COYOTE
QUEST

CoyoteQuest Mobile Navigation App

Eugene Kim (Product Manager/Software Engineer)
Gan Liu (Assistant Product Manager/Software Engineer)
Nicholas Perez (Software Engineer)
Jasmine Pena (Software Engineer)
Daniel Martinez (Software Engineer)

Software Architecture & Design

1. Purpose

The purpose of this document is to provide management and software engineers a blueprint and detailed understanding of how CoyoteQuest executes. It provides a high-level design perspective utilizing a UML component diagram to outline the major front and backend components and their relationships. In addition, a low-level design perspective utilizing a UML class diagram shows in more detail how the major classes, and their corresponding data members and functions, all work together to carry out the features provided in the app. These design perspectives provide a clearer understanding of the overall system, allowing an efficient way to implement in new code and changes to already existing code enabling future teams to maintain this software more effectively.

## 2. High-Level UML Component Diagram

3. Low-Level UML Class Diagram

+ MainActivity extends AppCompatActivity
    implements NavigationView.OnNavigationItemSelectedListener
               GridDialog.Communicator

fields
- − mMapView : MapView
- − mAddressSearchView : SearchView
- − mLocatorTask : LocatorTask
- − mGraphicsOverlay : GraphicsOverlay
- − mAddressGeocodeParameters : GeocodeParameters
- − mPinSourceSymbol : PictureMarkerSymbol
- − sublayers : SublayerList
- − mLocationDisplay : LocationDisplay
- − mSpinner : Spinner
- − mProgressDialog : ProgressDialog
- − mRouteTask : RouteTask
- − mRouteParams : RouteParameters
- − mRoute : Route
- − mRouteSymbol : SimpleLineSymbol
- − CurrentLocation : Location
- − PointPosition : Point
- − BasementButton : FloatingActionButton
- − FloorButton1 : FloatingActionButton
- − FloorButton2 : FloatingActionButton
- − FloorButton3 : FloatingActionButton
- − FloorButton4 : FloatingActionButton
- − FloorButton5 : FloatingActionButton
- − final TAG : String
- − final COLUMN_NAME_ADDRESS : String
- − final mColumnNames : String[]
- − requestCode : int
- − ~ reqPermissions : String[]

constructors

methods
- # onCreate (savedInstanceState : Bundle) : void
- + onRequestPermissionsResult(requestCode : int, permissions : String[], grantResults : int[]) : void
- + onBackPressed () : void
- + onCreateOptionsMenu (menu : Menu) : boolean
- + onOptionsItemSelected (item : MenuItem) : boolean
- + onNavigationItemSelected (item : MenuItem) : boolean
- + onDialogMessage (message : String) : void
- − setupAddressSearchView () : void
- − identifyGraphic (motionEvent : MotionEvent) : void
- − geoCodeTypedAddress(address : String) : void
- − displaySearchResult(geocodeResult : GeocodeResult) : void
- + CheckAddress(address : String) : boolean
- + floorsearchview (address : String) : void
- + returnfloorview (floor_integer : int, grid_index : int) : void
- + retrieveCoordinates (address : String) : Stop
- # onPause () : void
- # onResume() : void
- # onDestroy() : void

+ SplashScreen extends AppCompatActivity

fields
- − final SPLASH_DISPLAY_LENGTH : int

constructors

methods
- # onCreate (savedInstanceState : Bundle) : void

+ GridDialog extends AppCompatDialogFragment
    implements View.OnClickListener

fields
- ~ gridbutton1 : Button
- ~ gridbutton2 : Button
- ~ gridbutton3 : Button
- ~ gridbutton4 : Button
- ~ gridbutton5 : Button
- ~ gridbutton6 : Button
- ~ gridbutton7 : Button
- ~ gridbutton8 : Button
- ~ gridbutton9 : Button
- ~ gridbutton10 : Button
- ~ gridbutton11 : Button
- ~ gridbutton12 : Button
- ~ communicator : Communicator

