



Hermes Logistics Presents

PetasosExpress

By Illia Popov, Ahmad Aljawish, William Margalik and Dylan Ashton



Main Components of the Application



01

Authentication

The user can create an account with PetasosExpress or use Google to login

02

Shopping

The user can search for products add them to cart and purchase them.

03

Delivery Tracking

The user can track the delivery location and sensor readings of Petasos (delivery robot) assigned to their order

04

App Personalization

The user personalize the App and their Account for their liking.

05

Feedback loop

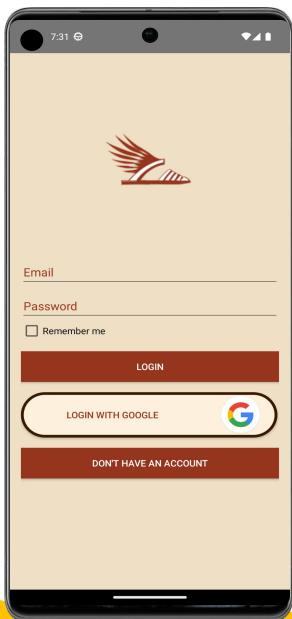
The user can submit their feedback to help us make the App better



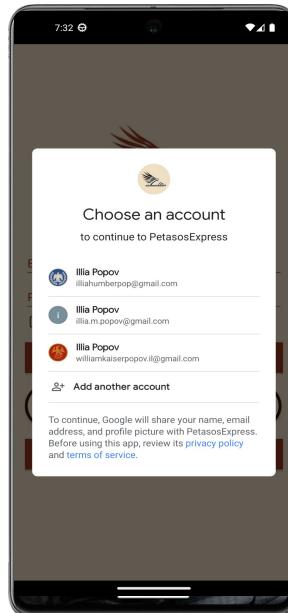
01

Authentication

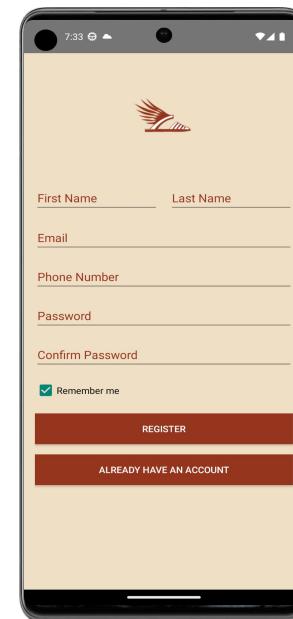
Login



Login with Google



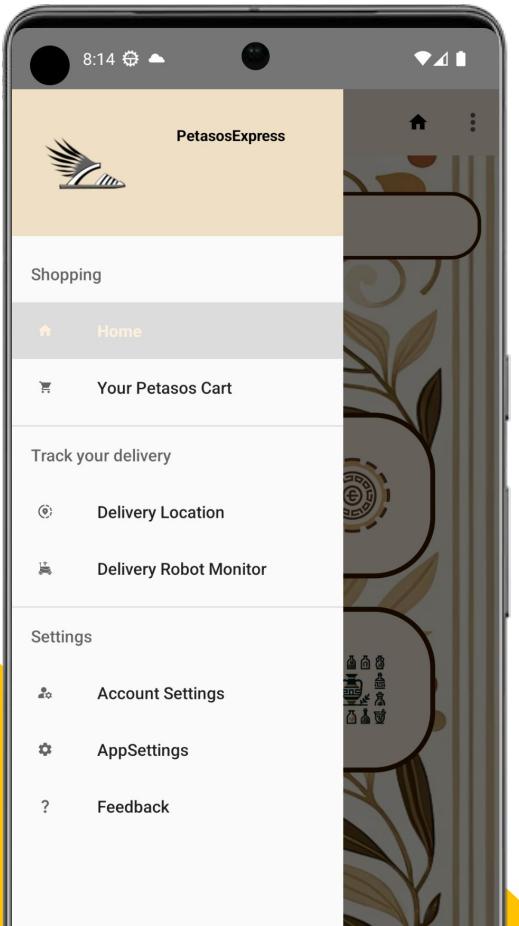
Registration





1.5

Navigation



Intuitive Access:

- Provides users easy access to diverse features like ordering, tracking, and sensor screens.
- Enhances user experience via intuitive, one-touch navigation.

