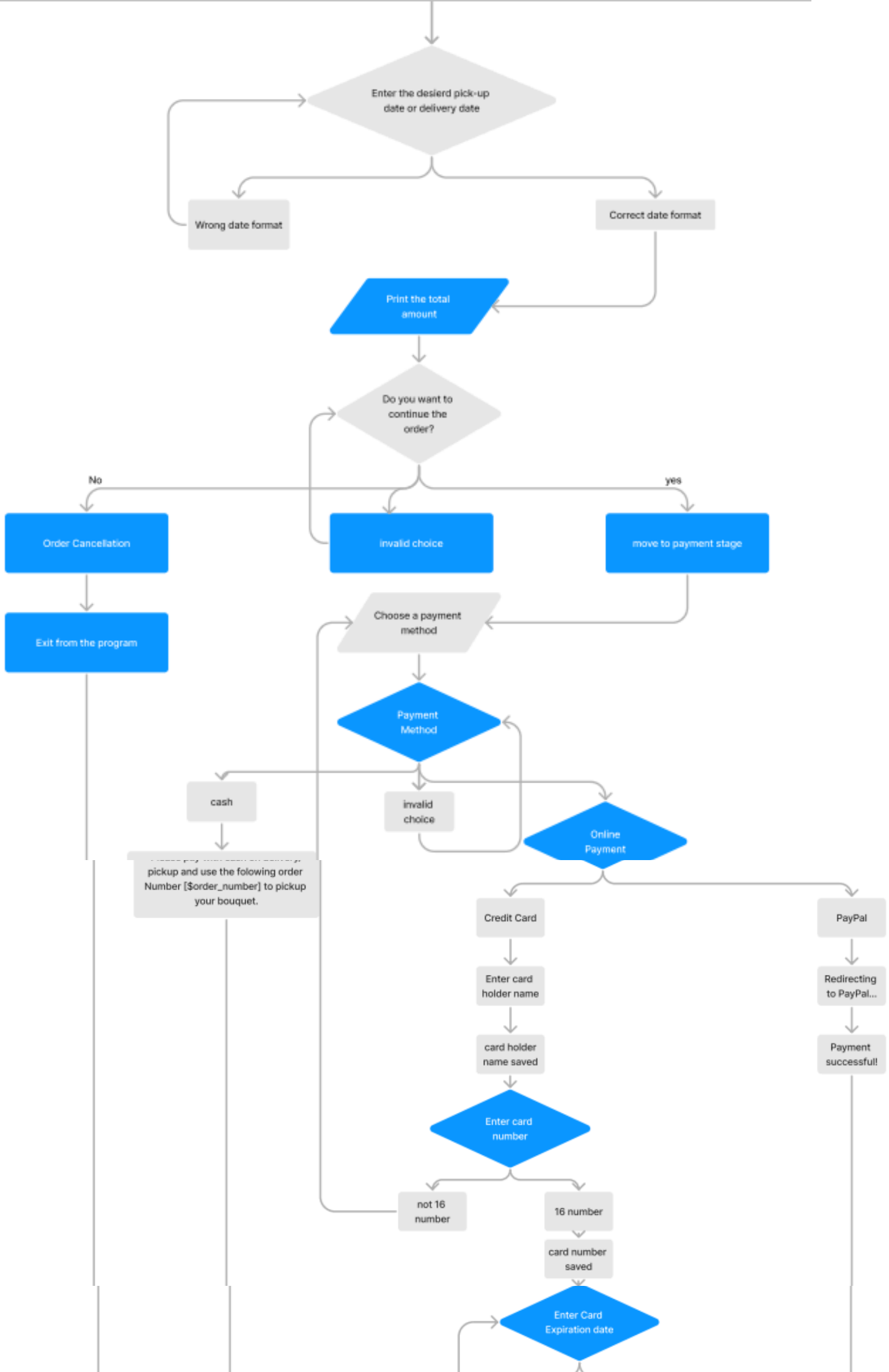
here is a link for compelet viwe

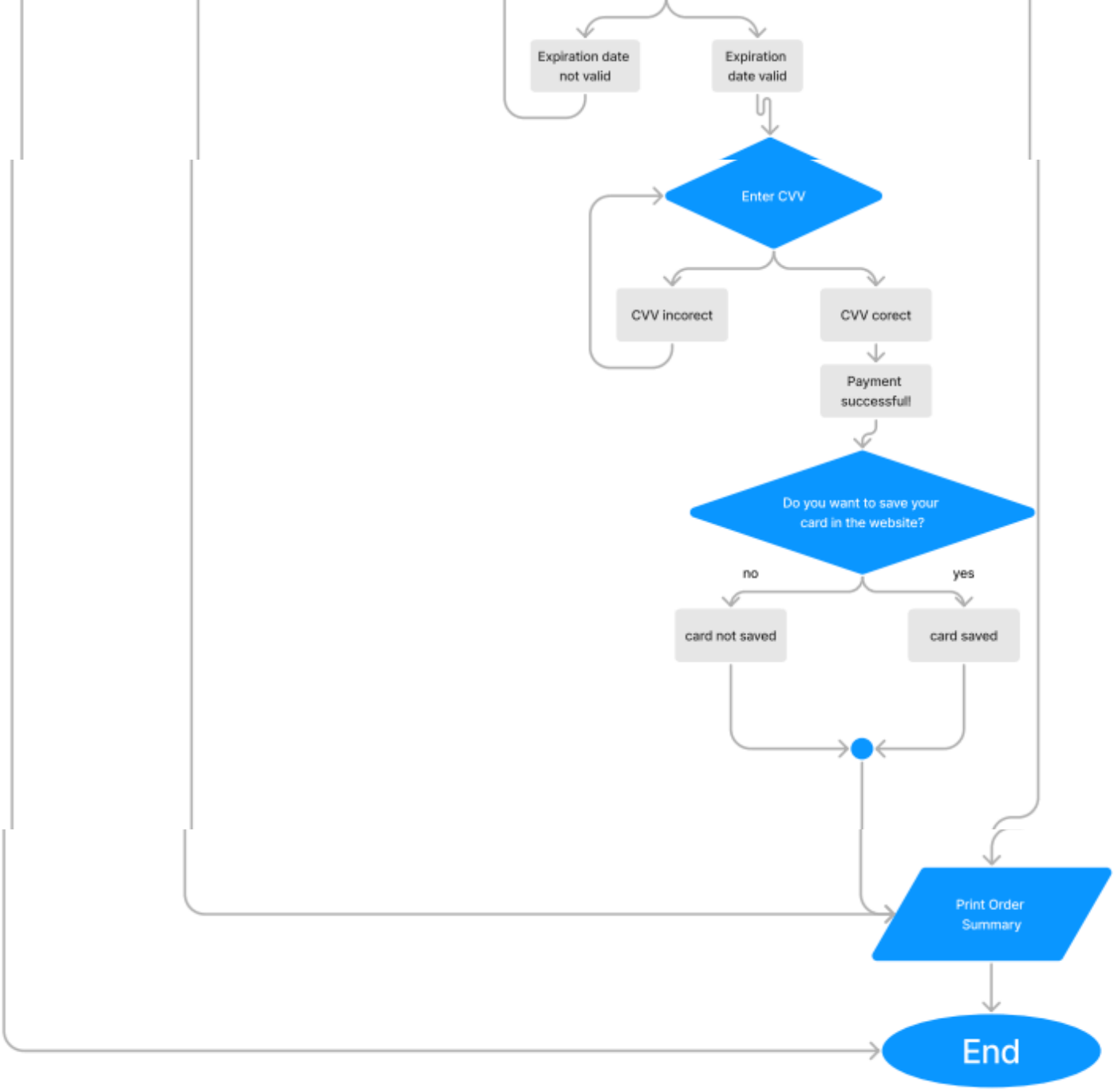
[flowchart link]




Start
Print a welcome message
Enter your name
Save user name
Enter your number
not 10 number
10 number
Save user number
Disply menu of the avalible cities
user type the number of the chosen city
user doesn't type 1,2or3
user type 1,2 or 3
save customer city
Disply a note about the wrapping price
Disply types of the available roses with there prices
Enter the name of the chosen type of roses
