Mark

西北工業大學
Northwestern Polytechnical University

Software Engineering Project Report

Title of project	NPU Restaurant
Student Name	Khan Md Shahedul Islam, Abid Ali, Tafsir Mahmud
Student ID	2018380130, 2019380141, 2019380179
Submit Date	24/05/2022

Directory

/*Template user guide：
This template can be used for documenting a complete software development project.
Please extend or tailor it according to the practical software size or special requirement.
But the following sections marked with "*"must be filled:

*/

Chapter 1 Development plan

1.1 Introduction

1.1.1 Purpose

We are bringing our new ideas putting in a core java app to deliver an online food parceling system. We have designed and implemented unique and easier user interfaces for faster access.

1.1.2 Scope of Project

With our app anyone can surf and order all of our foods. The checkout procedure is easy and done within a click. We have included our full restaurant menu and admin support as an extension to our app.

1.2 Developing tools and environment

1.2.1 Hardware

Impacting hardware details sequentially for team lead and each member: -

- Processor: Ryzen R7 3700x 8 core and 16 threads with 3.6GHz base and up to 4.4GHz boost clock speed, core i5 8th gen 8260U 4cores and 8 threads with 3.90GHz max turbo clock speed, core i5 6th gen 6350hq 4 core 4 threads with 3.20 GHz max turbo frequency clock speed.
- Disk Space: 50 MB utilized (Samsung 970 EVO 256 M.2 nvme ssd, Seagate 1 TB sata hdd, Toshiba 500 gb hdd)
- RAM: 16 GB 3200 MHz ddr4, 8 GB 2400 MHz ddr4, 8 gb 2666 MHz

1.2.2 Software

- content Operating System: For development we used windows 11(x64), Windows 10(x64), Windows 10(x64)
- Software: eclipse, MySQL workbench
- Eclipse IDE and IntelliJ

- MySQL Workbench:
  MySQL Workbench is a unified visual tool for database architects, developers, and DBAs. MySQL Workbench provides data modeling, SQL development, and comprehensive administration tools for server configuration, user administration, and much more. MySQL Workbench is available on Windows, Linux and Mac OS.Library-rs2ml.jar
- mysql-connector-java-8.0.22.jar

- Java JDK 16.0.1

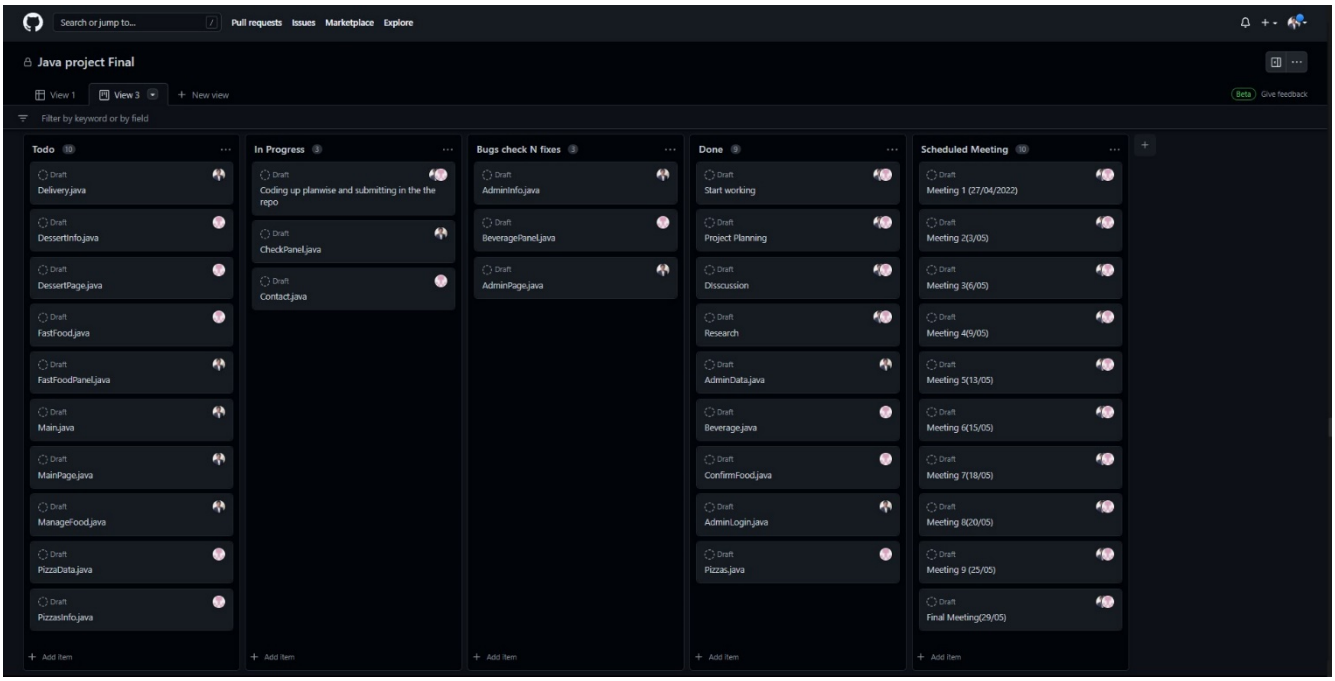
1.2.3 Other

- Github
- Tencent Meeting
- Discord

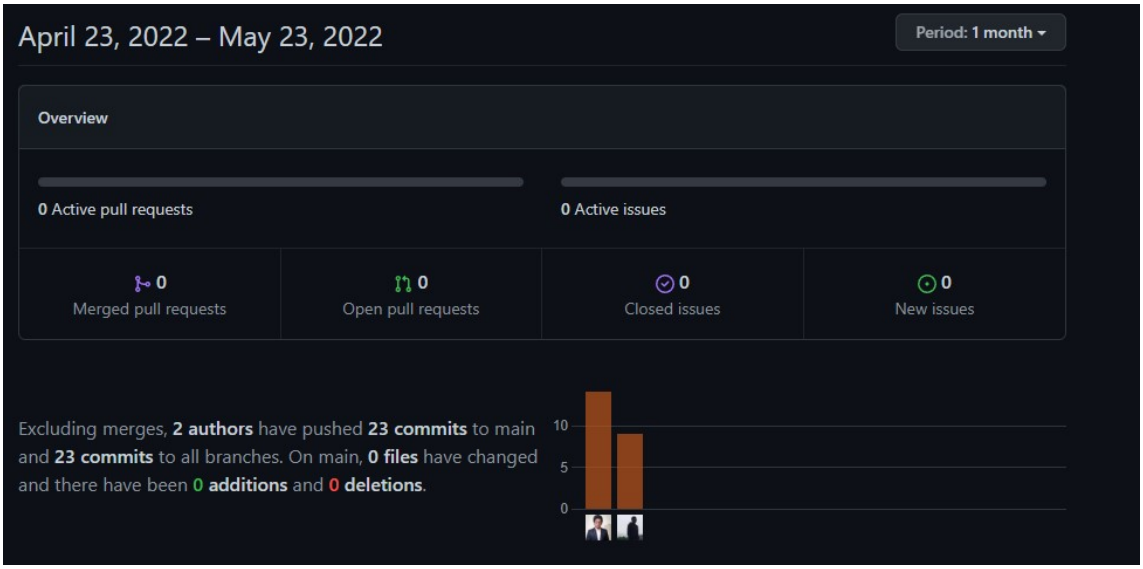
1.3 Project Management

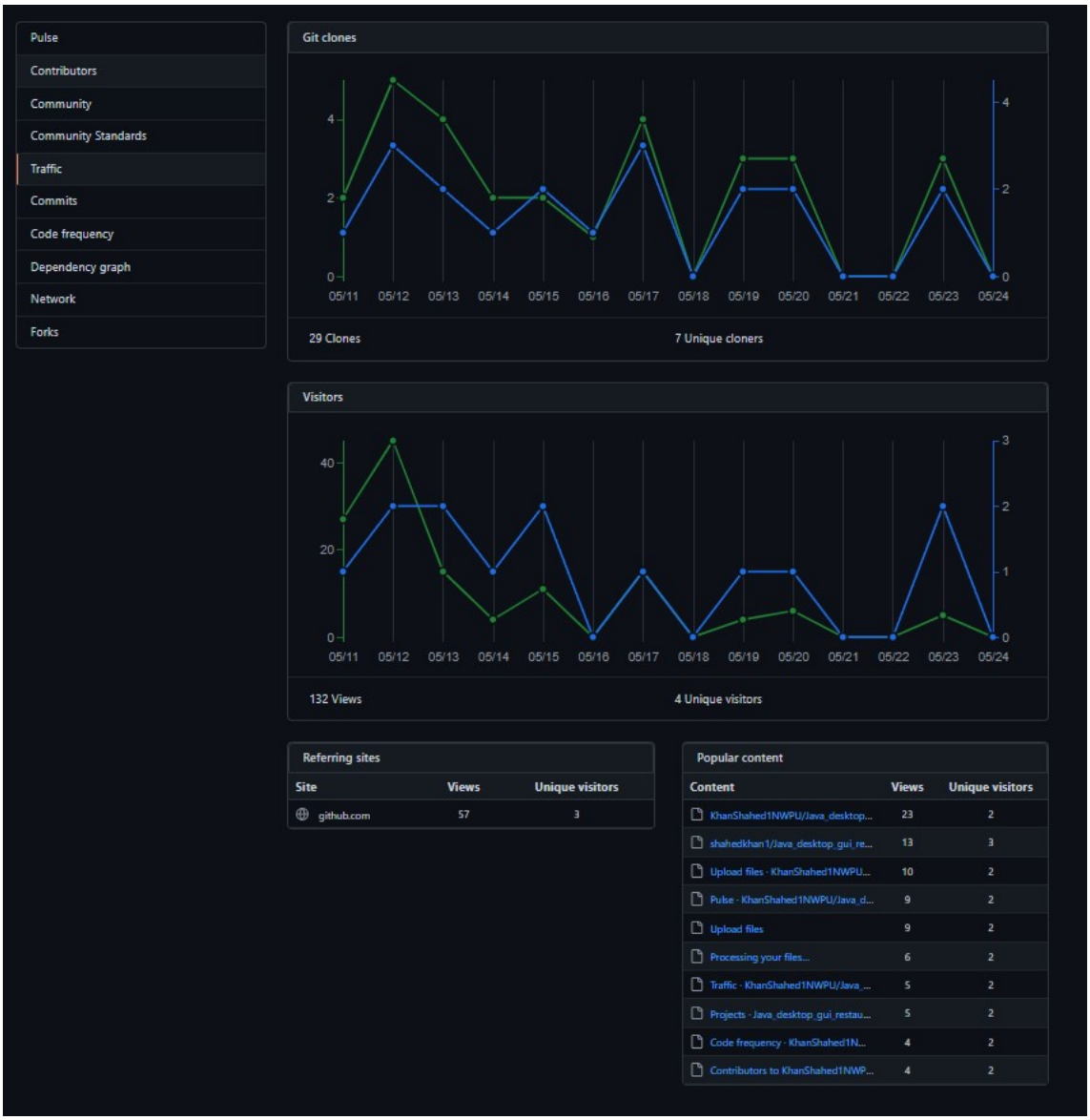
1.3.1 Schedule

Team lead had meeting after setting up the team and discussed in depth about the project outline functionalities and more importantly schedules. All together we came up with a schedule the we strictly followed and it was controlled via GitHub project.



