**Team Name**

STRIKER’S

**Project Documentation**

***On***

MOBILE GARAGE

*Submitted by*

UTTAM MEWADA

UJJWAL SRIVASTAVA

ATISHAY JAIN

JUHI KUMARI

ABDULLAH KHALID

INTRODUCTION

The proposed idea is an online solution application for all the mobile users. Mobile garage is a java-based application in which customers can buy or sell their mobile according to their necessity from a single application. In this application customers can put their existing mobiles over or purchase from the existing one moreover there is a feature of repairing the mobile as well. The customers can register the issue as well as check the status of the mobile while repairing process

FEATURES

* Admin can view all the registered data, all placed orders as well as delete the customer data with the help of customer id.
* Admin can view and update the issue raised by the customer.

* Admin can view, insert as well as delete all mobile data in the phone data repository.
* Buyer can register him/herself in the customer repository.
* Buyer can view all the mobile data as well as place the orders as many as they want.
* Sellers can put their old phones over for selling as well as check their status using customer id.
* The customer can raise an issue as well as check their issue status using the issue id.

MODULE DESCRIPTION

* ADMIN MODULE

In the admin module we have features like

1. Update Issue
2. Insert Mobile Data
3. View All Placed Orders
4. View Registered Data
5. View Old Mobile Data
6. View Old Mobile Data (Customer id)
7. View Issue
8. View All Mobile Data
9. Find By ID (Customer id)
10. Delete Mobile Data (Model no.)
11. Delete Customer Data (Customer id)

* BUYER MODULE

In the buyer module we have the following features

1. Place Order
2. Customer Registration
3. View Mobile Data

* SELLER MODULE

In the seller module we have the following features

1. Add Old Phone Data
2. Check Status (Customer Id)

* REPAIR MODULE

  In the seller module we have the following features

1. Raise Issue
2. View Issue Status (Issue id)

LAYERED ARCHITECTURE

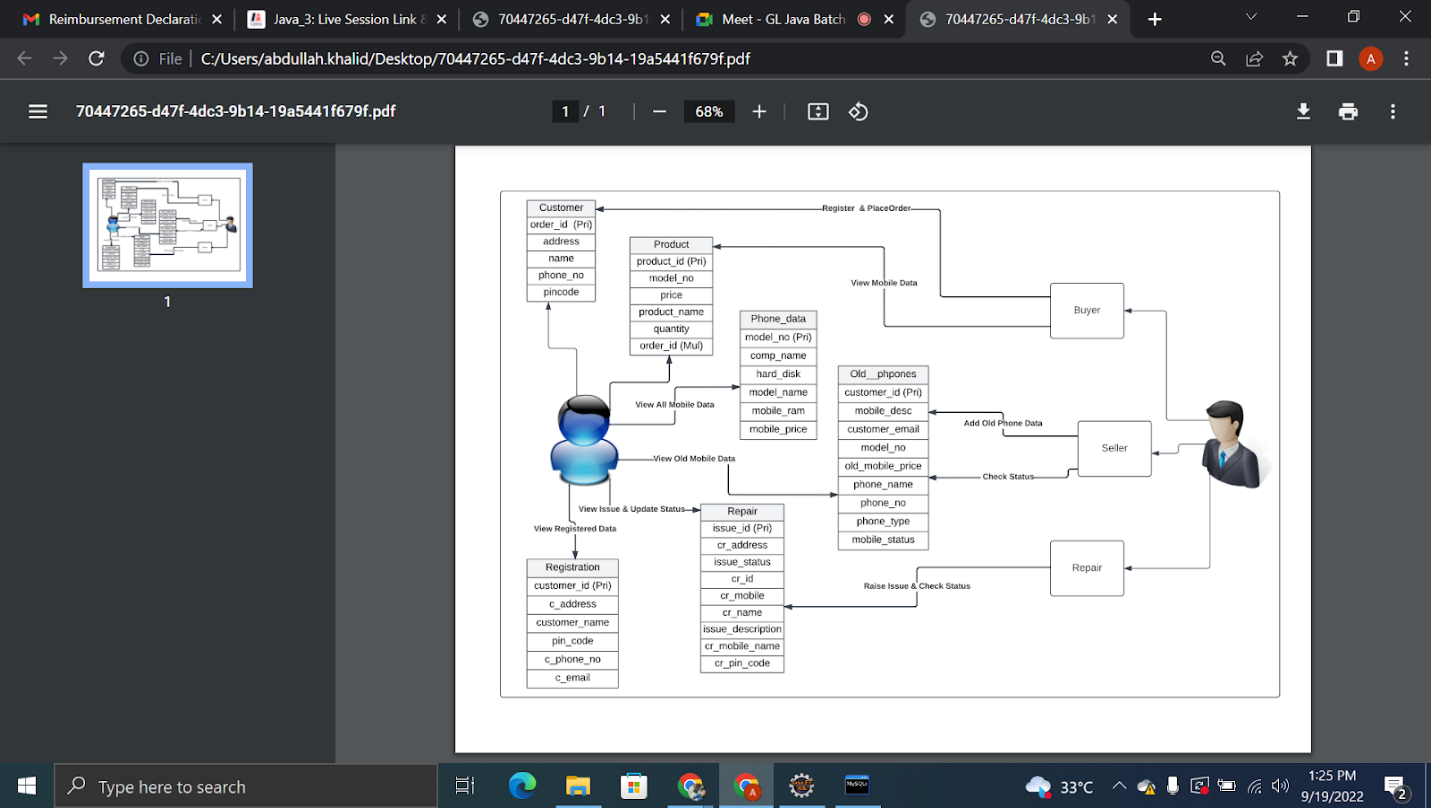
Layered architecture patterns are n-tiered patterns where the components are organized in horizontal layers. This is the traditional method for designing most software and is meant to be self-independent. This means that all the components are interconnected but do not depend on each other. There are four layers in this architecture where each layer has a connection between modularity and component within them. From top to bottom, they are

