PizzaT-Good

By
Ikemefuna Chukwunyerenwa

Introduction

PizzaT-Good (Pizza Taste Good), a pizza ordering app is an application, which helps restaurants may be large or small to optimize and have complete control over all their business and customers. Day to day, these applications are grabbing the market like anything. This application can help restaurants to do all functionalities more exactly and faster. It reduces work and improves the efficiency of restaurants. This application is helping pizza orderings to maintain the stock and cash flows. The software helps pizza orders to maintain day to day records in the system.

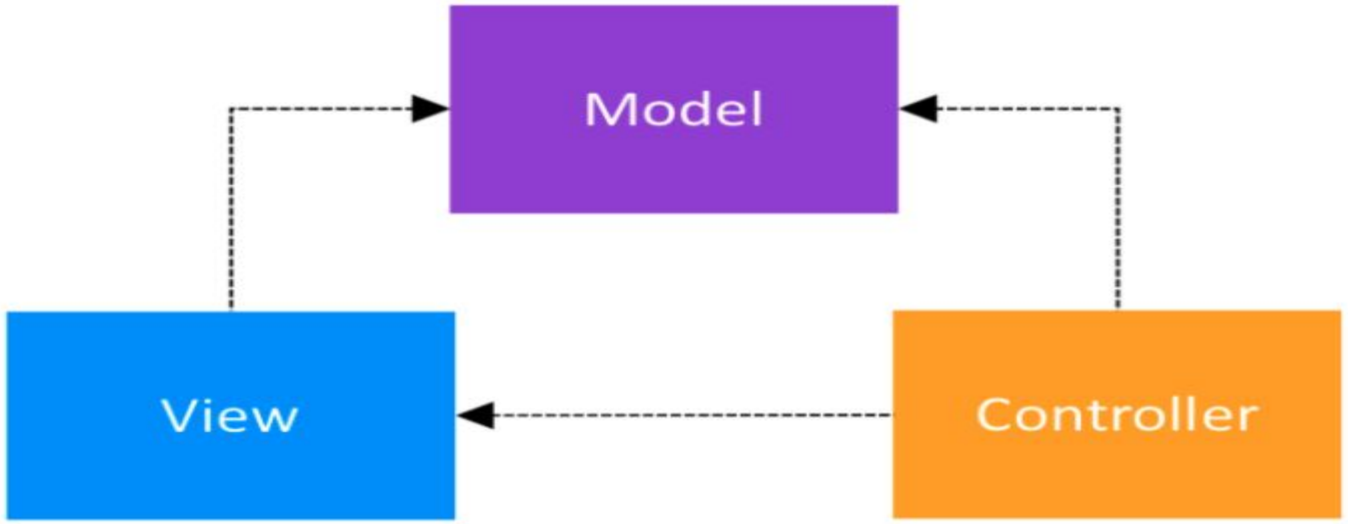
Model, View and Controller (MVC)

The **model** is the component that is in charge of the management of data and logic of an MVC application. As the model is the principal manager of all data and business logic, you can view it as the powerhouse of an MVC application.

The **view** is a visual representation of data that exists in and is generated by application. It is the primary point of interaction that a user has with application.

The **controller** is accepts input and converts it to commands for the model or view