	Title	Assignees	Status
1	Start working	abidnpu141 and Kh...	Done
2	Project Planning	abidnpu141 and Kh...	Done
3	Disscussion	abidnpu141 and Kh...	Done
4	Research	abidnpu141 and Kh...	Done
5	AdminData.java	KhanShahed1NWPU	Done
6	Beverage.java	abidnpu141	Done
7	ConfirmFood.java	abidnpu141	Done
8	AdminLogin.java	KhanShahed1NWPU	Done
9	Pizzas.java	abidnpu141	Done
10	Coding up planwise and submitting in the the repo	abidnpu141 and Kh...	In Progress
11	CheckPanel.java	KhanShahed1NWPU	In Progress
12	Contact.java	abidnpu141	In Progress
13	AdminInfo.java	KhanShahed1NWPU	Bugs check N fixes
14	BeveragePanel.java	abidnpu141	Bugs check N fixes
15	AdminPage.java	KhanShahed1NWPU	Bugs check N fixes
16	Delivery.java	KhanShahed1NWPU	Todo
17	DessertInfo.java	abidnpu141	Todo
18	DessertPage.java	abidnpu141	Todo
19	FastFood.java	abidnpu141	Todo
20	FastFoodPanel.java	KhanShahed1NWPU	Todo
21	Main.java	KhanShahed1NWPU	Todo
22	MainPage.java	KhanShahed1NWPU	Todo
23	ManageFood.java	KhanShahed1NWPU	Todo
24	PizzaData.java	abidnpu141	Todo
25	PizzasInfo.java	abidnpu141	Todo
26	Meeting 1 (27/04/2022)	abidnpu141 and Kh...	Scheduled Meeting
27	Meeting 2(3/05)	abidnpu141 and Kh...	Scheduled Meeting
28	Meeting 3(6/05)	abidnpu141 and Kh...	Scheduled Meeting
29	Meeting 4(9/05)	abidnpu141 and Kh...	Scheduled Meeting
30	Meeting 5(13/05)	abidnpu141 and Kh...	Scheduled Meeting
31	Meeting 6(15/05)	abidnpu141 and Kh...	Scheduled Meeting
32	Meeting 7(18/05)	abidnpu141 and Kh...	Scheduled Meeting
33	Meeting 8(20/05)	abidnpu141 and Kh...	Scheduled Meeting
34	Meeting 9 (25/05)	abidnpu141 and Kh...	Scheduled Meeting
35	Final Meeting(29/05)	abidnpu141 and Kh...	Scheduled Meeting


April 23, 2022 – May 23, 2022
Period: 1 month
Overview
0 Active pull requests
0 Active issues
0 Merged pull requests
0 Open pull requests
0 Closed issues
0 New issues
Excluding merges, 2 authors have pushed 23 commits to main and 23 commits to all branches. On main, 0 files have changed and there have been 0 additions and 0 deletions.
10
5
0


Pulse
Contributors
Community
Community Standards
Traffic
Commits
Code frequency
Dependency graph
Network
Forks
Git clones
4
2
0
05/11 05/12 05/13 05/14 05/15 05/16 05/17 05/18 05/19 05/20 05/21 05/22 05/23 05/24
29 Clones
7 Unique cloners
Visitors
40
20
0
3
2
1
0
05/11 05/12 05/13 05/14 05/15 05/16 05/17 05/18 05/19 05/20 05/21 05/22 05/23 05/24
132 Views
4 Unique visitors
Referring sites
Site Views Unique visitors
github.com 57 3
Popular content
Content Views Unique visitors
KhanShahed1NWPU/Java_desktop... 23 2
shahedkhan1/Java_desktop_gui_re... 13 3
Upload files · KhanShahed1NWPU... 10 2
Pulse · KhanShahed1NWPU/Java_d... 9 2
Upload files 9 2
Processing your files... 6 2
Traffic · KhanShahed1NWPU/Java_... 5 2
Projects · Java_desktop_gui_restau... 5 2
Code frequency · KhanShahed1N... 4 2
Contributors to KhanShahed1NWP... 4 2

1.3.2 Team Organization and Responsibilities

Team Leader Khan Md Shahedul Islam (2018380130):
Implementing structure design, working on classes, interface design, Code review, UML diagram design, bug fixing, system architecture designing, maintaining GitHub repository, creating data base, report writing.

Member1 Abid Ali (2019380141):
Implementing structure design, working on classes, interface design, Code review, UML diagram design, bug fixing, flowchart designing, ER diagram designing, creating data base, report writing.

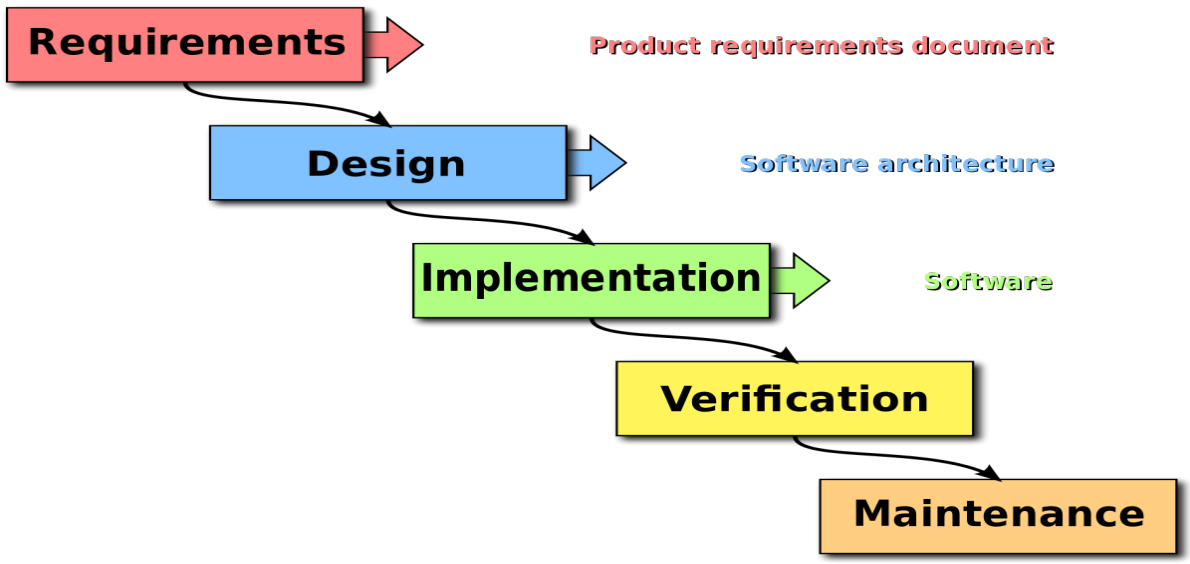
Member2 Tafsir (2019380179):
He joined the group late due to his two surgeries so he was given in charge of simple task such as interface design, Code review, Activity diagram, UML diagram design, Bug fixing, creating data base, report writing. He was given testing task for the project.

1.4 Software life cycle model

For software application development, we should follow Software Development Life Cycle (SDLC).

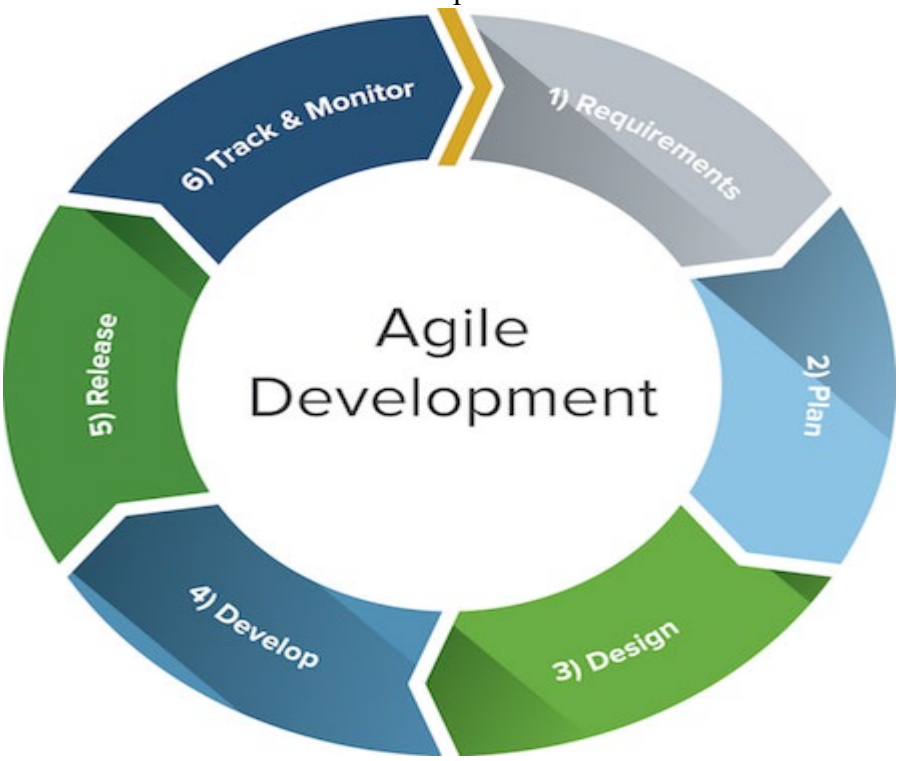
WaterFall Model

A typical waterfall model is a 5-step software development process which are ideation, design, implementation, testing, and deployment.
The method works in linear order. We should complete each individual process in order to start the next process.



