ILR 03 – Progress Review 2

Dorothy Kirlew

Team Daedalus Members: Mohak Bhardwaj, Shivam Gautam, Pranav Maheshwari, and Richa Varma

October 29, 2015

1. Individual Progress

For the progress review on October 29, I worked with Richa to complete our App Development goal of pairing the app with a laptop via Bluetooth. To better show the interactions between the phone and the laptop, I added two statuses to the top of the screen – one to show whether the message was sent, and one to show that the Bluetooth adaptor successfully connected, as seen in Figure 1: GUI **with status messages**.

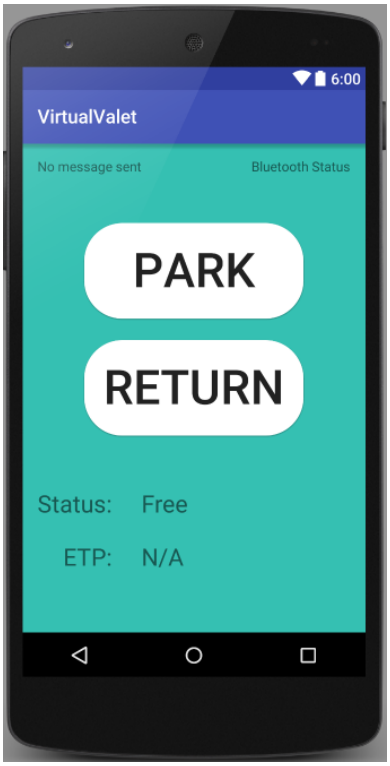

Figure 1: GUI with status messages

Previously, my work with the app had been done completely in XML to design the layout. To add the functionality to the buttons, I had to work with the MainActivity.java file. Currently, the app is in a neutral state when the app is opened; both buttons are enabled, the Status is free and the ETP (estimated time to park) or ETA is N/A, and no message has been sent.

The app will initially try to connect to the Bluetooth adaptor on the phone and, if it can, will display the message "Adaptor found! Yay!" The user can then press “Park” or “Return”. In either case, the app will ask the user for permission to turn on Bluetooth for 300 seconds, as seen in Figure 2 . After selecting “Allow”, the user will be asked which device to which to connect. For our purposes, we chose Richa’s laptop, which we had set to receive a file via Bluetooth.

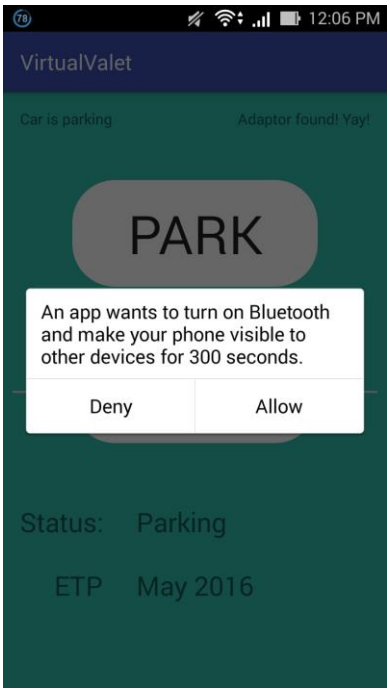

Figure 2: Bluetooth Permission, picture by Richa Varma

Upon connecting to a device, the app with send it a file called Message.txt, which contains "Park" or "Return", as appropriate. The app will then change the message status to "Car is parking" or "Car is returning". The status will change to "Parking" or "Returning" and the time status will change to "ETP: May 2016" or "ETA: One month", as appropriate. Figure 3 shows the App when "Park" is pressed the first time.