not avaible flower
availble flower
save the chosen

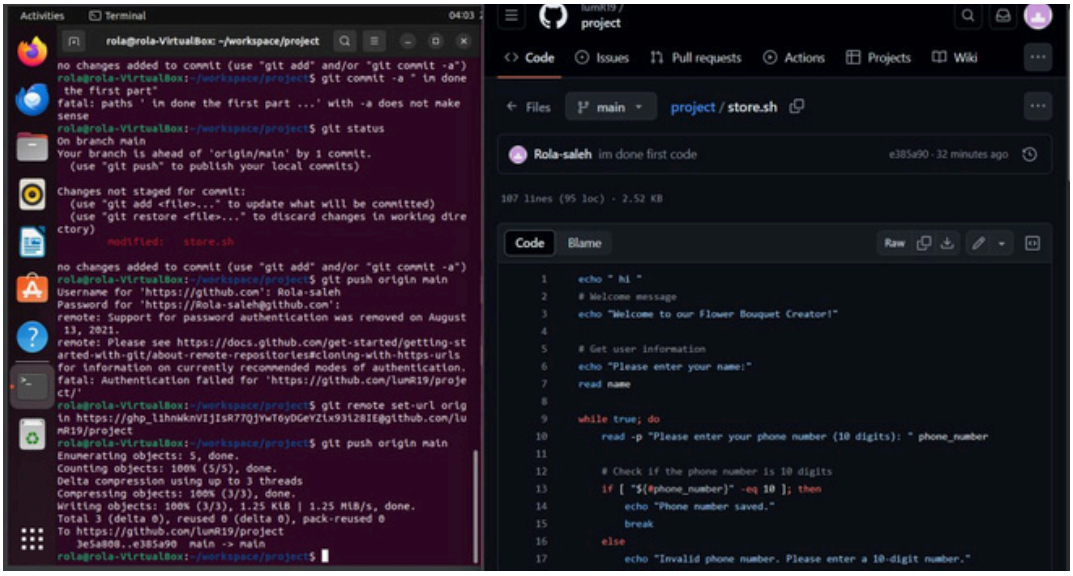

type
Enter how many roses you want from the chosen type
calculate the the price of the chosen type * number of the chosen flower
yes
Do you want to order more flowers in the bouquet?
No
Print Total Price
Enter the wrapping color
save the wrapping color
Enter the color of the ribbon
save the ribbon color
Do you want to pick up your order or delivery?
invalid choise
user choise delivery or pickup
1
2
delivery
pick up


