Android NanoDegree

Capstone Project

Description

Logistic App

By: Juan Jose Perez

 Index

GitHub Username: PlumaBrava

Logistic App

Description

Organize and Update your business logistic wherever you are.
Logistic App lets you administrate your logistic business, using proved process, minimizing your effort.

Enjoy our app wherever you are!!! And run easily your logistic busyness

Intended User

This app is for people who runs a small logistic busyness.
Especially those who are on the street all the time and need to organize their tasks.

Features

The logistic process has the following steps

Orders: in this step of the process the user receive the custom orders.

- Takes the client order selecting the client, the product and the quantity needed.
- Summarizes the total of product in ordered without picking or deliver.
- Marc an Order as standard order for a Client. When a New order is created, it will have the products and the quantities of standard order.

Picking: in this step of the process the user assign the custom order to a Picking Order. This is useful for determining how much of each product should be loaded into the truck.

- Crete Picking order
- Select a Picking order to assign custom orders.
- Assign Customer Orders to an Open Picking Order
- Close a Picking order in order to be prepared for delivery.
- Deliver a Picking Order, Send the order to the following step of the process.

Delivery: In this step of the process, the user deliver de products loaded in the truck. Some time, the quantities of products deliver is different to the product originally requested. So the user can modify this quantities.

- Display the total stock left in the truck
- Display the total of products in custom order to be deliver and show
    - Green if the product loaded in the truck is greater o equal to the products in custom orders
    - Red if the product loaded in the truck is small than the products in custom orders
- When the user arrives to the delivery address, confirm or modify de quantities delivered and closes the process, recording the payment and some final comments.

Customization

- Administrate a the Clients List (Name, delivery address, etc)
- Administrate de Product List (Name, picture, price)

SignIn

- The user must singIn to use the app. Is done using google play services Identity.

Widget

Widget to show the products in delivery (showing quantities required in custom orders, loaded and the balance)

User Interface Mocks

1) Products

Screen 1.1: Product: In this screen the user can see a list of the created product and

- Create a new product
- Select the product to modify (click)
- Delete a product

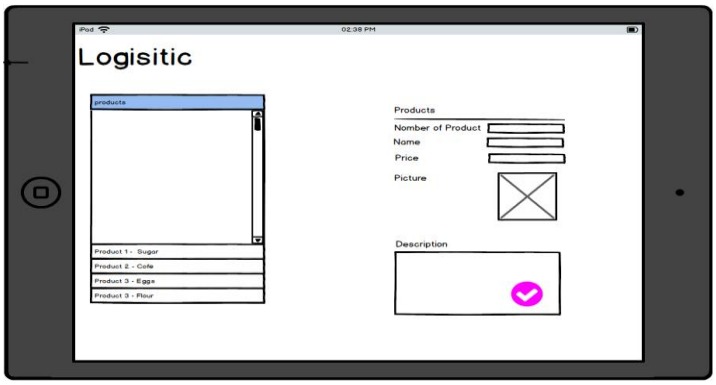
Screen 1.2: Product Detail: In this screen the user can see a Create or modify a customer. Use an Intent to take a picture and save it.

Screen 1.3: Product In a Tablet: In this screen the user can see the two Products Screen in one page.

Screen 1.1: Product:

Screen 1.2: Product Detail:

Screen 1.3: Product In a Tablet:



2) Customers

Screen 2.1: Customer: In this screen the user can see a list of the created customes and:

- Create a new customer (+)
- Select the customer to modify (click)
- Delete a customer (slide to the left)

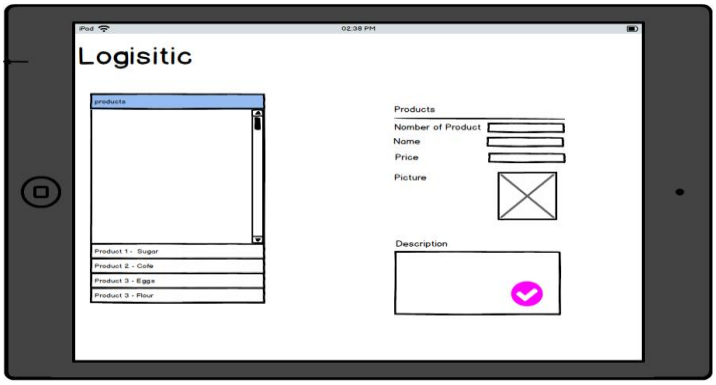
Screen 2.2: Customer Detail: In this screen the user can see a Create or modify a product. Use an Intent to take a picture and save it. The user can select a contact and link

Screen 2.3: Customer In Tablets: In this screen the user can see the client list and the details of the selected one page.

