**FINAL PROJECT REPORT**

SEMESTER 2, ACADEMIC YEAR: 2024-2025

CT312H: MOBILE PROGRAMMING

* **Project/Application name: Pizza Shop**
* **GitHub link: https://github.com/24-25Sem2-Courses/ct312hm03-project-Tanchuoi**
* **Link demo: https://www.youtube.com/watch?v=R9LpXpX4jbs**
* **Student ID 1: B2111950**
* **Student Name 1: Đinh Hồ Thanh Tân**
* **Student ID 2: B2111923**
* **Student Name 2: Lê Tào Quốc Hải**
* **Class/Group Number (e.g, CT312HM01): CT312HM03**

1. **Introduction**

* Project/application description:A selling pizza application.
* A task assignment sheet for each member if working in groups:

|  |  |
| --- | --- |
| Đinh Hồ Thanh Tân | Home page, login/signup page, product list page, product detail page |
| Lê Tào Quốc Hải | Cart page, order page, order detail page, settings page |

1. Feature / Application page 1:

- Description: What is this feature/application page? What is it used for?

- Screenshots: some screenshots of this feature/application page.

- Implementation details: students should answer the following questions:

+ List the widgets used for this feature/page. Is there any special widget (not introduced in the lesson) used? If so, state them.

+ Does the feature use any libraries or plugins? If so, state them and state the role of those libraries/plugins.

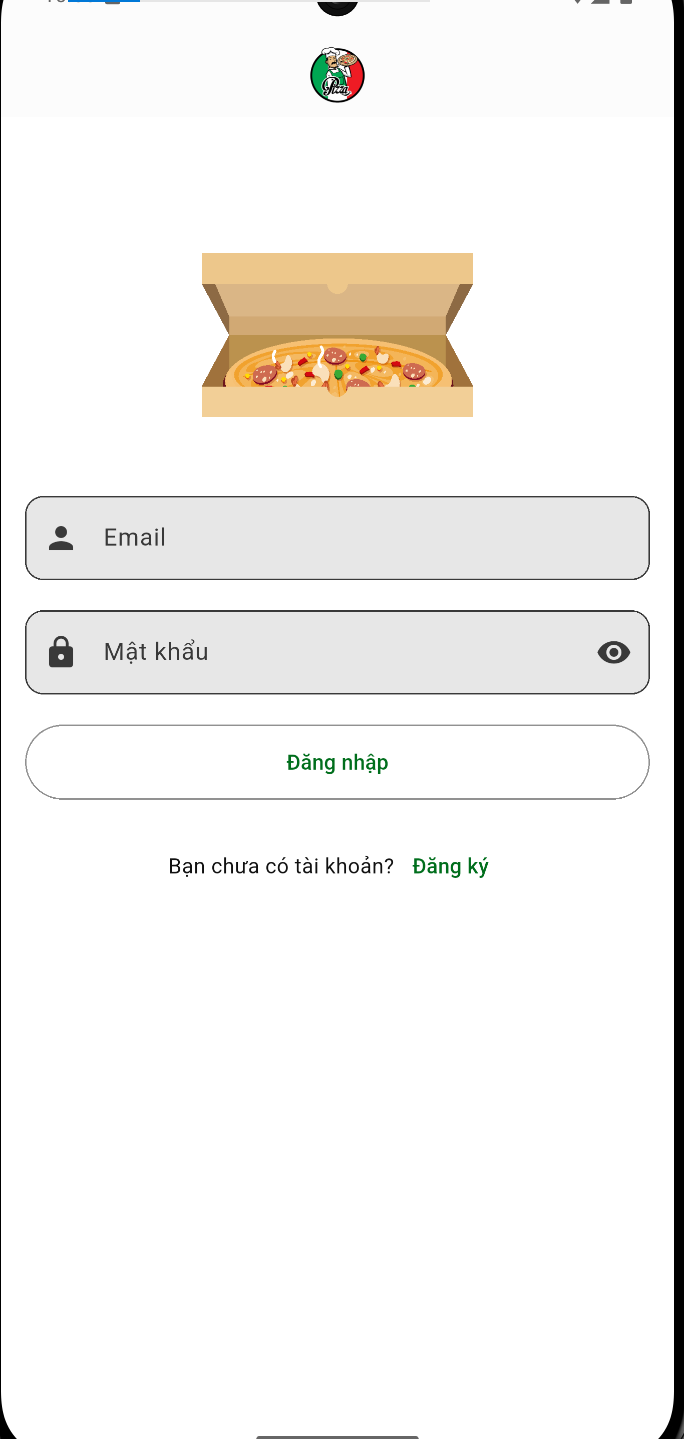
+ Does this feature use a shared state management solution? If so state briefly how your solution works and describe the code architecture.

+ Does this feature read or store any data? Locally or remotely? If so, state the data table structure. If you are using a REST API, briefly describe that API (how to call, input, output).

1. **Details of implemented features**
2. **Login page**

**- Description:** User login page

**- Screenshots:**



**- Implementation details:**

**+Widgets used:** Scaffold, AppBar, SingleChildScrollView, Padding, Column, Row, Image.asset, Lottie.asset, SizedBox, TextField, InputDecoration, OutlineInputBorder, ElevatedButton, Text, TextButton, Navigator.pushNamed

+ **libraries or plugins:**

- lottie – For displaying animated login illustrations.

+ **Shared state management:**

- LoginPage accesses UserManager using Provider.

- Upon successful login, the page navigates to /main.

+ **Data Storage and Retrieval:**

- **Remote Data**: This feature interacts with an authentication system UserManager.

- **Data table**: Users(id, email, username, phoneNumber, address, birthday, fullName)

1. **Signup page**

**- Description:** User signup page

**- Screenshots:**



**Implementation details**

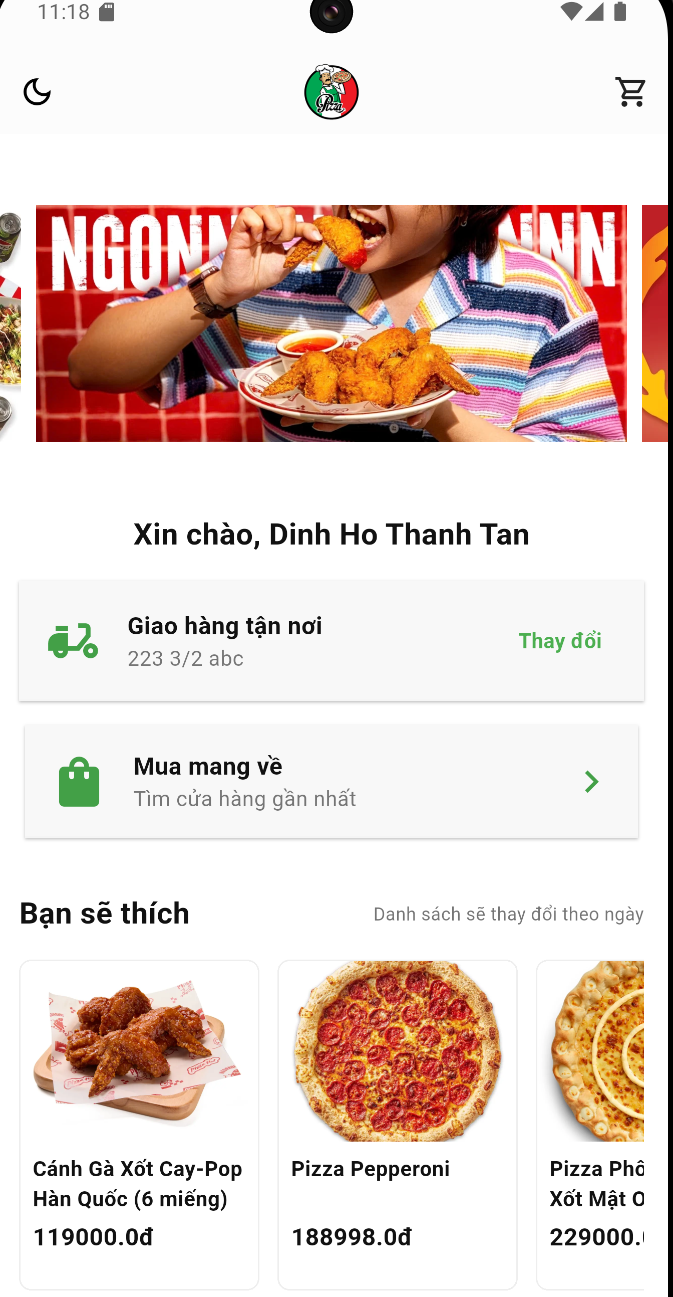
* **Widgets used:**Scaffold, AppBar, SingleChildScrollView, Padding, Column, Row, Image.asset, SizedBox, TextField, InputDecoration, OutlineInputBorder, IconButton, ElevatedButton, Text, TextButton, Navigator.pushNamed, SnackBar.
* **Libraries or plugins:**
  + provider – For state management (UserManager for handling user registration).
* RegisterPage accesses UserManager using Provider.
* Upon successful registration, the page navigates to /login.