Enter the desierd pick-up date or delivery date
Wrong date format
Correct date format
Print the total amount
Do you want to continue the order?
No
yes
Order Cancellation
invalid choice
move to payment stage
Exit from the program
Choose a payment method
Payment Method
cash
invalid choice
Online Payment
pickup and use the folowing order Number [$order_number] to pickup your bouquet.
Credit Card
PayPal
Enter card holder name
Redirecting to PayPal...
card holder name saved
Payment successful!
Enter card number
not 16 number
16 number
card number saved
Enter Card Expiration date

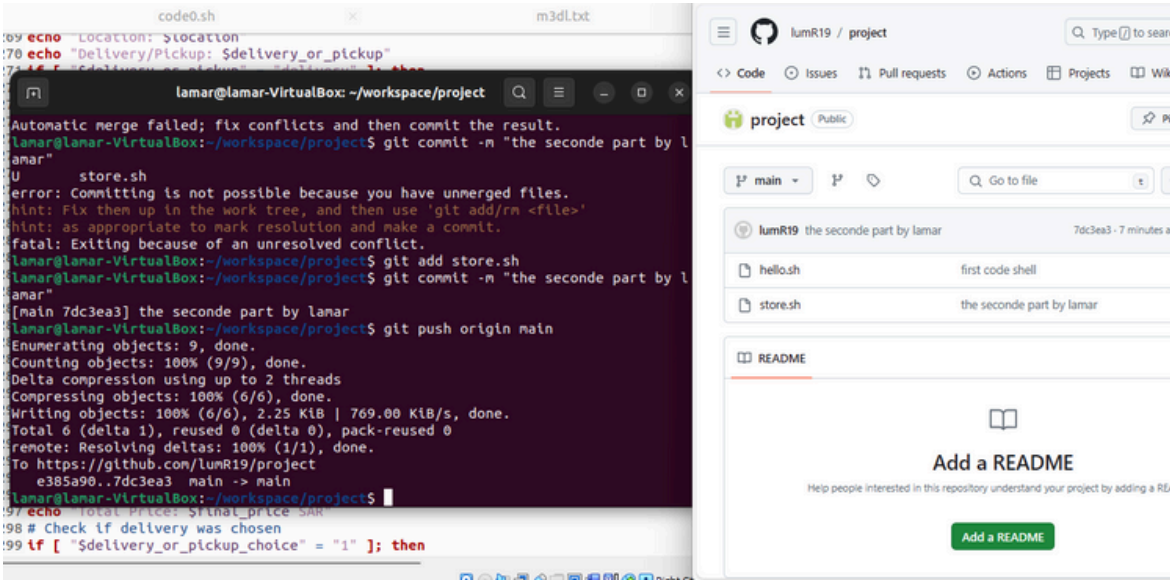

Expiration date not valid
Expiration date valid
Enter CVV
CVV incorect
CVV corect
Payment successfull
Do you want to save your card in the website?
no
yes
card not saved
card saved
Print Order Summary
End

- # Teamwork via GitHub:

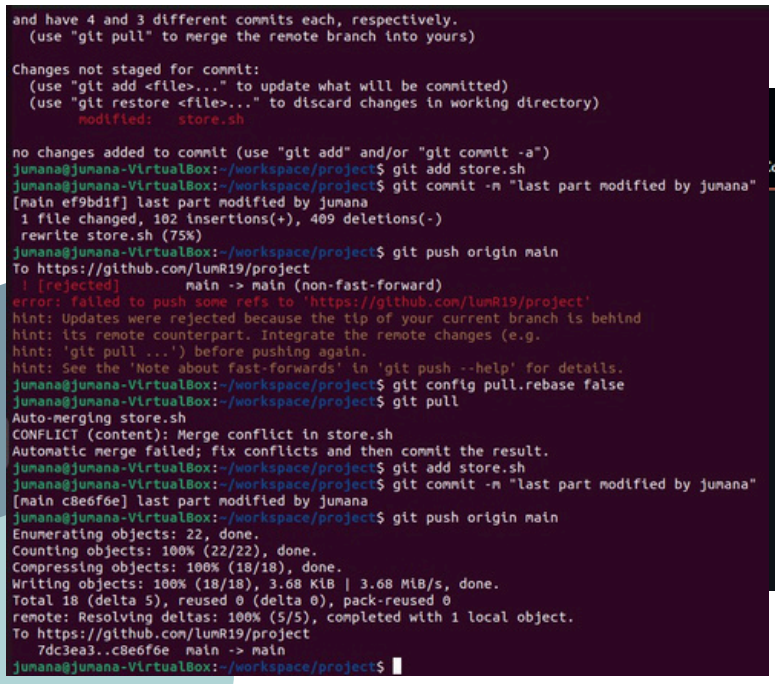
Rola's use of Github



Lamar's use of Github



Jumana's use of Github



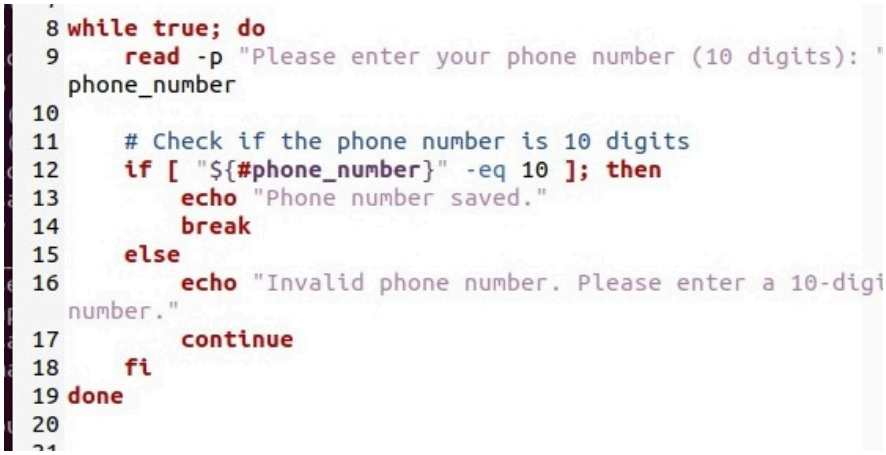
Software implementation:

- ## Rola’s code:

Let's walk through the code line by line, explaining what each part does:

```
1 # Welcome message
2 echo "Welcome to our Flower Bouquet Creator!"
3
4 # Get user information
5 echo "Please enter your name:"
6 read name
```

- This line displays a welcome message to the user, indicating the start of the flower bouquet creation process.
- Here, the script prompts the user to enter their name using the **read** command, which reads input from the user and assigns it to the variable **name**.



```
8 while true; do
9     read -p "Please enter your phone number (10 digits): "
  phone_number
10
11     # Check if the phone number is 10 digits
12     if [ "${#phone_number}" -eq 10 ]; then
13         echo "Phone number saved."
14         break
15     else
16         echo "Invalid phone number. Please enter a 10-digi
  number."
17         continue
18     fi
19 done
20
```

- This initiates a loop that continues until a valid phone number is provided. It prompts the user to enter their phone number and reads the input into the variable **phone_number**.
- the condition checks if the length of the phone number entered by the user is exactly 10 digits using the **${#phone_number}** expression, which returns the length of the string stored in the **phone_number** variable.
- If the phone number is valid, it prints a confirmation message and exits the loop using the break statement. If the phone number is invalid, it prompts the user to enter a 10-digit number again by continuing the loop with the continue statement.

```
22 # Generate order number
23 order_number=$RANDOM
```

- This line generates a random order number using the **$RANDOM** variable, which generates a random integer between 0 and 32767. However, it's important to note that this method may produce duplicate order numbers if the script is run multiple times in quick succession.

```
25 # Location options
26 location_selected=false
27 while [ "$location_selected" = false ]; do
28     echo "Please select your location:"
29     echo "1. Makkah"
30     echo "2. Jeddah"
31     echo "3. Taif"
32     read location_choice
33
34     case $location_choice in
35         1)
```

- This initiates a loop that continues until the user selects a valid location for delivery or pickup. The variable location_selected is initially set to false to indicate that a location has not been selected yet.

- Inside the loop, a **case** statement is used to handle different location choices entered by the user. Depending on the choice, the **location** variable is assigned the corresponding location name, and **location_selected** is set to **true** to exit the loop. If the user enters an invalid choice, an error message is displayed, and the loop continues.

```
52
53 # Flower options
54 declare -A flower_prices=(
55     [Rose]=37.5
56     [Lily]=45
57     [Tulip]=30
58     [Sunflower]=56.25
59     [Orchid]=75
60     [Carnation]=22.5
61     [Daisy]=26.25
62     [Peony]=67.5
63     [Chrysanthemum]=52.5
64     [Iris]=41.25
65 )
```

- This section declares an associative array named flower_prices where each flower type is associated with its price in SAR (Saudi Riyal).

```
67 echo "___________"
68 echo "Note : the wrapping price is = 25 SAR no matter how mucl
   flowers you chosed."
```

