

# **SoundHolics - “The Sound of Soul”**

- An E-Commerce web-app and android app

## **ABSTRACT-**

This application is an e-commerce website that is concentrated for music lovers only. According to the question, we should select 5 categories and thus, are listed below -

- 1) Headphones
- 2) Earphones
- 3) Speakers
- 4) Earbuds
- 5) Audio Players

## **TECHNICAL COMPONENTS-**

The application is divided into front-end for a smooth user experience and back-end for all the core activities. The technologies incorporated are listed below -

- 1) Backend - Spring boot, Firebase, MongoDB, Java
- 2) Frontend - Vue.js, HTML, CSS, JavaScript, Android Studio

## **LIBRARIES INCORPORATED-**

- 1) Elastic Search
- 2) Retrofit
- 3) Guide
- 4) Kafka
- 5) Mongo repositories
- 6) Spring Data Jpa
- 7) Spring Cloud
- 8) Vuex

## EXTERNAL API's INCORPORATED-

- 1) Java Mail API
- 2) Spring Cloud API (PROXY)

## MICROSERVICES-

- 1) Login
- 2) Merchant
- 3) Product
- 4) Order/Cart
- 5) Search
- 6) Rating

## ENTITIES-

- 1) Customer: To buy audio devices (Firebase)
- 2) Order: Confirmed orders (NoSQL)
- 3) Merchant: To sell their audio devices (Firebase)
- 4) Products: List of the products (SQL)
- 5) Review: Ratings of all the products (NoSQL)
- 6) Cart: To store products (NoSQL)

## OAUTH-

**OAuth** doesn't share password data but instead **uses authorization** tokens to prove an identity between consumers and service providers. **OAuth** is an **authentication** protocol that allows you to approve one application interacting with another on your behalf without giving away your password.

## THE USER INTERFACE-

The UI shall be the same for the web-app and android application. Our UI has been divided into 2 sections for easier understanding and concentration towards our business objective, namely, Customer UI and Merchant UI.

Customer	
PK	Cust-Email
Password	
Name	
Address	
Phone	
Photo	
Log-in[]	

Uses Login API

Orders	
PK	Order-Id
Email	
Products [{	
prod-id,	
qty	
merch-id	
}]	
Date	
Address	

Uses Order/Cart API

Merchant	
PK	Merc-Email
Password	
GST-No	
Rating	
Order-Id[]	

Uses Merchant API

Products	
PK	Product-Id
Category	
Stock	
Attribute [{	
}]	

Uses Product API - sync via  
kafka → Search  
API

Rating	
Product-Id	
Rating[]	

