IT - 314: Software Engineering

Project: Canteen Automation Application

Group 11 | Assigned by : Prof. Saurabh Tiwari

Dhirubhai Ambani Institute of Information and Communication Technology
Gandhinagar

RUSHIRAJ BAXI	(201701170)
AXIL VAGHELA	(201701220)
HARSH BHUT	(201701185)
HARSHIL PATEL	(201701177)
ADIT MEHTA	(201701153)
DHAVAL KANTARIA	(201701150)
DHAIRYA MANIYAR	(201701171)
PRIYANK SATASIYA	(201701183)
KUSHAL VIKANI	(201701167)
VAIDIK PATEL	(201701195)
DHARMIN SOLANKI	(201701198)
PAVAN BALDANIYA	(201701196)
ROHAN PRAJAPATI	(201701218)

Table of Contents

1 Introduction 3

2 Classification of Sprints 3

2.1 Sprint 1 3

2.2 Sprint 2 3

2.3 Sprint 3 4

2.4 Sprint 4 4

3 Functional and Non-functional Requirements 4

3.1 Functional Requirements 4

3.2 Non-Functional Requirements 5

3.3 Tools for Development: 5

4 Classification of Sprints with acceptance criteria 6

4.1 Sprint – 1 6

4.2 Sprint - 2 9

4.3 Sprint - 3 12

4.4 Sprint - 4 13

5 Class Diagrams 15

5.1 Sprint – 1 15

5.2 Sprint – 2 16

5.3 Sprint – 3 17

5.4 Sprint – 4 18

6 Activity Diagram 19

7 Deployment Diagram 23

8 Testing 24

8.1 Test Cases (Customer Module) 24

8.2 Test Cases (Canteen Module) 27

9 Unit Testing Code Snippets 29

9.1 Canteen Activity Test 29

9.2 Feedback Activity Test 30

9.3 Item Activity Test 31

9.4 Order Activity Test 32

9.5 User Activity Test 34

10 Minutes of Meetings 35

10.1 Meeting 1 35

10.2 Meeting 2 36

10.3 Meeting 3 36

10.4 Meeting 4 36

10.5 Meeting 5 36

1 Introduction

In today's fast-growing era of automation, almost everything is leaning towards making the existing systems smart and automated. Even many restaurants and food suppliers have chosen to focus on quick preparation and speedy delivery of orders. If we talk about university canteens in particular, nowadays students do not have much time to wait for their order near the counter endlessly until the order is served. These students may have a hectic schedule and limited amount of time for their lunch/dinner. In such situations they wouldn't want to wait too long for their orders. That is where the Canteen Automation system comes to the rescue. The main advantage of online ordering is that it reduces the hassle of both users as well as canteens. The project "Canteen Automation System" enables its users to register online, read and select food items from the e-menu and order food online by just selecting the food that the user wants using the android application. As soon as the order is placed the respective canteen will be notified and they can start preparing the order and notify the user when the order is ready. The project also aims at providing the facility of virtual money which reduces the workload of both users as well as canteens of managing accounts. Thus, the system aims to accelerate customer orders and reduce the latency of delivering an order to the customer.

2 Classification of Sprints

2.1 Sprint 1

- Login for customer and canteen staff
- Sign up for customers
- See canteens
- Browse canteen menu
- Cart for placing order
- Place order
- Add items to menu
- Navigate the canteen's menu

2.2 Sprint 2

- Change availability of canteen and items from menu
- Edit/Drop items from menu
- Order Status
- Pending Order
- Cooking Instructions
- Admin login
- Add money to customer account
- Settlement of account of canteen

2.3 Sprint 3

- Payment Methods
- Contact Info
- Suggestion /Feedback
- Order History
- See canteens

2.4 Sprint 4

- QR Code scanner
- OR Code Reader

3 Functional and Non-functional Requirements

3.1 Functional Requirements

- **User Side Module**.

  - Sign up page (Create an account)
  - Login page
  - Forgot Password
  - List of canteens
  - Navigate the canteen's menu
  - Adding items to the cart
  - Review the current order
  - Add/ Drop items in the cart
  - Add cooking instructions
  - Placing order with two payment options (Cash on Counter/ Virtual Money)
  - Generating order number
  - Order history
  - Contact information of respective canteens.
  - Feedback/ Suggestion.

- **Canteen Side Module**

  • Log in
  • Change availability of the canteen (Accepting/Not accepting current orders)
  • Add/Drop/Edit items to the menu with their prices.
  • Change the availability of items
  • Show pending orders
  • Scan the QR code shown in user's app when he/she picks up the order
  • Once the order is delivered remove it from pending list
  • Order history
  • Show feedbacks
  • Contact admin

- **Admin Side Module**

  • Login
  • Add virtual money to student's account when he pays cash
  • Settle account of Canteen by paying them the difference amount.

3.2 Non-Functional Requirements

• The Canteen Automation System should be more users friendly. The user interface should be kept simple, elegant and uncluttered.

• The project should be so flexible that whenever we want to make changes in it, it can be done easily.

• The system should be robust, maintainable, and reliable.

• The system should be able to handle multiple users at the same time.

3.3 Tools for Development:

- User Interface : XML
- Scripting : JAVA
- Operating System : Any Mobile Device supporting Android
- Database : Firebase
- IDE/Workbench : Android Studio

4 Classification of Sprints with acceptance criteria

4.1 Sprint – 1

- As a customer, I want a sign-up page so that I can create an account.
    - **Scenario:** Customer can successfully create an account.
        - Given that I'm an unregistered customer
          When I open the app for the first time
          Then the system shows a sign-in page
          And the system shows Sign Up option below 'LOGIN' button
          When I click on 'Don't have an account? Sign up here'
          Then the system shows sign up
          And the system shows field to enter Email, Name, Contact Number and password
          When I fill details
          And click 'Sign Up' button
          Then the system shows 'Registering user' flash prompt
          And the system shows the home-page.

- As a customer, I want a login page so that I can log in to my account.
    - **Scenario:** Customer can successfully log into the account.
        - Given that I'm a registered customer
          When I open the app
          Then the system shows a Sign-in page
          And the system shows fields to enter email and password
          When I fill details in the fields
          And I click 'LOGIN' button
          Then the system shows the home page
          And the system shows current balance and list of canteens.

- As a customer, I want to have a list of all available canteens which are accepting orders so that I can choose a canteen to order.
    - **Scenario:** Customer can choose a particular canteen which is available.
        - Given that I'm a registered customer
          When I login to the app
          Then the system shows home page
          And the system shows list of canteens
          And the system shows it's availability under its name
          When I click on any available canteen
          Then the system shows that canteen's page.

- As a customer, I want to have a list of all available canteens which are accepting orders so that I can see the menu of that particular canteen.
    - **Scenario:** Customer can see the list of all available canteens.
        - Given that I'm a registered customer
          When I login to the app
          Then the system shows home page
          And the system shows list of canteens
          And the system shows it's availability under its name.

- As a customer, I want to have the menu with the price of each item of the chosen canteen so that I can see what all things are available and I can order.
    - **Scenario:** Customers can choose a particular available canteen to see its menu and availability and price of food-items.
        - Given that I'm a registered customer
          When I login to the app
          Then the system shows list of canteens
          When I click on any available canteen
          Then the system shows that canteen's page and menu
          And the system shows available food-items in the menu
          And the system shows availability and price of that item under its name.

- As a canteen staff person I want a sign-in page so that I can log in to my account.
    - **Scenario:** Canteen staff member can successfully log into the account.
        - Given that I'm a registered canteen staff member
          When I open the app
          Then the system shows a customer sign-in page
          And the system shows 'Login as a Canteen Staff' option
          When I click on 'Login as a Canteen Staff'
          Then the system shows sign in page
          And the system shows field to enter Email and password
          When I fill details
          And click 'LOGIN' button
          Then the system shows home-page.

- As a canteen staff person, I want to add new items to the menu so that I can expand my menu.
    - **Scenario:** Canteen staff member can successfully add items to menu.
        - Given that I'm a registered canteen staff member
          When I login as a Canteen Staff
          Then the system shows home-page
          When I click '+ADD' button
          Then the system opens 'Enter Item Description' page
          And the system shows Item Name and Item Price fields and a 'SUBMIT' button
          When I fill the details and click the 'SUBMIT' button
          Then the system adds that item in the menu
          And system redirects to menu page.

- As a customer, I want to add the food item to cart so that I can order it.
    - **Scenario:** Customer can add items from a canteen into cart.
        - Given that I'm a registered customer and logged in the app
          When I click on any available canteen
          Then the system shows available food-items in the menu
          When I click on any available item from the menu page
          Then the system opens an item-page containing item's name and price
          And the system shows quantity
          And a '+' and a '-' button to increase and decrease quantity
          When I add desired quantity
          And I click 'SUBMIT' button
          Then the system adds that item into cart
          And the system redirects to the menu page.

- As a customer, I want to view my cart so that I can review my order before placing the order.
    - **Scenario:** Customer can view cart.
        - Given that I'm a registered customer and logged in the app
          When I add items into the cart and click on 'PROCEED TO CART' button
          Then the system shows cart page
          And the system shows Current Balance, details of items and its quantities
          And the system shows fields for Cooking instructions, Payment method and total amount
          And the system shows a 'CHANGE' button beside the Payment Method.

- As a customer, I want to add /drop items from carts so as to change my order.
    - **Scenario:** Customer can view cart and edit items from it.
        - Given that I'm a registered customer and logged in the app
          When I add items to cart and click on 'PROCEED TO CART' button
          Then the system shows cart page
          When I click on any item in the cart
          Then the system redirects to the item-page
          When I change quantity of the item and click on 'SUBMIT' button,
          Then the system updates the item into the cart
          And the system redirects to cart
          When I press back button
          Then the system redirects to menu page
          When I add a different item to cart
          Then the system updates the cart.

- As a customer, I want to place order so that my order is placed to respective canteen.
    - **Scenario:** Customer can place his order.
        - Given that I'm registered customer and logged in the app
          When I check my order in 'PROCEED TO CART' page
          Then the system shows 'PLACE ORDER' button
          When I click that button
          Then my order is placed to the canteen
          And the system will show the order number.

