Wireless Control of a Ground Vehicle through Internet and Gestures

ASSIGNMENT 1

SOFTWARE FOR EMBEDDED SYSTEMS

Vaibhav Gupta
2011A8PS255P

Tarang Shah
2011A8PS279P

PROBLEM STATEMENT

- To develop a ground vehicle which is capable of doing simple function like move Forward, Back, Left Turn, Right Turn and Stop. The vehicle will be wirelessly controlled using two technologies – Through Internet and through gestures. An embedded system will be used to perform these functions.
- For an internet controlled vehicle - a webpage will be developed which can be opened on the clients computer. It contains buttons which can be pressed by the user to send commands. When the user presses a button on the webpage, it will send the command to a webserver . The system -Rasberry-Pi which is connected to internet, will check for new commands on the server. If a new command is found, the system will execute it on the vehicle.
- In gesture control an android device is used. The device will use its accelerometer and Bluetooth capabilities. The user will tilt the android device which will send the commands over Bluetooth. The system will then execute the command. A switch will be provided on the bot to choose control configuration

PROBLEM DESCRIPTION

- Ground Vehicle
  - Interfacing with the motors
  - Receiver for Internet Commands and Bluetooth Commands
- Web Page and Web Server
  - Web page to Give commands to the Robot – Forward, Backward, Left, Right, Stop
  - Web Server will store the commands and Send them to the Robot
- Bluetooth Application (Android App)
  - Will contain Code to Interface with Bluetooth on phone
  - Interface with phone accelerometer
  - Detect Tilt Gestures and send corresponding command

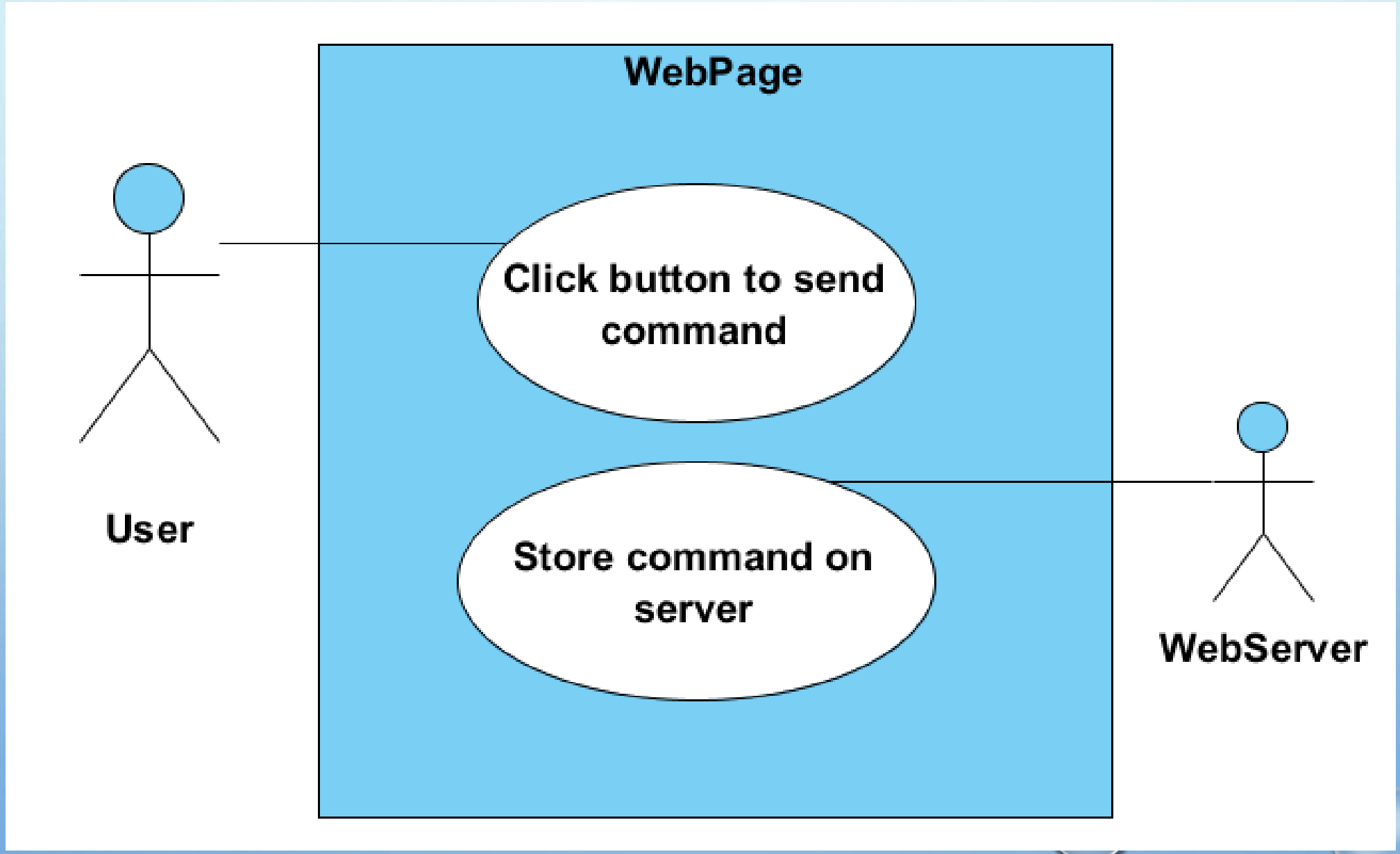
SYSTEM DESCRIPTION

- HARDWARE
  - Raspberry Pi interfaced with motors
  - Wireless Dongle to connect Raspberry pi to Internet
  - Android Phone with Bluetooth and Accelerometer
- SOFTWARE
  - Java Application on Raspberry Pi
  - Javascript based webpage for Internet Control
  - Android Application for Gesture control

