USAGE OF WEB SERVICE IN MOBILE APPLICATION FOR PARENTS AND STUDENTS IN BINUS SCHOOL SERPONG

Karto Iskandar[1]; Andrew Thejo Putranto[2]

[1,2]Computer Science Department, School of Computer Science, Bina Nusantara University
Jln. K.H. Syahdan No. 9 Palmerah, Jakarta Barat, 11480
karto_i@binus.edu[1]; archgod94@gmail.com[2]

ABSTRACT

A web service is a service offered by a device electronically to communicate with other electronic device using the World wide web. Smartphone is an electronic device that almost everyone has, especially student and parent for getting information about the school. In BINUS School Serpong mobile application, web services used for getting data from web server like student and menu data. Problem faced by BINUS School Serpong today is the time-consuming application update when using the native application while the application updates are very frequent. To resolve this problem, BINUS School Serpong mobile application will use the web service. This article showed the usage of web services with XML for retrieving data of student. The result from this study is that by using web service, smartphone can retrieve data consistently between multiple platforms.

Keywords: *web service, mobile application, BINUS School Serpong*

INTRODUCTION

The mobile development using operating system-based applications have become popular. Program applications running on smartphones and tablets called mobile applications. Mobile applications are typically developed and operated in accordance with their operating system and is usually provided on the app distribution platforms, such as Google Play and App Store.

Bina Nusantara (BINUS) University is a private organization in Indonesia that stir in the field of education that realizing the opportunity to implement their system in smartphones. One of the BINUS Group main strengths in creating and operating activities and academic services to BINA NUSANTARA Group stakeholders is the implementation of IT (one of them is a web application) in every aspect of the process of academic and academic support. IT Directorate was established in 1981 under the name EDP. In August 2005, the BINA NUSANTARA Foundation decided to change the status of the IT department of the Directorate. IT Directorate today is responsible for all Business Units in BINUS Group, not only supporting BINUS UNIVERSITY. IT Directorate is now led by the Director of IT. Currently, IT Directorate is located on the campus Syahdan. The office is located on the 2nd floor of Building H.

BINUS School Serpong is part of BINUS Group, which accept students from kindergarten to high school. BINUS School Serpong has mobile applications for students, parents, and staff. The application is called as MSIS Student-Parent and MSIS Staff. They are available on Android, iOS and Blackberry, but the future development will support Android and iOS only. MSIS Student-Parent usually used by students or parents in BINUS School Serpong to see class schedules, announcements, and score. Because of multiple platform development, data integrity is a serious problem that must be maintained.

Problems that are faced by BINUS School Serpong today is the time-consuming application update when using the native application while the application updates are very frequent. To resolve this problem, the mobile application of BINUS School Serpong will use the web service. In this article, the implementation of web service in the BINUS School Serpong mobile applications as a method that can maintain data integrity between multiple platforms will be discussed.

METHODS

This research uses two phases of methodology. The first methodology is literature study and the second is development using Objective-C, Java, and ASP.NET C#. The methodology of application development is incremental model.Some literature study of terms and technologies used in this study are XCode, Eclipse, and Microsoft Visual Studio. As for some literature used as references can be found in the Bibliography.

According to Pressman (2009), there are many circumstances where the business requirements of the initial software defined well enough, but the major scope of the development effort deters purely linear process. Additionally, there may be an urgent need to provide a limited set of software functions to users quickly and then improve and expand that function in later software releases. In such cases, you can choose a process model that is designed to produce the software in incrementally.

Referring to Figure 1, incremental model uses linear sequence in turns as working time. Each linear sequence produces an output of the software in a manner similar to the increase generated by the flow of the evolutionary process.