- As a customer, canteen staff person I want the UI to be simple and user-friendly so that I can use the app easily without any prior knowledge of how to use it.
    - **Scenario:** User can easily use the app.
        - Given that I'm a registered user
          When I log into my account
          Then the UI is explicit enough to understand
          And the UI is explicit enough to understand the functionality
          And the UI is easy enough to know what procedures to follow for different tasks.

4.2 Sprint - 2

- As a canteen staff person, I want an availability option so that I can change the availability of the canteen whether it is currently accepting an order or closed.
    - **Scenario:** Canteen staff member can change the availability mode.
        - Given that I'm a registered canteen staff member
          When I login as a Canteen Staff
          Then the system shows home-page
          And the system shows Availability, Current balance and menu and an additional '+ADD' button at the bottom right corner.
          And the system shows the 'CHANGE AVAILABILITY' button.
          When I click the 'CHANGE AVAILABILITY' button
          Then the system changes the availability mode.

- As a canteen staff person, I want to change the availability of a particular item so that I can change it to not available when I am short of a particular item.
    - **Scenario:** Canteen staff member can successfully change the availability of any item.
        - Given that I'm a registered canteen staff member
          When I login as a Canteen Staff
          Then the system shows home-page
          When I click on any item
          Then the system shows Edit Item Description page
          And the system shows Item Name, Item Price and Availability fields
          And the system shows 'CHANGE AVAILABILITY', 'SUBMIT' and 'DELETE' buttons
          When I click on 'CHANGE AVAILABILITY' button
          Then the system changes the availability for that item.

- As a canteen staff person, I want to delete an item from the menu so that I can remove it if I am no longer making an item.
    - **Scenario:** Canteen staff member can successfully delete any item.
        - Given that I'm a registered canteen staff member
          When I login as a Canteen Staff
          Then the system shows home-page
          When I click on any item
          Then the system shows Edit Item Description page
          And the system shows 'CHANGE AVAILABILITY', 'SUBMIT' and 'DELETE' buttons
          When I click on 'DELETE' button
          Then the system deleted that item from the menu
          And the system redirects to menu-page.

- As a customer, I want to see my order status so that I can see what my pending orders are.
    - **Scenario:** Customer can see his order status.
        - Given that I'm a registered customer
          When I login as a customer
          Then the system shows home-page and system shows options tab on the top right corner
          When I click on option tab
          Then the system shows Order Status, Order History and Logout options in a pull down list
          When I click on Order Status option
          Then the system shows all my orders.

- As a canteen staff person, I want to see all the pending orders so that I can see what all items I need to prepare to fulfil all orders.
    - **Scenario:** Canteen staff members can successfully check pending order details.
        - Given that I'm a registered canteen staff member
          When I login as a Canteen Staff
          Then the system shows home-page
          And the system shows options tab on the top right corner
          When I click on options tab
          Then the system shows Order Status, Order History and Logout options in a pull down list
          When I click on Order Status option
          Then the system opens the Pending Orders page.

- As a canteen staff person, I want to have the contact details of the customer so that I can contact the customer in case of some query/ inquiry.
    - **Scenario:** Canteen staff members can successfully contact customer.
        - Given that I'm a registered canteen staff member
          When I login as a Canteen Staff
          Then the system shows home-page
          When I click on Order Status option
          Then the system opens Pending Orders page
          And the system shows number of pending orders and list of pending orders
          When I click on any order
          Then the system opens Order Details page
          And the system shows Order no, Customer ID, cooking instructions, total amount, payment status and 'DELIVER ORDER' button.

- As a customer, I want to add cooking instructions so as to make food according to my taste and requirements.
    - **Scenario:** Customer can add cooking instructions.
        - Given that I am in a role of logged-in user
          When I open proceed to cart page
          Then this page shows me the cooking instructions text-field
          When I fill this field with my instructions
          And I click on the 'PLACE ORDER' button
          Then the system notes my cooking instructions.

- As admin, I want to add virtual money to a customer's account so that whenever he pays me some money I can add money to his account which he can use as virtual money to make payment online.
    - **Scenario:** Admin has right to add money to student account.
        - Given that I am in a role of logged-in admin
          When a student wants to add virtual money to his account
          Then I will enter student-ID
          And I enter the amount of money which student wants
          When I click on the 'SUBMIT' button
          Then the system adds money to student's current balance.

- As admin, I can settle the balance of the canteen so that the canteen will not be in loss.
    - **Scenario:** Admin will settle the balance of the canteen.
        - Given that I am in a role of logged-in admin
          When the canteen wants to settle the balance
          Then I will enter a valid ID of that canteen and amount
          When I click the 'GIVE MONEY' button
          Then whatever amount I entered that amount is increased in the canteen's balance
          And the system shows it.

4.3 Sprint - 3

- As a customer, I want to choose the payment method (online payment/ cash on the counter) so as to pay for my order.
    - **Scenario:** Customer can choose payment option.
        - Given that I am in a role of logged-in customer
          When I open proceed to cart page
          Then this page shows me the payment option
          When I choose payment option
          And I click on the PLACE ORDER button
          And I chose online option
          Then the system decreases my current balance.

- As a customer, I want to have contact information of respective canteens so that I can contact the canteen for inquiry.
    - **Scenario:** Customer can contact with canteen for inquiry.
        - Given that I am in a role of logged-in customer
          When I open a particular canteen page
          The system shows contact information for that canteen.

- As a customer, I want to have a suggestion box so that I can give suggestions to canteen vendors on how to improve.
    - **Scenario:** Customer can give suggestions to canteen.
        - Given that I am in a role of logged-in customer
          When I open a particular canteen page
          Then the system shows feedback section
          When I fill this section according to my opinion
          And I click on SUBMIT button
          Then the system shows "feedback submitted" flash message
          And the system clears the field of feedback section.

- As a canteen staff person, I want developer's contact information so that I can complain to the developer about bugs or give some suggestions.
    - **Scenario:** Canteen staff member can contact with developer for query.
        - Given that I am in a role of logged-in canteen staff member
          When there is a bug or I want to give some suggestion
          The system shows contact information of the developer.

- As a customer, I want to have an order history so that I can see what I bought in the past from which canteen.
    - **Scenario:** Customer can see his order history.
        - Given that I am in a role of logged-in customer
          When I want to know about my past orders
          The system shows an option of order history
          When I will click that button
          Then the system shows me my past history of orders according to canteens.

- As a canteen staff person, I want to have order history so that I can see which products I sold and total sales of a particular day.
    - **Scenario:** Canteen staff member can see his canteen's order history.
        - Given that I am in a role of logged-in canteen staff member
          When I want to know the total sale of a day
          The system shows an option of order history
          When I click the order history button
          Then the app shows me the whole history which I sold.

4.4 Sprint - 4

- As a customer, I want a QR code of my placed order so that I can show that to the counter while picking up my order to reduce the interaction time.
    - **Scenario:** Customer can generate QR code for his order.
        - Given that I am in a role of logged-in customer
          When my order is submitted successfully
          Then the system shows me an order number
          And the system shows an option of 'Generate QR code'
          When I will click that button
          Then the system will generate unique QR code for my order.

- As a canteen staff person, I want to scan the QR code from the customer's device so that I can view all the things which I have to give him and mark it as delivered/ finished.
    - **Scenario:** Canteen staff member can scan QR code of customer's order.
        - Given that I am in a role of logged-in canteen staff member
          When customer comes for his order
          And shows QR code for his order
          Then the system shows me an option of scan QR code
          When I click on that button
          Then the system opens a QR Code scanner.

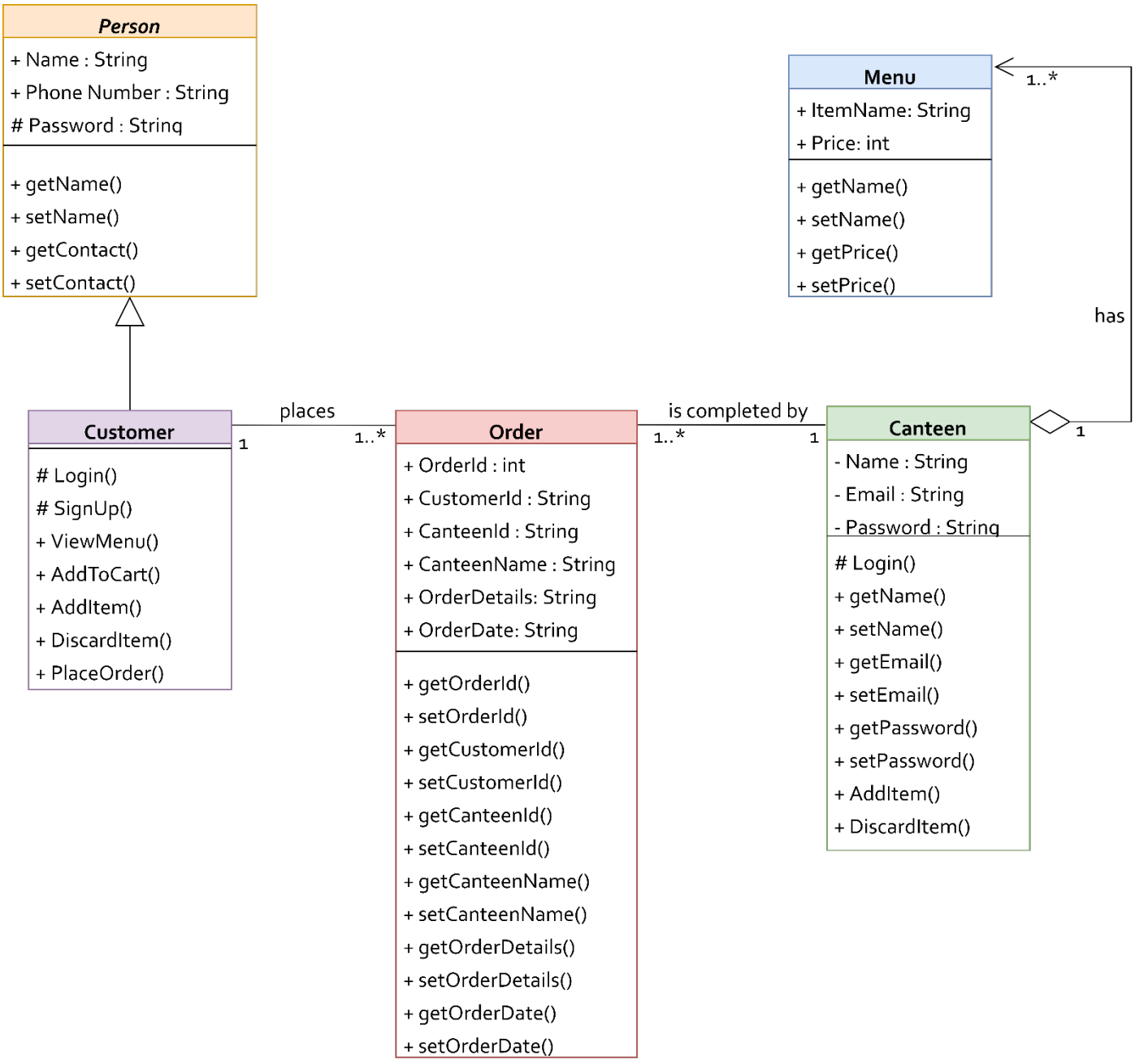
- As an owner I want the project to be flexible so that we can expand it.
    - **Scenario:** Owner can expand the project on different scales and platforms.
        - Given that I'm the owner of the app
          When I open the app project on developer's platform
          Then the system's functionalities are easy to implement
          And the system project is scalable on different operating systems
          And the system project is scalable on different types of devices
          And the system project is easy to change and edit.