constructors

methods
- + onAttach (context : Context) : void
- + onCreateView (inflater : LayoutInflater, container : ViewGroup, savedInstanceState : Bundle) : View
- + onClick(view : View) : void

```
+ SpinnerAdapter extends ArrayAdapter
─ fields ─────────────────────────────────
–  final   groupid:int
–  final   list:ArrayList<ItemData>
–  final   inflater :LayoutInflater
─ constructors ───────────────────────────
+  SpinnerAdapter(context:Activity, groupid:int, id:int, list:ArrayList<ItemData> )
─ methods ────────────────────────────────
+  getView (position:int, convertView: View , parent:ViewGroup):View
+  getDropDownView(position:int, convertView: View , parent:ViewGroup):View
```

```
+ ItemData
─ fields ─────────────────────────────
–  final   text :String
–  final   imageId:Integer
─ constructors ───────────────────────
+  ItemData (text:String, imageId:Integer)
─ methods ────────────────────────────
+  getText ():String
+  getImageId ():Integer
```
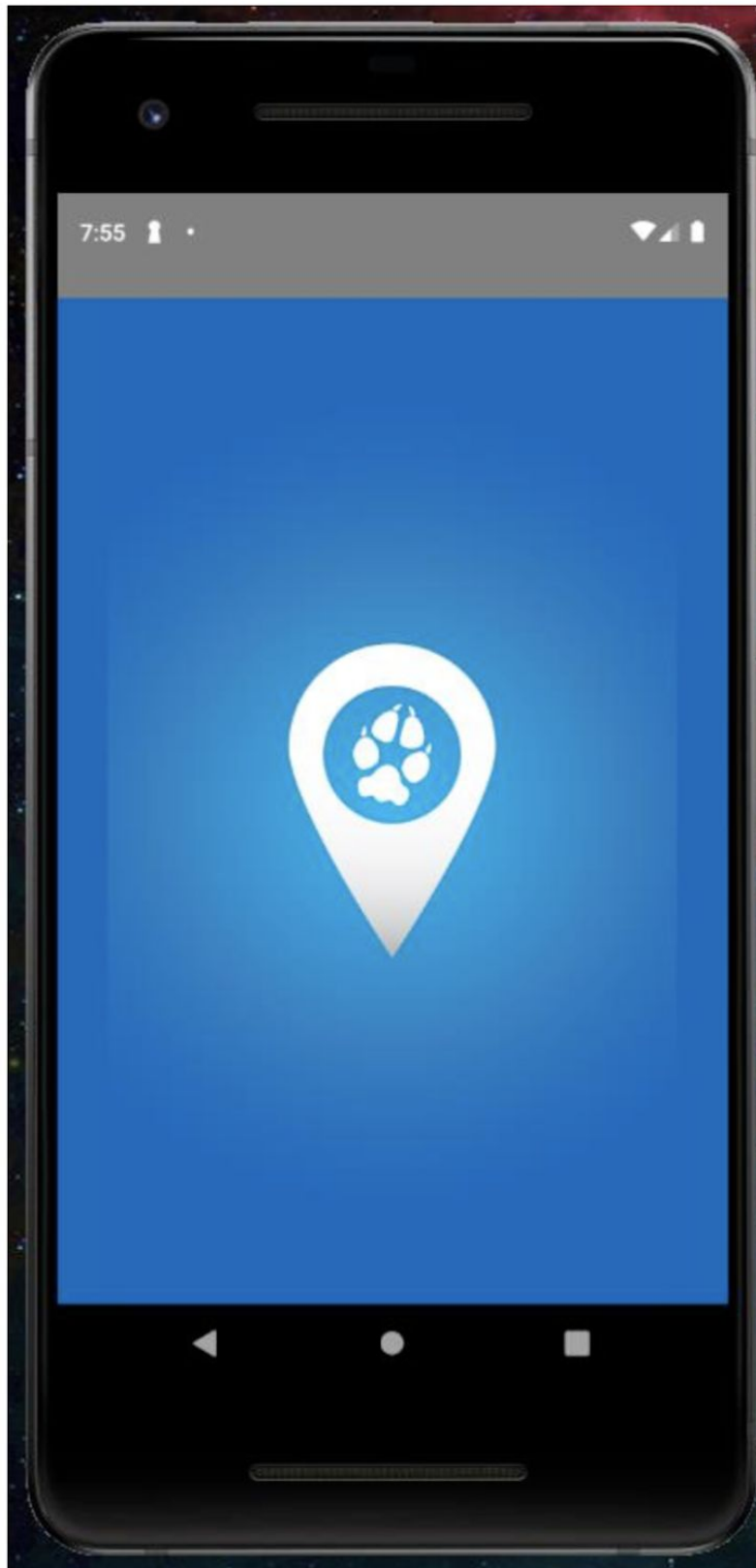
SplashScreen is displayed upon start-up

