

# Design of the Touchscreen Kiosk-Based Local Area Tour Guide System

Jaegeol Yim

*Dept. of Computer Engineering, Dongguk University at Gyeongju,  
Gyeongbuk Korea  
yim@dongguk.ac.kr*

## Abstract

*Touchscreen kiosks are widely used to automate various services. Automatic teller machines, certificate-issuing machines found on university campuses, and ticket machines found in airport terminals and bus terminals are examples of such kiosks. Since localized real-time information, interactive touch-control services, and value-added integrated marketing services are extremely important services to tourists, this paper introduces the design of a tourist guide system running on touch screen kiosks. A kiosk is connected to the bus information system, an Internet broadcast system, online shopping and a web portal, and allows users to select one of the services.*

**Keywords:** *Kiosk, Digital Signage, Video, Online Shopping*

## 1. Introduction

As electronic display devices and computing techniques advance, kiosks are widely used to provide localized real-time information, interactive touch-control services, and value-added integrated marketing services [1-2].

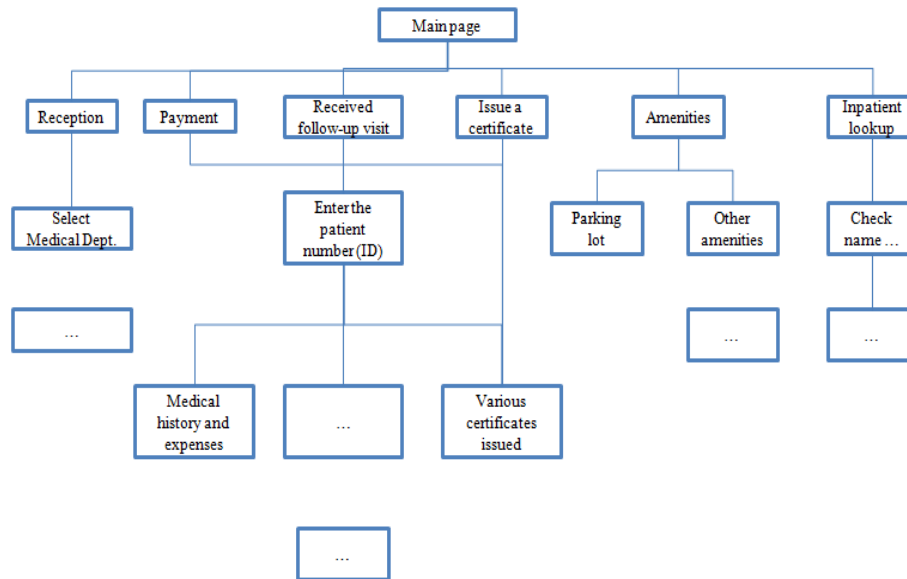
This paper reviews techniques recently published in the fields of kiosk usage. Development of information architecture for a user-centered hospital kiosk, aspects of trustworthy kiosk computing, and other topics will be discussed. A kiosk is a kind of digital signage. Audience measurement procedures for digital signage services will also be discussed [1].

The design of a user interface for a local-area tourism kiosk system will be introduced. The user interface consists of a video view, two image views, and an online shopping mall [1].

## 2. Related Works

Exercising Markov models and enhanced K-mean clustering methods, Hung *et al.* [2] analyzed log data recorded in eight kiosks that provide interactive services in order to understand self-seeking behavior by users, such as self-seeking use cycles, time, function numbers, and the depth and extent of services [1].

After conducting a usability test and in-depth interviews with each group of users, Park *et al.* [3] suggested a proper information architecture for an integrated user centered hospital kiosk. The suggested architecture is illustrated in Figure 1. The system provides the following functions: routes from where users are, parking fee payment, making appointment with a doctor by clicking the name, helping users to choose the right clinical program or doctor, medical bill payment, issuance of documents, input and search, a full screen menu, and so on. The system provides information on the location of amenities and transportation, parking guidance, clinical programs, general medical procedures, and so on [1].