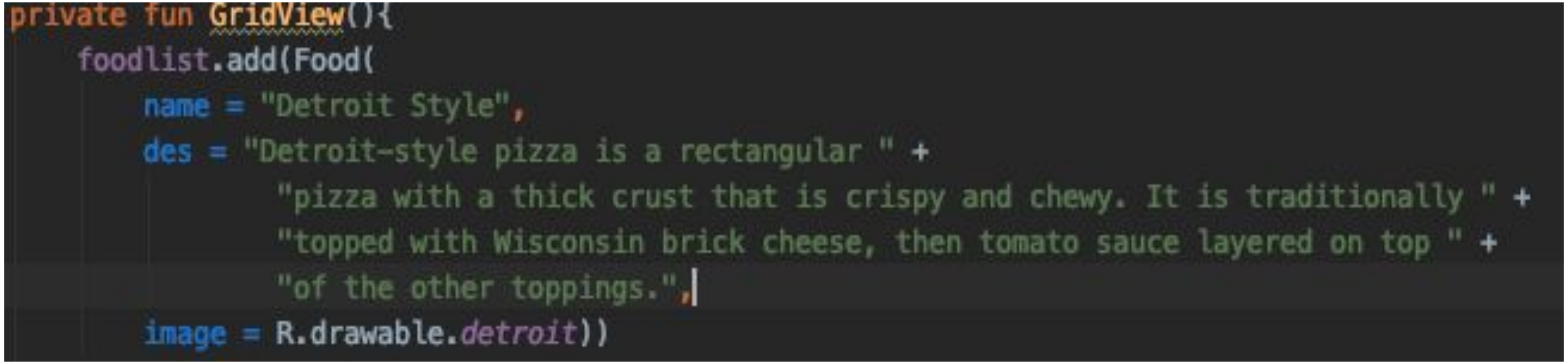
MVC Diagram



Model-View-Controller class structure

Model Code

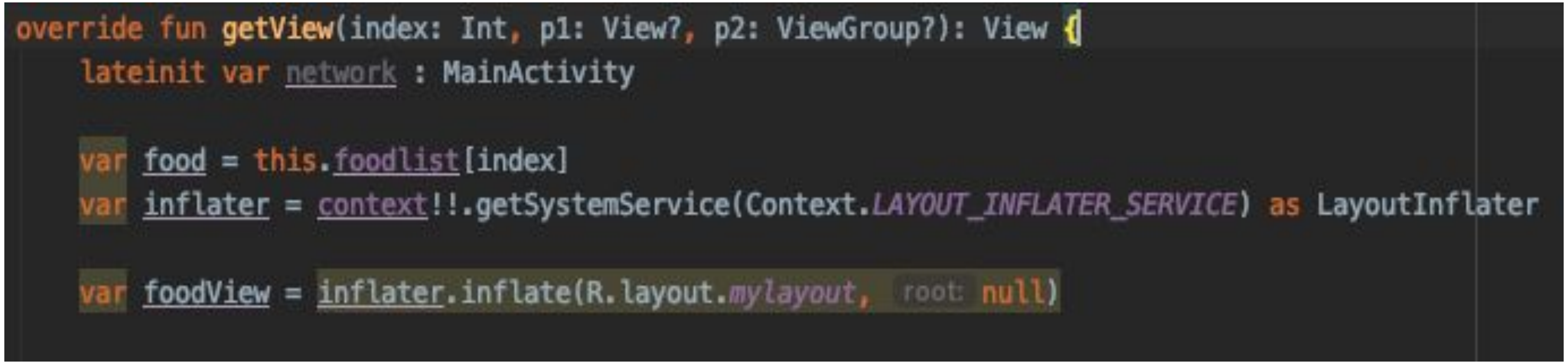
foodlist from Food class



```
private fun GridView(){
    foodlist.add(Food(
        name = "Detroit Style",
        des = "Detroit-style pizza is a rectangular " +
                "pizza with a thick crust that is crispy and chewy. It is traditionally " +
                "topped with Wisconsin brick cheese, then tomato sauce layered on top " +
                "of the other toppings.",
        image = R.drawable.detroit))
```

View Code

Read the mylayout.



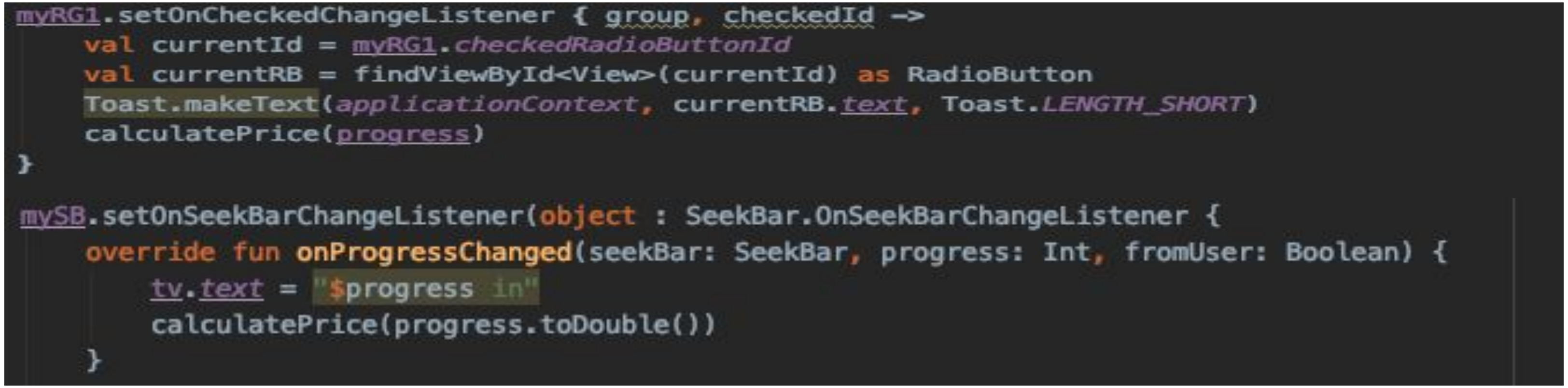
```
override fun getView(index: Int, p1: View?, p2: ViewGroup?): View {
    lateinit var network : MainActivity

    var food = this.foodlist[index]
    var inflater = context!!.getSystemService(Context.LAYOUT_INFLATER_SERVICE) as LayoutInflater

    var foodView = inflater.inflate(R.layout.mylayout, root: null)
```