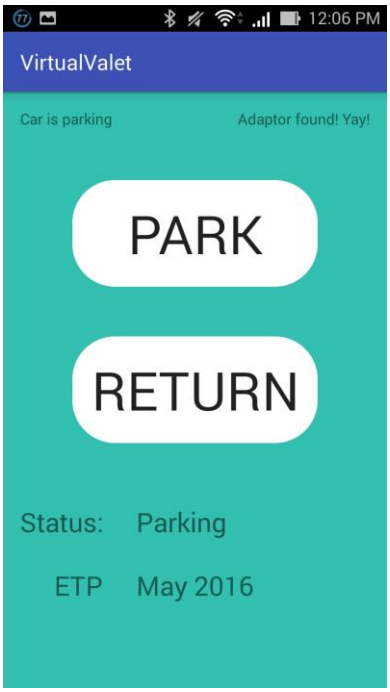

Figure 3: First "Park" Press

The user can then choose to press the same button or the other button. If they press the same button again, the app will not sent Message.txt, as it has already been sent. However, the message status will change to "Car is still parking, calm down", as seen in Figure 4.

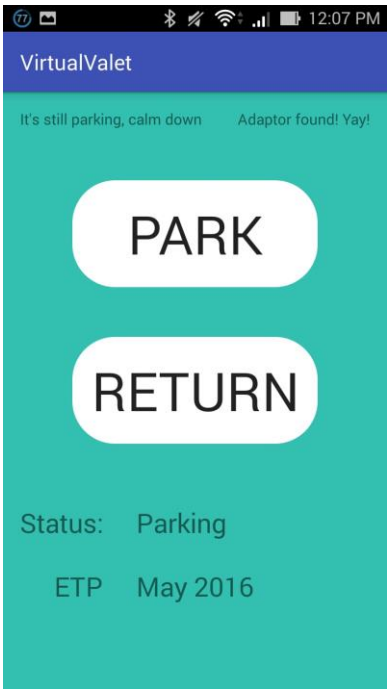

Figure 4: Second "Park" Press

If the user presses the same button a third time, the original button will be disabled. The user will only be able to press the remaining button. For example, if the user presses “Park” three times in a row, the Park button will be disabled and they will only be able to press the Return button, as seen in Figure 5. After pressing the Return button once, they will again be able to press the Park button.

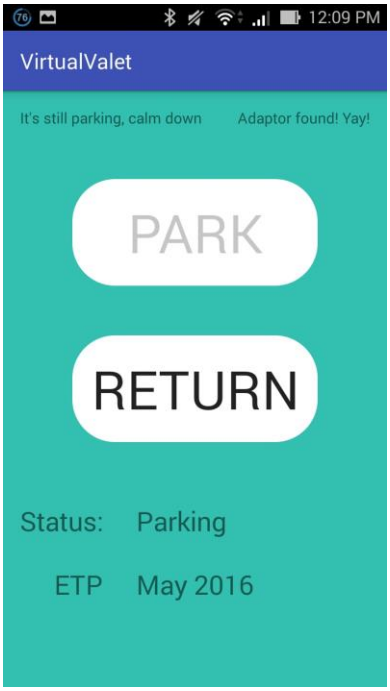

Figure 5: Final “Park” Press

In the final stages of the app, the user will only be able to press the button once before it becomes disabled. These extra messages are simply to test the capabilities of the app interface.

2. Challenges

Both interfacing the app with the Bluetooth adaptor and sending a file from the app to a connected device was quite a challenge. Thankfully, Android Studio is very well supported and there was plenty of documentation to assist Richa and me in doing this.

It has been quite some time since I have worked with Java, and it took me a while to get used to the language again, in addition to learning how to use the Java file to interface with the app.

3. Teamwork

We divided the work amongst our team members as evenly as possible, with respect to our experience and skills. Shivam interfaced the Arduino Mega and ROS with Pranav, began working with the Kinect with Mohak, ordered the communication equipment, and decided on the Oculus Rift as our mobile platform. Mohak worked on interfacing the Kinect with ROS, preliminary testing of Kinect and ROS, and interfacing ROS and the Bluetooth Serial Interface. Pranav interfaced the Arduino Mega and ROS with Shivam and interfaced Bluetooth serial communication and ROS with Mohak. Richa studied Android Studio and worked with me to set up the Bluetooth connection between the app and a laptop.