Space-Efficient Design:

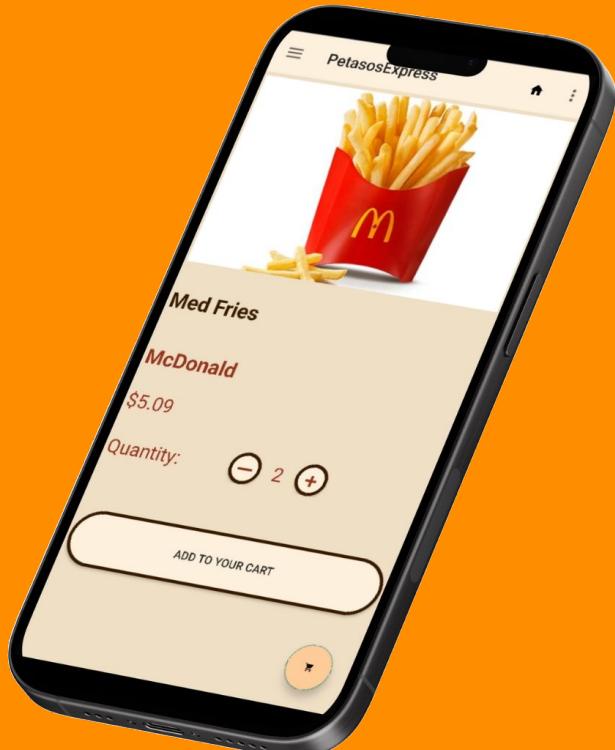
- Optimizes screen space by neatly organizing functions in a hidden drawer.
- Maintains a clutter-free layout, ensuring seamless user interaction.



02

Shopping

How to shop and buy things!





Search for Products!

Home & Search Screens

On the home screen users can type what they are looking for in the search tab to go to the search screen or click on one of the buttons filter specific type of products.



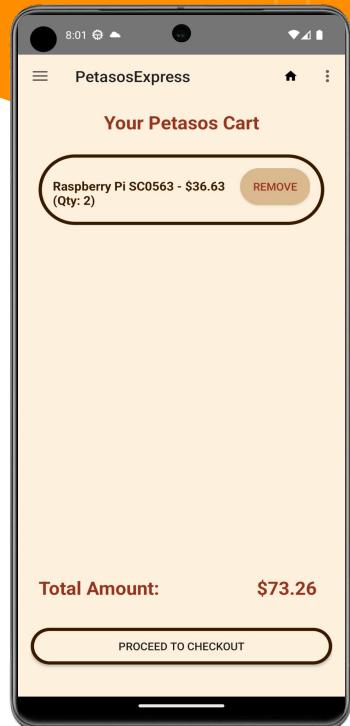
Users can search for the items by product's name, type or producer to find exactly what they need!

Our Search engine uses our database of approved products to fetch information about items' names, prices, category, etc, and display it on the search screen.





Cart Screen



Add them to your cart!

Product Screen

On the product screen user can get more information about the product. Even check at how it looks!

Users can chose the quantity of items they want to add to their cart and go there right away using FAB!

On the cart screen user can make final decision if they want to remove something before proceeding to checkout.



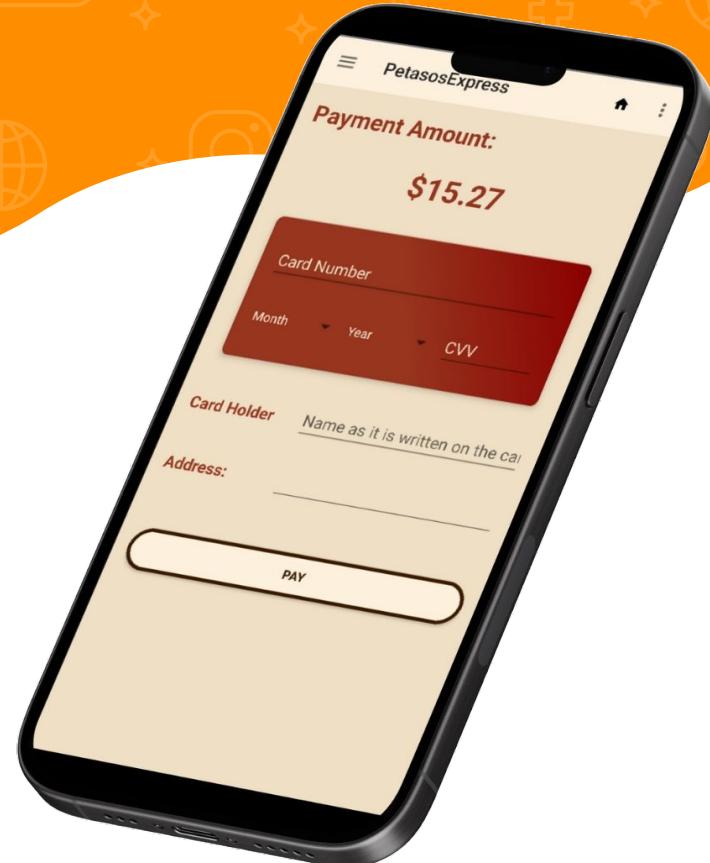


Place your order!

Payment Screen

Finally users can pay for their order!

Card number, Expiration date, CVV, card Holder Name and Address should be filled to assign the order.

