Course code: SE 231
Course Title: System Analysis Design projects

Project Title: DIU Delivery (An instant Delivery System)

Submitted By

Md. Mahfuzur Rahman
Student ID: **203-35-3137**

Esmat Jahan Khan Bristi
Student ID: **203-35-3146**

Section: **A**
Department of Software Engineering

Submitted to

Mr. Khalid Been Md. Badruzzaman Biplob

Lecturer (Senior Scale)

Department of Software Engineering

Date of Submission: 08/12/2022

ID. 203-35-3137
ID. 203-35-3146

Table of Contents Page No.

Chapter 1

1.1 Introduction

When I am studying at Daffodil International University, I face a major problem with food and stationery items. For example, after 1 or 2 classes I feel hungry but I couldn't go outside to buy some food due to lack of time because I have another class continuously, and as the campus is so big so we need enough time to go shop and buy and eat some food. The same thing happened for stationery items. When you need stationery items like a pen, pencil or khata emergency but you haven't enough time or energy to go outside. And that problem is not only mine, almost every student's. Even I talked with 20-30 students about that problem and they agree with this. So I found a solution to it. I want to make a software system like a "share & care" system by ordering some food and delivering it.

1.1.2. Objectives

This delivery system is a web application based on share & care services. Our software objective is to give students and teachers food and stationary item services by order and delivery by students and they get part-time voluntary services that will eventually help students on a daily basis with little income. Without login, no user can use any functionality of this system.

This system has mainly 3 types of users.

1. System Administrator or Admin
2. Teachers
3. Students

The list of operations that the system will provide are-

1. All users must register and login to use this system.
2. Teachers and students both can order food and stationery items.
3. Any student can deliver that food or stationery items.

4. After delivery of the items students got the items prize and delivery charge as a reward for their voluntary services.

5. Based on their behavior they can give ratings to each other.

1.1.3. Scope

Delivery food and stationery items mobile apps allow students and teachers to order food and stationery items in shortage of time. In this mobile application on the home page, there are two sections for students. One is a delivery timeline and the other is ordering. In delivery, they can see if there are any unpicked delivery available or not. So that they can pick a delivery and complete it. In the order section, they can choose their product and order that.

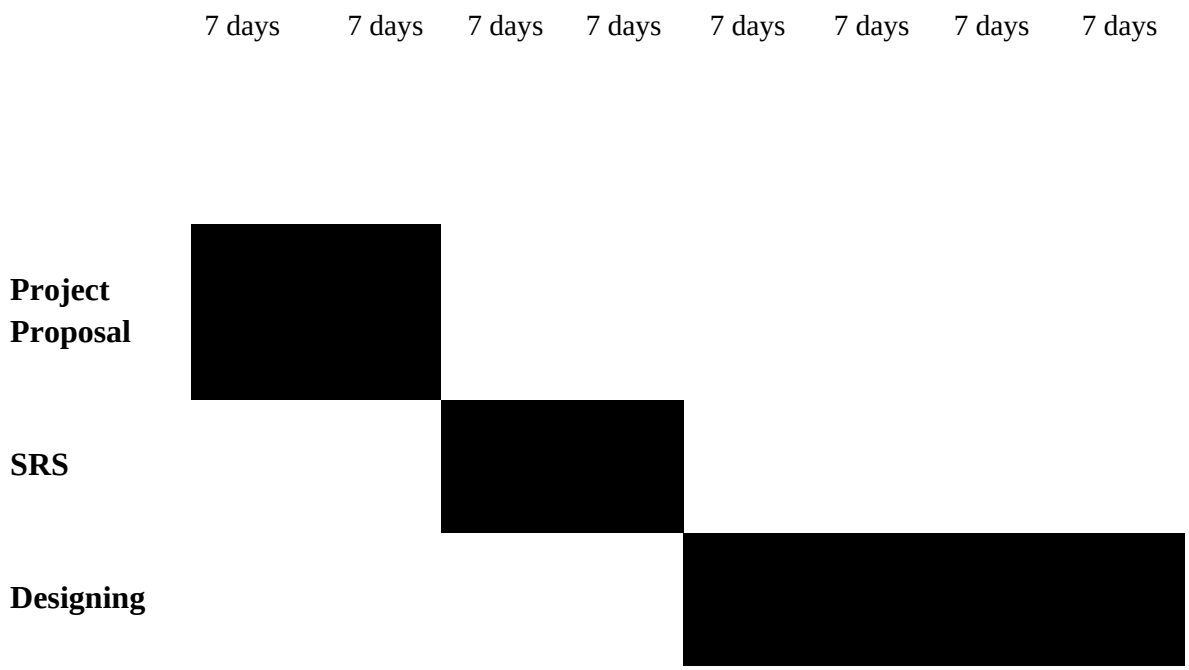
1.1.4. Assumptions and Constraints

As it is a pay and delivery system so no guest is allowed to use this system. Rather if anyone wants to use this system they must verify their ID card via KYC verification. Primarily there is no online payment system so there is no risk about money but later if the online payment system is included then this is a concern.

1.1.5. Dependencies and Risks

The user must have web access to use this system. The main risk behind implementing the project is Order management and security. If someone hacks the system then it will be a messy thing for all. Because in our system money is a major concern. Or if he finds a bug to order food without payment then it will be a disaster too. So in order to make quality software we make sure it's SQA testing and maintenance.

1.2.2 Timescales



1.2.3 Work Distribution

Project Proposal	Md. Mahfuzur Rahman Esmat Jahan Khan Bristi	14 days
Software Requirement Specification	Md. Mahfuzur Rahman Esmat Jahan Khan Bristi	14 days
Software Design	Md. Mahfuzur Rahman Esmat Jahan Khan Bristi	28 days

1.2.1 Deliverables

The following contents will be delivered with the project:

a) Project CD
   i. Project Demo
   ii. User manual along with Tutorial

b) Documentation

Hardware Requirements		
Processor	**RAM**	**Hard Disk Space**
Intel	256 Mb or higher	500 Mb or higher
Software Requirements		
Operating System		**Database**
For android users they must need android version 8 (Oreo) OS minimum and for apple users, they need ios 8 minimum requirement. The server machine must have 4 GB ram and a high-clock-speed processor. The server machine must have Windows 7/10/11 and ios along with .NET framework 4 and IIS.		SQL Server 2008
Browser		Best compatible Google Chrome and Safari

Table 1.2.2 Project Resources

1.3. Summary

With hunger we cannot concentrate on our studies. And for the majority of students that's a regular problem. The reason behind this problem is lack of time, continuous class, and long distance to the canteen. When they come to their university to study and they cannot concentrate on their study it's a concerning issue for everyone. So we take the initiative to mitigate this problem. We made a delivery system for students and teachers to order food and other stationery items when they need them.

1.4. References

1. HowTo: Write a project proposal [Online] URL:

https://www.indeed.com/career-advice/career-development/how-to-write-a-project-proposal

2. Foodpanda background case study:

https://www.academia.edu/22424501/Case_Study_on_Foodpanda

3. Case study:

https://www.routee.net/case-studies/foodpanda/

4. Online food delivery services.

https://www.coursehero.com/file/32151321/Case-Study-online-food-deliverypdf/

Chapter 2

REQUIREMENTS

2.1 System Analysis

Systems analysis is a technique for solving problems that breaks down systems into their component parts in order to examine how well those parts function independently and together to achieve the desired outcome.It was decided to use the traditional system development life cycle method because the needs for the software system were predictable. This procedure necessitates the development of software in a methodical, sequential manner, starting at the system level and moving through analysis, design, coding, testing, and maintenance. The stages that all software engineering paradigms must follow. The SDLC(Software Development Life Cycle) is used for the software.

2.2 System Engineering and Analysis

Every time there is a large system, each component must have requirements. Some subsets of these requirements must then be assigned to the software. A system perspective is vital when software needs to communicate with other components like hardware, humans, and databases. With a modest degree of top-level design analysis, requirements collection at the system level is included in system engineering and analysis.

2.3 Requirement Analysis

In systems engineering and software engineering, requirements analysis refers to the processes involved in identifying the requirements that must be satisfied for a new or modified project, taking into account the potentially conflicting requirements of the various stakeholders, as well as analyzing, documenting, validating, and managing software or system requirements.
The success of a systems or software project depends on the results of the requirements analysis. The requirements ought to be well-documented, usable, quantifiable, testable, traceable, tied to recognized business opportunities or needs, and sufficiently defined for system design.

2.4 Functional Requirements

No	Functional Requirements	Stakeholder
1	Registration	Students, Teachers
2	Login	
3	Available Items	
4	Search	
5	Add to Cart	
6	Order Items	
7	Confirm Order	
8	Cancel Order	
9	Order history	
10	Delivery Timeline	Students
11	Delivery history	
12	Track order	Students, Teachers
13	Make Payment	
14	User Rating	

15	**Give Feedback**	
16	**Chat System**	**Students, Teachers, Restaurant Owners**
17	**Add item**	**Restaurant Owners**

FR001	Registration
Description	Users must register on our system to use this software by providing their university email ID number.
Stakeholder	Students, Teachers

FR002	Login
Description	Both users have to log in before using our system
Stakeholder	Students and teachers

FR003	Available Items
Description	Users can see the food or stationery items that are available at that moment to order
Stakeholder	Students, Teachers

FR004	Search Items
Description	Users can search their items in the search box.
Stakeholder	Students, Teachers

FR005	Add to cart
Description	Users can add their items to the cart.

Stakeholder	Students, Teachers

FR006	Order Food
Description	Users can order their food or stationery item in this function. In this case, they have to choose the quantity of their food or stationery items and the delivery location.
Stakeholder	Students, Teachers

FR07	Confirm order
Description	Users must confirm their selected order or cart list by redirecting to the payment section and confirming payment. Before confirming the order, the user must select the location of his/her order. Like: AB4-612, AB1-313
Stakeholder	Students, Teacher

FR008	Cancel order
Description	Students and teachers can cancel the order request before it gets accepted by any students within 2 minutes.
Stakeholder	Students, Teachers

FR009	Order history
Description	Users can view all their previous orders they have been ordered.
Stakeholder	Students, Teachers

FR010	Delivery Timeline
Description	In this timeline section, Students can see all available orders which have to be delivered now.
Stakeholder	Students

FR011	Delivery history

Description	Students can see all the previous deliveries they have delivered.
Stakeholder	Students

FR012	Track order
Description	Users can track their orders.
Stakeholder	Students, Teacher

FR013	Make payment
Description	After delivery, they have to pay the money for the order. They can pay by cash or by online banking with a cashout charge.
Stakeholder	Students, Teacher

FR014	User Rating
Description	Users can rate Customers and delivery volunteers on their behavior and product delivery time.
Stakeholder	Students, Teacher

FR015	Give Feedback
Description	Users can give feedback or suggestions about their experience with the system.
Stakeholder	Students, Teacher

FR016	Chat system
Description	A chat system is integrated between the user and delivery volunteer when an order is pending between them. It's to find the perfect location or any request from the delivery volunteer.
Stakeholder	Students, Teacher, Restaurant Owners

FR017	Add item
Description	If a new Item comes on food or stationery items then the admin can add it to the bucket of the list for better user experience
Stakeholder	Restaurant Owners

2.5 Non-Functional Requirements

Reliability and Availability

The system has to be reliable so that users feel safe about their data. The system must be available around the clock to support smooth operations for many users at a time. This system must update regularly.

Speed, Latency, and Operational Requirements

This requirement specify a performance characteristic of a system or system component. The system requires a fair amount of speed, especially while browsing because on pickup time so many users use this system to order and deliver. This system must have the ability to handle so many users. While the user browses the system, the landing page will show within seconds. It also depends on the user's internet connection.

Quality and Acceptance Requirements

These Requirements define how to meet the physical and cognitive needs of the intended users of your website or application. The system is easy to use and can easily be understood.

Interface Requirements

The interface requirements describe the intended spirit, the mood, or the style of the product's appearance. These requirements specify the intention of the appearance. This requirement does not only define the necessity to use a css but also the requirements regarding the css's content as well as css frameworks like bootstrap and javascript.

Safety and Security Requirements

As this is a delivery system software and money is the currency that we use to deliver from one side to another side so security is a major concern of this system.