WaterFall Model

Agile Model
Agile Methodology is widely used in modern software application development. It is an iterative and incremental development w.r.t. SDLC.



Agile Model

Agile Development" is an umbrella term for several iterative and incremental software development methodologies. Scrum is one such framework within Agile. Scrum provides a lightweight process framework that embraces iterative and incremental practices, helping organizations deliver working software more frequently. Projects progress via a series of iterations called sprints; at the end of each sprint the team produces a potentially deliverable product increment.
'Agile Methodology for Zomato

1. Product Vision

1. Helping people discover great places around them.
2. Building amazing experiences around dining.
3. Enabling restaurants to create amazing experiences.

2. Creating Product Backlog

We need to create a Product Backlog where we can list down all the tasks to be done in the form of a user facing story, arranged in order of priority.

As a user, I want to discover restaurants in my city, so that I can decide where to dine or order from.

As a user, I want to see the ratings and reviews of restaurants, so that I can take an informed decision before going to the restaurant or ordering from it.
As a user, I want different filters to search restaurant, so that I can easily narrow down to the type of place I am interested in.
As a user, I want to order food of my choice from restaurants around me, so that I can enjoy a good meal without going to the restaurant
As a user I want an option to book tables for the restaurant of my choice, so that I don't have to wait when I visit the restaurant.

3. Creating Sprint Backlog

We selected to implement 1 epic from the Product Backlog — As a user, I want to discover restaurants in my city, so that I can decide where to dine or order from. This epic will be broken down into more specific user stories now. Example —

As a user, I want to see restaurants that are near my preferred location, so that I can decide where to dine or order from.
As a user, I want to see the top rated restaurants in my city, so that I can discover the most popular restaurants
As a user, I want to click on a restaurant, so that I can know more about the restaurant
As a user, I want to be suggested restaurants based on the time of day, so that I can plan the relative meals more easily

Chapter 2 Requirement analysis

2.1 Introduction

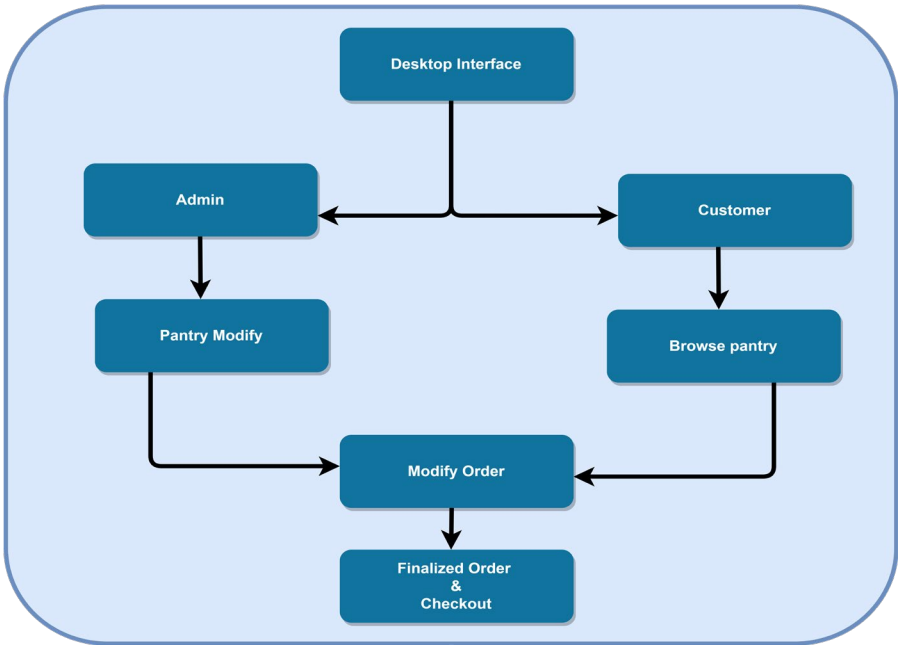
For our desktop app, we wanted our app to be very easy to use by both customers and admins. The checkout procedure is easy and done within a click. We have included our full restaurant menu and admin support as an extension to our app.

2.2 Functional Requirements

User:
A user can browse, place, cancel order. The system would be really user friendly, simple drag and drop to cancel. Can change quantity and/or modify orders during the checkout as well.
Admin:
Admins can easily access the database to change admin credentials(all the information including name, number, address, password).
Can modify food orders, modify food pantry(directly from app no need to access dabase additionally)

Chapter 3 System and function design

3.1 System architecture



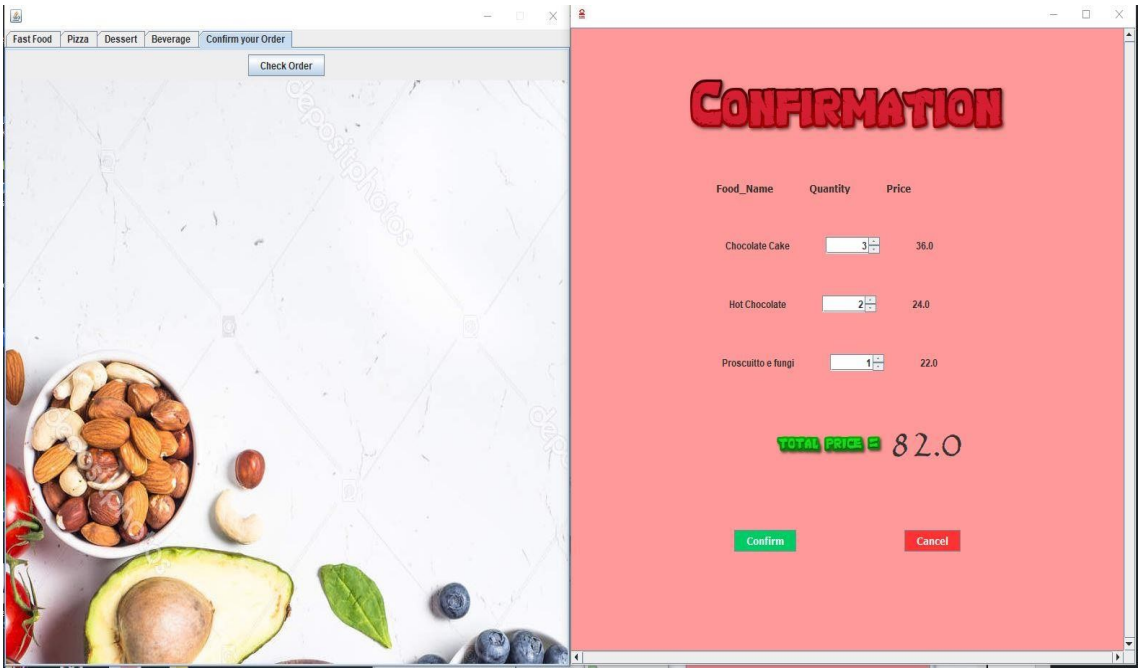
Our System architecture is illustrated through above.

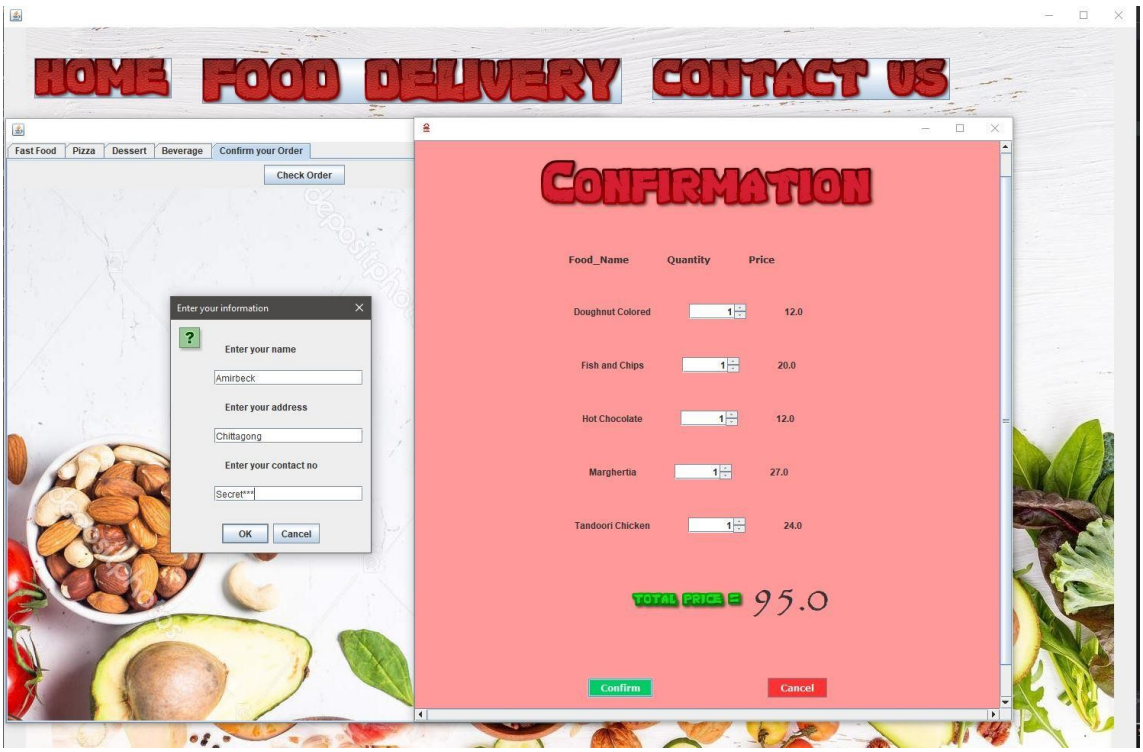
3.2 Functional Descriptions

3.2.1 Function1

Text description or use case diagram A user can access our food gallery from the app home page and can select any of the items we offer. Later moving on to check order option, user can add/drop, change quantity of the items and confirm the order after providing the user credentials (name, address, contact info).