4. Plans

The team has four goals to complete by Progress Review 3:

1. **Communication Subsystem**:
    a. Setup and test the mesh network using the DigiMesh networking components (Shivam and Pranav)
2. **Mobile Application Subsystem**:
    a. Integrate the mobile app with the ROS node to communicate data (Dorothy and Mohak)
    b. Send serial messages via Bluetooth (Dorothy and Richa)
3. **Vision Subsystem**:
    a. Show progress regarding obstacle detection algorithm implementation using Kinect (Shivam, Pranav, Mohak)
4. **Mobile Platform**:
    a. Acquire Oculus prime and integrate it with an SBC. (Pranav and Dorothy)

5. Code

a. activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
    android:layout_height="match_parent"
android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
tools:context=".MainActivity"
    android:background="#35c0b2">

    <Button
        android:layout_width="230dp"
        android:layout_height="100dp"
        android:textSize="50sp"
        android:background = "@drawable/unpressed_button"
        android:text="Park"
        android:id="@+id/parkButton"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="67dp"
        android:clickable="true"
        android:onClick="parkCar"/>

    <Button
        android:layout_width="230dp"
        android:layout_height="100dp"
        android:textSize="50sp"
        android:background = "@drawable/pressed_button"
        android:text="Return"
```

```
        android:id="@+id/returnButton"
        android:enabled="true"
        android:layout_centerVertical="true"
        android:layout_alignStart="@+id/parkButton"
        android:clickable="true"
        android:onClick="returnCar"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="25sp"
        android:text="Status:"
        android:id="@+id/justStatusLabel"
        android:layout_marginTop="57dp"
        android:layout_below="@+id/returnButton"
        android:layout_alignParentStart="true" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="25sp"
        android:text="Free"
        android:id="@+id/statusText"
        android:layout_marginStart="31dp"
        android:layout_alignTop="@+id/justStatusLabel"
        android:layout_toEndOf="@+id/justStatusLabel" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="25sp"
        android:text="ETP:"
        android:id="@+id/justETALabel"
        android:layout_below="@+id/justStatusLabel"
        android:layout_toStartOf="@+id/statusText"
        android:layout_marginTop="26dp" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="25sp"
        android:text="N/A"
        android:id="@+id/ETText"
        android:layout_alignTop="@+id/justETALabel"
        android:layout_alignStart="@+id/statusText" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="No message sent"
        android:id="@+id/messageTextView"
        android:layout_alignParentTop="true"
        android:layout_alignParentStart="true" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Bluetooth Status"
        android:id="@+id/bluetoothStatusText"
        android:layout_alignParentTop="true"
```

```
        android:layout_alignParentEnd="true" />

</RelativeLayout>
```

b. MainActivity.java

```
package teami.virtualvalet;

import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothServerSocket;
import android.bluetooth.BluetoothSocket;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.content.pm.ResolveInfo;
import android.net.Uri;
import android.os.Environment;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.util.List;

public class MainActivity extends AppCompatActivity {
    String message = "";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        BluetoothAdapter btAdapter = BluetoothAdapter.getDefaultAdapter();
        TextView bluetoothStatus = (TextView)
findViewById(R.id.bluetoothStatusText);

        if(btAdapter == null)
        {
            bluetoothStatus.setText("Adaptor not found");
        }
        else
        {
            bluetoothStatus.setText("Adaptor found! Yay!");
        }
    }

    public void parkCar(View view)
    {
        TextView messageToUser = (TextView)
findViewById(R.id.messageTextView);
        TextView status = (TextView) findViewById(R.id.statusText);
        TextView ETA = (TextView) findViewById(R.id.ETText);
        TextView ETALabel = (TextView) findViewById(R.id.justETALabel);
        Button parkButton = (Button) findViewById(R.id.parkButton);
        Button returnButton = (Button) findViewById(R.id.returnButton);

        if (messageToUser.getText().equals("No message sent")||
```

```
messageToUser.getText().equals("Car is returning") ||
messageToUser.getText().equals("It's still returning, calm down"))
        {
            messageToUser.setText("Car is parking");
            status.setText("Parking");
            ETA.setText("May 2016");
            ETALabel.setText("ETP");
            returnButton.setEnabled(true);
            message = "Park";
            enableBluetooth();
        }
        else if (messageToUser.getText().equals("Car is parking"))
        {
            messageToUser.setText("It's still parking, calm down");
        }
        else if (messageToUser.getText().equals("It's still parking, calm
down"))
        {
            parkButton.setEnabled(false);
        }
    }