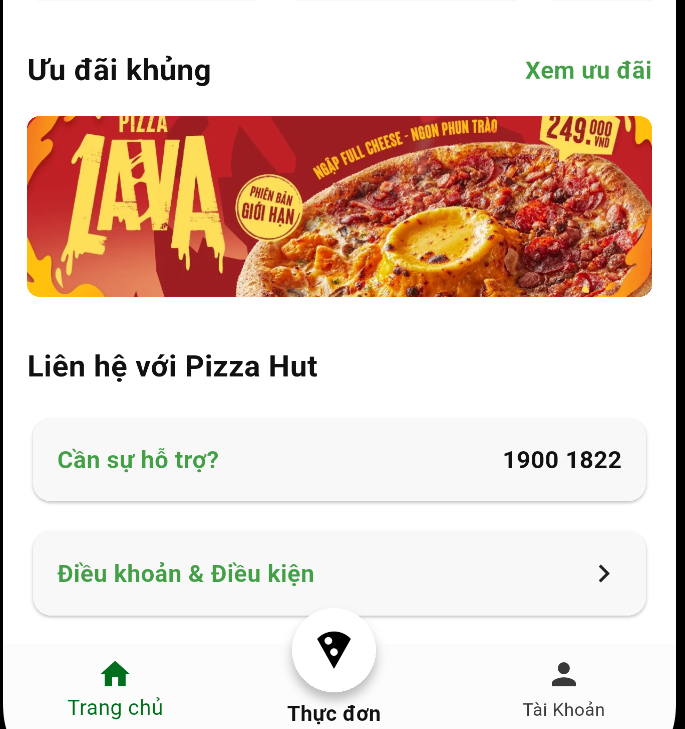
**Data Storage and Retrieval**

* **Remote Data**: This feature interacts with an authentication system via UserManager.
* **Data table**: Users(id, email, username, phoneNumber, address, birthday, fullName)

1. **Home Page**

* **Description:** This is the Home Page of the user, which contains banner, items you may like section, advertising…
* **Screenshots**





**Implementation details:**

* **Widgets used:**
  + SingleChildScrollView, Column, CarouselSlider, Align, Text, Card, Padding, Row, Expanded, IconButton, TextField, GestureDetector, PageView, ClipRRect
* **Libraries or plugins:**
  + carousel\_slider (for image carousel)
  + provider (for state management)

**Shared state management**

* The app uses **Provider** for managing user-related data through UserManager.
* The HomePage retrieves the logged-in user's information (userManager.user) and updates the delivery address.

**Data Storage and Retrieval:**

**Remote Data:**

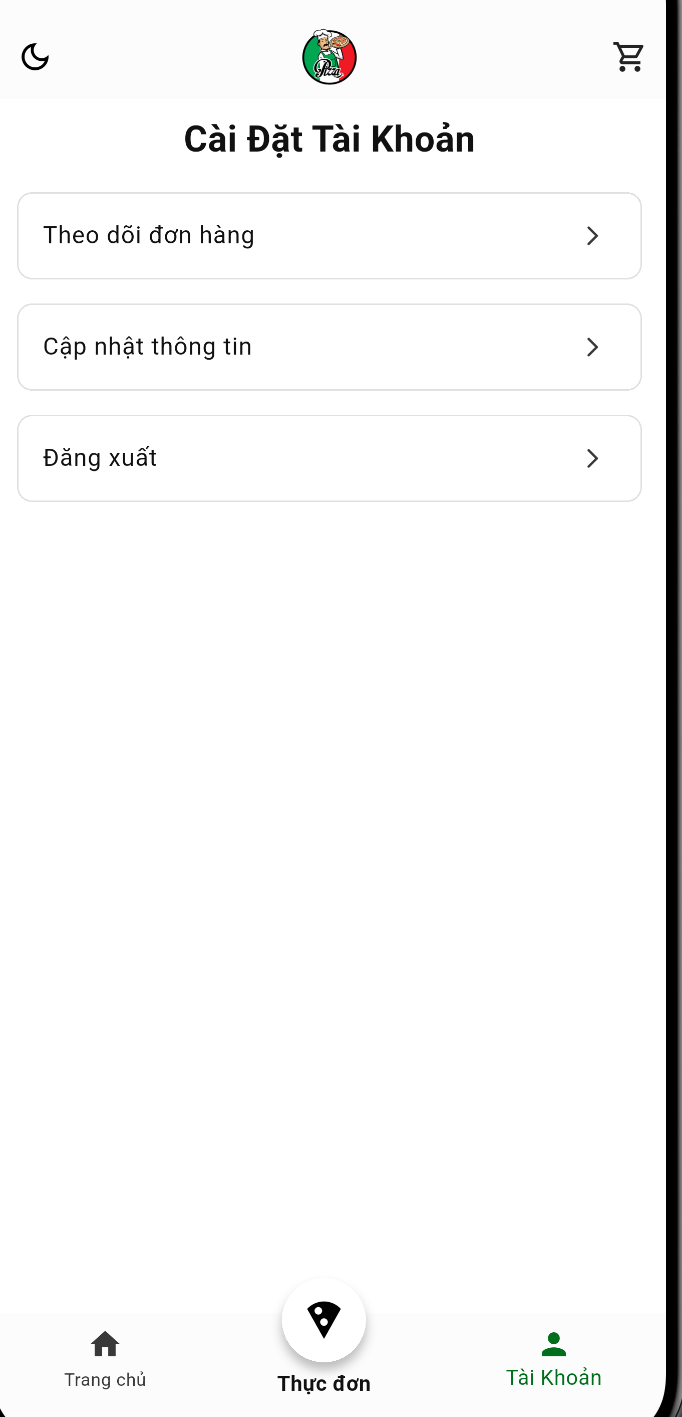
* ProductService().getRandomProducts() is used to fetch a random selection of products from the backend.
* userManager.updateAddress(newAddress) updates the user’s delivery address.

**Data table**: Users(id, email, username, phoneNumber, address, birthday, fullName)

1. **Profile Page**

**Description:** This is the Profile Page contains logout, update profile, track order.

**Screenshots**



**Implementation details:**

* **Widgets used:**
  + Column, SizedBox, Center, Expanded, ListView, Container, ListTile, Text, Icon
* **Libraries or plugins:**
  + flutter/material.dart (Core UI library)