MainActivity has a button. When clicked...

...**GridDialog** is instantiated and 12 buttons are displayed. When you click a button (e.g. "Campus Shuttle")...

...a shuttle icon is displayed in the **MainActivity**
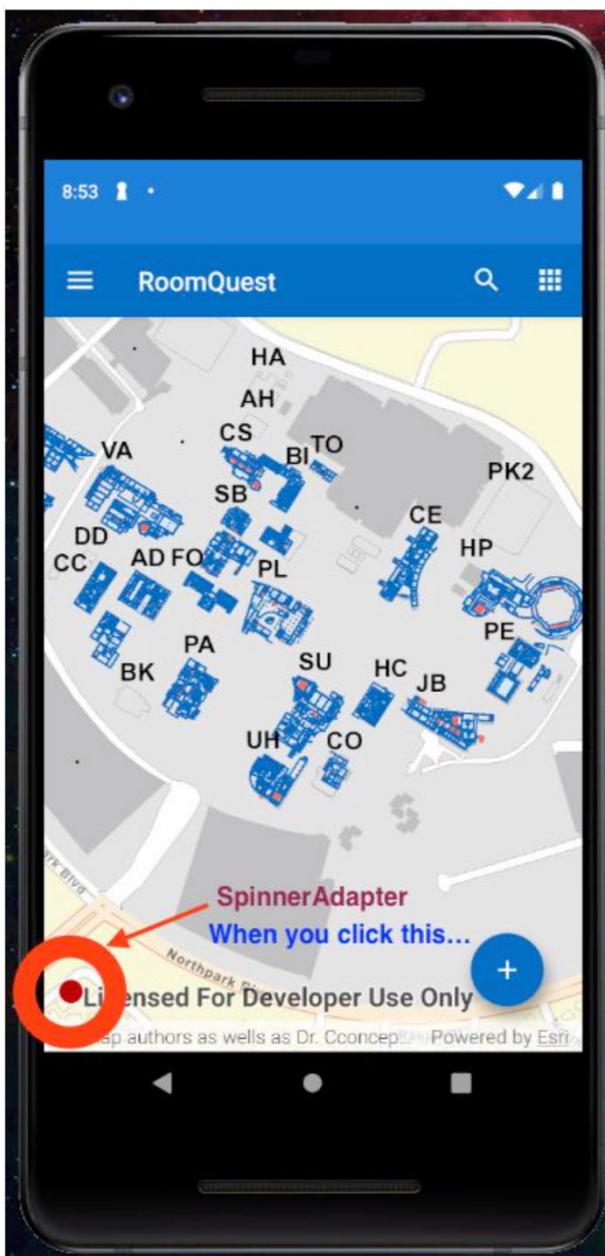
When you click the **SpinnerAdapter** on the **MainActivity**…
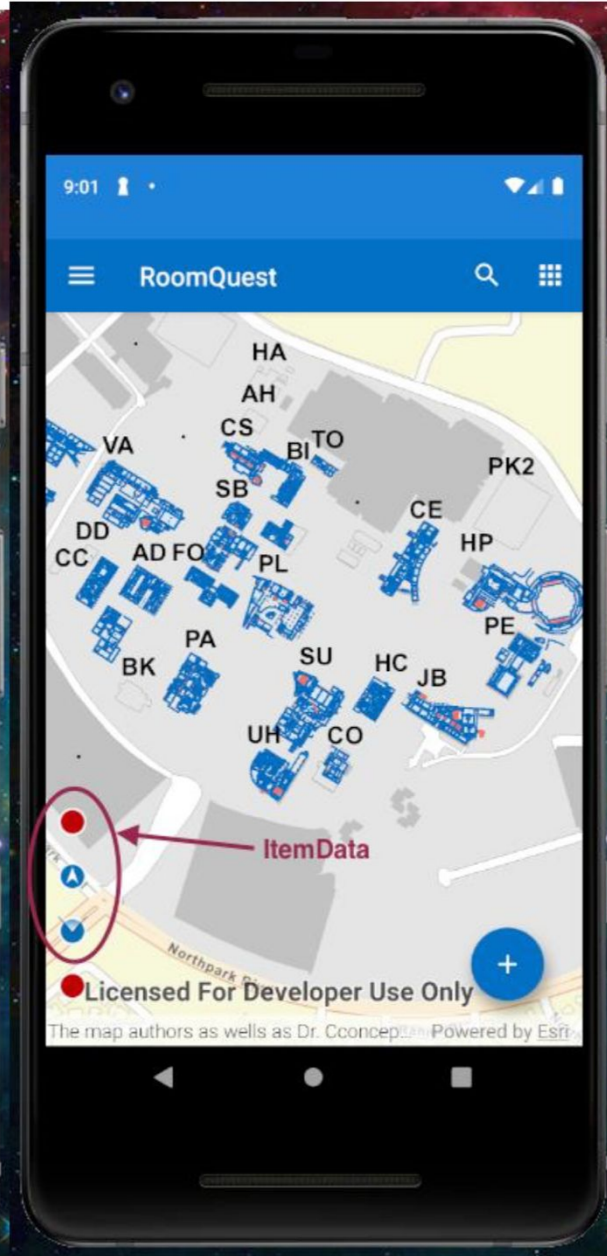
…the **ItemData** is instantiated and displayed on the **MainActivity**.

