

FIFI FLOWER SHOP

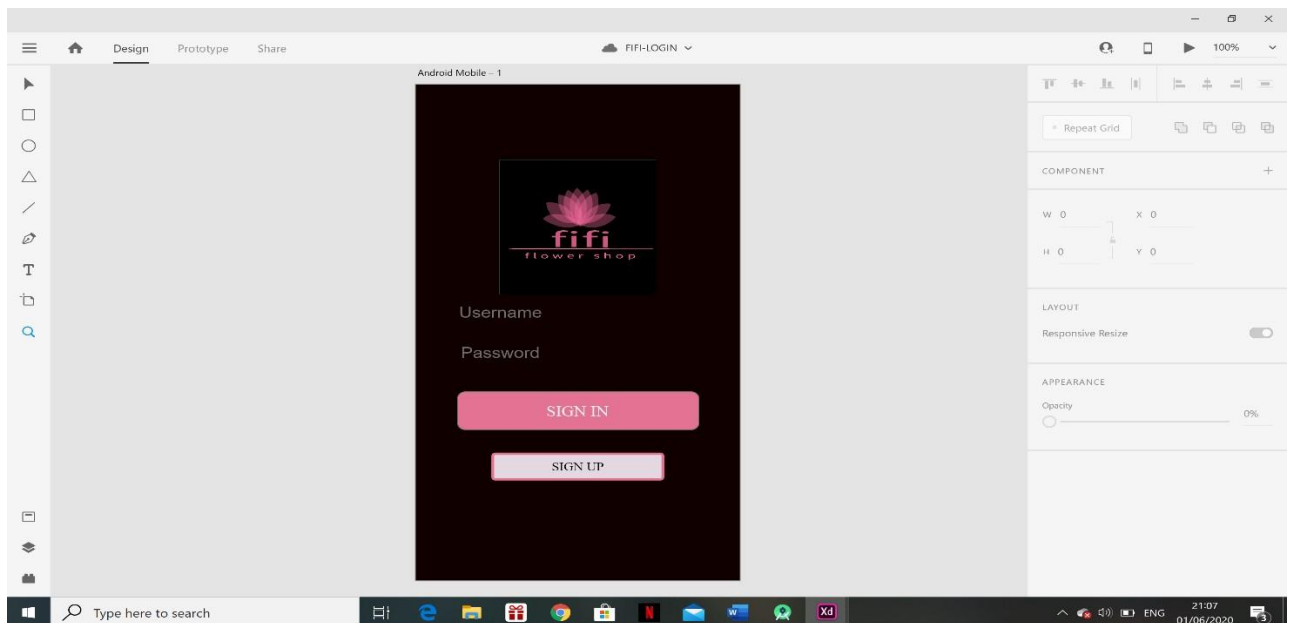
DESCRIEREA PROIECTULUI

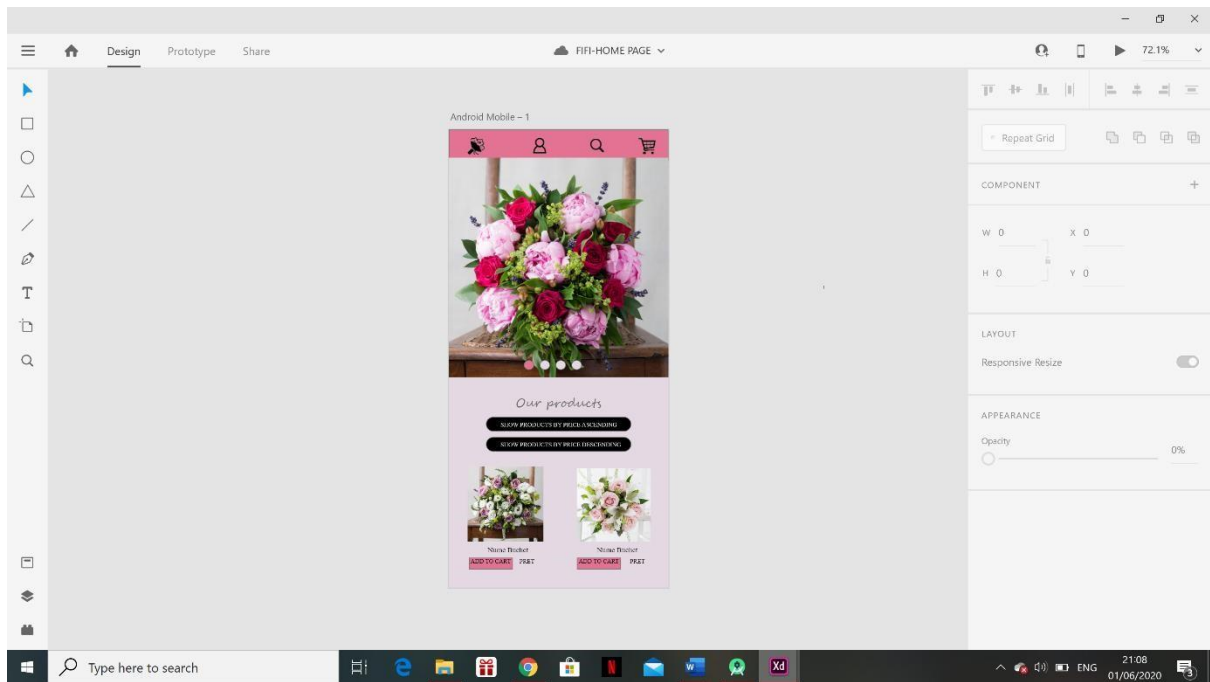
Ideea proiectului este realizarea unei aplicatii destinata dispozitivelor Android pentru o florarie digitala. Proiectul a fost realizat cu ajutorul limbajului Kotlin integrat in Android Studio.



