```as
package
{
    import com.vaden.ui.PanelBase;

    import flash.display.Sprite;

    [SWF(width="760",height="500",frameRate="30", backgroundColor="0x000000")]

    public class Main extends Sprite
    {
        public function Main()
        {
            super();

            var panel:PanelBase = new PanelBase;
            this.addChild(panel);
            panel.x = stage.stageWidth/2;
            panel.y = stage.stageHeight/2;


        }
    }
}
```

```as
package com.vaden.events
{
    import flash.display.Bitmap;
    import flash.events.Event;

    public class ImageEvent extends Event
    {
        public static const IMAGE_LOADED:String = "image_loaded";

        public var image:Bitmap;

        public function ImageEvent(type:String, bubbles:Boolean=false, cancelable:Boolean=false)
        {
            super(type, bubbles, cancelable);
        }

        public override function clone():Event
        {
            return new ImageEvent(type, bubbles, cancelable);
        }
    }
}
```

```as
package com.vaden.ui
{
    import com.vaden.managers.RollManager;
    import com.vaden.vo.WeatherVO;

    import flash.events.Event;
    import flash.events.MouseEvent;
    import flash.net.URLLoader;
    import flash.net.URLRequest;

    import libs.InputView;

    public class InputView extends libs.InputView
    {
        public function InputView()
        {
            super();
            //clear the input textfield for the zipcode and restrict to allow only 5 chars and only numbers
            this.tf_zipcode.text = "";
            this.tf_zipcode.restrict = "0-9";
            this.tf_zipcode.maxChars = 5;


            this.btn_enter.buttonMode = true;

            //create a rollover manager from library to manage rollover effect on btn_enter
            var roll:RollManager = new RollManager(this.btn_enter);




        }


    }
}
```

PanelBase.as

```as
package com.vaden.ui
{
    import com.vaden.ImageLoaderBasic;
    import com.vaden.events.ImageEvent;
    import com.vaden.vo.WeatherVO;

    import flash.events.Event;
    import flash.events.MouseEvent;
    import flash.net.URLLoader;
    import flash.net.URLRequest;

    import libs.PanelBase;

    import org.osmf.image.ImageLoader;

    public class PanelBase extends libs.PanelBase
    {
        private var _input:InputView;
        private var _result:ResultView;
        private var vo:WeatherVO;

        public function PanelBase()
        {
            super();

            _input = new InputView();
            this.addChild(_input);

            //add event listener to the button to listen for input text and process into the XML
            _input.btn_enter.addEventListener(MouseEvent.CLICK, processZip);
        }

        private function processZip(e:Event):void
        {
            if (_input.tf_zipcode.text == null)
            {
                _input.tf_instruct.text = "Please enter a valid zip code.";
            }else if (_input.tf_zipcode.text.length < 5)
            {
                _input.tf_zipcode.text = "";
                _input.tf_instruct.text = "Please enter a valid zip code.";
            }else
            {
                //create a variable zip and assign the input from the textfield
```

```
to the variable
```

```
83              this.removeChild(_input);
84
85              //remove the view and replace with results view
86              getImages();
87          }
88
89          private function getImages():void
90          {
91              //create a new URLLoader to load the graphics from the graphics xml
92              var ld:URLLoader = new URLLoader();
93              ld.load(new URLRequest("assets/xml/graphics.xml"));
94
95              //listen for the xml data to finish(complete) and then extract the
    data
96              ld.addEventListener(Event.COMPLETE, loadImages);
97          }
98
99          private function loadImages(e:Event):void
100         {
101             var xmldata:XML = XML(e.target.data);
102             var x:String = vo.todayWeatherCode;
103
104             if (x == "26"|| x == "27"||x == "29"||x == "31"||x == "33" )
105             {
106                 var ldMoon:ImageLoaderBasic = new ImageLoaderBasic("assets/
    images/moon.png");
107                 ldMoon.addEventListener(ImageEvent.IMAGE_LOADED, onLoad);
108             }
109             if(x == "0"||x == "1"||x == "2"||x == "9"||x == "10"||x == "11"||x
    == "12"||x == "17")
110             {
111                 var ldRain:ImageLoaderBasic = new ImageLoaderBasic("assets/
    images/rain.png");
112                 ldRain.addEventListener(ImageEvent.IMAGE_LOADED, onLoad);
113             }
114             if(x == "5"||x == "6"||x == "7"||x == "8"||x == "13"||x == "14"||x