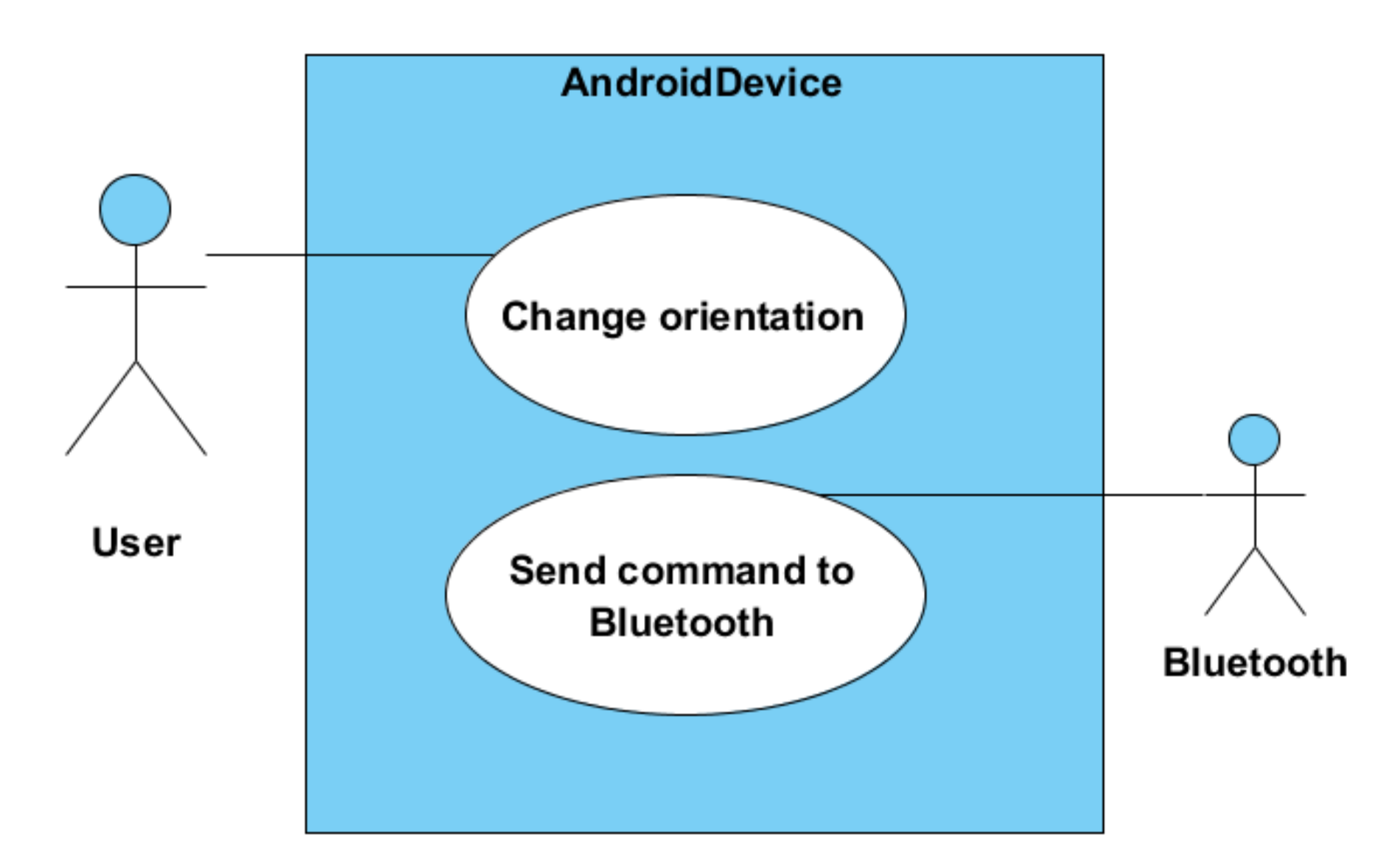
Use Cases

- Webpage Use case
- Android Device Use case
- Ground vehicle System Use Case

