

Smart Brochure

(paperless brochure using beacon)

Hyeonsu Lim
Information System in HYU
Email: tatto_hs@naver.com

Jaemook Kang
Information System in HYU
Email: kjm8475@naver.com

Kiseong Kim
Information System in HYU
Email: rltjd1231@naver.com

Abstract—When you go to the art museum, you will find out several things to help you see the exhibition better, such as brochure, program books, and audio-guide, etc. For the big size of exhibitions supported by big art gallery or of famous artist, there would be no problem to prepare the goods mentioned before. However, there are a lot of artists who are trying to open an exhibition in small art gallery and students who are preparing the exhibition for the graduation, and they have a lot of problems to possess those kinds of goods. The purpose of this software project is to help them. We can provide many kinds of IoT services, such as the explanation of the exhibition, explanations of each art, audio-guide, and so on, by using beacon and the mobile application.

Keywords — beacon; bluetooth; iot; museum; brochure; smart phone; application;

| Roles | Name | Task description and etc. |
|---------------------|--------------|---|
| User | Kiseong Kim | Use the mobile application and get some data about exhibition |
| Customer | Kiseong Kim | Purchase this software service and offer the data about exhibition |
| Software Developer | Jaemook Kang | Develop the mobile application and back-end server |
| Development manager | Hyeonsu Lim | Manage team and project and make the every plan of the process of project |

I. INTRODUCTION

If you have only a little interest, you will find many art galleries and museums easily. There are over 100 exhibition halls in Seoul, which means that a lot of exhibitions we can enjoy are opening every day. In order to help the audiences enjoy these exhibitions better and feel better, those exhibitions are providing many items that will help people understand the exhibition better. The audiences need to pay extra costs to buy or rent those goods.

However, not all the exhibitions provide those services. In the case of famous artists and big art museums, a lot of people will visit there and see the works, so there will be many people who will pay extra costs to get the chance to inspect better. Therefore, famous artists and big art museums can make extra incomes by making lots of guide goods and sell them. On the other hand, let's think about obscure artists and the university students who are preparing the exhibition for the graduations. They can hardly provide those items for their audiences. The first reason is the cost. The cost to make audio guide, brochure, and program books are not low, so the obscure artists and the students cannot afford it.(about 2,000,000 won)

The second reason is the difference of peoples interests. Practically, it is really hard for the small exhibition halls to have peoples attention. Even though they spend money more and produce the goods for their exhibitions, there will not be that many people who will pay extra money to buy or rent the goods.

We thought that people who are trying to open small exhibition and students who are preparing an exhibition doesnt want the extra income but the interests of people. We wanted to provide them more chances for the amateur artists to introduce themselves and to appeal their works to people. Therefore, we want to fulfill their requirements through SMART BROCHURE. Artists can provide good service to their audiences with less cost, and audiences dont have to pay extra money and get the chance to use many services that artists want to provide, only by installing an application. Though there is a similar application, named Jeonsi bogo, this service is only for the big exhibitions, so we still need to develop new system.

II. INDEX

I. Introduction

II. Requirement

2.1 The Environment for Using Beacon

2.1-1 Bluetooth 2.1-2 OS

2.2 Server requirement

2.2-1 How to send information 2.2-2 Need for

back-end server

2.3 User requirement 2.3-1 Accuracy

2.3-2 Unimportant data 2.3-3 Push alarm

2.3-4 Exhibition list

2.3-5 Map

2.3-6 Location of work the exhibition 2.3-7

Work button

2.3-8 Work picture

2.3-9 Text box1

2.3-10 Text box2

2.3-11 Voice button

2.3-12 Previous exhibition

2.3-13 Delete function

2.3-14 Reporting problem

2.4 Customer requirement

III. Develop environment

3.1 Choice of software development platform

3.1-1 Which platform and why?

3.1-2 Which programming language and why?

3.2 A cost estimation

3.3 Information of development environment

3.4 Using commercial cloud platform

IV. Specification

4.1 Modeling for Specifications

4.2 Prototype for Specifications

4.3 Specifications for front-end application

pages

-BLE searching outside the application

-My history page

-Information page

-Searching beacon page

-More page

4.4 Specifications for Server

V. Architecture Design and Implementation

VI. Use Cases

III. REQUIREMENTS

A. Requirement for The Environment for Using Beacon

1. Bluetooth module
 - Users need to have Bluetooth 4.0 or higher module.
2. Smartphone OS
 - 1) ios 7 or higher
 - 2) Android 4.3 or Higher
 - 3) OSX mavericks 10.9

B. Requirement for Server

1. How to send information
 - 1) The beacon installed in the gallery will send the id code to the smartphone which has the application, and the smartphone will send that id code and the customer information to the server.
 - 2) Lastly, the server will send the appropriate data, which is decided by combining the gallerys information and the customer information, to the smartphone.
2. Needs

We need to develop a back-end server that stores the information and judge the id code sent by beacon.