- As an owner I want the project to be robust**,** maintainable and reliable so that I can update and make my project run without crashing.
    - **Scenario:** Owner can successfully run the app without crashing.
        - Given that I'm the owner of the app
          When I open the app
          Then the system is strong enough to handle multiple queries
          And the system is smooth enough easily transact through different pages
          And the system database can easily store data
          And the system is small enough to take as minimum RAM as possible.

- As an owner I want the app to handle multiple customers so that I can earn profit out of it and make my app popular.
    - **Scenario:** Owner can successfully attract many customers and serve them simultaneously.
        - Given that I'm the owner of the app
          When I open the app
          Then the system is strong enough to handle multiple queries
          And the system UI is user-friendly and attractive
          And the system database is large and strong enough to save data.

5 Class Diagrams

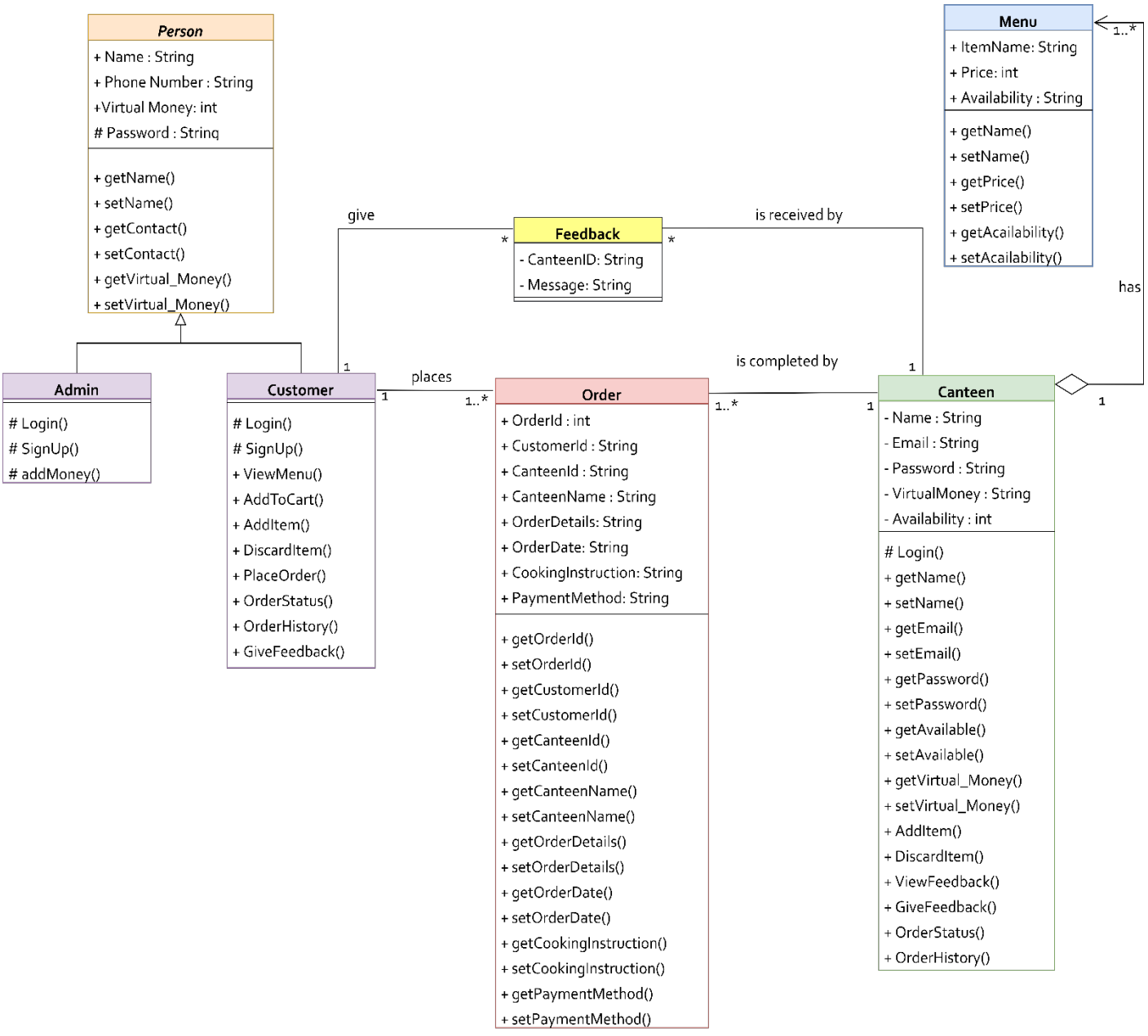
5.1 Sprint – 1



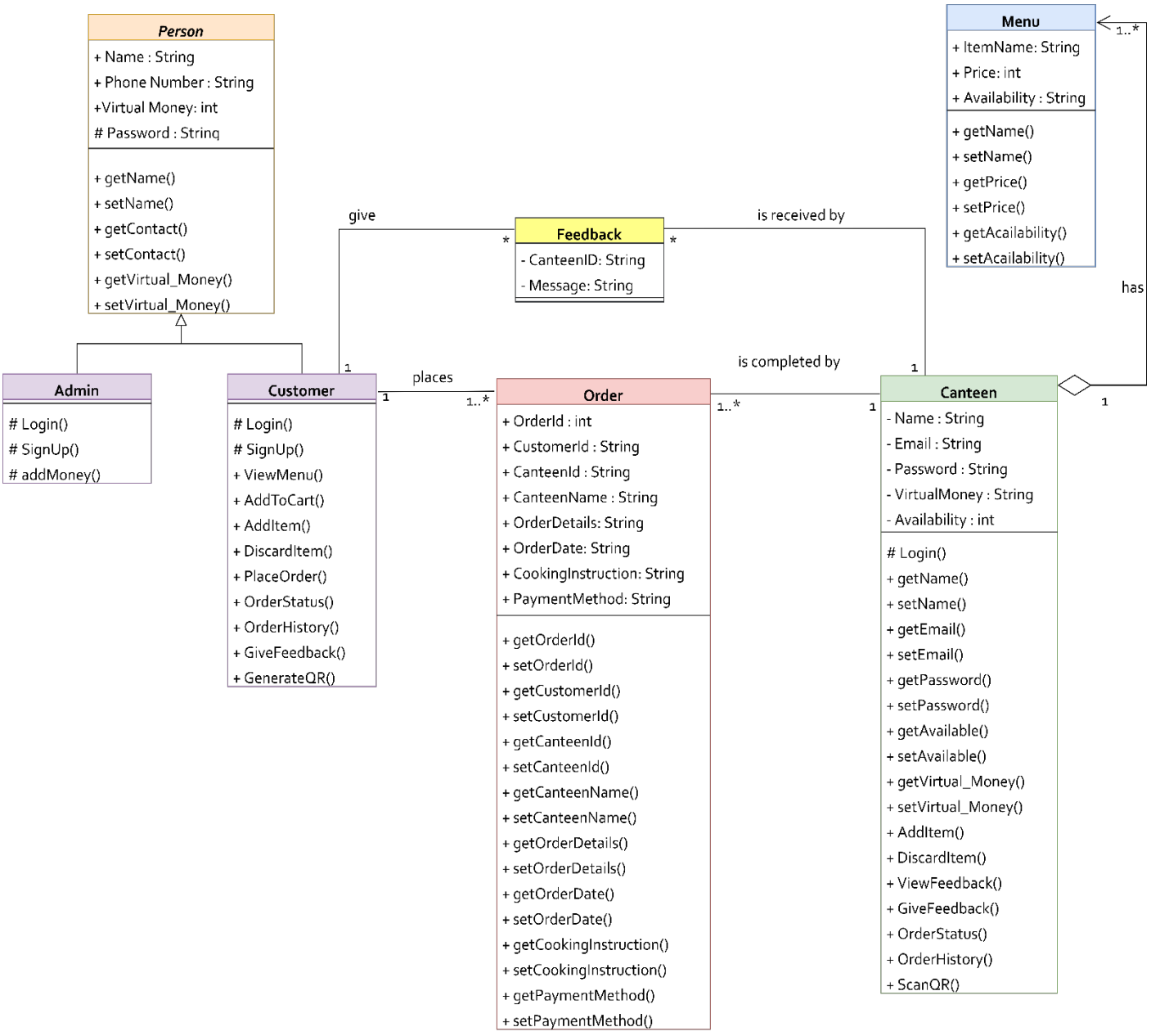
5.2 Sprint – 2

Person
+ Name : String
+ Phone Number : String
+Virtual Money: int
Password : String
+ getName()
+ setName()
+ getContact()
+ setContact()
+ getVirtual_Money()
+ setVirtual_Money()
Menu
+ ItemName: String
+ Price: int
+ Availability : String
+ getName()
+ setName()
+ getPrice()
+ setPrice()
+ getAcailability()
+ setAcailability()
1..*
has
Admin
Login()
SignUp()
addMoney()
Customer
Login()
SignUp()
+ ViewMenu()
+ AddToCart()
+ AddItem()
+ DiscardItem()
+ PlaceOrder()
places
1
1..*
Order
+ OrderId : int
+ CustomerId : String
+ CanteenId : String
+ CanteenName : String
+ OrderDetails: String
+ OrderDate: String
+ CookingInstruction: String
+ getOrderId()
+ setOrderId()
+ getCustomerId()
+ setCustomerId()
+ getCanteenId()
+ setCanteenId()
+ getCanteenName()
+ setCanteenName()
+ getOrderDetails()
+ setOrderDetails()
+ getOrderDate()
+ setOrderDate()
+ getCookingInstruction()
+ setCookingInstruction()
is completed by
1..*
1
Canteen
- Name : String
- Email : String
- Password : String
- VirtualMoney : String
- Availability : int
Login()
+ getName()
+ setName()
+ getEmail()
+ setEmail()
+ getPassword()
+ setPassword()
+ getAvailable()
+ setAvailable()
+ getVirtual_Money()
+ setVirtual_Money()
+ AddItem()
+ DiscardItem()
+ OrderStatus()
1