- These lines display a note informing the user about the fixed price of wrapping, which is 25 SAR, regardless of the number of flowers chosen.

```
69 # Function to display flower options
70 display_flower_options() {
71     echo "___________"
72     echo "Available flower types:"
73     for flower in "${!flower_prices[@]}"; do
74         echo "$flower (${flower_prices[$flower]} SAR)"
75     done
76     echo "___________"
77 }
```

- This defines a function named **display_flower_options**() that displays available flower types along with their respective prices. It iterates through the keys of the **flower_prices** associative array, which represent flower types, and prints each flower type along with its price.

```
# Main script
flower_types=()
flower_quantities=()
while true; do
    display_flower_options
    # Get flower selection
    while true; do
        echo "Please select the flower type ,please enter the name exactly as written (including capital and small letters)"
        read flower_type
        if [[ -n ${flower_prices[$flower_type]} ]]; then
            flower_types+=("$flower_type")
            echo "How many of $flower_type would you like?"
            read flower_quantity
            flower_quantities+=("$flower_quantity")
            break
        else
            echo "Invalid flower type. Please select from the list."
        fi
    done
```

- The main script starts here, initiating a loop that allows users to select one or more flowers for their bouquet. It will continue until the user decides not to order more flowers.
- This line calls the **display_flower_options**() function to show the available flower types and their prices to the user.
- Inside the loop, the script prompts the user to enter the name of the flower they want to add to the bouquet. The user's input is stored in the variable **flower_type**.
- This condition checks if the user's input matches one of the keys in the **flower_prices** associative array, ensuring that the entered flower type is valid.
- If the flower type entered by the user is valid, the script adds it to the flower_types array and prompts the user to enter the quantity of that flower. The quantity entered by the user is stored in the flower_quantities array.
- If the flower type entered by the user is invalid, the script notifies the user and prompts them to select a flower type again. This loop continues until a valid flower type is selected.

```
    # Ask if more flowers are needed
    echo "Do you want to order more flowers in the same bouquet? (yes/no)"
    read more_flowers
    if [[ $more_flowers == "no" ]]; then
        break
    fi
done
```

- After the user selects the desired quantity of a flower, the script asks if the user wants to order more flowers in the same bouquet. If the user answers "no," the loop breaks, and the script proceeds to the next section. Otherwise, the loop continues, allowing the user to select more flowers.

- **Lamar's code:**

```
110 # Function to calculate total price without wrapping
111 calculate_total_price() {
112     local total=0
113     for i in "${!flower_types[@]}"; do
114         price_per_unit=${flower_prices[${flower_types[$i]}]}
115         quantity="${flower_quantities[$i]}"
116         total=$(echo "scale=2; $total + $price_per_unit * $quantity" | bc)
117     done
118     echo "$total"
119 }
```

- This defines a function named **calculate_total_price**() that calculates the total price of the selected flowers without considering the wrapping cost. It iterates through the arrays flower_types and **flower_quantities**, retrieves the price per unit of each flower from the **flower_prices** associative array, and calculates the total price by multiplying the price per unit with the quantity.

```
121 # Calculate total price
122 total_price=$(calculate_total_price)
123 echo "Total price for the selected flowers: $total_price SAR"
124
```

- The script calculates the total price of the selected flowers by calling the calculate_total_price() function and assigns the result to the variable total_price. Then, it displays the total price to the user.

```
125 # Wrapping options
126 echo "Please select the wrapping color:"
127 read wrap_color
128 echo "Please select the ribbon color:"
129 read ribbon_color
```

- Next, the script prompts the user to select the wrapping color and ribbon color for the bouquet, allowing for customization options.

```
131 # Ask for delivery or pickup
132 while true; do
133     read -p "Do you want delivery or pickup? (1 for delivery, 2 for pickup): "
    delivery_or_pickup_choice
134     if [ "$delivery_or_pickup_choice" = "1" ]; then
135         delivery_or_pickup="delivery"
136         while true; do
137         read -p "Enter the desired delivery date (YYYY/MM/DD): " delivery_date
138         if [[ "$delivery_date" =~ ^[0-9]{4}/[0-9]{2}/[0-9]{2}$ ]]; then
139             break
140         else
141             echo "Invalid date format."
142         fi
143     done
144         break
145
157     else
158         echo "Invalid choice. Please try again."
159     fi
160 done
161 while true; do
162     read -p "Total amount of $total_price SAR ,Do you want to continue the paying? (y/n): "
    pay_choice
163     if [ "$pay_choice" = "y" ]; then
164         while true; do
165       read -p "Please select a payment method (1 for card, 2 for cash on delivery/pickup, or 3
    for PayPal): " payment_method
166
```