Uses Rating API

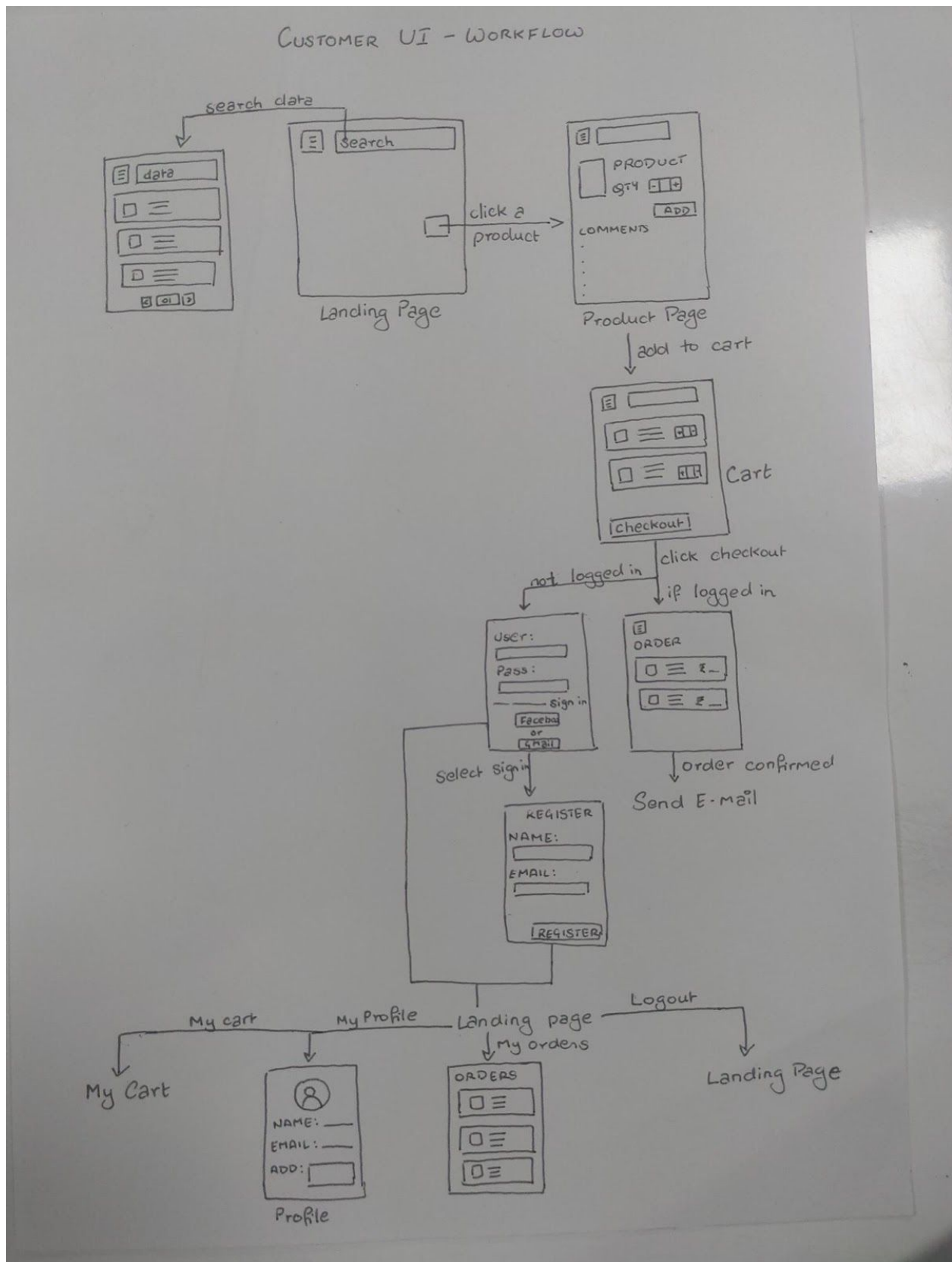
Merchant-Stock	
Key	Product-Id
	Merchant-Id
Price	
Stock	