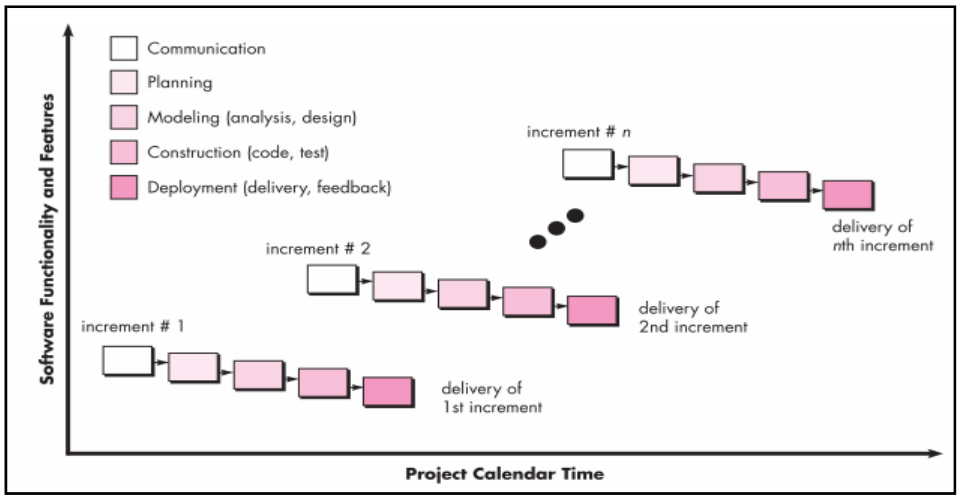


Figure 1 Incremental Process Model

When using a model of incremental, first iteration is often a repetition of the core. That is, the basic requirements are handled, but many additional features (some known, some unknown) remain undelivered. The main output has been used by the customers (or undergoes a detailed evaluation). As a result of the use and/or evaluation, a plan is developed for the next improvement. The plan discusses the modification of the core products better to meet customer needs and delivery of additional features and functionality. This process is repeated after increasing the previous software is complete, until the complete product produced.

The incremental process model focuses on delivering products with improved operations at each iteration. At the beginning of the hike is the version taken from the final stage, but the product can already be used by the user. The software has also been able to provide a platform for evaluation by the user.

The incremental models are very useful when staff is not available to complete the implementation deadline has been set for the project. In the early stages of implementation can be done with fewer people. If the main product has been well received, then the additional staff may be added to implement further incremental iteration. Additionally, incremental can be planned to reduce technical risk. For example, the primary system certainly requires new and expensive hardware while hardware delivery date is uncertain. It is possible to plan the initial incremental manner that avoids the use of this hardware device, thus allowing partial functions to be delivered to the end user without excessive delay.

According to Wagh and Thool (2014), mobile devices such as smart phones, tablets, tabs, with mobile web services and wireless communications, near 2020 would be hope to play a crucial role in all aspects of life. Mobile web services extend the concept of 'anywhere, anytime and on any device' to the new paradigm of mobile computing ubiquitous. It is used to improve access to information that is meaningful, rapid and necessary services and content through the mobile web. One of the most promising ways to make a decent web service for mobile devices is to add extra intelligence for web services, either from a web service provider and consumer web services.

According to Jackson (2011), at first, Android was created by Andy Rubin as an operating system for smartphone. In 2005, Android Inc acquired by Google, and made Andy Rubin as Director of Mobile Platforms for Google. The acquisition was largely in many think responses to the emergence of the Apple iPhone around thattime, Nokia Symbian, and Microsoft Windows Mobile. Then improvement for this things become popular with Internet 2.0 terminology. Internet 2.0 allows users to access everything content via networks and portable consumer electronic devices, such assmartphones and tablets.

According to Holla and Katti (2012), Android is a set of operating system, middleware and utiliy applications for mobile devices. The Android provides an SDK as tools and APIs to developing applications to run on Android platform using Java programming language. Android is based on Linux version 2.6. The system services such as security, memory management, process management are controlled by Linux operating system. Figure 2 shows the Android architecture.