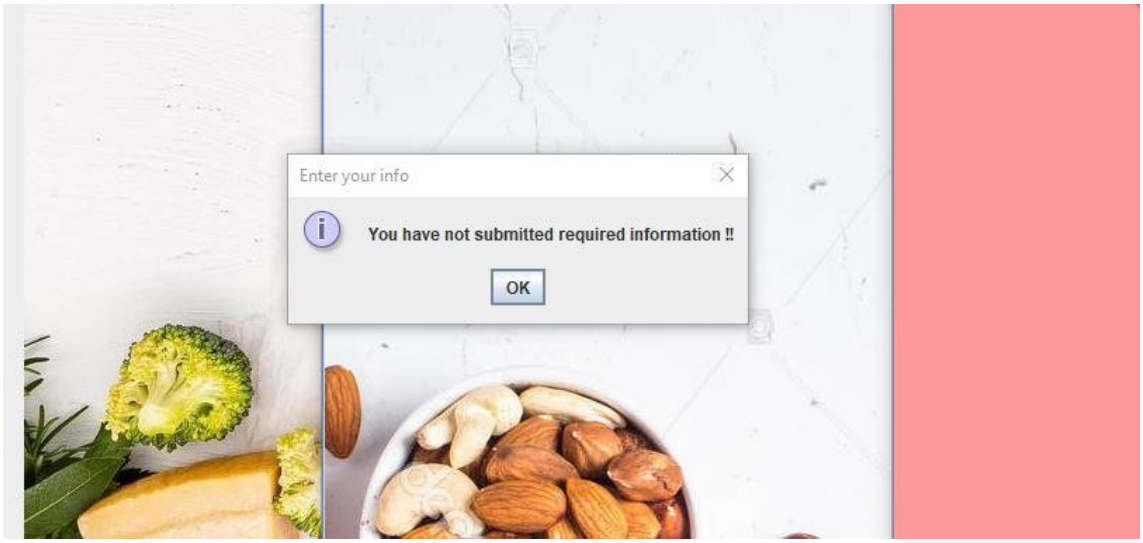

HOME
FOOD DELIVERY
CONTACT US
ADMIN


Fast Food
Pizza
Dessert
Beverage
Confirm your Order
Check Order
CONFIRMATION
Food_Name
Quantity
Price
Chocolate Cake
3
36.0
Hot Chocolate
2
24.0
Proscuitto e fungi
1
22.0
TOTAL PRICE = 82.0
Confirm
Cancel



3.2.2 Function2

If user does not input all the required information the system will give a pop-up reminding that.

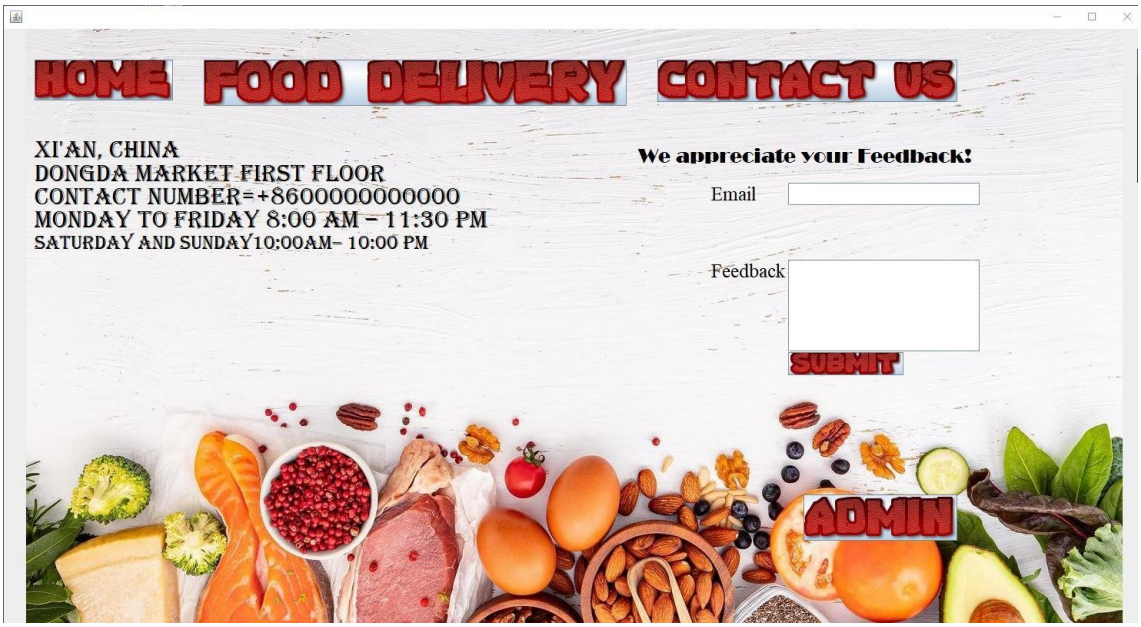


3.2.3 Function3

One click order cancellation

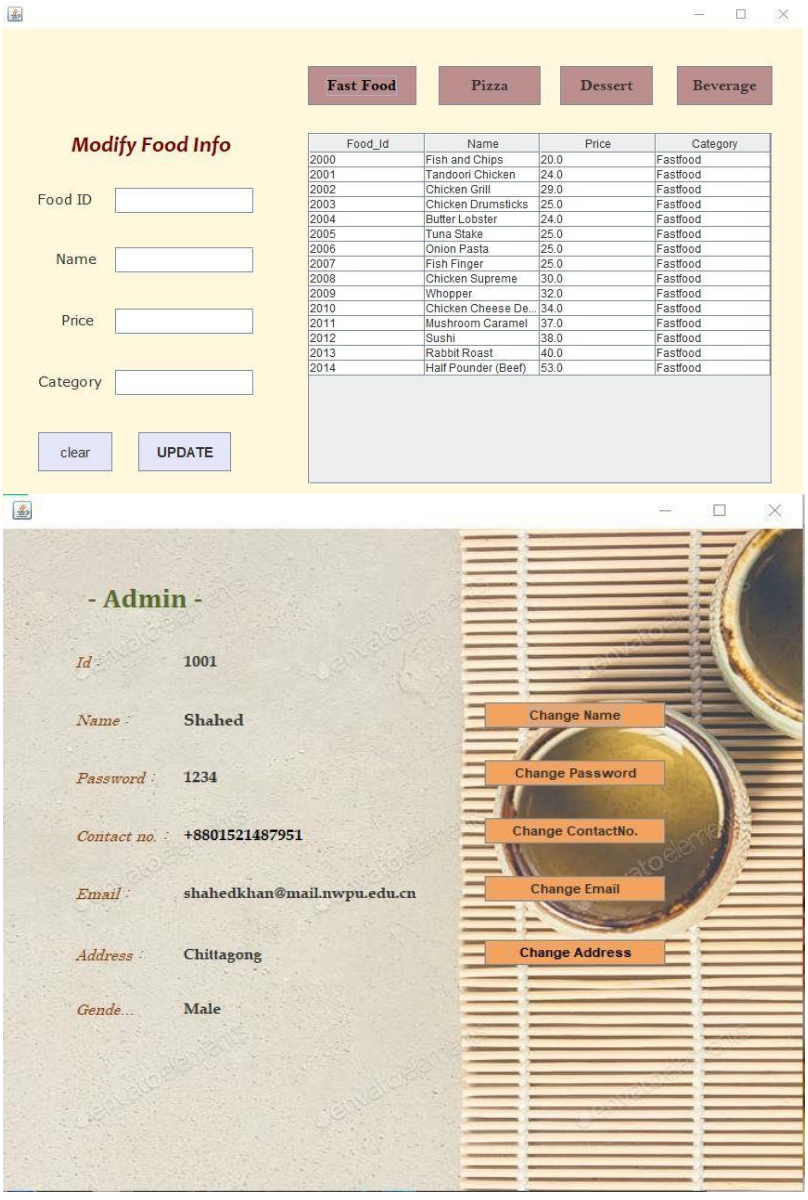
3.2.4 Function4

We have provided easy to interact system between system managers and users to help us grow and new customers to decide on foods. We have a contact us functionality embedded in the home page of our app for that.