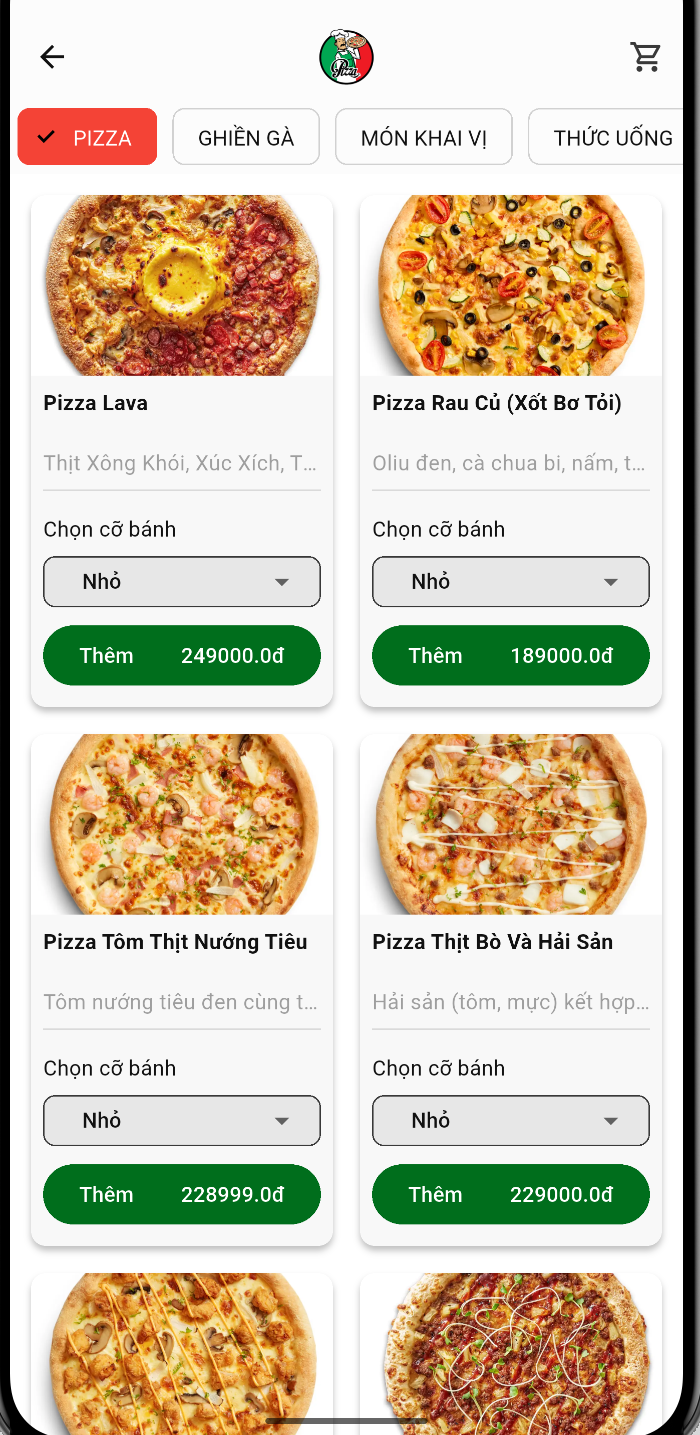
**Data Storage and Retrieval:**

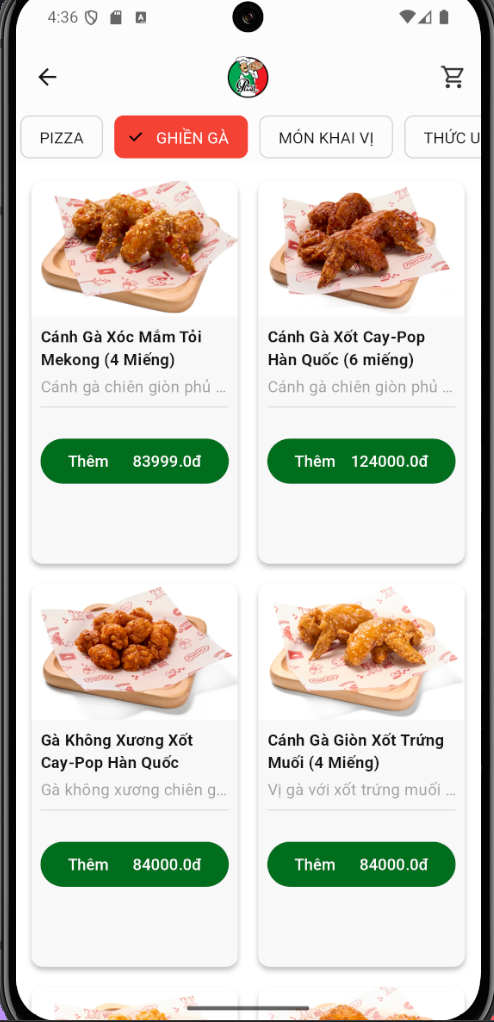
* **Remote Data:** Not used in this widget.
* **Data table:** No database integration in this snippet. However, if user information is retrieved from a backend, it could be stored in a User model and fetched via an API.

1. **Product list Page**

**Description:** This page show all products in shop filter by category

**Screenshots**

****

****

**Implementation details:**

* **Widgets used:**
  + Scaffold, AppBar, Image, IconButton, ChoiceChip, GridView.builder, CircularProgressIndicator, ProductCard, Padding, Center, Text
* **Libraries or plugins:**
  + provider (State management for product fetching)

**Shared state management:**

* The app uses Provider (ProductManager) to manage product data.
* getProductsByCategory(category) updates products based on the selected category.

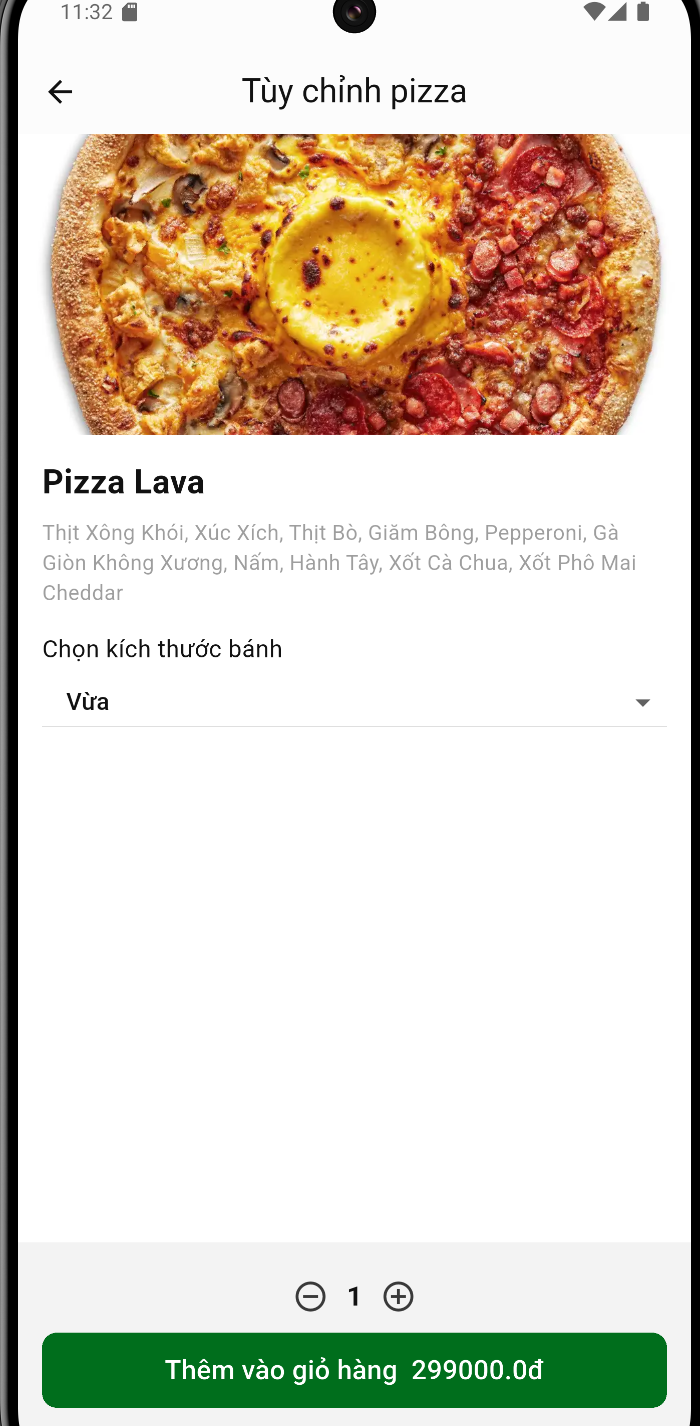
**Data Storage and Retrieval:**

* **Remote Data:** Products are fetched dynamically by category via ProductManager
* **Data table:** Products(id, name, description, price, featureImage, category, user)

1. **Product detail Page**

**Description:** This page show product detail and quantity selection.

**Screenshots**



**Implementation details:**

* **Widgets used:**
  + Scaffold, AppBar, Hero, Image.network, Column, DropdownButton, DropdownMenuItem, IconButton, FilledButton, Text, Padding, SizedBox, BottomAppBar, SnackBar.
* **Libraries or plugins:**
  + flutter/material.dart
  + provider (for CartManager).