**Figure 1. The User Interface Architecture of a Hospital Kiosk System [1]**

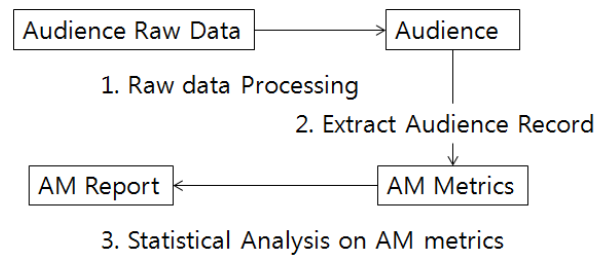
A kiosk must perform what the user asks it to do and must not do anything else. For example, a kiosk must not record a user's ID and password when the user asks to transfer money from one bank account to another. A kiosk is deemed trustworthy if it does only what the user wants it to do. Carriss *et al.* [4] presented a trust establishment protocol between mobile devices and kiosks [1].

Dipin *et al.* [5] proposed a kiosk-based employment generation system. Producers or service providers register their product or service at the kiosk using their telephones or accessing the Internet. Consumers also register their needs at the kiosk in the same manner. The administrator of the kiosk matches producers and consumers [1].

Digital signage is electronic displays placed in public or commercial areas. Multimedia contents consisting of text, audio, and video are displayed on digital signage. Digital signage systems receive the multimedia content stream from the server and display the stream as soon as it arrives. Users of digital signage systems can interact with the systems, *i.e.* users can select from a menu on the screen in order to tell the system to perform the instruction represented by the menu [1, 6].

A digital signage system can be equipped with various sensor devices, such as a camera, a microphone, a thermometer, *etc* A digital signage system may also support various short-range communications capabilities, such as radio frequency ID (RFID), near field communication (NFC), Bluetooth, and infrared-ray. Hyun *et al.* [6] described interfaces through which audience measurement will occur [1].

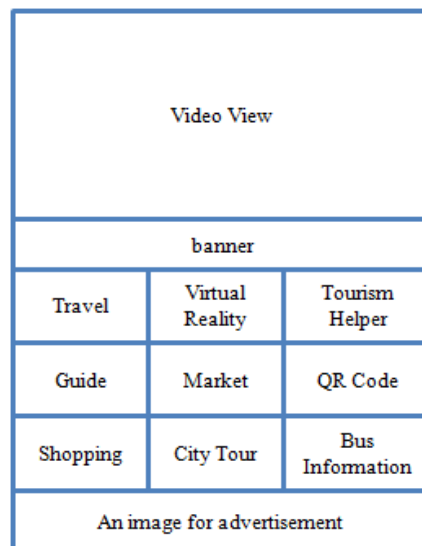
The authors of [7] introduced a context-aware Bluetooth device protection system that provides better security options to users in any situation for digital signage. The authors of [8] introduced generic steps shown in Figure 2 through which digital signage systems can extract audience measurement information. The authors of [9] presented a three-tier network architecture for healthcare monitoring. The authors of [10] introduced a flow-based conceptualization of services. We adapt multi-tier and web service in our design.



**Figure 2. Audience Information Extraction Steps [8]**

### 3 Design of the Kiosk System

Considering kiosk-related research results, a kiosk system that provides live broadcast and online shopping is proposed. The user interface of the system was designed as shown in Figure 3. Live broadcast is played in the video view. The interface has a banner displaying break news. At the bottom of the screen is an image for commercial advertisements. There are nine images in the user interface representing the main menus of the system: travel, virtual reality, tourism helper, tour guide, market, quick response (QR) code, shopping, city tour, and bus information.



**Figure 3. Design of the User Interface for the Kiosk [1]**

The service page for the 'Travel' menu is designed as shown in Figure 4. In the "Image view", an image representing the most distinguished tourism resources in the local area is displayed. Under this image, there is a list of images representing the categories of tourism resources. At the bottom, the buttons representing the main menu items are located. If the user selects a particular category in the category list, the system moves to a new page with tourism resources that are included in that category, as shown in Figure 5.