C. Requirement for User

1. Accuracy : Users want more precise sensor when they use beacon technology
2. Unimportant data : Users dont want information which is not necessary
3. Push alarm
 - 1) Push alarm is popped up on the users smart-phone when passes by an exhibition.
 - 2) The user can get some information by push the yes button.
4. Exhibition list
 - If push the button, you can see list of exhibitions you watched. Latest exhibition is

located in top of the list.

5. Map
 - 1) Map provides a course how to see the exhibition.
 - 2) If users push one of the mutton, Smart Brochure gives users the map of the exhibition.
6. Location of work in the exhibition
 - In the map, users can see some buttons which indicates work name and where works are.
7. Work button
 - If users push button on the map, they can get screen which has information about the work.
8. Work picture
 - In the information screen, picture of the work is located left-top.
 - Users can check on whether explanation corresponds to the work by picture.
9. Text box1
 - Text box1 is located next to work picture. There are work name, artist name, and techniques in the box.
10. Text box2
 - Text box2 has explanation of the work. If explanation is so long, uses can use scroll technique.
11. Voice button
 - In the text box2, users can use voice button. If users push the button, they can hear explanation of the work.
12. Previous exhibition
 - This application can saved data about previous exhibition.
13. Delete function
 - If users want to delete previous exhibition information, they can delete the data.
14. Reporting problem
 - When use Smart Brochure, users can find some problem. In this situation, users can report this problem to developer.

IV. DEVELOPMENT ENVIRONMENT

A. Choice of software development platform

1. Which platform and why? (e.g., Windows, Linux, Web, or etc.)

- 1) Windows for android application developing
- 2) Mac OS for iOS application developing.

2. Which programming language and why?

- 1) java for android application developing

Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. As of 2015, Java is one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers. Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them.

2) objective-C and Swift for iOS application developing.

Objective-C is a general-purpose, object-oriented programming language that adds Smalltalk-style messaging to the C-programming language. It is the main programming language used by Apple for the OS X and iOS operating systems, and their respective application programming interfaces (APIs), Cocoa and Cocoa Touch. Objective-C's features often allow for flexible, and often easy, solutions to programming issues. Delegating methods to other objects and remote invocation can be easily implemented using categories and message forwarding. Swizzling of the isa pointer allows for classes to change at runtime. Typically used for debugging where freed objects are swizzled into zombie objects whose only purpose is to report an error when someone calls them. Swizzling was also used in Enterprise Objects Framework to create database faults. Swizzling is used today by Apple's Foundation Framework to implement Key-Value Observing.

Swift is a multi paradigm, compiled programming language created by Apple Inc. for iOS and OS X development Swift is designed to work with Apple's Cocoa and Cocoa Touch frameworks and the large body of existing Objective-C code written for Apple products. Swift is intended to be more resilient to erroneous code ("safer") than Objective-C, and also more concise. It is built with the LLVM compiler framework included in Xcode6, and uses the Objective-C runtime, allowing C, Objective-C, C++ and Swift code to run within a single program, but its proprietary nature may hinder Swift's adoption outside the Apple ecosystem.

3) JSON

JSON is an open standard format that uses human-readable text to transmit data objects consisting of attribute-value pairs. It is used primarily to transmit data between a server and web application, as an alternative to XML. Although originally derived from the JavaScript scripting language, JSON is a language-independent data format.

B. Provide a cost estimation for your built.

(including any purchase of software/hardware)

1. cost for server : 1 year for free. And after 1year, there will be additional prices. We predict maybe about 1,000 people will use our service, and DAU(Daily Activity User) will be 300 around. So we will use t1micr instance (AWS), and its prices are about 30permonth.Somaybethere will be additional 360 per year.

2. cost for beacon : We will use the RECO beacon. Reco beacon is authorised by iBeacon. Its prices are 229,000 (10 pieces).

3 cost for developer :

C. Provide clear information of your development environment.

(e.g., version of software, OS version, your computer resources)

1. iOS develop

1) Mac : OS X Yosemite ver 10.10.1

2) iOS : iOS 8.3

2. Android develop

1) Windows : Windows7 ultimate

2) Android OS : Android 4.4 kitkat

D. Using any commercial cloud platform (e.g., Amazons EC2) is definitely a BONUS.

1. We will use Apache Web Server because it is the world's most widely used web server software. As of June 2013, Apache was estimated to serve 54.2percent of all active website and 53.3percent of the top servers across all domains. (The most important reason is we have already apache web server.)

Apache Web Server has some features:

- easy and fast customizing using module.
- It can handle many traffic easily.
- It can control web server more delicately
- It is tested enough, so it is very stable.