    == "15"||x == "16"||x == "18"||x == "25"||x == "41"||x == "42"||x == "43"||x ==
    "46")
115             {
116                 var ldSnow:ImageLoaderBasic = new ImageLoaderBasic("assets/
    images/snow.png");
117                 ldSnow.addEventListener(ImageEvent.IMAGE_LOADED, onLoad);
118             }
119             if(x == "19"||x == "3"||x == "4"||x == "20"||x == "35"||x == "36"||x
    == "37"||x == "38"||x == "39"||x == "40"||x == "44"||x == "45"||x == "47" )
```

```
120             {
121                 var ldStorm:ImageLoaderBasic = new ImageLoaderBasic("assets/
    images/storm.png");
122                 ldStorm.addEventListener(ImageEvent.IMAGE_LOADED, onLoad);
123             }
124             if(x == "21"||x == "22"||x == "23"||x == "24"||x == "28"||x == "30"||
    x == "32"||x == "34"||x == "36"||x == "3200")
125             {
126                 var ldSun:ImageLoaderBasic = new ImageLoaderBasic("assets/images/
    sun.png");
127                 ldSun.addEventListener(ImageEvent.IMAGE_LOADED, onLoad);
128             }
129         }
130 
131         private function onLoad(e:ImageEvent):void
132         {
133             //add the ResultView to the stage
134             _result = new ResultView();
135             this.addChild(_result);
136 
137             //process the vo items into the textfields within the ResultView
138             _result.tf_city.text = vo.city;
139             _result.tf_currentTemp.text = vo.currentTemp;
140             _result.tf_sunrise.text = vo.sunrise;
141             _result.tf_sunset.text = vo.sunset;
142             _result.tf_todayHigh.text = vo.todayHigh;
143             _result.tf_todayLow.text = vo.todayLow;
144             _result.tf_tomorrowHigh.text = vo.tomorrowHigh;
145             _result.tf_tomorrowLow.text = vo.tomorrowLow;
146 
147             _result.addChild(e.image);
148             e.image.x = -110;
149             e.image.y = -200;
150 
151             //add EventListener for the restart button
152             _result.btn_restart.addEventListener(MouseEvent.CLICK, restart);
153         }
154 
155         private function restart(e:Event):void
156         {
157             this.removeChild(_result);
158             this.addChild(_input);
159         }
160     }
161 }
```

```as
package com.vaden.ui
{
    import com.vaden.managers.RollManager;
    import com.vaden.vo.WeatherVO;

    import libs.ResultView;

    public class ResultView extends libs.ResultView
    {
        public function ResultView()
        {
            super();

            //clear the textfields
            this.tf_city.text = "";
            this.tf_currentTemp.text = "";
            this.tf_sunrise.text = "";
            this.tf_sunset.text = "";
            this.tf_todayHigh.text = "";
            this.tf_todayLow.text = "";
            this.tf_tomorrowHigh.text = "";
            this.tf_tomorrowLow.text = "";

            var roll:RollManager = new RollManager(this.btn_restart);

        }
    }
}
```

```as
package com.vaden.vo
{
    public class WeatherVO
    {
        public var city:String;
        public var currentTemp:String;
        public var sunrise:String;
        public var sunset: String;
        public var todayHigh:String;
        public var todayLow: String;
        public var tomorrowHigh:String;
        public var tomorrowLow:String;
        public var todayWeatherCode:String;
        public var tomorrowWeatherCode:String;

        public function WeatherVO()
        {
        }
    }
}
```

```as
package com.vaden
{
    import com.vaden.events.ImageEvent;
    import flash.display.Loader;
    import flash.events.Event;
    import flash.events.EventDispatcher;
    import flash.net.URLRequest;


    public class ImageLoaderBasic extends EventDispatcher
    {
        private var ld:Loader;

        public function ImageLoaderBasic(file:String)
        {
            super();
            ld = new Loader();
            ld.load(new URLRequest(file));
            //complete event goes on the contentLoaderInfo property not the loader class
            ld.contentLoaderInfo.addEventListener(Event.COMPLETE, onLoad);
        }

        private function onLoad(e:Event):void
        {
            //make a new custom event and dispatch it here
            var evt:ImageEvent = new ImageEvent(ImageEvent.IMAGE_LOADED);

            //attach the bitmap that was loaded before dispatching the event
            evt.image = e.target.content;
            dispatchEvent(evt);

            //need to remove the event listener and purge elements from RAM
            ld.contentLoaderInfo.removeEventListener(Event.COMPLETE, onLoad);
            ld.unload();
            ld = null;
        }
    }
}
```