Uses Merchant API

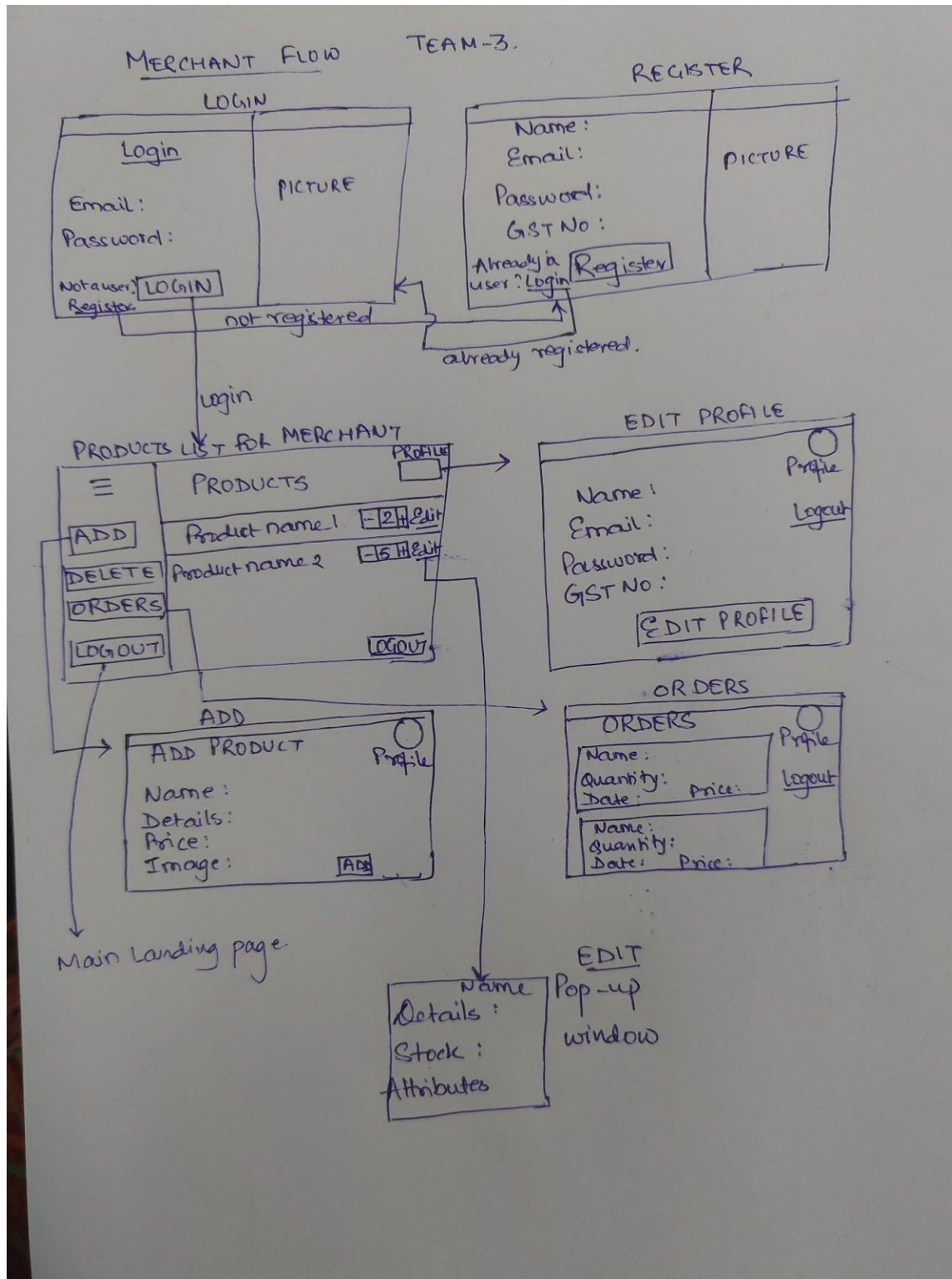
Cart	
Products[]	
Cust-Email	

Uses Order/Cart API

## UserInterface WORKFLOW for CUSTOMERS -



## UserInterface WORKFLOW for **MERCHANTS** -



## APIs INCORPORATED-

Login Service	
Post	Register User
Put	Update Detail
Get	Authentication
Get	Login History
Get	get Customer Details

Product Service	
Post	add New Product
Put	price Change
Put	quantity Change
Get	get Stock
Get	get Product Details

Merchant Service	
Post	Register Merchant
Get	get Orders
Get	get Authentication
Put	Update Details

Order / Cart	
Post	postOrder
Get	getOrder Details
Get	get Cart
Put	update Cart
Post	postCart
Delete	delete Cart

Search	
Get	getByTerm
Get	get By USP
Get	get By Category
Get	get By Filters

Rating	
Get	getRating
Put	putRating
Post	postRating

↑↓ Connects to other API

Proxy Server  
(Spring Cloud Gateway)

↑↓

User Interface

### **TEAM MEMBERS-**

- 1) Shubham
- 2) Raja
- 3) Swetha
- 4) Anirudh
- 5) Abhinav
- 6) Aakash