2. Software in use

Any existing software or algorithm in use? (doing a similar task as your proposal; provide a proper reference if there is any)

1. Trello :

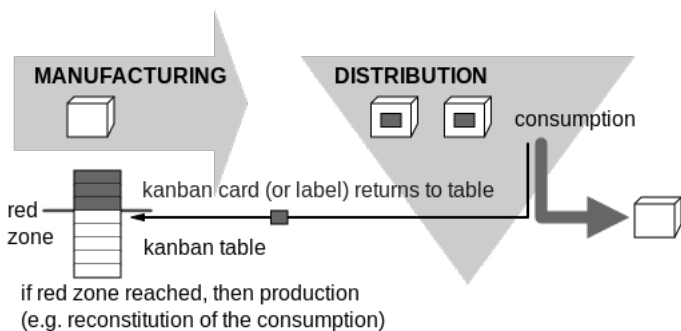


Fig. 1. Kanban Paradigm

Trello is a free web-based project management application. Trello uses the kanban paradigm for managing project. Kanban is a scheduling system

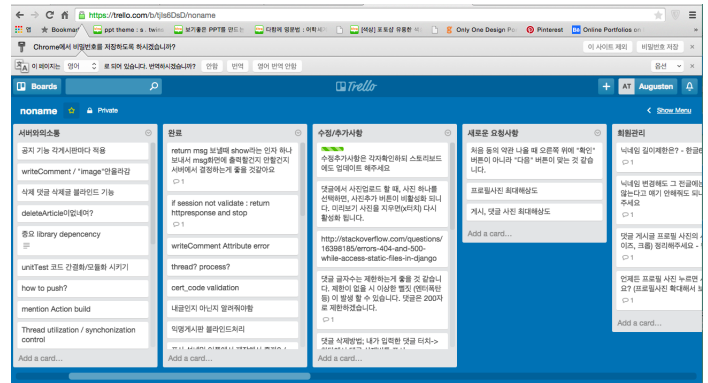


Fig. 2. trello

for lean and just-in-time (JIT) production.

Projects are represented by boards, which contain lists (corresponding to task lists). Lists contain cards (corresponding to tasks). Cards are supposed to progress from one list to the next (via drag-and-drop), for instance mirroring the flow of a feature from idea to implementation. Users can be assigned to cards. Users and boards can be grouped into organizations.

There is a little similar software in Korea, GO But this software service is only for big exhibition so small exhibition artist or students cant use that services.

3. Task distribution (If you want, you can provide this later at the next phase - design)

Which member is responsible for what?

V. SPECIFICATIONS

A. Modeling for Specifications

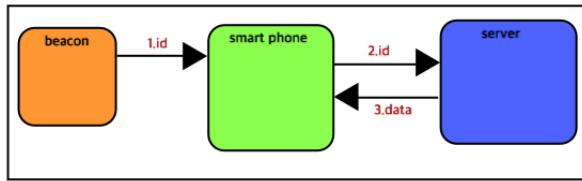


Fig. 3. basic structure

Beacon sends a specific ID value to the smart phone, when smart phone comes into its signal area. Then smart phone application recognizes this ID value and sends this value to server. Server which has this ID value check the location of beacon. After that, server sends the information or data about exhibition to smart phone.

B. Prototype for Specifications

Our application is divided into two parts. One is server side with Amazon Web Service EC2 and Ruby on Rails. The other part is client side acting at the smart phone. Client side, smart phone application, is structured by objective-C (iOS application), and java(Android application)

C. Specification for front-end application pages

0) BLE searching outside the application



Fig. 4. Information Page02

[BLE]

In our application, there is the service class [SearchBLE.java] which is searching the [BLE]. This service searches the BLE, if the bluetooth module on the smartphone is on. BLE(Bluetooth low energy) is a wireless personal area network(PAN) technology. It is designed and marketed for applications in the healthcare, fitness, security, home entertainment industries, and [beacon].

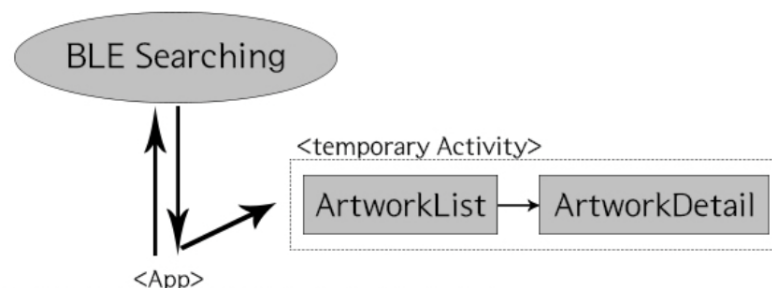


Fig. 5. how temporary activity is operating

If user let bluetooth module [on], smartphone will be searching BLE automatically. That is, it is searching beacon signal. If it finds beacon signal, smartphone get [beacon id]. And smartphone sends this beacon id to server, then server sends to smartphone the all data about exhibition which is stored at that beacon id. After, every data about exhibition is downloaded to smartphone. This downloaded data is showed through [Temporary

Activity].

