

```

package com.mistershorr.birthdaytracker

import android.R.attr
import android.app.Activity
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import androidx.activity.result.ActivityResult
import androidx.activity.result.contract.ActivityResultContracts
import com.backendless.Backendless
import com.mistershorr.birthdaytracker.databinding.ActivityLoginBinding
import com.backendless.exceptions.BackendlessFault

import com.backendless.BackendlessUser

import com.backendless.async.callback.AsyncCallback

import android.R.attr.password
import android.util.Log
import android.widget.Toast

//import com.mistershorr.loginandregistration.databinding.ActivityLoginBinding

class LoginActivity : AppCompatActivity() {

    private lateinit var binding: ActivityLoginBinding

    // any time you need what would have been a "static" variable in java, you use
    // companion object in Kotlin. You access with ClassName.VARIABLE_NAME like Math.PI
    companion object {
        // keys for the key-value pairs for the intent extras
        val EXTRA_USERNAME = "username"
        val EXTRA_PASSWORD = "password"
    }

    // starting an activity for a result
    // Step 1: Register the Activity with a contract
    val startRegistrationForResult = registerForActivityResult(ActivityResultContracts.StartActivityForResult
()) {
        result: ActivityResult ->
        // decide what to do if the result is ok (if it was successful)
        if (result.resultCode == Activity.RESULT_OK) {
            val intent = result.data
            // Handle the Intent
            // note: editTexts are different from textViews in that you have to call setText
            binding.editTextLoginUsername.setText(intent?.getStringExtra(EXTRA_USERNAME))
            binding.editTextLoginPassword.setText(intent?.getStringExtra(EXTRA_PASSWORD))
        }
        // decide what to do if it was unsuccessful with a RESULT_CANCELED
    }

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityLoginBinding.inflate(layoutInflater)
        setContentView(binding.root)

        Backendless.initApp( this, Constants.APP_ID, Constants.API_KEY)

        binding.textViewLoginCreateAccount.setOnClickListener {
            // launch the registration activity
            // pass the values of username and password along to the new activity
            // 1. extract any information you might need from edit texts
            val username = binding.editTextLoginUsername.text.toString()
            val password = binding.editTextLoginPassword.text.toString()

            // 2. create the intent and package the extras
            // the context is the activity you are in (here we can use this)
            val registrationIntent
            = Intent(this, RegistrationActivity::class.java).apply {
                putExtra(EXTRA_USERNAME, username)
                putExtra(EXTRA_PASSWORD, password)
            }

            // 3. launch the activity
            startActivity(registrationIntent)
        }
    }
}

```

```
// 3b. Alternate: Could launch the activity for a result instead
// use the variable from the register for result contract above
startRegistrationForResult.launch(registrationIntent)
// do not forget to call Backendless.initApp in the app initialization code

// do not forget to call Backendless.initApp in the app initialization code

}

binding.buttonLoginLogin.setOnClickListener {
    val username = binding.editTextLoginUsername.text.toString()
    val password = binding.editTextLoginPassword.text.toString()
    Backendless.UserService.login(
        username,
        password,
        object : AsyncCallback<BackendlessUser?> {
            override fun handleResponse(user: BackendlessUser?) {
                // user has been logged in
                Log.d("LoginActivity", "handleResponse: ${user?.email}")
                Toast.makeText(this@LoginActivity, "${user?.getProperty("username")} has logged in
successfully", Toast.LENGTH_SHORT).show()

                // launch the birthday list activity
                val birthdayListIntent = Intent(this@LoginActivity, BirthdayListActivity::class.java)
                startActivity(birthdayListIntent)
                // finish this activity
                finish()
            }

            override fun handleFault(fault: BackendlessFault) {
                // login failed, to get the error code call fault.getCode()
                Log.d("LoginActivity", "handleFault: ${fault.message}")
            }
        })
}
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".BirthdayDetailActivity">

    <CheckBox
        android:id="@+id/checkBox_birthdayDetail_giftBought"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Purchased?"
        app:layout_constraintBottom_toTopOf="@+id/textView_birthdayDetail_birthdate"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editText_birthdayDetail_desiredGift" />

    <EditText
        android:id="@+id/editText_birthdayDetail_budget"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Budget"
        android:inputType="none"
        android:enabled="false"
        app:layout_constraintBottom_toTopOf="@+id/editText_birthdayDetail_desiredGift"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editText_birthdayDetail_name" />

    <EditText
        android:id="@+id/editText_birthdayDetail_desiredGift"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Desired Gift"
        android:inputType="none"
        android:enabled="false"
        app:layout_constraintBottom_toTopOf="@+id/checkBox_birthdayDetail_giftBought"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editText_birthdayDetail_budget" />

    <EditText
        android:id="@+id/editText_birthdayDetail_name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Name"
        android:inputType="none"
        android:enabled="false"
        app:layout_constraintBottom_toTopOf="@+id/editText_birthdayDetail_budget"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/textView_birthdayDetail_birthdate"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintBottom_toTopOf="@+id/textView_birthdayDetail_daysUntil"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/checkBox_birthdayDetail_giftBought"
        tools:text="Birthdate" />

    <TextView
        android:id="@+id/textView_birthdayDetail_daysUntil"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
```

```
    app:layout_constraintHorizontal_bias="0.5"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView_birthdayDetail_birthdate"
    tools:text="Calculated Days Until Birthday" />

<Button
    android:id="@+id/button_birthdayDetail_save"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Save"
    android:visibility="gone"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