- This part of the script prompts the user to choose between delivery and pickup options. If the user selects delivery, they are asked to input the desired delivery date. If pickup is chosen, the script prompts the user to enter the desired pickup date. The loop ensures that the user provides a valid choice.

```
167       if [ "$payment_method" = "2" ]; then
168           echo "Please pay with cash on delivery/pickup and use the folowing order Number
    [$order_number] to pickup your bouquet."
169
170           break
171       elif [ "$payment_method" = "1" ]; then
172           read -p "Enter the card holder name: " card_holder_name
173           read -p "Enter the card number: " card_number
174
175           # Check if the card number is 16 digits
176           if [ "${#card_number}" -eq 16 ]; then
177
178 while true; do
179       # Prompt for card expiration date
180       read -p "Enter card expiration date (MM/YY): " exp_date
181
182       # Validate card expiration date format
183       if [[ "$exp_date" =~ ^[0-9]{2}/[0-9]{2}$ ]]; then
184           # Extract month and year from expiration date
185           exp_month=${exp_date:0:2}
186           exp_year=${exp_date:3:2}
187
188           # Check if expiration date is in the future
189           current_year=$(($(date +%y)))
190           if [[ "$exp_year" -gt "$current_year" || ("$exp_year" -eq "$current_year" &&
    "$exp_month" -gt $(date +%-m)) ]]; then
191               echo "Valid expiration date."
192               break
193           else
194               echo "Card expired. Please enter a valid expiration date."
```

- Here, the script asks the user if they want to proceed with payment. If the user chooses to continue ("y"), the script proceeds to the payment section.
- Within this nested loop, the script prompts the user to select a payment method: card payment, cash on delivery/pickup, or PayPal.
- If the user chooses cash on delivery/pickup, the script displays a message instructing the user to pay with cash and provides the order number for pickup.
- If the user chooses card payment, the script prompts them to enter the card holder's name and card number.
- checks if the entered card number has exactly 16 digits.
- If the card number is valid, the script prompts the user to enter the card's expiration date.

```
195               continue
196           fi
197       else
198           echo "Invalid expiration date format. Please try again."
199           continue
200       fi
201 done
202
203           while true; do
204       # Prompt the user to enter the CVV
205       read -p "Enter the card CVV: " cvv
```

- **Jumana's code:**

```
206
207       # Check if the CVV is a 3-digit number
208       if [[ ${#cvv} -eq 3 && $cvv =~ ^[0-9]+$ ]]; then
209           echo "Valid CVV: $cvv"
210           break
211       else
212           echo "Invalid CVV. Please enter a 3-digit number."
213           continue
214       fi
215 done
```

- The script ensures that the CVV (Card Verification Value) entered by the user is exactly 3 digits.
- If all payment information is provided correctly, the script confirms the payment success and asks the user if they want to save the card information.
- Depending on the user's choice, the script informs whether the card information has been saved or not.

```
216
217             echo "Payment successful!"
218             break
219
220             read -p "Do you want to save this card on our website? (y/n) " save_card
221             if [ "$save_card" = "y" ]; then
222                 echo "Card information saved."
223             else
224                 echo "Card information not saved."
225             fi
226             break
227         else
228             echo "Invalid card number. Please try again."
229             continue
230         fi
231     elif [ "$payment_method" = "3" ]; then
232         echo "Redirecting to PayPal..."
233         echo "Payment successful!"
234        break
235     else
236         echo "Invalid payment method. Please try again."
237     fi
238 done
239
240         break
241     elif [ "$pay_choice" = "n" ]; then
242         echo "Order cancelled."
243         break
244     else
245         echo "Invalid choice. Please try again."
246     fi
247 done
248
249 # Function to calculate total price with wrapping
250 calculate_total_price1() {
251     local flower_total=0
252     for i in "${!flower_types[@]}"; do
253         price_per_unit=${flower_prices[${flower_types[$i]}]}
254         quantity="${flower_quantities[$i]}"
```

- This function, calculate_total_price1(), calculates the total price of selected flowers including the wrapping cost.
- It iterates over the selected flower types stored in the flower_types array and calculates the total price by adding the individual flower prices.
- The wrapping cost is then added to the total.
- The resulting final price is returned.

```
256         flower_total=$(echo "scale=2; $flower_total + $price_per_unit *
    $quantity" | bc)
257     done
258
259     local wrapping_price=25
260     local final_price=$(echo "scale=2; $flower_total + $wrapping_price" | bc)
261     echo "$final_price"
262 }
263
264 # Display order summary
265 while true; do
266 if [ "$pay_choice" == "y" ]; then
267      echo "_____"
268      echo "Order Summary:"
269      echo "Order Number: $order_number"
270      echo "Name: $name"
271      echo "Phone: $phone_number"
272      echo "Location: $location"
273
274     if [ "$delivery_or_pickup_choice" == "1" ]; then
275         echo "Delivery/Pickup: Delivery"
276     else
277         echo "Delivery/Pickup: Pickup"
278     fi
279
280     if [ "$delivery_or_pickup_choice" = "1" ]; then
281         echo "Delivery Date: $delivery_date"
282     else
283         echo "Pickup Date: $pickup_date"
284     fi
```