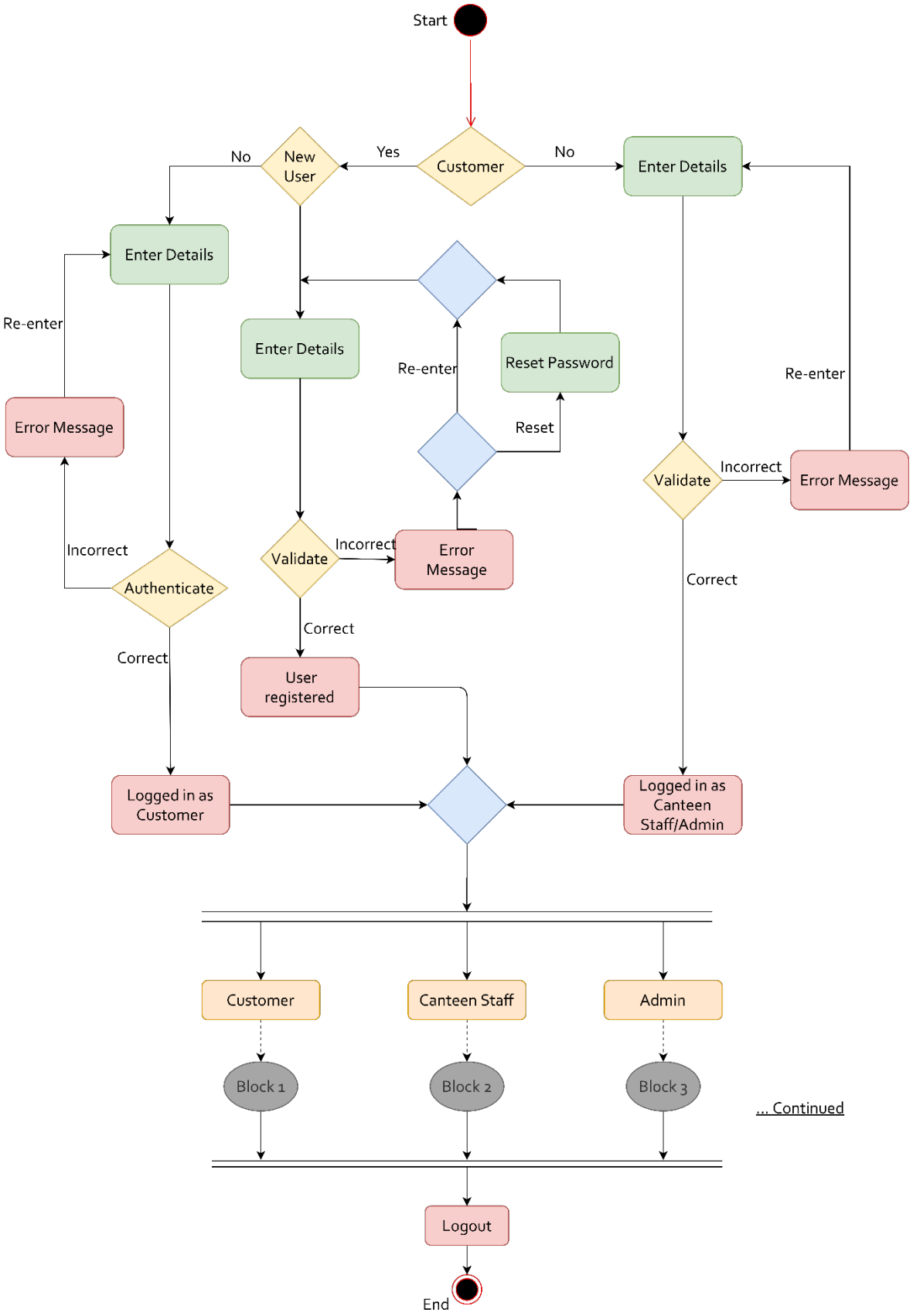
5.3 Sprint – 3

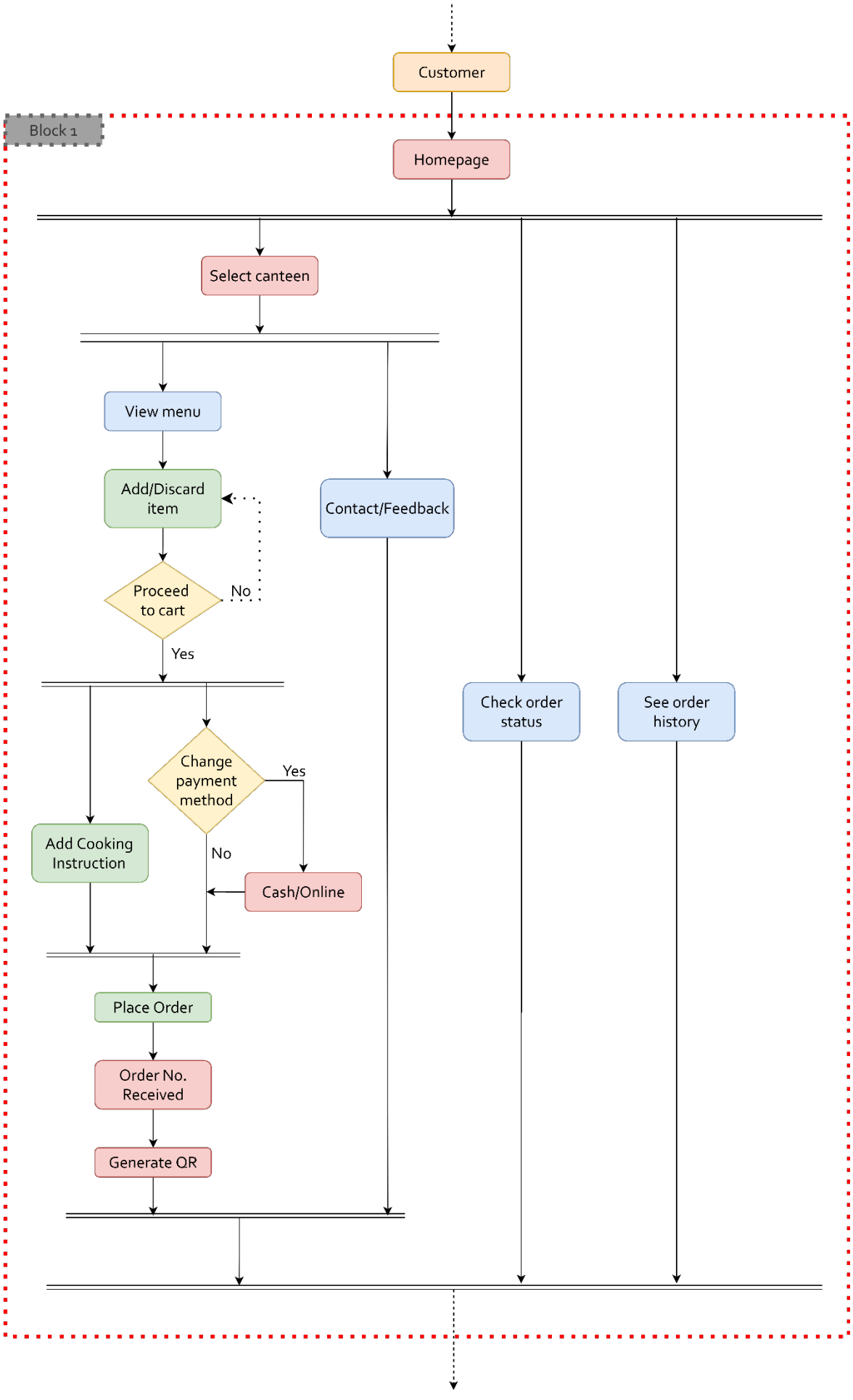

Person
+ Name : String
+ Phone Number : String
+Virtual Money: int
Password : String
+ getName()
+ setName()
+ getContact()
+ setContact()
+ getVirtual_Money()
+ setVirtual_Money()
Menu
+ ItemName: String
+ Price: int
+ Availability : String
+ getName()
+ setName()
+ getPrice()
+ setPrice()
+ getAcailability()
+ setAcailability()
1..*
give
Feedback
- CanteenID: String
- Message: String
is received by
has
Admin
Login()
SignUp()
addMoney()
Customer
Login()
SignUp()
+ ViewMenu()
+ AddToCart()
+ AddItem()
+ DiscardItem()
+ PlaceOrder()
+ OrderStatus()
+ OrderHistory()
+ GiveFeedback()
places
is completed by
Order
+ OrderId : int
+ CustomerId : String
+ CanteenId : String
+ CanteenName : String
+ OrderDetails: String
+ OrderDate: String
+ CookingInstruction: String
+ PaymentMethod: String
+ getOrderId()
+ setOrderId()
+ getCustomerId()
+ setCustomerId()
+ getCanteenId()
+ setCanteenId()
+ getCanteenName()
+ setCanteenName()
+ getOrderDetails()
+ setOrderDetails()
+ getOrderDate()
+ setOrderDate()
+ getCookingInstruction()
+ setCookingInstruction()
+ getPaymentMethod()
+ setPaymentMethod()
Canteen
- Name : String
- Email : String
- Password : String
- VirtualMoney : String
- Availability : int
Login()
+ getName()
+ setName()
+ getEmail()
+ setEmail()
+ getPassword()
+ setPassword()
+ getAvailable()
+ setAvailable()
+ getVirtual_Money()
+ setVirtual_Money()
+ AddItem()
+ DiscardItem()
+ ViewFeedback()
+ GiveFeedback()
+ OrderStatus()
+ OrderHistory()