Verification Requirements

For our user identity verification requirements are necessary to make sure they are students or teachers of Daffodil International University.

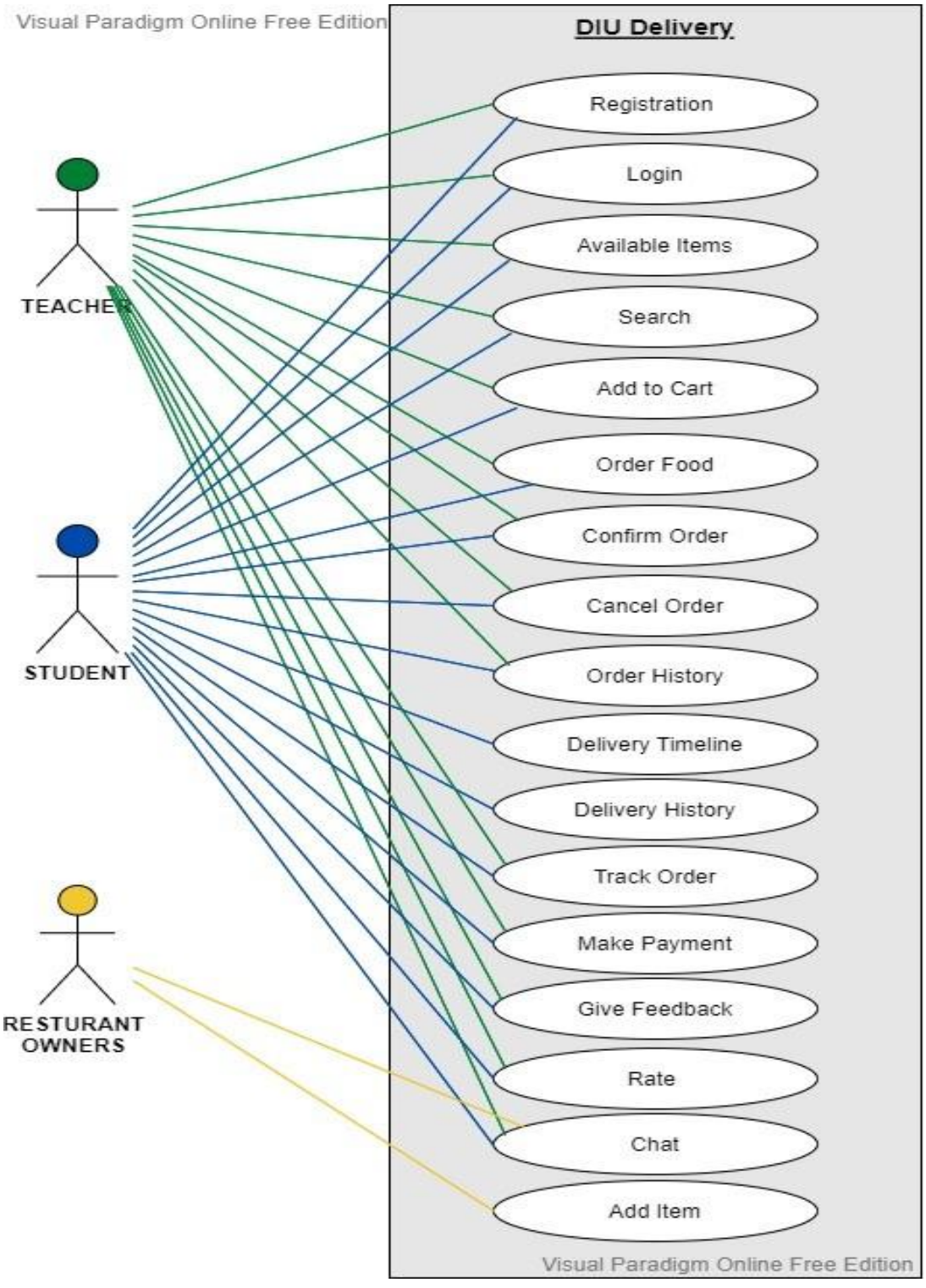
Maintainability Requirements

As this is an online base mobile application so it needs regular maintenance for smooth and secure performance. So that the system is not down on pickup time.

Chapter 3

System Analysis with Diagram

3.1 Use case diagram



3.2 Use case description

3.2.1 Registration

Use Case	Registration	
Goal <a longer statement of the goal in context if needed>	To use system features, users must log in, and to log in, they have to register first.	
Preconditions <what we expect is already the state of the world>		
Success End Condition <the state of the world upon successful completion>	1. Users have a unique daffodil mail address. 2. Users have a valid student ID card or teacher ID card.	
Failed End Condition <the state of the world if goal abandoned>	1. Email or password is not valid. 2. Email is already used 3. The student or teacher ID card is not valid.	
Primary Actors: **Secondary Actors:**	Teacher and Students	
Trigger <the action upon the system that starts use case>	Signup request comes in.	
Description / Main Success Scenario <the scenario steps from trigger to goal delivery and any clean-up after>	**Step**	**Action**
	1	User clicks on the Signup button to create a new user account.
	2 2.1 2.2 2.3 2.4 2.5 2.6 2.7	User fill-ups all empty boxes. Fill up the full name box. Fill up the mail address box. Fill up the student or teacher ID box. Fill up the user name box. Fill up the phone number box. Fill up the password box by providing a new password. Fill up the confirm password box by rewriting the given password..
	3	User clicks on the DONE button to complete creating an account.
	4	Users are registered.
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	2.6a	Do not set password properly
	2.6a1	Set password properly
Quality Requirements	**Step**	**Requirement**
	3	Users need a stable good internet connection to complete registration
	3.1	Users get 180 seconds time to complete registration after triggering the signup button.

3.2.2 Login

Use Case	Login	
Goal <a longer statement of the goal in context if needed>	To use system features and deliver and order, users must log in to the system.	
Preconditions <what we expect is already the state of the world>	Users must be registered.	
Success End Condition <the state of the world upon successful completion>	User fills up the sign-in form with a registered email and authentic password. Successfully logged in. Users see an order menu or delivery menu.	
Failed End Condition <the state of the world if goal abandoned>	1. The user is not registered yet. 2. Email or password is not valid. 3. Login Timeout	
Primary Actors: **Secondary Actors:**	Teacher and Students	
Trigger <the action upon the system that starts use case>	Login request comes in.	
Description / Main Success Scenario <the steps of the scenario from trigger to goal delivery and any clean-up after>	**Step**	**Action**
	1	User sees email and password entry box
	1.1 1.2	User enter a valid email User enter a valid password
	2	Under email and password box users will see a login button and create a new account button.
	2.1	User clicks login button
	3	User sees a new homepage.
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	1.2.a	If a user forgets a password then he can trigger the forget password button.
	1.2.b	A new page came to reset the password.
Quality Requirements	**Step**	**Requirement**
	2.1	Users get 120 seconds time to complete login after triggering the login button.

3.2.3 Available Items

Use Case	Available Items	
Goal <a longer statement of the goal in context if needed>	Customers can see the available items in the canteen and stores of DIU.	
Preconditions <what we expect is already the state of the world>	Customers must go to the order section	
Success End Condition <the state of the world upon successful completion>	If items are available, then users see the available items. User sees the available items with their prices.	
Failed End Condition <the state of the world if goal abandoned>	User is not logged in yet.	
Primary Actors: **Secondary Actors:**	Students, Teachers	
Trigger <the action upon the system that starts use case>		
Description / Main Success Scenario <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	Users logged into the system.
	2	Users view the home screen which shows available items, search box, cart, etc.
Alternative Flows <a: condition causing branching> <a1: action or name of sub use case>	**Step**	**Branching Action**
	3a	
	3a1	
Quality Requirements	**Step**	**Requirement**
	1	Users need a stable good internet connection to use the system.

3.2.4 SEARCH

Use Case	Search	
Goal <a longer statement of the goal in context if needed>	Customers will be able to search for items they want.	
Preconditions <what we expect is already the state of the world>	Customers must be logged in to the system.	
Success End Condition <the state of the world upon successful completion>	Searched item is successfully found.	
Failed End Condition <the state of the world if goal abandoned>	1. Searched item is not found because of a spelling mistake. 2. Searched item is not available in the system.	
Primary Actors: **Secondary Actors:**	Students, Teachers	
Trigger <the action upon the system that starts use case>	Search request comes in.	
Description / Main Success Scenario <the scenario steps from trigger to goal delivery and any clean-up after>	**Step**	**Action**
	1	Click on the Search icon or box.
	2	Write the name of the item the user wants to search.
	3	Click on the enter button to start searching.
	4	If the item is available, it will be shown on the screen to the user.
Alternative Flows <a: condition causing branching> <a1: action or name of sub use case>	**Step**	**Branching Action**
	2a	Name is not correct.
	2a1	Set the correct name.
Quality Requirements	**Step**	**Requirement**
	3	Users need a stable good internet connection to search for items.
	4	Users need to wait for a maximum of 10 seconds time to see if the searched item is found after triggering the enter button.

3.2.5 Add to Cart

Use Case	Add to Cart	
Goal <a longer statement of the goal in context if needed>	Users can select and add that selected item or items to the cart.	
Preconditions <what we expect is already the state of the world>	View items and select one or more items.	
Success End Condition <the state of the world upon successful completion>	Item or items added to cart	
Failed End Condition <the state of the world if goal abandoned>	Item or items failed to be added to the cart	
Primary Actors: **Secondary Actors:**	Students, Teachers	
Trigger <the action upon the system that starts use case>	Users will request the system to add items or items to cart.	
Description / Main Success Scenario <the scenario steps from trigger to goal delivery and any clean-up after>	**Step**	**Action**
	1	Click on food or stationery item
	2	Click on the Cart icon of that food or stationery item if you want to add that to the cart
	3	Food or stationery item added on the cart.
Alternative Flows <a: condition causing branching> <a1: action or name of sub use case>	**Step**	**Branching Action**
	2a	DId not add an item to cart
	2a1	Select item and click on Cart icon to add it to cart
Quality Requirements	**Step**	**Requirement**
	2	Users need a stable good internet connection to add items on the cart.
	3	Users need to wait for a maximum of 5 seconds to see if the item is added on the cart after triggering the cart icon or button.

3.2.6 Order Items

Use Case	Order Food or Stationery Item	
Goal <a longer statement of the goal in context if needed>	Customers will order food or stationery item from the selected items in the cart	
Preconditions <what we expect is already the state of the world>	Food or the stationery item is added to the cart.	
Success End Condition <the state of the world upon successful completion>	Food or the stationery item is ordered.	
Failed End Condition <the state of the world if goal abandoned>	1. Food or stationery item is not in the cart 2. Quantity not entered.	
Primary Actors: **Secondary Actors:**	Students, Teachers	
Trigger <the action upon the system that starts use case>	Order Food request comes in.	
Description / Main Success Scenario <the scenario steps from trigger to goal delivery and any clean-up after>	**Step**	**Action**
	1	User clicks on the cart icon.
	1.1	Before ordering, users have to make sure they have what they want to order in the cart.
	2	Provide the quantity of the items in the cart to order.
	3.	Click on the Order button to place an order.
	4	Order of food or stationery item is placed.
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	1.1a	Item is not on the cart
	1.1a1	Add item in the cart
	2a	Did not set the quantity
	2a1	Set the quantity
Quality Requirements	**Step**	**Requirement**
	3	Users need a stable, good internet connection to order food or stationery items.
	4	Users need to wait for a maximum of 5 seconds time to see if the order is placed after triggering the Order button.

