국민대학교
전자정보통신대학
컴퓨터공학부

캡스톤 디자인 2
종합설계 프로젝트

프로젝트 명	Bitpay
팀 명	Firstcoin
문서 제목	2차 중간보고서

Version	1.0
Date	2015-10-29

팀원	장 예진 (조장)
	이 서연
	신 지은
	정 다운
	김 학균
지도교수	윤 성혜 교수

CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 전자정보통신대학 컴퓨터공학부 및 컴퓨터공학부 개설 교과목 캡스톤 디자인2 수강 학생 중 프로젝트 "Bitpay"를 수행하는 팀 "Firstcoin"의 팀원들의 자산입니다. 국민대학교 컴퓨터공학부 및 팀 " Firstcoin"의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

문서 정보 / 수정 내역

Filename	2차중간보고서-Bitpay.doc
원안작성자	장예진, 이서연
수정작업자	장예진, 이서연, 신지은

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2015-10-25	장예진 이서연	0.5	최초 작성	
2015-10-27	장예진 이서연 신지은	0.9	내용 수정	오타 및 내용 수정
2015-10-29	장예진 이서연 신지은	1.0	내용 수정	오타 및 내용 수정

목 차

1 프로젝트 목표

현재 널리 사용되고 있는 결제수단은 크게 카드와 현금으로 나눌 수 있다. 이와 더불어 최근에 등장한 Mobile, NFC, 기타 간편결제 등에 의해 결제수단이 점점 늘어나고 있는 추세이다. 기존 프로젝트에서는 이 결제수단에 비트코인을 추가하고자 하였다.

기본적으로 비트코인의 거래는 인터넷 상에서 P2P 형식으로 이루어진다. 이러한 특징에 따라 현재 비트코인은 스마트폰으로 구동되는 전자지갑 application을 이용하여 결제가 이루어지고 있다. 전자지갑은 일종의 은행계좌와도 같은 것으로 각 전자지갑마다 고유의 비트코인 주소가 할당되어 이 주소가 은행 계좌번호와 같은 역할을 한다. 이 주소를 통해 비트코인 사용자들은 서로의 비트코인을 교환할 수 있다.

스마트폰을 갖고 있고, 인터넷이 되는 환경이라면 누구나 비트코인을 사용할 수 있다. 현재 국내에서 비트코인을 받는 오프라인 상점은 100여 곳에 불과하다. 이는 비트코인을 접할 수 있는 환경이 주어지더라도 사용처가 마땅치 않아 상점의 확산이 더딘 것이라 생각된다. 지난 프로젝트에서는 상인과 소비자 모두 비트코인을 편리하게 사용할 수 있는 'bitpay' 개발을 완료하였다. bitpay는 비트코인 지갑이 제공하는 비트코인 전송 기능을 이용하여 비트코인 특유의 간편 결제모듈을 통해 결제수단으로서의 촉진을 돕는다. 그 뿐만 아니라 통계 기능을 제공하여 상인의 재고 관리, 자산 관리, 서비스 관리 등을 도우며, 원거리 주문 등의 편리한 기능을 제공하여 비트코인 결제 플랫폼을 구축하였다.

그러나 현재까지의 application은 보안이 유지되지 않았고, 서비스가 불안정한 점 등 여러 문제점을 인식하였다. 따라서 본 프로젝트는 데이터, 통신 등의 암호화와 서비스 안정화를 위한 리팩토링, 기능추가를 추진하여 'bitpay'의 안정화를 최종 목표로 한다.

2 수행 내용 및 중간결과

2.1 계획서 상의 연구내용

2.1.1 SSL

웰드 와이드 웹 브라우저와 웹 서버 간에 데이터를 안전하게 주고받기 위한 업계 표준 프로토콜이다. SSL 은 웹 제품뿐만 아니라 파일 전송 규약(FTP) 등 다른 TCP/IP 애플리케이션 에 적용할 수 있으며, 인증 암호화 기능을 제공한다. SSL 통신의 구축 확인은 와이어샤크를 통한 패킷 분석으로 가능할 것이다.

구축 전 패킷은 다음과 같이 보여진다.

```
0230  2c 20 64 65 66 6c 61 74  65 0d 0a 41 63 63 65 70   , deflat e..Accep
0240  74 2d 4c 61 6e 67 75 61  67 65 3a 20 6b 6f 2d 4b   t-Langua ge: ko-K
0250  52 2c 6b 6f 3b 71 3d 30  2e 38 2c 65 6e 2d 55 53   R,ko;q=0 .8,en-US
0260  3b 71 3d 30 2e 36 2c 65  6e 3b 71 3d 30 2e 34 0d   ;q=0.6,e n;q=0.4.
0270  0a 43 6f 6f 6b 69 65 3a  20 4a 53 45 53 53 49 4f   .Cookie:  JSESSIO
0280  4e 49 44 3d 44 46 32 38  44 32 38 41 32 46 42 41   NID=DF28 D28A2FBA
0290  41 46 31 36 33 36 33 45  34 30 41 34 46 38 46 37   AF16363E 40A4F8F7
02a0  30 38 37 33 0d 0a 0d 0a  49 6e 70 75 74 49 64 3d   0873.... InputId=
02b0  74 65 73 74 31 26 49 6e  70 75 74 50 61 73 73 77   test1&In putPassw
02c0  6f 72 64 3d 31 32 33 34  35                       ord=1234 5
```

다음을 통해 패스워드가 그대로 노출되고 있음을 알 수 있다.

SSL 통신을 적용 한 이후 패킷의 모습은 다음과 같다.

```
0000  00 08 9f b9 0c 3a d0 50  99 55 60 25 08 00 45 00   .....:.P .U`%..E.
0010  00 8a 58 bb 40 00 80 06  00 00 c0 a8 00 07 cb f6   ..X.@... ........
0020  70 83 ce bb 0c ea 81 66  f4 41 0e 4b e7 28 50 18   p......f .A.K.(P.
0030  3f 09 fd a5 00 00 5e 00  00 00 03 73 65 6c 65 63   ?.....^. ...selec
0040  74 20 2a 20 66 72 6f 6d  20 6d 65 72 63 68 61 6e   t * from  merchan
0050  74 20 77 68 65 72 65 20  6c 6f 67 69 6e 5f 69 64   t where  login_id
0060  3d 27 74 65 73 74 31 27  20 61 6e 64 20 70 61 73   ='test1'  and pas
0070  73 77 6f 72 64 3d 27 38  32 37 63 63 62 30 65 65   sword='8 27ccb0ee
0080  61 38 61 37 30 36 63 34  63 33 34 61 31 36 38 39   a8a706c4 c34a1689
0090  31 66 38 34 65 37 62 27                           1f84e7b'
```

다음과 같이 패스워드가 암호화되어 보여지는 것을 알 수 있으며, 이를 통해 안전한 통신이 가능하게 되었다.

2.1.2 SHA256

SHA(Secure Hash Algorithm, 안전한 해시 알고리즘) 함수들은 서로 관련된 암호학적 해시 함수들의 모음이다. 미국 국가 안보국(NSA)이 1993 년에 처음으로 설계했으며 미국 국가 표준으로 지정되었다.

SHA 함수들 중 가장 많이 쓰이는 SHA-1 은 TLS, SSL, PGP, SSH, IPSec 등 많은 보안 프로토콜과 프로그램에서 사용되고 있다. SHA-1 은 이전에 널리 사용되던 MD5 를 대신해서 쓰이기 도 하지만, 최근 MD5 의 보안취약점이 발견되어 SHA-256 이나 그 이상의 알고리즘을 사용 하는 것을 권장하고 있다.