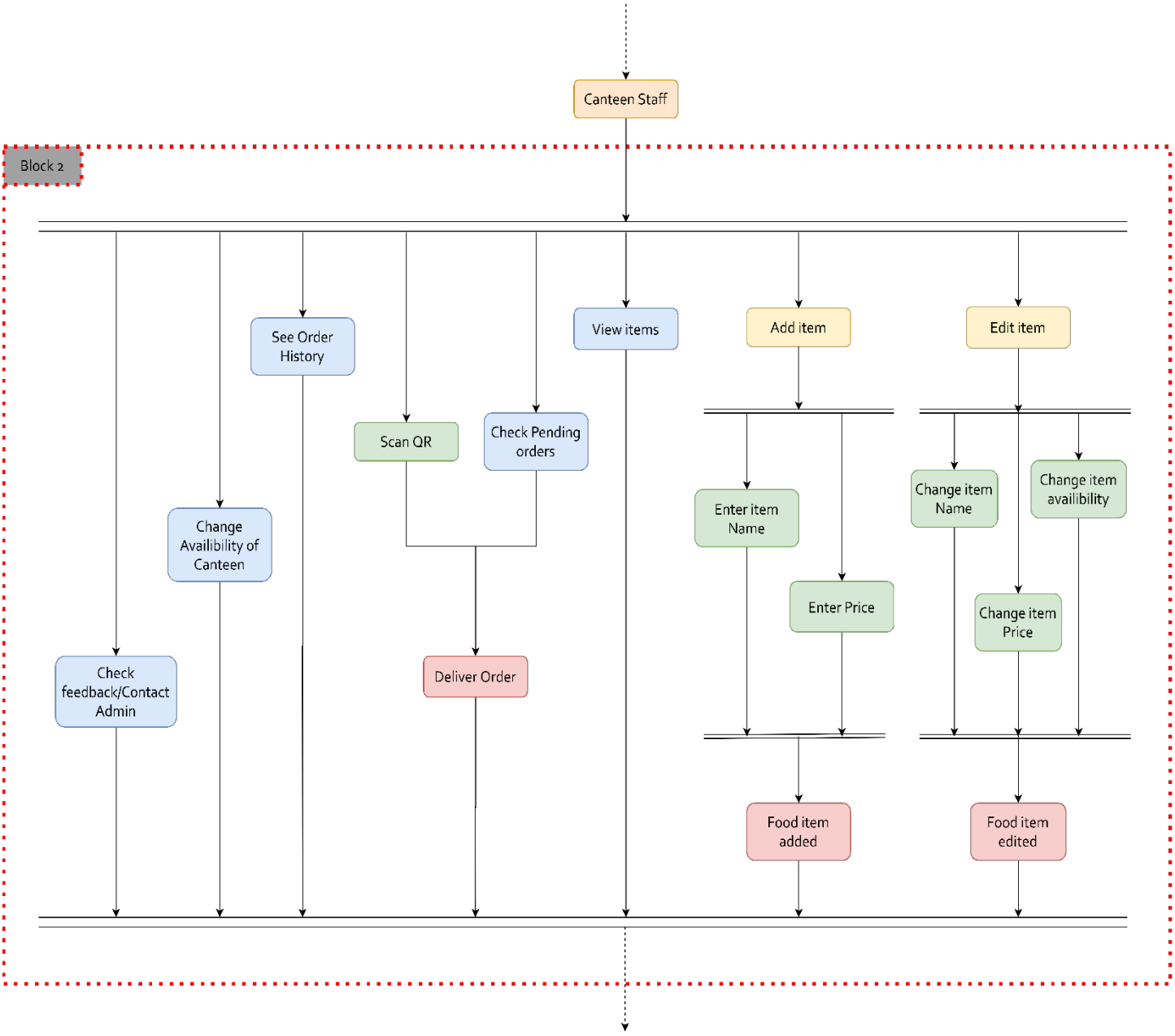
5.4 Sprint – 4

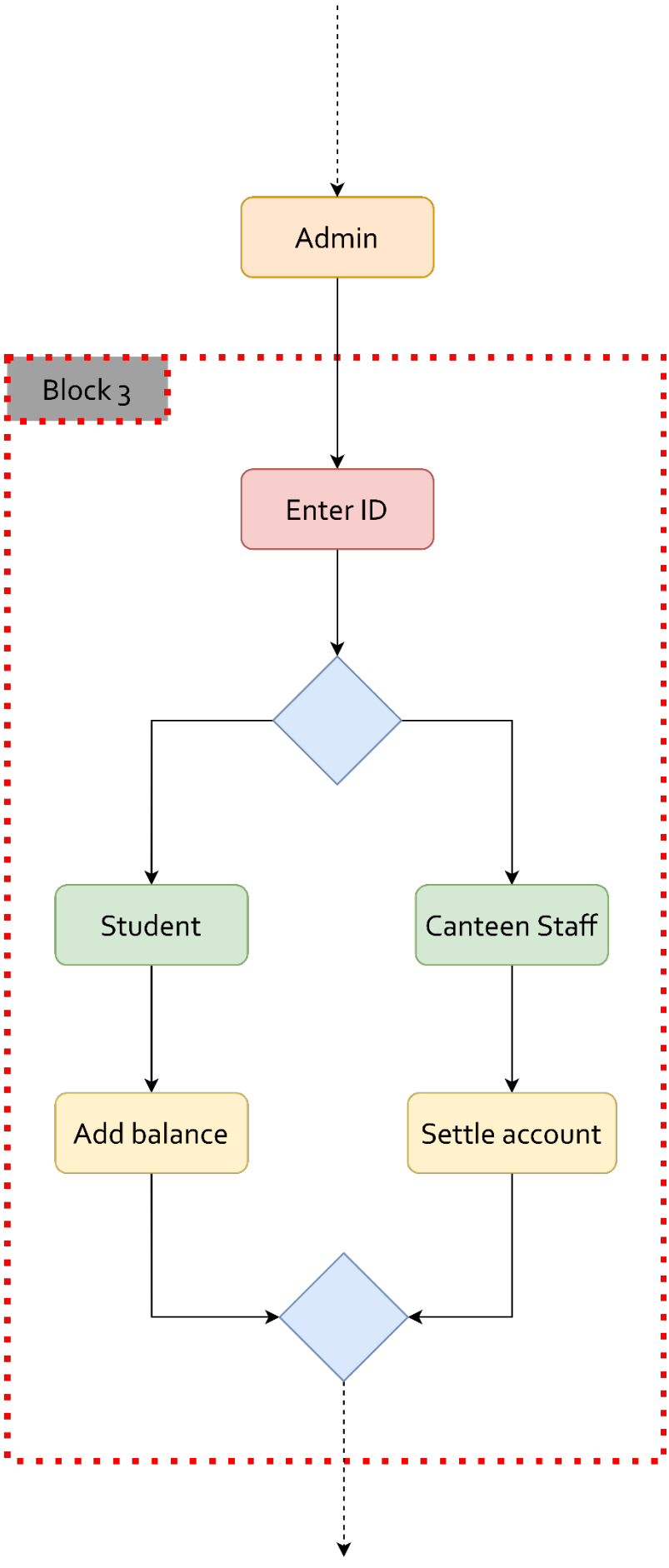

Person
+ Name : String
+ Phone Number : String
+Virtual Money: int
Password : String
+ getName()
+ setName()
+ getContact()
+ setContact()
+ getVirtual_Money()
+ setVirtual_Money()
Menu
+ ItemName: String
+ Price: int
+ Availability : String
+ getName()
+ setName()
+ getPrice()
+ setPrice()
+ getAcailability()
+ setAcailability()
1..*
has
give
Feedback
- CanteenID: String
- Message: String
is received by
Admin
Login()
SignUp()
addMoney()
Customer
Login()
SignUp()
+ ViewMenu()
+ AddToCart()
+ AddItem()
+ DiscardItem()
+ PlaceOrder()
+ OrderStatus()
+ OrderHistory()
+ GiveFeedback()
+ GenerateQR()
places
Order
+ OrderId : int
+ CustomerId : String
+ CanteenId : String
+ CanteenName : String
+ OrderDetails: String
+ OrderDate: String
+ CookingInstruction: String
+ PaymentMethod: String
+ getOrderId()
+ setOrderId()
+ getCustomerId()
+ setCustomerId()
+ getCanteenId()
+ setCanteenId()
+ getCanteenName()
+ setCanteenName()
+ getOrderDetails()
+ setOrderDetails()
+ getOrderDate()
+ setOrderDate()
+ getCookingInstruction()
+ setCookingInstruction()
+ getPaymentMethod()
+ setPaymentMethod()
is completed by
Canteen
- Name : String
- Email : String
- Password : String
- VirtualMoney : String
- Availability : int
Login()
+ getName()
+ setName()
+ getEmail()
+ setEmail()
+ getPassword()
+ setPassword()
+ getAvailable()
+ setAvailable()
+ getVirtual_Money()
+ setVirtual_Money()
+ AddItem()
+ DiscardItem()
+ ViewFeedback()
+ GiveFeedback()
+ OrderStatus()
+ OrderHistory()
+ ScanQR()