Screen 2.1: Customers:

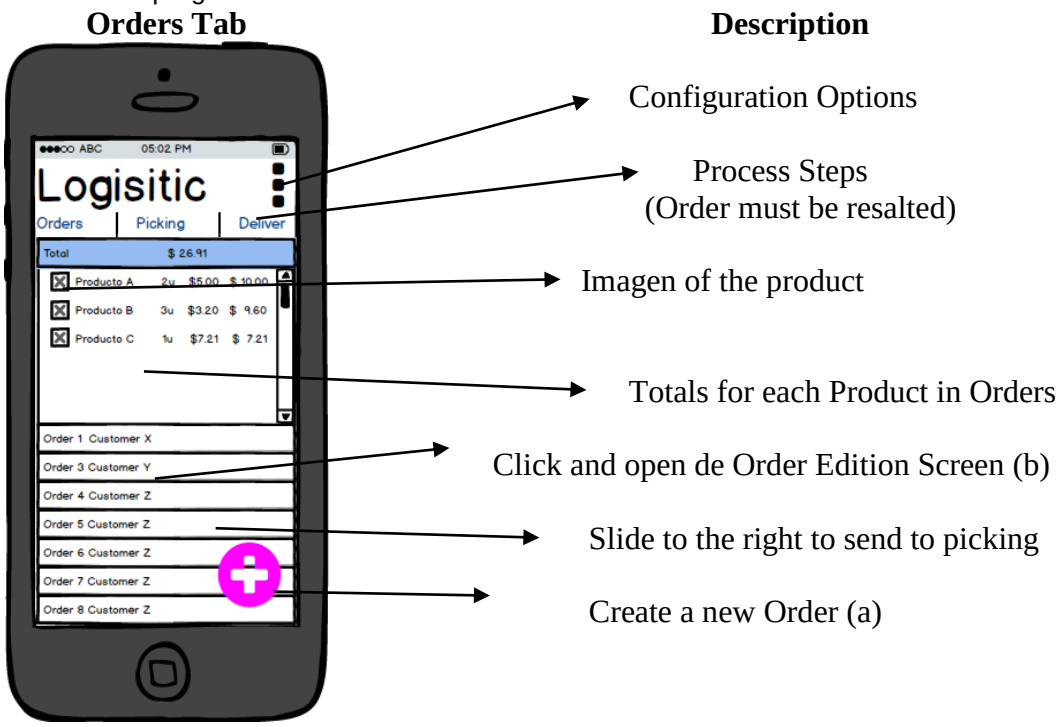
Screen 1.2: Customer Detail:

Screen 1.3: Customers In a Tablet:



Screen 3: Main Screen

This screen display 3 tabs steps of the logistic process.(orders, picking and delivery).The user through the process flows as this progresses.



- Configuration options, open the menu to administrate product and customers.
- When the user touch + (a) : Create a new Order.
  Open OrderActivity with Extra=Create_New_Order

- When the user click an order (b) : Edit the Order touched.
  Open OrderActivity with Extra=Edit_Order and the id of the Order touched.