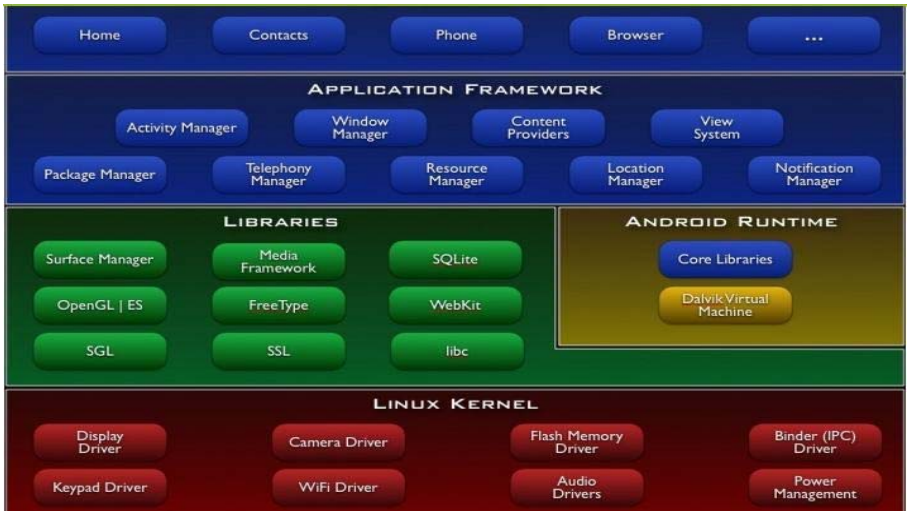


Figure 2 Android Architecture

According to Epifani (2015), all the devices described in this chapter have in common the use of the iOS operating system. Originally known as iPhone OS up to Version 3, it was developed by Apple specifically for iPhone, iPad, and iPod touch. It was unveiled for the first time in January 2007 and was introduced with the first model of iPhone in June of the same year. iOS is an operating system, based on the older forefather Mac OS X, it uses four levels of abstraction: (1) core OS, this level consists of file system, security, memory management, power management, network management, programming sockets, and any encryption purpose. (2) Core services: this level consists of networking, SQLite, geolocation, and threads. (3) Media: this level consists of OpenAL, audio, image, video, and OpenGL. (4) Cocoa touch: this level consists of core animation, multitasking, and gesture recognizer.

According to Sheikh, Ganai, Malik, and Dar (2013), Apple's mobile operating system or iOS, have a lot of different features developed compare to first versions of iOS since 2007. This improvement is introduced by Apple as technology innovative, even groundbreaking advances for mobile and smartphone device.

According to Lee (2014), web services provide a standard way to inter-operate between different software applications, which operated on a variety of platforms and/or frameworks. The WSA (Web service architecture) model provides a conceptual and contextual understanding of web services and the relationship between the components of this model. Figure 2 presents web service architecture. It has three components; namely XML web service broker (UDDI), XML web service consumer, and XML web service producer. A service provider provides a specific service, and it publishes the service specification to service registry. The XML web service provider receives web service call message (bind) from XML web service consumer. A XML Web service consumer finds a service specification published in the service registry. This service consumer calls or binds the corresponding service after it searches a service specification provided by XML Web service provider. A XML Web Service Broker manages such as directory to use and search XML Web Service consumer the provided service. The WSA describes both the minimal characteristics, and a number of characteristics that are needed by many, but not all web services. The WSA configuration is seen in Figure 3.

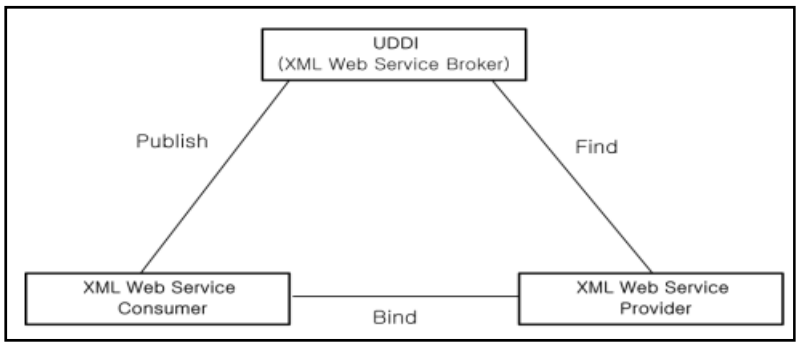


Figure 3 Web Service Architecture

In developing solution for mobile, three elements of technology namely, XML Web Service Broker (UDDI), WSDL, and SOAP are needed to implement the above described web service architecture. Figure 4 shows the technology elements.