The first page has the brief explains for exhibition and the list of artworks (similar with [My History Page02]). Among these artworks, if you pick one, you can show the details of that artworks such like photos and detail explanations (similar with [My History Page 03]).

You have to know this [Temporary Activity] is totally different page with main application. This is [only] temporary. This activity cannot access to main application, and also main application cannot access this activity neither. This activity is only for showing the data from the server. And in main application, not every data showed at temporary activity is stored. Only the name of exhibition and the downloaded date are stored.

1) My history page

| My Exhibition | | |
|----------------------------|------|---------|
| Exhibition 01 / 2015.05.25 | | |
| Exhibition 02 / 2015.05.30 | | |
| Exhibition 03 / 2015.05.31 | | |
| Exhibition 04 / 2015.06.04 | | |
| | | |
| My | Info | Setting |

Fig. 6. BLE

[My History Page01]

This is the first page of application. And you can

access this page by tab menu under the display. There is the list of exhibitions which user have already seen.

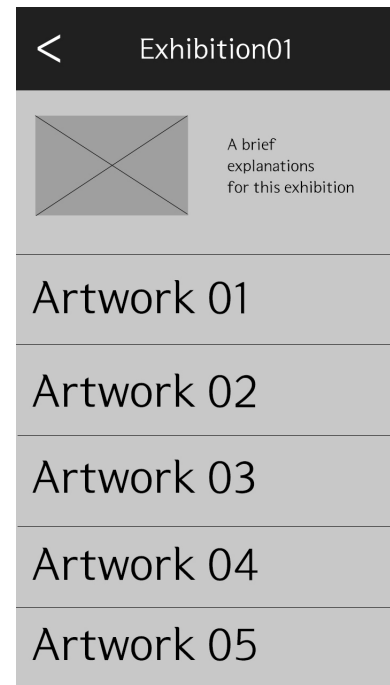


Fig. 7. My History Page02

[My History Page02]

This is the detail page of the exhibition01. To access this page, applications has to communicate with the back-end server. Server will send the data of Exhibition 01 which the user already downloaded by beacon communication at that Exhibition. On Navigation bar, there is the name of the exhibition. Under the navigation bar, the left side of the first cell, there is main image of exhibition. And Next to the main image, the right side of the first cell, there is the brief explanation for this exhibition. There will be the information of this exhibition such like the theme of exhibition, the name of exhibition center, address of exhibition center. Below First cell, there is the list of artwork which is displayed in this exhibition. user can see the detail information about the artwork such as image of artwork, text explanation of artist, or voice-audio explanation.

[My History Page03]

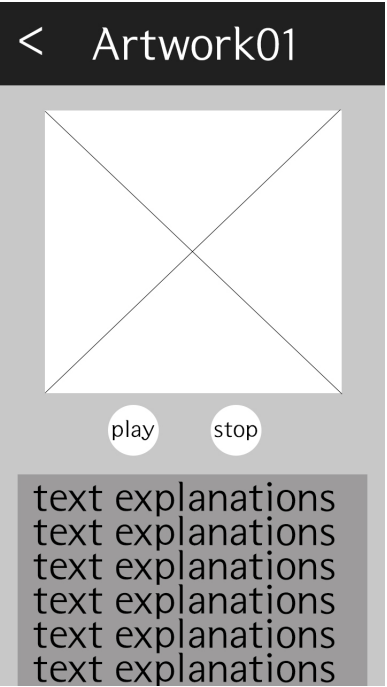


Fig. 8. My History Page03

This is the detail page of artwork. On the Navigation bar, there is the name of artwork. Under the Navigation bar, there is the Image of the artwork(If user touch the small image, the pop up window will appear, and user can see the big size image). Below the artwork image, there is the play and stop button. This button is for voice-audio explanation. Voice-audio explanation is not for every artwork. We offer the voice-audio explanation only for the artwork that artist want, and artwork that artist offer the voice-audio explanation data. So if there is the voice-audio explanation, there will be play and stop buttons. And if there is no voice-audio explanation, the play and stop buttons will not exist. Under the Image and buttons, there is the text explanations that artist offer.

2) Information page



Fig. 9. Information Page01

[Information Page01]

user can access the Information page by touching tab menu button under the screen. When user touch the [Info] button under the display, every data is downloaded from server. This page is for noticing the exhibition. There is the list of the exhibitions which is on going now or which will be started. user can check the detail information about the exhibition that user like by touching the name of the exhibition.

[Information Page02]



Fig. 10. Information Page02

This page shows the detail information of the exhibition. user can move to the exhibition list page by back button on the navigation bar. On navigation bar, there is the name of the exhibition. Below the navigation bar, user can find the brief information about the exhibition such as the main image of the exhibition, theme of the exhibition, and map or address of exhibition.