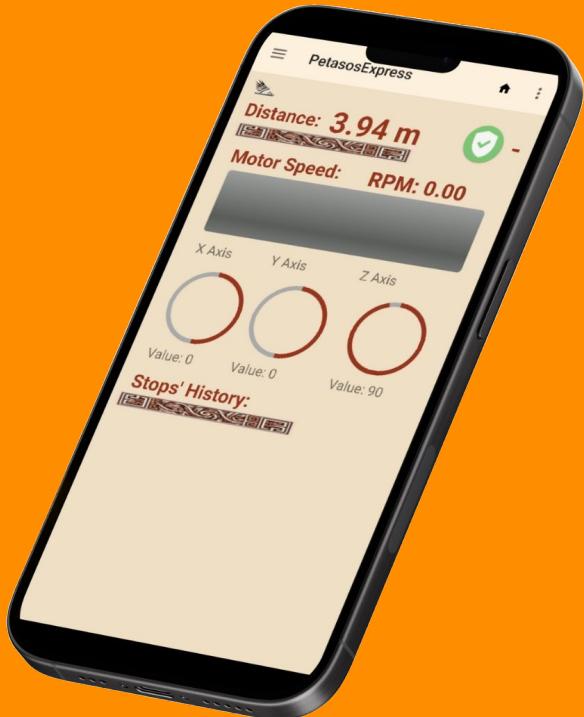




03

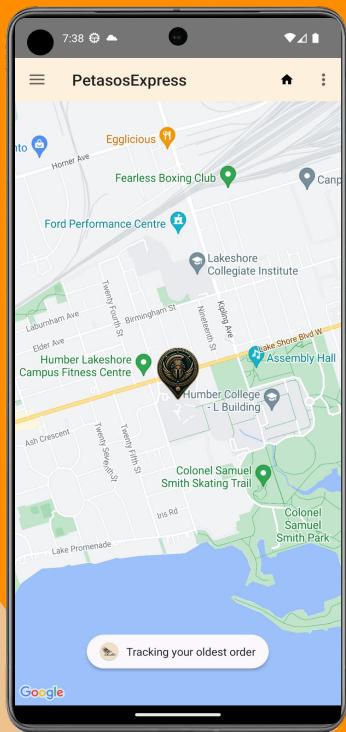
Delivery Tracking

Track your order!





Tracking Location of the delivery

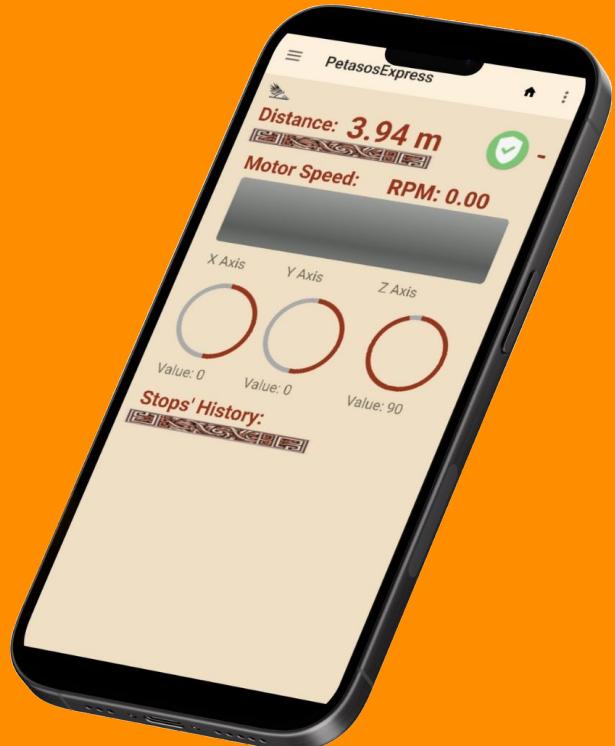


Geolocation Screen

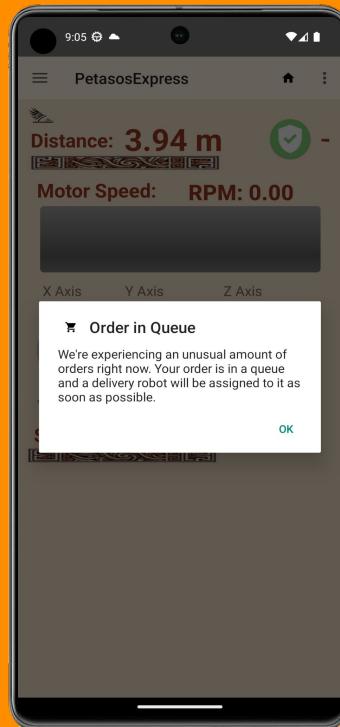
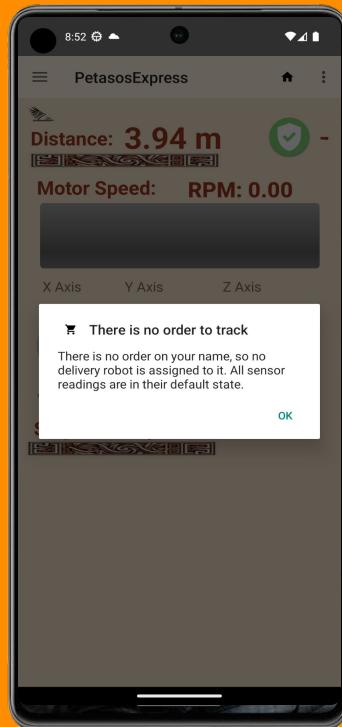
After the order was placed users can track location of the delivery robot using “Delivery Location” screen to know exactly how close their delivery is to them!