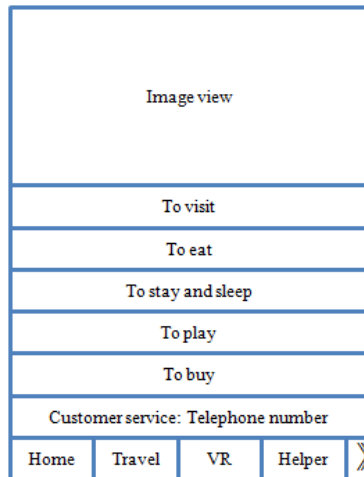


Figure 4. The Layout of the 'Travel' Page

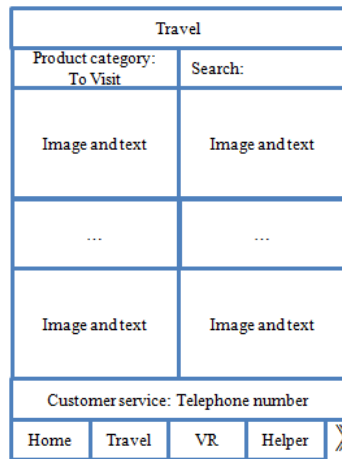


Figure 5. The Layout of the "To Visit" Page



Figure 6. A Sample User Interface for the "Virtual Reality" Service

If the "Virtual reality (VR)" button is clicked, then the user interface for 3D views of tourist attractions appears, as shown in Figure 6. Users are allowed to select any VR button in order to enjoy a 3D view of the selected tourist attraction. The "Shopping" button leads users to the online shopping mall. From the mall, the users can select items to buy and can send their telephone number to the server. Then, the server returns the uniform resource locator (URL) where users can place orders for the items.

The top level structure of the kiosk system is described in Figure 7. A kiosk is connected to the streaming server through the Internet and simultaneously receives content from this server and plays received content. It is also connected to the bus information system (BIS). The BIS recommends a bus route for the given departure and destination bus stops and provides arrival times for buses approaching the departure bus stop.

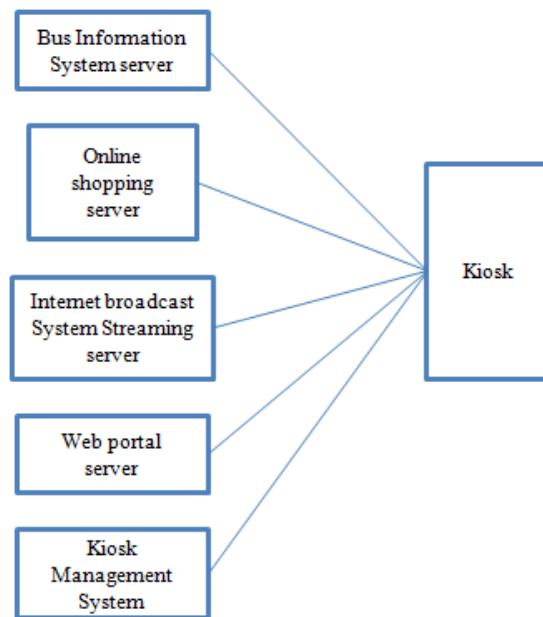


Figure 7. The Top Level Structure of the Kiosk System

#### 4. Implementation Considerations

Part of the system was implemented as a Windows Presentation Foundation (WPF) application with the Visual C# integrated development environment (IDE). This IDE provides a ToolBox from which developers can drag controls, such as TextBox and Button, and drop in the Form in order to implement a user interface. Given a string and data consisting of many phrases delimited by the "/" character, the setList(String data) method deploys controls on the user interface. This method splits the given string or data into elements of a string array using the following sentence:

```
String[] arrData = datas.Split(new char[] { '/' });
```