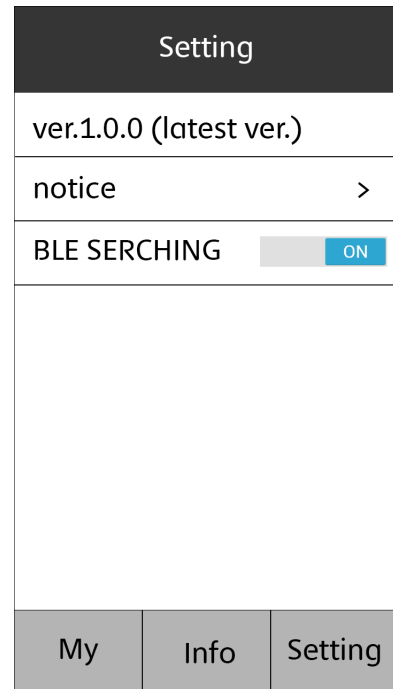


Fig. 11. Setting Page

[Setting Page]

In Setting page, there will be the additional function. For example, there will be the on/off button that control the searching BLE. If the user makes the button-state [on], smartphone will search beacon signal and get beacon id (explained and [0)BLE searching outside the application]). And if user makes the button-state [off], smartphone will not search any beacon signal outside the application. And there will be another board for inform the latest version or notice etc.

3) Setting page

D. Specifications for Server

VI. ARCHITECTURE DESIGN AND IMPLEMENTATION

A. Directory organization

Image

B. Code analysis

1) Splash

-purpose : To show specific page during downloading date before program starting.

-functionality : Splash is a kind of loading screen. The function of splash is gaining time to get data which is necessary for program. Additionally, we can promote our application name during splash time.

-location of source code : Smart-Brochure/appsrc/java/com/example/jay/smart_brochure/Splash

-class components

a. 'onCreate' method is implemented firstly in activity class like main() in JavaSE. In the onCreate, setContentView shows basic layout which is used to compose main page. By using postDelayed in handler, this method can do function how long loading page appear.

b. onCreateOptionsMenu method does initializing option menu of activity.

c. In onOptionsItemSelected method, Handle action bar item clicks here. The action bar will automatically handle clicks on the Home/Up button, so long as you specify a parent activity in AndroidManifest.xml.

-how/why you used it : If you want to implement some program, the program needs time to get source which is vital for program. Splash does function which gains time getting data.

2) SearchBle

-purpose : To search beacon which uses Bluetooth Low Energy(BLE).

-functionality : SearchBle is implemented on application background. Usually almost classes are inherited from Activity, but SearchBle is inherited from Service. Because searching function isn't seen in the smartphone screen but implemented background. Moreover, Users can turn on/off Bluetooth searching function by this class.

-location of source code : Smart-Brochure/appsrc/java/com/example/jay/smart_brochure/SearchBle

-class components

a. BroadcastReceiver has function which receives intent type message. It is used to realize operation when application completed specific task. If beacon send signal, starts BLE searching when Bluetooth is on by using Bluetooth manager.

b. Timer is basic class in Android. By using Timer class, we can set schedule when searching starts and how long search beacon. When we use Timer class, Android appreciate '1 = 1 millisecond'. Therefore if we want to set searching time 3 seconds, we have to put value 3*1000.

c. onLeScan's main function is comparison.

This method compares searched beacon device and information in the database. When we search beacon, there are so many beacon even if don't need. If we don't comparison task, we have to send all beacon data which are not necessary to server. That can provoke data waste. By using this method, we can send appropriate beacon data to server. Additionally, if an appropriate beacon data sent to server, we stop searching the beacon because if we don't stop searching, push alarm will come continuously while audiences watch the exhibition.

d. NotificationCompat is function which makes push alarm. If you turn on Bluetooth and application, push alarm notifies that message "Brochure come", whether getting the brochure or not and go to Push_Clicked.

-how/why you used it : We have to use Timer class because if we don't use this class, our application will search continuously. That means that smartphone battery consume rapidly. SearchBle is core function because we use beacon device.

3) My

-purpose : My shows an exhibition list that users visited before to customer.

-functionality : First tap button on the main page and if you execute SmartBrochure, you can see my page firstly. If you click one of the list, list of the exhibition work appears.

-location of source code : Smart-Brochure/appsrc/java/com/example/jay/smart_brochure/My

-class components

a. sendId method is the main method in My class. If SmartBrochure got the beacon data which is correct with server data, this method send the beacon id to server. After server gets the beacon data, sends the exhibition data (exhibition name, list of work, image and so on) to SmartBrochure.

b. customAdapter's main function is giving data to

list view. In the customAdapter, getView method implement above function.

-how/why you used it : My is first page when user implement Smartbrochure. They can see list and My will provide all of exhibition information which users want. As a result, users can collect special their brochure list.

4) My_Clicked

-purpose : My_Clicked shows the list of art work. The list includes name and image of each art work.

-functionality : My_Clicked class must need communication with server. If application sends the id value which is in the database, after that server sends information about the exhibition which is correct with id to application.