3.2.7 Confirm Order

Use Case	Confirm Order	
Goal <a longer statement of the goal in context if needed>	Customers will be able to confirm the order after providing the delivery location	
Preconditions <what we expect is already the state of the world>	To confirm the order, customers have to order food first	
Success End Condition <the state of the world upon successful completion>	Food or stationery item order confirmed.	
Failed End Condition <the state of the world if goal abandoned>	Confirmation denied because the requirements are fulfilled	
Primary Actors: **Secondary Actors:**	Students, Teachers	
Trigger <the action upon the system that starts use case>	Customers send confirmation order requests to confirm their order.	
Description / Main Success Scenario <the steps of the scenario from trigger to goal delivery and any clean-up after>	**Step**	**Action**
	1 1.1 1.2 1.3	After an order is placed, the user has to provide the location of delivering the order. Fill in the building name. Fill in the floor number. If users want, they can fill in the optional box to provide more information.
	2.	Click on the Confirm button to confirm that order for delivery.
	3.	Order is confirmed.
Alternative Flows <a: condition causing branching> <a1: action or name of sub use case>	**Step**	**Branching Action**
	1a	User did not give the location properly
	1a1	Set the location properly
Quality Requirements	**Step**	**Requirement**
	2	Users need a stable good internet connection to confirm an order.
	3	Users need to wait for a maximum of 5 seconds time to see if the order is placed after triggering the Confirm button.

3.2.8 Cancel Order

Use Case	Cancel Order	
Goal <a longer statement of the goal in context if needed>	Customers can cancel the order within 2 minutes of confirming the order if it is already not accepted by a volunteer.	
Preconditions <what we expect is already the state of the world>	Food or stationery item order needs to be confirmed.	
Success End Condition <the state of the world upon successful completion>	Food or stationery item order is canceled.	
Failed End Condition <the state of the world if goal abandoned>	1. The 2 minutes time limit is over. 2. Food or stationery item order is already accepted.	
Primary Actors: **Secondary Actors:**	Students, Teachers	
Trigger <the action upon the system that starts use case>	Customers send cancellation order requests to cancel their order.	
Description / Main Success Scenario <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	After confirming an order, customers will be shown a Cancel order screen.
	1.1	Order is not accepted by a volunteer, so the customer can click on the Cancel button to cancel that order within two minutes.
	2	Order is canceled
Alternative Flows <a: condition causing branching> <a1: action or name of sub use case>	**Step**	**Branching Action**
	2a	Order not canceled
	2a1	Click on the cancel button
Quality Requirements	**Step**	**Requirement**
	1.1	Users need a stable good internet connection to cancel an order.
	2	Users need to wait for a maximum of 5 seconds time to see if the order is canceled after triggering the Cancel button.

3.2.9 Order History

Use Case	Order History	
Goal <a longer statement of the goal in context if needed>	Customers can view the previous order they made.	
Preconditions <what we expect is already the state of the world>	Food or stationery items are ordered and delivered.	
Success End Condition <the state of the world upon successful completion>	Successfully shown the Order History.	
Failed End Condition <the state of the world if goal abandoned>	Failed to show the Order History	
Primary Actors: **Secondary Actors:**	Students, Teachers	
Trigger <the action upon the system that starts use case>	Order History request comes in.	
Description / Main Success Scenario <the steps of the scenario from trigger to goal delivery and any clean-up after>	**Step**	**Action**
	1 1.1	Customers have to click on the Order History icon to see the date and time of an order that was delivered to them. Customers can also see the previous order list and the place they ordered from.
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	1a	Did not click on Order History icon
	1a1	Click on the icon to view order history
Quality Requirements	**Step**	**Requirement**
	1	Users need a stable good internet connection to see the Order History

3.2.10 Delivery Timeline

Use Case	Delivery Timeline	
Goal <a longer statement of the goal in context if needed>	Students can pick any order of customers based on their suitable location.	
Preconditions <what we expect is already the state of the world>	Must log in as a delivery volunteer.	
Success End Condition <the state of the word upon successful completion>	Customers must order an item. They got a notification of an order is pending in the timeline	
Failed End Condition <the state of the world if goal abandoned>	Failed to show the Delivery Timeline because there is no order in the timeline.	
Primary Actors: **Secondary Actors:**	Students	
Trigger <the action upon the system that starts use case>	Pick up request comes in. by clicking, you are assigned for that delivery.	
Description / Main Success Scenario <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	First Login as a student .
	2	After login, users see two different sections.
	2.1 2.2	One is the Order section to order other one is the delivery timeline to pickup delivery Their preferred location is close, delivery comes first, then others come.
	3	Every Student will see the same order request at the same time.
	4.	When a student clicks pick up he has to be assigned to deliver that order request.
Alternative Flows <a: condition causing branching> <a1: action or name of sub use case>	**Step**	**Branching Action**
	3a	
	3a1	
	4a	
	4a1	
	7a	
	7a1	
Quality Requirements	**Step**	**Requirement**
	1	Users need a stable good internet connection to see the Delivery Timeline

3.2.11 Delivery History

Use Case	Delivery History	
Goal <a longer statement of the goal in context if needed>	Volunteered students can view the previous order they delivered	
Preconditions <what we expect is already the state of the world>	Food or stationery items are ordered and delivered.	
Success End Condition <the state of the world upon successful completion>	Successfully showed delivery history	
Failed End Condition <the state of the world if goal abandoned>	Failed to show the Delivery History	
Primary Actors: **Secondary Actors:**	Students	
Trigger <the action upon the system that starts use case>	Delivery History request comes in.	
Description / Main Success Scenario <the steps of the scenario from trigger to goal delivery and any clean-up after>	**Step**	**Action**
	1 1.1	Volunteered students have to click on the Delivery History icon to see the date and time of an order they delivered to customers. Students can also see the previous order list and the place the order was made
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	1a	Did not click on Delivery History icon
	1a1	Click on the icon to view delivery history
Quality Requirements	**Step**	**Requirement**
	1	Users need a stable good internet connection to see the Delivery History

3.2.12 Track Order

Use Case	Track Order	
Goal <a longer statement of the goal in context if needed>	To track a placed order till it is delivered.	
Preconditions <what we expect is already the state of the world>	Must place an order first.	
Success End Condition <the state of the world upon successful completion>	Successfully tracked an order.	
Failed End Condition <the state of the world if goal abandoned>	Failed to track the Order.	
Primary Actors: **Secondary Actors:**	Students, Teachers	
Trigger <the action upon the system that starts use case>	Track Order request comes in.	
Description / Main Success Scenario <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1 1.1 1.2 1.3	After an order is placed, customers will be shown on screen the track of that order. The preparation of the order. The pick-up of the order. The delivery of the order.
	2	After delivery is done, tracking of the order is complete
Alternative Flows <a: condition causing branching> <a1: action or name of sub use case>	**Step**	**Branching Action**
	3a	
	3a1	
	4a	
	4a1	
	7a	
	7a1	
Quality Requirements	**Step**	**Requirement**
	1	Users need a stable good internet connection to track the Order

3.2.13 Make Payment

Use Case	Make Payment	
Goal <a longer statement of the goal in context if needed>	After getting the delivery of their own order of customer he/she has to pay the bill of their items.	
Preconditions <what we expect is already the state of the world>	Customers must order a product.	
Success End Condition <the state of the world upon successful completion>	Payment successfully received.	
Failed End Condition <the state of the world if goal abandoned>	1. Due to a poor internet connection I failed to make payment. 2. OTP doesn't match. 3. Timeout	
Primary Actors: **Secondary Actors:**	Students, Teachers	
Trigger <the action upon the system that starts use case>	Make Payment request comes in when order is in the near door.	
Description / Main Success Scenario <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	First login into the system.
	2	Then go to the order section. In the beginning, the customer sees the last order. Besides the order, there is a red color popup button for payment.
	3	Click the red payment button it will redirect to the online payment system
	4	After payment the red button turns into a green button.
Alternative Flows <a: condition causing branching> <a1: action or name of sub use case>	**Step**	**Branching Action**
	3a	
	3a1	
	4a	
	4a1	
	7a	
	7a1	
Quality Requirements	**Step**	**Requirement**
	1	Complete all the processes before timeout.

3.2.14 Give Feedback

Use Case	Give Feedback	
Goal <a longer statement of the goal in context if needed>	Customers can comment on the system	
Preconditions <what we expect is already the state of the world>		
Success End Condition <the state of the world upon successful completion>	Successfully gave feedback	
Failed End Condition <the state of the world if goal abandoned>	Failed to give feedback	
Primary Actors: **Secondary Actors:**	Students, Teachers	
Trigger <the action upon the system that starts use case>	A Feedback request comes in.	
Description / Main Success Scenario <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	Customer clicks on the Comment button .
	2	Customers can write about their complaint or compliment about the system
	3	After writing, they click on the Enter button.
	4	Feedback is given
Alternative Flows <a: condition causing branching> <a1: action or name of sub use case>	**Step**	**Branching Action**
	3a	Did not click Enter button
	3a1	Click Enter button
Quality Requirements	**Step**	**Requirement**
	2	Users need a stable good internet connection to give feedback

	3	Users need to wait for a maximum of 5 seconds time to see if feedback is given after triggering Enter button

3.2.15 Rate

Use Case	Rate	
Goal <a longer statement of the goal in context if needed>	Customers can rate the system	
Preconditions <what we expect is already the state of the world>	An order is delivered first.	
Success End Condition <the state of the world upon successful completion>	Successfully rated the system	
Failed End Condition <the state of the world if goal abandoned>	Failed to rate	
Primary Actors: **Secondary Actors:**	Students, Teachers	
Trigger <the action upon the system that starts use case>	Rate request comes in.	
Description / Main Success Scenario <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	After an order is delivered, customers will get a rating option on the system to rate the order
	2	Customers will get 1 to 5 stars on which they can click according to their satisfaction.
	3	After clicking on the star, customers will click on the RATE button
	4	Rating is done
Alternative Flows <a: condition causing branching> <a1: action or name of sub use case>	**Step**	**Branching Action**
	2a	Did not click on a star
	2a1	Click on a star
	3a	Did not click on Rate button
	3a1	Click on Rate button
Quality Requirements	**Step**	**Requirement**
	2,3	Users need a stable good internet connection to rate.

3.2.16 Chat

Use Case	Chat	
Goal <a longer statement of the goal in context if needed>	Customers will be able to chat on the system with the volunteered student for delivery during an order	
Preconditions <what we expect is already the state of the world>	Customers have to place an order first	
Success End Condition <the state of the world upon successful completion>	Chat is done successfully.	
Failed End Condition <the state of the world if goal abandoned>	Failed to chat	
Primary Actors: **Secondary Actors:**	Students, Teachers, Restaurant Owners	
Trigger <the action upon the system that starts use case>	Chat request comes in.	
Description / Main Success Scenario <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	After an order is placed and picked up by a volunteer student, a delivery chat option will be available.
	2 2.1	Customer and delivery person can click on the Chat button Users can write in the text section.
	3	After delivery, the chat option is closed.
Alternative Flows <a: condition causing branching> <a1: action or name of sub use case>	**Step**	**Branching Action**
	2a	Did not click on the Chat button
	2a1	Click on the Chat button

Quality Requirements	Step	Requirement
	4	Users need a stable, good internet connection to chat.