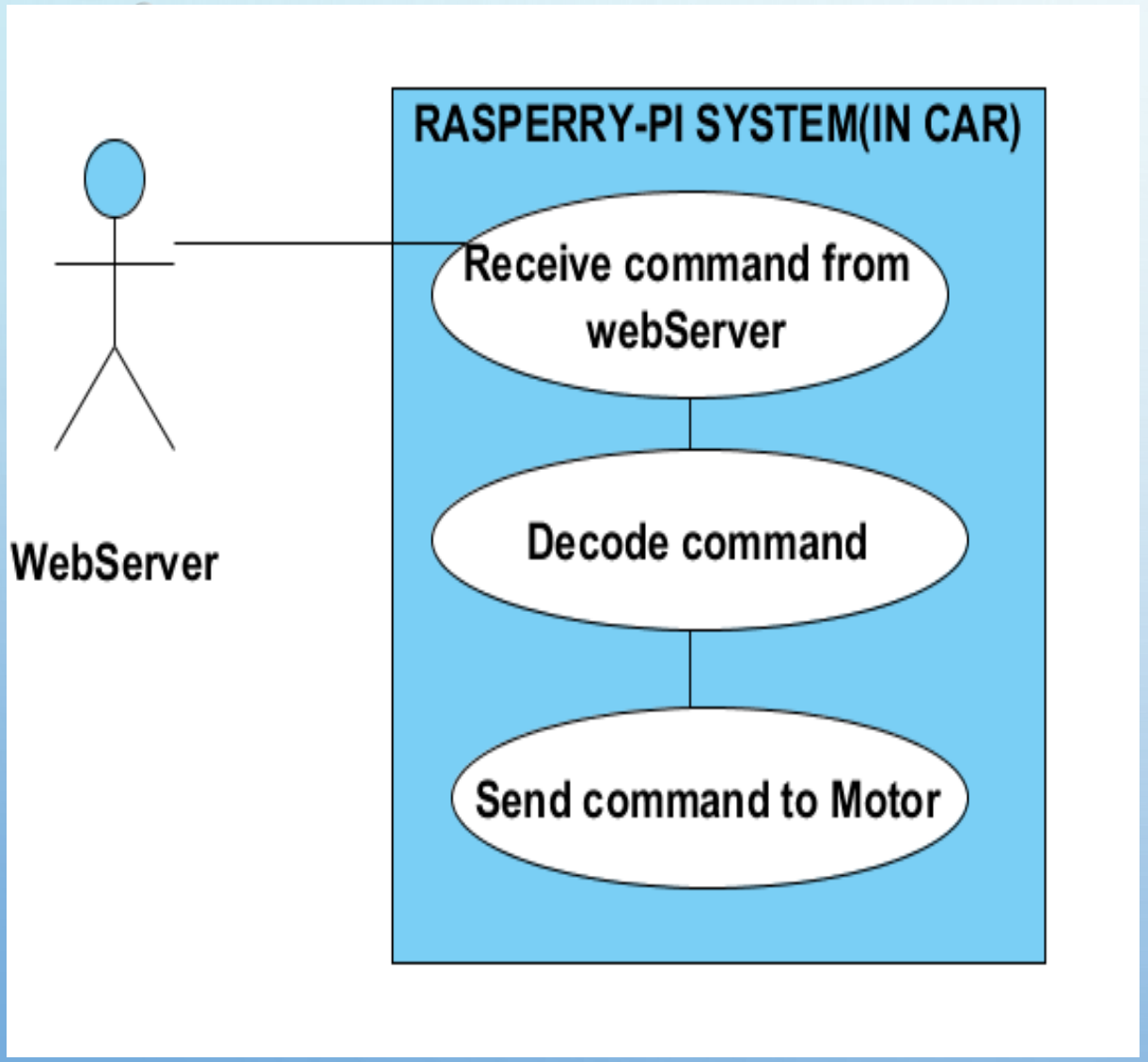
Webpage Use case

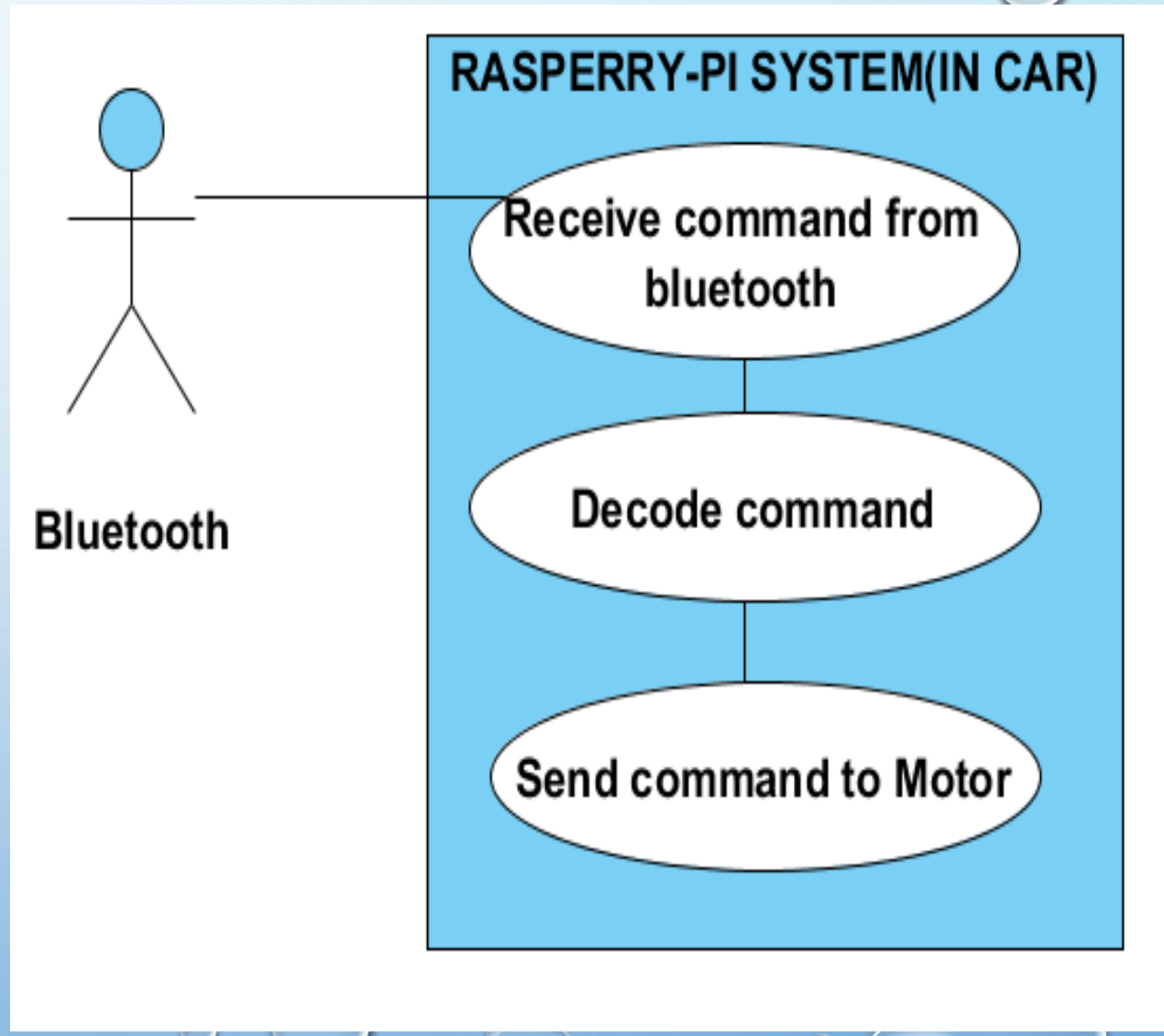