**Shared state management:**

* Provider (CartManager) manages the cart state.
* setState is used for:
  + selectedSize updates.
  + quantity updates.
* ScaffoldMessenger provides user feedback when an item is added.

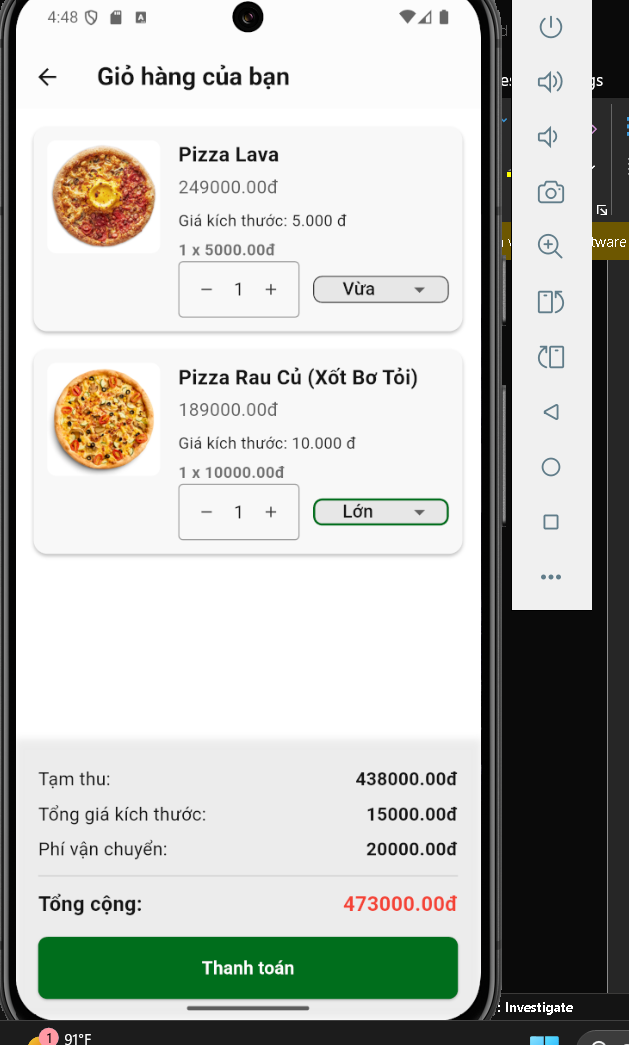
**Data Storage and Retrieval:**

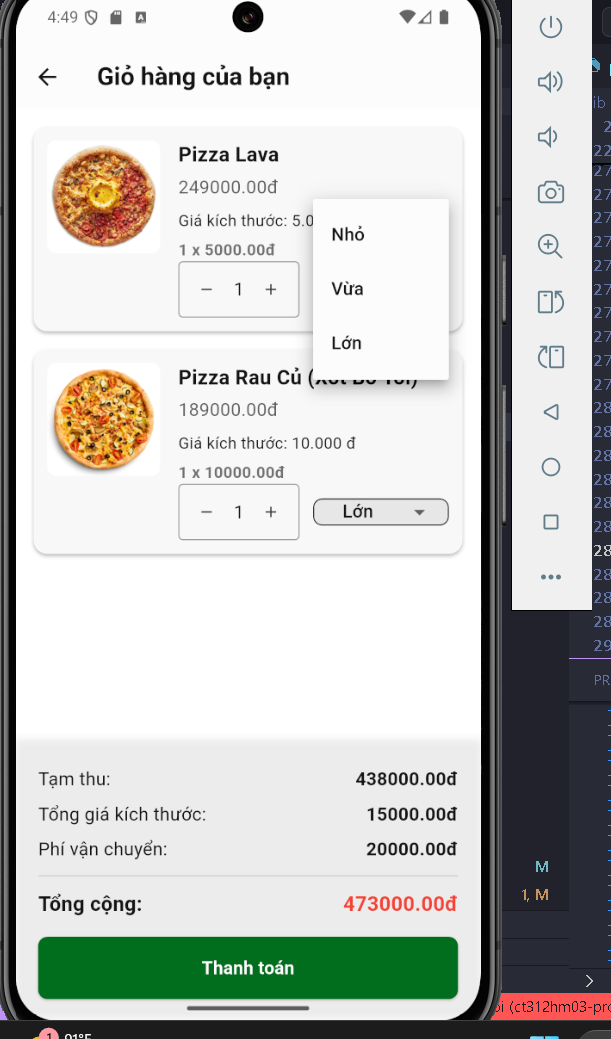
* **Remote Data:** Product details passed as a parameter.
* **Data table:** Products(id, name, description, price, featureImage, category, user)

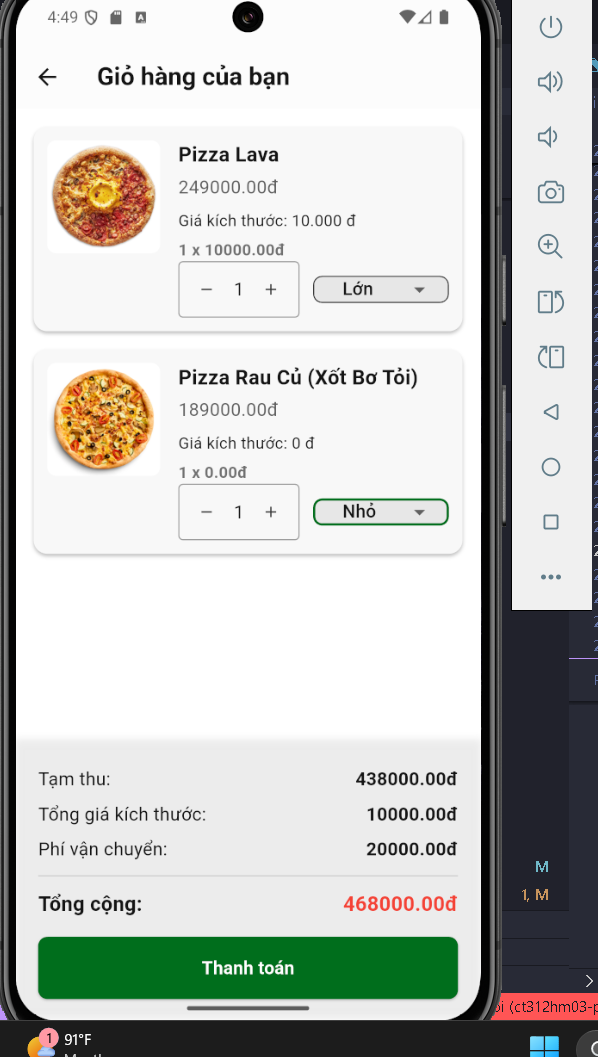
1. **Cart Page**

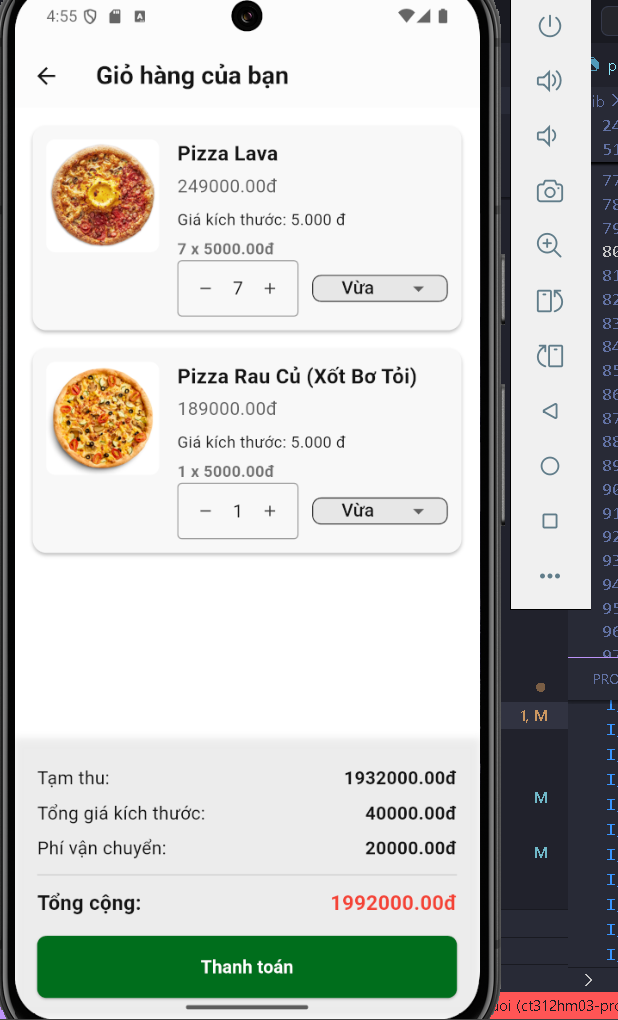
**Description:** This page cart item that was added by user, user can change quantity and size in this page and process to checkout