* The presentation layers
* The business layers
* The persistence layers
* The database layers

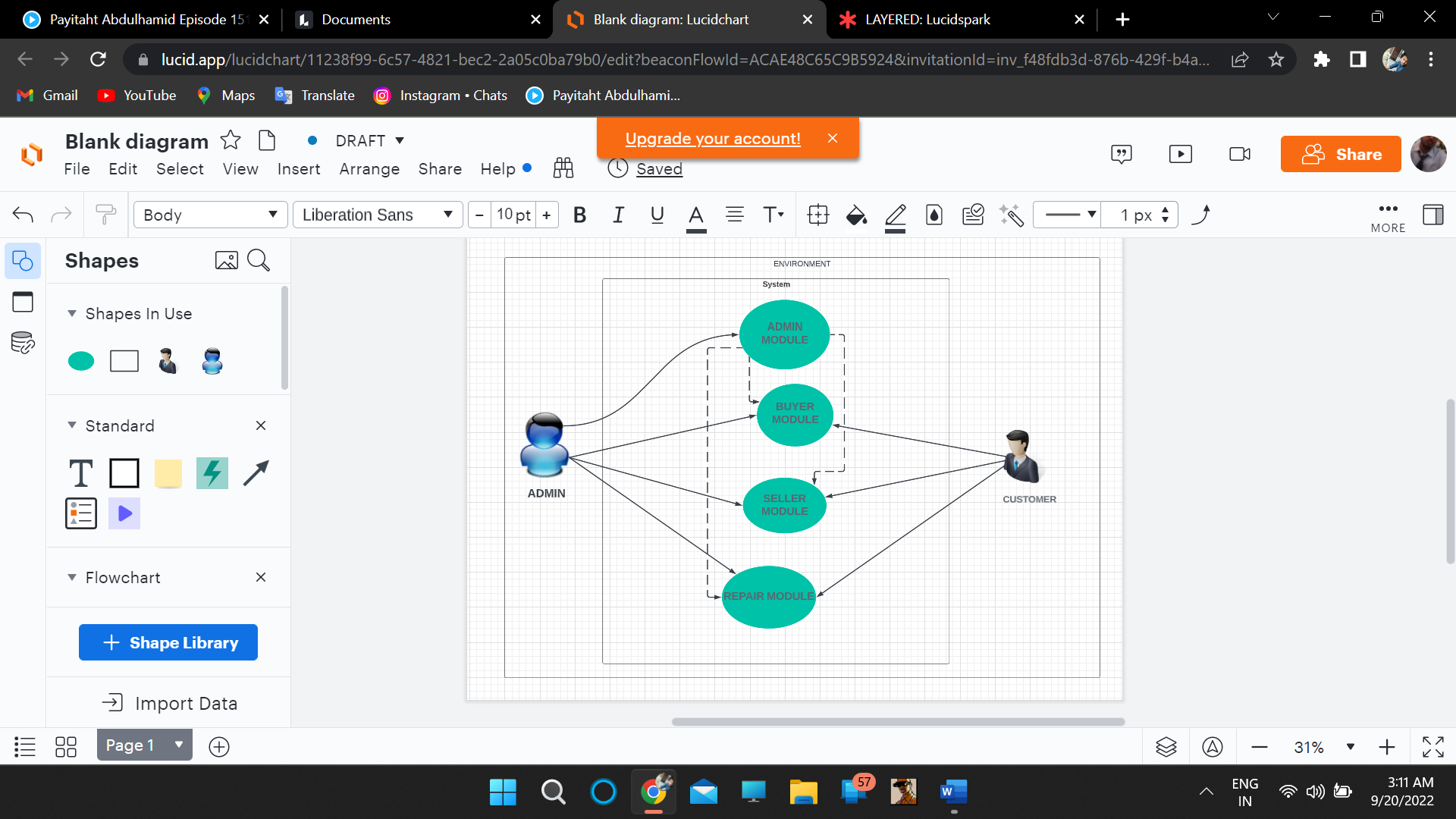
TOOLS & TECHNOLOGY

* IDE : Eclipse
* Language : JAVA
* Framework : Spring Boot
* API Documentation : SWAGGER
* DATABASE : MYSQL

DIAGRAMS



**Data Flow Diagram**



**Use Case Diagram**