Android Device Use Case



Ground vehicle Use Cases

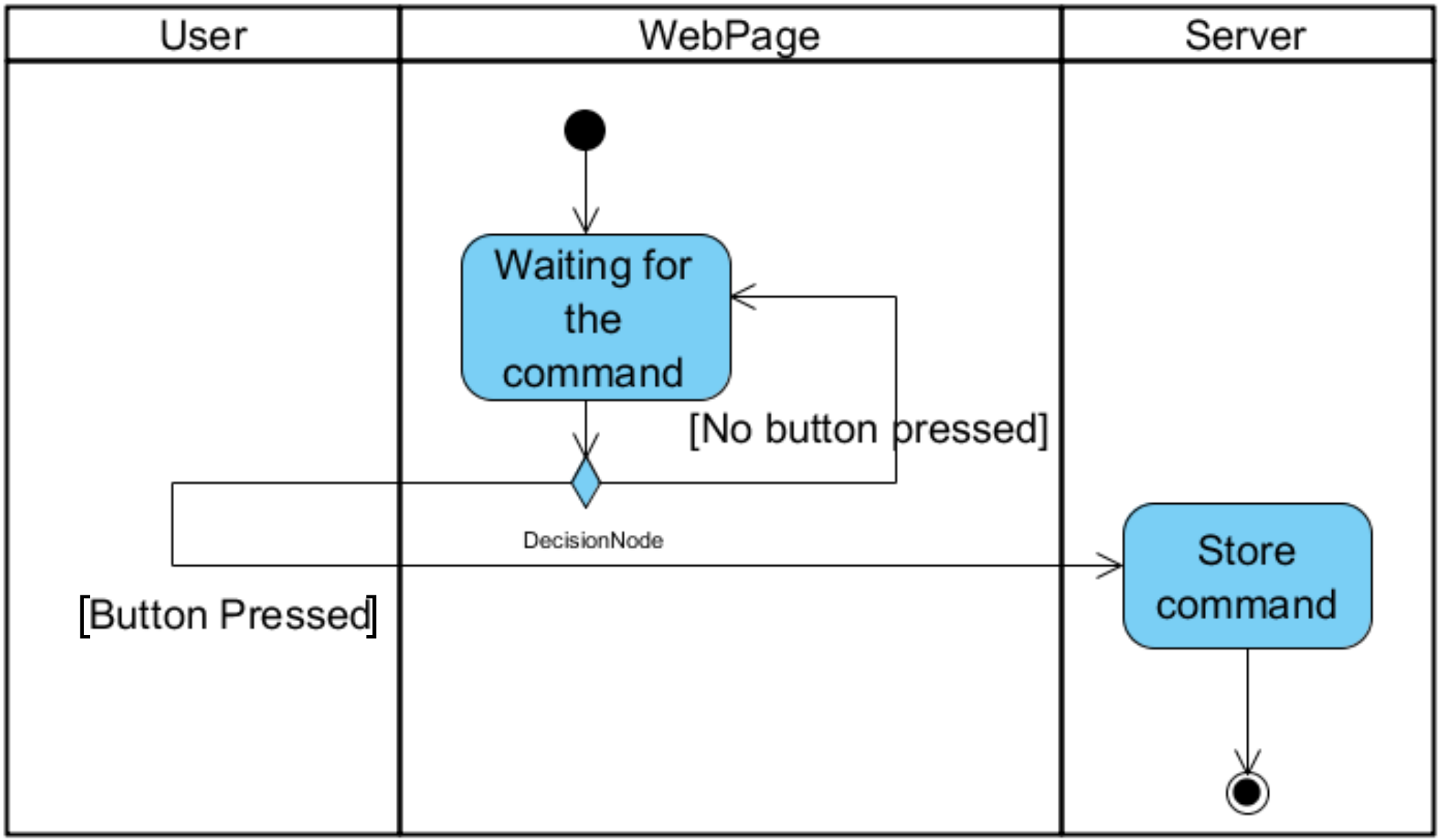




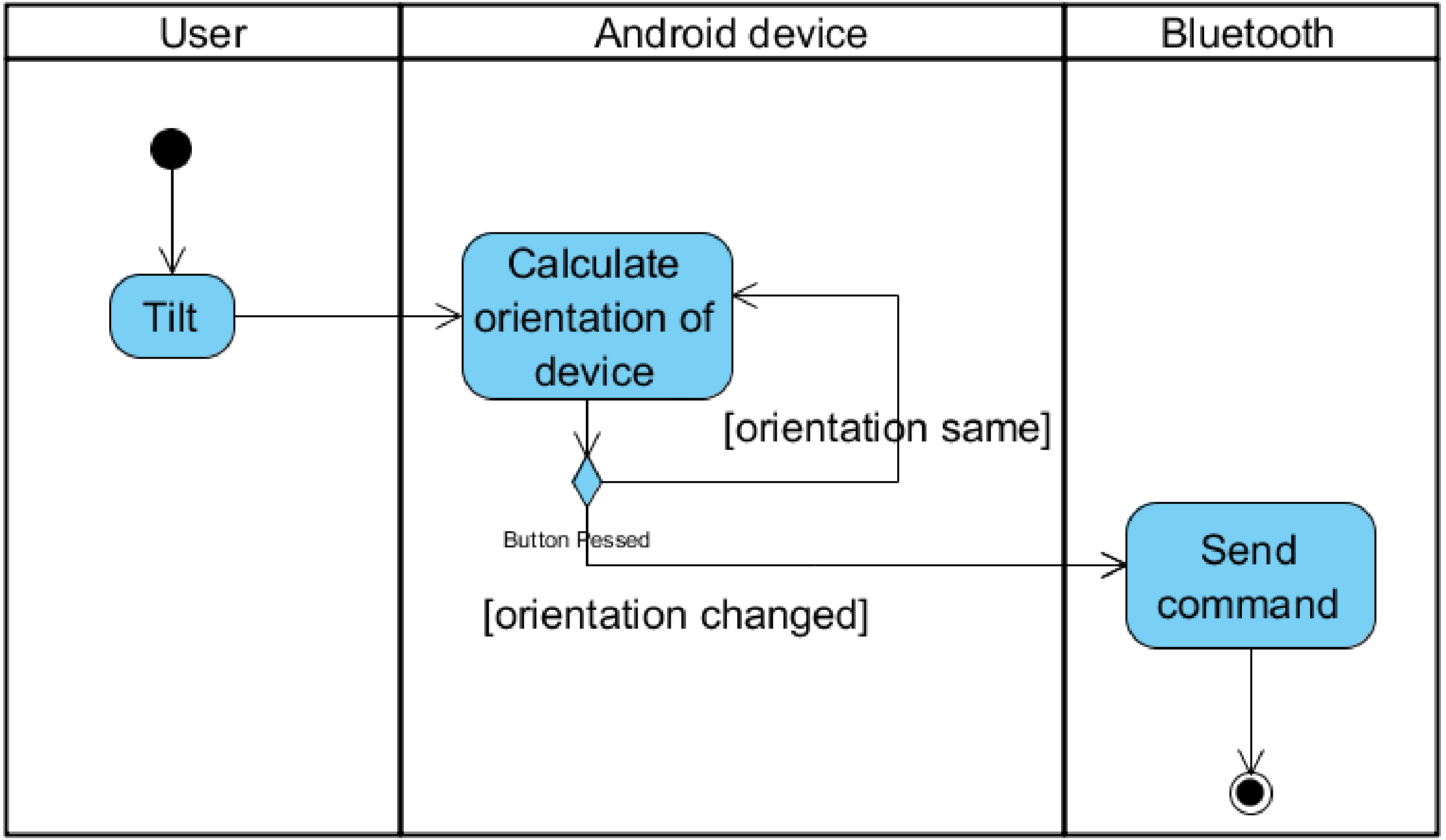
Activity Diagrams

- Webpage
- Android Device
- Ground Vehicle
  - Internet Control
  - Gesture Control