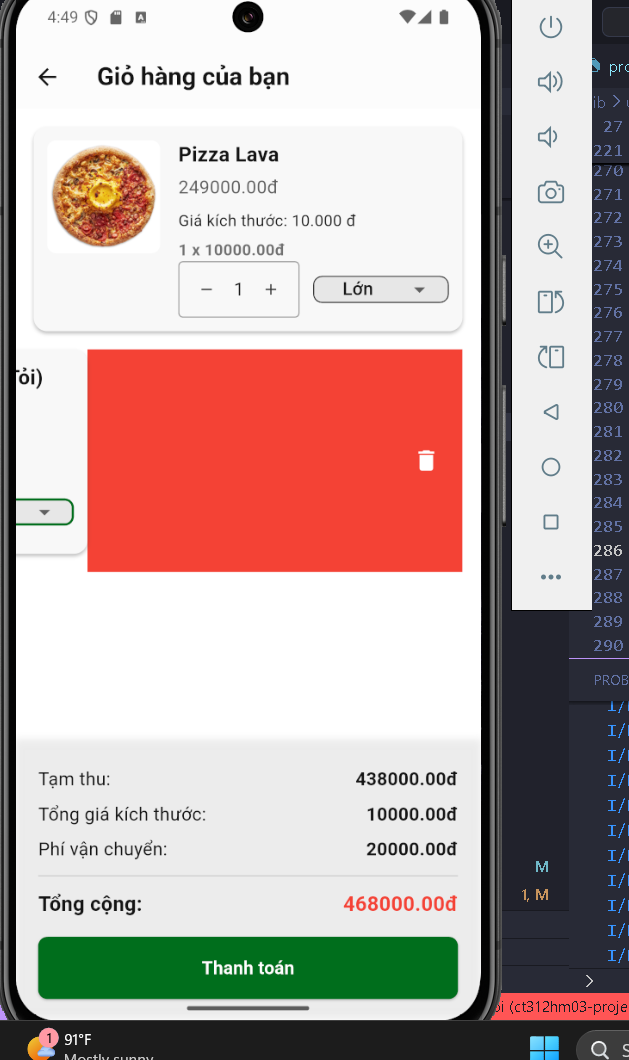
**Screenshots:**

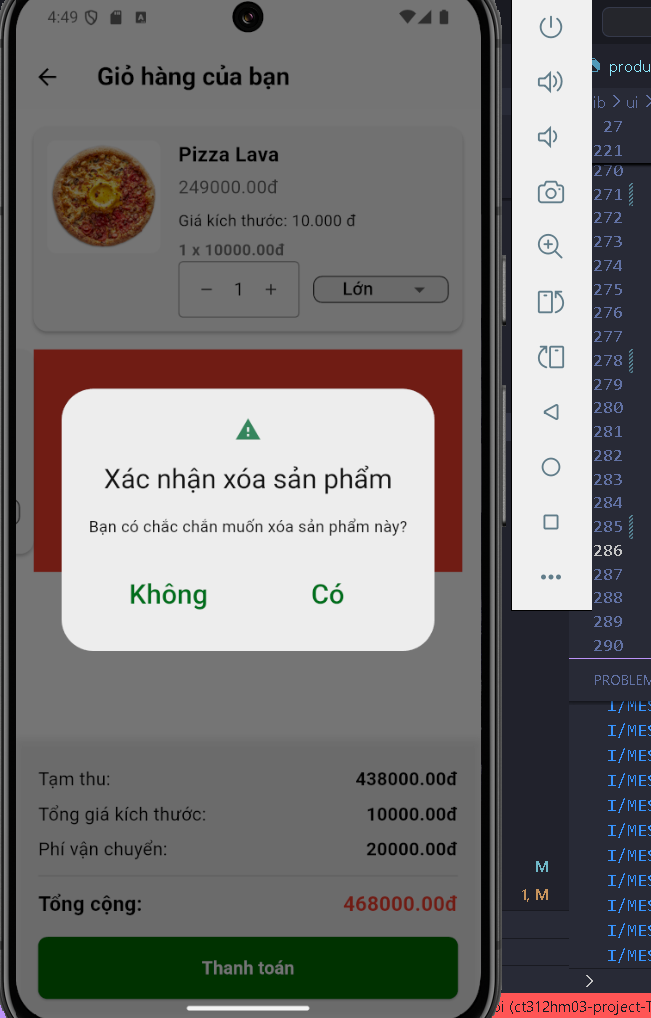


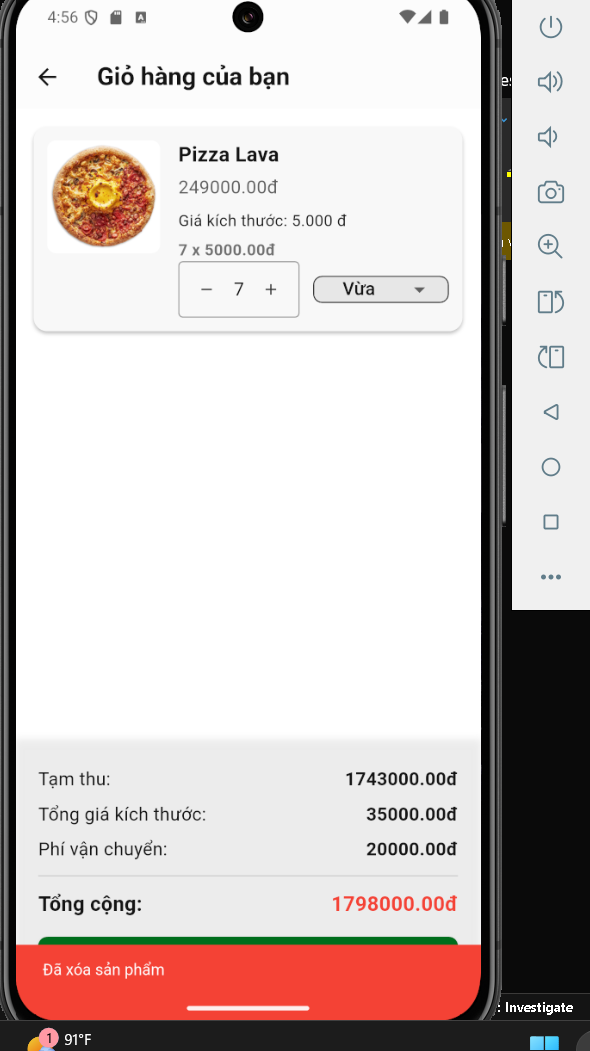


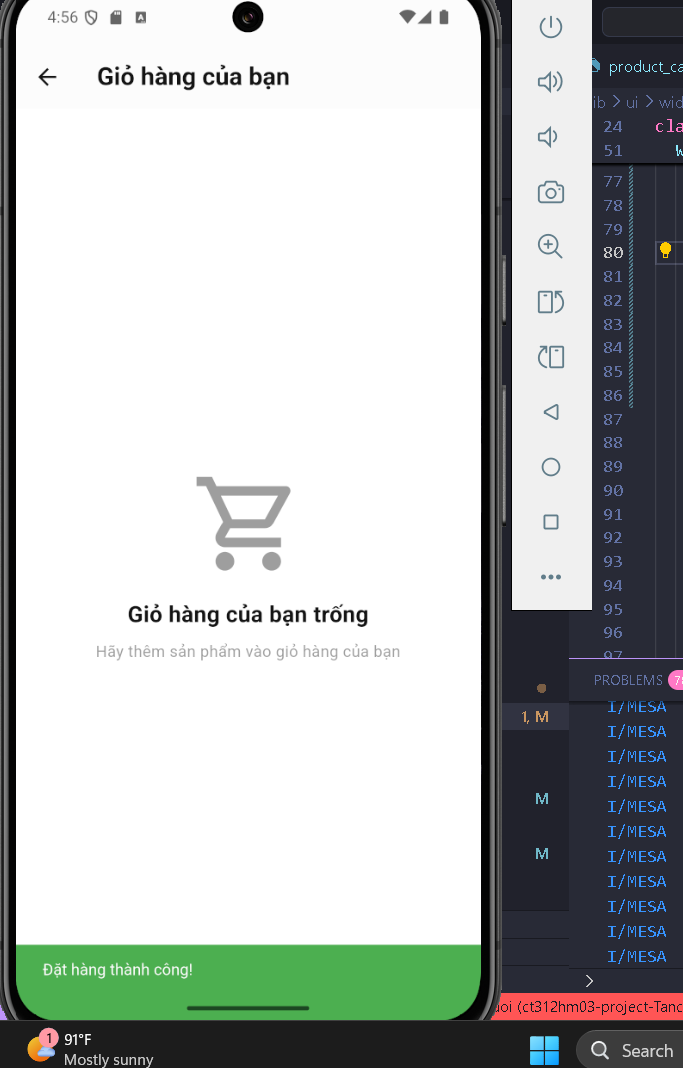








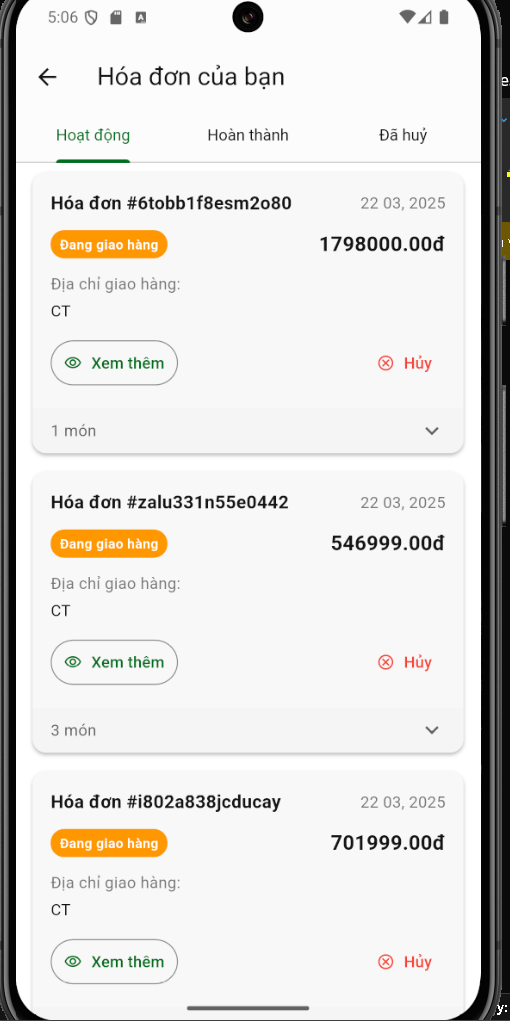


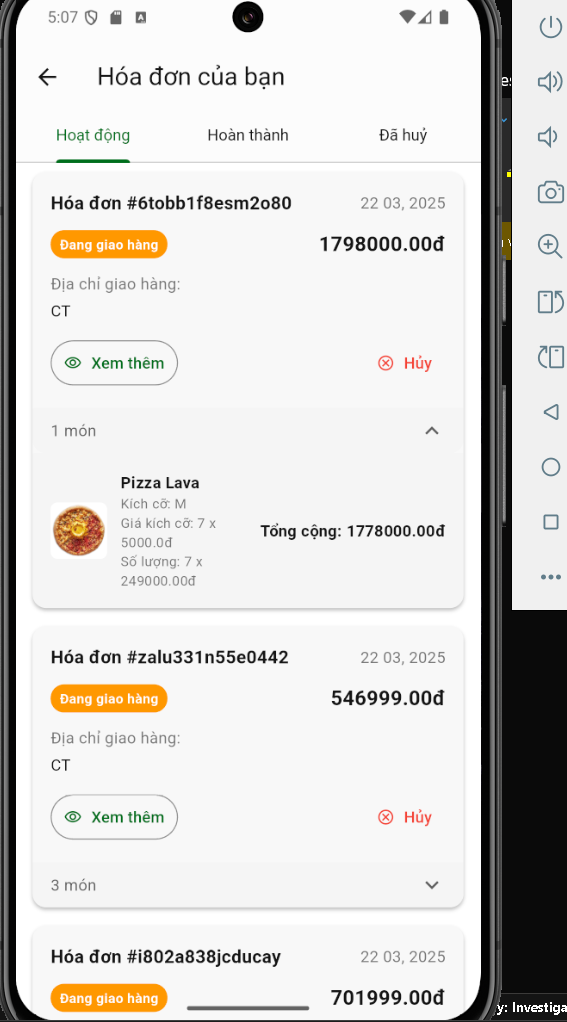


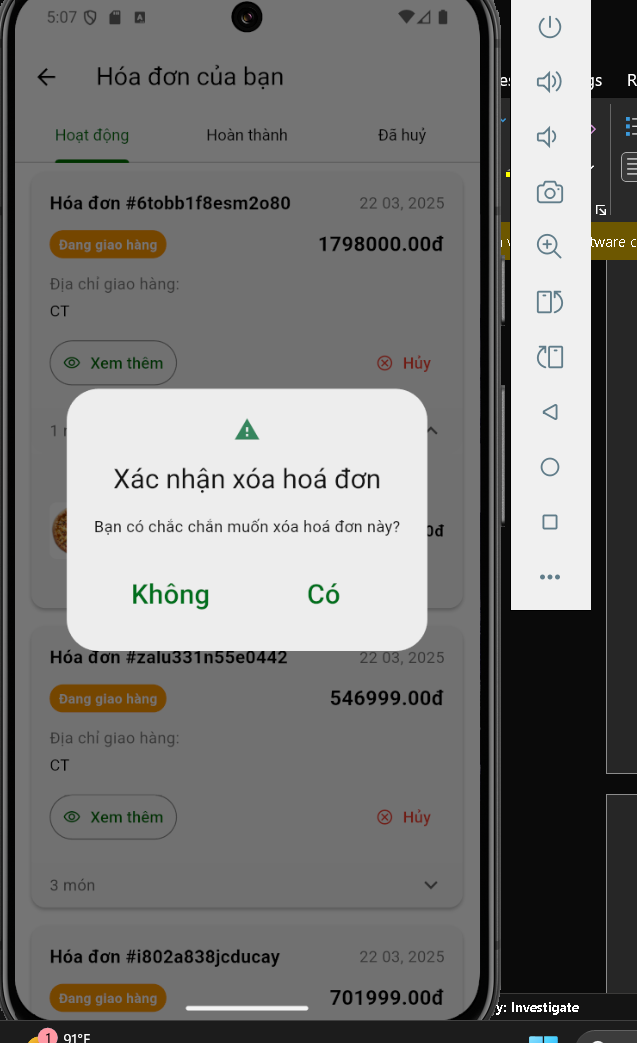
* **Implementation details:** students should answer the following questions:
  + **Widget:** Scaffold, AppBar, Column, Row, Expanded, Container, SizedBox, Divider, Text, Icon, ListView, FutureBuilder, Consumer, BoxDecoration, BoxShadow, ElevatedButton, SnackBar, ScaffoldMessenger, StatefulWidget, DropdownButtonFormField, Dismissible, Card, ClipRRect, Image, IconButton, DropdownMenuItem, OutlineInputBorder, InputDecoration, ScaffoldMessenger, EdgeInsets, BorderRadius, SizedBox, Expanded, Row, Column
  + **Libraries:** intl, provider
  + **State Manager and Code Structure:** 
    - **State Manager:**
      * Provider(use for quantity change and size change)
      * setState: on update quantity and size
    - **Code Structure:**
      * Models/cart.dart: store cart class
      * Page/cart.dart: Cart Page
      * Widget/cart\_item\_card.dart: Card of Item
      * Manager/cart\_manager.dart: Store cart item model data and cart item data function
      * Service/cart\_service.dart: Cart Item Pocketbase function
* **Database/Rest API:** Cart Item(id, name, quantity, price, product, status, size)