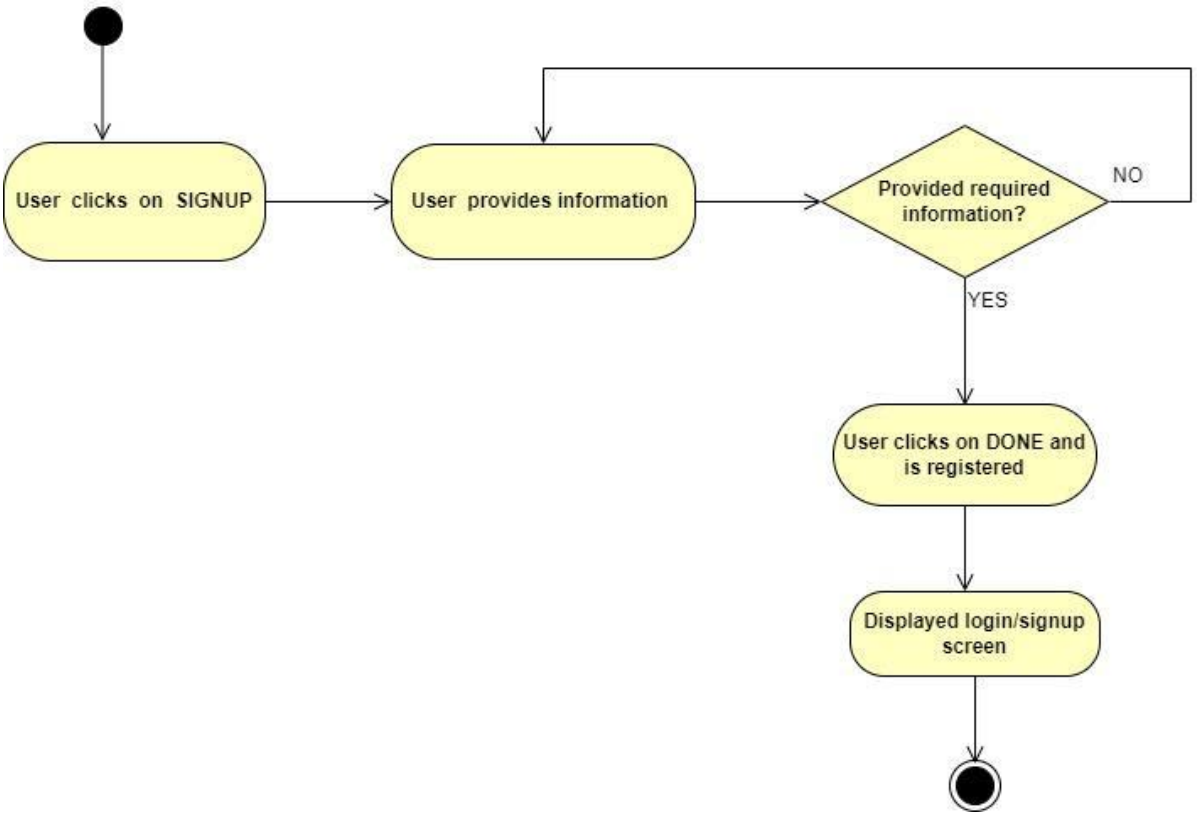
3.2.17 Add Item

Use Case	Add Item	
Goal <a longer statement of the goal in context if needed>	Restaurant owners can add new items to the system.	
Preconditions <what we expect is already the state of the world>		
Success End Condition <the state of the world upon successful completion>	Successfully added an item with a name, price, and picture.	
Failed End Condition <the state of the world if goal abandoned>	Failed to add an item	
Primary Actors: **Secondary Actors:**	Restaurant Owners	
Trigger <the action upon the system that starts use case>	Add item request comes in.	
Description / Main Success Scenario <the steps of the scenario from trigger to goal delivery and any clean-up after>	**Step**	**Action**
	1 1.1 1.2 1.3	Click on the Add item button. Fill up the name box of the item Fill up the price box of the item Upload an image of the item in JPG format.
	2	Click on the ADD button.
	3	Items are added on the system.
Alternative Flows <a: condition causing branching> <a1: action or name of sub use case>	**Step**	**Branching Action**
	1.1a	Did not fill name box
	1.1a1	Fill up the name box
	1.2a	Did not fill price box
	1.2a1	Fillup price box
	1.3a	Did not upload an image of item
	1.3a1	Upload an image of item
	2a	Did not click ADD button
	2a1	Click ADD button
Quality Requirements	**Step**	**Requirement**

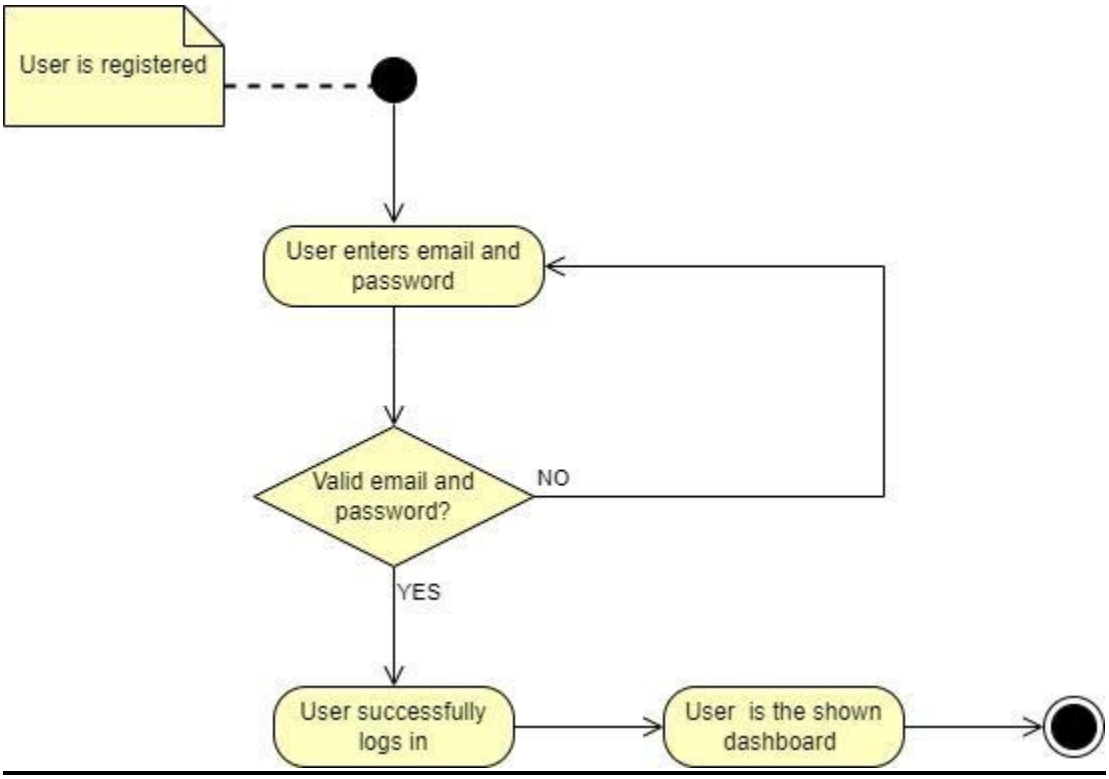
	1	Users need a stable good internet connection to add an item.
	2	Users need to wait for a maximum of 10 seconds to see if an item is added after triggering the ADD button.