4. Pseudo-Code Algorithms of Critical Functions in Class Diagram

Main Activity Algorithm:

Include necessary Libraries

Class MainActivity
{Declared variables}
MapView mMapView
SearchView mAddressSearchView
LocatorTask mLocatorTask
GraphicsOverlay mGraphicsOverlay
GeocodeParameters mAddressGeocodeParameters
PictureMarkerSymbol mPinSourceSymbol
SublayerList sublayers
LocationDisplay mLocationDisplay
Spinner mSpinner
ProgressDialog mProgressDialog
RouteTask mRouteTask
RouteParameters mRouteParams
Route mRoute
SimpleLineSymbol mRouteSymbol
Location CurrentLocation
Point PointPosition
FloatingActionButton BasementButton, FloorButton1, FloorButton2, FloorButton3,
        FloorButton4, FloorButton5
String COLUMN_NAME_ADDRESS = "address"
private final String[] mColumnNames = { BaseColumns._ID, COLUMN_NAME_ADDRESS }
private int requestCode = 2
String[] reqPermissions =
        {"ACCESS_FINE_LOCATION","ACCESS_COARSE_LOCATION"}

        Function onCreate(Bundle savedInstanceState)
                Basement = menu_item_resource_id6
                FloorButton1 = menu_item_resource_id5
                FloorButton2 = menu_item_resource_id4
                FloorButton3 = menu_item_resource_id3
                FloorButton4 = menu_item_resource_id2
                FloorButton5 = menu_item_resource_id1

                mLocatorTask = new LocatorTask("GeocodeServer URL")
                mMapView = find mapView resources