-location of source code : Smart-Brochure/appsrc/java/com/example/jay/smart_brochure/My_Clicked

-class components

a. ImageLoaderConfiguration is used for implanting image. In the application, image file doesn't store anywhere. However, in order to use work image, we can use server. When we need some image file, we get URL which has image from server. In other words, just borrow image file when we need. If we go out of the page which has image, image file is removed from application.

-how/why you used it : We can't store all image file in the application because application can't accept many image file. So we use server to use adequate image, because image data is very important in exhibition information.

5) Push_Clicked

-purpose : Push_Clicked also shows the list of art work. The list includes name and image of each art work.

-functionality : Push_Clicked also class must need communication with server. However Push_Clicked is different from My_Clicked. If application sends the id value searched from beacon not database, after that server sends information about the exhibition which is correct with id to application.

-location of source code : Smart-Brochure/appsrc/java/com/example/jay/smart_brochure/Push_Clicked

-class components

a. ImageLoaderConfigurationbition is also used in Push_Clicked. In the application, image file doesn't store anywhere. However, in order to use work image, we can use server. When we need some image file, we get URL which has image from server. In other words, just borrow image file when we need. If we go out of the page which has image, image file is removed from application.

-how/why you used it : Push_Clicked is very similar with My_Clicked. However they are different in sending id value to server. My_Clicked is focus on existing exhibition information. In contrast, Push_Clicked is focus on new information from beacon search. Because of this difference, we divided them into two classes.

6) Explanation

-purpose : Explanation class has function which gives the detailed description about art work. Customers can see image and text in this step.

-functionality : When you click one of the art work list(in the My_Clicked or Push_Clicked), Explanation class is implemented. In this process, communication with server is very essential like My_clicked or Push_Clicked. Explanation also use image by using URL from server.

-location of source code : Smart-Brochure/appsrc/java/com/example/jay/smart_brochure/Explanation

-class components

a. In the Explanation class, onCreate method's main function is setting image and text about art work. They are given from server.

b. setExplanation method uses some function like DisplayImageOptions and ImageLoaderConfiguration. By using them, we can set image size and etc.

-how/why you used it : Explanation class is the most informative section in our application. Like My_Clicked, We use server to use adequate image, because image data is very important in exhibition information.

VII. USE CASES

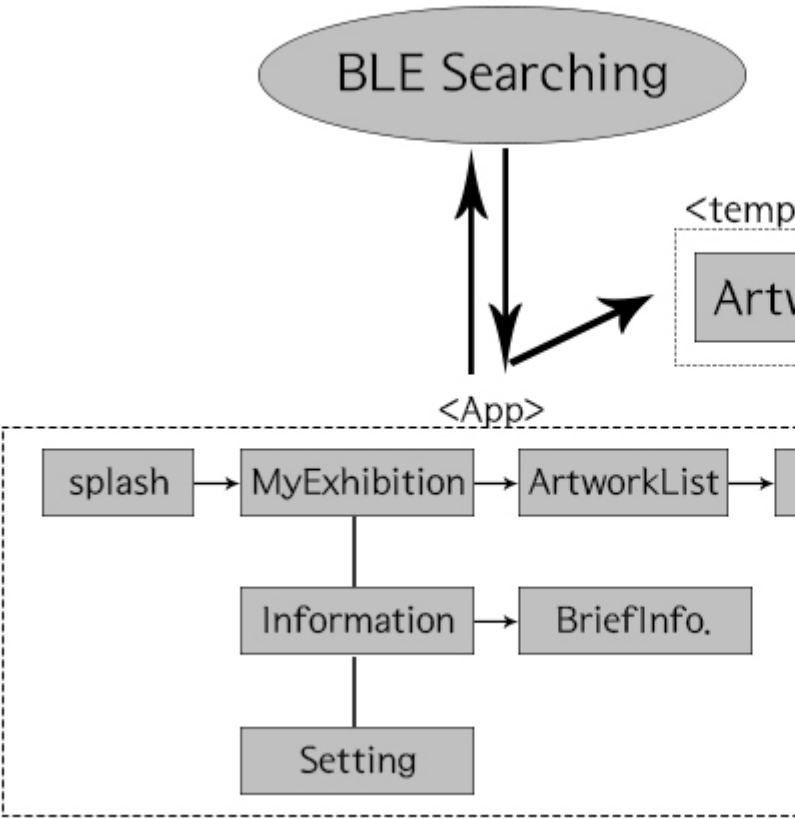


Fig. 12. Beacon Page03

6-1) BLE searching and push notice

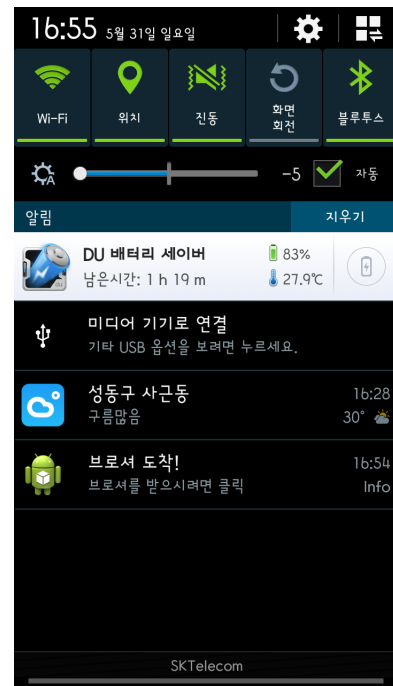


Fig. 13. BLE searching and push notice

If User turn on the bluetooth module, smartphone finds the BLE signal, and then Server sends to notice that information data about Exhibition is downloaded. If user touch the push notice, then temporary activity is open.