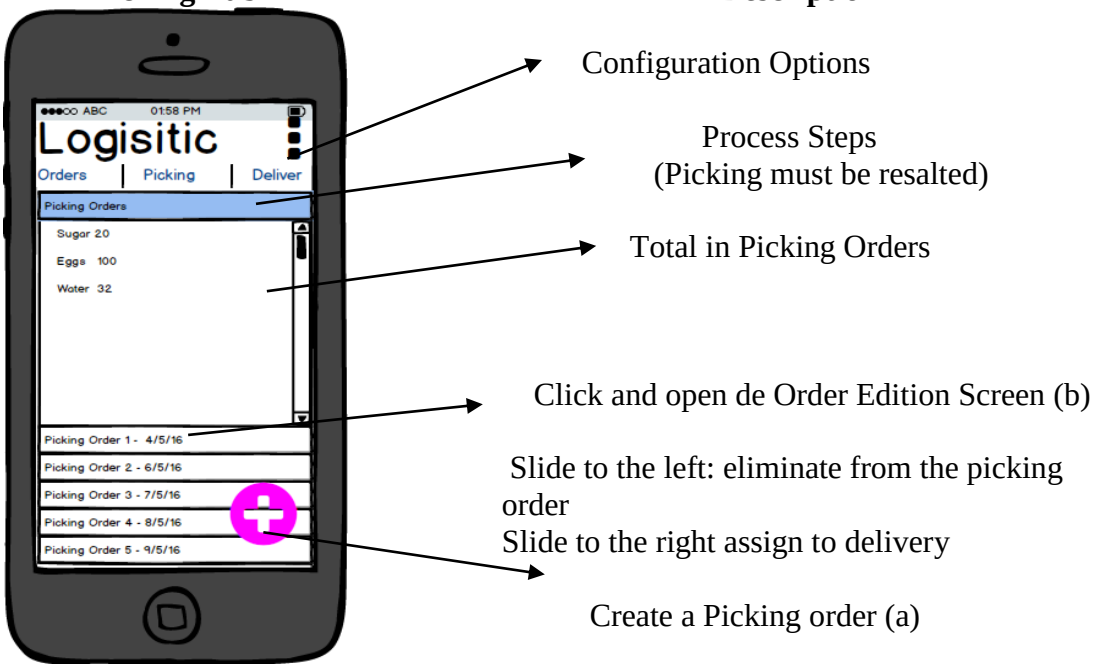
- When the user slide to the right (b) : Edit the Order touched.
  1) Assign the order to the open picking Order
     1. If there are no picking order sugest to create one or select .

The Custom Order have a Status, The status represents the Custom Order State. They are:

Initial
Deleted
Assigned to a Picking Order
In delivering
Delivered

Notes: when there are no orders to display show “No Orders to display”

Picking Tab **Description**



- Configuration options, open the menu to administrate product and customers.
- This tab, shows the Picking Orders in Status Initial
- When the user touch Plus (a) : Create a New picking order Assigning a Number and the date, Status Initial.
- When the user c*lick* an order (b) : Add a transition effect. Open de Picking Detail Screen to see the totals of the
- When the user *slide to the left* (b) :
  If the are no custom orders assigned , The status is changed to Deleted before a dialog to confirm the action
- When the user *slide to the right* (b) :

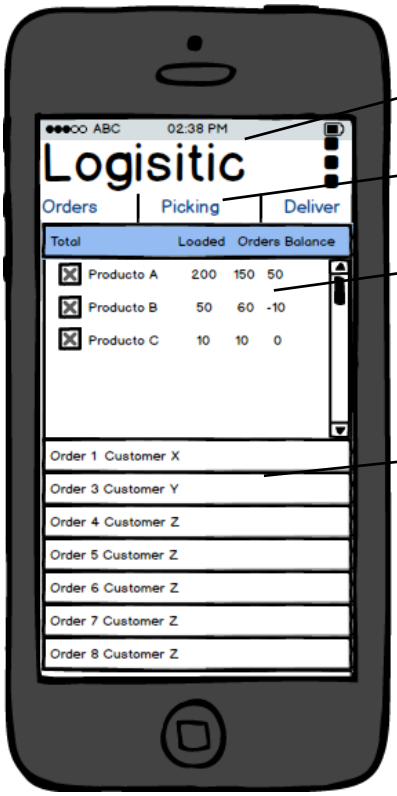
If there are no custom orders assigned, Display a Dialog say that this Picking order can't be delivered because its empty. Else, modify the status to InDelivery. Only one picking order can stay "in delivering" status.

- when there are no orders to display show "No Orders to display"
- Share (a) can be user to

The Picking Order have a Status, The status represents the Picking Order State. They are:

Initial
Deleted
In delivering
Delivered

Deliver Tab



Description

Configuration Options

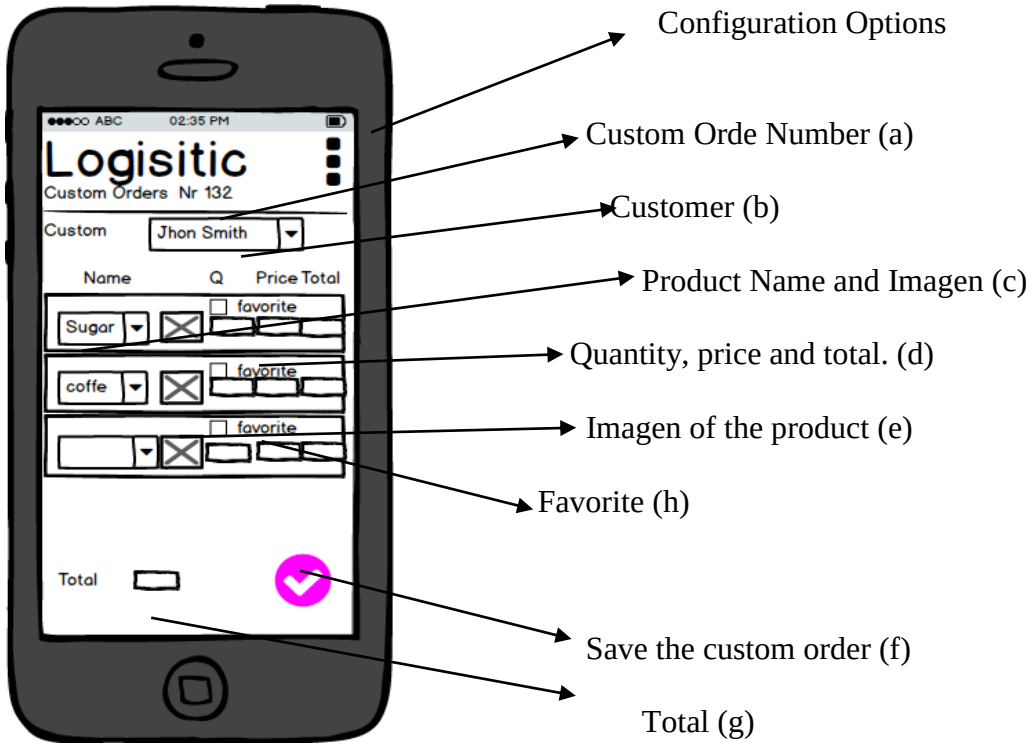
Process Steps
(Delivery must be emphasize)

Display the total Loaded, total in orders and the balance

Click and open de Order Edition Screen (b)

Slide to the left: eliminate from the picking order

Screen 4 : Custom Order Detail



- Custom Order Number (a): is the Id of the order, It will be in black if it's a new order

- Custom (b) if the user choose a client and it has and Order in Initial stats, alert saying "This customer has x orders open"

In order to made this process easy, when a new order is created and the client is selected, add all the products and quantities marked as favorite.

- Product (c), Select a Product from the table of products if the product already exist in this order, Show a dialog saying "The product *name* was assigned"

- Imagen of the product (e), display the picture of the product or a default

- Quantity, price and total(d)
  The user can modify the only the quantities
  Price and Total are blocked to modify.
  Price, shows the price loaded in the custom price, if the quantities changed or the order si new load the price from the product table.

  Total=prices*quantity

- Imagen of the product (e)
- Save (f): Save the custom order in the table Orders and Detail Orders.