Returnfloorview(1,0)
Set mMapView Viewpoint to (x_coordinate, y_coordinate, Spatial Reference)
mMapView.OnTouchListener
        Function Boolean onSingleTapConfirmed {Used for when tapping the
screen}

                Viewpoint vp = Get current mMapView Viewpoint Coordinates
                Set mMapView Viewpoint to vp
                Set mAddressSearch to false
                Close Options Menu
                Set mLocationDisplay to stop
                Return true
mGraphicsOverlay = new Graphics Layer
setupAddressSearchView()

BasementButton.setOnClickListener
        Function Void onClick
                Set mLocationDisplay to stop
                Viewpoint vp = Get current mMapView Viewpoint Coordinates
                Returnfloorview(0,0)
                Set mMapView Viewpoint to vp
                Remove existing graphics layer on mMapView

FloorButton1.setOnClickListener
        Function Void onClick
                Set mLocationDisplay to stop
                Viewpoint vp = Get current mMapView Viewpoint Coordinates
                Returnfloorview(1,0)
                Set mMapView Viewpoint to vp
                Remove existing graphics layer on mMapView

FloorButton2.setOnClickListener
        Function Void onClick
                Set mLocationDisplay to stop
                Viewpoint vp = Get current mMapView Viewpoint Coordinates
                Returnfloorview(2,0)
                Set mMapView Viewpoint to vp
                Remove existing graphics layer on mMapView

FloorButton3.setOnClickListener
        Function Void onClick
                Set mLocationDisplay to stop
                Viewpoint vp = Get current mMapView Viewpoint Coordinates
                Returnfloorview(3,0)
                Set mMapView Viewpoint to vp
                Remove existing graphics layer on mMapView

FloorButton4.setOnClickListener
 Function Void onClick
  Set mLocationDisplay to stop
  Viewpoint vp = Get current mMapView Viewpoint Coordinates
  Returnfloorview(4,0)
  Set mMapView Viewpoint to vp
  Remove existing graphics layer on mMapView

FloorButton5.setOnClickListener
 Function Void onClick
  Set mLocationDisplay to stop
  Viewpoint vp = Get current mMapView Viewpoint Coordinates
  Returnfloorview(5,0)
  Set mMapView Viewpoint to vp
  Remove existing graphics layer on mMapView

mLocationDisplay.addDataSourceStatusChangedListener
 Function Void onStatusChanged(LocationDisplay Event)
  If (LocationDisplay Event is true)
   Return
  If (LocationDisplay Event Error == null)
   Return
  Boolean permissionCheck1 = reqPermissions[0]) ==
 "PERMISSION_GRANTED"
  Boolean permissionCheck2 = reqPermissions[1]) ==
 "PERMISSION_GRANTED"
  If (permissionCheck1 is not true && permissionCheck2 is not true)
  Request GPS permissions
  Else
  Set mSpinner selection to disabled display

mSpinner.setOnItemSelectedListener {Used for GPS buttons}
 switch position
  Case 0
   If (mLocationDisplay is true)
    Stop Location Display
   Break Operation
  Case 1
   Set mLocationDisplay to Navigation view
   If (mLocationDisplay is false)
    Start Navigation Mode
   Break Operation
  Case 2
   Set mLocationDisplay to Compass view

If (mLocationDisplay is false)
Start Compass Mode
Break Operation
Function Void onNothingSelected {Left Empty}

PinDrawable = getDrawableResources.coyote_paw
Set PinDrawable Width
Set PinDrawable Height
Create Route_Button {Route Button is hidden from UI}
Function Void onClick()
mRouteTask = new Route(getResources)
{Left empty but is used for navigation drawer and search view}

Function onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults)
If permission granted
Begin mLocationDisplay.startAsync()
Else
Show Error message

