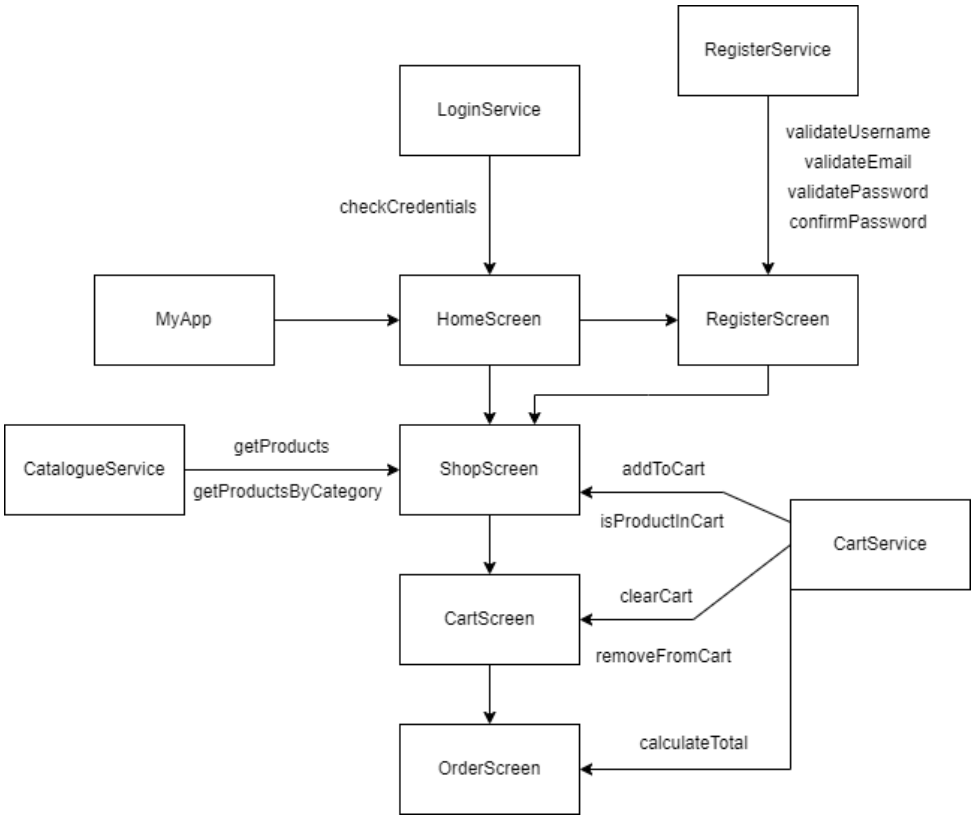
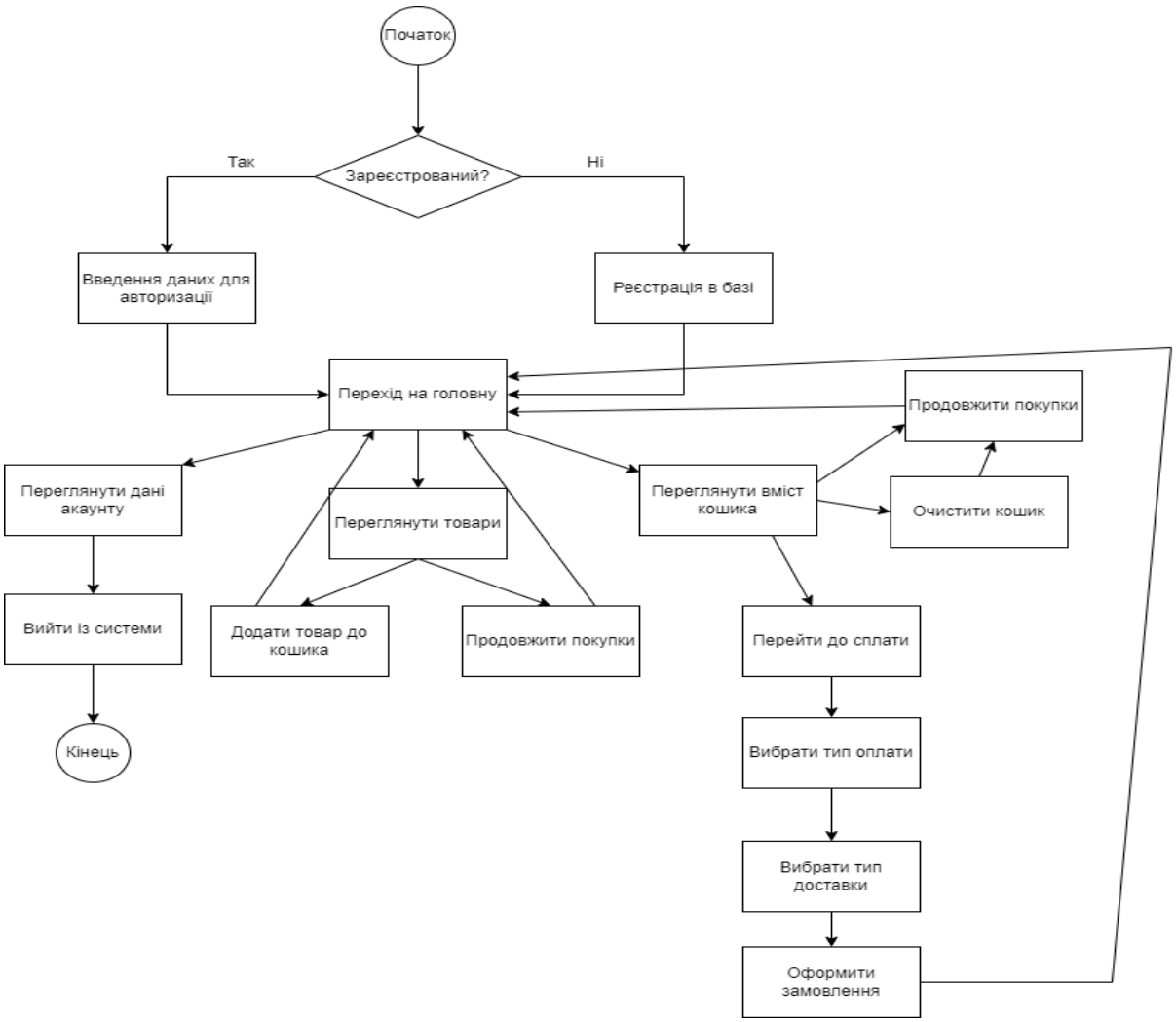
Архітектура