```
package com.mistershorr.birthdaytracker

object Constants {
    val APP_ID = "6AF83806-4398-D062-FF57-E1A25740F700"
    val API_KEY = "AB6050CA-13DE-4AC6-B675-3E17B1D2C725"
}
```

```
package com.mistershorr.birthdaytracker

import android.os.Parcelable
import kotlinx.parcelize.Parcelize
import java.util.*

// backendless and other online Baas (backend as a service)
// apis often require your model class to have a
// default, no parameter constructor
// val blah = Person()
// in kotlin, you give each field a default value
// and then you can use a no parameter constructor
@Parcelize
data class Person(
    var name : String = "Your mom",
    var birthday : Date = Date(1646932056741),
    var budget : Int = 1,
    var desiredGift : String = "String",
    var previousGifts : List<String> = listOf(),
    var previousGiftsToMe : List<String> = listOf(),
    var isPurchased : Boolean = false,
    var ownerId : String? = null,
    var objectId : String? = null
) : Parcelable {
    // TODO: have methods to return the calculated values of age, days until birthday
}
```

```

package com.mistershorr.birthdaytracker

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.util.Log
import androidx.recyclerview.widget.LinearLayoutManager
import com.mistershorr.birthdaytracker.databinding.ActivityBirthdayListBinding
import com.backendless.exceptions.BackendlessFault

import com.backendless.async.callback.AsyncCallback

import com.backendless.Backendless
import com.backendless.persistence.DataQueryBuilder

class BirthdayListActivity : AppCompatActivity() {

    lateinit var binding : ActivityBirthdayListBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityBirthdayListBinding.inflate(layoutInflater)
        setContentView(binding.root)

        // make an onclicklistener for the floating action button
        // that launches the detailActivity
        binding.floatingActionButton.setOnClickListener {
            // make the intent
            val detailIntent = Intent(this, BirthdayDetailActivity::class.java)
            // launch the activity
            detailIntent.
        }
    }

    override fun onStart() {
        super.onStart()

        // load the data from the database
        // put it into the recyclerView
        // this is in onStart so when we return from the detail view
        // it will reload the data from the database
        loadDataFromBackendless()
    }

    private fun loadDataFromBackendless() {

        // retrieve only objects whose ownerId matches the user's
        val objectId = Backendless.UserService.CurrentUser().objectId
        val whereClause = "ownerId = '$objectId'"
        val queryBuilder = DataQueryBuilder.create()
        queryBuilder.whereClause = whereClause