6 Activity Diagram

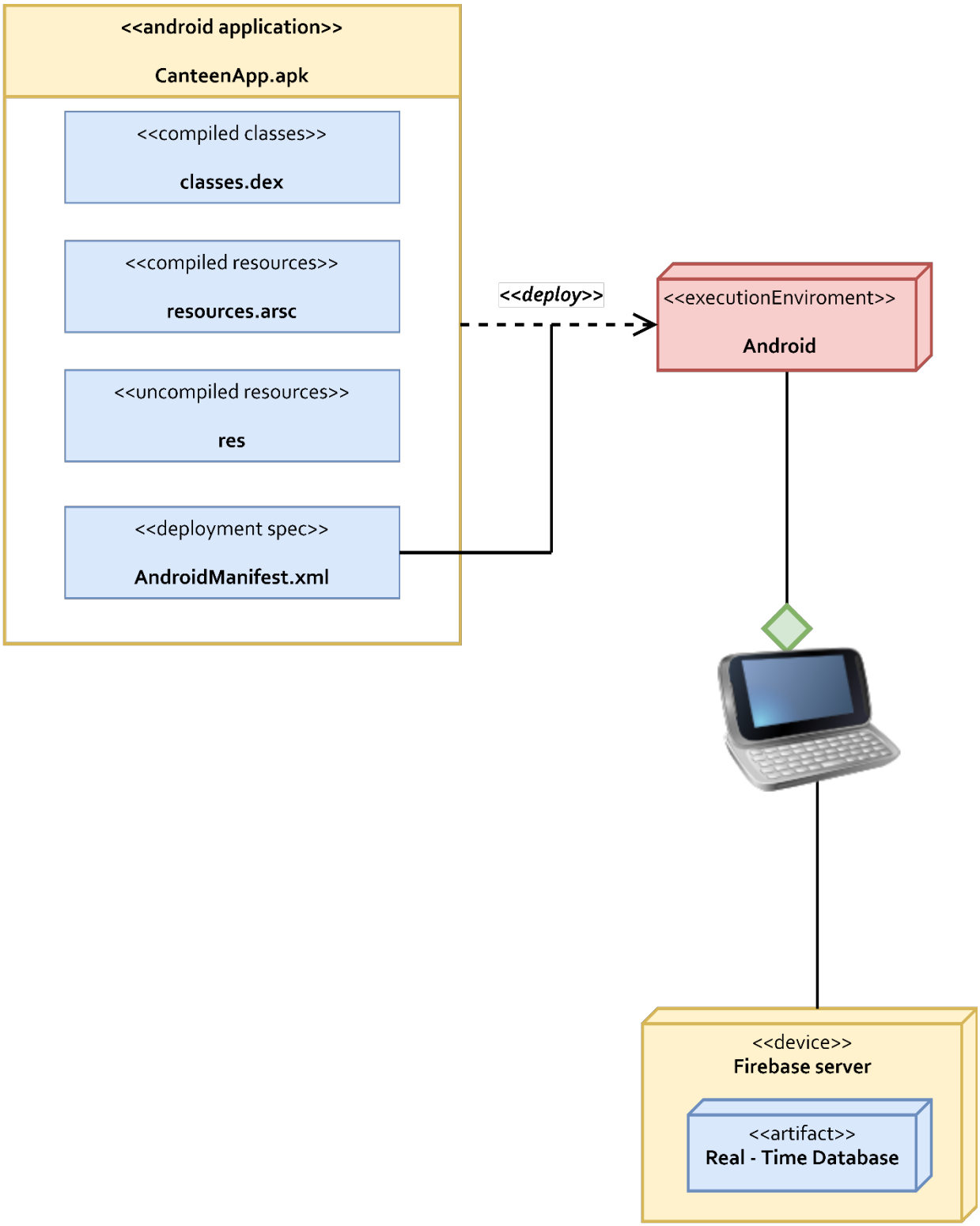



Customer
Block 1
Homepage
Select canteen
View menu
Add/Discard item
Proceed to cart
No
Yes
Contact/Feedback
Check order status
See order history
Change payment method
Yes
No
Add Cooking Instruction
Cash/Online
Place Order
Order No. Received
Generate QR


Canteen Staff
Block 2
See Order History
View items
Add item
Edit item
Scan QR
Check Pending orders
Change item Name
Change item availibility
Change Availibility of Canteen
Enter item Name
Enter Price
Change item Price
Check feedback/Contact Admin
Deliver Order
Food item added
Food item edited


Admin
Block 3
Enter ID
Student
Canteen Staff
Add balance
Settle account

7 Deployment Diagram



8 Testing

8.1 Test Cases (Customer Module)

Test Scenario Group	Test Case Id	Test Case Description	Test Priority	Pre-requisite	Action	Inputs	Expected Output	Actual Output	Test Results
SPRINT - 1									
Log in	1.01	Log in - Positive test case	High	User must have an account	Enter correct email and password	Email: 201701xxx@gmail.com Password:xxxxx	Home page	Home page	Pass
Sign Up	1.02	Sign up - Positive test case	High	NA	Enter non-used email and valid password	Enter all the required details	Sign up Success	Sign up Success	Pass
Log in	1.03	Log in - Negative test case	High	NA	Incorrect email or password	Email: 201701xxx@gmail.com Password:xxxxx	The Email address or password is not match any account	The Email address or password is not match any account	Pass
Sign Up	1.04	Sign up - Negative test case	High	NA	Enter used email or invalid password	Enter details along with invalid password or used email	Email or Password is not valid	Email or Password is not valid	Pass
Home Page	1.05	User name must be displayed	High	User must be logged in	--	--	User Name on the top of the screen	User Name on the top of the screen	Pass
Home Page	1.06	List of available canteen	High	User must be logged in	--	--	All available canteen must be displayed	All canteen displayed with its status	Pass
Available items	1.07	List of available items (If canteen is available)	High	At least one canteen is available	select any available canteen	--	All items should be displayed	All items should be displayed	Pass
Available items	1.08	List of available items (If canteen is not available)	High	At least one canteen is unavailable	select any unavailable canteen	--	No items should be displayed	No items should be displayed	Pass
Add food item to Cart	1.09	Add item and set Qty to more than 0	High	Items must be displayed	Select item, Qty and hit submit	Set Qty more than 0	Home page	Home page	Pass

Add food item to Cart	1.10	Add item and set Qty to 0	High	Items must be displayed	Select item, Qty and hit submit	Set Qty to 0	Home page	Home page	Pass
Cart Page	1.11	Total cost	High	Must be on cart page	--	--	Total Cost	Total cost	Pass
Cart Page	1.12	Enter NO items and open cart	High	No items should be selected	Hit Cart Button	--	Message must be displayed	Message must be displayed	Pass
Cart Page	1.13	Change Qty of selected item	High	At least one item should be selected	Hit Cart Button	--	Display all items whose Qty is greater than 0 with name and Qty	Display all items whose Qty is greater than 0 with name and Qty	Pass
Drop food item to Cart	1.14	Drop item which is Added(Minimum one should be there at the end)	High	At least one item should be in cart	Set Qty to 0 of some items	--	That item must not be displayed in the cart page	That item is not available in the cart page	Pass
Drop food item to Cart	1.15	Drop all items	High	At least one item should be in cart	Set Qty to 0 of all items	--	Item list	Item list	Pass
Successfully placed	1.16	I must get an conformation	High	At least one item should present in the cart	Hit Submit Button	--	There should be confirmation message	Get an order id	Pass
Logout	1.17	User can be logout from his account	High	User must be logged in	Click on option button and select logout	--	Sign in page	Sign in page	Pass

SPRINT - 2

Cart Page	2.01	Add Cooking Instruction	High	At least one item should present in cart	Give instruction and hit submit button	Give input instruction	There should be conformation message	Get an order id	Pass

Home page	2.02	To know about ongoing order	High	Must be logged in	Click on option button and select order status	--	Must be show ongoing order if any	I can see total number of pending order, order number and canteen name	Pass

SPRINT - 3

Cart page	3.01	I can select payment method	High	At least one item should present in the cart	Select payment method as cash or online payment	--	Payment method must be change	Displaying payment method	Pass
Cart page	3.02	If balance is not sufficient then I could not pay online	High	Balance must be low than total cost	Select payment method to online	--	Option should be unavailable to me	I cannot select online payment	Pass
Available items	3.03	I should give feedback to respective canteen	High	User should be in canteen's item page	Select option and click on feedback	Enter instructions in given space	Feedback submitted successfully	Feedback submitted successfully	Pass
Home page	3.04	Order History	High	Must be logged in	Select option and click on Order History	--	I should see all my previous order	I can see all my previous order with total cost and canteen name	Pass

SPRINT - 4

Get QR	4.01	I should get QR code of my ongoing order	High	There should be at least one pending order	Go to 'order status' and Click on 'Generate QR'	--	Display QR code	I can see QR code	Pass

8.2 Test Cases (Canteen Module)

Test Scenario Group	Test Case Id	Test Case Description	Test Priority	Pre-requisite	Action	Inputs	Expected Output	Actual Output	Test Results
SPRINT - 1									
Log in	1.01	Log in - Positive test case	High	NA	Incorrect email or password	Email: xxx@ gmail.com Password:x xxxx	Home Page	Home Page	Pass
Log in	1.02	Log in - Negative test case	High	NA	Enter used email or invalid password	Enter details along with invalid password or used email	Email or Password is not valid	Email or Password is not valid	Pass
Home Page	1.03	List of items must be displayed	High	User must be logged in	--	--	List of all items	List of all items	Pass
Add item	1.04	Add new item to the menu	High	Item shouldn't added already	Select add item option	enter all details of food item	New item is added successfully	New item is added successfully	Pass
SPRINT - 2									
Change Availability	2.01	Make item available to menu	High	User must be logged in	Click on 'CHANGE AVAILABILITY ' button	--	item's availability changed to 'Available'	item's availability changed to 'Available'	Pass
Change Availability	2.02	Make item unavailable to menu	High	User must be logged in	Click on 'CHANGE AVAILABILITY ' button	--	item's availability changed to 'Not Available'	item's availability changed to 'Not Available'	Pass

Delete Item	2.03	Delete item from menu	High	User must be logged in	Click on 'DELETE' button	--	item's deleted successfully	item's deleted successfully	Pass
Check status	2.04	Check pending orders	High	User must be logged in	Click on 'Order status' button	--	List of pending orders	List of pending orders	Pass
Contact Customer	2.05	Contact Customer to pick up order	Low	User must be logged in	Click on 'Order Details' button	--	it shows order and customer details	it shows order and customer details	Pass
Add money	2.06	Add virtual Money to customer's account	High	admin must be logged in	Click on 'Add Money' button	enter the amount of money and customer id	Money balance should updated	Money balance is updated	Pass
SPRINT - 3									
Contact Developer	3.01	Complain to the developer about bugs or give some suggestions.	High	User must be logged in	Click on 'Contact or feedback' button	--	shows Contact details	shows Contact details	Pass
Check History	3.02	Check previous order history	Low	User must be logged in	Click on 'Order history' button	--	List of previous orders	List of previous orders	Pass
SPRINT - 4									
Scan QR	4.01	Scan the QR code from the customer's device to mark it as delivered/ finished.	Low	Customer should have valid QR code	Click on 'Scan QR' button	mark as 'delivered'	Order Delivered successfully	Order Delivered successfully	Pass

9 Unit Testing Code Snippets

Entire unit testing is done by using ***Junit*** Framework.