Webpage
Activity Diagram

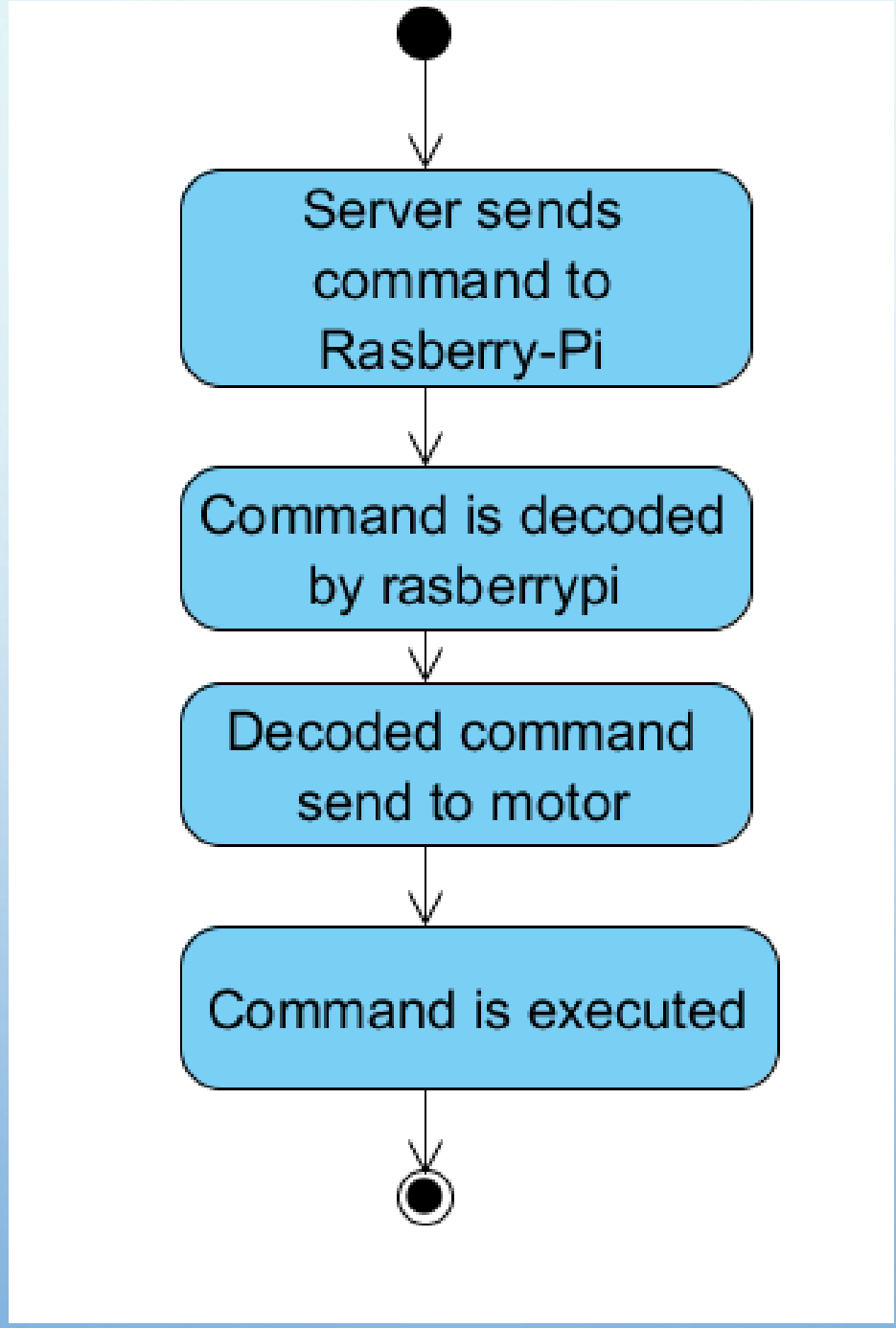

User
WebPage
Server
Waiting for the command
[No button pressed]
DecisionNode
[Button Pressed]
Store command

Android Device Activity Diagram


User
Android device
Bluetooth
Tilt
Calculate orientation of device
[orientation same]
Button Pressed
[orientation changed]
Send command

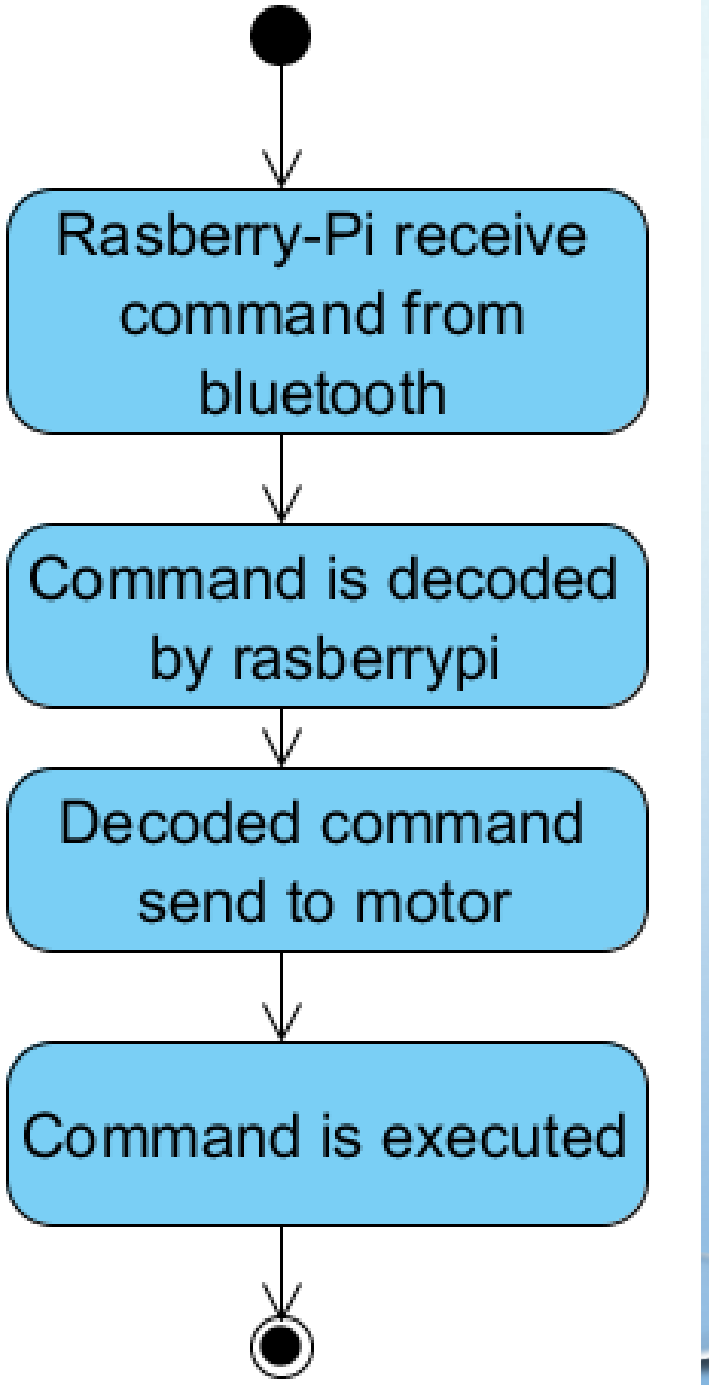
Ground Vehicle

Internet Control Activity Diagram


Server sends command to Rasberry-Pi
Command is decoded by rasberrypi
Decoded command send to motor
Command is executed