3.2.5 Function5

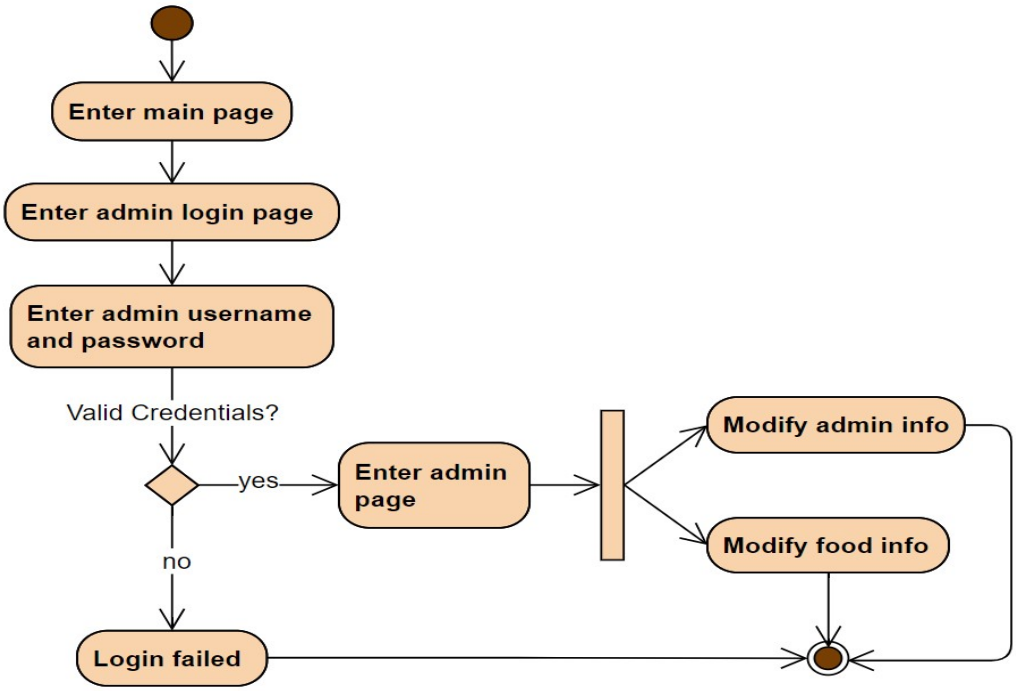
App managers/admins can easily access the database to change admin credentials (all the credentials), update foods (price, name, category, Id in the database)



3.3 Package/Module diagram

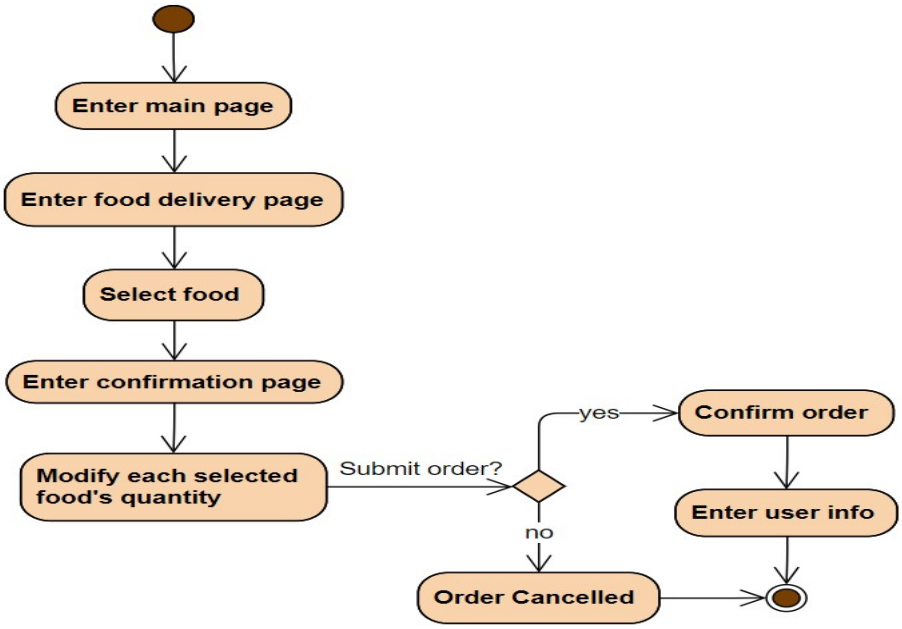
3.3.1 Activity Diagram (User)

Shows all the steps users can follow to confirm/cancel order after selection.



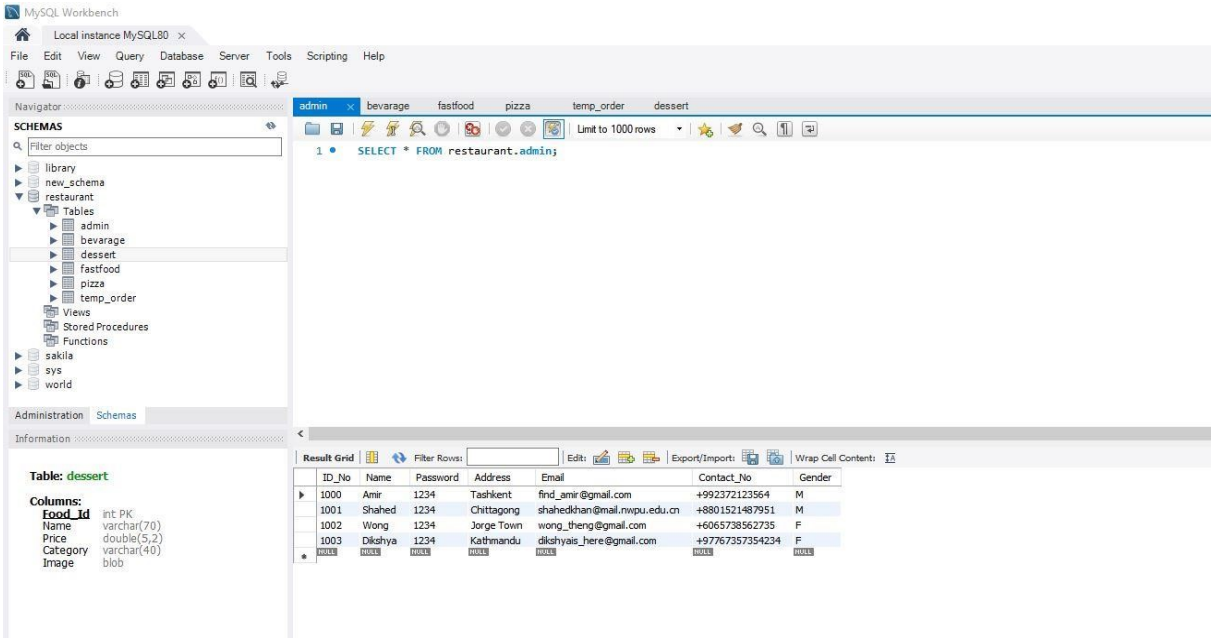
3.3.2 Activity Diagram (Admin)

Shows all the steps admins need to follow to modify admin credential or update/modify food database.

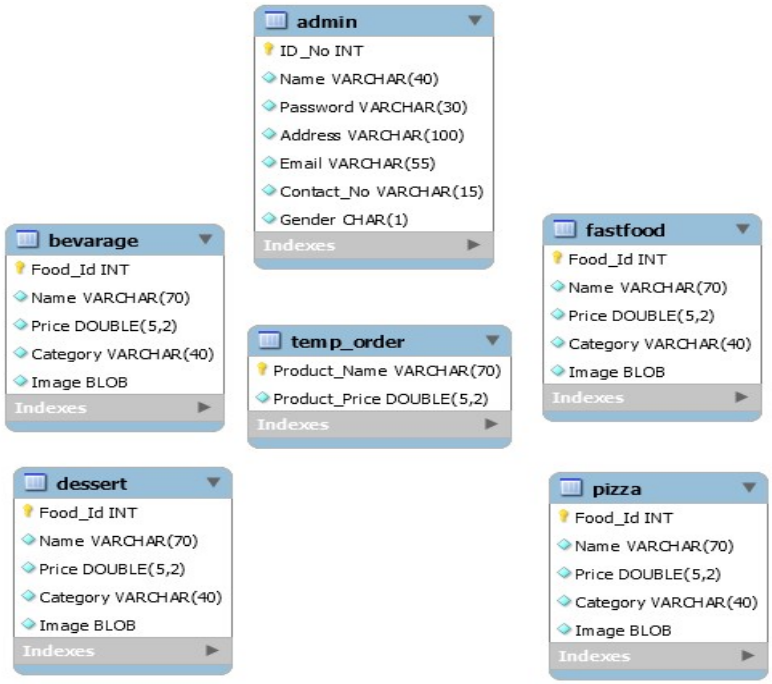


3.4 Data description

Affiliated database: - In addition to textual data, in our database we have imported all the images for our available foods. We went with BLOB (64kb limit) as it was the best fit for our combination.



