Lappeenrannan teknillinen yliopisto

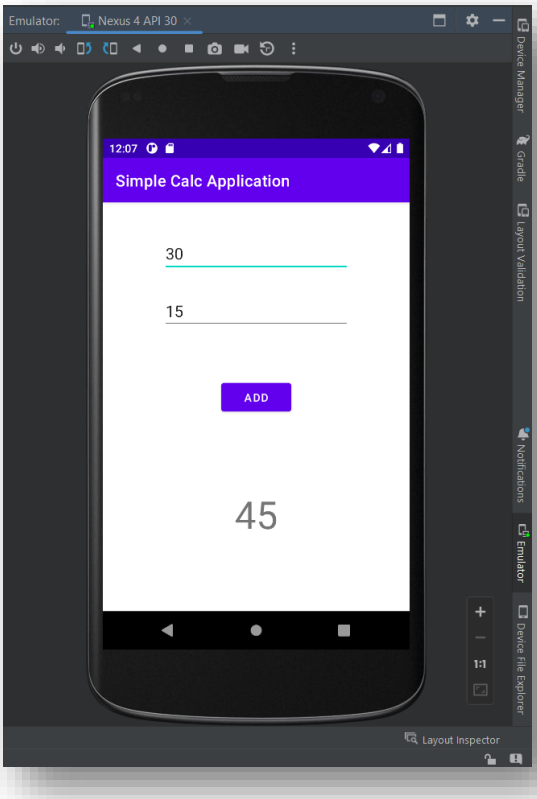
Software Development Skills: Mobile 2024-25, Online course

