

**Московский государственный технический
университет им. Н.Э. Баумана**

Факультет «Радиотехнический»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Разработка интернет-приложений»

Отчет по домашнему заданию
«Разработка мобильного приложения»

Выполнил:
студент группы РТ5-51Б
Борисочкин М.И.
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю. Е.
Подпись и дата:

Москва, 2022 г.

Описание задания

На основе методических указаний разработайте мобильное приложение на языке Kotlin (платформа Android) или языке Swift (платформа iOS).

Для создания приложения необходимо решить следующие задачи:

1. Создать стартовый проект мобильного приложения.
2. Добавить компонент таблицы в Ваше приложение.
3. Добавить подключение из мобильного приложения к веб-сервису, разработанному в лабораторной работе №6.
4. Наполнить таблицу элементами, полученными из вашего веб-сервиса: с помощью запросов к API получить из него данные.
5. Разработать дополнительный экран с UI компонентами для отображения детальной информации об элементе таблицы при касании по нему.

Расширенное задание, добавляющее +1 балл на экзамене:

1. Реализовать возможности редактирования, добавления и удаления данных таблицы в мобильном приложении с помощью запросов к веб-сервису.

Текст программы

com/example/pizzaapp/network/Pizza.kt

```
package com.example.pizzaapp.network

data class Pizza( // Data-класс "Пицца"
    val pk: Int?,
    val name: String,
    val topping: String,
    val diameter: Int
)
```

com/example/pizzaapp/network/PizzaApi.kt

```
package com.example.pizzaapp.network

import retrofit2.Call
import retrofit2.Response
import retrofit2.http.*

interface PizzaApi { // Интерфейс для реализации HTTP-методов

    @GET("/pizza/") // GET-запрос для получения списка всех пицц
    suspend fun getPizzas() : Response<List<Pizza>>

    @FormUrlEncoded
    @POST("/pizza/") // POST-запрос для добавления пиццы
    fun createPizza(@Field("name") name: String,
        @Field("topping") topping: String,
        @Field("diameter") diameter: Int): Call<Pizza>

    @PUT("/pizza/{pk}/") // PUT-запрос для изменения пиццы
    fun updatePizza(@Path("pk") pk: Int, @Body pizza: Pizza): Call<Pizza>

    @DELETE("/pizza/{pk}/") // DELETE-запрос для удаления пиццы
    fun deletePizza(@Path("pk") pk: Int): Call<Pizza>
}
```

com/example/pizzaapp/network/RetrofitInstance.kt

```
package com.example.pizzaapp.network

import retrofit2.Retrofit
import retrofit2.converter.gson.GsonConverterFactory

object RetrofitInstance { // Экземпляр объекта Retrofit

    val api: PizzaApi by lazy {

        Retrofit.Builder() // Построение объекта класса Retrofit
            .baseUrl("http://10.0.2.2:8000")
            .addConverterFactory(GsonConverterFactory.create())
            .build()
            .create(PizzaApi::class.java)

    }

}
```