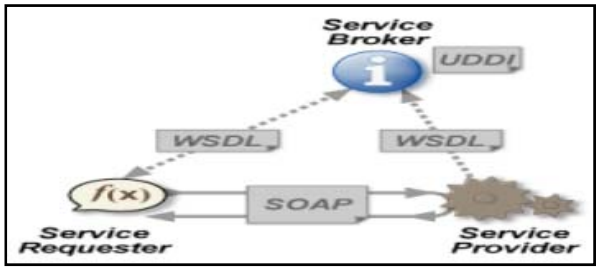


Figure 4 Web Service Standard Technology Elements

SOAP Model or Simple Object Access Protocol is a protocol used to deliver messages logically to exchange structured information between described services by WSDL interface. Figure 5 shows that there is a simple architecture concept to implement SOAP Model messages through transport protocol that is set between several web services.

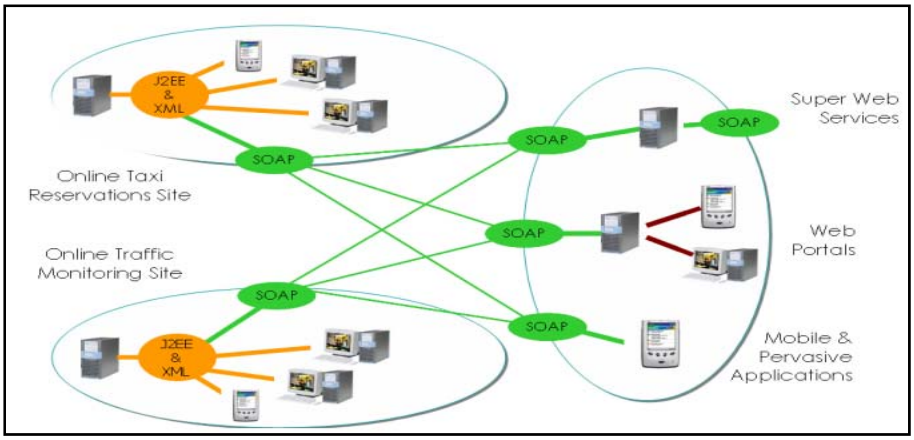


Figure 5 SOAP Transfer Model

SOAP is the most important protocol in supporting to web services. SOAP messages are consisted of XML documents with route element. There have two knowledge elements on basic SOAP based structure; header and body. These elements envelope in which there is an application. Figure 6 shows SOAP message structure.

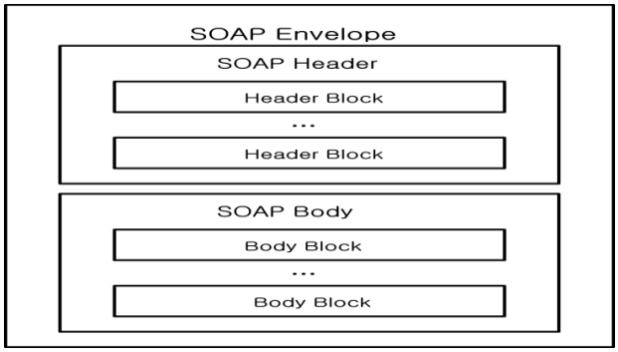


Figure 6 SOAP Message Structure

SOAP is a protocol standard that defines a particular method to provide the data encoded in XML form as a binding protocol. SOAP function to call a web service method using an XML format that has been provided. SOAP uses XML as a text-based protocol that uses a mild exchange of information within the distribution environment to have an advantage that is not dependent on the hardware platform, OS, programming languages, and hardware platforms network.

A SOAP messages provide conveying the message function in one direction (one-way messages). All SOAP basic coding scheme consists of an XML specification and provides two grammar to express the type of data.

RESULTS AND DISCUSSIONS

MSIS Serpong is created with the Java language and indeed, uploading your application to Google Play Store is quite time-consuming because the application cannot be directly accessed, but will be in the first review advance by the Google to filter the applications, not in accordance with the terms and conditions of the Google Play Store. Other than that, Android will also have its API that allows developers use it to access the functions of Android and also mobile phone hardware. One example is when a login with BINUSIAN id and password are correct; then the view will switch to the section after login. To move to the part, the API calling is required.