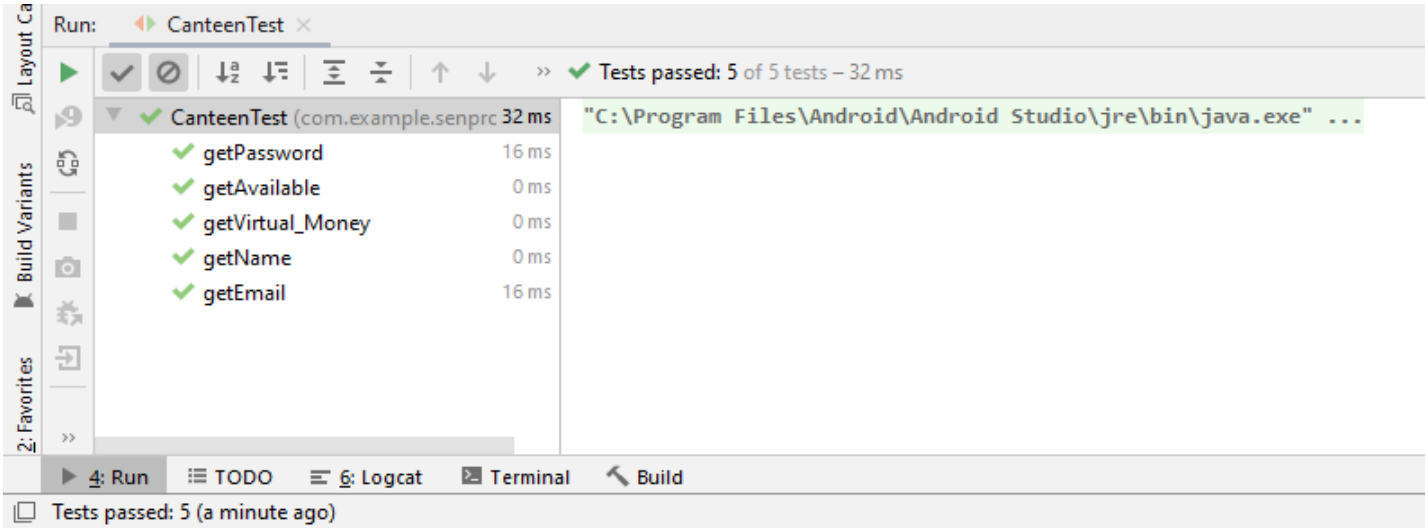
```
package com.example.senproject;

import org.junit.Test;
import static org.hamcrest.CoreMatchers.is;
import static org.junit.Assert.*;
```

9.1 Canteen Activity Test

- **Code:**

```
public class CanteenTest {

    private Canteen c;

    @Test
    public void getName() {
        c = new Canteen("1","HoneyOne@gmail.com","HoneyOne_C1","Canteen@1","0");
        assertThat("HoneyOne_C1",is(c.getName()));
    }

    @Test
    public void getEmail() {
        c = new Canteen("1","HoneyOne@gmail.com","HoneyOne_C1","Canteen@1","0");
        assertThat("HoneyOne@gmail.com",is(c.getEmail()));
    }

    @Test
    public void getPassword() {
        c = new Canteen("1","HoneyOne@gmail.com","HoneyOne_C1","Canteen@1","0");
        assertThat("Canteen@1",is(c.getPassword()));
    }

    @Test
    public void getAvailable() {
        c = new Canteen("1","HoneyOne@gmail.com","HoneyOne_C1","Canteen@1","0");
        assertThat("1",is(c.getAvailable()));
    }

    @Test
    public void getVirtual_Money() {
        c = new Canteen("1","HoneyOne@gmail.com","HoneyOne_C1","Canteen@1","0");
        assertThat("0",is(c.getVirtual_Money()));
    }
}
```

- **Screenshot:**

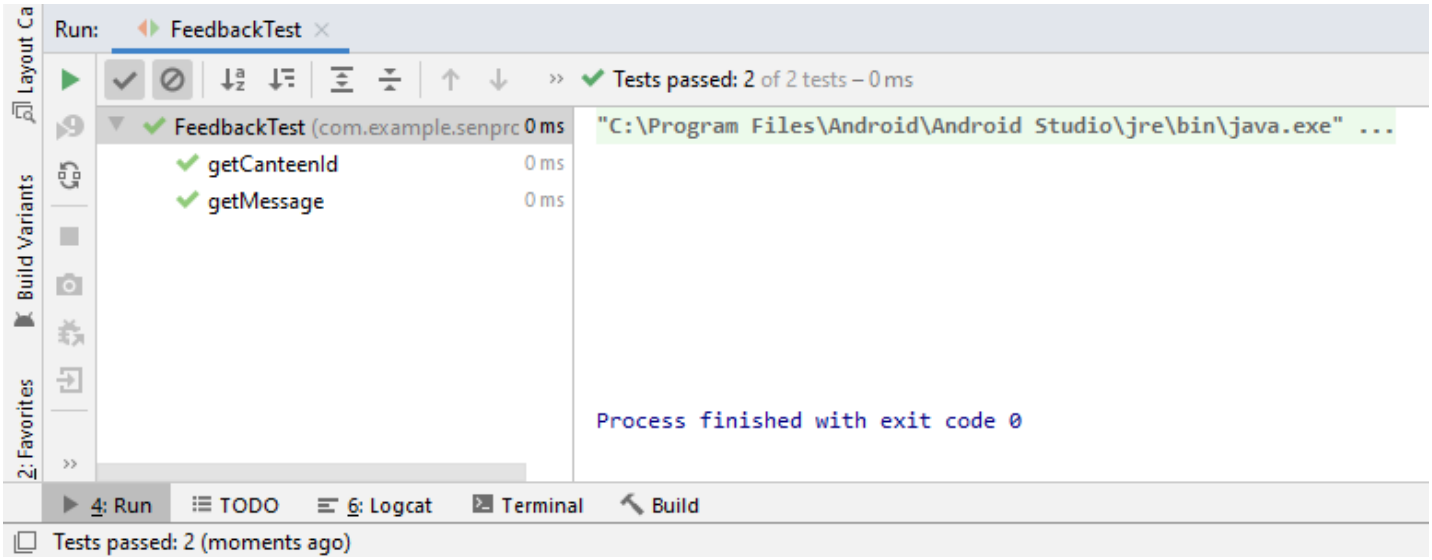


9.2 Feedback Activity Test

- **Code:**

```
public class FeedbackTest {

    private Feedback feedback_obj;
    @Test
    public void getCanteenId() {
        feedback_obj = new Feedback("1","Great Canteen");
        assertThat("1",is(feedback_obj.getCanteenId()));
    }

    @Test
    public void getMessage() {
        feedback_obj = new Feedback("1","Great Canteen");
        assertThat("Great Canteen",is(feedback_obj.getMessage()));
    }
}
```

- **Screenshot:**

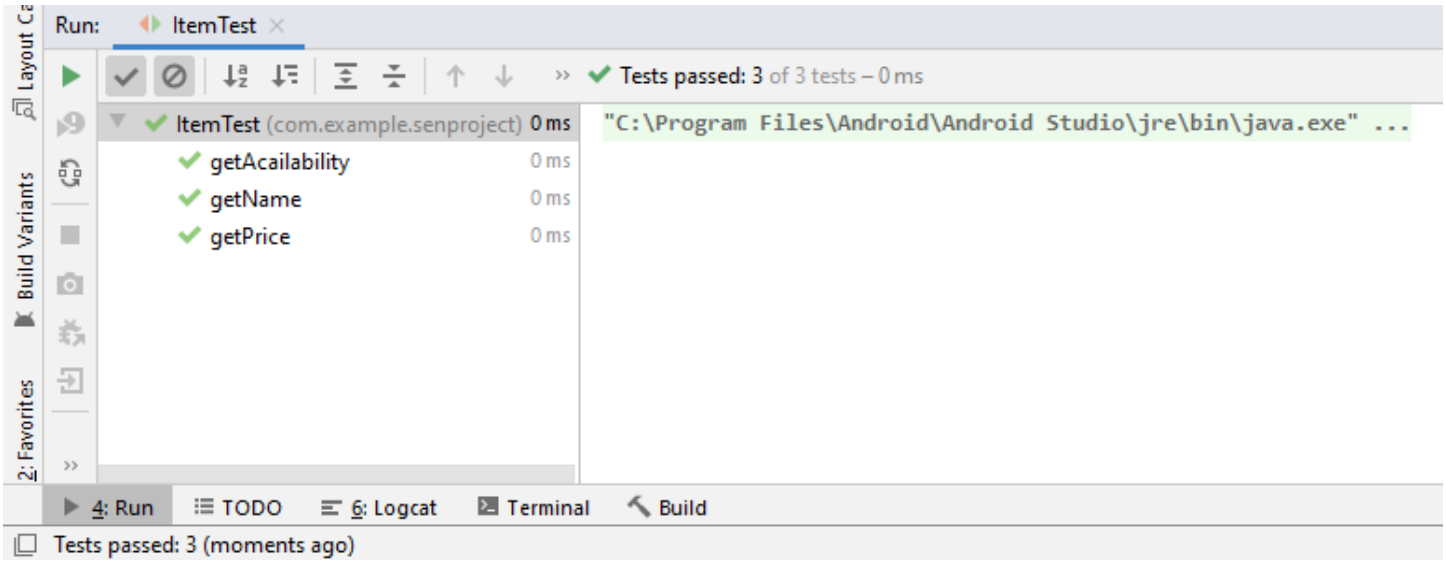


9.3 Item Activity Test

- **Code:**

```
public class ItemTest {

    private Item item_obj;
    @Test
    public void getName() {
        item_obj = new Item("Bhel","30","1");
        assertThat("Bhel",is(item_obj.getName()));
    }

    @Test
    public void getPrice() {
        item_obj = new Item("Bhel","30","1");
        assertThat("30",is(item_obj.getPrice()));
    }
    @Test
    public void getAcailability() {
        item_obj = new Item("Bhel","30","1");
        assertThat("1",is(item_obj.getAcailability()));
    }
}
```