2.2 수행내용

2.2.1 데이터암호화

비밀번호 등의 중요 데이터는 암호화하여 저장해야 한다. 이를 위해서 해시함수를 사용하여, 기존 데이터의 해시값을 데이터베이스에 저장한다. 또한 데이터의 일치 여부를 확인할 때에도 데이터를 해시함수를 적용한 값으로 비교해야 한다.

MD5.java

```
package kr.ac.kookmin.cs.firstcoin.posdata;
import java.security.MessageDigest;

public class MD5 {
    public static String encrypt(String plainText) {
        try {
            MessageDigest md = MessageDigest.getInstance("MD5");
            md.update(plainText.getBytes());
            byte byteData[] = md.digest();

            StringBuffer sb = new StringBuffer();
            for(int i=0; i<byteData.length; i++) {
                sb.append(Integer.toString((byteData[i] & 0xff) + 0x100, 16).substring(1));
            }

            StringBuffer hexString = new StringBuffer();
            for(int i=0; i<byteData.length; i++) {
                String hex = Integer.toHexString(0xff & byteData[i]);
                if(hex.length() == 1) {
                    hexString.append('0');
                }
```

```
                hexString.append(hex);
            }
                return hexString.toString();
            } catch(Exception e) {
                e.printStackTrace();
                throw new RuntimeException();
            }
        }
}
```

MD5 적용

```
final String userId = idEdit.getText().toString();
String userPassword = passEdit.getText().toString();
final String securePassword = MD5.encrypt(userPassword);

nameList = new ArrayList<NameValuePair>(2);

nameList.add(new BasicNameValuePair("id", userId));
nameList.add(new BasicNameValuePair("password", securePassword));
httpPost.setEntity(new UrlEncodedFormEntity(nameList));
httpResponse = httpClient.execute(httpPost);
ResponseHandler<String> responseHandler = new BasicResponseHandler();
```

2.2.2 SSL통신

MD5의 결함을 보완하고, 데이터 보안을 강화하기 위해 통신 프로토콜로 SSL통신을 선택하였다. 데이터베이스 또한 SSL 통신으로 설정을 변경하였고, SSL 이외의 접근은 허용이 불가하도록 설정하였다.

2.2.3 WebView

웹페이지에서만 가능하던 회원가입을 애플리케이션에서도 가능하도록 기능을 추가하였다. 웹 페이지와 애플리케이션의 통일감을 주기 위해 WebView기능을 활용하였고, 이는 유지

보수를 좀 더 용이하게 할 수 있을 것이라고 생각한다.

웹페이지를 좀 더 적극 활용하기 위해 통계 탭에 WebView기능을 추가하였고, 텍스트 외에 그래프로 통계내역을 확인할 수 있게 되었다.

SignupActivity.java

```
public class SignupAcivity extends ActionBarActivity {
    private WebView mWebView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_signup);
        mWebView = (WebView)findViewById(R.id.webView1);
        //웹뷰에서 자바스크립스 사용
        mWebView.getSettings().setJavaScriptEnabled(true);
        mWebView.loadUrl("http://203.246.112.137/bitpay/signup.jsp");
        mWebView.setWebViewClient(new SignupWebViewClient());

    }
```