Tracking Petasos Condition

- Proximity/Distance
- Motor Speed in RPM
- Magnetometer for tilt and direction
- Stops History



Tracking works only there is something to track!

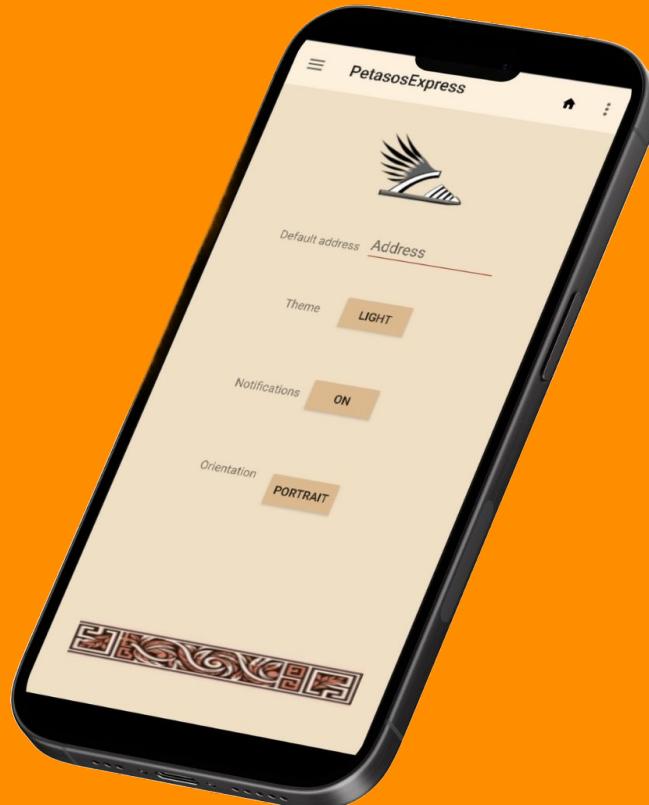




04

Personalization

Personalize your Petasos

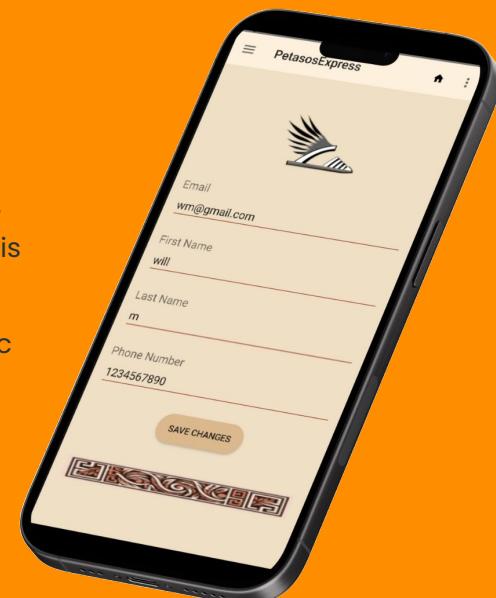
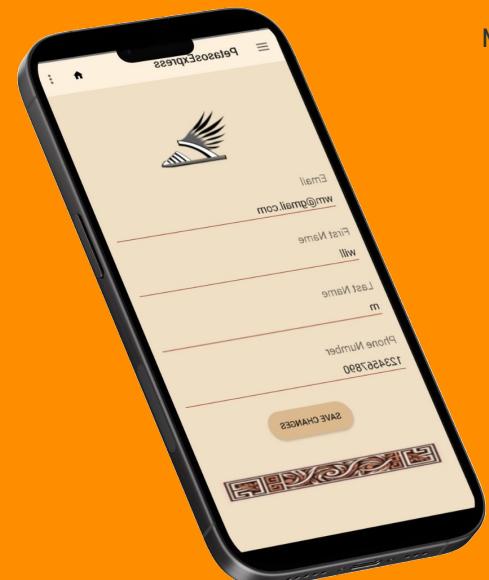