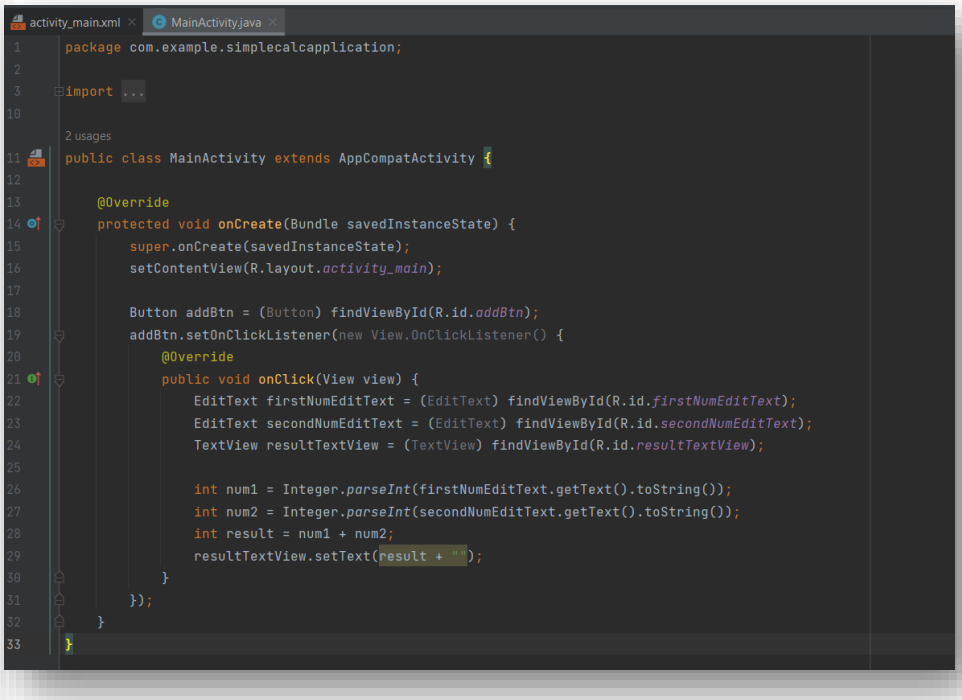
Tommi Uponen, 001134917

Learning Diary

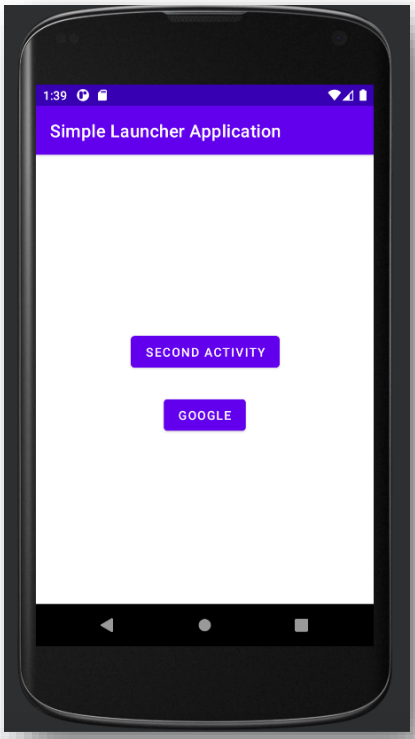
5.12.2024

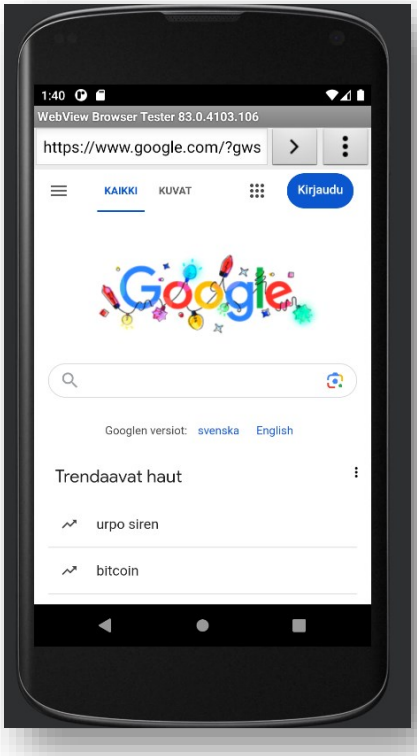
Today, I started the online course by looking at the introductory beginner video in Moodle. I have previously used Android Studio, so I think I'll manage quite well with the tasks on this course. The first task in the introductory video is to build an application which has 2 input fields, a button and a result screen. The application is a simple calculator that adds up the numbers of both input fields and displays them in the result box. The introductory video covered the setup of Android Studio, developing the described application and debugging as well. I have previous experience with this, so the setup wasn't necessary for me but developing the application and debugging refreshed my understanding of how Android Studio works again so it was useful overall. Here are the results of the first video:

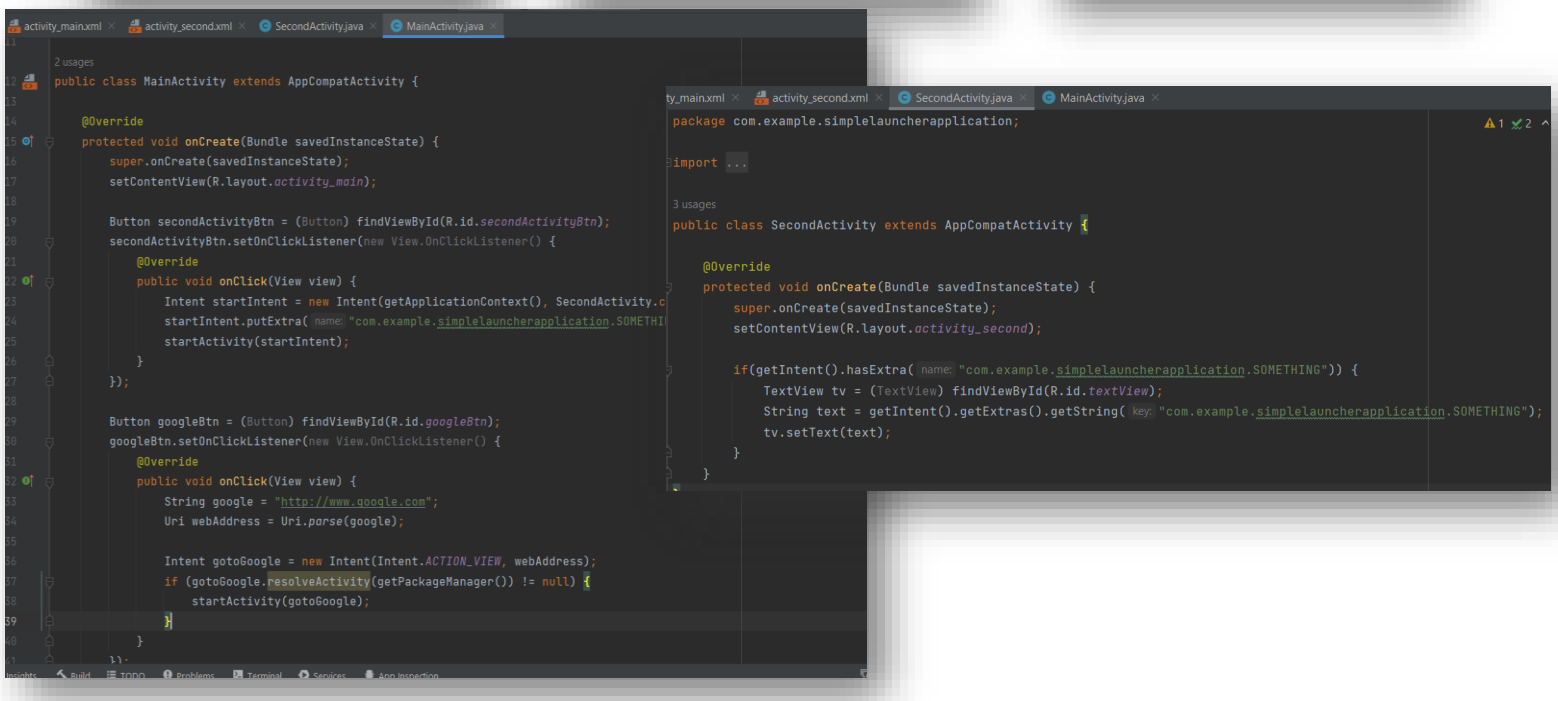




The first introductory video was so simple that I decided to continue my work and do the application mentioned in the second beginner video of this course. The second video covers information mostly about activities and intents which are something that I need some practice because I haven't touched Android Studio for a while even though I have a pretty good understanding of it. This time the application had 2 buttons which will direct the user either to a second activity or to the google homepage based on the URL. I created the second activity as instructed with the extra text which will be attached to the second activity's TextView once I click the button to direct me there. Lastly, I created the code for the website button which will direct the user to the homepage of google. I think this was also a good refresher with the basics. However, here are the results of the second video:

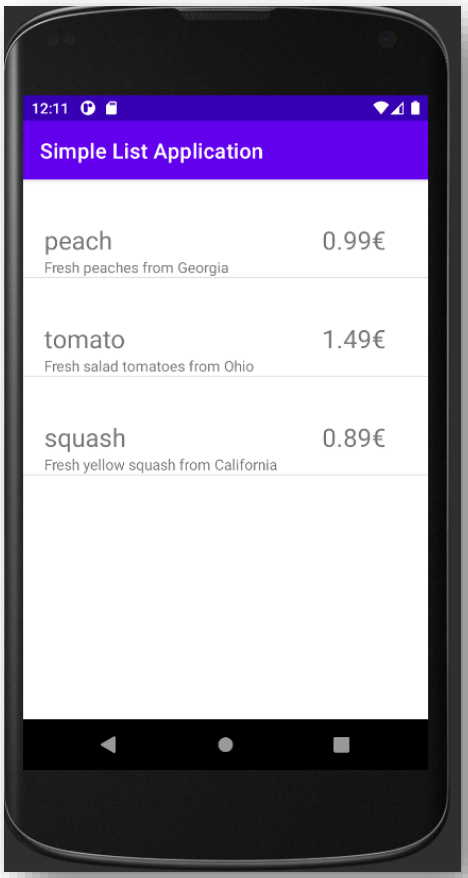






6.12.2024

Today, I continued my work and decided to do the third beginner video of this course. This video covers ListViews, custom layouts and ImageViews which are all familiar to me but a refresher on ListViews will come in handy. I started delving straight into the video to learn other features of Android Studio. Today, I encountered some problems with Android Studio failing to launch an empty project because of NullPointerException. However, I figured it out and the problem was that my dependencies were changed to newer versions which caused errors with building gradle. After that error I started creating the described app that will have a ListView which will show you three items and when you click one of those items, it will direct the user to a new activity which will show an image of that specific item. This was quite easy to implement, however I had completely forgotten how adapters worked so creating an ItemAdapter was good practise. I learned a lot today because I didn't remember how adapters worked with list views, so this was overall a good learning experience. Here are the results of my application:

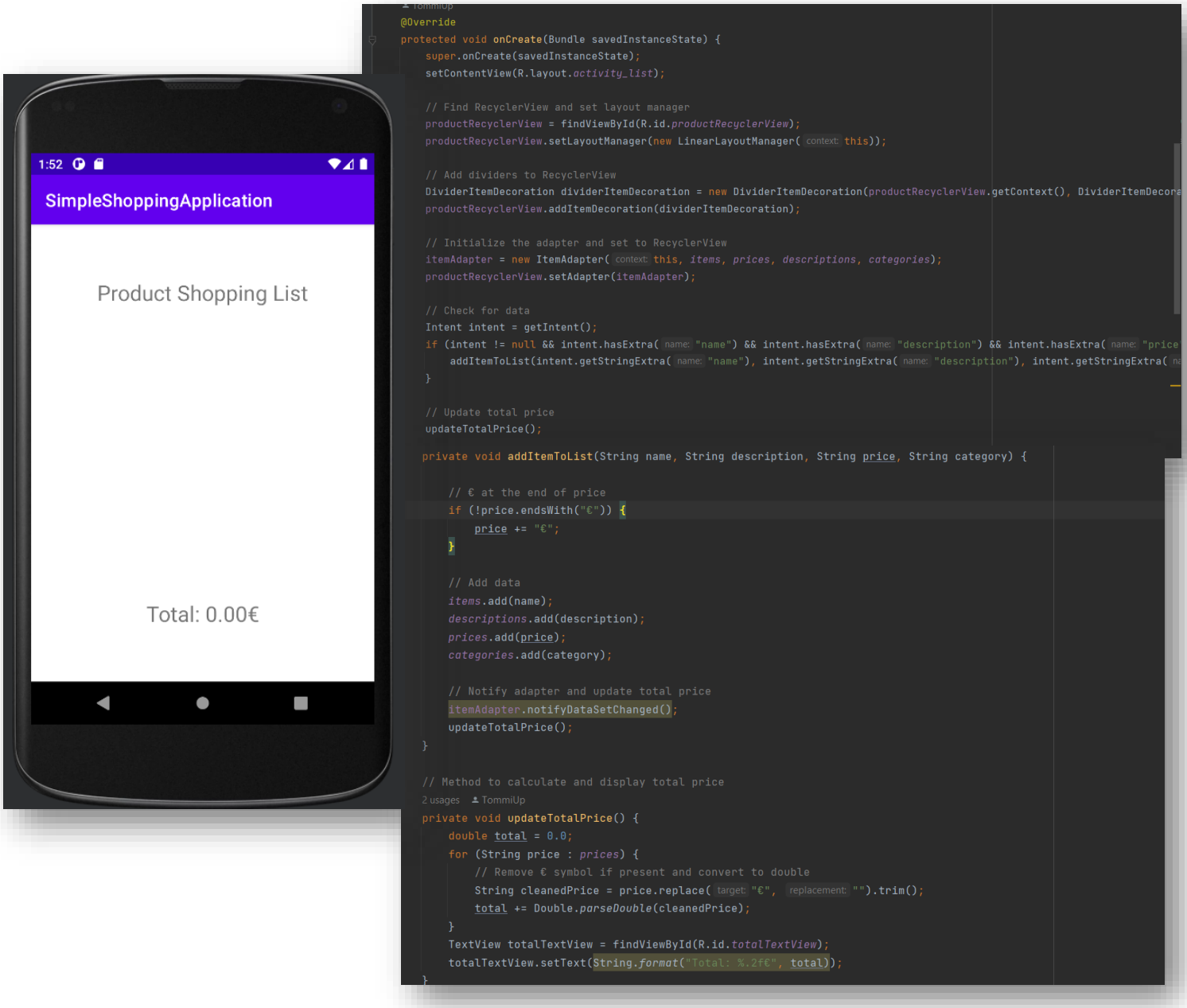


10.12.2024

Today, I decided to start my project for the course. I decided to combine all the information I have acquired from the beginner videos to make a simple shopping application. The shopping application has a main screen which has 2 buttons leading to the shopping list and product addition screen. The shopping list screen has a list containing all the products you have added and a total price of the products. The product addition screen has 4 sections: name, description, price and category. I started by implementing the main screen by adding 2 simple buttons and a TextView asking the user what he/she wants to do. I handled the click events which directs the user to the correct activities. I will try to show all the necessary code fields of my project, but rest of the code can be seen through my git. Here is the result of the main screen (MainActivity):