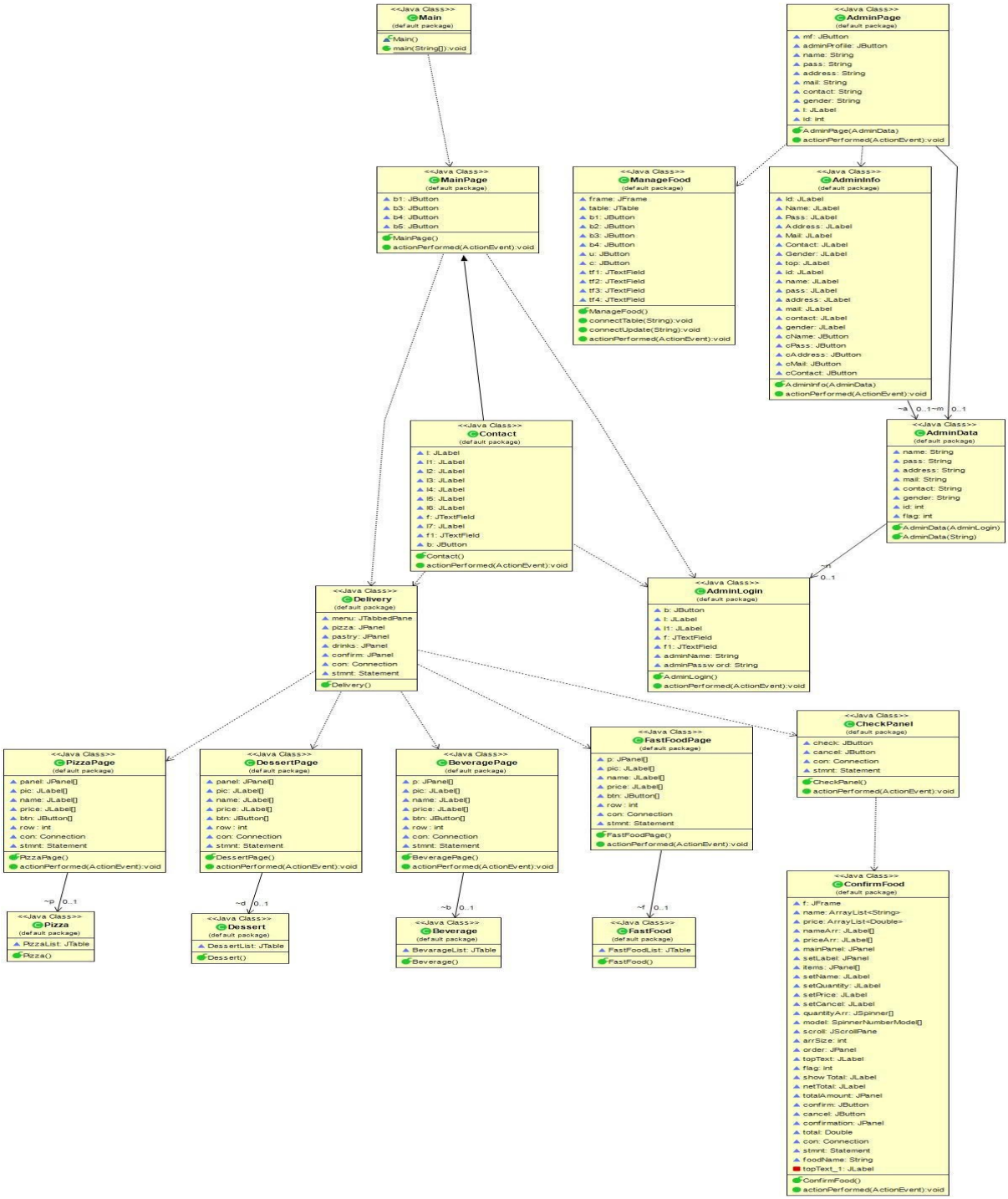
ER diagram



Chapter 4 Detail design

4.1 Class diagram

Below is given our app's class organization. From the diagram we can see different relations (Inheritance: contact us from main page, dependencies, directed association etc.). All the functions, structures used in each class can also be seen in the diagram.



4.2 ClassA/ModuleA

4.2.1 Variable list

AdminPage

Field Name	Data type	Size	Constants	Description	Extra
name	String	25	PRI	*Admin Name*	
pass	String	25	Not Null	Admin Password	
address	String	100	Not Null	Admin's Address	
Mail	String	20	Not Null	Admin's Maild	

AdminData

Field Name	Data type	Size	Constants	Description	Extra
name	String	25	PRI	*Admin Name*	
pass	String	25	Not Null	Admin Password	
address	String	100	Not Null	Admin's Address	
mail	String	20	Not Null	Admin's Mail	
contact	String	100	Not Null	Admin's Contact	
gender	String	20	Not Null	Admin's Data	
id	int	20	Not Null	ID of admin	
flag	int	20	Not Null	Flag	

CofirmFood

Field Name	Data type	Size	Constants	Description	Extra
name	String	25	Name	*Food Name*	
price	Int	25	Not Null	Food price	
setName	String	100	Not Null	Food's Name	
setQuantity	Int	20	Not Null	Food Quantity	
setPrice	String	100	Not Null	Price detail	
order	String	20	Not Null	Order list	
Total	Int	20	Not Null	Total cost	
confirm	String	20	Not Null	Confirmation	

PizzaPage

Field Name	Data type	Size	Constants	Description	Extra
name	String	25	Not Null	Name of dish	
price	Int	25	Not Null	Price of Pizza	

DessertPage

Field Name	Data type	Size	Constants	Description	Extra
name	String	25	Not Null	Name of dish	
price	Int	25	Not Null	Price of Dessert	

BeveragePage

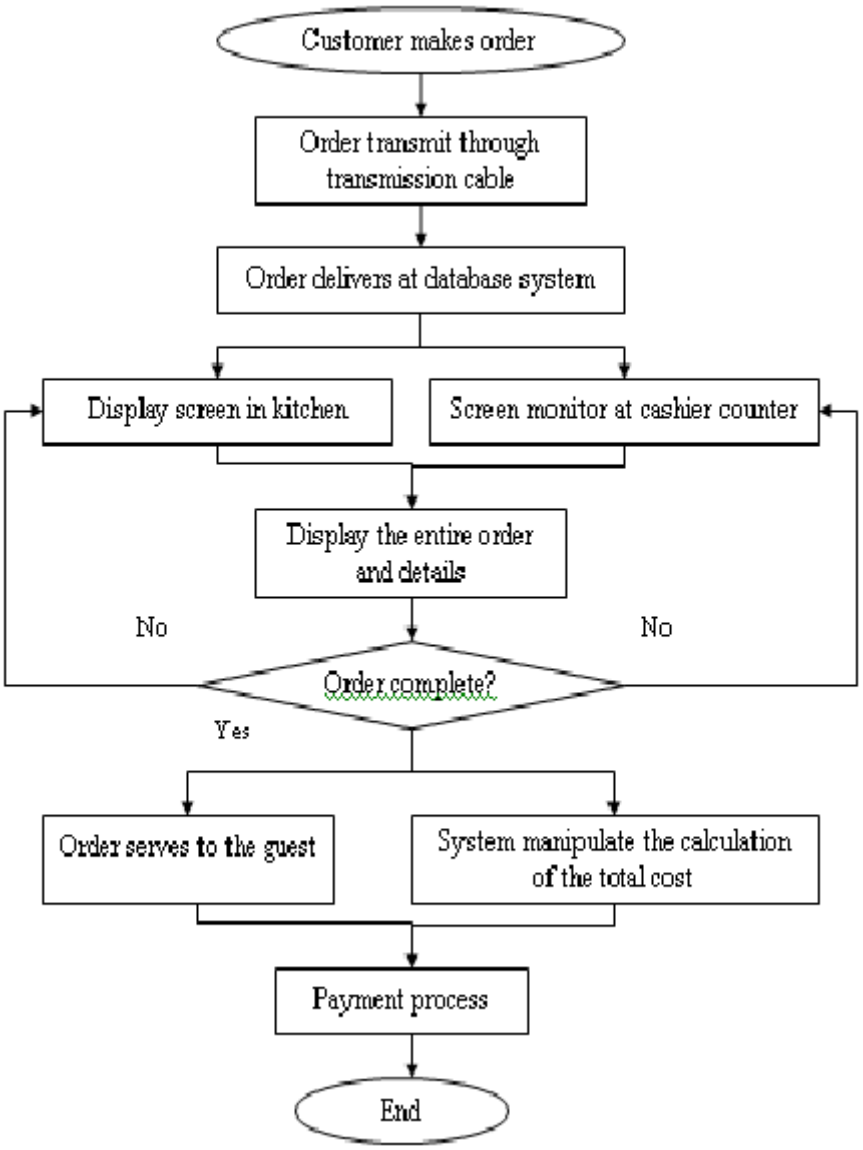
Field Name	Data type	Size	Constants	Description	Extra
name	**String**	**25**	**Not Null**	**Name of dish**	
price	**Int**	**25**	**Not Null**	**Price of Beverage**	

4.2.2 Function list

ConfirmFood, FastFood, Beverages, Dessert, Pizza, BeveragePage, PizzaPage, DessertPage, BeveragePage, FastFoodPage, CheckPanel, Delivery, AdminLogin, Contact, AdmnData, MainPage, ManadeFood, AdminPage, Main

Function Name	Meaning	Parameter	Data Type	Return Value
Beverages	Beverage option in menu	Select beverage, quantity	String, Int	Get beverage
ConfirmFood	Confirm food from menu	Click confirmation button.	String	Confirmation done
Pizza	Pizza option in menu	Select pizza, quantity	String, Int	Get pizza
BeveragePage	Open the page of beverage	Choose beverage	String	Beverage choosen
PizzaPage	Open the page of pizza	Choose pizza	String	Pizza choosen
DessertPage	Open the page of dessert	Choose dessert	String	Beverage dessert
Delivery	What will be delivered to customer and quatity of food	Customer's desired choice food and quantity	String, int	Delivering List Items
AdminPage	Admin login page	Type Admin's user name and admin's password	String, int	Admin homepage is opened

4.2.3 FunctionA



Chapter 5 Programming

5.1 Coding rules

Coding rules and guidelines ensure that software is:

- **Safe:** It can be used without causing harm.
- **Secure:** It can't be hacked.
- **Reliable:** It functions as it should, every time.
- **Testable:** It can be tested at the code level.

- **Maintainable:** It can be maintained, even as your codebase grows.
- **Portable:** It works the same in every environment.

5.2 Techniques in programming

Pair programming

Pair programming is an agile software development technique in which two programmers work together at one workstation. One, the driver, writes code while the other, the observer or navigator, reviews each line of code as it is typed in. The two programmers switch roles frequently.

While reviewing, the observer also considers the "strategic" direction of the work, coming up with ideas for improvements and likely future problems to address. This is intended to free the driver to focus all of their attention on the "tactical" aspects of completing the current task, using the observer as a safety net and guide.
Knowledge is constantly shared between pair programmers, whether in the industry or in a classroom. Many sources suggest that students show higher confidence when programming in pairs, and many learn whether it be from tips on programming language rules to overall design skill. In "promiscuous pairing", each programmer communicates and works with all the other programmers on the team rather than pairing only with one partner, which causes knowledge of the system to spread throughout the whole team. Pair programming allows programmers to examine their partner's code and provide feedback, which is necessary to increase their own ability to develop monitoring mechanisms for their own learning activities.

Kanban

We used Kanban method in our programming. Kanban manages workflow directly on the kanban board. The WIP limits for development steps provide development teams immediate feedback on common workflow issues.

5.3 Difficulties in programming

In this day and age of complete mobility, going out to eat is a hard option. But business and consumption have to go on. Thinking to tackle this Covid Pandemic era we have come out with our solution through this app. Users can access the previous reviews on app as well as visual representation of what they are going to get when they place orders as well.

1)Trying to create a desktop core java app we faced problem with GUI: Java Swing, AWT (First we were implementing design part manually, later we found out that there is drag drop option for design, which is more convenient and time saving.)