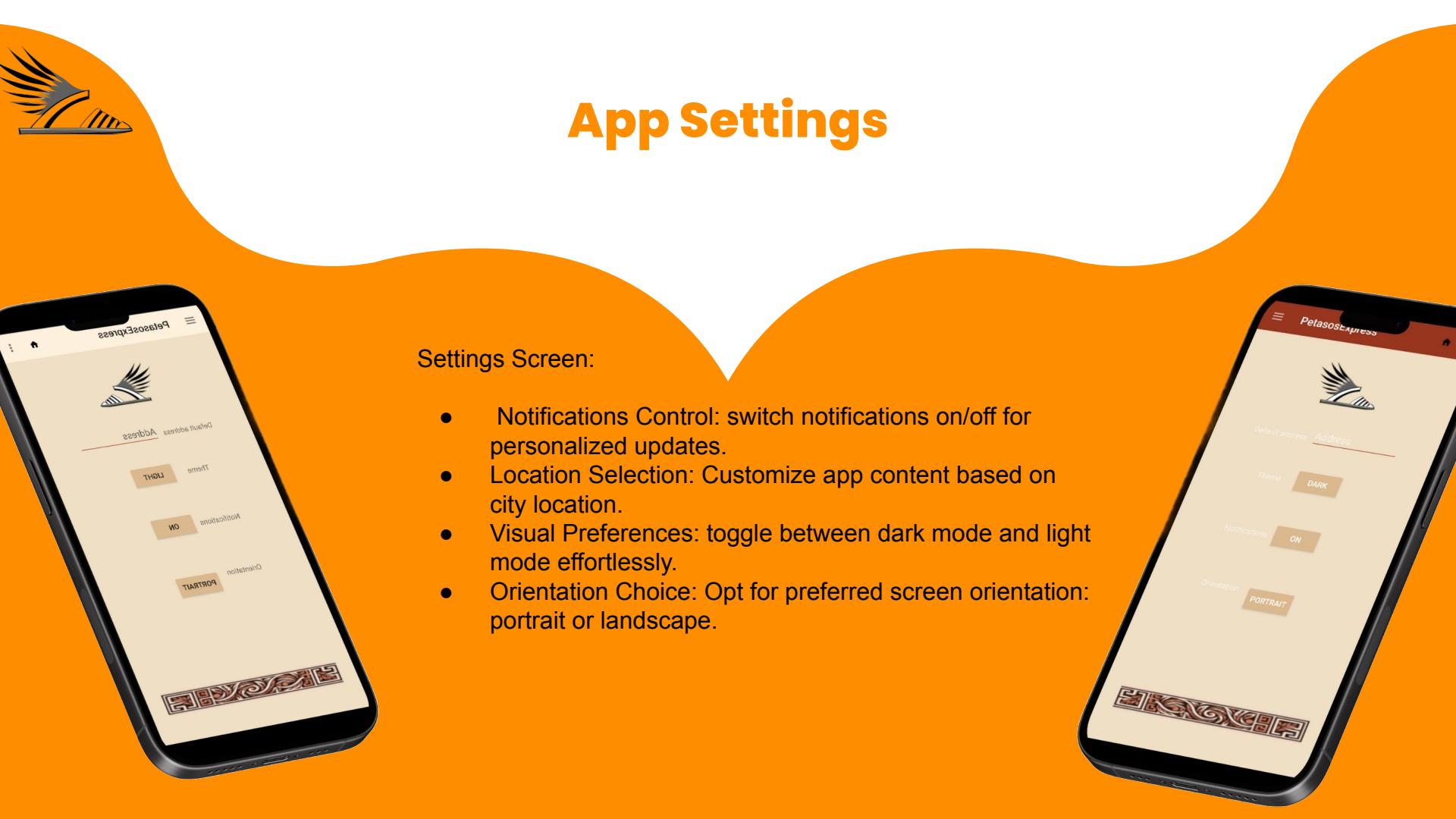


Manage Account

Manage Account screen :

- Email Display: Shows the user's current email address, helping them to verify their account identity.
- Personal Information Update: Allows users to edit and save their first name, last name, and phone number, ensuring their profile is up-to-date.
- Phone Number Formatting: Automatically removes non-numeric characters from the phone input, maintaining a standardized format for contact information.
- Data Validation: Includes checks for empty fields and correct phone number length, ensuring all entered information is valid before saving.
- User Feedback: Displays Toast messages for successful data updates or errors, providing immediate feedback on user actions.





App Settings

Settings Screen:

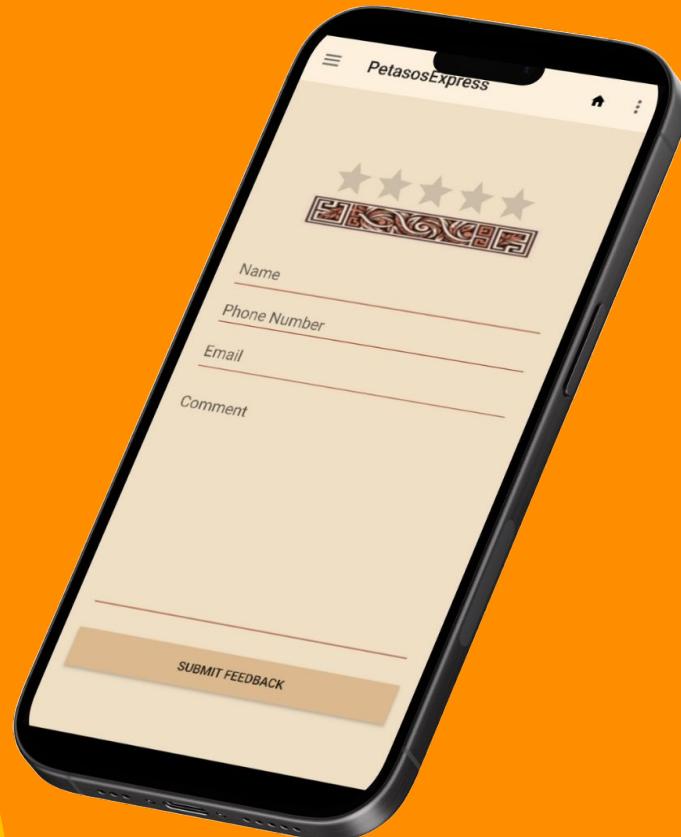
- Notifications Control: switch notifications on/off for personalized updates.
- Location Selection: Customize app content based on city location.
- Visual Preferences: toggle between dark mode and light mode effortlessly.
- Orientation Choice: Opt for preferred screen orientation: portrait or landscape.



05

Feedback

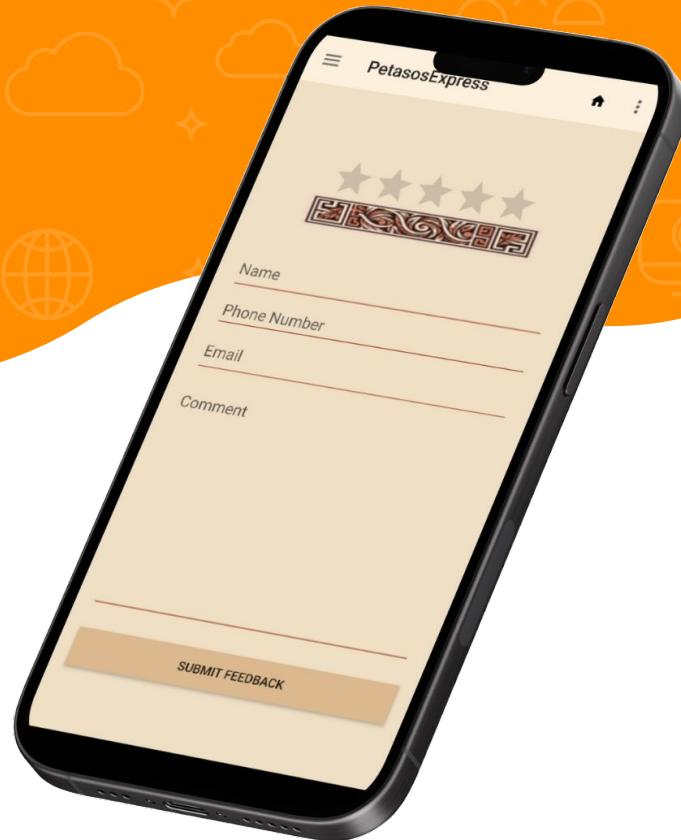
Contact us, let's make
PetasosExpress better together!





Feedback page

- Star Rating system
- Store feedback and respond
- Responsive UI
- Unique responses



Offline Mode





How did we test:



Junit Test cases

```
no usages  ahmadaljawish5348 +1
9  public class GeneralValidationUtilsTest {
10
11      no usages  ahmadaljawish5348 +1
12      @Test
13      public void testIsValidName_ValidName() {
14          assertTrue(GeneralValidationUtils.isValidName("John Doe"));
15      }
16
17      no usages  ahmadaljawish5348 +1
18      @Test
19      public void testIsValidName_InvalidName_Empty() {
20          assertFalse(GeneralValidationUtils.isValidName(""));
21      }
22
```

Junit Test cases

```
20
no usages ahmadaljawish5348 +1
21
@Test
22 public void testIsValidPhoneNumber_ValidPhoneNumber() {
23     assertTrue(GeneralValidationUtils.isValidPhone("1234567890"));
24 }
25
no usages ahmadaljawish5348 +1
26
@Test
27 public void testIsValidPhoneNumber_InvalidPhoneNumber() {
28     assertFalse(GeneralValidationUtils.isValidPhone("12345"));
29 }
30
no usages ahmadaljawish5348 +1
31
@Test
32 public void testIsValidRating_ValidRating() {
33     assertTrue(GeneralValidationUtils.isValidRating(3.5f));
34 }
35
```

Junit Test cases

```
35      no usages  ahmadaljawish5348 +1
36
37  @Test
38  public void testIsValidRating_InvalidRating() {
39      assertFalse(GeneralValidationUtils.isValidRating(0f));
40
41      no usages  ahmadaljawish5348 +1
42  @Test
43  public void testIsValidComment_ValidComment() {
44      assertTrue(GeneralValidationUtils.isValidComment("This is a valid comment."));
45
46      no usages  ahmadaljawish5348 +1
47  @Test
48  public void testIsValidComment_InvalidComment_Empty() {
49      assertFalse(GeneralValidationUtils.isValidComment(""));
50  }
```