3.3 Activity Diagram

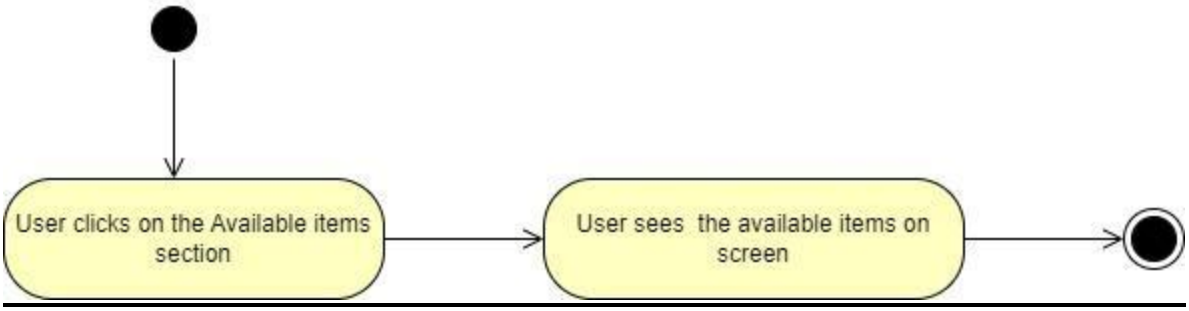
- **Registration**



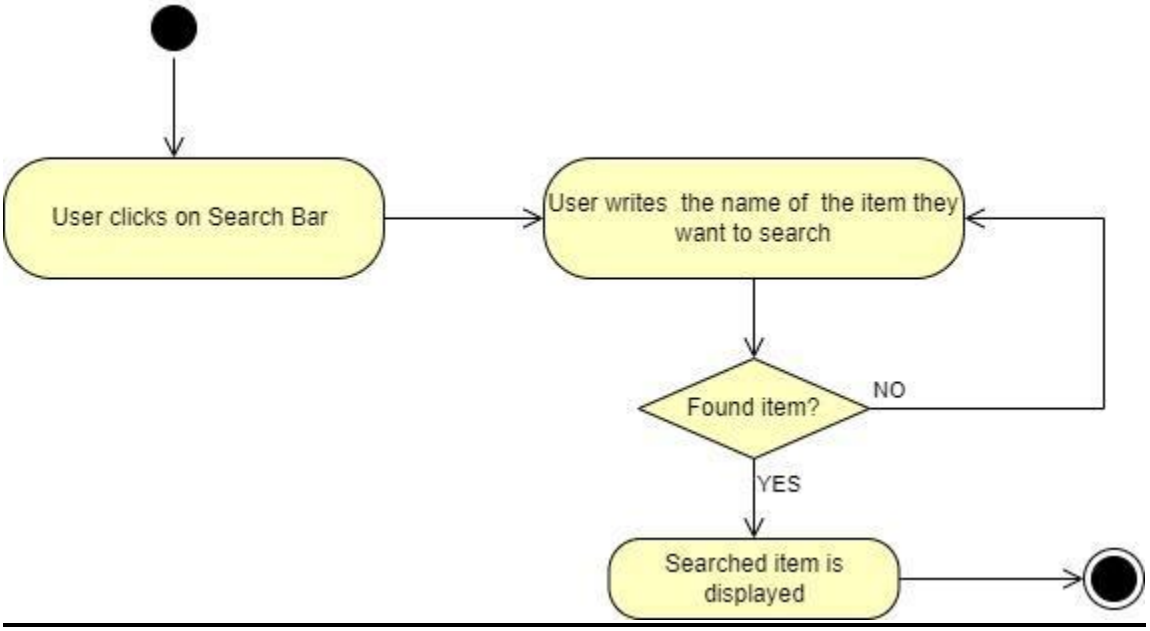
- **Login**



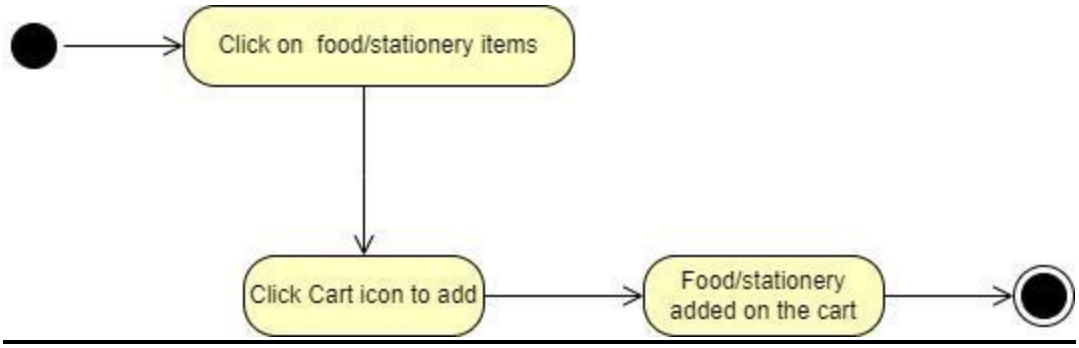
- **Available Items**



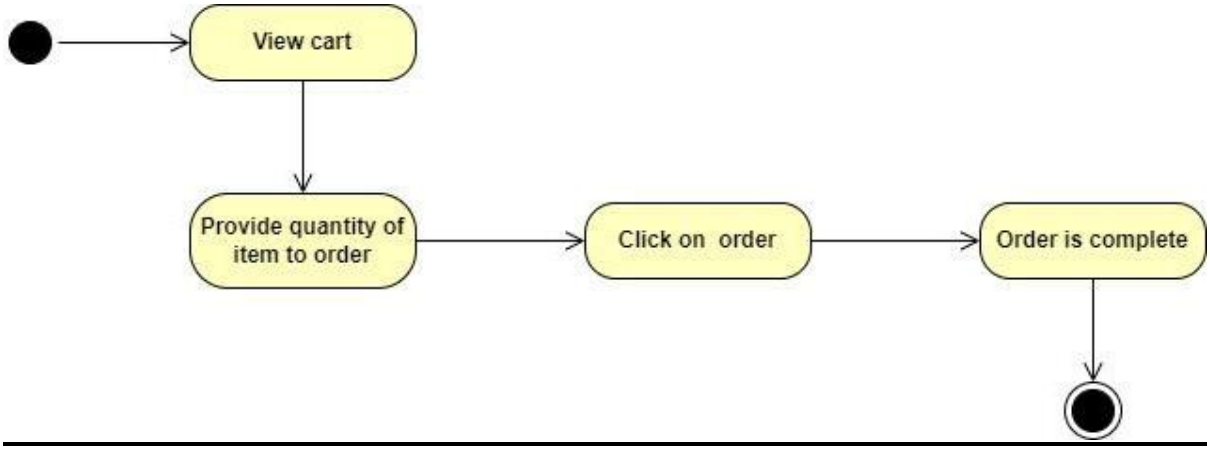
- **SEARCH**



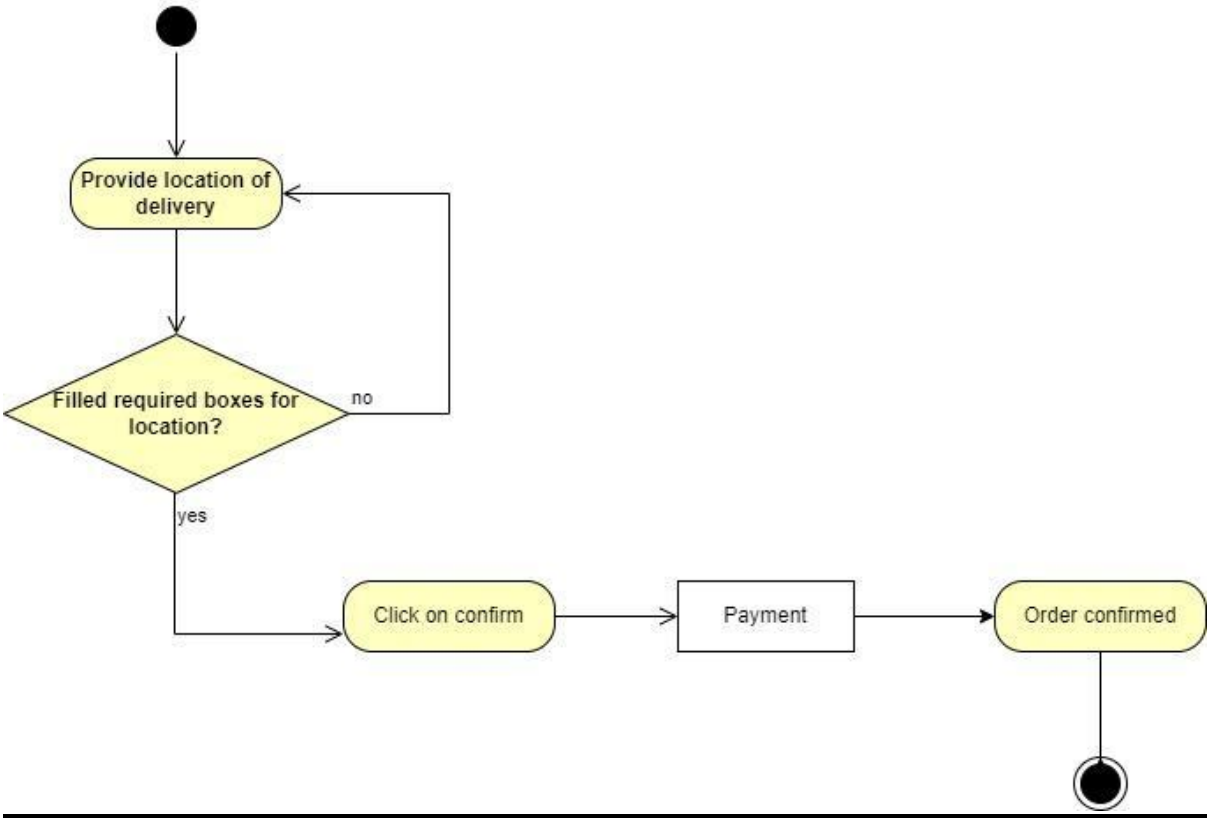
- **Add to Cart**



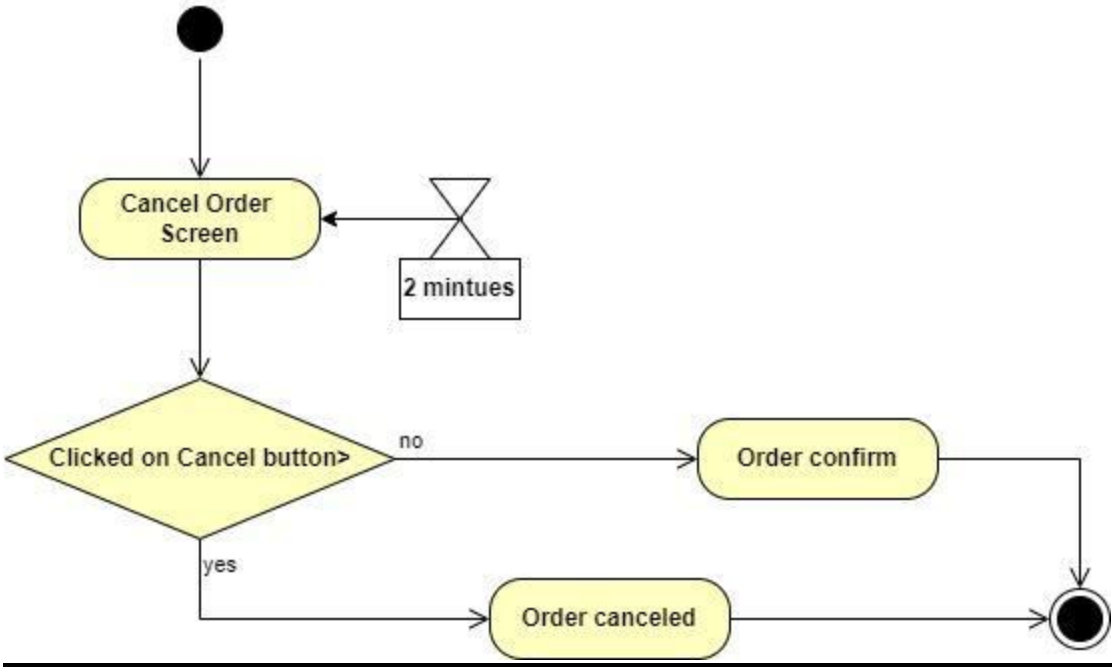
- **Order Items**



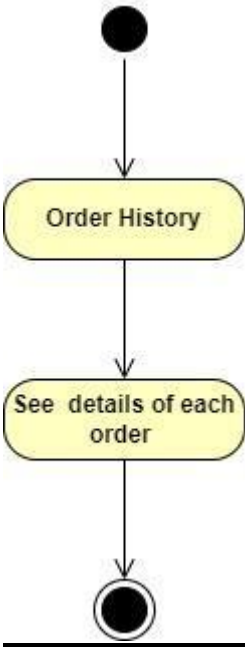
- **Confirm Order**



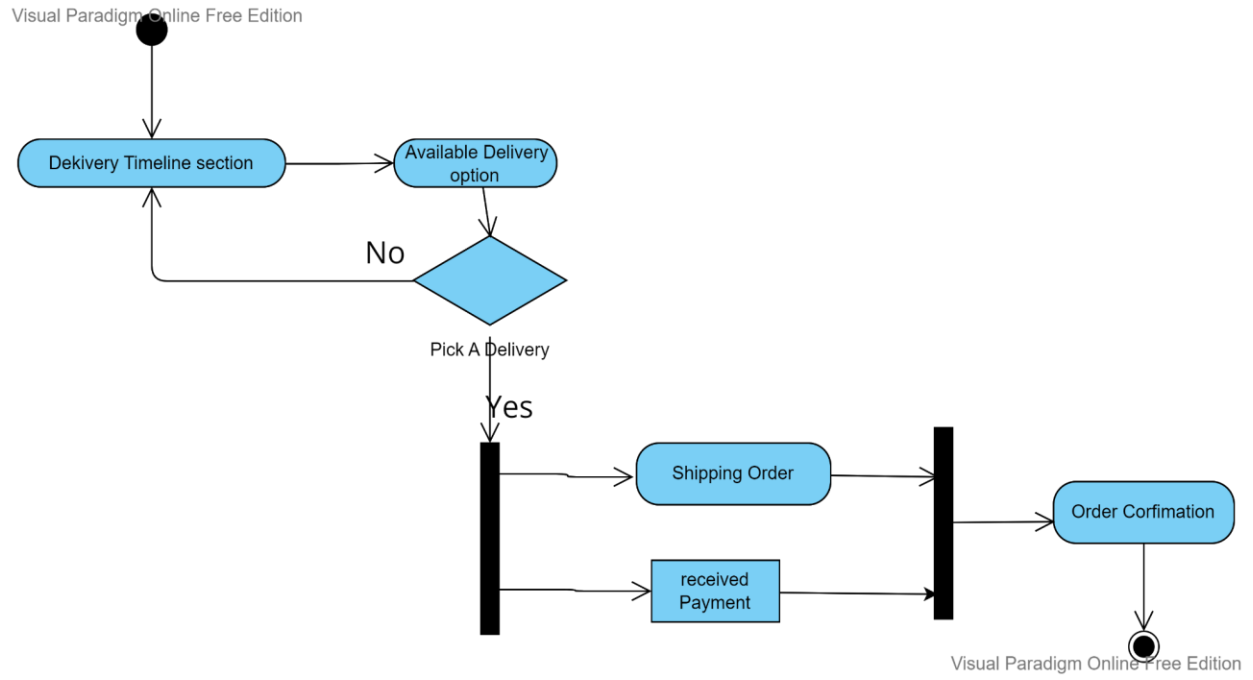
- **Cancel Order**



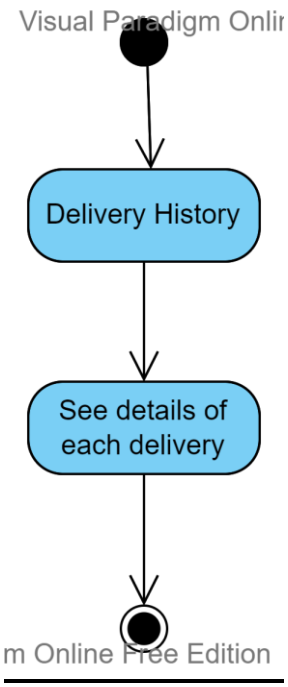
- **Order History**



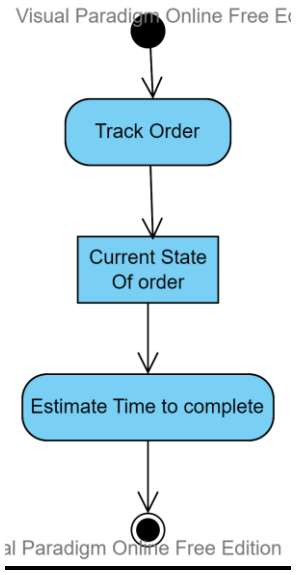
- **Delivery Timeline**



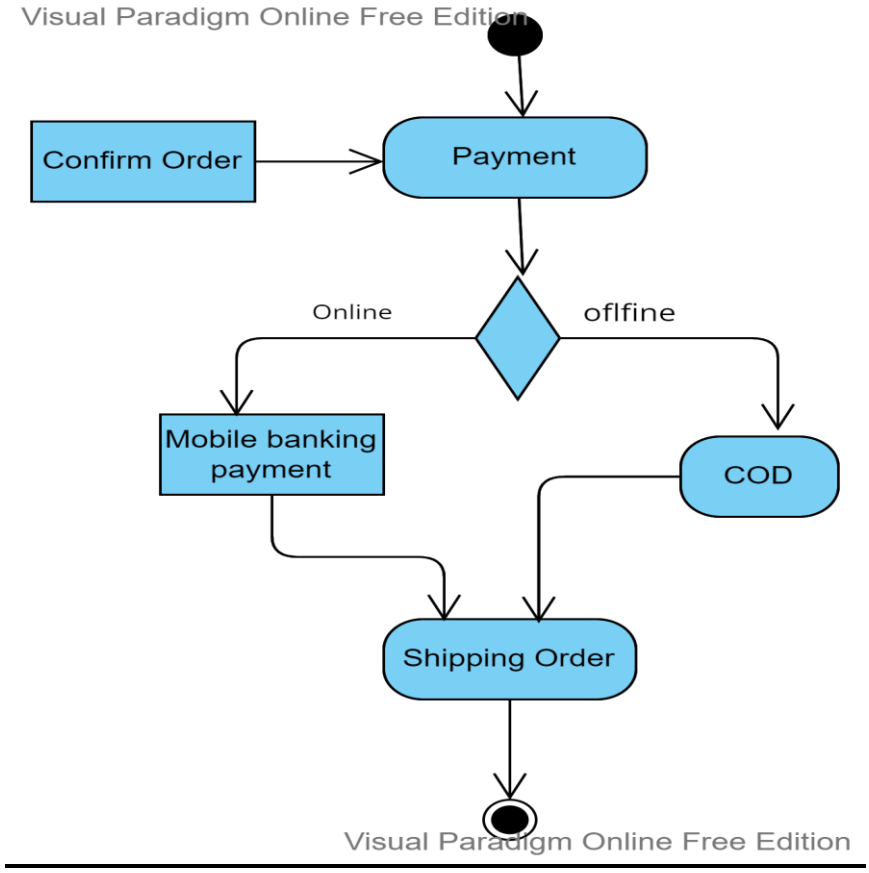
- **Delivery History**



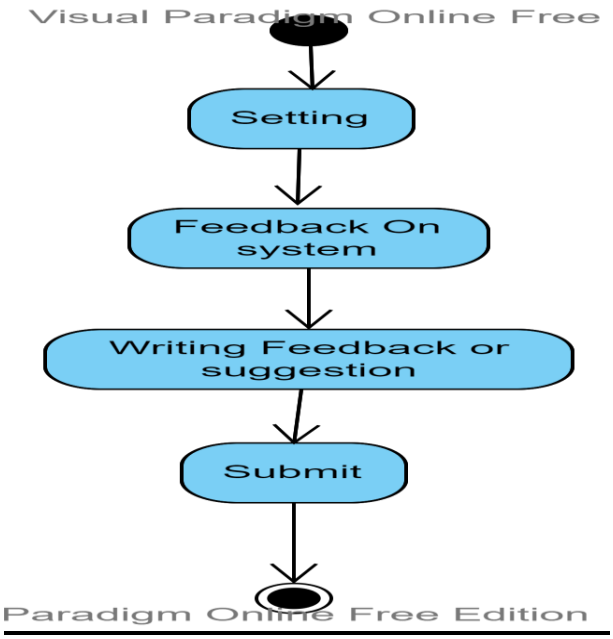
- **Track Order**