Ground Vehicle

Gesture Control Activity Diagram

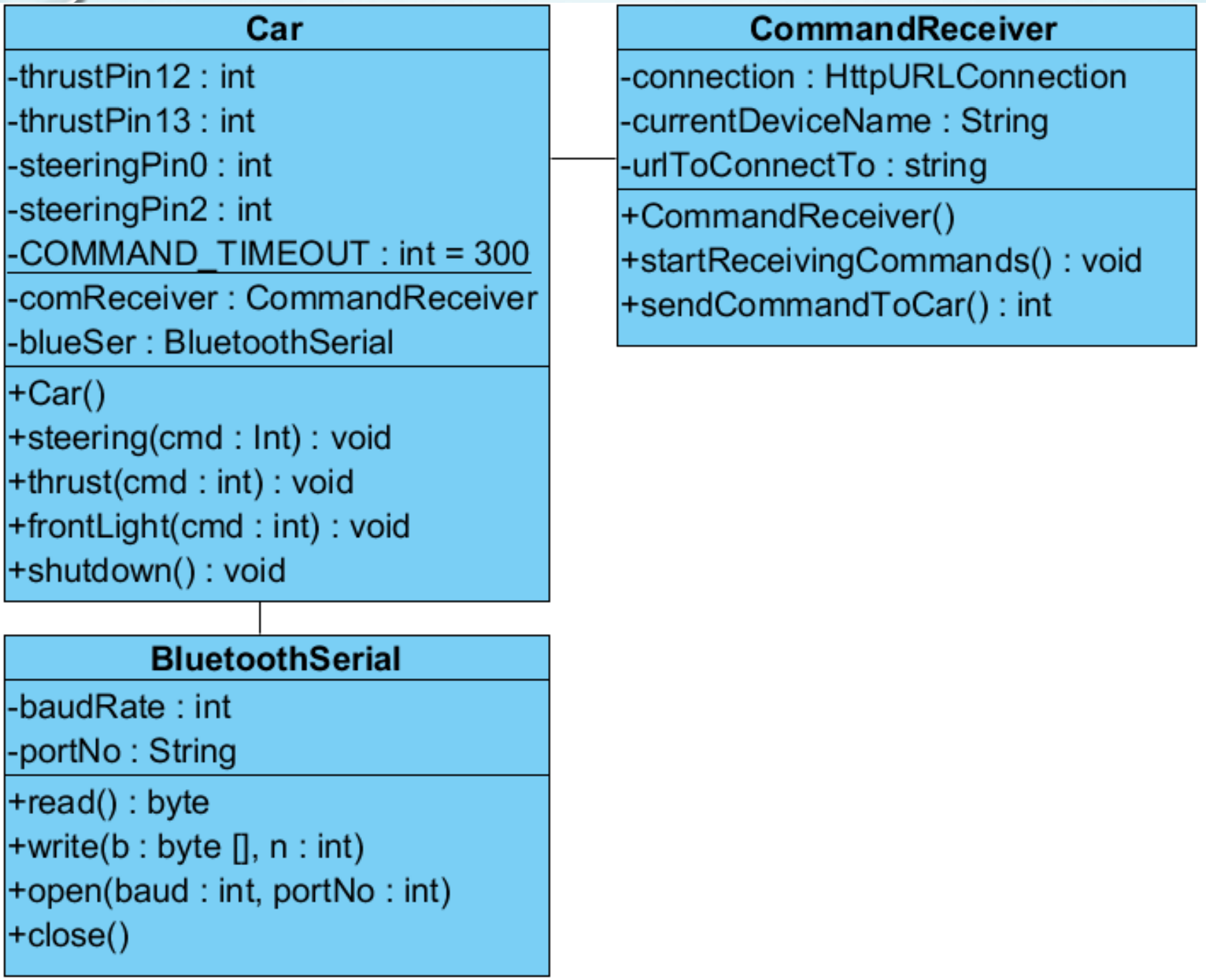

Rasberry-Pi receive command from bluetooth
Command is decoded by rasberrypi
Decoded command send to motor
Command is executed

Class Diagrams

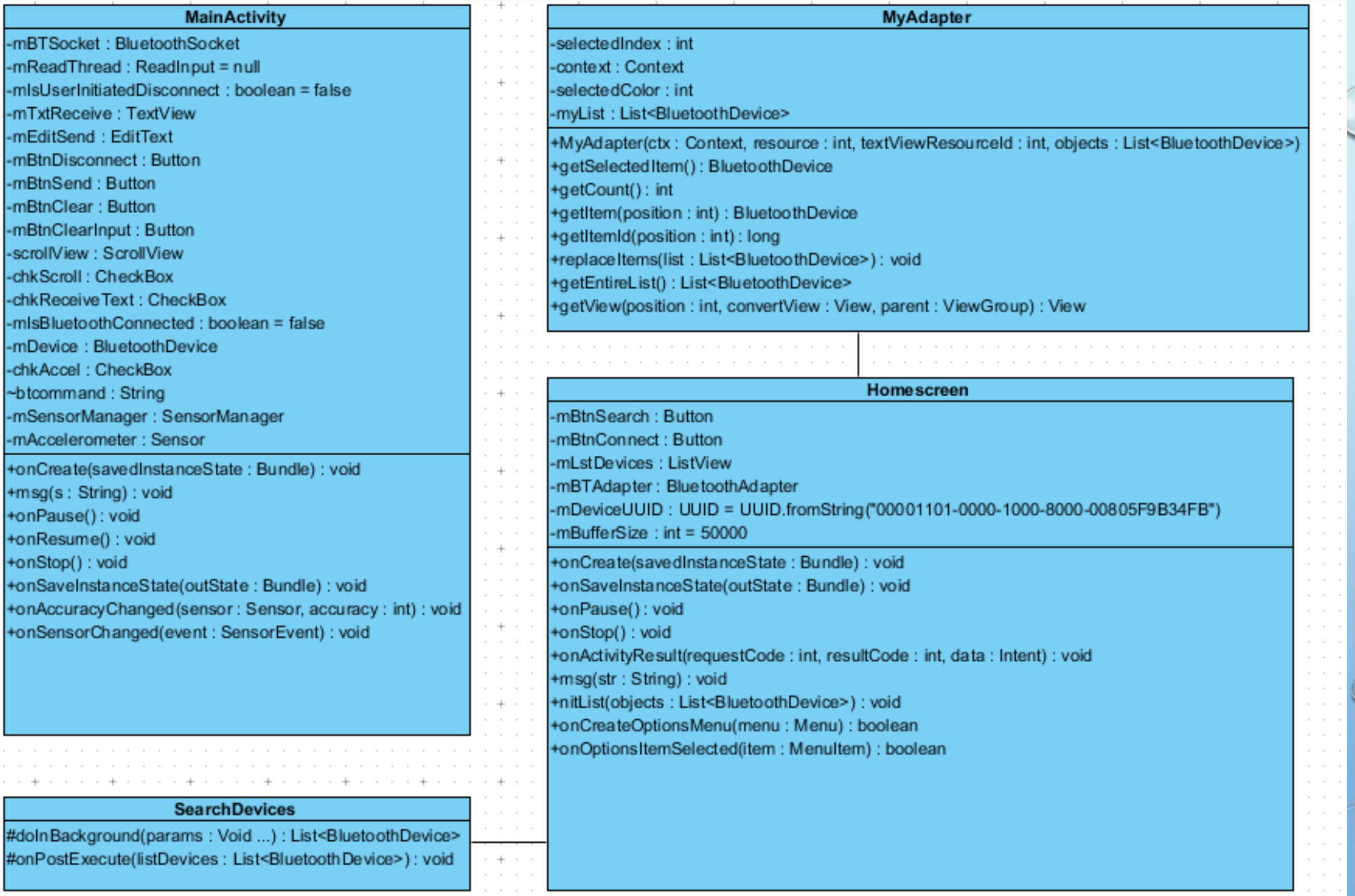
- Ground vehicle System
- Bluetooth Controller App