2)In our database we have imported all the images for our available foods. But it was

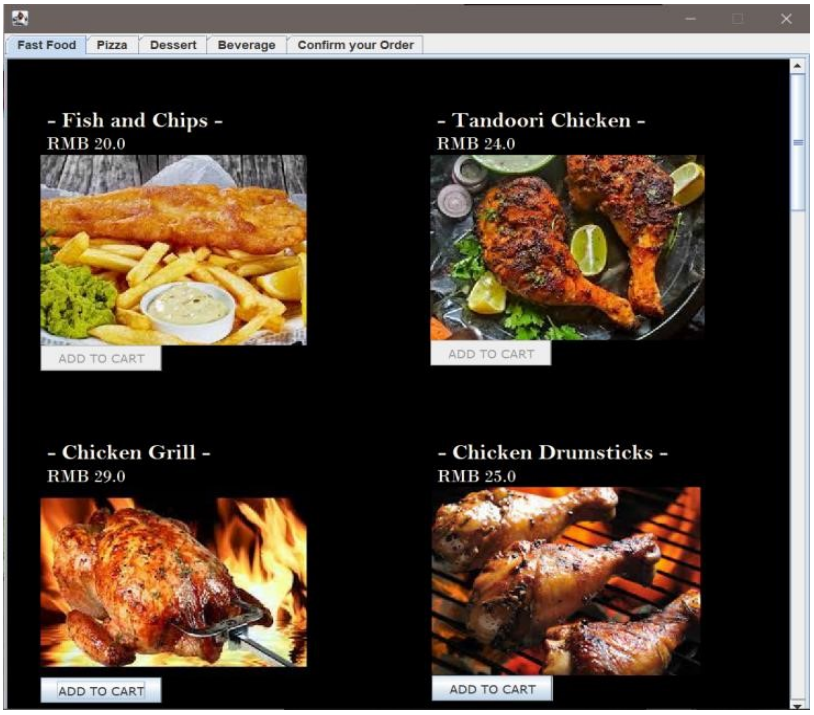
hard to import and later we went with BLOB(64kb limit) as it was the best fit for our combination.

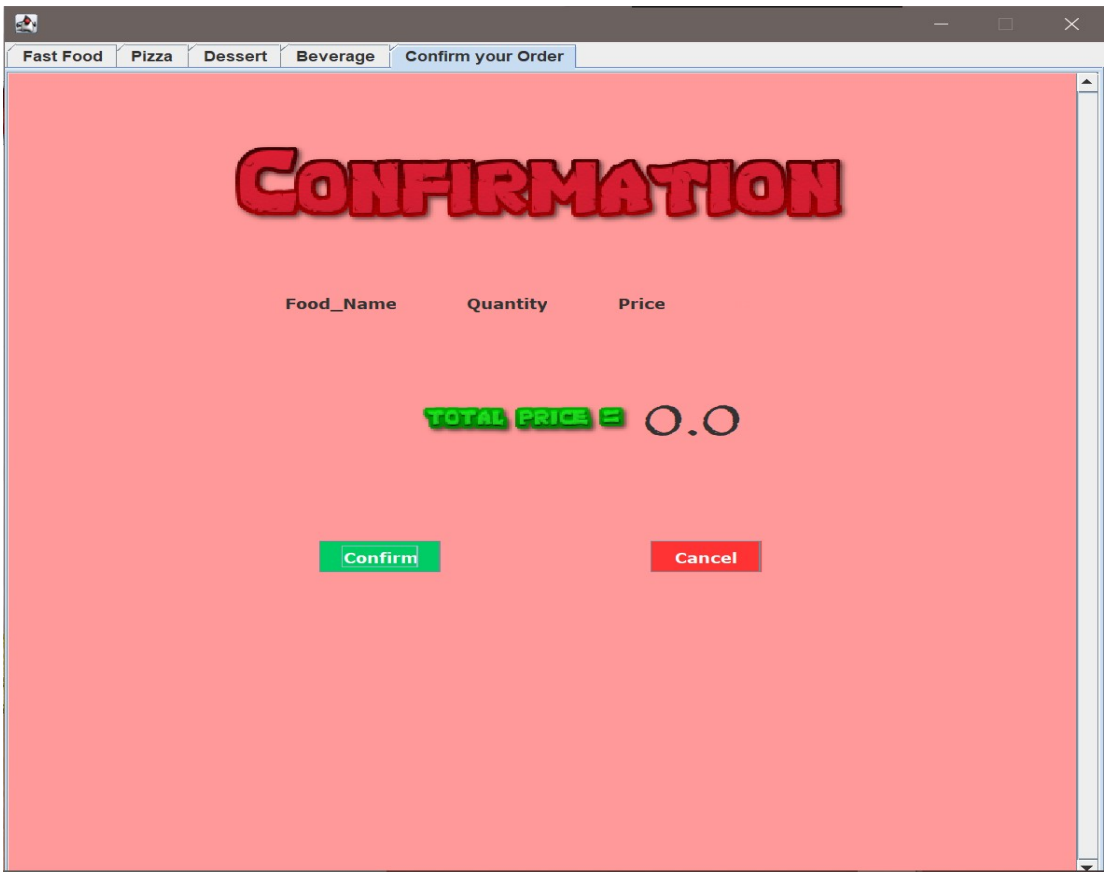
3) Structure: -

> We encountered some problems to implement structure according to our desired design (app interface and layout)

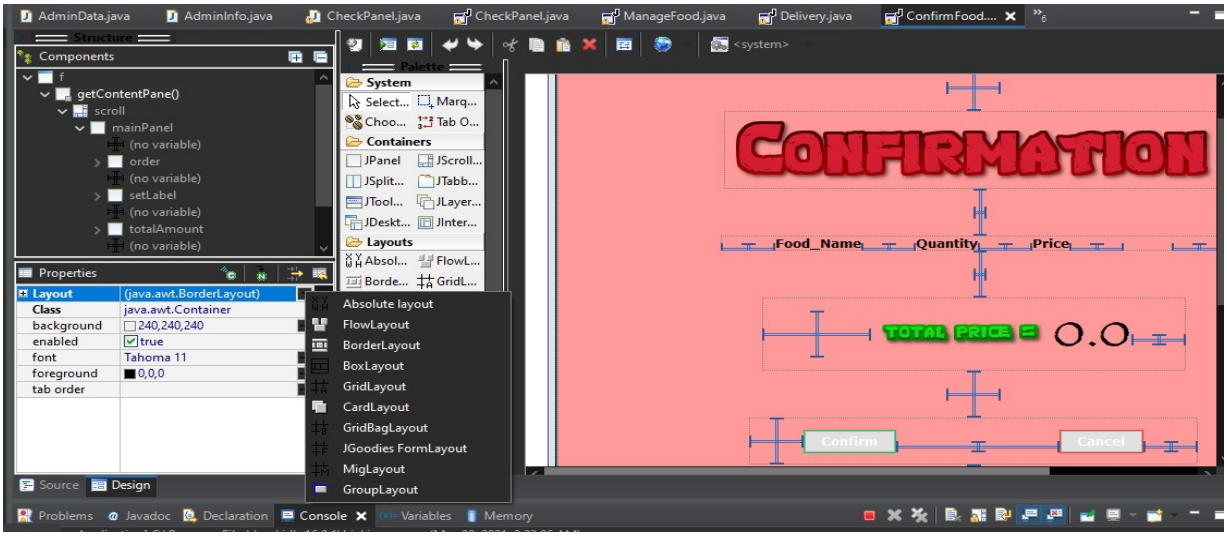
> We had issues creating admin page for the app and add data manipulation of data base through the app.

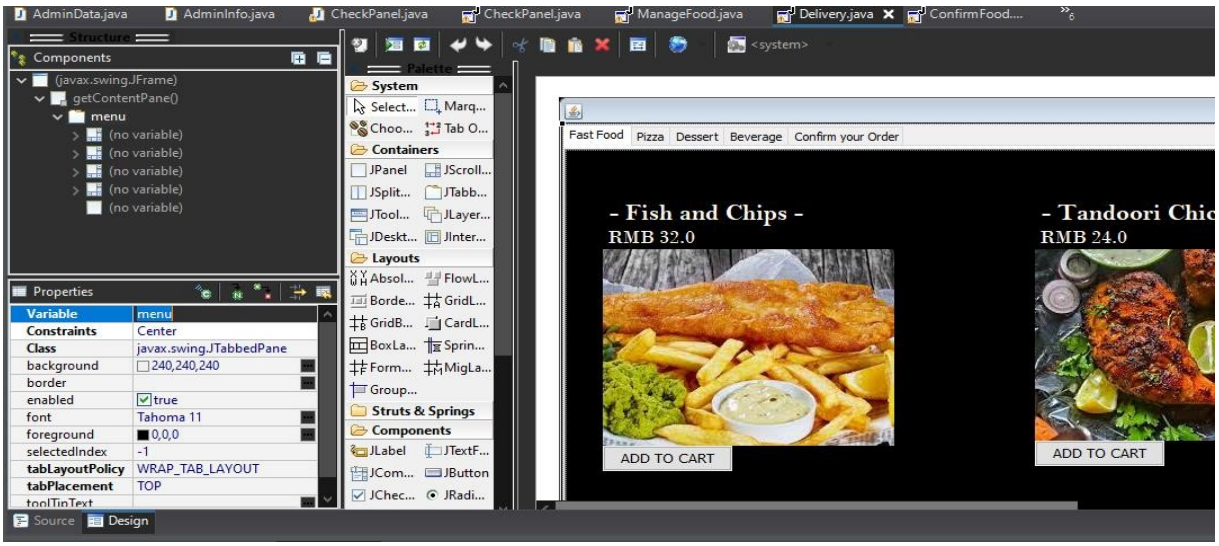
> We wanted to make our menu appear like this following picture, but we were having troubles with the synchronization(input) of information in order confirmation page.