- Display the summery

```
    for ((i=0; i<${#flower_types[@]}; i++)); do
        echo "Flower: ${flower_quantities[$i]} x ${flower_types[$i]} ($
{flower_prices[$   {flower_types[$i]}]} SAR each)"
    done
    echo "Wrapping: $wrap_color with $ribbon_color ribbon (25 SAR)"

    # Function to calculate total price with wrapping
    calculate_total_price1() {
        local flower_total=0
        for i in "${!flower_types[@]}"; do
            price_per_unit=${flower_prices[${flower_types[$i]}]}
            quantity="${flower_quantities[$i]}"
            flower_total=$(echo "scale=2; $flower_total + $price_per_unit *
$quantity" | bc)
        done

     local wrapping_price=25
     local final_price=$(echo "scale=2; $flower_total + $wrapping_price" | bc)
     echo "$final_price"
     }
    # Calculate total price
    final_price=$(calculate_total_price1)
    echo "Total Price: $final_price SAR"
    # Check if delivery was chosen
    if [ "$delivery_or_pickup_choice" = "1" ]; then
    # Display success message if payment was successful
        if [ "$payment_method" = "1" ] || [ "$payment_method" = "3" ]; then
             echo "Thank you, the delivery representative will contact you to
deliver it to your door."
        else

             echo "Your order will be ready for pickup on $pickup_date
$delivery_date ."

        fi

    else

        echo "Your order will be ready for pickup on $pickup_date
$delivery_date ."

    fi
    break
else

   break

 fi
done
```

- Here, the total price of the bouquet, including the wrapping cost, is calculated using the calculate_total_price1() function and stored in the variable final_price.
- ThenThe script prints out the total price.

Validating Software Output:

Ensuring that the software generates valid output without errors is crucial for maintaining reliability and user satisfaction. Validation procedures involve verifying that the output conforms to expected standards and accurately reflects the input provided by the user. Below are examples of how the flower shop software output can be validated:

```
Welcome to our Flower Bouquet Creator!
Please enter your name:
rola
Please enter your phone number (10 digits): 123456
Invalid phone number. Please enter a 10-digit number.
Please enter your phone number (10 digits): 1234567899
Phone number saved.
Please select your location:
1. Makkah
2. Jeddah
3. Taif
2

__________
Note : the wrapping price is = 25 SAR no matter how much flowers you chosed.
__________
Available flower types:
Lily (45 SAR)
Carnation (22.5 SAR)
Daisy (26.25 SAR)
Orchid (75 SAR)
Tulip (30 SAR)
Rose (37.5 SAR)
Iris (41.25 SAR)
Sunflower (56.25 SAR)
Chrysanthemum (52.5 SAR)
Peony (67.5 SAR)
_____________________________
Please select the flower type ,please enter the name exactly as written (includi
ng capital and small letters):
Rose
How many of Rose would you like?
3
Do you want to order more flowers in the same bouquet? (yes/no)
```

If the user selected choise number (eg; 4,5..) it will disply an Error massege "Invalid location choice. Please try again."

If the user entered wrong name or the name of the flower without accounting for capitalization, or small letters it will disply an error massege "Invalid flower type. Please select from the list."

Con..

- If the user typed “yes” it will Continue asking the user about the type and the number for flowers from the chosen type and stops once the user type “no” :

```
Peony (67.5 SAR)
_____________________________
Please select the flower type ,please enter the name exactly as written (includi
ng capital and small letters):
Iris
How many of Iris would you like?
2
Do you want to order more flowers in the same bouquet? (yes/no)
yes
_____________________________
Available flower types:
Lily (45 SAR)
Carnation (22.5 SAR)
Daisy (26.25 SAR)
Orchid (75 SAR)
Tulip (30 SAR)
Rose (37.5 SAR)
Iris (41.25 SAR)
Sunflower (56.25 SAR)
Chrysanthemum (52.5 SAR)
Peony (67.5 SAR)
_____________________________
Please select the flower type ,please enter the name exactly as written (includi
ng capital and small letters):
Rose
How many of Rose would you like?
3
Do you want to order more flowers in the same bouquet? (yes/no)
no
Total price for the selected flowers: 82.50 SAR
Please select the wrapping color:
```

- The user chose the desired colors for wrapping and ribbon:

```
Please select the wrapping color:
White
Please select the ribbon color:
black
```

Con..

- The user select between “Delivery” or “Pickup”. if he selectd invalid choise (eg; 3,4..) it will disply an error masege as shown:

```
Do you want delivery or pickup? (1 for delivery, 2 for pickup): 3
Invalid choice. Please try again.
Do you want delivery or pickup? (1 for delivery, 2 for pickup): 1
Enter the desired delivery date (YYYY/MM/DD):
```

- The user enter the desired delivery date , if he enterd the date in wrong format it will disply an error massege as shown :

```
Enter the desired delivery date (YYYY/MM/DD): 2024-07-01
Invalid date format.
Enter the desired delivery date (YYYY/MM/DD): 2024/07/01
Total amount of 165.00 SAR ,Do you want to continue the paying? (y/n):
```

- Display the total amont, if the user okay with the total, the user type “y” if No type “n”.

```
Total amount of 165.00 SAR ,Do you want to continue the paying? (y/n): yah
Invalid choice. Please try again.
Total amount of 165.00 SAR ,Do you want to continue the paying? (y/n): y
```