For each element of arrData, this method creates a Label as follows:

```
System.Windows.Controls.Label CatNM = new  
System.Windows.Controls.Label();  
CatNM.Content = " " + arrData[i];
```

A label is associated with an image visualizing the concept represented by the label:

```
bitmap.UriSource = new Uri(C...IMG_DIR + "List_L_" + C...String(num) +  
".png");
```

Then, the method put the label and image together in a StackPanel, as shown in Figure 8. SurfaceListBoxItem represents an item that users can select from a SurfaceListBox control. Defined are constants and commonly used utility methods, like the setList method in the clsCommon class.

```
private void setList(String datas)  
{  
    String[] arrData = datas.Split(new char[] { '/' });  
    ...  
    for (int i = 0; i < arrData.Length; i++)  
    {  
        SurfaceListBoxItem sfLstBitem = new SurfaceListBoxItem();  
        SurfaceButton sb = new SurfaceButton();  
        StackPanel sp = new StackPanel();  
  
        System.Windows.Controls.Label          CatNM          =          new  
System.Windows.Controls.Label();  
        CatNM.Content = " " + arrData[i];  
        CatNM.FontSize = 35;  
        bitmap.UriSource = new Uri(C...DIR + "List_L_" + Con...String(num) +  
".png");  
        ...  
        Rectangle NumImage = new Rectangle();  
        ...  
        sp.Children.Add(NumImage);  
        sp.Children.Add(CatNM);  
        ...  
    }  
}
```

**Figure 8. Part of the SetList Method**

The DB\_Control method, which retrieves menu items from the database, is defined and shown in Figure 9. The DataSet class represents an in-memory cache of data. The XmlReader class represents a reader that provides fast, non-cached, forward-only access to XML data.

```
public void DB_Control ...
{
    string strSql;
    DataSet ds = new DataSet();
    if (clsCommon.Menu == "Coupon")
    {
        XmlReader reader = XmlReader.Create("http...ID=...password=...");
        ...
    }
    else if (clsCommon.Menu == "Tour")
    {
        strSql = "SEL... categoryName ... FROM GJTour where catid in('kgj2...26')";
        ...
    }
    else if (clsCommon.Menu == "Food")
    {
        strSql = "SEL ... catid) ... FROM ... in('kgj3_11',..., 'kgj3_10')";
    }
    else if (clsCommon.Menu == "Stay")
    ...
    else if (clsCommon.Menu == "Heritage")
    ...
    else if (clsCommon.Menu == "Cultural")
    ...
    Title.Source = ...GetBitmap... Image...(...IMG_DIR + ... Language +
    @"\Time.png");
    ...
}
```

**Figure 9. Part of the DB\_Control Method**

APP\_DIR, TravelImgPath, CultureImgPath, and MarketImgPath are examples of constants defined in the clsCommon class. For example, APP\_DIR, the release path, is defined as follows:

```
readonly ... APP_DIR = AppDomain.BaseDirectory;
```

The following commonly used methods are also defined in clsCommon class:

GetBitmapImageFromFilePath(string strImagePath) returns the BitmapImage of the image designated by strImagePath.

ImageLoad\_ForLocalFile\_NotUseThread(System.Windows.Controls.Panel pnl, Uri uriSource) draws the image designated by uriSource on the panel.

loadData() reads data needed to create the main page of the user interface from designated URLs. For example, the following is a part of the code to read "shopping categories":

```
List<Dic ...>> lstShopCouponCat =
getXmlTravelData("http://.../getCategorySub");
```

One of the most important elements of the user interface of the kiosk system is the web page. The main web page is defined in GJTour\_Main, as shown in Figure 10. This is a Partial class, and it will be combined with GJTour\_Main.xaml when the application is compiled. This class extends the Web.UI.Page class representing an .aspx file, also known as a Web Forms page. One of the constructors of this class requires a container argument. This constructor invokes the InitializeComponent() method in order to load the "...component/gjtour\_main.xaml" xaml file. Then it assigns the container argument to the local variable. When the page is loaded, it invokes ImgControl() in order to initialize all the variables representing images. For example, the following sentence shows the typical form of the sentences found in the ImgControl() method.

```
Btn_Shop.NormalImage = Con...NEW_IMG_DIR + @"\main_btn_02.png";
```