6-2) Temporary Activity01

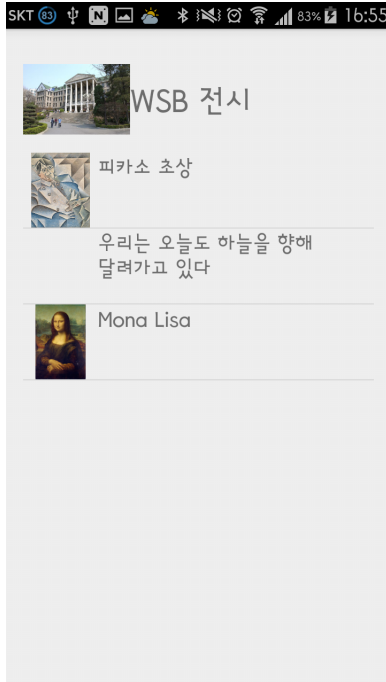


Fig. 14. Temporary Activity 01

This is the page when temporary page is open. It shows the information received from server. Upper-side of the page, there are the name and main image of exhibition. Under, there are the artwork lists which are displayed at the exhibition now user is seeing.

6-2) Temporary Activity02

If the user touch one artwork, he can watch the detail of artwork. Of course every data is received from server. There are image, name, and detail descriptions of the artwork.

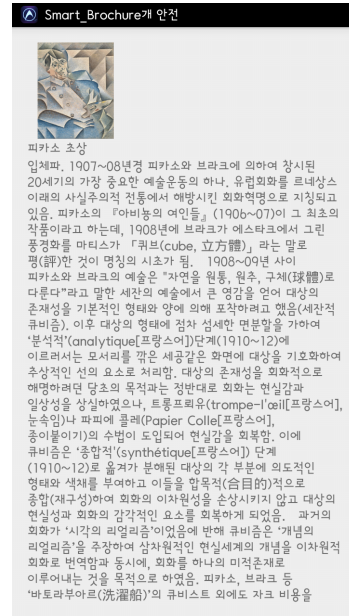


Fig. 15. Temporary Activity02

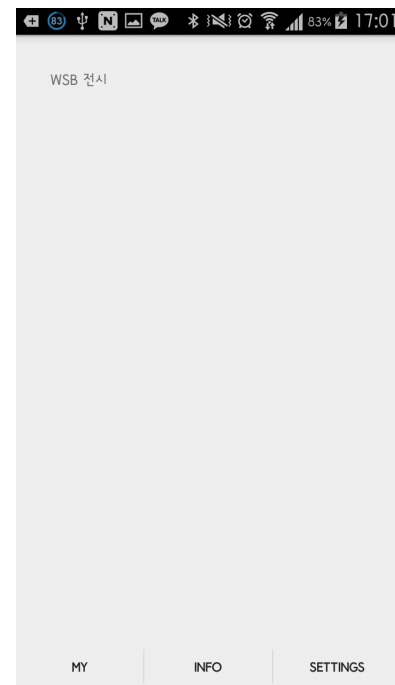


Fig. 16. My History01

6-3) Main Application - My History Page01

User will see this page when he run the application.

It is the first page of main application. First, there is tab menu under the screen. By this tab menu, user can move to page he want to see. This page is for user own. The data which he received from server and saw through temporary activity is stacked at this page.

6-3) Main Application - My History Page02

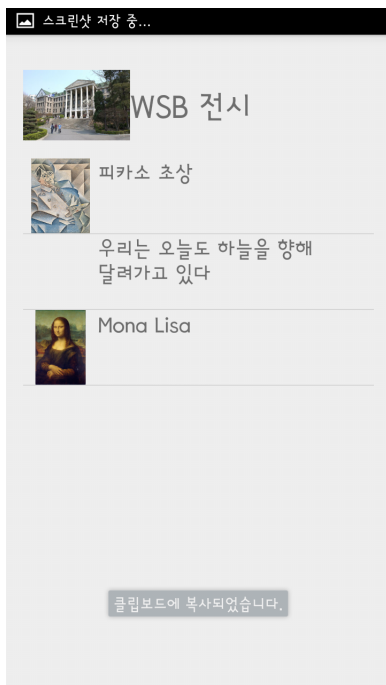


Fig. 17. My History02

This page shows the detail information about the exhibition. User can see the brief details about the exhibition theme, the Artist, and the list of artwork which is displayed on that exhibition.