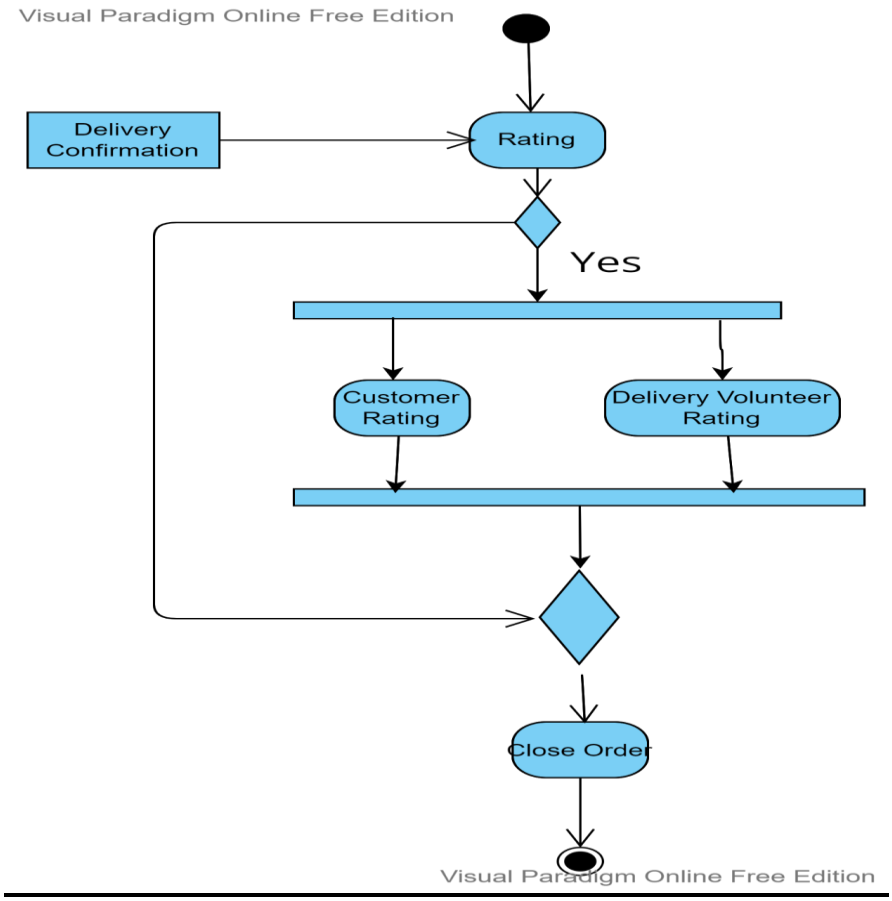
- **Make Payment**



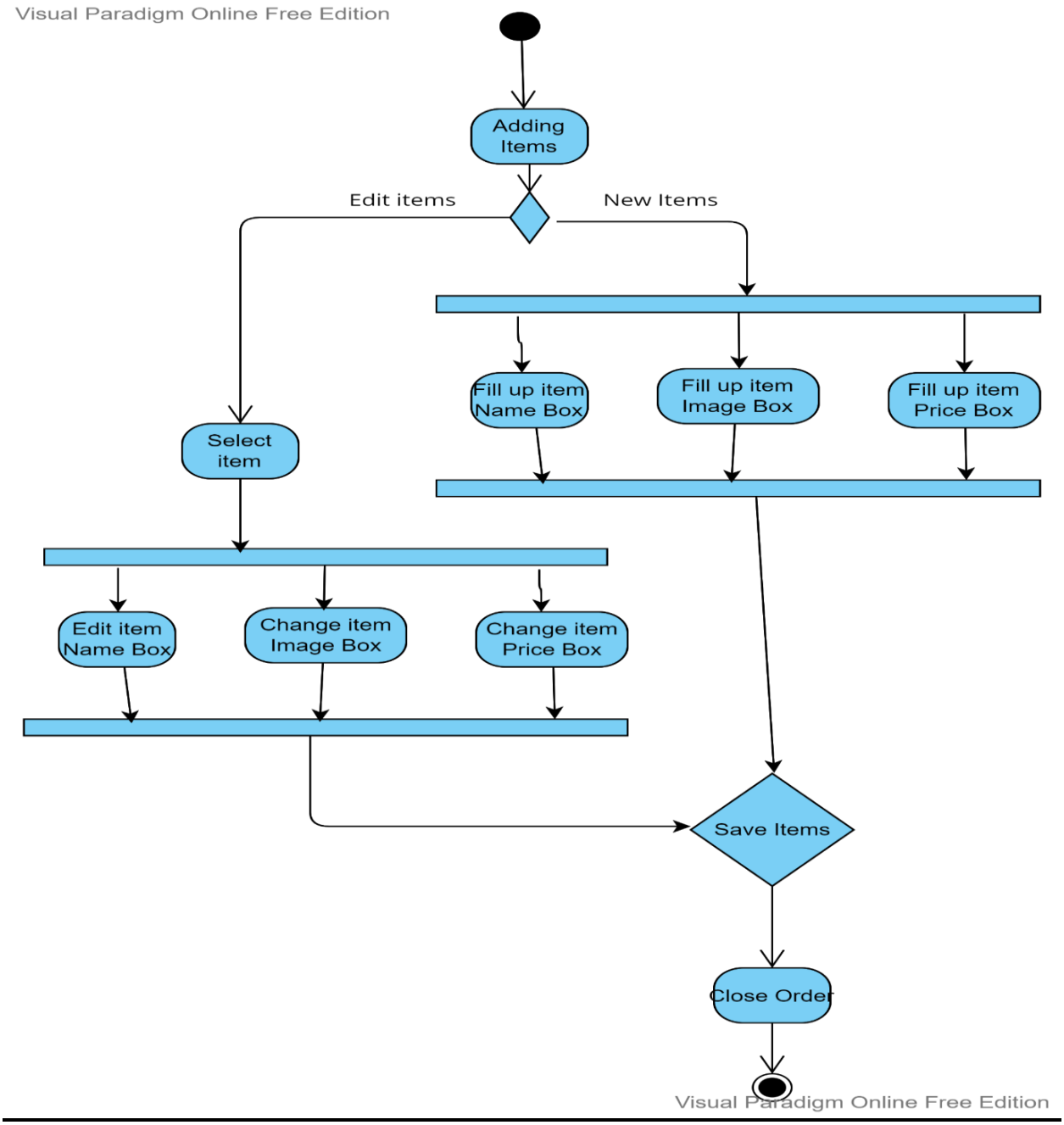
- **Give Feedback**



- **Rate**


Visual Paradigm Online Free Edition
Delivery Confirmation
Rating
Yes
Customer Rating
Delivery Volunteer Rating
Close Order
Visual Paradigm Online Free Edition

- **Add Item**


Visual Paradigm Online Free Edition
Adding Items
Edit items
New Items
Select item
Fill up item Name Box
Fill up item Image Box
Fill up item Price Box
Edit item Name Box
Change item Image Box
Change item Price Box
Save Items
Close Order
Visual Paradigm Online Free Edition

3.4 ERD Diagram

Our primary actors are

1. Students
2. Teachers
3. Restaurant/stationary Owner

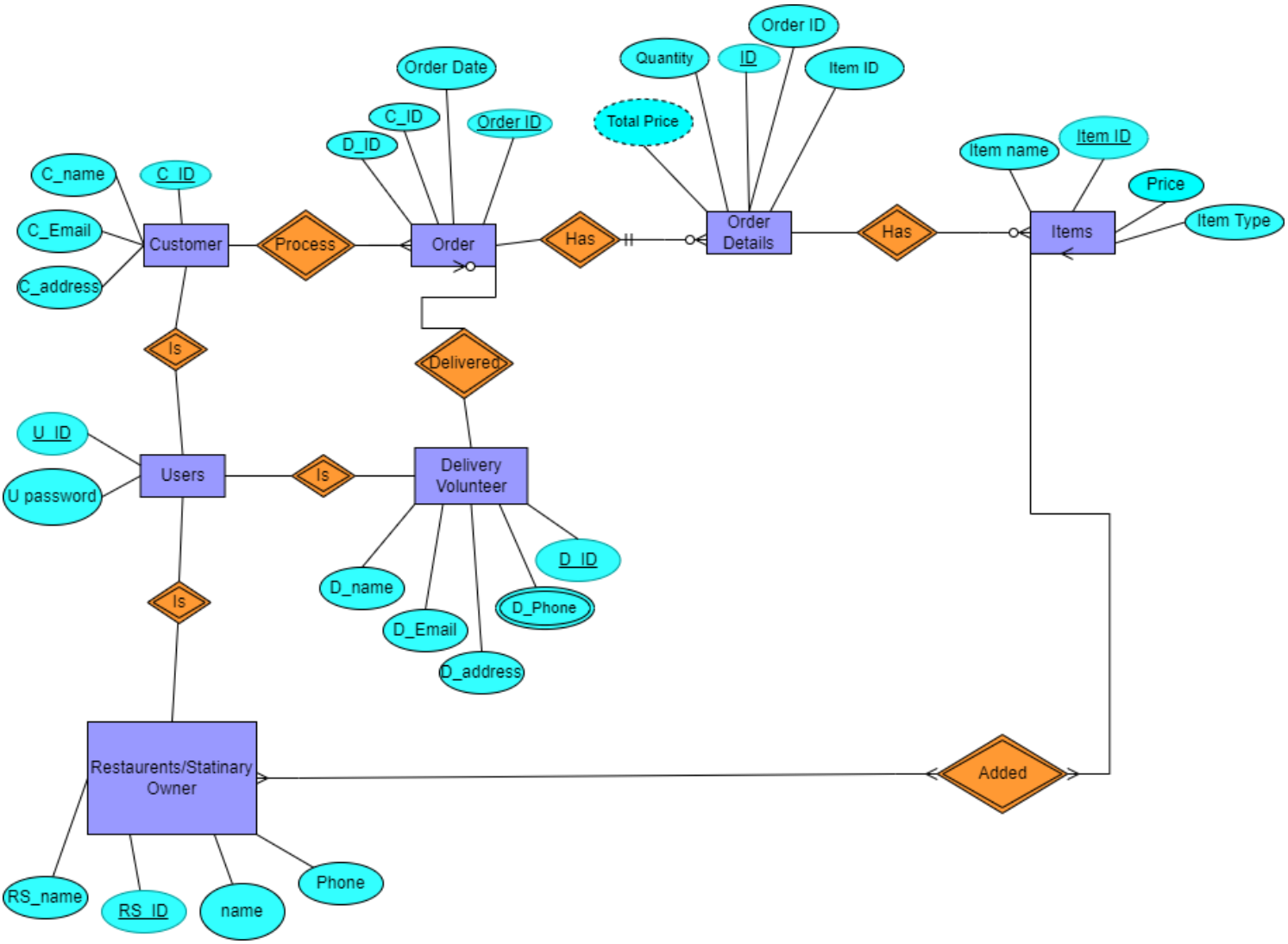
In this Entity Relation Diagram our entities are,