The webBrowser1 in the Page\_Loaded is defined in the GJTour\_Main.xaml.

```
public partial class GJTour_Main : Page
...
public GJTour_Main(Container aContainer)
{
    InitializeComponent();
    theContainer = aContainer;
    ...
}

private void Page_Loaded(object sender, RoutedEventArgs e)
{
    ImgControl();
    ...
    webBrowser1.Navigate("http://localhost/gsb_kiosk/gsb_video.html");
}
...
```

**Figure 10. Part of the Definition for GJ Tour\_Main**

The NaverClass class, which renders the Naver map, is defined as shown in Figure 11, where the WebBrowserOverlay class creates a Window, and the vrf argument represents a Canvas. The Canvas, vrf, is hosted by the Window created by WebBrowserOverlay, and WebBrowser in the Canvas will display the Naver map.

```
public partial class NaverClass : Window
...
public NaverClass()
{
```



```
        InitializeComponent();
    }

    private void Window_Loaded(object sender, RoutedEventArgs e)
    {
        wbo = new WebBrowserOverlay(vrf);
        ...
        System.Windows.Controls.WebBrowser wb = wbo.WebBrowser;
        wb.Navigate(new Uri("http://map.naver.com/"));
        ...
    }
}
```

**Figure 11. Part of the Definition of NaverClass**

PopupClass extends the Window in order to handle the menu items that display messages on a pop-up window. Part of the definition of this class is shown in Figure 12 where popupBG, btnOK, and lbl\_Content represent an image, an ImageControl, and a label located at the Canvas, respectively. ImageControl is a class inheriting the Control.

```
...
popupBG.Source = ...GetBit...Path(clsCommon.IMG_DIR + @"\...png");
btnOK.NormalImage = clsCommon.IMG_DIR + @"\...png";
btnOK.PressImage = clsCommon.IMG_DIR + ...png";
lbl_Content.Content = ...;
...
```

**Figure 12. Part of the Definition for PopupClass**

KeyPadClass extends a Window in order to create a user interface that looks like a telephone keypad. Part of the definition of this class is shown in Figure 13. The constructor of this class reads the designated xaml file and instantiates a canvas in which images, labels, and ImageControls are deployed to make a telephone keypad. Then the constructor assigns the prompt "Please input your telephone number" to the label.

```
public partial class KeyPadClass : Window
{
    ...
    public KeyPadClass(String _contID, String _contName)
    {
        InitializeComponent();
        ...
        lbPhoneNumber.Content = " Please input your telephone number ";
    }
}
```

```
...  
  
private void Window_Loaded(object sender, RoutedEventArgs e)  
  
...  
    keypadBG.Source = ...GetBitmap...Path(...IMG_DIR + @"\keypad\bg.png");  
    btnClose.Source = ...GetBitmap...Path(...IMG_DIR +  
@"\keypad\key_close.png");  
    btnSend.NormalImage = ...IMG_DIR + @"\keypad\btn_send.png";  
    ...  
    btn0.NormalImage = clsCommon.IMG_DIR + @"\keypad\0.png";  
    ...  
    btn9.NormalImage = clsCommon.IMG_DIR + @"\keypad\9.png";  
    ...
```

Figure 13. A Part of Our Definition of the KeyPadClass

#### 4. Experiments

Part of the Kiosk system was implemented. The main page of the system is shown in Figure 14. The nine boxes filled with images and texts shown in the figure are menu buttons. By touching these menu buttons, users can move to the service page associated with the button. For example, by clicking the 'TRAVEL' button, users can move to the "TRAVEL" page shown in Figure 15.