Controller Code

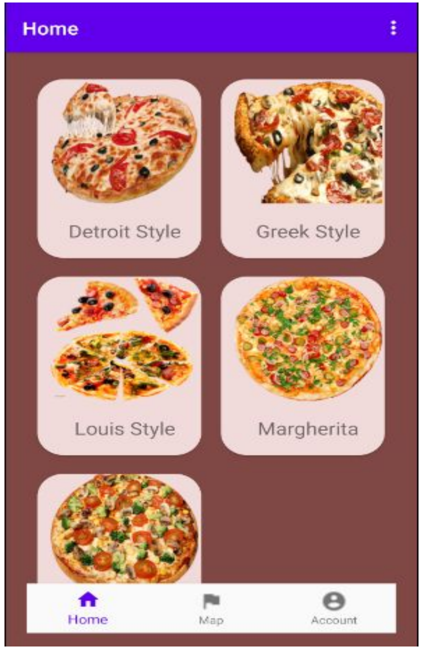
Event listener represent the interfaces responsible to handle **events**

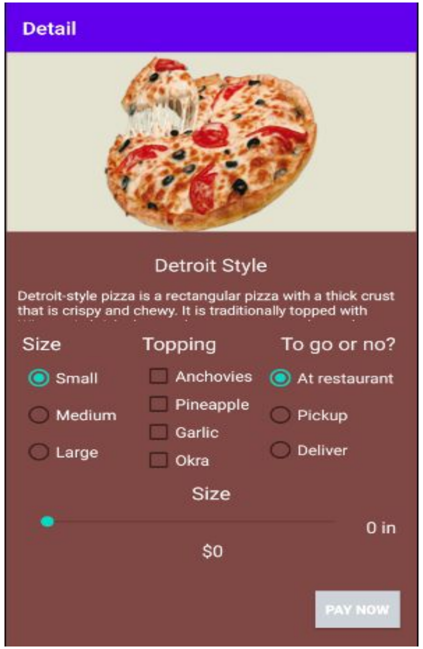


```
myRG1.setOnCheckedChangeListener { group, checkedId ->
    val currentId = myRG1.checkedRadioButtonId
    val currentRB = findViewById<View>(currentId) as RadioButton
    Toast.makeText(applicationContext, currentRB.text, Toast.LENGTH_SHORT)
    calculatePrice(progress)
}

mySB.setOnSeekBarChangeListener(object : SeekBar.OnSeekBarChangeListener {
    override fun onProgressChanged(seekBar: SeekBar, progress: Int, fromUser: Boolean) {
        tv.text = "$progress in"
        calculatePrice(progress.toDouble())
    }
```

GRIDVIEW, DETAILVIEW and CheckoutView





Detroit Style

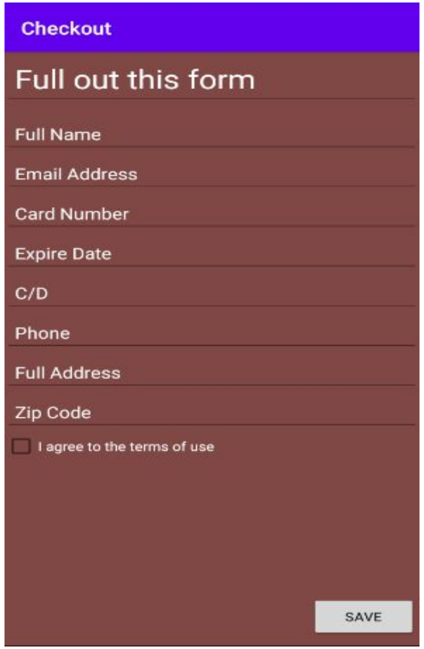
Detroit-style pizza is a rectangular pizza with a thick crust that is crispy and chewy. It is traditionally topped with

Size	Topping	To go or no?
[✓ Small]	☐ Anchovies	[✓ At restaurant]
Medium	☐ Pineapple	Pickup
Large	☐ Garlic	Deliver
	☐ Okra	

Size

0 in

$0



Full out this form

Full Name

Email Address

Card Number

Expire Date

C/D

Phone

Full Address

Zip Code

☐ I agree to the terms of use

Challenge I faced

- I was struggling that the fragment did not show the view data on the MainActivity.
- The issue with listener events when I clicked the button that often crashes.
- I was struggling about how I could allow multiple fragments to show on MainActivity but somehow they get crash often.
- There was working for the map but it did not show the location where the user stood at.
- I struggled with how I could understand the measurement of UI for layout.
- With the COVID-19 situation, there was a limited time for me to tutors for helping me with my assignments.

Resources

- GridView: https://grokonez.com/android/kotlin-gridview-example-show-list-of-items-on-grid-android
- Pizza codes: https://www.youtube.com/watch?v=IBE_719AqKs
- Pizza codes: https://stackoverflow.com/questions/22405582/android-checkbox-enable-and-disable
- https://www.androhub.com/android-radiobutton/
- Listener events: https://stackoverflow.com/questions/25102156/radio-group-setoncheckedchangelistener https://antonioleiva.com/listeners-several-functions-kotlin/