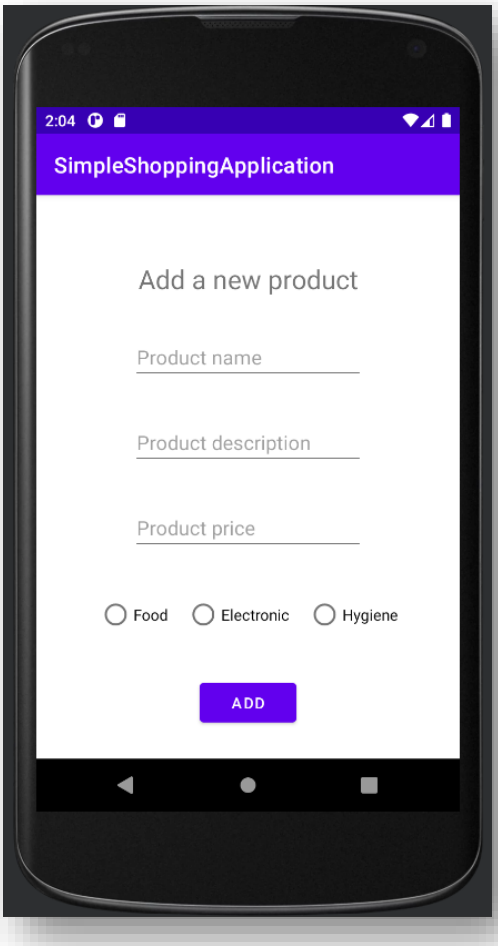
After finishing the main screen, I moved on to creating the shopping list screen. For this, I used a RecyclerView to display the list of products. I learned how to use an adapter (ItemAdapter) to manage and bind data to the RecyclerView. I also added a divider to visually separate the items in the list, which I found makes the interface clearer. Additionally, I included a TextView at the bottom of the screen to show the total price of all products in the list, which updates dynamically whenever a new product is added. Here

are the results of the shopping list screen (ListActivity) now without any products added as I haven't implemented that yet:



Next, I worked on the product addition screen. This screen includes EditText fields for entering the product name, description, and price, along with a RadioGroup for selecting the product category (e.g., food, electronic, hygiene). Implementing the RadioGroup was interesting because it allowed me to add visual categorization to the items in the shopping

list. I also made sure that all the fields must be filled, and the price field appends a euro € symbol. Here is the result of the product addition screen (AddActivity):




```
// Find views
EditText nameEditText = findViewById(R.id.nameEditText);
EditText descriptionEditText = findViewById(R.id.descriptionEditText);
EditText priceEditText = findViewById(R.id.priceEditText);
RadioGroup categoryRadioGroup = findViewById(R.id.categoryRadioGroup);
Button addBtn = findViewById(R.id.addBtn);

// Handle Add Button Click
TommiUp
addBtn.setOnClickListener(new View.OnClickListener() {
    TommiUp
    @Override
    public void onClick(View view) {
        String name = nameEditText.getText().toString().trim();
        String description = descriptionEditText.getText().toString().trim();
        String price = priceEditText.getText().toString().trim();

        // Get selected category
        int selectedCategoryId = categoryRadioGroup.getCheckedRadioButtonId();
        String category = "";

        if (selectedCategoryId == R.id.foodRadioBtn) {
            category = "food";
        } else if (selectedCategoryId == R.id.electronicRadioBtn) {
            category = "electronics";
        } else if (selectedCategoryId == R.id.hygieneRadioBtn) {
            category = "hygiene";
        }
```

```
// Check for empty fields
if (name.isEmpty() || description.isEmpty() || price.isEmpty() || category.isEmpty()) {
    Toast.makeText( context: AddActivity.this, text: "Please fill all fields and choose a category", Toast.LENGTH_SHO
} else {
    // € at the end of price
    if (!price.endsWith("€")) {
        price += "€";
    }

    // Pass data to ListActivity
    Intent intent = new Intent( packageContext: AddActivity.this, MainActivity.class);
    intent.putExtra( name: "name", name);
    intent.putExtra( name: "description", description);
    intent.putExtra( name: "price", price);
    intent.putExtra( name: "category", category);

    // Add item to lists in ListActivity
    ListActivity.items.add(name);
    ListActivity.descriptions.add(description);
    ListActivity.prices.add(price);
    ListActivity.categories.add(category);

    // Finish Activity
    setResult(RESULT_OK, intent);
    finish();
}
```

While testing the application, I encountered challenges with passing data between activities using Intents. To fix this, I passed the product information from the product addition screen back to the main list by adding the data to static lists managed in the ListActivity class. Also, the positioning of the RadioGroup was painful to address but I managed to align it perfectly.

Finally, I focused on adding finishing touches to improve the user experience. One of the key learning moments was implementing icons for the product categories. Each item in the shopping list uses a layout (product_list_detail), where I added an ImageView to display an appropriate icon based on the selected category. Configuring this logic in the ItemAdapter class gave me valuable knowledge into dynamically binding resources to views, making the application clearer.

Through this project, I've applied multiple concepts, including RecyclerView, adapters, ConstraintLayout, RadioGroup, and activity communication with Intents. I also learned to troubleshoot issues like layout misalignments and data passing errors. Overall, it was a great learning experience to build this Android application and here are the results of the final look: