**작품 요약서**

* **Basic Data**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **작품명** | 찍Go | | | **지원분야** | Software |
| **개발인원** | 1 명 | **본인역할** | PL | **개발기간** | 2013.10  ~2013.12 |
| **개발환경** | Platform : Android 4.1.2 , Jelly Bean | | | | |
| **개발언어** | Java, xml | | | | |
| **개발툴** | Eclipse, Android SDK | | | | |
| **작품소개**  **(요약)** | 여행을 가면 사람들이 ‘흔적 남기기’를 할 때 항상 하는 일 중 하나가 바로 스탬프 투어입니다. 하지만 스탬프 투어를 할 때 가지고 다니는 종이는 훼손이나 분실의 위험이 있고, 정해진 시간에 도착하지 않으면 종이에 스탬프를 찍기가 힘든 단점이 있습니다.  이 프로그램은 위와 같은 문제점을 해결하기 위해 종이의 단점인 훼손과 분실의 위험을 제거하고 시간에 구애 받지 않게 도장을 찍어주어 여행에 대한 추억을 간직할 수 있는 프로그램 입니다.  잘쓴거 같다 .. ㅋㅋ | | | | |
| **작품내용** | 스탬프 투어란 각 지역에 있는 관광명소들을 돌며 해당 관광안내소에서 스탬프를 종이에 받는 것입니다. 우리나라에서는 부산과 서울, 대구, 제주, 경주, 군산 등에서 스탬프투어 프로그램을 운영하고 있습니다.  하지만 스탬프를 종이에 받기 때문에 종이 특성상 손상될 우려가 있어 여행에 대한 추억을 소중하게 간직하기가 힘듭니다. 또 스탬프 투어를 할때 정해진 시간이 아니면 도장을 찍을 수 없는 상황도 생겨서 여행을 다녀온 것에 대해 흥미를 잃어버리기도 합니다.  이러한 문제점을 해결하고자 기존의 스탬프투어 방식 대신 증강현실을 이용하여 현실감은 살리고 종이 특성상 손상될 우려가 있고 잃어버릴 수 있는 단점을 보완하고, 시간에 구애 받지 않는 여행으로 소중한 추억을 간직할 수 있도록 해주는 스마트폰 어플리케이션을 만들게 되었습니다.  이 프로그램의 메뉴로는 첫째로 내가 여행을 가서 찍은 도장을 확인할 수 있는 **‘나의 도장 보기’** 메뉴가 있습니다. 나의 도장 보기에서는 여행을 갔을 때 도장판을 지역별로 볼 수 있습니다. 둘째로는 **‘도장 찍기’** 메뉴가 있습니다. 도장 찍기 메뉴에서는 증강현실로 구현된 도장 마크가 카메라 서피스뷰에 보여집니다. 도장을 터치하면 도장이 찍는 애니메이션이 나오고, 나의 도장 보기에서 내가 찍었던 도장을 볼 수 있습니다. | | | | |
| **개발내용**  **(본인 구현부분)** | 이 프로그램은 웹 서버를 사용하여 데이터베이스로부터 현재 위치에 알맞은 도장 정보를 가져옵니다. 가져온 도장 정보는 안드로이드 기반의 클라이언트에서 **Mixare Engine** 라이브러리를 이용하여 SurfaceView 위에 도장 이미지로 나타내어 줍니다. 또 안드로이드 기반의 클라이언트에서 SharedPreference로 저장한 사용자의 고유 정보로 자신이 여행을 가서 찍었던 도장을 서버를 이용하여 데이터베이스에서 가져와 볼 수 있습니다.  SurfaceView 에 나타나는 도장 이미지는 Mixare Engine 라이브러리를 이용하여 표현할 수 있습니다. 도장 이미지를 터치 하였을 때는 표시되고 있는 도장의 정보를 사용자 정보와 함께 데이터베이스에 저장합니다. 그러면 사용자는 나의 도장 보기 메뉴에서 도장판에 도장이 찍혀있음을 확인할 수 있습니다.  안드로이드 기반 클라이언트는 현재 자신의 위치에 있는 위도와 경도 정보를 서버에 전송합니다. 서버는 클라이언트로부터 받아온 요청을 **Servlet**으로 응답할 수 있도록 만들었습니다. Servlet으로 요청을 받은 서버는 위도와 경도 정보로 현재 위치에서 불러올 수 있는 도장의 정보들을 데이터베이스로부터 받을 수 있도록 하였습니다.  데이터베이스에서 도장의 정보를 불러오거나, 사용자의 도장을 저장하는 것은 MyBatis 프레임워크로 구현하여 객체 단위로 저장하고 불러올 수 있도록 유연하게 만들었습니다. 데이터베이스에서 받아온 객체 단위는 ArrayList타입으로 되어있는데, Mixare Engine은 JSON 형태로 도장의 정보를 받기 때문에 Servlet 에서 **JSON Array** 형식으로 바꿔서 클라이언트로 도장의 정보를 보낼 수 있도록 만들었습니다.  안드로이드 기반 클라이언트에서 도장을 터치하였을 경우 도장의 정보는 SharedPreference로 참조된 사용자의 정보와 함께 데이터베이스에 저장이 됩니다. 도장 보기를 눌렀을 때에는 지역별 테이블에 저장되어 있는 사용자 정보와 비교해서 데이터베이스로 불러 오도록 하였고, 찍었던 도장을 도장판에 보여주기 위해서 xml의 android:tag 를 이용하여 View 의 형태를 Visible로 바꿔서 찍었던 도장을 볼 수 있도록 만들었습니다.  데이터베이스에는 관광명소를 왔을 때 SurfaceView 위에 도장으로 보여주기 위한 관광 명소들의 정보가 있는 테이블과 도장을 터치 하였을때에 저장이 되어 나의 도장보기에서 도장 정보를 가져올 수 있는 지역별 테이블이 있습니다.  문단문단 형식이 잘 나누어져 있는거 같다. 문장의 단위가 길지 않고 주어 동사 목적어 등의 구분이 명확한거 같다. 잘쓴거 같음 | | | | |
| **기타** | 제 프로그램은 시간에 구애 받지 않는 여행을 할 수 있는 것이 가장 큰 장점이라고 생각합니다. 그리고 제 프로그램이 시,군별과 연동하게 된다면 지역 홍보 효과와 여행 문화에 큰 도움이 될 것입니다.  여기 좀더 길게.. ㅋㅋ 상세하기 달랑 3줄이니 성의의 적음이 보임 ^^ | | | | |