Function onBackPressed() {Detects the back button on the Home Buttons}
GridDialog gridDialog = new GridDialog()
Dismiss gridDialog
If (Navigation Drawer is open)
Close Navigation Drawer
Else
onBackPressed()

Function onCreateOptionsMenu(Menu menu)
Inflate menu items {Creates the menu buttons at the top of the UI}

Function onOptionsItemSelected(MenuItem item)
int id = item.getItemId()

If (id == action_search_integer)
Open search view
Else If (id == grid_integer)
Initiate GridDialog Class Layout
Else If (id != action_search_integer)
close search view
Return Item Selected

Function onNavigationItemSelected(Menu item)
int id = item.getItemID

```
If (id == "building1")
        geoCodeTypedAddress("Administration")
Else If (id == "building2")
        geoCodeTypedAddress("Auto Fleet Services")
Else If (id == "building3")
        geoCodeTypedAddress("Vivarium")
.
.
.
Else If (id == "building46")
        geoCodeTypedAddress("Yasuda Center")

Return true

Function onDialogMessage(String message)
        switch(message)
                case "Bicycle Racks"
                        Returnfloorview(1,1)
                case "Parking Permit Dispensers"
                        Returnfloorview(1,2)
                case "Disabilities Parking Areas"
                        Returnfloorview(1,3)
                .
                .
                .
                case "Dining"
                        Returnfloorview(1,12)

Function Void setupAddressSearchView()
        Get place name and address attributes from mAddressGeocodeParameters

Function Void identifyGraphic(MotionEvent motionEvent)
        Get screen point
        Get graphics from the overlay near the tapped location of mMapView
                Function run()
                        If graphic has been identified
                                Get the first graphic identified
Function Void geoCodeTypedAddress(String address)
        If (address != null)
                mLocatorTask.addDoneLoadingListener
                        Function Void run()
                                If(mLocatorTask.getLoadStatus() == Loaded)
                                        Function geocodeResult
                                                Function Void run()
```

If (geocodeResults.size > 0)

    If(CheckAddress(address)

&& mLocationDisplay.isStarted())

        Begin Route_Button

        Stop

DevicePointLocation = Retrieve Device Location

        Stop

Destination Point = retrieveCoordinates(address)

        Add

destination stops

        Route stop

locations

displaySearchResult(geocodeResults.get(0))

    Else

displaySearchResult(geocodeResults.get(0))

floorsearchview(address)

    Else

        If(address == "Secret Code"

            Display Secret Map

Layer

        Else

            Show Error Message

    Else

        Show Message mLocatorTask cannot be Loaded

    mLocatorTask.loadAsync()
    Set mAddressSearchView to close

    Function Void displaySearchResult(GeocodeResult geocodeResult)
        If(mMapView.getCallout() != null && mMapView.getCallout().isShowing())
            Dismiss mMapView Callout()
        Load Graphic layer and Point Location of geocodeResult

    Function Boolean CheckAddress (string address)
        String[] AddressList = {"Administration","Auto Fleet Services", "Vivarium", … , "Yasuda Center"}
        for (int i ; i < AddressList.length ; i++)
            If (address == AddressList[i])
                Return true
        Return false

    {This Function is used to display the correct floor when searching for a specIfic room number}

{This only works for floors 1 through 5}
Function Void floorsearchview(String address)
        for(int i ; i < address.length() ; i++)
                If(address.character[i] == '1')
                        Returnfloorview(1,0)
                        Show message "1st Floor"
                        Break operation
                Else If(address.character[i] == '2')
                        Returnfloorview(2,0)
                        Show message "2nd Floor"
                        Break operation
                Else If(address.character[i] == '3')
                        Returnfloorview(3,0)
                        Show message "3rd Floor"
                        Break operation
                Else If(address.character[i] == '4')
                        Returnfloorview(4,0)
                        Show message "4th Floor"
                        Break operation
                Else If(address.character[i] == '5')
                        Returnfloorview(5,0)
                        Show message "5th Floor"
                        Break operation