MSIS on IOS was made using Objective-C programming language, and to make this program certainly should have Mac and devices for testing, due to the manufacture and development application using XCode which can only be downloaded from the AppStore on the Mac. To be able to download or upload XCode downloads the necessary application to the Apple developer accountobtained by making on Apple's website. On Android, to be able to use the web service, it is necessary to use the plugin that called ksoap2. As for iOS, no need to use the plugin because there are libraries that support to use the web service. Example method for calling web service can be seen in Figure 7 and Figure 8.

```
String SOAP_ACTION = "http://tempuri.org/LoginLogoutStudentParent";
String METHOD_NAME = "LoginLogoutStudentParent";
String NAMESPACE = "http://tempuri.org/";
String URL = AppConfig.URLWebService+"loginservice.asmx";


String BinusianID,Password;
BinusianID = getFromPrefs(MainActivity.this, AppConfig.PREFS_LOGIN_USERNAME_KEY,"");
Password = getFromPrefs(MainActivity.this, AppConfig.PREFS_LOGIN_PASSWORD_KEY,"");
PackageManager manager = this.getPackageManager();
PackageInfo info = manager.getPackageInfo(
        this.getPackageName(), 0);
String version = info.versionName;

TelephonyManager mngr = (TelephonyManager) this.getSystemService(this.TELEPHONY_SERVICE);
String imei = mngr.getDeviceId();

String ApplicationID =  getFromPrefs(MainActivity.this, AppConfig.PREFS_LOGIN_APPID_KEY,"");
SoapObject Request = new SoapObject(NAMESPACE, METHOD_NAME);

Request.addProperty("LoginTypeID", "2");
Request.addProperty("BinusianID", BinusianID);
Request.addProperty("Password", Password);
Request.addProperty("ApplicationID", ApplicationID);
Request.addProperty("DeviceID", imei);
Request.addProperty("OSVersion", "Android "+ Build.VERSION.RELEASE);
Request.addProperty("Appver", version);



SoapSerializationEnvelope soapEnvelope = new SoapSerializationEnvelope(SoapEnvelope.VER11);
soapEnvelope.dotNet = true;
soapEnvelope.setOutputSoapObject(Request);

HttpTransportSE transport = new HttpTransportSE(URL);

transport.call(SOAP_ACTION, soapEnvelope);
```

Figure 7 Web Service Calling with ksoap2 in Java Android

```
questionnaireURL = @"";
matchingElement = @"GetQuestionnairePeriodResult";
NSString *soapMsg = [NSString stringWithFormat:
                     @"<?xml version=\"1.0\" encoding=\"utf-8\"?>"
                     "<soap12:Envelope xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"
                     xmlns:xsd=\"http://www.w3.org/2001/XMLSchema\" xmlns:soap12=\"http://www.w3.org/2003/05/soap-envelope\">"
                     "<soap12:Body>"
                     "<GetQuestionnairePeriod xmlns=\"http://tempuri.org/\">"
                     "<studentid>%@</studentid>"
                     "<applicationid>%@</applicationid>"
                     "<APIKey>%@</APIKey>"
                     "</GetQuestionnairePeriod>"
                     "</soap12:Body>" "</soap12:Envelope>", studentId, [KeychainHelper objectForKey:UD_APPLICATION_ID],
                     [KeychainHelper objectForKey:UD_API_KEY]];

NSURL *url = [NSURL URLWithString: [NSString stringWithFormat:@"%@Questionnaire.asmx", WEBSERVICE_URL]];
NSMutableURLRequest *req = [NSMutableURLRequest requestWithURL:url];
NSString *msgLength = [NSString stringWithFormat:@"%d", [soapMsg length]];

[req addValue:@"text/xml; charset=utf-8" forHTTPHeaderField:@"Content-Type"];
[req addValue:msgLength forHTTPHeaderField:@"Content-Length"];
[req setHTTPMethod:@"POST"];
[req setHTTPBody: [soapMsg dataUsingEncoding:NSUTF8StringEncoding]];

NSLog(@"Calling %@", url);

NSError *error = nil;
NSURLResponse *response = nil;
NSData *webData = [NSURLConnection sendSynchronousRequest:req returningResponse:&response error:&error];
```

Figure 8 Web Service Calling in iOS Using Swift

When the address of the web service is opened, several functions will be appeared in the web just like in Figure 9. If one of these functions is selected, the parameters of what is needed if you want to use the function will appear. This can be seen in Figure 10. Also, there are parameters which in return/generated from the function; which can be seen in Figure 11.