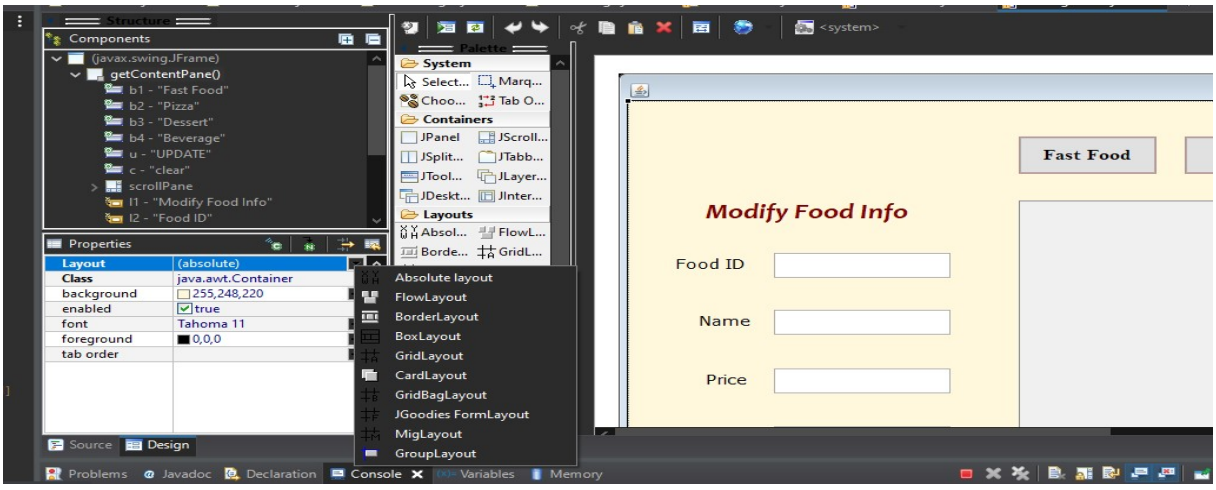




> Configuring Layout was a bit hard (we didn't know priorly where to apply which one) as the implication was unknown to us and we had use different layouts for different pages to find the suitable ones we were looking for.







4)**Library-rs2ml.jar:** We had problems as we didn't exactly know which library to use for GUI table manipulation, later after an extensive stressful search session we ended up with rs2ml.jar that provided easy solution for us.

5) **Time Zone problem-** We have worked over 200 hours together for this group from 3

different time zones, while during the holiday it was possible to push 13-15 hours session each day, during weekdays it was really tough on to work more than 3-4 hours/day. Luckily through pushing each other we have managed to pull that off.

Chapter 6 Testing

6.1 Test plan

6.1.1 Test environment

Eclipese IDE
Windows 10
Windows 11
IntelliJ
Localization environment in English Language

6.1.2 Test team responsibility

Unit testing is done by ABID ALI (2019380141)
Integrating testing done by Shahedul Islam (2018380130) and Tafsir (2019380179)

6.2 Unit Test

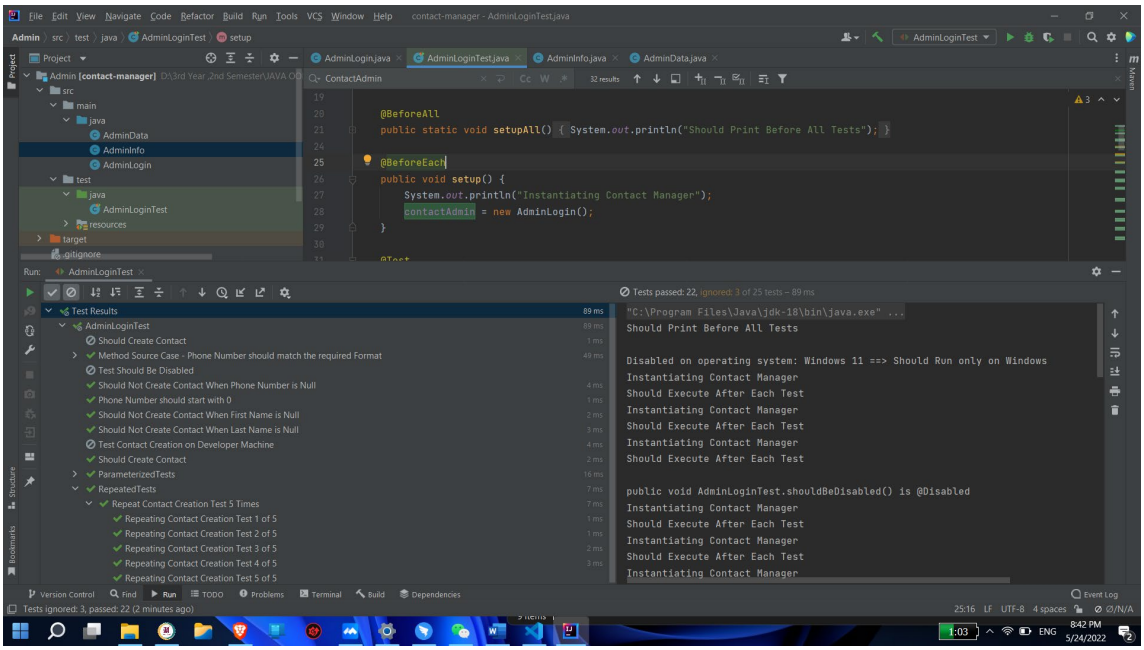
6.2.1 Test case

Project Name	NPU Retaurant
Module Name	AdminLogin
Created By	ABID ALI
Creation Date	05/21/2022
Reviewed By	Shahedul Islam
Reviewed Date	05/23/2022

No.	Test Scenario Description	Test Steps	Test Case Description	Test Data	Result	Expected
1	Verify the login functionality of Admin login page.	1)Enter valid username of Admin. 2) Enter valid password of Admin. 3)Check login button	Enter a valid username, valid password.	username :Abid password :1234	Admin should be able to see Admin homepage	Successful Login

2	Verify the login functionality of Admin login page.	1)Enter valid username of Admin. 2) Enter invalid password of Admin. 3)Check login button	Enter a valid username, valid password.	username :Abid password :Abid	Login failed	A pop up message to show Login failed
3	Verify the login functionality of Admin login page.	1)Enter invalid username of Admin. 2) Enter valid password of Admin. 3)Check login button	Enter a valid username, valid password.	username :Shahed password :1234	Login failed	Same as above

6.2.2 Test result



6.3 Function Test

6.3.1 Test case

Admin Functunality

Field Name	Valid Data	Invalid Data
Admin Username	Shahed Islam /Abid	Shahed / Other names
Admin Password	1234	Except 1234

Test Case	Test steps	Expected Result	Actual Result
Valid Admin Name and valid password	1.Enter'Abid' in name 2.Enter'1234'as password 3.Press Admin Login	Should save data in database	Correct
Valid Admin Name and invalid password	1.Enter'Abid' in name 2.Enter'Shahed'as password 3.Press Admin Login	System should prompt the user as Login failed	Correct
Invalid Admin Name and valid password	1.Enter'Shahed' in name 2.Enter'1234'as password 3.Press Admin Login	System should prompt the user as Login failed	Correct
Invalid Admin Name and invalid password	1.Enter'Shahed' in name 2.Enter'Shahed'as password 3.Press Admin Login	System should prompt the user as Login failed	Correct

6.3.2 Test result

Functional Testing is blackbox testing.

6.4 Test Summary

Unit testing is done by ABID ALI (2019380141)
From the unit testing we isolate the module then run some testcases and checked did we get the expected result according to our plan.

Integrating testing done by Shahedul Islam (2018380130) and Tafsir (2019380179)
From the integrating testing we run the whole program then run some testcases and checked did we get the expected result according to our plan.
Test Environment were given below
Eclipese IDE
Windows 10
Windows 11
IntelliJ
Localization environment in English Language

Chapter 7 Summary

7.1 Project summary

Through our app we have focused on increasing work efficiency of the existing systems. Our app is very user friendly, easy to use and navigate. Multiple admins can access the database according to the need and working time. Modifying food in the database can be done from the app itself without touching database. Same goes for admin credentials.

We have focused on a feasible fast purchasing system. Therefore, and user can do all he/she wants (select, drop, cancel order, change quantity, provide customer info) with 4/5 clicks and the order would be placed. Our app is visually soothing and app interface is attractive creating a rich user experience for the customers.
So, the main issues we have addressed during designing and developing the app are quick, easy to use, user friendly and foremost accessible.

Work List:

Member	Work	Size (LOC)
Team Leader Khan Md Shahedul Islam 2018380130	Implementing structure design, working on classes, interface design, Code review, UML diagram design, Bug fixing, creating data base, report writing.	**1012** LOC per day is 26 Started from04/14/2022

Member 1 Abid Ali 2019380141	Implementing structure design, working on classes, interface design, Code review, UML diagram design, Bug fixing, creating data base, report writing.	**1120** LOC per day is 29 Started from04/14/2022
Member 2 Tafsir Mahmud 2019380179	He joined the group late due to his two surgeries of stomach bleeding and still didn't recover yet. As he joined late, we gave him simple task for example layout design and different kind of testing and he brought different perspective towards the project	As he was performing testing and design layout So, his LOC is low. **178**

7.2 Technical summary

To create our application, we followed **Behavioral pattern** for controlling action events.

Besides that, we also used features like exception handling (to catch SQLException), collections (*Vector*, *ArrayList*), built in interface *ActionListener* and its *actionPerformed*() function for event handling (When the action event occurs, that object's *actionPerformed* method is invoked.)

The way we have designed and implemented the design it can be best described with the UML diagrams. We have used 19 classes in total developing this application and the required functionalities by the teacher were implemented as a necessity to make this app.

Reference

1. https://docs.oracle.com/en/java/

2.https://docs.oracle.com/javase/7/docs/api/javax/swing/package-summary.html

3. https://youtu.be/g0PrXoWKM2Y

4. https://www.youtube.com/watch?v=iWtxEDE1IR4

5. https://www.youtube.com/watch?v=xk4_1vDrzzo

6. https://www.youtube.com/watch?v=0nuGm8HBjWk

7. https://idratherbewriting.com/learnapidoc/docapis_sdks.html

8. https://www.w3schools.com/sql/

9. https://www.youtube.com/watch?v=HXV3zeQKqGY

10. https://www.eclipse.org/