{This Function Returns all the necessary map layers for basement, 1st, 2nd, 3rd, 4th, and 5th floor}
{grid_index is for Returning a grid layer}
{Grid layers can only be show with the 1st floor map layers}
Function Void returnfloorview(int floor_integer, int grid_index)
        If(floor_integer == 0)
                mapImageLayer.addDoneLoadingListener
                        Get sublayers from web service
                        for(int i ; i < 26 ; i++)
                                {Certain layers are open by default. This Function sets all of them to false}
                                        Set sublayers(i) to false
                        Set sublayers(5) to true
                        Set sublayers(6) to true
                        Set sublayers(8) to true
                        Set sublayers(19) to true
                        Set sublayers(20) to true
                        Set sublayers(26) to true
        Else If (floor_integer == 1)
                mapImageLayer.addDoneLoadingListener
                        Get sublayers from web service

```
for(int i ; i < 26 ; i++)
        Set sublayers(i) to false
Set sublayers(4) to true
Set sublayers(6) to true
Set sublayers(8) to true
Set sublayers(17) to true
Set sublayers(18) to true
Set sublayers(25) to true
If ( grid_index != 0 )
        Remove all graphic layers
        Set sublayers(7) to true
        for (int i ; i < 18 ; i++)
                Set sublayers(7).SubLayerContents(i) to false
        If ( grid_index == 1 )
                Set sublayers(7).SubLayerContents(0) to true
        Else If ( grid_index == 2 )
                Set sublayers(7).SubLayerContents(1) to true
        Else If ( grid_index == 3 )
                Set sublayers(7).SubLayerContents(2) to true
        Else If ( grid_index == 4 )
                Set sublayers(7).SubLayerContents(3) to true
        Else If ( grid_index == 5 )
                Set sublayers(7).SubLayerContents(4) to true
        Else If ( grid_index == 6 )
                Set sublayers(7).SubLayerContents(5) to true
        Else If ( grid_index == 7 )
                Set sublayers(7).SubLayerContents(6) to true
        Else If ( grid_index == 8 )
                Set sublayers(7).SubLayerContents(7) to true
        Else If ( grid_index == 9 )
                Set sublayers(7).SubLayerContents(8) to true
        Else If ( grid_index == 10 )
                Set sublayers(7).SubLayerContents(9) to true
        Else If ( grid_index == 11 )
                Set sublayers(7).SubLayerContents(10) to true
                Set sublayers(7).SubLayerContents(11) to true
        Else If ( grid_index == 12 )
                Set sublayers(7).SubLayerContents(12) to true
Else If (floor_integer == 2)
        mapImageLayer.addDoneLoadingListener
                Get sublayers from web service
                for(int i ; i < 26 ; i++)
                        Set sublayers(i) to false
                Set sublayers(3) to true
                Set sublayers(6) to true
```

```
                    Set sublayers(8) to true
                    Set sublayers(15) to true
                    Set sublayers(16) to true
                    Set sublayers(24) to true
        Else If (floor_integer == 3)
                mapImageLayer.addDoneLoadingListener
                    Get sublayers from web service
                    for(int i ; i < 26 ; i++)
                            Set sublayers(i) to false
                    Set sublayers(2) to true
                    Set sublayers(6) to true
                    Set sublayers(8) to true
                    Set sublayers(13) to true
                    Set sublayers(14) to true
                    Set sublayers(23) to true
        Else If (floor_integer == 4)
                mapImageLayer.addDoneLoadingListener
                    Get sublayers from web service
                    for(int i ; i < 26 ; i++)
                            Set sublayers(i) to false
                    Set sublayers(1) to true
                    Set sublayers(6) to true
                    Set sublayers(8) to true
                    Set sublayers(11) to true
                    Set sublayers(12) to true
                    Set sublayers(22) to true
        Else If (floor_integer == 5)
                mapImageLayer.addDoneLoadingListener
                    Get sublayers from web service
                    for(int i ; i < 26 ; i++)
                            Set sublayers(i) to false
                    Set sublayers(0) to true
                    Set sublayers(6) to true
                    Set sublayers(8) to true
                    Set sublayers(9) to true
                    Set sublayers(10) to true
                    Set sublayers(21) to true


    Function Stop retrieveCoordinates (String address)
        If (address == "Administration")
                Stop DestinationPoint = new Stop (new Point (x_coordinate,
y_coordinate, Spatial reference)
                Return Destination Point
        Else If (address == "Auto Fleet Services")
```