1. Users
2. Customers
3. Delivery Volunteers
4. Restaurants/stationary owner
5. Order
6. Order Details
7. Items

Attributes:

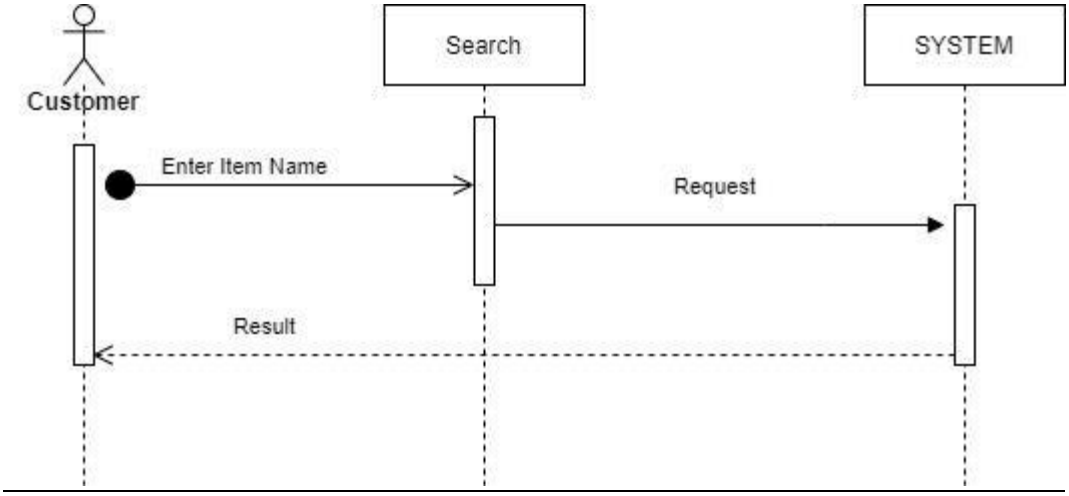
1. Users (U_id (primary key), U password)
2. Customer - (C_ID(primary key), C_name, C_address,, C_email)
3. Delivery Volunteers (D_ID(primary key), D_name, D_address, D_email)
4. Restaurants/stationary owner (RS_name, RS_ID(primary key), name, phone)
5. Order (Order_id(primary key), C_id(foreign key), D_id(foreign key), C_id(foreign key), Order Date)
6. Order Details(ID(primary key), item_id(foreign key), order_id(foreign key),quantity, total price)
7. Items (item_id(primary key), item_name, item type, price)