HomeScreen

Тут користувач може ввести дані для входу в систему. Перевірка даних для входу виконується класом LoginService. Якщо він не зареєстрований, він може створити обліковий запис, перейшовши на RegisterScreen.

LoginService

CheckCredentials – метод, який перевіряє чи є у Firebase Auth користувач з такими даними. Він звертається до Firebase Auth за допомогою методу signInWithEmailAndPassword перевіряючи чи відповіають введені пошта та пароль зареєстрованому користувачу.

RegisterScreen

Тут користувач може ввести дані для створення облікового запису. За створення та первірку введних даних відповідає клас RegisterService. Після реєстрації користувач входить в систему і переходить на головну сторінку магазину.

RegisterService

RegisterUser – метод, який створює користувача у Firebase Auth та зберігає додаткову інформацію у CloudFirestore. Для цього він звертається до Firebase Auth за допомогою методу createUserWithEmailAndPassword та створює користувача з поштою та паролем. Потім генерує унікальний id і використовуючи його зберігає у CloudFirestore пошту, пароль та ім’я користувача.

validateUsername, validateEmail, validatePassword – методи, які перевіряють правильність введення імені, пошти та паролю. Ім’я користувача може містити латиницю, кирилицю та цифри. Пошта може містити латиницю, цифри і обов’язково мати @. Пароль може містити латиницю, цифри та мати мінімум 6 символів (обов’язкова умова від Firebase Auth)

confirmPassword – метод призначений для підтвердження паролю.

ShopScreen

Головна сторінка магазину, тут можна переглянути товари в магазині, дані користувача, вийти із системи та перейти до кошика. Також можна вибрати

виведення товарів за категорією. За це відповідає клас CatalogueService. За додавання товарів до кошика відповідає CartService.

loadUserData – метод, який звертається до Firestore, щоб отримати ім’я користувача.

CatalogueService

getProducts – метод для отримання всіх продуктів з Firestore. Він створює масив із об’єктів у колекції products у Firestore. Масив представляє собою List із інтерфейсом Map.

getProductsByCategory – метод для отримання продуктів за певною категорією. Працює подібно до getProducts, тільки в запиті до бази вказується тип продуктів.

CartScreen

Тут можна переглянути товари в кошику. Є можливість видалити товари з кошика та перейти до оформлення замовлення. За контроль вмісту кошика відповідє CartService.

CartService

Клас створює динамічний список cartItems для збереження товарів у кошику.

addToCart – метод для додавання продукту до кошика. Додає новий елемент до масиву cartItems.

isProductInCart – метод для перевірки, чи продукт вже є у кошику. Для цього він звертається до Firestore і перевіряє чи є продукт з таким id у кошику.

removeFromCart – метод для видалення продукту з кошика. Видаляє елемент з масиву cartItems.

clearCart – метод для видалення всіх продуктів з кошика. Очищує масив cartItems.

calculateTotal – метод для обчислення суми продуктів в кошику. Додає ціну всіх продуктів у кошику.

OrderScreen

Тут можна переглянути суму замовлення, вибрати спосіб оплати та варіант доставки. За суму замовлення відповідає метод calculateTotal із CartService. Для оплати є 2 варіанти: картка та готівка. При виборі оплати карткою з'являються поля для введення реквізитів картки. Для доставки є 2 варіанти: доставка до дверей та самовивіз. При виборі доставки дверей з'являється поле для введення адреси. Якщо вибрати самовивіз із відділення, то з'явиться меню з вибором відділень. Відділення генеруються за допомогою randomuser.me і зберігаються у динамічному списку у форматі: «місто, вулиця, номер».

Колекції у Cloud Firestore

Users – колекція де зберігаються дані користувачів. Об'єкт колекції має поля: email(string), password(string), username(string).

Products – колекція де зберігаються дані про продукти. Об'єкт колекції має поля: available(boolean), description(string), image(string), name(string), price(number), type(string).