* **System Architecture**

|  |
| --- |
|  |
| **그림 1 – System Architecture** |
| 제 프로젝트는 Mixare Engine 라이브러리를 이용하여 도장 이미지를 Surfaceview에 표시합니다. 데이터베이스에 저장되어 있는 도장 정보들은 웹 서버를 통해 안드로이드 기반의 클라이언트로 전달되고, 클라이언트로부터 입력 받은 도장 정보는 서버를 통해 데이터베이스에 저장이 됩니다.  **데이터베이스**에는 관광 명소들의 위도와 경도 정보를 담고 있는 Stamp Mark Table 이 있고, 사용자가 찍은 도장을 지역별로 보관하기 위한 Place Table 이 있습니다.  **클라이언트**에서는 Mixare Engine 에서 가속도센서와 자기센서를 이용하여 방향을 알아내고, SurfaceView 에 비추면 현재 위치에 맞게 도장 이미지가 보여집니다. 또한 클라이언트는 자기가 찍은 도장을 표시하기 위해 사용자 정보가 있습니다.  **웹 서버**에서는 Servlet이 실행되어 클라이언트로부터 현재 위치 정보를 입력받아 JSON Array로 통신을 할 수 있도록 하였고 데이터베이스와 유연한 접근을 위해 MyBatis 라는 프레임워크를 사용하였습니다. |

* **Software/Hardware Architecture**

|  |
| --- |
|  |
| **그림 2 – Software Architecture ( SurfaceView initialize , Request )** |
| 제 프로젝트는 Mixare Engine 라이브러리를 이용하여 도장 이미지를 Surfaceview에 표시합니다. 데이터베이스에 저장되어 있는 도장 정보들은 웹 서버를 통해 안드로이드 기반의 클라이언트로 전달되고, 클라이언트로부터 입력 받은 도장 정보는 서버를 통해 데이터베이스에 저장이 됩니다.  **데이터베이스**에는 관광 명소들의 위도와 경도 정보를 담고 있는 Stamp Mark Table 이 있고, 사용자가 찍은 도장을 지역별로 보관하기 위한 Place Table 이 있습니다.  **클라이언트**에서는 Mixare Engine 에서 가속도센서와 자기센서를 이용하여 방향을 알아내고, SurfaceView 에 비추면 현재 위치에 맞게 도장 이미지가 보여집니다. 또한 클라이언트는 자기가 찍은 도장을 표시하기 위해 사용자 정보가 있습니다.  **웹 서버**에서는 Servlet이 실행되어 클라이언트로부터 현재 위치 정보를 입력받아 JSON Array로 통신을 할 수 있도록 하였고 데이터베이스와 유연한 접근을 위해 MyBatis 라는 프레임워크를 사용하였습니다. |

* **Software/Hardware Architecture**

|  |
| --- |
| **ddd2** |
| **그림 2 – Software Architecture ( SurfaceView Initialize, Server Request )** |
| Main Activity에서 도장 찍기 버튼을 click 하면 Mixare Engine의 MixView Activity로 전환하고 증강현실을 표현하기 위해 SurfaceView로 Camera의 open 함수로 화면을 비춥니다. 그리고 증강현실을 표현하기 위해 방향을 표현할 때 쓰이는 자기 센서와 가속도 센서들의 변하는 값을 잡아내기 위해 OnSensorChanged 에서 계속 값을 업데이트 합니다.  MixView 클래스 안에는 증강현실을 표현하기 위한 여러 클래스들이 있는데 그중에서DataView 는 Location 클래스로 현재위치의 위도와 경도를 알아냅니다. DataView 클래스에서 Location 클래스로 현재 위도와 경도를 가져오면 Mixare Engine은 현재 위치에 있는 도장 정보를 가져오기 위해 Thread로 서버와 통신을 하게 됩니다.  Mixare는 Runnable 인터페이스를 구현한 Thread를 이용하여 URLConnection 클래스를 통해 서버와 통신을 합니다. URLConnection 은 getInputStream 함수를 호출하여 현재 위치의 위도,경도,고도 등의 정보를 담아서 서버로 전송합니다.  URLConnection에서 요청을 받은 서버는 Service에서 Get Method 임을 확인하고 Get Method를 호출합니다. 서버는 클라이언트 기반 안드로이드에서 보낸 현재 위치인 위도와 경도 정보를 받습니다. |

|  |
| --- |
| **ddd3** |
| **그림 3 – Software Architecture ( Get datas from Server )** |
| 안드로이드의 요청을 보고 서버는 위도와 경도를 입력 받습니다. stamp\_mark 테이블에는 관광 명소의 정보가 저장이 되어있는데, 현재 위치에서 근처에 있는 도장의 정보를 가져오기 위해서는 현재 위치인 위도, 경도에 일정한 범위를 주고 테이블에 있는 도장의 정보를 가져와야 합니다. 서버는 데이터베이스와 연결하여 현재 위치에서 일정한 범위에 안에 있는 도장 정보들을 가져 옵니다.  서버에서 데이터베이스로 가져올 때 MyBatis 프레임 워크를 쓰기 때문에 ArrayList형태로 오게 되는데, Mixare 는 증강현실을 표현하기 위한 정보를 JSON 으로 표현하기 때문에 ArrayList형태의 데이터들을 JSON 형태로 바꾸어 줍니다. 그리고 서버는 도장의 정보를 안드로이드로 보내기 위해 PrintWriter로 http응답을 합니다.  Mixare는 서버에서 보낸 응답을 DownloadMgrImpl 이라는 클래스에서 받습니다. JSON형태로 현재위치에서 찍을 수 있는 도장의 정보를 가져오면 Dataconvertor 클래스로 JSON 형태의 도장 정보를 파싱하여 도장 정보를 생성합니다. |

|  |
| --- |
| **ddd4** |
| **그림 4 – Software Architecture ( Show Stamp and Touch Event)** |
| DataConvertor클래스가 도장 정보를 load하는 함수를 쓰게 되면, Mixare의 MixView에 있는 AugmentedView의 onDraw 함수를 호출 합니다. onDraw 함수로 SurfaceView에 도장 이미지가 표시가 되면서 카메라 화면에 증강현실로 도장 이미지가 실시간으로 구현이 됩니다.  카메라 화면에 표시되고 있는 도장 이미지를 클릭 하였을 때에는 Touch 이벤트가 발생하여 Marker를 상속받은 LocalMarker 의 fClick 함수가 호출됩니다.  도장을 저장하기 위하여 fClick함수에서는 도장의 정보와 SharedPreference 로 저장된 이름과 전화번호의 정보를 서버에 보냅니다. HttpClient 로 도장 정보를 담아서 서버에 요청을 하게 되면 서버는 도장 정보를 가지고 알맞은 지역에 데이터를 저장합니다. |