com/example/pizzaapp/network/PizzaAdapter.kt

```
package com.example.pizzaapp.network

import android.view.LayoutInflater
import android.view.ViewGroup
import androidx.recyclerview.widget.AsyncListDiffer
import androidx.recyclerview.widget.DiffUtil
import androidx.recyclerview.widget.RecyclerView
import com.example.pizzaapp.databinding.ItemViewBinding

class PizzaAdapter: RecyclerView.Adapter<PizzaAdapter.PizzaViewHolder>() { //
Адаптер для RecyclerView

    private lateinit var mListener : onItemClickListener

    interface onItemClickListener{ // Интерфейс обработки события нажатия на
ViewHolder
        fun onItemClick(position: Int)

    }

    fun setOnItemClickListener(listener: onItemClickListener) { // Подписка
на обработку события нажатия кнопки
        mListener = listener
    }

    inner class PizzaViewHolder(val binding: ItemViewBinding, listener:
onItemClickListener) : RecyclerView.ViewHolder(binding.root) { // ViewHolder
для пиццы
        init {
            itemView.setOnClickListener{
                listener.onItemClick(adapterPosition)
            }
        }
    }

}
```

```

        private val diffCallback = object : DiffUtil.ItemCallback<Pizza>() {
            override fun areItemsTheSame(oldItem: Pizza, newItem: Pizza): Boolean {
                // Реализация метода areItemsTheSame
                return oldItem.pk == newItem.pk
            }

            override fun areContentsTheSame(oldItem: Pizza, newItem: Pizza): Boolean {
                // Реализация метода areContentsTheSame
                return oldItem == newItem
            }
        }

        private val differ = AsyncListDiffer(this, diffCallback)
        var pizzas: List<Pizza>
        get() = differ.currentList
        set(value) { differ.submitList(value) }

        override fun getItemCount(): Int = pizzas.size // Реализация метода getItemCount

        override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): PizzaViewHolder {
            // Реализация метода onCreateViewHolder
            return PizzaViewHolder(ItemViewBinding.inflate(LayoutInflater.from(parent.context), parent, false), mListener)
        }

        override fun onBindViewHolder(holder: PizzaViewHolder, position: Int) {
            // Реализация метода onBindViewHolder
            holder.binding.apply {
                val pizza = pizzas[position]
                itemText.text = pizza.name
            }
        }
    }
}

```

com/example/pizzaapp/MainActivity.kt

```
package com.example.pizzaapp

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.util.Log
import androidx.core.view.isVisible
import androidx.lifecycle.LifecycleScope
import androidx.recyclerview.widget.LinearLayoutManager
import com.example.pizzaapp.databinding.ActivityMainBinding
import com.example.pizzaapp.network.Pizza
import com.example.pizzaapp.network.PizzaAdapter
import com.example.pizzaapp.network.RetrofitInstance
import retrofit2.HttpException
import retrofit2.Response
import java.io.IOException

const val TAG = "MainActivity"

class MainActivity : AppCompatActivity() {

    private lateinit var binding: ActivityMainBinding

    private lateinit var pizzaAdapter: PizzaAdapter

    lateinit var response : Response<List<Pizza>>

    override fun onCreate(savedInstanceState: Bundle?) { // Переопределение
        метода onCreate
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)
    }

    override fun onStart() { // Переопределение метода onStart
        super.onStart()
        setupRecyclerView() // Установка RecyclerView

        // Корутины для загрузки данных из API
        lifecycleScope.launchWhenStarted {
            binding.progressBar.isVisible = true
            response = try {
                RetrofitInstance.api.getPizzas()
            } catch (e: IOException) { // Обработка ошибки ввода-вывода
                Log.e(TAG, "IOException, you might not to have internet
connection")

                binding.progressBar.isVisible = false
                return@launchWhenStarted
            } catch (e: HttpException) { // Обработка ошибки HTTP
                Log.e(TAG, "HttpException, unexpected response")
                binding.progressBar.isVisible = false
                return@launchWhenStarted
            }

            // Проверка успешного ответа
            if(response.isSuccessful && response.body() != null) {
                pizzaAdapter.pizzas = response.body()!!
            } else {
                Log.e(TAG, "Response not successful")
            }
        }
    }
}
```

```

        binding.progressBar.isVisible = false
    }

    // Обработка события нажатия кнопки для добавления записи
    binding.btnCreatePizza.setOnClickListener {
        val intent = Intent(this@MainActivity,
CreatePizzaActivity::class.java)
        startActivity(intent)
    }
}

// Функция для установки RecyclerView
private fun setupRecyclerView() = binding.rvPizzas.apply {
    pizzaAdapter = PizzaAdapter()
    adapter = pizzaAdapter
    layoutManager = LinearLayoutManager(this@MainActivity)

    pizzaAdapter.setOnItemClickListener(object
:PizzaAdapter.OnItemClickListener {
        override fun onItemClick(position: Int) { // Обработка нажатия на
ViewHolder

            val intent = Intent(this@MainActivity,
DetailActivity::class.java)
            intent.putExtra("pk", response.body()!![position].pk)
            intent.putExtra("name", response.body()!![position].name)
            intent.putExtra("topping",
response.body()!![position].topping)
            intent.putExtra("diameter",
response.body()!![position].diameter)
            startActivity(intent)
        }
    })
}
}

