Nathan Brathwaite, Zachary Aird, Khareen Proverb

Abstract

For this project, we chose to create an android mobile application for use on multiple devices.  
The purpose of the application was to provide users with the ability to purchase tickets, view animals as well as the ability to navigate throughout the zoo.

seaside developemnt: Android Development report writing

Analysis of Product Design and Implementation

Contents

[1. Introduction 2](#_Toc60719299)

[2. Registration/Login 3](#_Toc60719300)

[3. View Animals 5](#_Toc60719301)

[4. Map 7](#_Toc60719302)

[5. Ticket Master Screen 9](#_Toc60719303)

[6. Database 11](#_Toc60719304)

[7. Future of the Business/App 11](#_Toc60719305)

[8. Conclusion 11](#_Toc60719306)

# Introduction

We are expected as a group to formulate a working program which provides functionality for the idea in which we conceptualized.

The report aims to look at how the functionality is brought about, the appropriate software using the android studio as the platform for development and Kolin as the designated language.

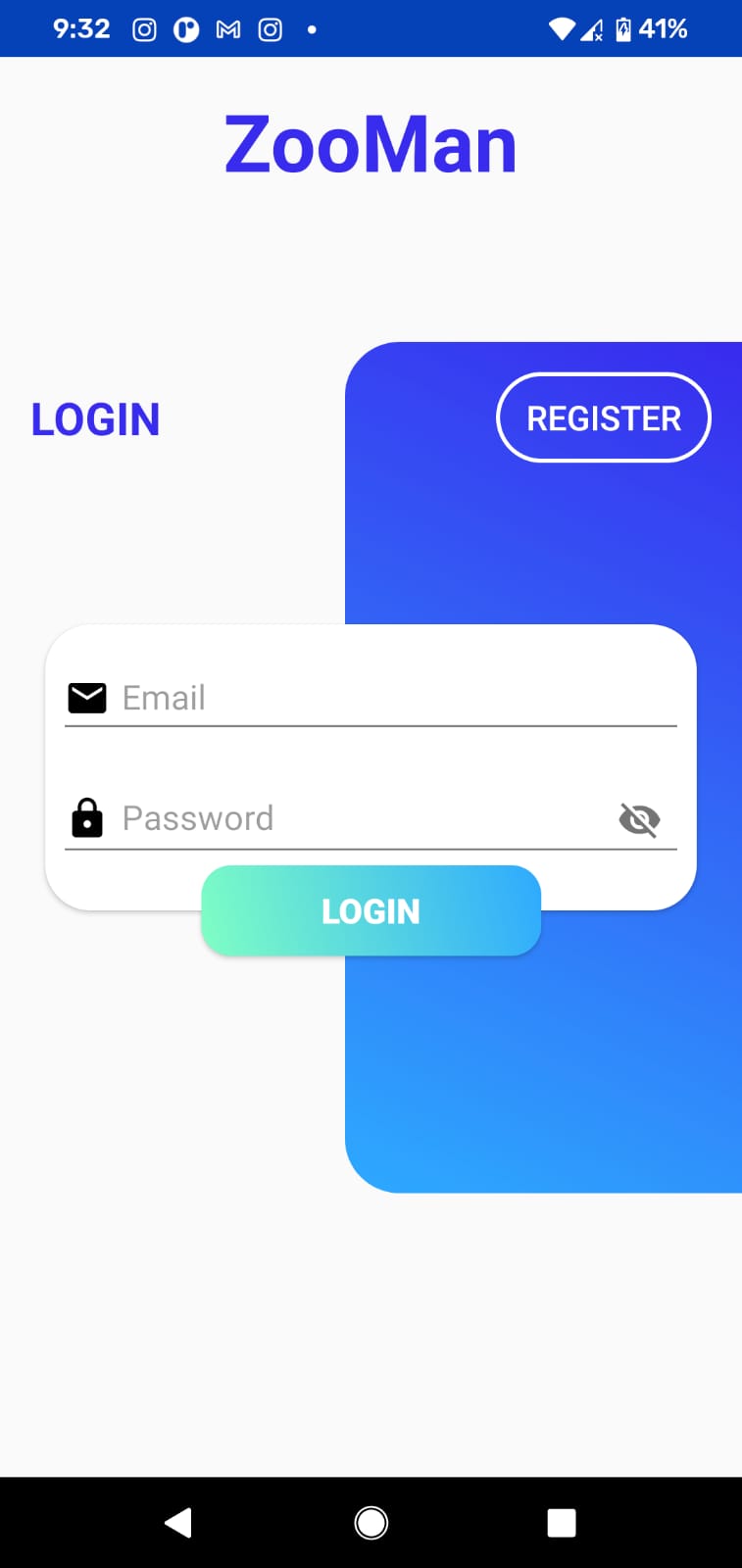
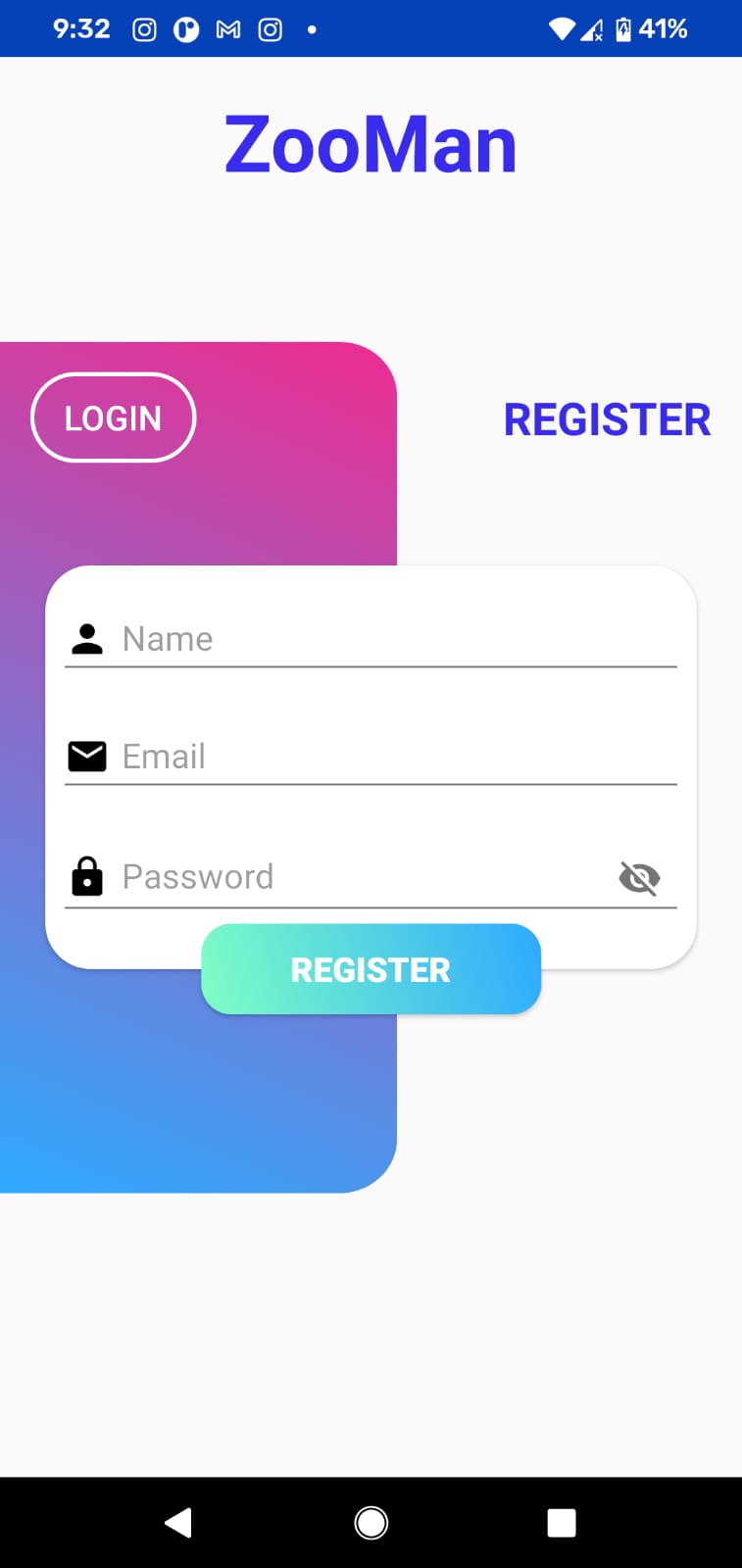
* 1. **Background**

When the project began there was a general lack of kotlin as a programming language among the team. No previous knowledge. Time was dedicated to learning and get an understanding of the capabilities using Kotlin. Mostly knowledge was gained from answers provided in StackOverflow, as well as various other tutorial sites/videos. But as we got more familiar with the language, we knew it was perfect for the application.