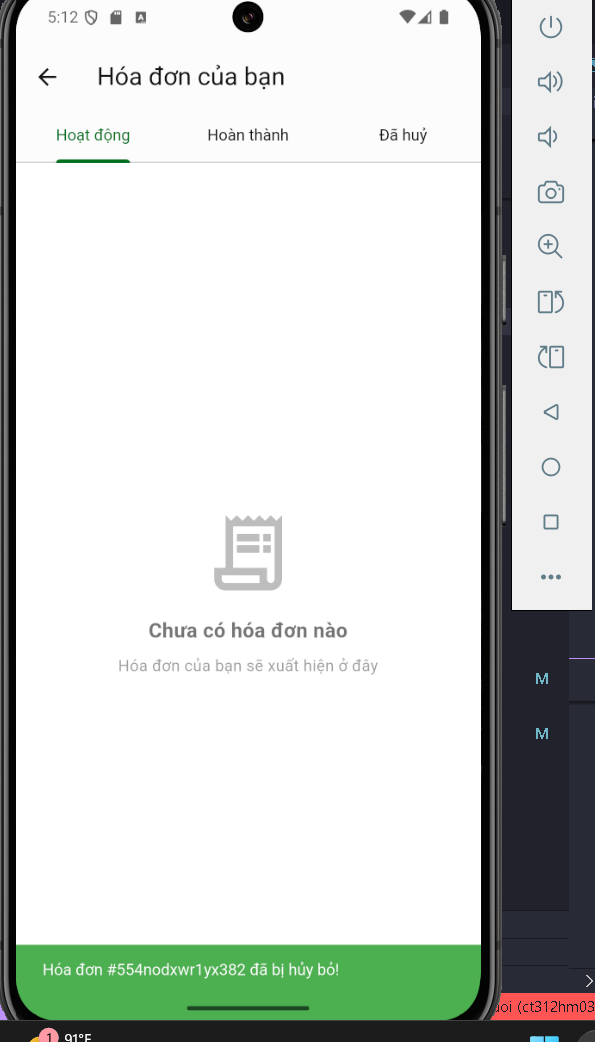
1. **Order Page**

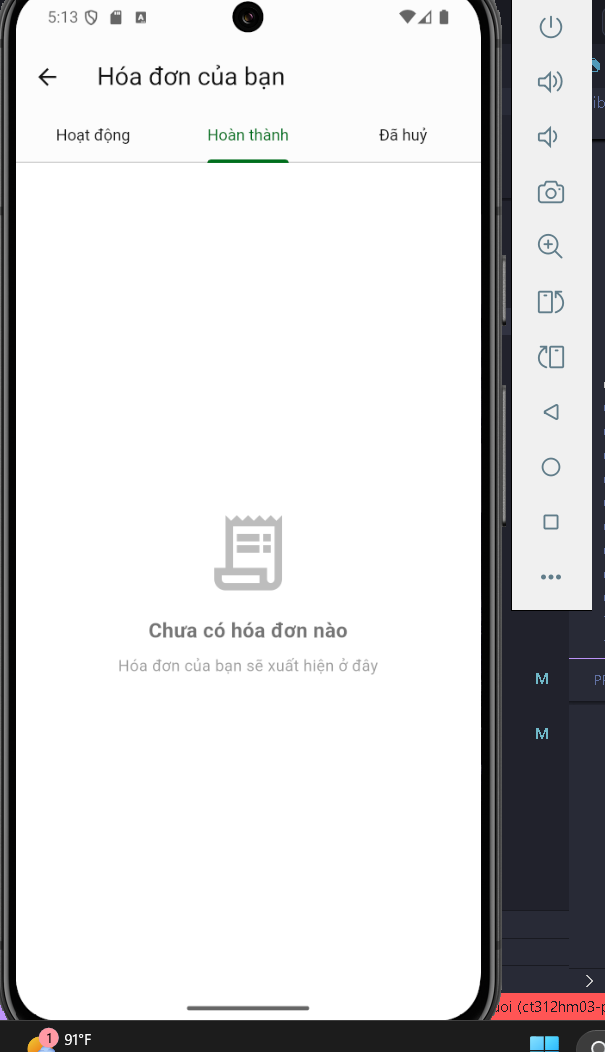
* **Description:** This is the Order of the user, where user could see all the orders that had made, the user can reorder in here, cancel the order or see order detail.
* **Screenshots:**

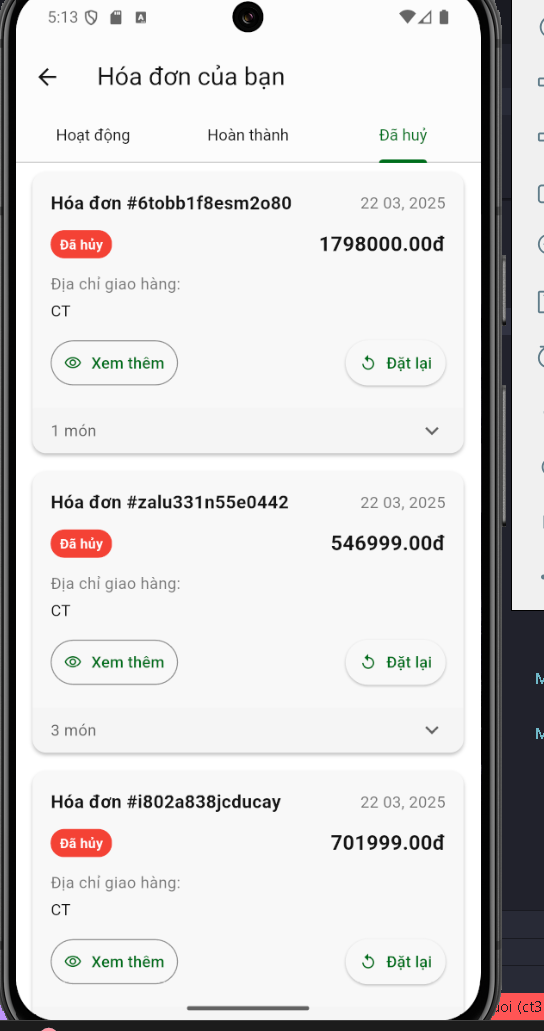
****

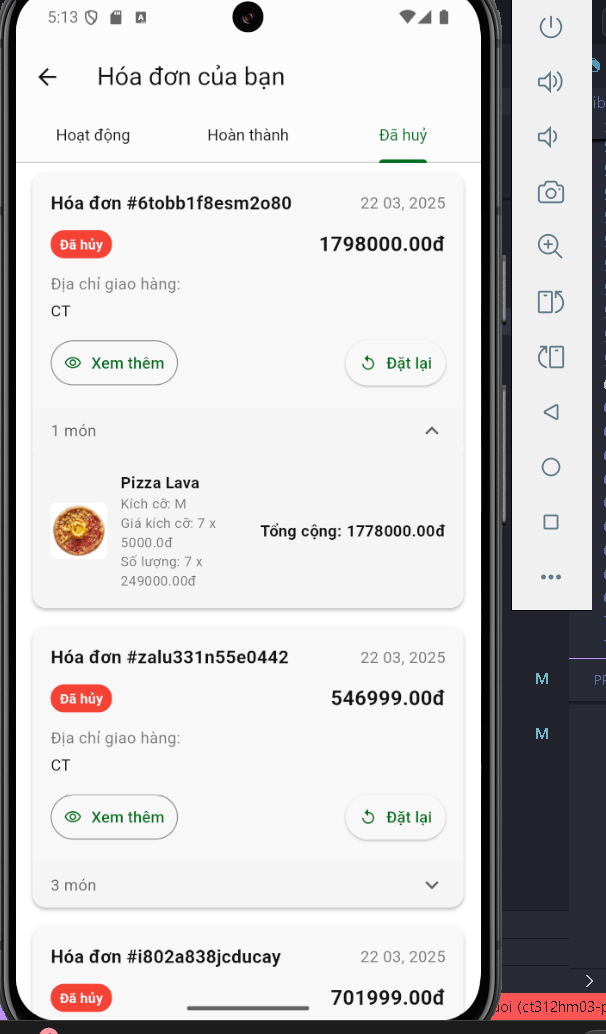
****

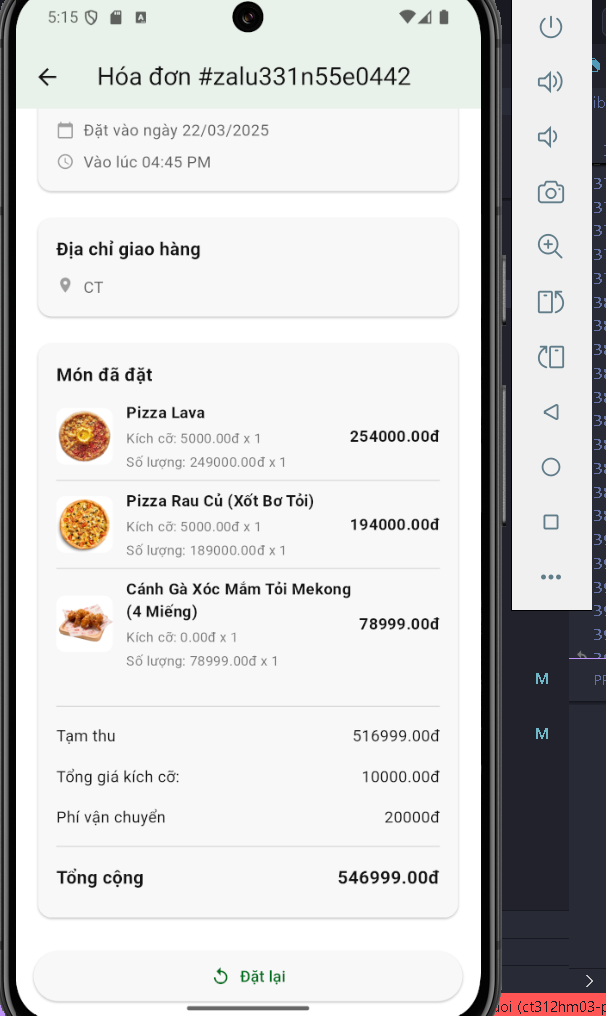
****

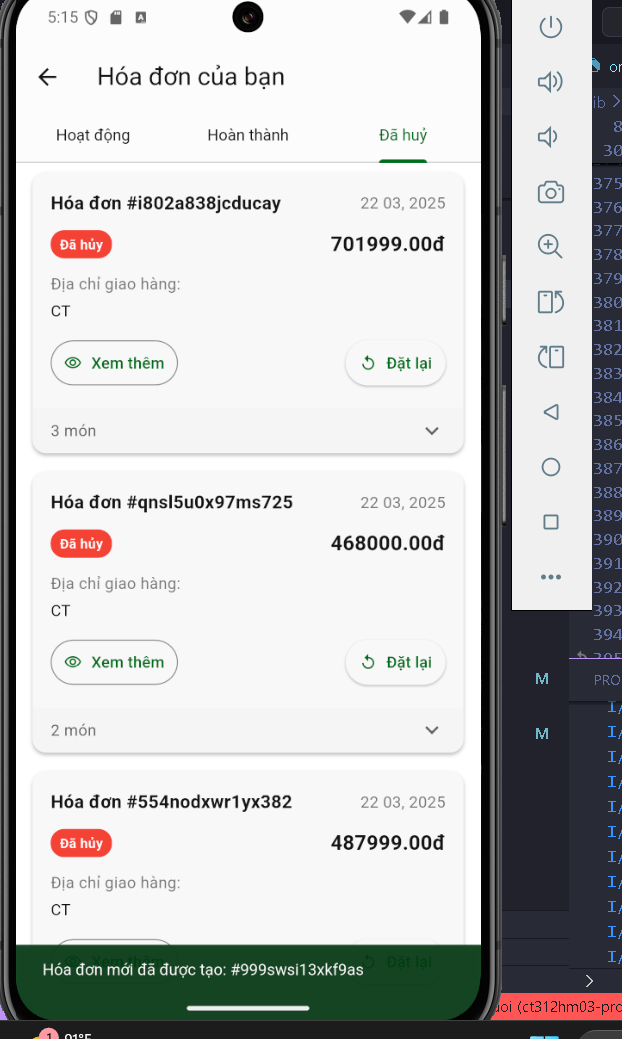
****

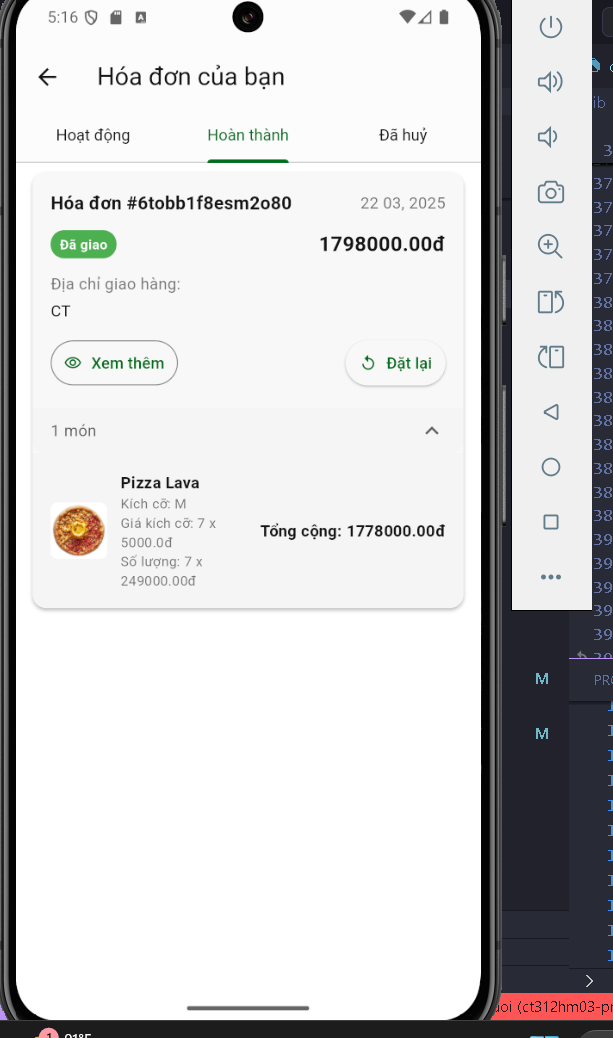
****

****

****

****

****

****

* **Implementation details:** students should answer the following questions:
  + **Widget:** Scaffold, SingleChildScrollView, Padding, Column, SizedBox, EvaluatedButton, ScaffoldMessager, SnackBar, Navigator, Text, Row, Expanded, Container, ClipRRect, ListView, Card, FutureBuilder, Consumer, TabBarView, InkWell, Icon, TextButton, OutlinedButton
  + **Libraries:** intl, provider
  + **State Manager and Code Structure:** 
    - **State Manager:**
      * Provider and ChangeNotifier to manager order state
      * Order Manager:
        + Fetch Orders,
        + Get Orders by ID to go to order detail
        + Cancel an order
        + Reorder
    - **Code Structure:**
      * Models/order.dart: Store order class
      * Page/order\_page.dart: Order Page
      * Page/order\_detail\_page.dart: Order Detail Page
      * Widget/order\_item.dart: Order Item
      * Widget/order\_list.dart: List of Order Item
      * Manager/order\_manager.dart: Store Order model data and state function
      * Service/order\_service.dart: Order Pocketbase function
* **Database/Rest API:** orders(id, amount, dateTime, user(single), cart\_items(multiple), deliveryAddress)