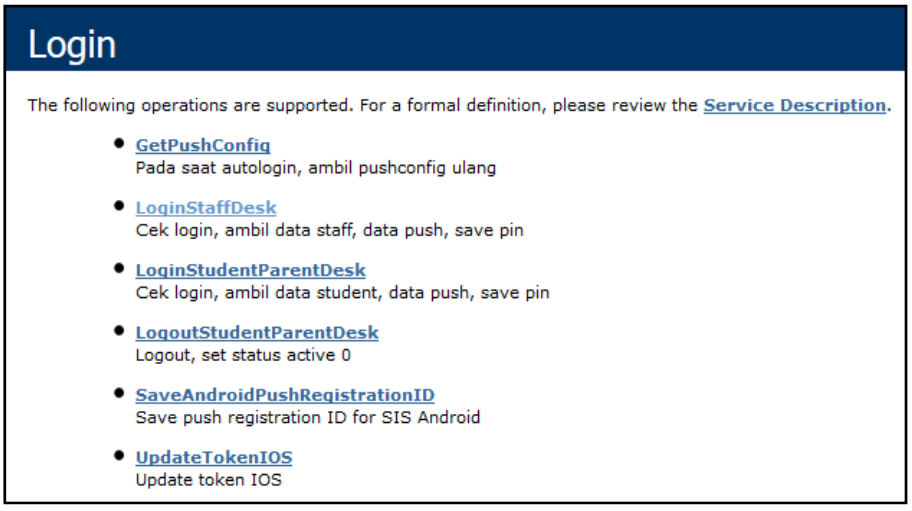
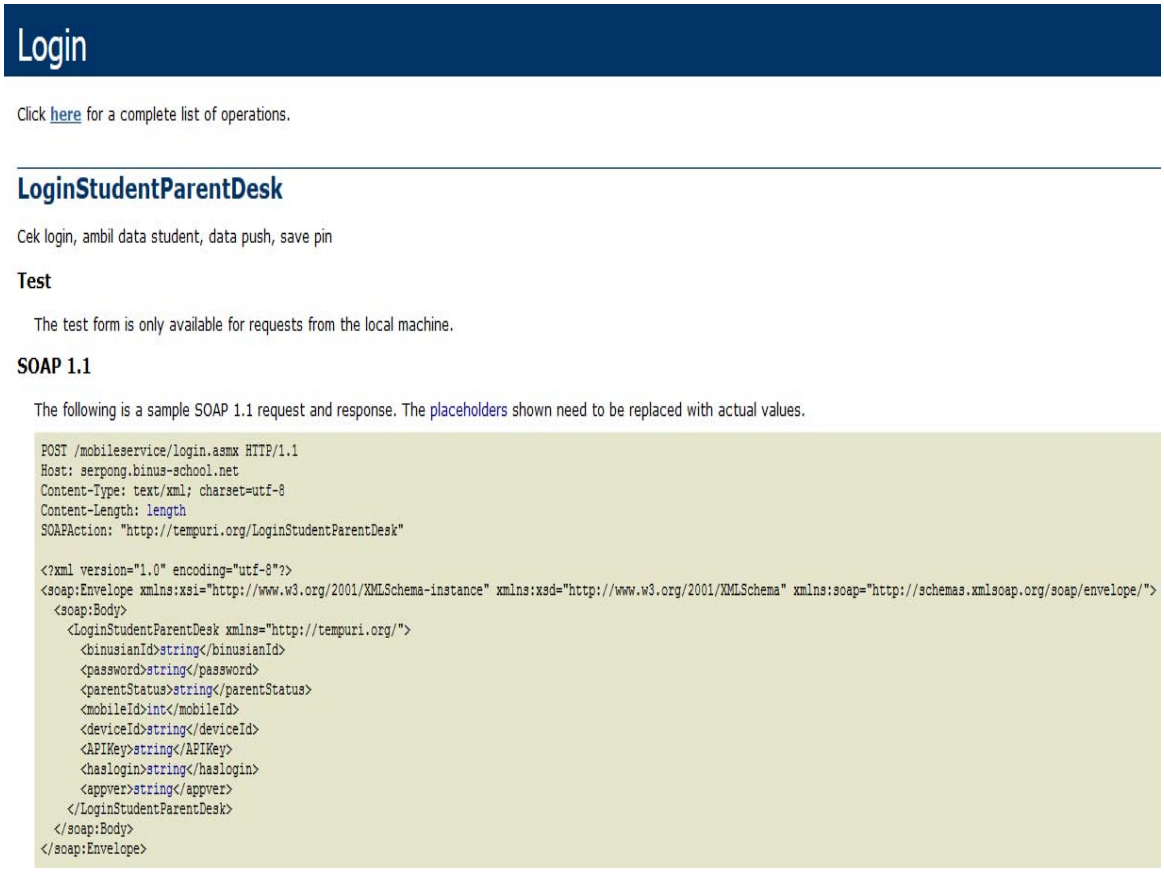