6-3) Main Application - My History Page03

If user touch any artwork at the My History Page01, user can see the detail information about that artwork like this page.

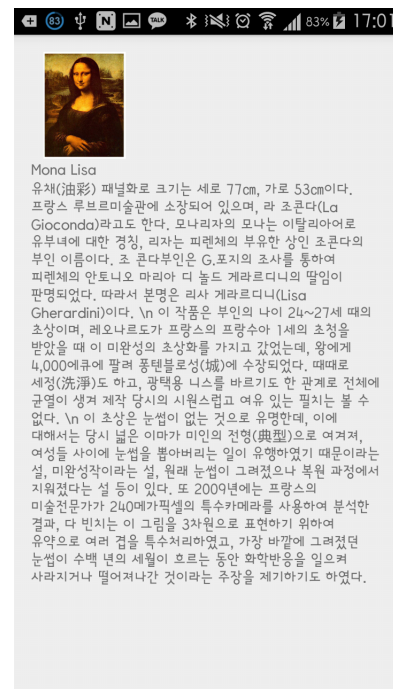


Fig. 18. My History03

6-3) Main Application - Information Page

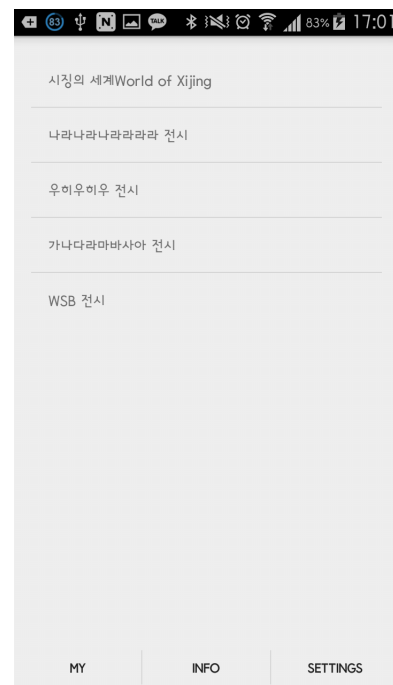


Fig. 19. Information Page01

This is the second menu page of the application. User can move to this page from the other using tab menu under the screen. This page is for inform about the other exhibition. On the screen, there will be the list of exhibition that user has not seen yet. If user touch the INFO menu from other page, or swipe down the list, then application connect to the server and get the data of the list of the exhibition. It means, user can refresh the list whenever he wants.



Fig. 20. Information Page02

If user select one exhibition, he can see the detail information about that exhibition such like the main image of the exhibition, information about the artist, the address of the exhibition and so on.

6-3) Main Application - Setting Page

User can turn on or off the function searching the BLE. - Turn on

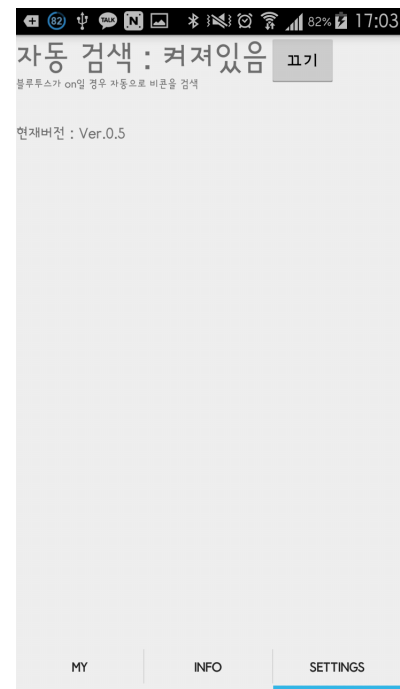


Fig. 21. Setting Page01

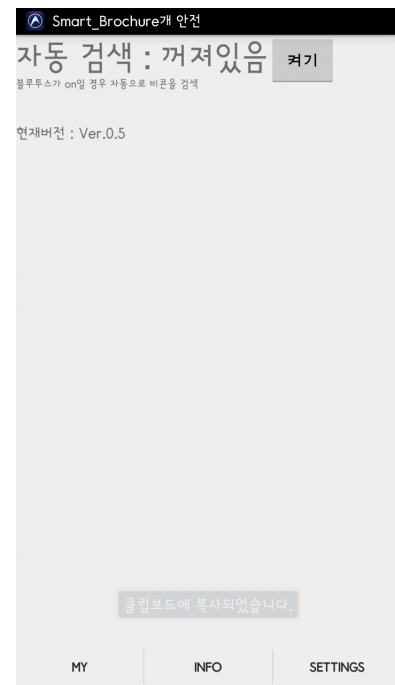


Fig. 22. Setting Page02

User can turn on or off the function searching

the BLE. - Turn off

.