    public void returnCar(View view)
    {
        TextView messageToUser = (TextView)
findViewById(R.id.messageTextView);
        TextView status = (TextView) findViewById(R.id.statusText);
        TextView ETA = (TextView) findViewById(R.id.ETText);
        TextView ETALabel = (TextView) findViewById(R.id.justETALabel);
        Button parkButton = (Button) findViewById(R.id.parkButton);
        Button returnButton = (Button) findViewById(R.id.returnButton);

        if (messageToUser.getText().equals("No message sent")||
messageToUser.getText().equals("Car is parking") ||
messageToUser.getText().equals("It's still parking, calm down"))
        {
            messageToUser.setText("Car is returning");
            status.setText("Returning");
            ETA.setText("One Month");
            ETALabel.setText("ETA");
            parkButton.setEnabled(true);
            message = "Return";
            enableBluetooth();
        }
        else if (messageToUser.getText().equals("Car is returning"))
        {
            messageToUser.setText("It's still returning, calm down");
        }
        else if (messageToUser.getText().equals("It's still returning, calm
down"))
        {
            returnButton.setEnabled(false);
        }
    }

    public void enableBluetooth()
    {
        Intent discoveryIntent = new
Intent(BluetoothAdapter.ACTION_REQUEST_DISCOVERABLE);
```

```
        discoveryIntent.putExtra(BluetoothAdapter.EXTRA_DISCOVERABLE_DURATION,
300);

        startActivityForResult(discoveryIntent, 1);
    }

    protected void onActivityResult(int requestCode, int resultCode, Intent
data)
    {
        if(resultCode == 300 && requestCode == 1)
        {
            Intent intent = new Intent();
            intent.setAction(Intent.ACTION_SEND);
            intent.setType("text/plain");
            try {

                File root = new
File(Environment.getExternalStorageDirectory(), "Notes");
                if (!root.exists()) {
                    root.mkdir();
                }
                File gpxFile = new File(root, "Message.txt");

                FileWriter writer = new FileWriter(gpxFile);
                writer.append(message);
                writer.flush();
                writer.close();

                intent.putExtra(Intent.EXTRA_STREAM, Uri.fromFile(gpxFile));

                Toast.makeText(this, "Sent!", Toast.LENGTH_SHORT).show();
            }
            catch (IOException e)
            {
                e.printStackTrace();
            }

            PackageManager pm = getPackageManager();
            List <ResolveInfo> appsList = pm.queryIntentActivities( intent,
0);
            if(appsList.size() > 0)
            {
                String packageName = null;
                String className = null;
                boolean found = false;

                for(ResolveInfo info: appsList)
                {
                    packageName = info.activityInfo.packageName;
                    if(packageName.equals("com.android.bluetooth"))
                    {
                        className = info.activityInfo.name;
                        found = true;
                        break;
                    }
                }

                if(!found)
                {
                    TextView bluetoothStatus = (TextView)
```

```
findViewById(R.id.bluetoothStatusText);
                    bluetoothStatus.setText("Bluetooth not found!!");
                }
                else
                {
                    intent.setClassName(packageName, className);
                    startActivity(intent);
                }
            }

            else
            {
                TextView bluetoothStatus = (TextView)
findViewById(R.id.bluetoothStatusText);
                bluetoothStatus.setText("Bluetooth is cancelled.");
            }
        }
    }
}
```

c. button_states.xml

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

    <item android:state_pressed="false"
        android:drawable="@drawable/unpressed_button"/>

    <item android:state_pressed="true"
        android:drawable="@drawable/pressed_button"/>

</selector>
```

d. pressed_button.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
    <corners android:radius="40dp"/>
    <gradient android:startColor="#FFFFFF" android:endColor="#FFFFFF"  />
</shape>
```

e. unpressed_button.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
    <corners android:radius="40dp"/>
    <gradient android:startColor="#FFFFFF" android:endColor="#FFFFFF"  />
</shape>
```