Espresso Test cases

Espresso Test cases



```
no usages IlliaPopov1791
@Test
public void testElectronicsButton() {
    onView(withId(R.id.electronicsButton)).perform(click());
    onView(withId(R.id.searchEditText)).check(matches(withText("technology")));
    onView(withId(R.id.search_list)).check(matches(isDisplayed()));
}

no usages IlliaPopov1791
@Test
public void testMedicineButton() {
    onView(withId(R.id.medicalButton)).perform(click());
    onView(withId(R.id.searchEditText)).check(matches(withText("medicine")));
    onView(withId(R.id.search_list)).check(matches(isDisplayed()));
}
```



```
no usages ahmadaljawish5348
@Test
public void testOtherButton() {
    onView(withId(R.id.otherButton)).perform(click());
    onView(withId(R.id.searchEditText)).check(matches(withText("other")));
    onView(withId(R.id.search_list)).check(matches(isDisplayed()));
}

no usages IlliaPopov1791
@Test
public void testFabCartButton() {
    onView(withId(R.id.fab_cart)).perform(click());
    onView(withId(R.id.cart_items_list)).check(matches(isDisplayed()));
}
```

Espresso Test cases



HomeTest

Status: 6 passed | 6 tests, 39 s 311 ms

Filter tests:

Tests	Duration	Pixel_7_Pro_API_34
Test Results	14 s	6/6
HomeTest	14 s	6/6
testFoodButton	4 s	✓
testSearchFunctionality	4 s	✓
testFabCartButton	1 s	✓
testMedicineButton	1 s	✓
testOtherButton	1 s	✓
testElectronicsButton	1 s	✓

Robletric Test cases

```
no usages IlliaPopov1791+1
11 @RunWith(RoboletricTestRunner.class)
12 public class PaymentValidationTest {
13
14     no usages IlliaPopov1791
15     @Test
16     public void testValidCreditCardNumber() {
17         assertTrue(validateCreditCardNumberLength("1234567890123456"));
18     }
19
20     no usages IlliaPopov1791
21     @Test
22     public void testInvalidCreditCardNumber() {
23         assertFalse(validateCreditCardNumberLength("1234"));
24     }
25
26     no usages IlliaPopov1791
27     @Test
28     public void testValidExpirationMonth() { assertTrue(validateExpirationMonth( selectedMonthPosition: 5)); //May }
```

Robletric Test cases

```
no usages ▲ IlliaPopov1791
30
31 @Test
32     public void testInvalidExpirationMonth() {
33         assertFalse(validateExpirationMonth( selectedMonthPosition: 0)); //'no selection'
34     }
35
36 no usages ▲ IlliaPopov1791
37 @Test
38     public void testValidExpirationYear() { assertTrue(validateExpirationYear( selectedYearPosition: 3)); //year 2023 }
39
40 no usages ▲ IlliaPopov1791
41 @Test
42     public void testInvalidExpirationYear() {
43         assertFalse(validateExpirationYear( selectedYearPosition: 0)); //'no selection'
44     }
45 no usages ▲ ahmadaljawish5348
46 @Test
47     public void testValidSecurityCode() { assertTrue(validateSecurityCodeLength("123")); }
48
49 no usages ▲ ahmadaljawish5348
50 @Test
51     public void testInvalidSecurityCode() { assertFalse(validateSecurityCodeLength("12")); }
52
53 no usages ▲ ahmadaljawish5348
54 @Test
55     public void testValidCardHolderName() { assertTrue(validateCardHolderName("John Doe")); }
```



Submission To Google Play

Dashboard Last 30 days ▾

 PetasosExpress
ca.hermeslogistics.itservices.petasosexpress • [View on Google Play](#) View releases overview

Update status In review • [Go to publishing overview](#)

Production
Active • 0 active devices • 1 country/region

App has been submitted to google play on 05/12/2023

Currently waiting on google to publish the App after reviewing





Sprints:

▼ Epic 4: Story 12: Implement Offline Data Storage and Synchronization

	Task	Person	Status	Priority	Size	Timeline	Date of comple...	i
<input type="checkbox"/>	Task 1: Relocate unnecessary for the server data to Shared Preferences		IP	Done		Small	Nov 15 - 19	Nov 19
<input type="checkbox"/>	Task 2: Implement Online Data Synchronization		IP	Done		Medium	Nov 15 - 19	Nov 19
<input type="checkbox"/>	Task 3: Handle Data Conflicts		IP	Done		Small	Nov 15 - 19	Nov 19
<input type="checkbox"/>	Task 4: Ensure Smooth Offline-Online Transitions		IP	Done		Small	Nov 15 - 19	Nov 19
<input type="checkbox"/>	+ Add task						Nov 15 - 19	