* 1. **Project Brief**

The goal of this project was to create an android application, capable of being used on multiple devices. The application allows users to register a new account and login to currently existing accounts. Once logged in, the user can purchase tickets, view animals and use the navigation feature to help find their way throughout the zoo.

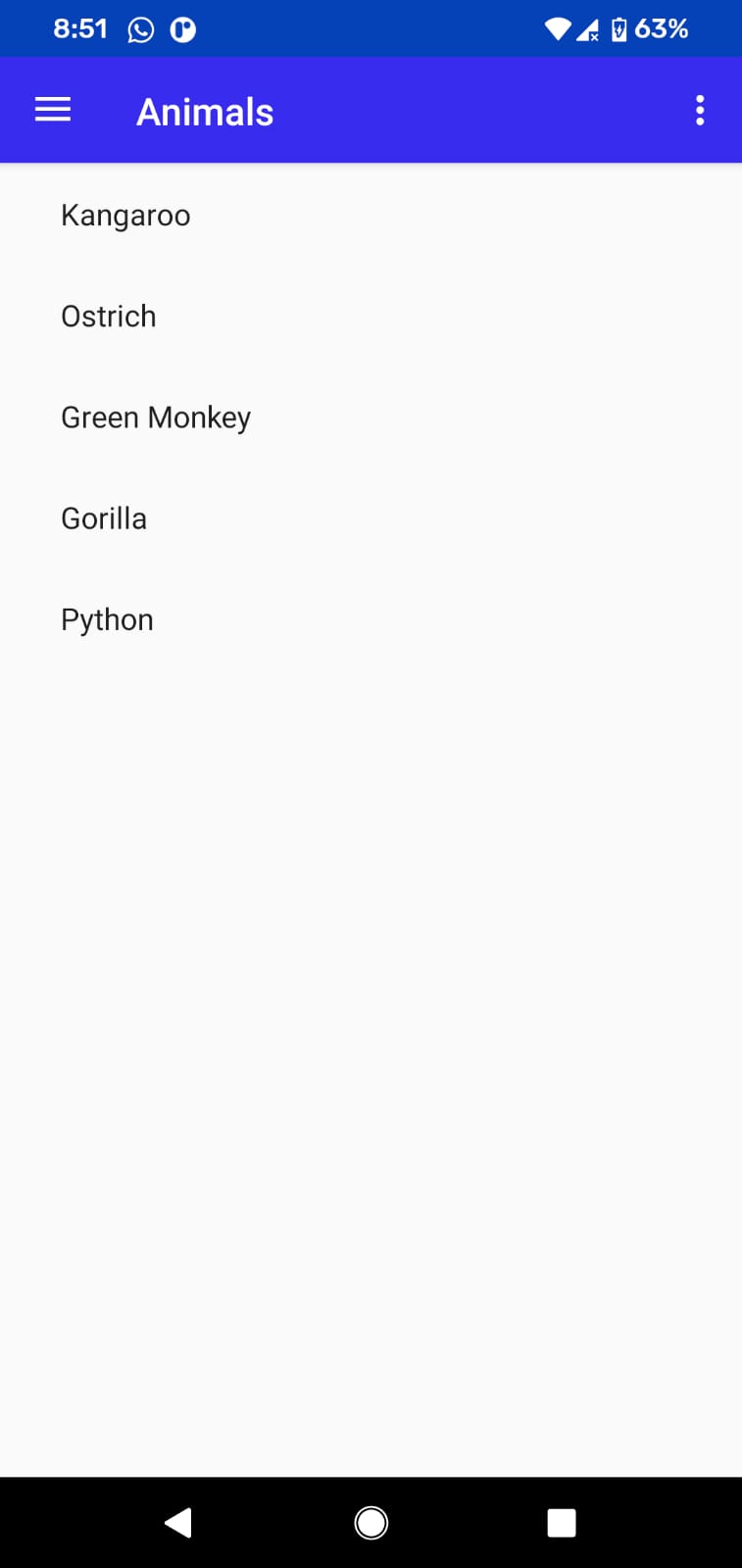
# Registration/Login



These pages have a very simple yet appealing design. There was little to no hardcoding used for these pages. Android studio has a very unique interface which allows design implantation with drag and drops features. Seamless yet effective. Now here is some code to show the creation of the login screen partially.

<TextView  
 android:id="@+id/textView2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="16dp"  
 android:text="ZooMan"  
 android:textColor="@color/colorPrimary"  
 android:textSize="42sp"  
 android:textStyle="bold"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
<View  
 android:id="@+id/view"  
 android:layout\_width="210dp"  
 android:layout\_height="450dp"  
 android:background="@drawable/ic\_login\_bg"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />

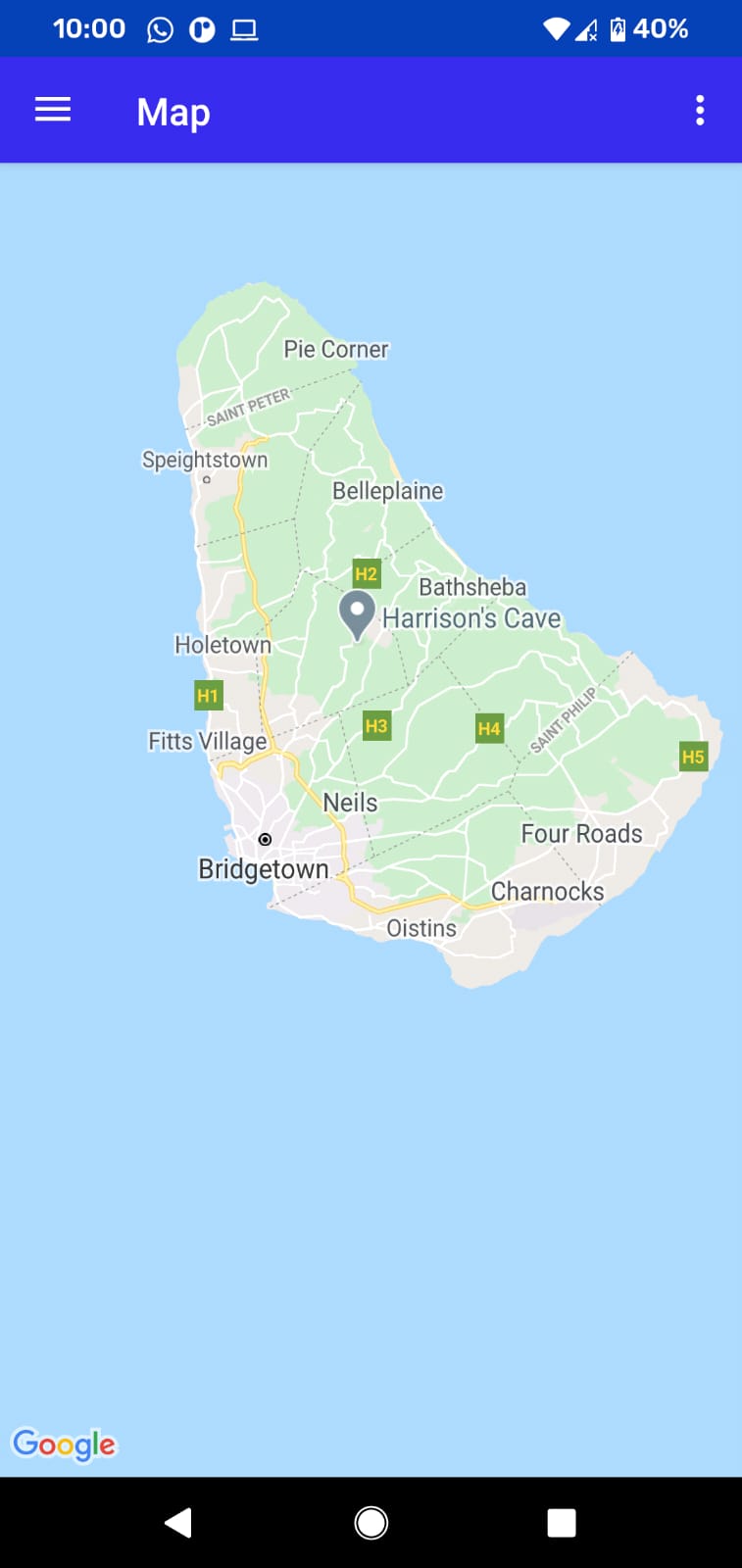
# View Animals



This is Animal View Page. It allows for a user to select the animal(s) they wish to view. It’s seamless and efficient. In the backend multiple animals can be added no matter the category, there is no limit to adding animals other than the physical capacity of the zoo, as you would not add more than can be stored. Here we have partial code showing how animals are being stored within the database and can be view via the app.

class AnimalItemRecyclerViewAdapter(  
 private val values: List<String>  
) : RecyclerView.Adapter<AnimalItemRecyclerViewAdapter.ViewHolder>() {  
  
 override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {  
 val view = LayoutInflater.from(parent.*context*)  
 .inflate(R.layout.*fragment\_animal\_item*, parent, false)  
 return ViewHolder(view)  
 }  
  
 override fun onBindViewHolder(holder: ViewHolder, position: Int) {  
 val item = values[position]  
 holder.idView.*text* = item  
 *//holder.contentView.text = item* }  
  
 override fun getItemCount(): Int = values.size  
  
 inner class ViewHolder(view: View) : RecyclerView.ViewHolder(view) {  
 val idView: TextView = view.findViewById(R.id.*item\_number*)  
 val contentView: TextView = view.findViewById(R.id.*content*)

# Map



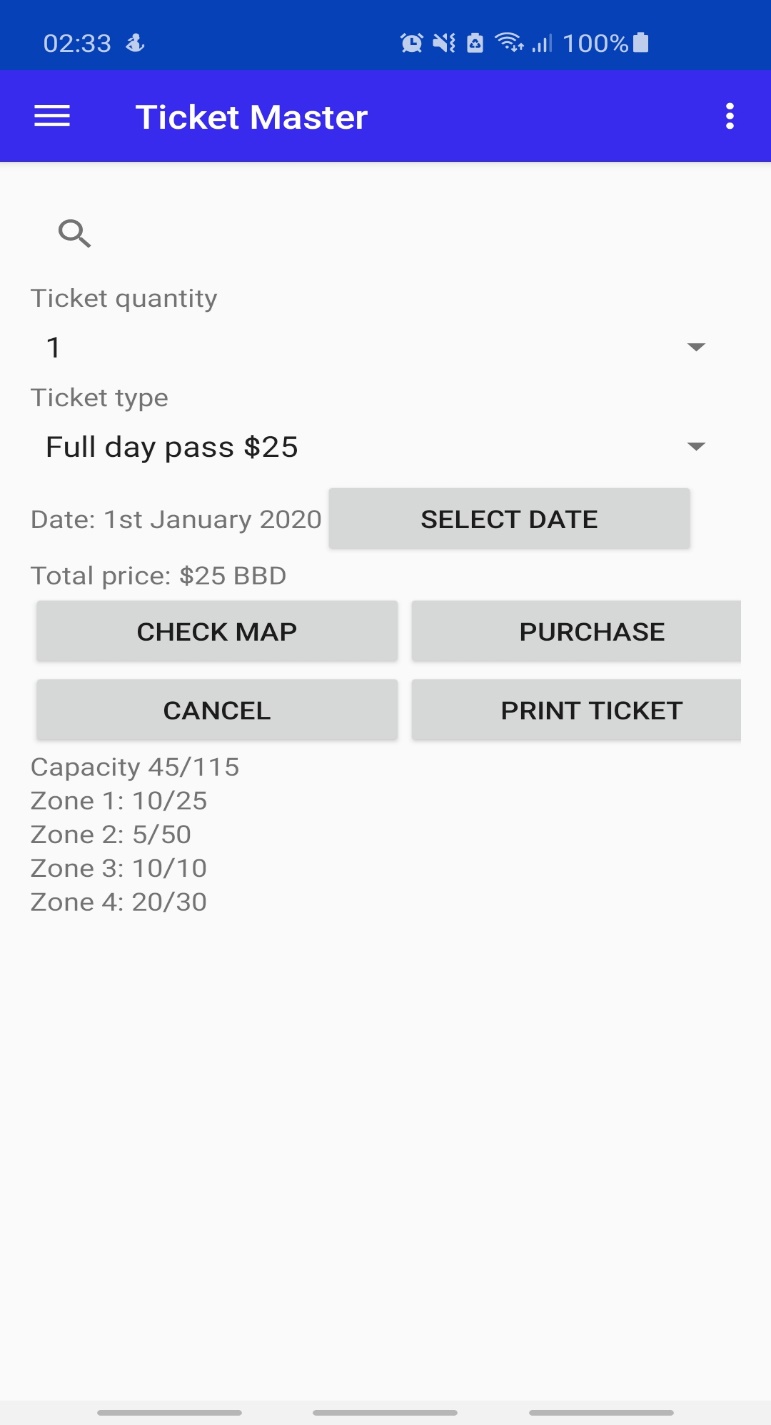
This is powered by google maps. It allows the user to view their current location and navigate throughout the compound.

Below is a sample of the code used to embed the map functionalities within the app:

MapsFragment.kt

package com.swen3.zooman.ui.map  
import androidx.fragment.app.Fragment  
import android.os.Bundle  
import android.view.LayoutInflater  
import android.view.View  
import android.view.ViewGroup  
import com.google.android.gms.maps.CameraUpdateFactory  
import com.google.android.gms.maps.GoogleMap  
import com.google.android.gms.maps.OnMapReadyCallback  
import com.google.android.gms.maps.SupportMapFragment  
import com.google.android.gms.maps.model.LatLng  
import com.google.android.gms.maps.model.MarkerOptions  
import com.swen3.zooman.R  
  
class MapsFragment : Fragment() {  
  
 private val callback = *OnMapReadyCallback* **{** googleMap **->** */\*\*  
 \* Manipulates the map once available.  
 \* This callback is triggered when the map is ready to be used.  
 \* This is where we can add markers or lines, add listeners or move the camera.  
 \* In this case, we just add a marker near Sydney, Australia.  
 \* If Google Play services are not installed on the device, the user will be prompted to  
 \*, install it inside the SupportMapFragment. This method will only be triggered once the  
 \* the user has installed Google Play services and returned to the app.  
 \*/* val sydney = LatLng(-34.0, 151.0)  
 googleMap.addMarker(MarkerOptions().position(sydney).title("Marker in Sydney"))  
 googleMap.moveCamera(CameraUpdateFactory.newLatLng(sydney))  
 **}**

# Ticket Master Screen



On this screen, Ticket Master is a very easy to read, efficient ticket purchase screen where the customer can select the number of tickets he/she would like to order if they require a full day pass or a 2-hour pass and the date(s) they would like each ticket to be available for. The customer would see their subtotal before they checkout and see which zones are filled with occupants within those time frames according to ticket purchases for those date/times. They can choose to print their ticket to be scanned in person upon arrival and check the map to see where zones are located.

Now here is some of the code used to show the current amount of persons within the animal zones at each location.

<TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Capacity 45/115" />  
  
<TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Zone 1: 10/25" />  
  
<TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Zone 2: 5/50" />  
  
<TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Zone 3: 10/10" />  
  
<TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Zone 4: 20/30" />

# Database

Here we have a database that runs locally for now until we get a remote database. It has all the basic requirements needed to run the application for business and its ease of use makes it very efficient for maintenance and upgrades to it.

# Future of the Business/App

For the future of the business, we would like to first expand our reach across the globe, giving off an experience that would be remembered for a lifetime, something like the Disney World effect but for the animal kingdom and its forever expanding hierarchy as new species and subspecies are found and recorded into history.

We would also like to implement a remote database for a more seamless approach to what our true goal for the app and business to feel like as they go hand in hand for the customer’s experience. Using the app while present within the zoo which would allow for real-time tracking of animals, knowing their back story, names they were given, habitats they thrive in or can’t, their potential mates and also their rescue location/breed location.

# Conclusion

We are excited to share with the world our views on the animal kingdom and with the use of technology to bridge the gap of the old school with the new school so kids can have an interactive experience instead of hearing a boring old tour guide and not understanding the true nature of the animal and how sometimes they can think like humans to protect the ones they love or be very social once trust is earned and once it is earned it is hard to be broken as animals tend to never forget the scent or face of their companions.