        // this retrieves all objects regardless of owner
        // but adding the builder will now search with the where clause
        Backendless.Data.of(Person::class.java).find(queryBuilder, object : AsyncCallback<List<Person>?> {
            override fun handleResponse(foundPeople: List<Person>?) {
                // all person instances have been found
                Log.d("BirthdayList", "handleResponse: ${foundPeople}")
                val adapter = BirthdayAdapter((foundPeople ?: listOf<Person>()))
                binding.recyclerViewBirthdayListPeople.adapter = adapter
                binding.recyclerViewBirthdayListPeople.layoutManager =
                    LinearLayoutManager(this@BirthdayListActivity)
            }

            override fun handleFault(fault: BackendlessFault) {
                // an error has occurred, the error code can be retrieved with fault.getCode()
                Log.d("BirthdayList", "handleFault: ${fault.message}")
            }
        })
    }
}

```

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".RegistrationActivity">

    <Button
        android:id="@+id/button_registration_register"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="24dp"
        android:text="Register"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editText_registration_email" />

    <EditText
        android:id="@+id/editText_registration_username"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="24dp"
        android:ems="10"
        android:hint="Username"
        android:inputType="textPersonName"
        android:minHeight="48dp"
        app:layout_constraintBottom_toTopOf="@+id/editText_registration_password"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editText_registration_name" />

    <EditText
        android:id="@+id/editText_registration_password"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="24dp"
        android:ems="10"
        android:hint="Password"
        android:inputType="textPassword"
        android:minHeight="48dp"
        app:layout_constraintBottom_toTopOf="@+id/editText_registration_confirmPassword"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editText_registration_username" />

    <EditText
        android:id="@+id/editText_registration_confirmPassword"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="24dp"
        android:ems="10"
        android:hint="Confirm Password"
        android:inputType="textPassword"
        android:minHeight="48dp"
        app:layout_constraintBottom_toTopOf="@+id/editText_registration_email"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editText_registration_password" />

    <EditText
        android:id="@+id/editText_registration_email"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="24dp"
        android:ems="10"
        android:hint="Email"
        android:inputType="textEmailAddress"
        android:minHeight="48dp"
        app:layout_constraintBottom_toTopOf="@+id/button_registration_register"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
```

```
    app:layout_constraintTop_toBottomOf="@+id/editText_registration_confirmPassword" />

<EditText
    android:id="@+id/editText_registration_name"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="24dp"
    android:ems="10"
    android:hint="Name"
    android:inputType="textPersonName"
    android:minHeight="48dp"
    app:layout_constraintBottom_toTopOf="@+id/editText_registration_username"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.5"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_chainStyle="packed" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/layout_birthdayItem"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="8dp">

    <CheckBox
        android:id="@+id/checkBox_birthdayItem_giftBought"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Gift Bought?"
        app:layout_constraintBottom_toBottomOf="@+id/textView_birthdayItem_daysUntil"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="@+id/textView_birthdayItem_name" />

    <TextView
        android:id="@+id/textView_birthdayItem_name"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:textSize="20sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toStartOf="@+id/checkBox_birthdayItem_giftBought"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        tools:text="Some Name" />

    <TextView
        android:id="@+id/textView_birthdayItem_daysUntil"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toStartOf="@+id/checkBox_birthdayItem_giftBought"
        app:layout_constraintStart_toStartOf="@+id/textView_birthdayItem_name"
        tools:text="3 days until" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

```

package com.mistershorr.birthdaytracker

import android.app.DatePickerDialog
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.text.InputType
import android.view.Menu
import android.view.MenuInflater
import android.view.MenuItem
import android.view.View
import android.widget.DatePicker
import com.mistershorr.birthdaytracker.databinding.ActivityBirthdayDetailBinding
import java.text.DateFormat
import java.text.SimpleDateFormat
import java.util.*
import com.backendless.exceptions.BackendlessFault

import com.backendless.async.callback.AsyncCallback

import com.backendless.Backendless

import android.R.string.no
import android.util.Log
import android.widget.Toast

class BirthdayDetailActivity : AppCompatActivity() {

    lateinit var binding : ActivityBirthdayDetailBinding
    var personIsEditable = false
    var cal = Calendar.getInstance()

    companion object {
        val EXTRA_PERSON = "person"
    }

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityBirthdayDetailBinding.inflate(layoutInflater)
        setContentView(binding.root)

        // get the intent and fill in all the values
        var person = intent.getParcelableExtra<Person>(EXTRA_PERSON)
        if (person == null) {
            person = Person()
        }
        binding.editTextBirthdayDetailName.setText(person.name)
        binding.editTextBirthdayDetailDesiredGift.setText(person.desiredGift)
        binding.editTextBirthdayDetailBudget.setText(person.budget.toString())
        binding.checkBoxBirthdayDetailGiftBought.isChecked = person.isPurchased ?: false
        binding.textViewBirthdayDetailBirthdate.text = person.birthday.toString()

        binding.buttonBirthdayDetailSave.setOnClickListener {
            Toast.makeText(this, "Button Clicked", Toast.LENGTH_SHORT).show()

            person!!.name = binding.editTextBirthdayDetailName.text.toString()
            person!!.budget = binding.editTextBirthdayDetailBudget.text.toInt()

            Backendless.Data.of(Person::class.java).save(person, object : AsyncCallback<Person?> {
                override fun handleResponse(response: Person?) {
                    Toast.makeText(this@BirthdayDetailActivity, "save successful", Toast.LENGTH_SHORT).show()
                }

                override fun handleFault(fault: BackendlessFault) {
                    Log.d("BirthdayActivity", "handleFault: ${fault.message}")
                }
            })
        }
    }

    // datepicker code from https://www.tutorialkart.com/kotlin-android/android-datepicker-kotlin-example
}

val dateSetListener = object : DatePickerDialog.OnDateSetListener {
    override fun onDateSet(view: DatePicker, year: Int, monthOfYear: Int,
                         dayOfMonth: Int) {
        cal.set(Calendar.YEAR, year)
        cal.set(Calendar.MONTH, monthOfYear)
        cal.set(Calendar.DAY_OF_MONTH, dayOfMonth)
    }
}

```

```

        updateDateInView()
    }

    binding.textViewBirthdayDetailBirthdate.setOnClickListener {
        if(personIsEditable) {
            DatePickerDialog(this, dateSetListener,
                cal.get(Calendar.YEAR),
                cal.get(Calendar.MONTH),
                cal.get(Calendar.DAY_OF_MONTH)).show()
        }
    }
}

private fun updateDateInView() {
    val myFormat = "MM/dd/yyyy" // mention the format you need
    val sdf = SimpleDateFormat(myFormat, Locale.US)
    binding.textViewBirthdayDetailBirthdate.text = sdf.format(cal.getTime())
}

override fun onCreateOptionsMenu(menu: Menu): Boolean {
    val inflater: MenuInflater = menuInflater
    inflater.inflate(R.menu.menu_birthday_detail, menu)
    return true
}

override fun onOptionsItemSelected(item: MenuItem): Boolean {
    // Handle item selection
    return when (item.itemId) {
        R.id.menu_item_birthday_detail_edit -> {
            toggleEditable()
            true
        }
        else -> super.onOptionsItemSelected(item)
    }
}

private fun toggleEditable() {
    if (personIsEditable) {
        personIsEditable = false
        binding.buttonBirthdayDetailSave.isEnabled = false
        binding.buttonBirthdayDetailSave.visibility = View.GONE
        binding.checkBoxBirthdayDetailGiftBought.isEnabled = false
        binding.editTextBirthdayDetailName.inputType = InputType.TYPE_NULL
        binding.editTextBirthdayDetailName.isEnabled = false
        binding.editTextBirthdayDetailDesiredGift.inputType = InputType.TYPE_NULL
        binding.editTextBirthdayDetailDesiredGift.isEnabled = false
        binding.editTextBirthdayDetailBudget.inputType = InputType.TYPE_NULL
        binding.editTextBirthdayDetailBudget.isEnabled = false
        binding.textViewBirthdayDetailBirthdate.isClickable = false
    } else {
        personIsEditable = true
        binding.buttonBirthdayDetailSave.isEnabled = true
        binding.buttonBirthdayDetailSave.visibility = View.VISIBLE
        binding.checkBoxBirthdayDetailGiftBought.isEnabled = true
        binding.editTextBirthdayDetailName.inputType = InputType.TYPE_TEXT_VARIATION_PERSON_NAME
        binding.editTextBirthdayDetailName.isEnabled = true
        binding.editTextBirthdayDetailDesiredGift.inputType = InputType.TYPE_TEXT_VARIATION_PERSON_NAME
        binding.editTextBirthdayDetailDesiredGift.isEnabled = true
        binding.editTextBirthdayDetailBudget.inputType = InputType.TYPE_NUMBER_VARIATION_NORMAL
        binding.editTextBirthdayDetailBudget.isEnabled = true
        binding.textViewBirthdayDetailBirthdate.isClickable = true
    }
}
}

```

```
package com.mistershorr.birthdaytracker

// object keyword makes it so all the functions are
// static functions
object RegistrationUtil {
    // use this in the test class for the is username taken test
    // make another similar list for some taken emails
    var existingUsers = listOf("cosmicF", "cosmicY", "bob", "alice")
    // you can use listOf<type>() instead of making the list & adding individually
    // List<String> blah = new ArrayList<String>();
    // blah.add("hi")
    // blah.add("hello")
    //

    // isn't empty
    // already taken
    // minimum number of characters is 3
    fun validateUsername(username: String) : Boolean {
        return true
    }

    // make sure meets security requirements (deprecated ones that are still used everywhere)
    // min length 8 chars
    // at least one digit
    // at least one capital letter
    // both passwords match
    // not empty
    fun validatePassword(password : String, confirmPassword: String) : Boolean {
        return true
    }

    // isn't empty
    fun validateName(name: String) : Boolean {
        return true
    }

    // isn't empty
    // make sure the email isn't used
    // make sure it's in the proper email format user@domain.tld
    fun validateEmail(email: String) : Boolean {
        return true
    }
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".LoginActivity">

    <Button
        android:id="@+id/button_login_login"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="64dp"
        android:text="@string/login"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView_login_createAccount" />

    <EditText
        android:id="@+id/editText_login_username"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="@string/username"
        android:inputType="textPersonName"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="@+id/guideline2" />

    <EditText
        android:id="@+id/editText_login_password"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="32dp"
        android:ems="10"
        android:inputType="textPassword"
        android:hint="@string/password"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editText_login_username" />

    <TextView
        android:id="@+id/textView_login_createAccount"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="@string/register_an_account"
        android:textColor="@color/links"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editText_login_password" />

    <androidx.constraintlayout.widget.Guideline
        android:id="@+id/guideline2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        app:layout_constraintGuide_percent="0.3" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

```

package com.mistershorr.birthdaytracker

import android.app.Activity
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Toast
import com.backendless.Backendless
import com.backendless.BackendlessUser
import com.backendless.async.callback.AsyncCallback
import com.backendless.exceptions.BackendlessFault
import com.mistershorr.birthdaytracker.databinding.ActivityRegistrationBinding
import java.util.*

class RegistrationActivity : AppCompatActivity() {

    private lateinit var binding: ActivityRegistrationBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityRegistrationBinding.inflate(layoutInflater)
        setContentView(binding.root)

        Backendless.initApp( this, Constants.APP_ID, Constants.API_KEY)

        // access any values that were sent to us from the intent that launched this activity
        val username = intent.getStringExtra(LoginActivity.EXTRA_USERNAME)
        val password = intent.getStringExtra(LoginActivity.EXTRA_PASSWORD)
        Toast.makeText(this, "user:$username pwd $password", Toast.LENGTH_SHORT).show()

        binding.buttonRegistrationRegister.setOnClickListener {

            if(!RegistrationUtil.validateName(
                binding.editTextRegistrationName.text.toString())) {
                Toast.makeText(this, "Invalid name", Toast.LENGTH_SHORT).show()
            }

            else {
                registerUser()
            }
        }
    }

    private fun registerUser() {
        val user = BackendlessUser()
        user.setProperty("email", "${binding.editTextRegistrationEmail.text.toString()}")
        user.password = "${binding.editTextRegistrationPassword.text.toString()}"
        user.setProperty("username", "${binding.editTextRegistrationUsername.text.toString()}")
        user.setProperty("name", "${binding.editTextRegistrationName.text.toString()}")


        Backendless.UserService.register(user, object : AsyncCallback<BackendlessUser?> {
            override fun handleResponse(registeredUser: BackendlessUser?) {
                Toast.makeText(this@RegistrationActivity, "registration successful", Toast.LENGTH_SHORT).show()
            }

            override fun handleFault(fault: BackendlessFault) {
                // an error has occurred, the error code can be retrieved with fault.getCode()
                Toast.makeText(this@RegistrationActivity, "${fault.message}", Toast.LENGTH_SHORT).show()
            }
        })
    }

    private fun returnToLogin(username: String?, password: String?) {
        var returnToLoginIntent = Intent().apply {
            putExtra(
                LoginActivity.EXTRA_USERNAME,
                binding.editTextRegistrationUsername.text.toString())
            putExtra(
                LoginActivity.EXTRA_PASSWORD,
                binding.editTextRegistrationPassword.text.toString())
        }
    }
}

```

```
        }
        setResult(Activity.RESULT_OK, returnToLoginIntent)
        finish()
    }
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".BirthdayListActivity">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recyclerView_birthdayList_people"
        android:layout_width="0dp"
        android:layout_height="0dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <com.google.android.material.floatingactionbutton.FloatingActionButton
        android:id="@+id/floatingActionButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="32dp"
        android:layout_marginBottom="32dp"
        android:clickable="true"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:srcCompat="@android:drawable/ic_input_add"
        tools:ignore="SpeakableTextPresentCheck" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

```

package com.mistershorr.birthdaytracker

import android.content.Intent
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.CheckBox
import android.widget.TextView
import androidx.constraintlayout.widget.ConstraintLayout
import androidx.recyclerview.widget.RecyclerView
import java.util.*

class BirthdayAdapter(var birthdayList: List<Person>) : RecyclerView.Adapter<BirthdayAdapter.ViewHolder>() {

    class ViewHolder(view: View) : RecyclerView.ViewHolder(view) {

        val textViewName : TextView
        val textViewDaysUntil: TextView
        val layout : ConstraintLayout
        val checkBoxGiftBought : CheckBox

        init {
            textViewName = view.findViewById(R.id.textView_birthdayItem_name)
            textViewDaysUntil = view.findViewById(R.id.textView_birthdayItem_daysUntil)
            layout = view.findViewById(R.id.layout_birthdayItem)
            checkBoxGiftBought = view.findViewById(R.id.checkBox_birthdayItem_giftBought)
        }
    }

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {
        val view = LayoutInflater.from(parent.context)
            .inflate(R.layout.item_birthday, parent, false)

        return ViewHolder(view)
    }

    override fun onBindViewHolder(holder: ViewHolder, position: Int) {
        val context = holder.layout.context
        val person = birthdayList[position]
        holder.textViewName.text = person.name
        var daysDiff = calculateDaysLeft(person)

        holder.textViewDaysUntil.text = "${daysDiff} left"
        holder.checkBoxGiftBought.isChecked = person.isPurchased
        holder.layout.setOnClickListener {
            // pass the clicked on Person object to the detail activity
            val detailIntent = Intent(context, BirthdayDetailActivity::class.java)
            detailIntent.putExtra("person", person)
            context.startActivity(detailIntent)
        }
    }

    private fun calculateDaysLeft(person: Person): Long {
        // need to calculate how many days until the person's next birthday
        // someone's birthday is April 5th coming up
        // Date of their birthday - Today's date
        // someone's birthday
        val calendarBday = Calendar.getInstance()
        calendarBday.time = person.birthday
        val calendarToday = Calendar.getInstance()
        val bdayMonth = calendarBday.get(Calendar.MONTH)
        val bdayDay = calendarBday.get(Calendar.DAY_OF_MONTH)
        val todayMonth = calendarToday.get(Calendar.MONTH)
        val todayDay = calendarToday.get(Calendar.DAY_OF_MONTH)
        val todayYear = calendarToday.get(Calendar.YEAR)
        // when should calendarBday be the current year?
        // if the month/day are after the current date
        // same month, need the day to be after
        // different month, just need month to be after
        if (bdayMonth > todayMonth ||
            bdayMonth == todayMonth && bdayDay > todayDay) {
            calendarBday.set(Calendar.YEAR, todayYear)
        }
        // when should calendarBday be next year?
        // of the month/day are before the current date
        // if month/day are the same, display TODAY!!!
        else if (bdayMonth == todayMonth && bdayDay == todayDay) {
    }
}

```

```
    calendarBday.set(Calendar.YEAR, todayYear)
}
else {
    calendarBday.set(Calendar.YEAR, todayYear + 1)
}
var difference = calendarBday.timeInMillis - calendarToday.timeInMillis
var daysDiff = difference / (1000 * 60 * 60 * 24)
return daysDiff
}

override fun getItemCount(): Int {
    return birthdayList.size
}
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:app="http://schemas.android.com/apk/res-auto"
      xmlns:android="http://schemas.android.com/apk/res/android">

    <item
        android:icon="@drawable/ic_baseline_edit_24"
        android:title="Edit Person"
        android:id="@+id/menu_item_birthday_detail_edit"
        app:showAsAction="always"/>

</menu>
```