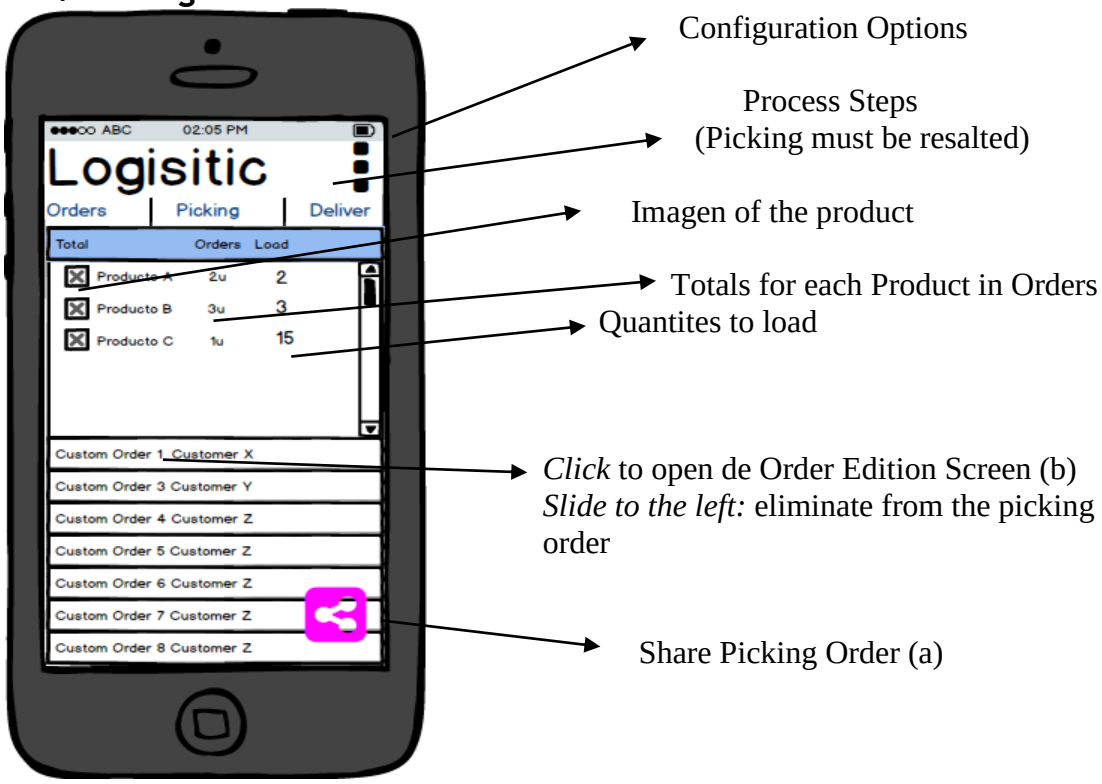
After that show an interstitial advertising in the free flavor.

- Total (g) Sum of the total of each product.
- Favorite (h) : this marc is used to generate automatically custom orders.

  This marc must be unique for each customer and Product. When marc, verify is this combination of customer and Product exists if false marc, if true make a dialog to confirm if this new is the favorite If its confirmed delete de previous marc and select this new.

- Pressing back, return to screen. Return to Order's tab without saving. If there are unsaved changes must ask if this will be discarded
-

Screen 5 : Picking Detail

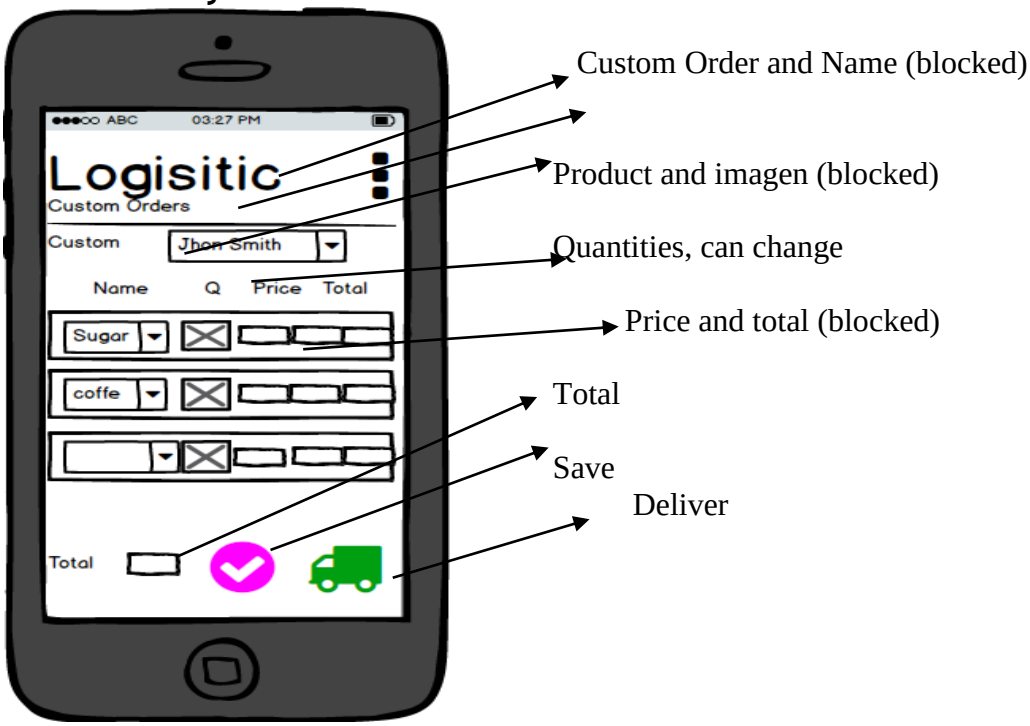


- When the user touch Share (a) : share the totals of the Picking order. The user can send this information to someone who helps him.

- When the user click an order (b) : Edit the Order touched.
  Open OrderActivity with Extra=Edit_Order and the id of the Order touched.
  In this case modify quantities. Quantities delivered is modified in the delivery step.
  Add a transition effect.
- When the user slide to the right (b) : nothing is done. Add a transition effect.

- When the user slide to the left (b) : cancel the assign of the custom order to the picking order and Modify the status of the custom order to Initial.
  Add a transition effect.
- Pressing back, return to screen. Picking tab

Screen 6 : Delivery Order Detail



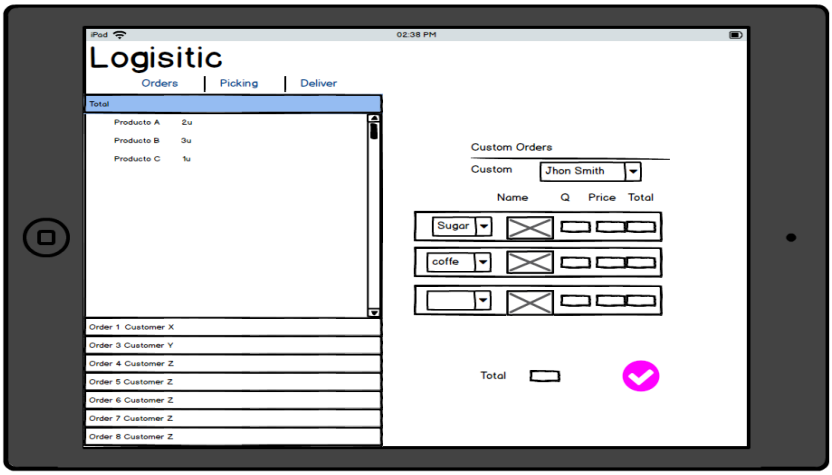
- This screen is similar to the Custom Order Screen, but the use can modify the quantities only.

It the quantities change, total change using the new quantities

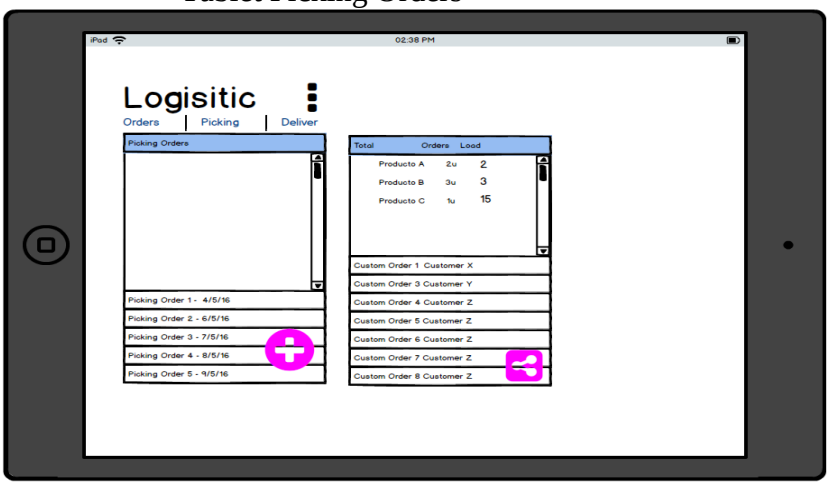
- Save: save the quantities modification in the CustomOrderDetail Table
- Deliver
  Is blocked if same changed weren't saved.
  When deliver is clicked, Display a Dialog to confirm the delivery.. If its confirmed, Modify the state of the custom order to delivered.

Screen 7 : Tablet Screens

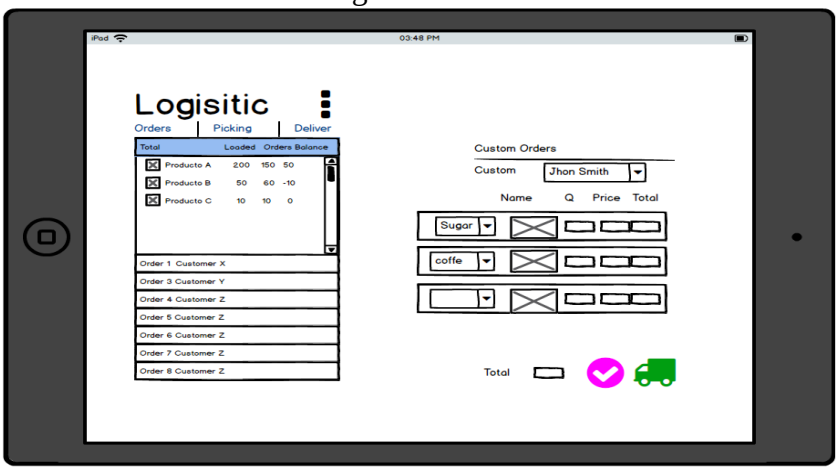
Tablet Custom orders



Tablet Picking Orders



Tablet Picking Orders



- In table, in landscape Mode, display the list and detail in the same screen.

Key Considerations

app data persistence

Implement a custom Provider using schematic

The Table **Products** have the following fields.

Field	Type	Referene
Id	Integer-Primary key	PRODUCT_ID
Name	String (20)	PRODUCT_NAME
Description	String (70)	PRODUCT_DESCRITION
Imagen	String	PRODUCT_IMAGEN
Price	Long (**REAL**)	PRODUCT_PRICE
Status	integer	PRODUCT_STATUS

The Table **Customers** have the following fields.

Field	Type	Referene
Id	Integer-Primary key	CUSTOMER_ID
Name	String (20)	CUSTOMER_NAME
LastName	String (70)	CUSTOMER_LASTNAME
DeliveryAddress	String (70)	CUSTOMER_DELIVEYADDRES
Imagen	String	CUSTOMER_IMAGEN
ContactReference	String	CUSTOMER_REFERENCE

Use the Content Provider for contacts to get a reference when create a Customer.

The Table **CustomOrders** have the following fields.

Field	Type	Referene
Id	Integer-Primary key	CUSTOM_ORDER_ID
Ref_Customer	Integer	CUSTOM_ORDER_REF_CUSTOMER
Creation_Date	Date	CUSTOM_ORDER_CREATION_DATE
StatusCustomOrder	Integer	CUSTOM_ORDER_STATUS_CUSTOM_ORDER
Date_of_Picking_Asignation	Date	CUSTOM_ORDER_DATE_OF_PICKING_ASGNATION
Date_of_Deliver	Date	CUSTOM_ORDER_DATE_OF_DELIVER
TotalPrice	Long (**REAL**)	CUSTOM_ORDER_TOTAL_PRICE

The Table **CustomOrdersDetail** have the following fields.

Field	Type	Referene
Id	Integer-Primary key	CUSTOM_ORDER_DETAIL_ID
Ref_CustomOrder	Integer	CUSTOM_ORDER_DETAIL_REF_CUSTOM_ORDER
Ref_Product	Integer	
Product_Name	String	
Quantity	Long (**REAL**)	
Price	Long (**REAL**)	
Favorite	Binary	

The Table **PickingOrders** have the following fields.

Field	Type	Referene
Id	Integer-Primary key	PICKING_ORDER_ID
Creation_Date	Date	PICKING_ORDER_CREATION_DATE
Comments	String	PICKING_ORDER_COMMENTS
Status	Int	PICKING_ORDER_STATUS

The Table **PickingOrdersDetail** have the following fields.

Field	Type	Referene
Id	Integer-Primary key	PICKING_ORDER_DETAIL_ID
Ref_PickingOrder	Integer	PICKING__ORDER_DETAIL_REF_CUSTOM_ORDER
Ref_Product	Integer	PICKING__ORDER_DETAIL_REF_PRODUCT
Product_Name	String	PICKING__ORDER_DETAIL_NAME
Quantity	Long (**REAL**)	PICKING__ORDER_DETAIL_QUANTITY

Note: When the user signIn to use the app, save in preference the result of the transaction. If its OK, don’t sign.