Delivery case Output:

1. Credit card :

- Entering Card information and saving output:

```
Do you want delivery or pickup? (1 for delivery, 2 for pickup): 1
Enter the desired delivery date (YYYY/MM/DD): 2024/06/01
Total amount of 412.50 SAR ,Do you want to continue the paying? (y/n): y
Please select a payment method (1 for card, 2 for cash on delivery/pickup, or 3
for PayPal): 1
Enter the card holder name: lamar bandar felemban
Enter the card number: 1234567890123456
Enter card expiration date (MM/YY): 07/27
Valid expiration date.
Enter the card CVV: 234
Valid CVV: 234
Payment successful!
```

- Order Summery for delivery output:

```
Order Summary:
Order Number: 25949
Name: lamar felemban
Phone: 0555779488
Location: Makkah
Delivery/Pickup: delivery
Delivery Date: 2024/06/01
Flower: 10 x Iris (41.25 SAR each)
Wrapping: black with Gold ribbon (25 SAR)
Total Price: 437.50 SAR
Thank you, the delivery representative will contact you to deliver it to your do
or.
```

2. paypal :

```
Please select a payment method (1 for card, 2 for cash on delivery/pickup, or 3
for PayPal): 3
Redirecting to PayPal...
Payment successful!
```

- Order Summery for delivery output:

```
Order Summary:
Order Number: 25949
Name: lamar felemban
Phone: 0555779488
Location: Makkah
Delivery/Pickup: delivery
Delivery Date: 2024/06/01
Flower: 10 x Iris (41.25 SAR each)
Wrapping: black with Gold ribbon (25 SAR)
Total Price: 437.50 SAR
Thank you, the delivery representative will contact you to deliver it to your do
or.
```

3. Cash on delivery:

```
Please select a payment method (1 for card, 2 for cash on delivery/pickup, or 3
for PayPal): 2
Please pay with cash on delivery/pickup and use the folowing order Number [5154]
 to pickup your bouquet.
```

- Order Summery for delivery output:

```
Order Summary:
Order Number: 25949
Name: lamar felemban
Phone: 0555779488
Location: Makkah
Delivery/Pickup: delivery
Delivery Date: 2024/06/01
Flower: 10 x Iris (41.25 SAR each)
Wrapping: black with Gold ribbon (25 SAR)
Total Price: 437.50 SAR
Thank you, the delivery representative will contact you to deliver it to your do
or.
```

Pickup case Output:

1. Credit card :

- Entering Card information and saving output:

```
Please select a payment method (1 for card, 2 for cash on delivery/pickup, or 3
for PayPal): 1
Enter the card holder name: lamar felemban
Enter the card number: 1234567890123456
Enter card expiration date (MM/YY): 07/27
Valid expiration date.
Enter the card CVV: 123
Valid CVV: 123
Payment successful!
Do you want to save this card on our website? (y/n) y
Card information saved.

------------------------------------
Order Summary:
Order Number: 22300
Name: lamar
Phone: 1234567890
Location: Makkah
Delivery/Pickup: Pickup
Pickup Date: 2024/07/01
Flower: 4 x Iris (41.25 SAR each)
Wrapping: Red with Blue ribbon (25 SAR)
Total Price: 190.00 SAR
Your order will be ready for pickup on 2024/07/01  .
```

2. paypal :

```
Please select a payment method (1 for card, 2 for cash on delivery/pickup, or 3
for PayPal): 3
Redirecting to PayPal...
Payment successful!

------------------------------------
Order Summary:
Order Number: 29036
Name: lamar
Phone: 1234567890
Location: Makkah
Delivery/Pickup: Pickup
Pickup Date: 2024/06/01
Flower: 4 x Iris (41.25 SAR each)
Wrapping: Red with White ribbon (25 SAR)
Total Price: 190.00 SAR
Your order will be ready for pickup on 2024/06/01  .
```

3. Cash on pickup:

```
Please select a payment method (1 for card, 2 for cash on delivery/pickup, or 3
for PayPal): 2
Please pay with cash on delivery/pickup and use the folowing order Number [6463]
 to pickup your bouquet.

------------------------------------
Order Summary:
Order Number: 6463
Name: lamar
Phone: 1234567890
Location: Makkah
Delivery/Pickup: Pickup
Pickup Date: 2024/07/01
Flower: 4 x Iris (41.25 SAR each)
Wrapping: Red with Blue ribbon (25 SAR)
Total Price: 190.00 SAR
Your order will be ready for pickup on 2024/07/01  .
```

Conclusion:

In summary, the flower shop software streamlines the purchasing process for customers through intuitive design and efficient functionality. Leveraging Unix shell scripts and GitHub for development and collaboration, our team created a user-friendly platform that enhances the overall customer experience.

References:

1. Unix shell scripting

2. GitHub for teamwork
   https://github.com/lumR19/project

3. Canva for the report
   https://www.canva.com/design/DAGFhgWFMpc/pfwx_hvwqD7IOlK8Qlbcjag/edit