# Bring-It Project Aim

Bring**-**It project, aims to simplify the provision of alcoholic drinks, to any event who require it, being small like it would be a family meeting, or a medium-large event like it would be a party, or official event of “X” company.

# Project Description

BringIt is an e-commerce app, that deliver to the User a quick provision of alcoholic drinks, it is designed to satisfy the customer with the drinks of choice, having plenty of choices.

# Functional and non-functional requirements

### Functional requirements

* The system allows users to register if they wish to create an account.
* Users can sign in of they already have an existing account.
* The system updates the database in real time with the data entered by users.
* The application includes the following pages: Start, Home, Catalog, Product, Cart, Checkout, Profile and Notifications pages.
* The system performs full CRUD operations.
* The system stores all data in a Firebase database

### non-functional requirements

* The UI (user Interface) is user friendly and easy to navigate
* The system provides real time performance when updating or retrieving data.
* The app ensures data reliability and availability through Firebase.

# User stories

As a family meeting host, I want to order alcohol for the family gathering while I prepare the house for my guests, so that the drinks are delivered to the house and I don’t waste time and gasoline going to the supermarket or my next **SAQ**.

### Accepted criteria

* The User can order different types of drinks.
* The system confirms delivered address and estimated time
* The system sends a notification to the User.

# Test cases

Test 1

Description: Verify that a registered user can log in successfully.

Steps made:

* Open the app.
* Enter valid email and password.
* Click “Sign In.”

**Expected results:** The user is redirected to the home screen and sees a welcome message.

**Result:** Success.

Test 2

Description: Verify that the system rejects login with an incorrect password.

Steps made:

* Open the app.
* Enter valid email and incorrect password.
* Click “Sign In.”

**Expected results:** The system displays an error message in a Toast and stays on the login page.

**Result:** Success

Test 3

Description: Verify that the user can add a drink to the cart.

Steps made:

* Open the catalog.
* Select a drink.
* Click “Add to cart.”

**Expected results:** The selected drink appears in the cart with correct quantity and price.

**Result:** Success.

**Test 4**  
**Description:** Verify that payment using Stripe is processed successfully.  
**Steps made:**

* Add drinks to the cart.
* Proceed to checkout.
* Select “Pay with Stripe.”
* Enter valid card details.
* Click “Pay now.”

**Expected results:** Payment is accepted, and the system displays an order confirmation.  
**Result:** Success.

Test 5

Description: User Registration verification by sending an e-mail to the respected e-mail address input

**Steps made**:

* Click register on the login page
* Input the respected e-mail address and its password (and confirmation of the same)
* Verify e-mail address on google g-mail.

**Expected results:** The app sends an e-mail to g-mail, and, by clicking in the verification link send to the e-mail (most likely in spam) the user can now log in without a problem.

**Result:** Success.

# Individual’s roles and responsibilities

José Ailton Jiménez Mota: DB management, Login page logic, Cart page logic, Products pages logic, Register page logic, Home page logic, Menu page Logic.

Francisco Guzman: Frontend, Ui design, Bug corrections, Flow of the app, Documentation, Presentation.

Lucas Meira Lobato: API callings, notification management, Payment page logic.