- ONLY CLASSES THAT WE ARE GOING TO IMPLEMENT HAVE BEEN MENTIONED. STANDARD CLASSES FROM LIBRARIES HAVE NOT BEEN INCLUDED TO MAKE THE DIAGRAMS CLUTTER FREE.

Ground Vehicle System


Car
-thrustPin12 : int
-thrustPin13 : int
-steeringPin0 : int
-steeringPin2 : int
-COMMAND_TIMEOUT : int = 300
-comReceiver : CommandReceiver
-blueSer : BluetoothSerial
+Car()
+steering(cmd : Int) : void
+thrust(cmd : int) : void
+frontLight(cmd : int) : void
+shutdown() : void
CommandReceiver
-connection : HttpURLConnection
-currentDeviceName : String
-urlToConnectTo : string
+CommandReceiver()
+startReceivingCommands() : void
+sendCommandToCar() : int
BluetoothSerial
-baudRate : int
-portNo : String
+read() : byte
+write(b : byte [], n : int)
+open(baud : int, portNo : int)
+close()

Bluetooth Controller App


MainActivity
-mBTSocket : BluetoothSocket
-mReadThread : ReadInput = null
-mIsUserInitiatedDisconnect : boolean = false
-mTxtReceive : TextView
-mEditSend : EditText
-mBtnDisconnect : Button
-mBtnSend : Button
-mBtnClear : Button
-mBtnClearInput : Button
-scrollView : ScrollView
-chkScroll : CheckBox
-chkReceiveText : CheckBox
-mIsBluetoothConnected : boolean = false
-mDevice : BluetoothDevice
-chkAccel : CheckBox
~btcommand : String
-mSensorManager : SensorManager
-mAccelerometer : Sensor
+onCreate(savedInstanceState : Bundle) : void
+msg(s : String) : void
+onPause() : void
+onResume() : void
+onStop() : void
+onSaveInstanceState(outState : Bundle) : void
+onAccuracyChanged(sensor : Sensor, accuracy : int) : void
+onSensorChanged(event : SensorEvent) : void
MyAdapter
-selectedIndex : int
-context : Context
-selectedColor : int
-myList : List<BluetoothDevice>
+MyAdapter(ctx : Context, resource : int, textViewResourceId : int, objects : List<BluetoothDevice>)
+getSelectedItem() : BluetoothDevice
+getCount() : int
+getItem(position : int) : BluetoothDevice
+getItemId(position : int) : long
+replaceItems(list : List<BluetoothDevice>) : void
+getEntireList() : List<BluetoothDevice>
+getView(position : int, convertView : View, parent : ViewGroup) : View
Homescreen
-mBtnSearch : Button
-mBtnConnect : Button
-mLstDevices : ListView
-mBTAdapter : BluetoothAdapter
-mDeviceUUID : UUID = UUID.fromString("00001101-0000-1000-8000-00805F9B34FB")
-mBufferSize : int = 50000
+onCreate(savedInstanceState : Bundle) : void
+onSaveInstanceState(outState : Bundle) : void
+onPause() : void
+onStop() : void
+onActivityResult(requestCode : int, resultCode : int, data : Intent) : void
+msg(str : String) : void
+nitList(objects : List<BluetoothDevice>) : void
+onCreateOptionsMenu(menu : Menu) : boolean
+onOptionsItemSelected(item : MenuItem) : boolean
SearchDevices
#doInBackground(params : Void ...) : List<BluetoothDevice>
#onPostExecute(listDevices : List<BluetoothDevice>) : void

STAGES OF DEVELOPMENT

Stage1

- Developing Android Application & implementing class diagrams
- Testing and debugging the gesture control

Stage 2

- Developing the hardware
- Developing the software for controlling the hardware.

Stage 3

- Developing the webpage and web server
- Testing the data transfer through internet

Stage 4

- Final Integration of all components

APPLICATIONS AND FURTHER DEVELOPMENTS

- POSSIBLE ADDITION OF CAMERA
  - HELP IN REMOTE MONITORING OF HAZARDOUS AREAS
  - USER CAN SEE THE LIVE IMAGES ON A WEBPAGE AND CONTROL THE BOT AT THE SAME TIME
- ROBOTIC TROLLEY FOR TRANSPORTING MATERIAL FROM ONE PLACE TO ANOTHER
- WEB BASED CONTROL SYSTEM CAN BE USED TO CONTROL VARIOUS EMBEDDED DEVICES OVER INTERNET.
- HOME AUTOMATION