Figure 14. A Snapshot of the Kiosk Main Page



Figure 15. A Snapshot of the "Travel" Page (To See, to Eat, to Stay, to Play, to Buy)



Figure 16. A Snapshot of the "QR Code" \*age (Read QR Code with Your Smartphone)



Figure 17. A Snapshot of the Keypad Page (Type in Your Phone Number)

If the user touches the "QR code" button shown in Figure 14, then a pop-up window that tells the user how to use the system appears, as shown in Figure 16. Users can purchase local products using the kiosk. When the user selects a product item, a keypad window appears, as shown in Figure 17, so the user can type in his/her phone number. Then, the system sends the URL of the site that takes customer orders to the phone.

## 6. Conclusions

This paper introduced the design of a kiosk system for local area tourism. This system is integrated with the live broadcast and an online shopping mall. Part of the system has been implemented, and plans include implementing the whole system and launching the service soon.

## Acknowledgment

This work was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (NRF-2011-0006942), by 'Development of Global Culture and Tourism IPTV Broadcasting Station' Project through the Industrial Infrastructure Program for Fundamental Technologies funded by the Ministry of Knowledge Economy (10037393) and the Dongguk University Research Fund of 2015.

## References

- [1] J. Yim, "Review of the Kiosk Related Research Results," *Advanced Science and Technology Letters* 85 (ASTL 85). It is also Proceedings International Workshop Information Technology and Computer Science 2015, April 15-18, 2015, Jeju Island, Korea, (2015), pp.51-54.
- [2] Y. Hung, K. Chen, C. Yang, G. Deng, Y. Yeh and N. Wu, "Data mining for analyzing kiosk usage behavior patterns," *IEEE International Conference on Automation Science and Engineering*, (2014), pp. 1115-1120.
- [3] M. Park, D. Song, H. Kim and S. Shin, "Development of information architecture for user-centered hospital KIOSK," *3rd International Conference on Data Mining and Intelligent Information Technology Applications*, (2011), pp. 201-203.
- [4] S. Garriss, R. Sailer, S. Berger, R. Sailer, L. V. Doorn and X. Zhang, "Towards Trustworthy Kiosk Computing," *Eighth IEEE Workshop on Mobile Computing Systems and Applications*, (2007), pp.41-45.
- [5] K. Dipin, J. Bose and V. Vivek, "A kiosk based model for employment generation in rural areas," *IEEE Global Humanitarian Technology Conference - South Asia Satellite*, (2014), pp. 195-200.
- [6] W. Hyun, M. Huh, S. Kim and S. Kang, "Considerations on Audience Measurement Procedures for Digital Signage Service," *IJCA*, vol. 5 no. 2, (2012), pp.123-130.
- [7] S. Jung, Y. Chae, J. Shin, U. Baek and Seoksoo K., "Design of Context-Aware based Information Prevention for Digital Signage," *IJMUE*, vol.8 no.3, (2013), pp. 359-368.
- [8] W. Hyun, M. Huh, S. Kim and S. Kang, "Considerations on Audience Measurement Procedures for Digital Signage Service," *IJCA*, vol. 5 no. 2, (2012), pp.123-130.
- [9] J. Wang, Z. Zhang, Y. Zheng, L. Zuo and J. Kim, "A MultiTiers Service Architecture based Diabetes Monitoring for Elderly Care in Hospital," *IJMUE*, vol.8 no.3, (2013), pp.387-398.
- [10] S. A. Fedaghi, "Developing Web Applications," *IJSEIA*, vol. 5 no.2, (2011), pp.57-68.

## Author



**Jaegeol Yim** received the M.S. and Ph.D. degrees in Computer Science from the University of Illinois at Chicago, in 1987 and 1990, respectively. He is a Professor in the Department of Computer Science at Dongguk University at Gyeongju Korea. His current research interests include Petri net theory and its applications, Location Based Service, AI systems, and multimedia systems. He has published more than 50 journal papers, 100 conference papers (mostly written in Korean Language), and several undergraduate textbooks.