ID. 203-35-3137
ID. 203-35-3146

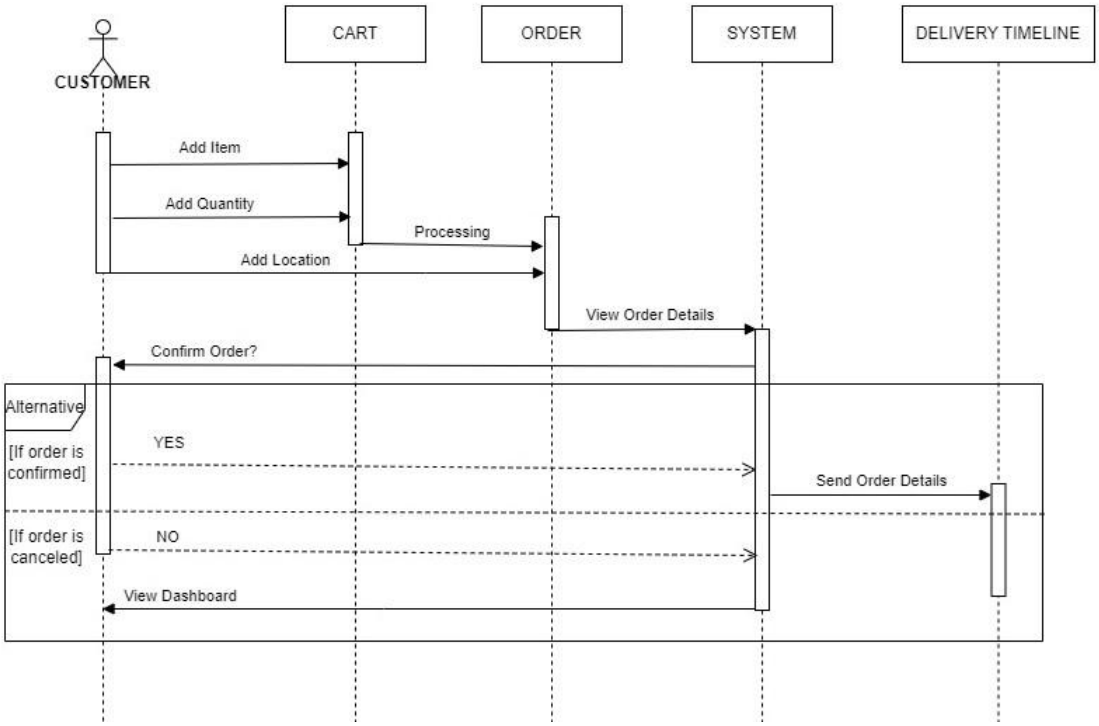


3.5 Sequence Diagram

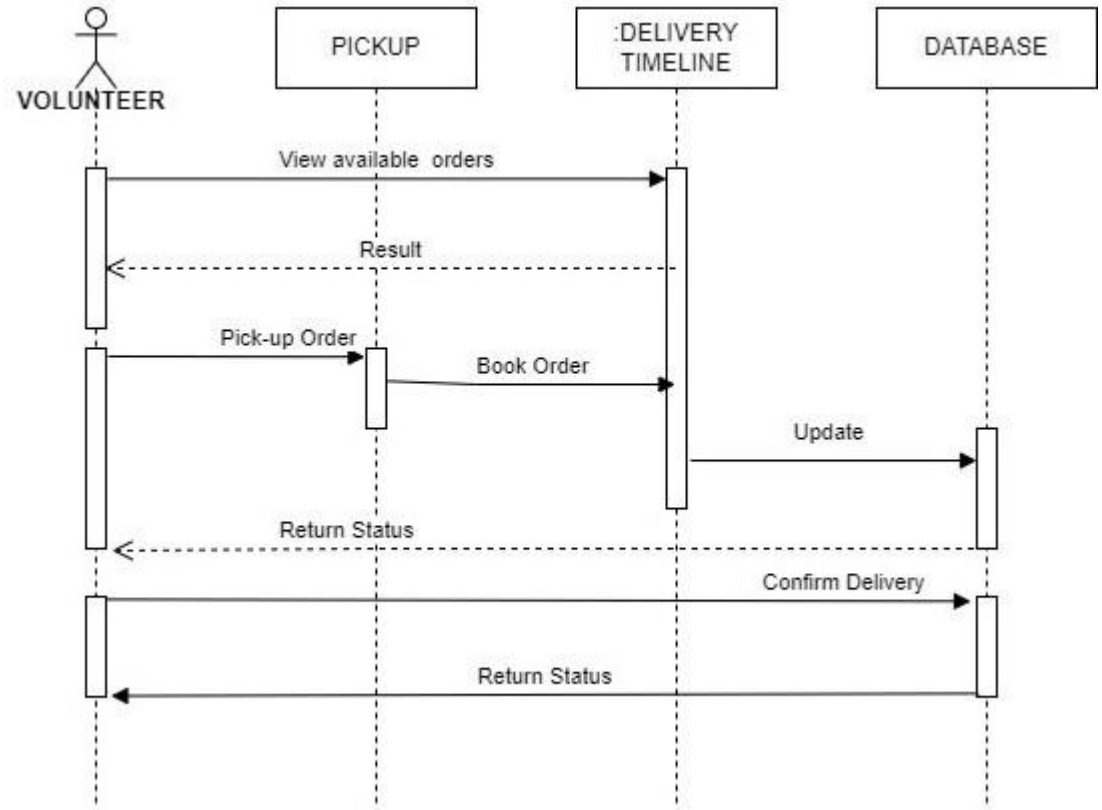
Search



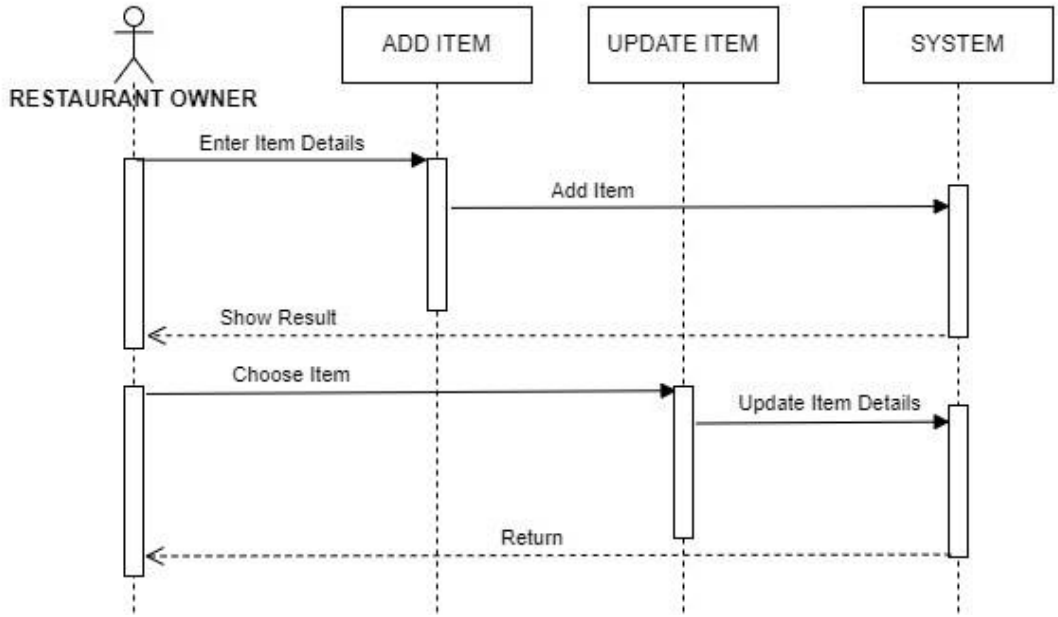
Order



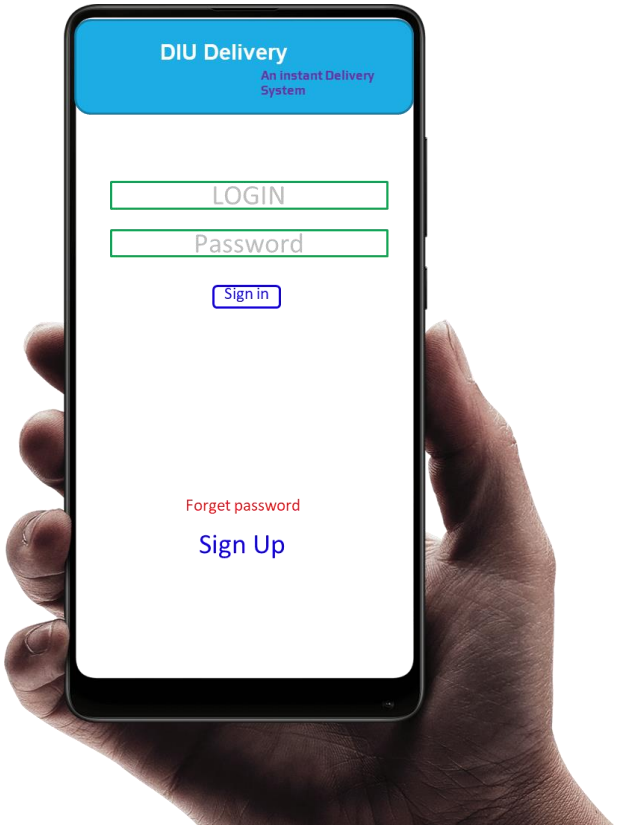
Delivery

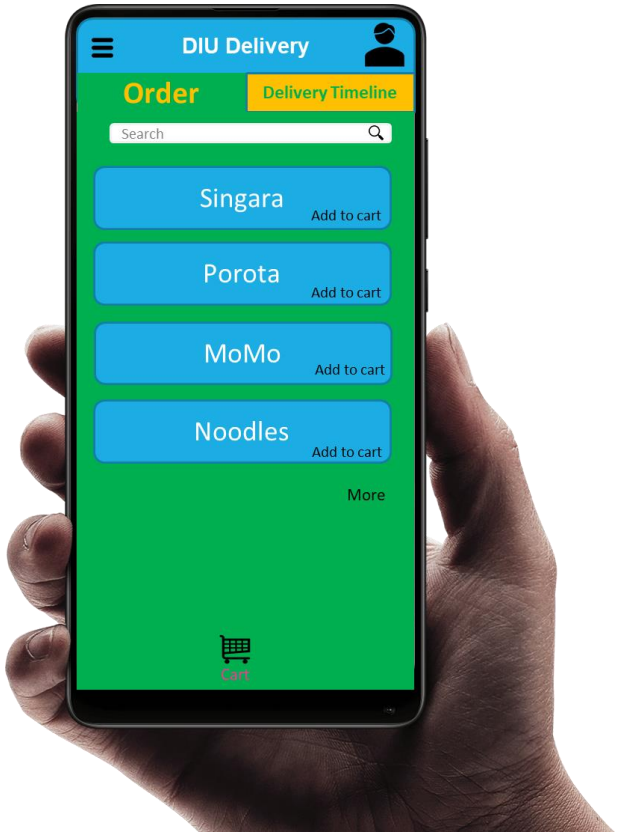


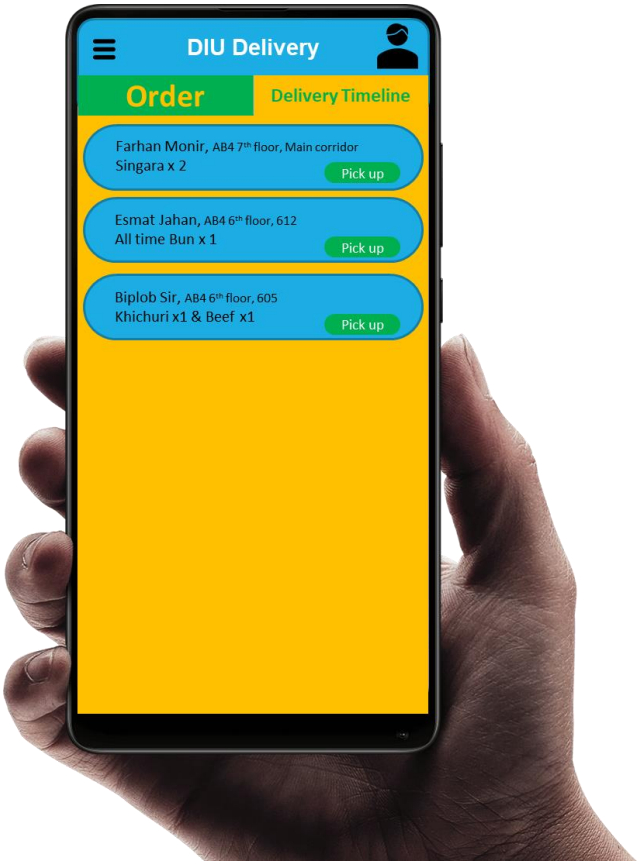
ADD ITEM

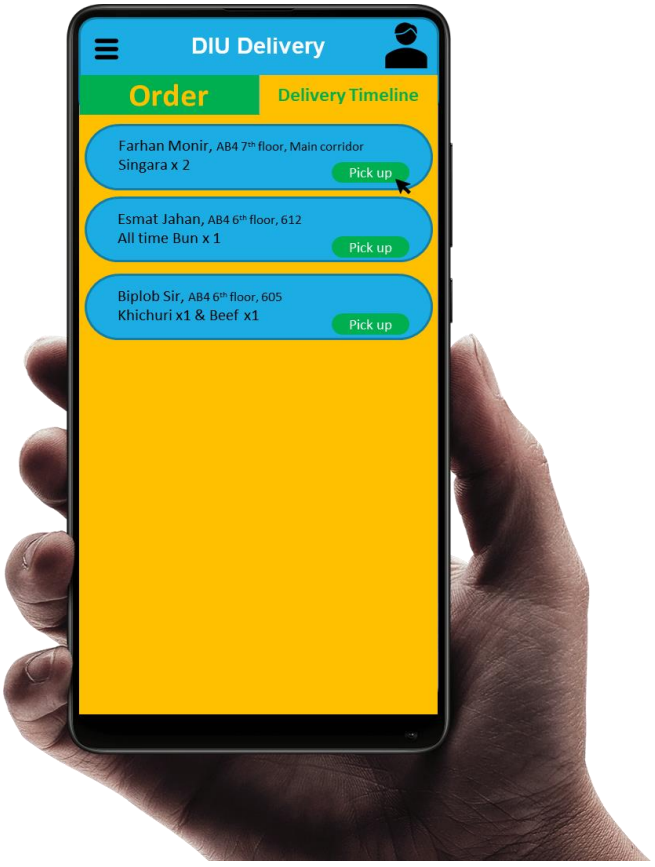


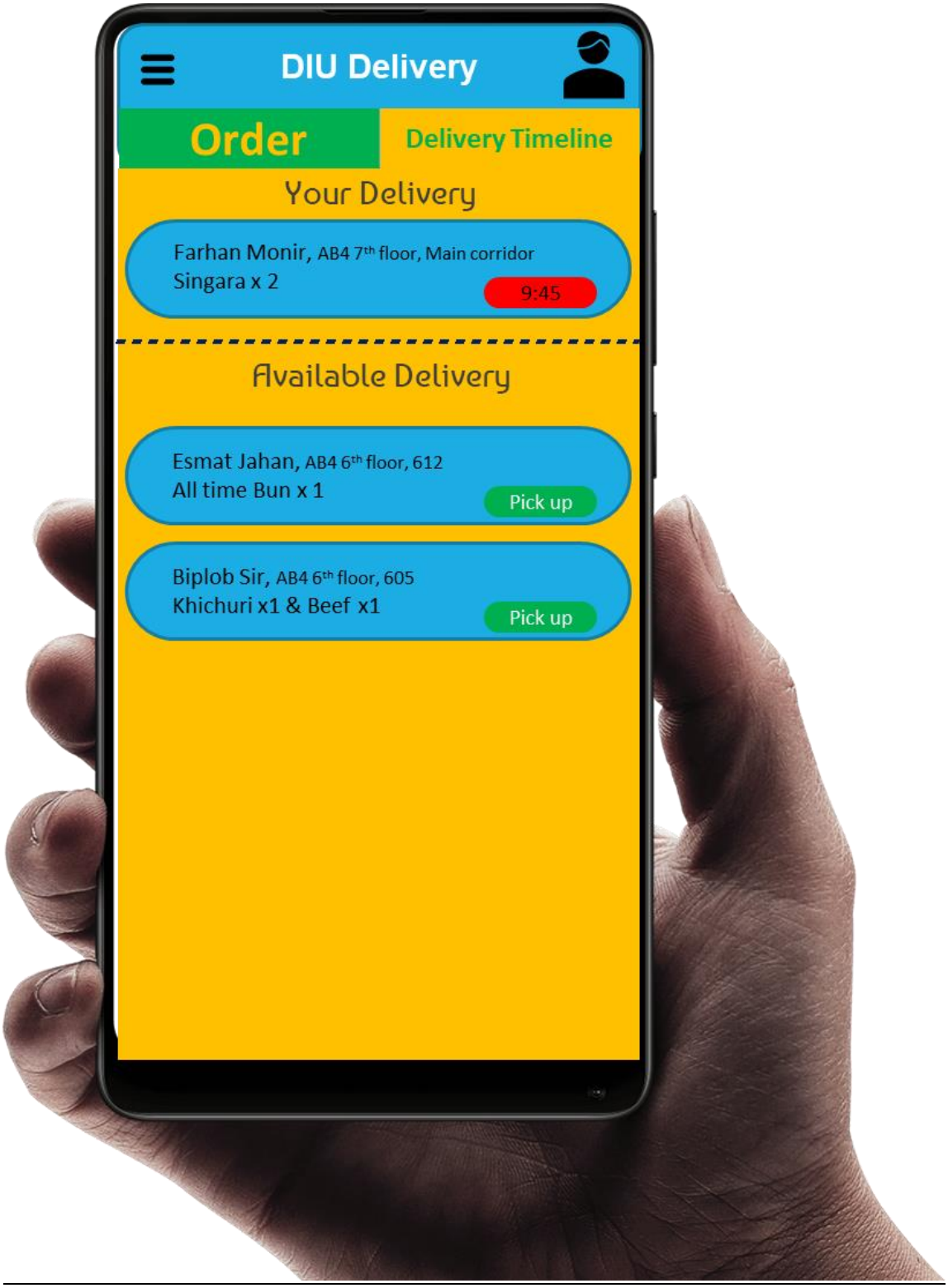
3.6 Prototype


DIU Delivery
An instant Delivery System
LOGIN
Password
Sign in
Forget password
Sign Up


DIU Delivery
Order
Delivery Timeline
Search
Singara
Add to cart
Porota
Add to cart
MoMo
Add to cart
Noodles
Add to cart
More
Cart


DIU Delivery
Order
Delivery Timeline
Farhan Monir, AB4 7th floor, Main corridor
Singara x 2
Pick up
Esmat Jahan, AB4 6th floor, 612
All time Bun x 1
Pick up
Biplob Sir, AB4 6th floor, 605
Khichuri x1 & Beef x1
Pick up


DIU Delivery
Order
Delivery Timeline
Farhan Monir, AB4 7th floor, Main corridor
Singara x 2
Pick up
Esmat Jahan, AB4 6th floor, 612
All time Bun x 1
Pick up
Biplob Sir, AB4 6th floor, 605
Khichuri x1 & Beef x1
Pick up


DIU Delivery
Order
Delivery Timeline
Your Delivery
Farhan Monir, AB4 7th floor, Main corridor
Singara x 2
9:45
Available Delivery
Esmat Jahan, AB4 6th floor, 612
All time Bun x 1
Pick up
Biplob Sir, AB4 6th floor, 605
Khichuri x1 & Beef x1
Pick up

Chapter 4

Conclusion

The application, DIU delivery software is a fast delivery system software inside Daffodil International University to deliver food and other stationery items to students and teachers. As Daffodil International University is the largest private green university in Bangladesh so sometimes it is tough for teachers and students to eat food due to time shortages during rush hours. It is difficult to concentrate in class with hunger. And sometimes you need stationery items but you can not go out of class or you have back-to-back classes. So here is our software to mitigate this problem along with the stationery items problem. When they order, students who are just now entering university can just buy food or stationery items and deliver them to the customers. Besides this system is helpful for some of the students working as delivery volunteers because they can earn some money for their day-to-day work. All these problems will be solved with just the help of our application; an instant delivery system where everyone will be benefited.

This documentation will help users and stakeholders to understand what this project is about and how to use it behind the diagrams of this application.