▼ Epic 4: Story 13: Create a Floating Action Button

	Task	Person	Status	Priority	Size	Timeline	Date of comple...	i
<input type="checkbox"/>	Task 1: Create a Floating Action Button in the targeted classes		DA	Done		Small	Nov 21 - 29	Nov 28
<input type="checkbox"/>	Task 2: Redesign targeted classes' portrait layouts to include FAB		DA	Done		Small	Nov 21 - 29	Nov 28
<input type="checkbox"/>	Task 3: Redesign targeted classes' landscape layouts to include FAB		DA	Low		Medium	Nov 21 - 29	Nov 28
<input type="checkbox"/>	Task 4: Implement functionality of the Floating Action Button		DA	High		Small	Nov 21 - 29	Nov 28
<input type="checkbox"/>	Task 5: Ensure that FAB works as intended on all screens		DA	Done		Small	Nov 21 - 29	Nov 28
<input type="checkbox"/>	+ Add task						Nov 21 - 29	



Sprints:

▼ Epic 4: Story 14: Modify the feedback screen

	Task	Person	Status	Priority	Size	Timeline	Date of comple...	ⓘ
<input type="checkbox"/>	Task 1: Implement delay for submission of the feedback		Done			Nov 21 - 30	Nov 26	
<input type="checkbox"/>	Task 2: Add the progress bar to represent submission process in the desi...		Done			Nov 21 - 30	Nov 26	
<input type="checkbox"/>	Task 3: Implement the progression on the progress bar		Done			Nov 21 - 30	Nov 30	
<input type="checkbox"/>	Task 4: Restrict Feedback Submission to once in 24 hours		Done			Nov 21 - 30	Nov 26	

▼ Epic 4: Story 15: Payment and Order Tracking Implementation 10 Tasks

	Task	Person	Status	Priority	Size	Timeline	Date of comple...	ⓘ
<input type="checkbox"/>	Task 1: Creating portrait layout for Product Screen		Done			Nov 21 - Dec 10	Nov 23	
<input type="checkbox"/>	Task 2: Creating landscape layout for Product Screen		Done			Nov 21 - Dec 10	Nov 29	
<input type="checkbox"/>	Task 3: Implementing fetching of the correct data for the Product screen		Done			Nov 21 - Dec 10	Nov 23	
<input type="checkbox"/>	Task 4: Designing portrait layout for Cart Screen		Done			Nov 21 - Dec 10	Nov 25	
<input type="checkbox"/>	Task 5: Designing landscape layout for Cart Screen		Done			Nov 21 - Dec 10	Nov 30	
<input type="checkbox"/>	Task 6: Implement Storing and Displaying Cart content		Done			Nov 21 - Dec 10	Nov 25	
<input type="checkbox"/>	Task 7: Developing portrait UI for Payment Screen		Done			Nov 21 - Dec 10	Nov 21	
<input type="checkbox"/>	Task 8: Developing landscape UI for Payment Screen		Done			Nov 21 - Dec 10	Nov 2	
<input type="checkbox"/>	Task 9: Implementing Order placing		Done			Nov 21 - Dec 10	Nov 27	
<input type="checkbox"/>	Task 10: Changing sensors code to work with assigned Delivery Robot O...		Done			Nov 21 - Dec 10	Nov 27	
<input type="checkbox"/>	+ Add task					Nov 21 - D...		Help



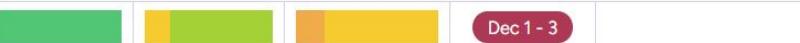
Sprints:

▼ Epic 4: Story 16: Testing

	Task	Person	Status	Priority	Size	Timeline	Date of comple...	i
<input type="checkbox"/>	Task 1: Junit4 testing for General Verification Processes		Done			Nov 15 - Dec 1	Dec 1	
<input type="checkbox"/>	Task 2: Junit4 testing for Product		Done			Nov 15 - Dec 1	Dec 2	
<input type="checkbox"/>	Task 3: Robletric Testing		Done			Nov 15 - Dec 1	Dec 2	
<input type="checkbox"/>	Task 4: Espresso Testing		Done			Nov 15 - Dec 1	Dec 1	
<input type="checkbox"/>	Task 5: Validate all tests pass and function as expected		Done			Nov 15 - Dec 1	Dec 2	

▼ Epic 4: Story 17: Code refactoring and Cleanup

	Task	Person	Status	Priority	Size	Timeline	Date of comple...	i
<input type="checkbox"/>	Task 1: Removing all Commented-Out Code		Done			Dec 1 - 3	Dec 3	
<input type="checkbox"/>	Task 2: Implement Proper Code Formatting		Done			Dec 1 - 3	Dec 3	
<input type="checkbox"/>	Task 3: Implement Proper Naming convention		Done			Dec 1 - 3	Dec 3	
<input type="checkbox"/>	Task 4: Refactoring the App		Done			Dec 1 - 3	Dec 3	
<input type="checkbox"/>	Task 5: Verifying Functionality after the refactoring		Done			Dec 1 - 3	Dec 3	
<input type="checkbox"/>	+ Add task							





THANK YOU