- **Screenshot:**



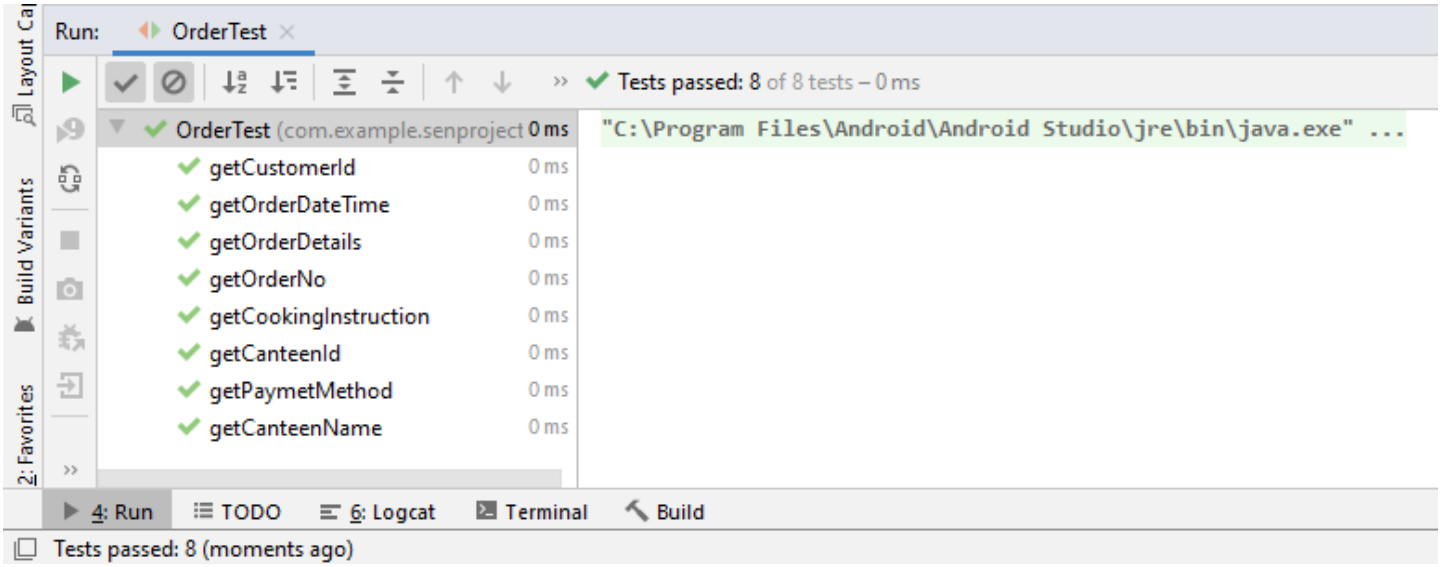
9.4 Order Activity Test

- **Code:**

```
public class OrderTest {
    private Order order_obj;
    @Test
    public void getPaymetMethod() {
        order_obj = new Order( "15","201701153","1","HoneyOne_C1","Bhel,22,1","12/06/2
020,23:13:12","Make Food Hot","Cash");
        assertThat("Cash",is(order_obj.getPaymetMethod()));
    }
    @Test
    public void getCookingInstruction() {
        order_obj = new Order("15","201701153","1","HoneyOne_C1","Bhel,22,1","12/06/20
20,23:13:12","Make Food Hot","Cash");
        assertThat("Make Food Hot",is(order_obj.getCookingInstruction()));
    }


    @Test
    public void getOrderNo() {
        order_obj = new Order("15","201701153","1","HoneyOne_C1","Bhel,22,1","12/06/20
20,23:13:12","Make Food Hot","Cash");
        assertThat("15",is(order_obj.getOrderNo()));
    }
```

```
    @Test
    public void getCustomerId() {
        order_obj = new Order("15","201701153","1","HoneyOne_C1","Bhel,22,1","12/06/20
20,23:13:12","Make Food Hot","Cash");
        assertThat("201701153",is(order_obj.getCustomerId()));
    }


    @Test
    public void getCanteenId() {
        order_obj = new Order("15","201701153","1","HoneyOne_C1","Bhel,22,1","12/06/20
20,23:13:12","Make Food Hot","Cash");
        assertThat("1",is(order_obj.getCanteenId()));
    }


    @Test
    public void getCanteenName() {
        order_obj = new Order("15","201701153","1","HoneyOne_C1","Bhel,22,1","12/06/20
20,23:13:12","Make Food Hot","Cash");
        assertThat("HoneyOne_C1",is(order_obj.getCanteenName()));
    }


    @Test
    public void getOrderDetails() {
        order_obj = new Order("15","201701153","1","HoneyOne_C1","Bhel,22,1","12/06/20
20,23:13:12","Make Food Hot","Cash");
        assertThat("Bhel,22,1",is(order_obj.getOrderDetails()));
    }


    @Test
    public void getOrderDateTime() {
        order_obj = new Order("15","201701153","1","HoneyOne_C1","Bhel,22,1","12/06/20
20,23:13:12","Make Food Hot","Cash");
        assertThat("12/06/2020,23:13:12",is(order_obj.getOrderDateTime()));
    }
}
```

- **Screenshot:**

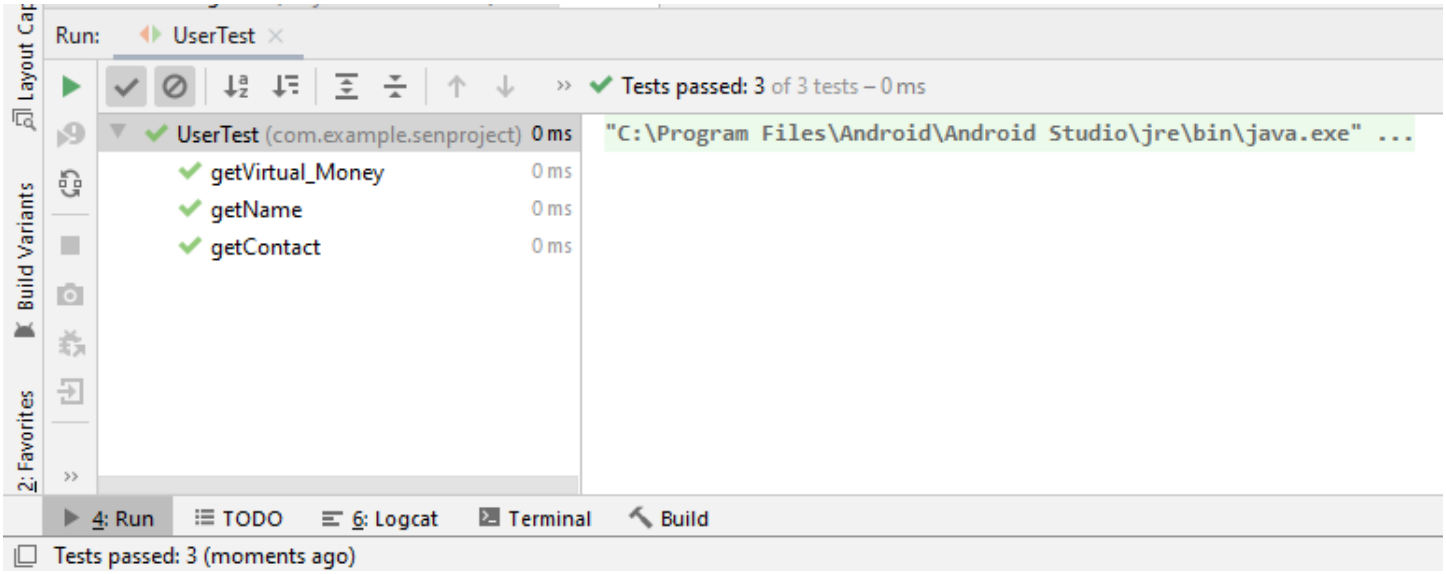


9.5 User Activity Test

- **Code:**

```
public class UserTest {

    private User user_obj;
    @Test
    public void getName() {
        user_obj = new User("Adit Mehta","7698459935","0");
        assertThat("Adit Mehta",is(user_obj.getName()));

    }

    @Test
    public void getContact() {
        user_obj = new User("Adit Mehta","7698459935","0");
        assertThat("7698459935",is(user_obj.getContact()));
    }

    @Test
    public void getVirtual_Money() {
        user_obj = new User("Adit Mehta","7698459935","0");
        assertThat("0",is(user_obj.getVirtual_Money()));
    }
}
```

- **Screenshot:**



10 Minutes of Meetings

10.1 Meeting 1

Date: 2/2/2020 Sunday

Agenda: Introduction of group members and decision on project to be selected.

Duration: 2 hours.

Description: All the members introduced themselves to the group. Then we all started discussing all the projects ideas that were given to us. Then we shortlisted some 5-7 projects in which all were interested and discussed what can be done in this projects and what are our limitations. Then we further shortlisted to 3-4 projects and at last we voted for each of our individual preferences for top 3 projects. The projects which we had selected were as follows:

1. Online Faculty Staff Directory of Multi University

2. Canteen Automation

3. Question Paper generation.

10.2 Meeting 2

Date: 5/2/2020 Wednesday

Agenda: Lab session and submission of functional and non-functional requirements of the project.
Duration: 1.5 hours.
Description: All the members started brainstorming about the different requirements that were needed and also we started various food delivering apps such as Uber Eats, Zomato etc. and saw the important, basic and necessary things that were implemented there and then we all had a discussion upon which requirements, for our project, should be finalized.

10.3 Meeting 3

Date: 12/2/2020 Wednesday

Agenda: Regarding Concept Poster and User Stories.
Duration: 3 hours.
Description: We all started up by writing up the user stories that we thought by considering our self as a user and then we discussed and narrowed it down to make the final document. We also made the concept poster by following the procedure that was done in 2nd lab and then discussed what to add and what to not add during that time in it.

10.4 Meeting 4

Date: 19/2/2020 Wednesday

Agenda: Regarding prioritizing and scheduling of sprints and user stories.
Duration: 2.5 hours.
Description: We all started up by looking up at the user stories and functional requirements that we had done earlier before and the discussed upon what are to be added in which sprints and started to edit the document.

10.5 Meeting 5

Date: 6/3/2020 Wednesday

Agenda: Regarding discussion about activity diagram class diagram and future planning.
Duration: 1.5 hours.
Description: We all discussed how to make the activity diagram and class diagram and also assigned different tasks to different people to do and made future planning on how to develop the app.

====XXXX====