DESIGN

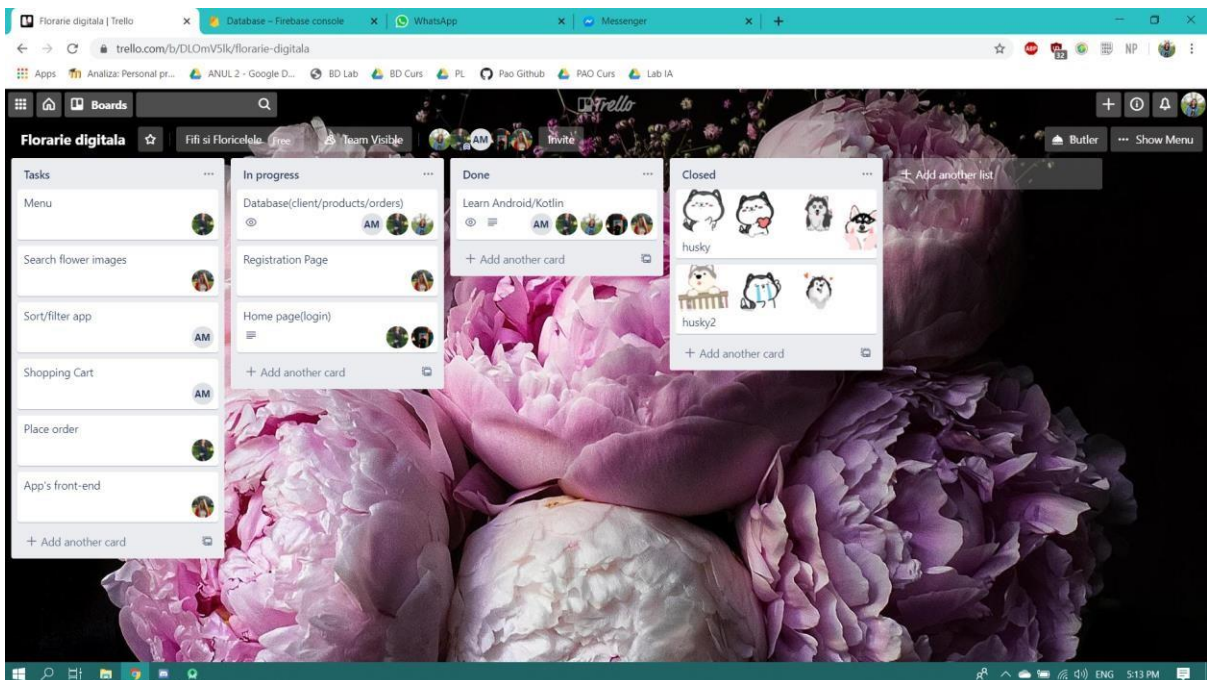
Am ales sa facem schita designului mai intai in Adobe XD.