|  |
| --- |
| **ddd5** |
| **그림 5 – Software Architecture ( MyStamp Activity )** |
| MainActivity 에서 MyStamp 버튼을 click하고 지역별로 도장판의 Activivty 를 click하게되면 SharedPreferene클래스로 초기 어플리케이션 실행때 저장했던 사용자의 이름과 전화번호를 가져옵니다. 도장판은 각 지역별로 나뉘어져 있기 때문에 지역별 Activity 마다 Place 이름을 가지고 있습니다.  도장판에 찍힌 도장을 보기 위해서 사용자 정보와 함께 도장판의 지역이름을 HttpClient 라이브러리로 서버에 전송을 합니다. 서버는 지역 이름에 맞게 그 지역의 place table에서 사용자의 이름과 전화번호로 사용자가 도장을 찍었던 장소가 있는지 검색합니다. 장소가 있다면 서버는 다시 안드로이드 기반 클라이언트로 갔었던 장소를 보내줍니다.  안드로이드 액티비티를 .xml 로 표현할 때 이미지뷰 속성중 tag를 달 수 있는 속성이 있는데, 지역별로 각 액티비티에는 이미지뷰로 표현되어 있는 도장들이 관광 명소의 이름을 tag로 달고 INVISIBLE 형태로 있습니다.  서버에서 받아온 장소와 액티비티에 있는 이미지뷰들의 tag 를 비교하여 장소와 tag가 같은 경우 이미지뷰를 VISIBLE 로 바꿉니다. 이렇게 함으로써 사용자가 도장을 찍었던 도장을 볼 수 있습니다. |

* **Function Explanation (Applicant)**

|  |  |
| --- | --- |
| 설치시작 화면 | 이 화면은 찍Go 어플리케이션을 처음 설치하여 나타나는 Dialog입니다. 여기에서 이름과 핸드폰 번호를 적게되면 SharedPreference 클래스에 저장이 되어 다음 실행할때부터는 이 화면이 생략되고 이름과 전화번호를 기억하면서 어플리케이션을 실행 할 수 있습니다. |
| **그림 6 – 실행 첫 화면** |

|  |  |
| --- | --- |
| 초기 화면 | 실행 첫 화면 이후의 메인 화면입니다. MainActivity에서는 나의 도장보기 버튼과 도장 찍기 버튼이 있습니다. 각 버튼을 클릭하면 알맞은 Activity 로 전환합니다. |
| **그림 7 – 메인 화면** |

|  |  |
| --- | --- |
| 진주성 | 이 화면은 서버로부터 받은 도장의 정보가 SurfaceView를 통하여 도장이미지를 보여줌으로써 증상현실을 구현하고 있습니다. |
| **그림 8 – 증강현실로 표현된 도장** |

|  |  |
| --- | --- |
| 도장 애니메이션  도장 찍힌거 | 도장 이미지를 클릭하게 되면 터치이벤트가 발생하여 Dialog가 생성됩니다. Dialog 위에서는 안드로이드의 Animation클래스가 프레임 형식으로 도장을 실제로 찍는 것처럼 표현 하여 보여줍니다. |
| **그림 9 – 도장이 찍히는 모습** |

|  |  |
| --- | --- |
| 지역별 화면 | 이 화면은 MainActivity 에서 나의 도장 보기 버튼을 클릭하였을 때 나오는 화면입니다. 지역별로 볼 수 있도록 버튼을 나누었습니다. |
| **그림 8 – 나의 도장보기 화면** |

|  |  |
| --- | --- |
| 도장이 찍힘 | 진주에서 증강현실로 표현된 도장을 클릭하고 난 뒤, 데이터베이스에서 도장이 저장이 됩니다. 저장이 된 후 나의 도장 보기에서 진주에서 사용자가 찍은 도장을 확인한 화면입니다. |
| **그림 7 – 진주 도장판** |