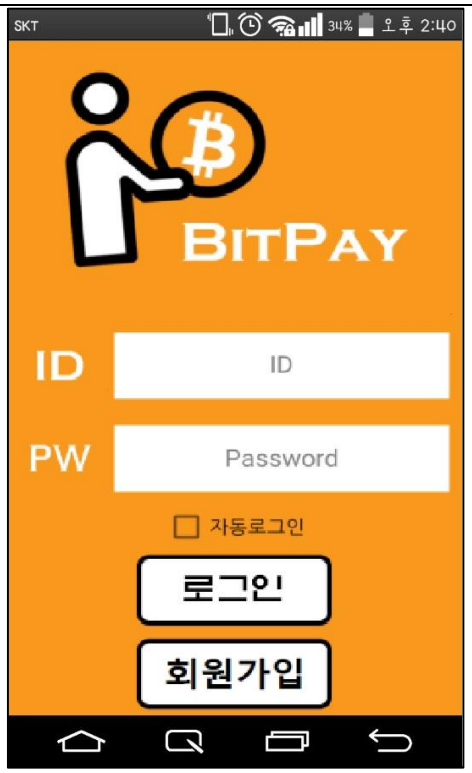
activity_signup.xml

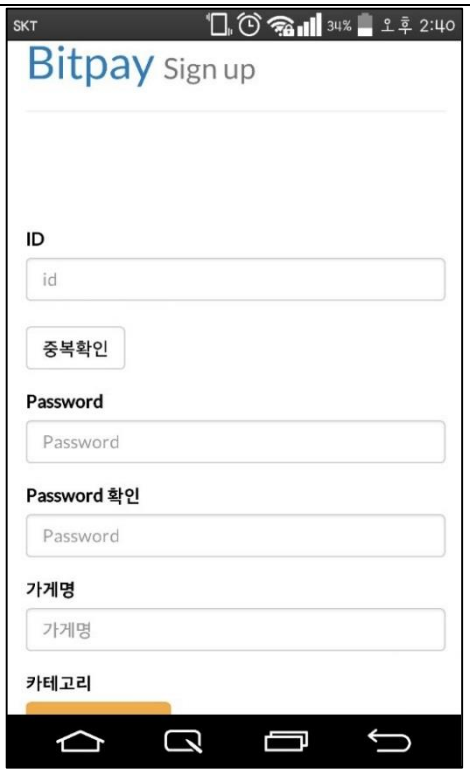
```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/login_layer"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/login_orange_bg"
    android:gravity="center"
    android:orientation="vertical" >

    <WebView
        android:id="@+id/webView1"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</LinearLayout>
```

WebView 실행화면





3 수정된 연구내용 및 추진 방향

3.1 MD5

SHA256 해시함수를 이용하여 비밀번호를 암호화하였으나 데이터베이스에서 저장하지 못하는 문제를 발견하였다. 또한, 웹에서 SHA256 해시함수를 실행한 결과와 애플리케이션에서 SHA256 해시함수를 실행한 결과값이 서로 달라, 로그인에 실패하는 에러가 발생하였다. 따라서 SHA256을 비밀번호 암호화 해시함수로 사용하는 것이 부적절하다는 판단을 하게 되었고, MD5로 해시함수를 변경하게 되었다.

MD5는 해시함수의 결함이 발견되어 데이터암호화를 위한 용도로 사용을 권장하지 않는 해시 함수이다. 하지만 통신 프로토콜로 SSL을 사용하여 데이터보안을 강화시켰다.

이는 2.2 수행내용과 같이 수정을 완료하였다.

4 향후 추진계획

4.1 향후 계획의 세부 내용

4.1.1 기능추가

1학기에 진행하는 동안 시간이 부족하여 구현하지 못했던 기능이 있다. 또한 지난 최종 발표 이후 받은 여러 가지 피드백을 토대로 앞으로 서비스를 제공하는 데에 필요하다고 생각하는 기능을 정리해보았다. 이들을 토대로 새로운 기능들을 추가하기로 하였다.

첫째, 주문 수정 기능이다. 현재는 주문이 완료된 후에는 주문 취소 또는 주문 수정이 불가능하다. 때문에 주문내역과 주문자를 수정하는 기능을 추가할 예정이다.

두 번째는 주문 취소 기능이다. 결제가 되기 전이라면 주문 취소가 가능하도록 하여 사용자에게 더욱 편리한 서비스를 제공할 수 있을 것이다.

5 고충 및 건의사항

5.1 데이터암호화 및 SSL

데이터베이스 SSL 설정 과정에서 어려움을 겪었다. 인증서를 발급받고, 설정하는 과정에서 SSL이 적용되지 않아 한 동안 애플리케이션의 개발이 중단되기도 하였다. 다행히 원인을 찾아 해결할 수 있었고, SSL 통신을 성공적으로 완료할 수 있었다.