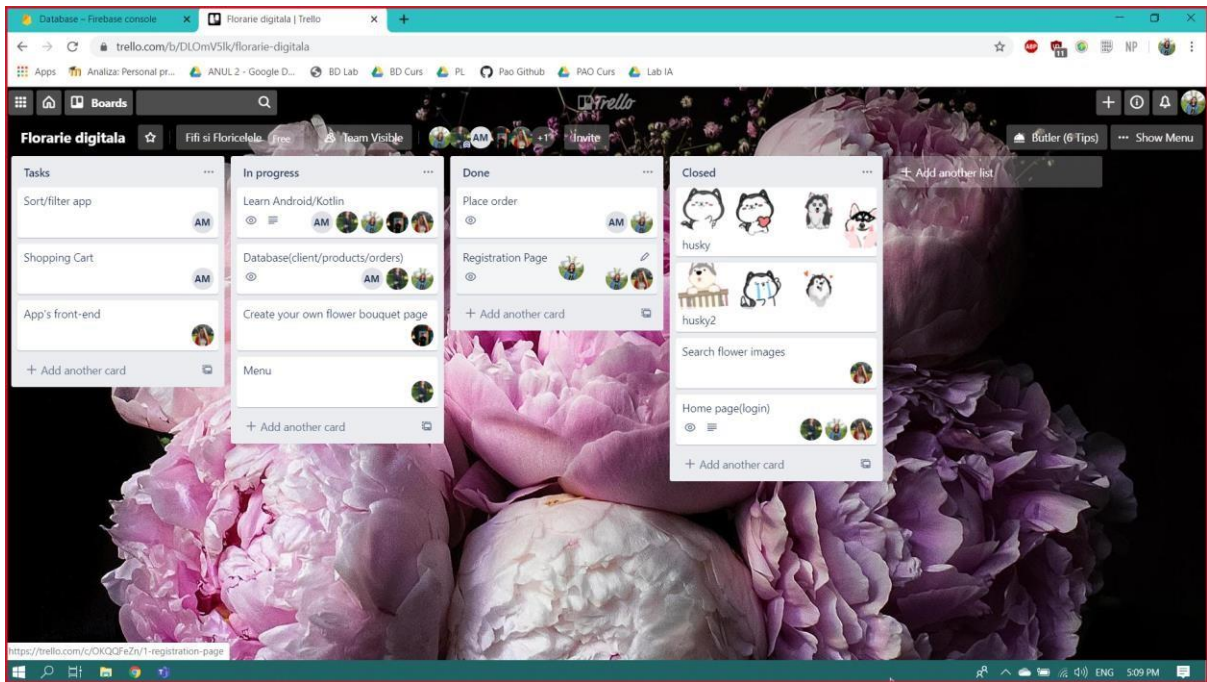
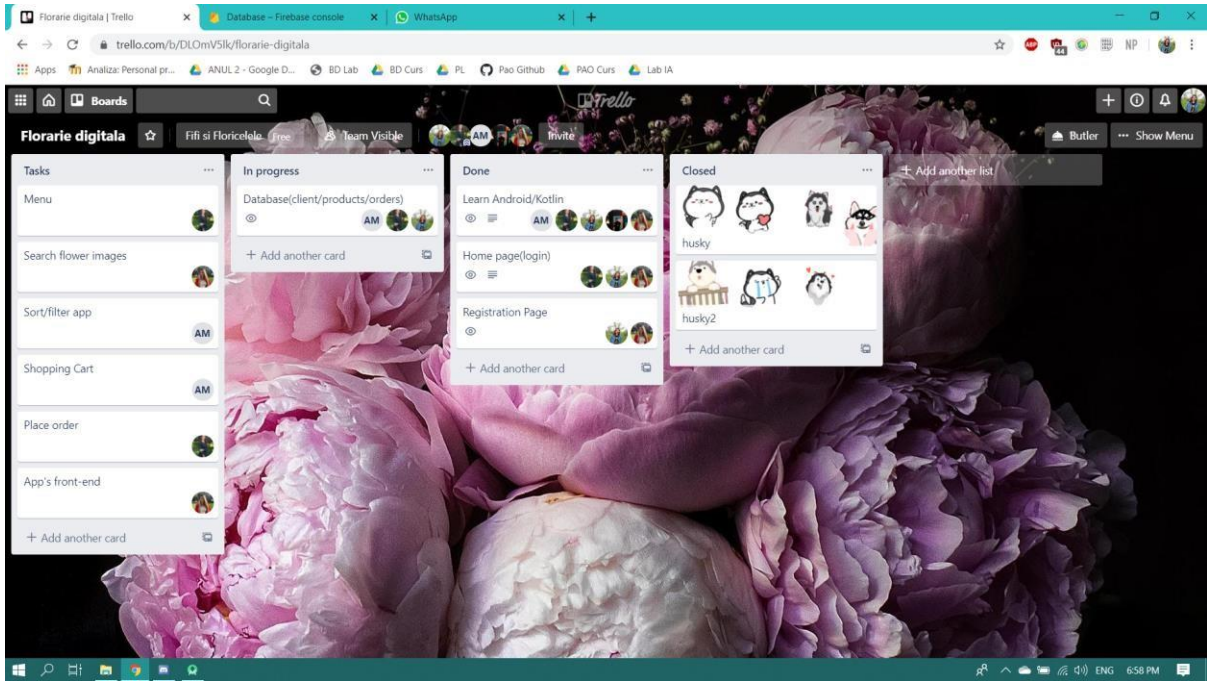