Describe any corner cases in the UX.

This item is in the screen description.

Libraries to be used

Use the following libraries

- **Schematic** to create and manage a Content Provider
- **Picasso:** to manage the imagen of the products
- **Expresso:** to run the test

Required Tasks

Task 1: Project Setup

- Configure libraries
- Manifest permission

Task 2: Implement UI for Main Activity and Fragment

- Build UI for MainActivity (Tabbed)
- Configure the menus
- Build UI for Activity for Client and client Detail
- Build UI for Activity for Product and product Detail

Task 3: Implement the Content Provider

- Build the Content Provider using the tables of this document.

Task 4: Products Activity and ProductoDetailActivity

- Develop Products Activity and ProductsDetailActivity.
- Build the layout, Loaders, Validate data and save it.
- Use Intents to take a picture for products
- Create an expresso test for this activity and verify woks properly.

Task 5: Customs Activity and CustomsDetailActivity

- Develop Customs Activity and CustomsDetailActivity.
- Build the layout, Loaders, Validate data and save it.
- Use Intents to take a picture for products
- Create an expresso test for this activity and verify woks properly.

Task 4: Implement Customs Orders Fragment and CustomOrdersDetailActiviy

- Build UI for the Custom Order activity
- Create layout
- Configure Plus button of main activity to create a Custom Order
- Implement the validation
    - warn if there are other Order in Initial State.
    - Verify that the products are only one time in an order
    - Verify that the quantities are >0
    - Build the validation to select a standard order.

- Estimate the total Price
- Implement the save Button, cancel the operation using back.
- Create an expresso test for this activity and verify woks properly.

Task 4: Implement Orders and CustomOrdersDetail Fragment

- Build UI for the Orders and CustomOrdersDetail fragments
- Implement the adapters for custom orders
- Implement the Loaders
- Implement the querys to show the totals.
- Build the onClick operation to modify a Custom Order. Reuse the Custom Order Activity.
- Build onSlidetoTheRight to assign and order to a picking Order.
    - Assing to the picking Order Selected as Open if doesn't find one, give a message to select one.
      If the order was assigned to a picking order modify it status to assigned to Assigned to a Picking Order
    - Up date the adapter and de totals.
- Create an expresso test for this activity and verify woks properly.

Task 5: Implement Picking Fragment

- Build UI for the picking Fragment and the layout
- Implement the loaders and adapters to show the picking orders not Deliver or deleted.
- Implement onClick to Open the PickingFragmeDetail.

Task 6: Implement Picking Detail Fragment

- Build UI for the picking Fragment and the layout
- Implement the loaders and adapters to show the Orders Assigned and totals.

- Implement the querys to show the totals.
- Build the onClick operation to modify a Custom Order. Reuse the Custom Order Activity.
- Build onSlidetoTheleft to release and order from the picking Order.
    - Assign Status to Initial
    - Update the adapter and de totals.
- Create an expresso test for this activity and verify woks properly.

Task 6: Implement Delivey Fragment

- Build UI for the delivery Fragment and the layout
- Implement the loaders and adapters to show the Orders Assigned and totals.

- Implement the querys to show the totals.
- Build the onClick operation to modify a Custom Order. Reuse the Custom Order Activity.
- Build onSlidetoTheleft to release and order from the picking Order.
    - Assign Status to Initial
    - Update the adapter and de totals.
- Build onSlidetoTheRigth on an custom order to open Close Activity

- Create an expresso test for this activity and verify woks properly.

Task 7: Implement Close Activity

- Build UI for the Close Activity and the layout
- Implement the confirmation Button to change the status to Delivered
- Create an expresso test for this activity and verify woks properly.

Task 8: Implement Google Play Services

- Implement and Interstitial add after the creation of a Custom Order
- Implement the adds in the bottom of each principal screen
- Create an expresso test for this activity and verify woks properly.

Task 9: Create Build Variant

- Implement a variant without adds.
- Create an expresso test for this activity and verify woks properly.

Task 10: Create widget

- Implement a widget to show the products quantities in custom orders, load and balance in the delivery process..

Task 11: Final Testing

- Run all the tests and verify the app works ok.