Stop DestinationPoint = new Stop (new Point (x_coordinate, y_coordinate, Spatial reference)

Return Destination Point

Else If (address == "Vivarium")

Stop DestinationPoint = new Stop (new Point (x_coordinate, y_coordinate, Spatial reference)

Return Destination Point

.

.

.

Else If (address == "Yasuda Center")

Stop DestinationPoint = new Stop (new Point (x_coordinate, y_coordinate, Spatial reference)

Return Destination Point

Else

Return null

Function Void onPause()
Returnfloorview(1,0)
Set mMapView viewpoint to (x_coordinate,y_coordinate,Spatial Reference)

Function Void onResume()
Returnfloorview(1,0)
Set mMapView viewpoint to (x_coordinate,y_coordinate,Spatial Reference)
Function Void onDestroy()
Delete mMapView

GridDialog Algorithm:

Include necessary libraries

Class GridDialog
Button gridbutton1, gridbutton2, gridbutton3, gridbutton4, gridbutton5, gridbutton6, gridbutton7,
gridbutton8, gridbutton9, gridbutton10, gridbutton11, gridbutton12

Communicator communicator

Function Void onAttach()
communicator = context

Function View onCreateView()
Get dialog window
gridbutton1 = get grid_button_one resources
gridbutton1.setOnClickListener(this)

gridbutton2 = get grid_button_one resources
gridbutton2.setOnClickListener(this)
gridbutton3 = get grid_button_one resources
gridbutton3.setOnClickListener(this)
.
.
.
gridbutton12 = get grid_button_one resources
gridbutton12.setOnClickListener(this)
Set cancelable to true {Used to exit the grid layout window}
Return true

Function Void onClick(View view)
    If (view id == grid_button_one)
        Show communicator message "Bicycle Racks"
        Dismiss dialog
    Else If (view id == grid_button_two)
        Show communicator message "Parking Permit Dispensers"
        Dismiss dialog
    Else If (view id == grid_button_three)
        Show communicator message "Disability Parking Areas"
        Dismiss dialog
    .
    .
    .
    Else If (view id == grid_button_twelve)
        Show communicator message "Dining"
        Dismiss dialog

Function interface Commincator
    Void onDialogMessage(String message)

ItemData Algorithm:

Include necessary libraries

Class ItemData
    String text
    Integer imageId

    Function ItemData (String text, Integer imageId)
        text = this text
        imageId = this imageId

    Function String getText()

Return text

Function Integer getImageId()
Return imageId

SpinnerAdapter Algorithm:

Include necessary libraries

Class SpinnerAdapater
int groupid
ArrayList<ItemData> list
LayoutInflator inflator

Function SpinnerAdapter (Activity context, int groupid, int id, ArrayList<ItemData> list)
list = this list
inflator = get system service of context LAYOUT_INFLATER_SERVICE
groupid = this groupid

Function View getView (int position, View convertView, ViewGroup parent)
View itemView = inflator.(groupid, parent, false)
ImageView imageView = img id
Get imageView position id
TextView textView = (TextView) itemView.findViewById(R.id.txt)
TextView textView = txt id
Get textView position
Return itemView

Function View getDropDownView (int position, View convertView, ViewGroup parent)
Return getView(position, convertView, parent)