IMPARTIREA TASK-URILOR

Pentru colaborarea eficienta in echipa, impartirea si evidenta task-urilor am folosit Trello. Pe parcursul realizarii proiectului, am efectuat cateva screenshoturi cu evolutia sa.





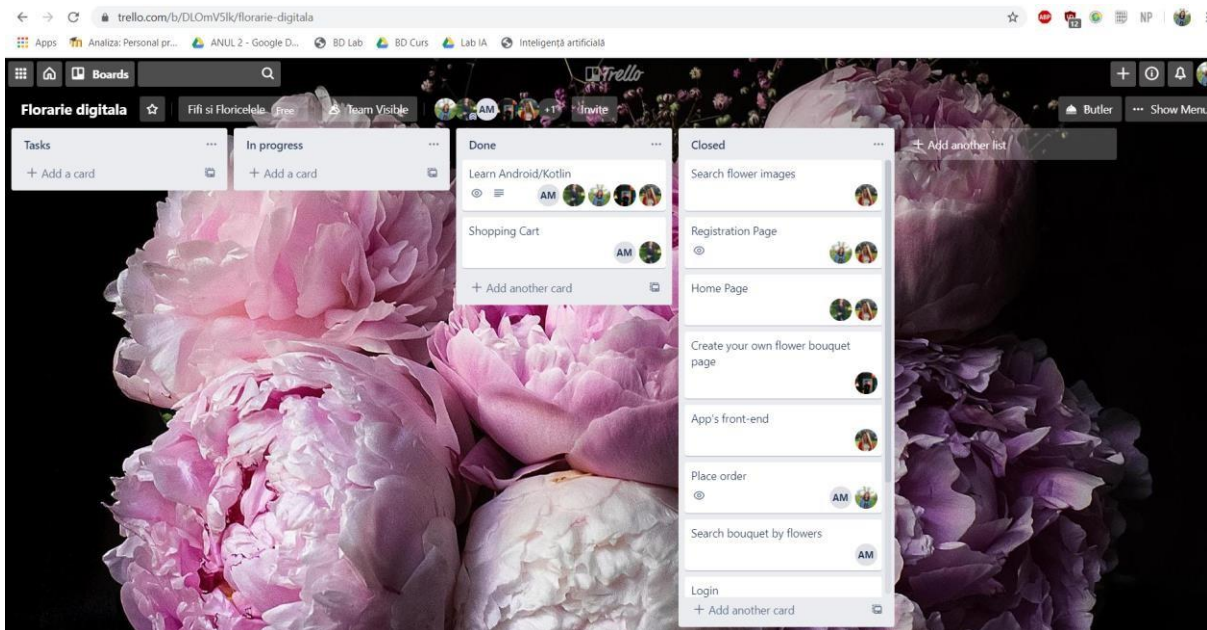
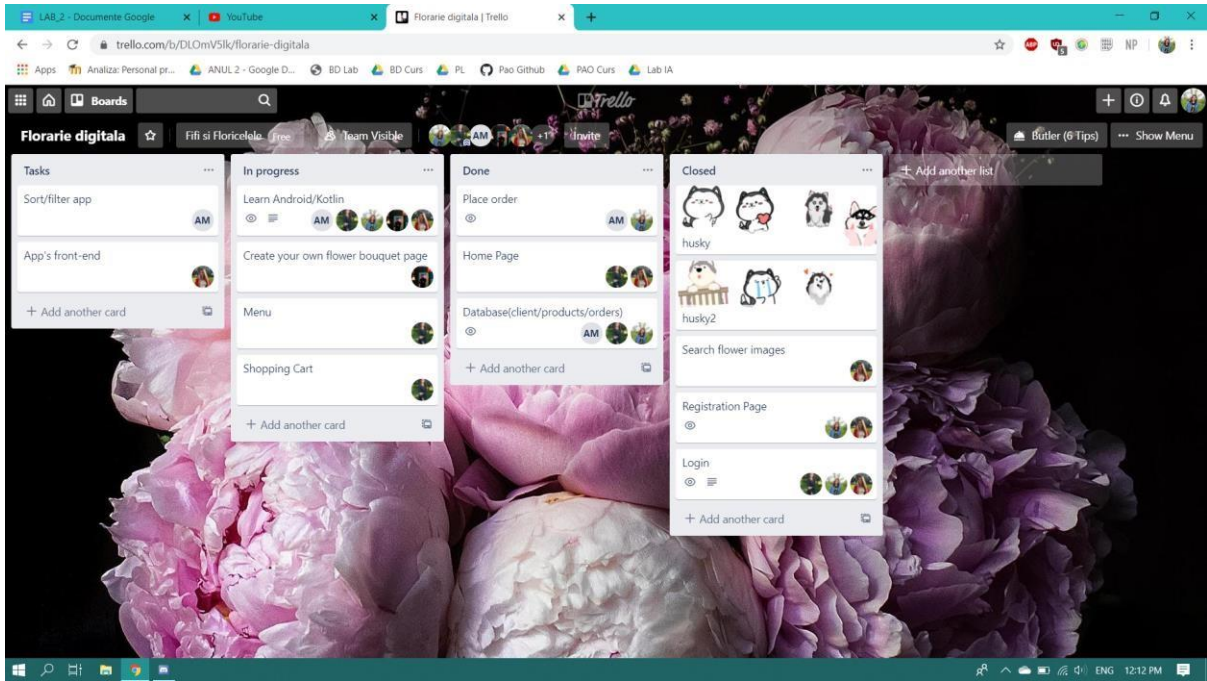
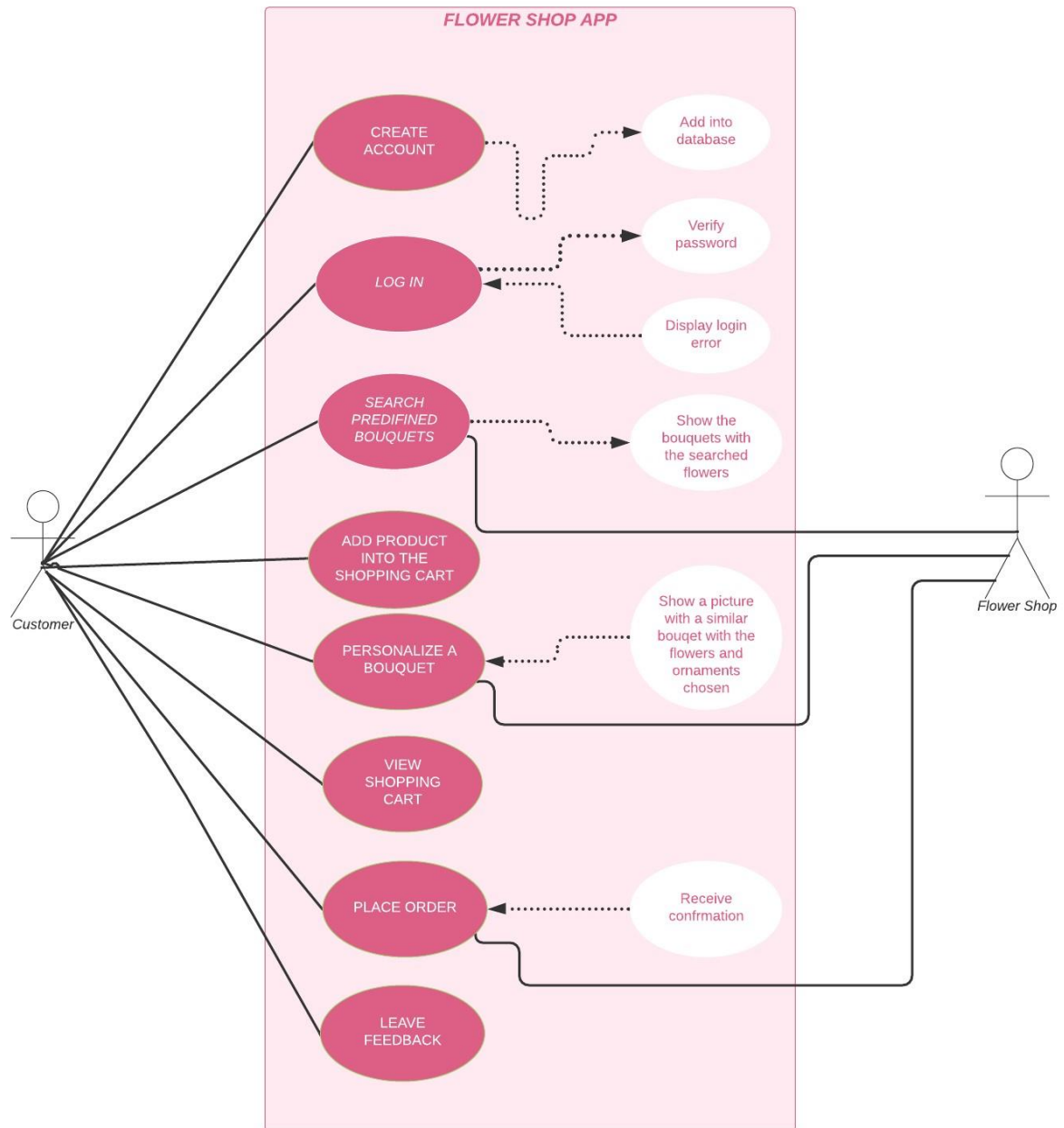