```

com/example/pizzaapp/DetailActivity.kt

```
package com.example.pizzaapp

import android.annotation.SuppressLint
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Toast
import com.example.pizzaapp.databinding.ActivityDetailBinding
import com.example.pizzaapp.network.Pizza
import com.example.pizzaapp.network.RetrofitInstance
import retrofit2.Call
import retrofit2.Callback
import retrofit2.Response

class DetailActivity : AppCompatActivity() {

    private lateinit var binding: ActivityDetailBinding

    @SuppressLint("SetTextI18n")
    override fun onCreate(savedInstanceState: Bundle?) { // Переопределение
        метода onCreate
        super.onCreate(savedInstanceState)
        binding = ActivityDetailBinding.inflate(layoutInflater)
        setContentView(binding.root)

        // Получение данных из главной activity
        val bundle: Bundle? = intent.extras
        val pizzaName = bundle!!.getString("name")
        val pizzaTopping = bundle.getString("topping")
        val pizzaDiameter = bundle.getInt("diameter")
        val pizzaPK = bundle.getInt("pk")

        binding.tvPizzaName.text = pizzaName
        binding.tvPizzaTopping.text = "Состав: $pizzaTopping"
        binding.tvPizzaDiameter.text = "Диаметр: $pizzaDiameter см"

        // Обработка события нажатия кнопки для изменения записи
        binding.btnChange.setOnClickListener {
            val intent = Intent(this, UpdateActivity::class.java)
            intent.putExtra("pk", pizzaPK)
            intent.putExtra("name", pizzaName)
            intent.putExtra("topping", pizzaTopping)
            intent.putExtra("diameter", pizzaDiameter)

            startActivity(intent)
        }

        // Обработка события нажатия кнопки для удаления записи
        binding.btnDelete.setOnClickListener {
            RetrofitInstance.api.deletePizza(pizzaPK).enqueue(object :
            Callback<Pizza> {
                override fun onResponse(call: Call<Pizza>, response:
                Response<Pizza>) {
                    Toast.makeText(applicationContext, "Удалено",
                    Toast.LENGTH_SHORT).show()
                }

                override fun onFailure(call: Call<Pizza>, t: Throwable) {
                    Toast.makeText(applicationContext, "Ошибка",
                    Toast.LENGTH_SHORT).show()
                }
            })
        }
    }
}
```



```

        })

        val intent = Intent(this, MainActivity::class.java)
        startActivity(intent)
    }
}

```

com/example/pizzaapp/CreateActivity.kt

```

package com.example.pizzaapp

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Toast
import androidx.core.view.get
import com.example.pizzaapp.databinding.ActivityCreatePizzaBinding
import com.example.pizzaapp.network.Pizza
import com.example.pizzaapp.network.RetrofitInstance
import retrofit2.Call
import retrofit2.Response

class CreatePizzaActivity : AppCompatActivity() {

    private lateinit var binding : ActivityCreatePizzaBinding

    override fun onCreate(savedInstanceState: Bundle?) { // Переопределение
        метода onCreate
        super.onCreate(savedInstanceState)
        binding = ActivityCreatePizzaBinding.inflate(layoutInflater)
        setContentView(binding.root)

        // Обработка события нажатия кнопки для сохранения изменений
        binding.btnSave.setOnClickListener {

            val name = binding.etPizzaName.text.toString()
            val topping = binding.etPizzaTopping.text.toString()

            val diameter = when(binding.rgDiameter.checkedRadioButtonId) {
                binding.rgDiameter[0].id-> 23
                binding.rgDiameter[1].id -> 30
                binding.rgDiameter[2].id -> 35
                binding.rgDiameter[3].id -> 40
                else -> 23
            }
        }
    }
}

```

```

        // Проверка пустых полей
        if (name.isNotEmpty() && topping.isNotEmpty())
        {
            RetrofitInstance.api.createPizza(name, topping, diameter)
                .enqueue(object : retrofit2.Callback<Pizza> {
                    override fun onResponse(call: Call<Pizza>, response:
Response<Pizza>) {
                        Toast.makeText(applicationContext, "Успех",
Toast.LENGTH_LONG).show()
                    }

                    override fun onFailure(call: Call<Pizza>, t:
Throwable) {
                        Toast.makeText(applicationContext, "Ошибка",
Toast.LENGTH_SHORT).show()
                    }
                })

            val intent = Intent(this, MainActivity::class.java)
            startActivity(intent)
        }
        else
        {
            Toast.makeText(applicationContext, "Неверные данные",
Toast.LENGTH_SHORT).show()
        }
    }
}

```

com/example/pizzaapp/UpdateActivity.kt

```

package com.example.pizzaapp

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Toast
import androidx.core.view.get
import com.example.pizzaapp.databinding.ActivityUpdateBinding
import com.example.pizzaapp.network.Pizza
import com.example.pizzaapp.network.RetrofitInstance
import retrofit2.Call
import retrofit2.Callback
import retrofit2.Response

class UpdateActivity : AppCompatActivity() {

    private lateinit var binding: ActivityUpdateBinding

    override fun onCreate(savedInstanceState: Bundle?) { // Переопределение
метода onCreate
        super.onCreate(savedInstanceState)
        binding = ActivityUpdateBinding.inflate(layoutInflater)
        setContentView(binding.root)

        // Получение данных из detail activity
        val bundle: Bundle? = intent.extras
        val pizzaName = bundle!!.getString("name")
        val pizzaTopping = bundle.getString("topping")
        val pizzaDiameter = bundle.getInt("diameter")
        val pizzaPK = bundle.getInt("pk")
    }
}

```

```

// Установка полученных значений во View
binding.etPizzaName.setText(pizzaName)
binding.etPizzaTopping.setText(pizzaTopping)

when(pizzaDiameter){
    23 -> binding.rbD23.toggle()
    30 -> binding.rbD30.toggle()
    35 -> binding.rbD35.toggle()
    40 -> binding.rbD40.toggle()
}

// Обработка события нажатия кнопки для сохранения изменений
binding.btnSave.setOnClickListener {

    val name = binding.etPizzaName.text.toString()
    val topping = binding.etPizzaTopping.text.toString()

    val diameter = when(binding.rgDiameter.checkedRadioButtonId) {
        binding.rgDiameter[0].id -> 23
        binding.rgDiameter[1].id -> 30
        binding.rgDiameter[2].id -> 35
        binding.rgDiameter[3].id -> 40
        else -> 23
    }

    val pizza = Pizza(pizzaPK, name, topping, diameter)
    RetrofitInstance.api.updatePizza(pizza.pk!!,
pizza).enqueue(object : Callback<Pizza> {
    override fun onResponse(
        call: Call<Pizza>,
        response: Response<Pizza>
    ) {
        Toast.makeText(applicationContext, "Успех",
Toast.LENGTH_SHORT).show()
    }

    override fun onFailure(call: Call<Pizza>, t: Throwable) {
        Toast.makeText(applicationContext, "Ошибка",
Toast.LENGTH_SHORT).show()
    }
})

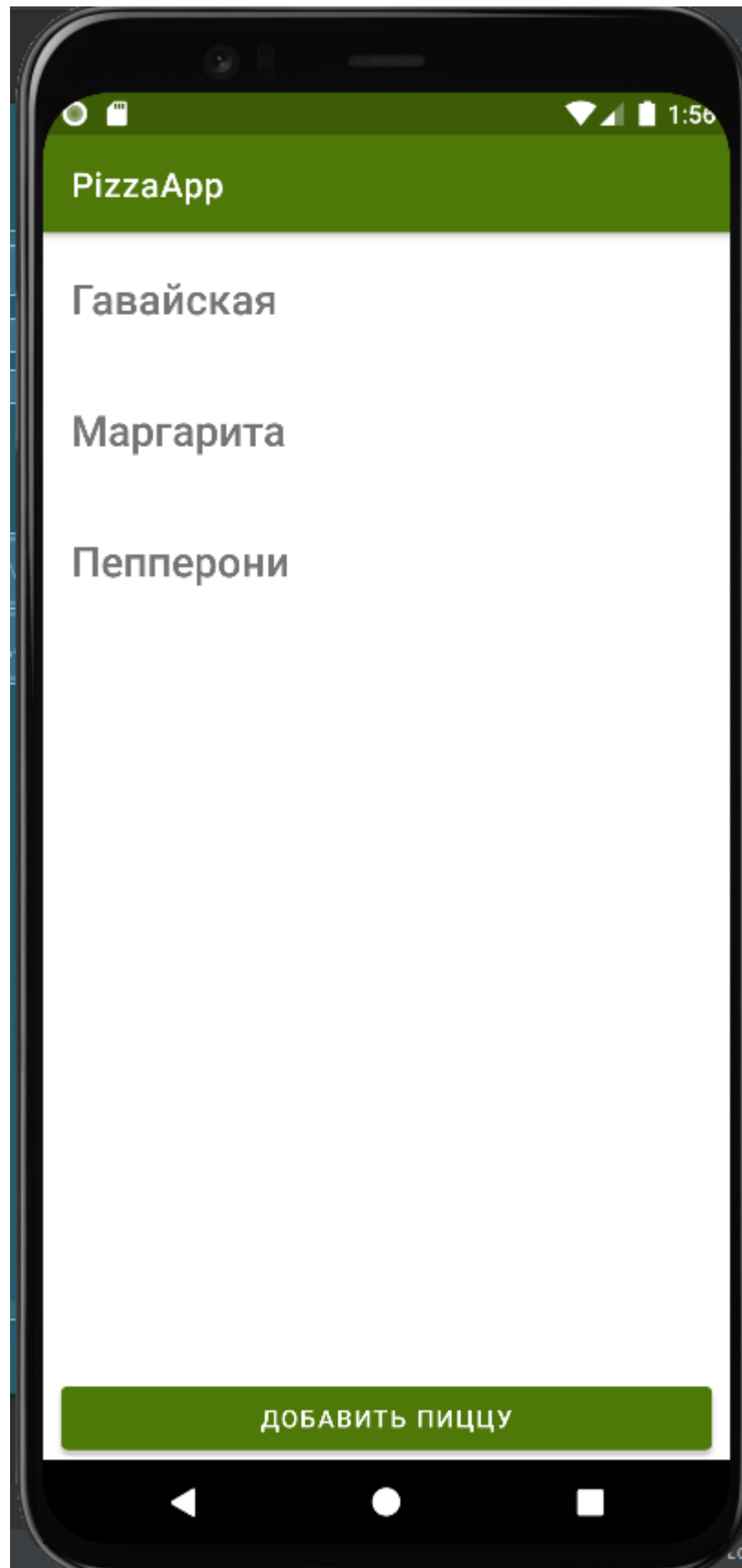
    val intent = Intent(this, MainActivity::class.java)

    startActivity(intent)
}
}
}

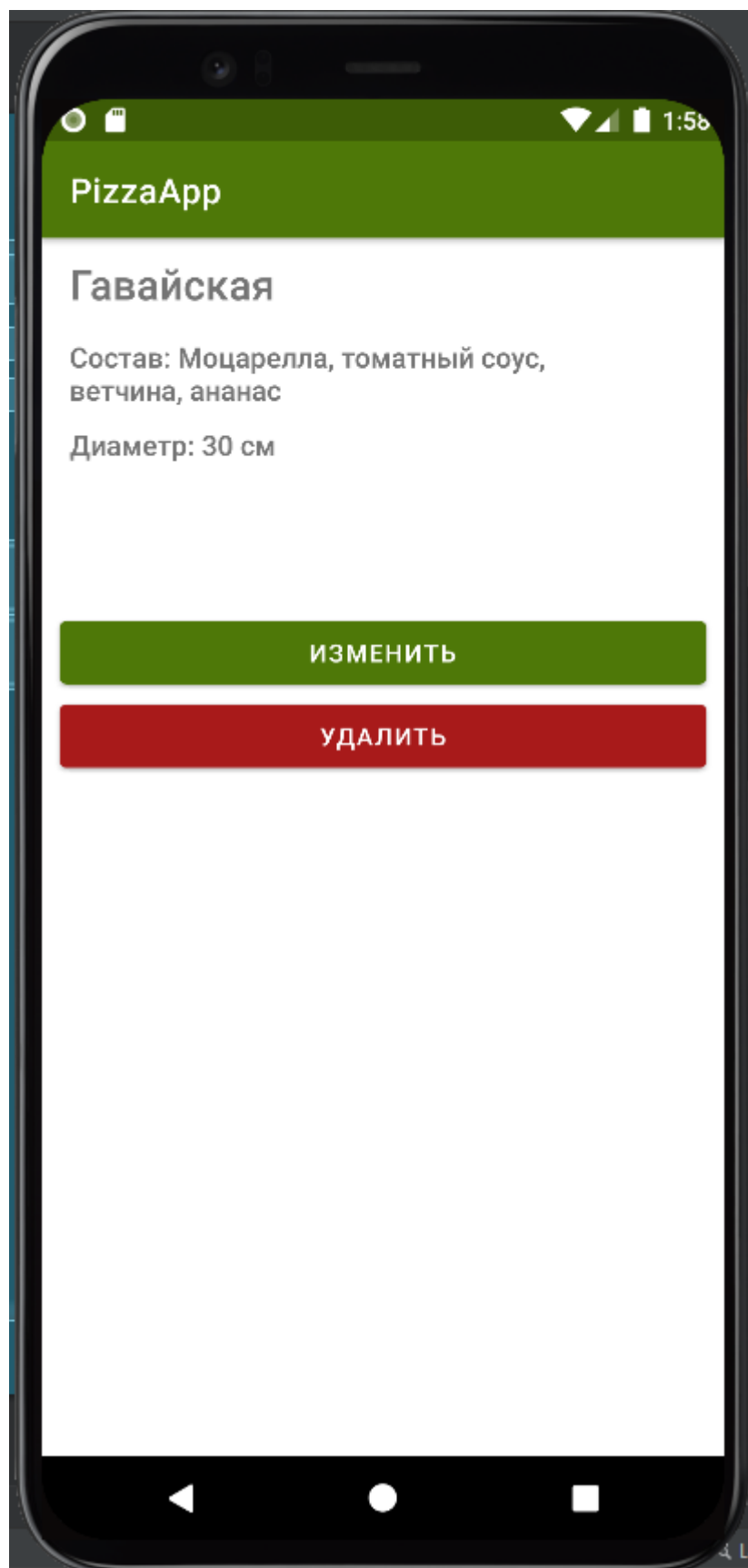
```

Примеры выполнения программы

Скриншот № 1 (Main Activity)



Скриншот № 2
(Detail Activity)



Скриншот № 3
(Create Activity)

PizzaApp

Название: Введите название

Состав: Введите состав

Диаметр

☐ 23 см

☐ 30 см

☐ 35 см

☐ 40 см

СОХРАНИТЬ

Скриншот № 3
(Update Activity)

PizzaApp

Название: Гавайская

Состав: Моцарелла, томатный соус, ветчи

Диаметр

☐ 23 см

☒ 30 см

☐ 35 см

☐ 40 см

СОХРАНИТЬ