DIAGRAMA UML – Use Case



CODE STANDARDS

Inainte de a incepe proiectul, ne-am setat niste standarde ale codului pe care toti a trebuit sa le respectam astfel incat tot codul proiectului sa respecte aceleasi reguli.

	E.g.	Correct	Wrong	Comments
Esthetics	Variable and function names should be camelCase:	camelCase	myCombined_Styles();	
			PascalCaseStyle();	
			snake_case_style();	
			function camelCase(); int snake_case	
	#define should be in CAPITAL LETTERS using SNAKE_CASE (attention, only for define! Not for const int as well):	#define LED_PIN 13	led_pin 13 etc	We encourage using const int though.
			LEDPIN 13	
			ledpin 13	
			ledPin 13	
	Use const int for immutable variables (such as pins):	const int ledPin = 13;	int ledPin = 13;	
	Write const int with camel case, not like define:	const int ledPin = 13;	const int LED_PIN = 13;	
	Correct spacing:	blinkInterval = 20;	blinkInterval=20;	
			blinkInterval= 20;	
			blinkInterval =20;	
		blinkInterval = (ledDelay + buzzerDelay);	blinkInterval=(ledDelay+buzzerDelay);	
			blinkInterval=(ledDelay+buzzerDelay);	
			blinkInterval =(ledDelay+buzzerDelay);	
			blinkInterval = (ledDelay+buzzerDelay);	
		if (xAxisValue + yAxisValue) {	if(xAxisValue + yAxisValue) {	
			if (xAxisValue + yAxisValue) {	
			if (xAxisValue+yAxisValue) {	

		for (int i; i < n; i++) {	for(int i; i < n; i++) {	
--	--	---------------------------	--------------------------	--

			for (int i; i < n; i++) {	
			for (int i; i < n; i++) {	
		while (gameState == "ON") { doSomething();	while(gameState == "ON") {	
			while (gameState== "ON") { doSomethnig();	
			while (gameState== "ON") {	
			while (gameState == "ON"){	
	Suggestive variable names:	int displayLedPin = 9;	pin13; pinLed; myPin; unPin; x; y; var1; var2	
		Use only English	Questa e una lingua amestecata cu romana and english from time to time (attention! Variables imported from libraries are OK to be left with their default names)	
Functionality	Modular code:	Loop should only call a few functions.	all the functionality is in the loop() function instead of breaking it into smaller functions	
			all the functionality is in only 1 function that gets called in the loop() function	
	Modular functionality:	Write functions that are modular and general. For example, in a game you should have a level function that receives parameters, including the level number and generate the specified level.	Hardcoded levels (level1(); level2();)	

	Global blockage:	<pre> loop() { if (buttonValue == HIGH && !gameStarted) { startGame(); // gameStarted = true; + other functionality } if (gameStarted) { runGame(); } } </pre>	<pre> loop() { while (buttonValue != HIGH && !gameStarted) { startGame(); // gameStarted = true; + other functionality } runGame(); } </pre>	Use if's instead of while. A while() in the for is the same as a global block. You already have a infinite loop(). Use variables to direct the workflow instead of blocking them in another local loop.
		Use millis() instead of delay()	Using delay() and having a global block.	
	Don't add unnecessary code:	<pre> int potValue = analogRead(potPin); </pre>	<pre> int myPot = analogRead(potPin); int potValue = myPot; </pre>	Careful though. Do not confuse this with adding a new variable for a different component. Something like ledValue = buttonValue; is OK, because you might want to manipulate one of them later without affecting the other.
	Code comments:	/* the levels are generated	No comments at all or too many comments.	

		procedurally, based on control variables such as level number, score and number of lives left. The last 2 variables are global. */ void setLevel(int levelNo) { // do something }	// sets color of the RGB LED int setRGBColor(red, green, blue) {	Simple and intuitive commenting, to understand the logic. Each function and variable name should be self explanatory. Rule of thumb: less comments than code.
	Understanding:	Clear understanding and knowledge of the written code.	Copied code from the internet or some colleague with no clear understanding of it's functionality.	It's ok to get inspired off the internet, but write the code yourself and make sure you understand every line.

USER STORIES

De asemenea, am conceput user stories care contin toate functionalitatile aplicatiei:

SIGN UP

As a user, I can create an account by completing a form.

As a user, I can choose my own username and password

ACCEPTANCE CRITERIA: The user should complete every text box from the form.

The user should choose a password with minimum 4 characters.

The user should introduce only valid data.

The user should agree that their data will be added into the app's database.

SIGN IN

As a user, I can login anytime.

As a user, I can reset my password

As a user, I can stay logged in until I close the app

As a user, after login I will be redirected to the Home Page

SIGN OUT

As a user, I can logout anytime.

LEAVE FEEDBACK

As a user I can write a feedback.

ACCETANCE CRITERIA: The user has to be logged in.

SEARCH

As a user, I can use the search bar and type the flowers I want to see into the predefined bouquets.
The app will shop them with colored text.

CHOOSE ONE PREDEFINED BOUQUET

As a user, I can look through the predefined bouquets and choose one, adding it to the cart.

VIEW MY SHOOPING CART

As a user, I can see and change the number of the products that I've added into the cart.

As a user, I can see the total price of the products that are in my shopping cart.

PLACE ORDER

As a user, I can make an order with the products that I have in the shopping cart by completing a form.

As a user, I can choose the date and the address of the delivery.

As a user, I can add a message to the receiver.

As a user, after placing an order I will receive a message with the confirmation.

ACCEPTANCE CRITERIA: The user should complete every text box from the form.

The user should introduce only valid data.

ADD A PRODUCT INTO THE SHOPPING CART

As a user, I can choose any bouquet and add it only once into my shopping cart by pressing the 'ADD TO CART' button.

NATIVAGATION MENU

As a user, I can switch the page I'm viewing with the Navigation Menu that is located into the header of the app.

PERSONALIZE YOUR BOUQUETS

As a user, I can choose the flowers (and how many of each kind) that I want in my bouquet, the ornaments and the wrap and I will see a picture of a similar bouquet.

As a user, I still can place an order with a combination that doesn't exist in the app.

PERSONAS

Acestea reprezinta cateva tipuri de persoane pentru care aplicatia noastra ar fi ideala:

Albert

Age 25-35

-Hectic schedule

-In a rush, wants to finish this task quickly/fast

Advantages:

-card payment method

-predefined bouquets

-online order

-selection of delivery date

Ricardo

-Le boss

-has a lot of money

-wants to surprise all his employers(females)

-buys in huge quantities

Advantages:

- doesn't have to worry about transport
- card payment
- diverse and available in large quantities stock

Andrew

Simple man

- caring for his wife
- knows her favorite flowers
- he likes the comfort of online shopping

Advantages:

- is able to customize his bouquet
- multiple payment methods
- card payment

Mary

- likes to always have flower in her home
- online shopaholic
- makes cute and thoughtful gifts for her girl friends

Advantages:

- reasonable prices
- unique/cute bouquets
- online shopping/card payment