Figure 9 Web Service Login



Login

Click here for a complete list of operations.

LoginStudentParentDesk

Cek login, ambil data student, data push, save pin

Test

The test form is only available for requests from the local machine.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /mobileservice/login.asmx HTTP/1.1
Host: serpong.binus-school.net
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/LoginStudentParentDesk"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <LoginStudentParentDesk xmlns="http://tempuri.org/">
      <binusianId>string</binusianId>
      <password>string</password>
      <parentStatus>string</parentStatus>
      <mobileId>int</mobileId>
      <deviceId>string</deviceId>
      <APIKey>string</APIKey>
      <haslogin>string</haslogin>
      <appver>string</appver>
    </LoginStudentParentDesk>
  </soap:Body>
</soap:Envelope>
```

Figure 10 Web Service Login Function for Students and Parents

```
<LoginData>
  <BinusianId>string</BinusianId>
  <FirstName>string</FirstName>
  <StudentStatusId>int</StudentStatusId>
  <ClassPackageSetName>string</ClassPackageSetName>
  <ClassPackageSetId>int</ClassPackageSetId>
  <SchoolLevelId>string</SchoolLevelId>
  <YearLevelId>int</YearLevelId>
  <PathwayId>int</PathwayId>
  <SemesterId>string</SemesterId>
  <AcademicYear>string</AcademicYear>
  <ApplicationId>int</ApplicationId>
  <TermId>int</TermId>
  <PushAppId>string</PushAppId>
  <PushPort>string</PushPort>
  <PushBPSURL>string</PushBPSURL>
  <FullName>string</FullName>
  <YearLevelName>string</YearLevelName>
</LoginData>
```

Figure 11 Return Parameters By Login Student Parent Desk Function

CONCLUSIONS

BINUS SCHOOL SERPONG mobile application or often called MSIS SERPONG available on Android and iOS, which Android is made with the Java Programming Language and for IOS with Objective-C. The program used is Eclipse for Android and XCode for iOS.

MSIS SERPONG uses the web service to connect to SQL Server database so that the data obtained will be the same though different operating systems. So even though using Android or IOS, there will not be a problem because the same information obtained. Web Service is implemented only in some parts of the application, as most use the mobile web so that when there is an update then it is not long to raise the application. Use of this web service is on the module Login, About, and Feedback.

REFERENCES

Epifani, M. (2015). *Learning iOS Forensics*. Birmingham: Packt Publishing Ltd

Holla, S., & Katti, M. M. (2012). Android Based Mobile Application Development and Its Security. *International Journal of Computer Trends and Technology*, *3*(3), 486-490.

Jackson, W. (2011). *Android Apps For Absolute Beginners*. New York: Apress

Lee, S.H. (2014). A Study on Web Service Analysis and Bio-information based Web Service Security Mechanism. *International Journal of Security and Its Applications*, *8*(2), 77-86.

Pressman, R. S. (2009). *Software Engineering: A Practitioner's Approach*. New York City: McGraw-Hill

Sheikh, A. A., Ganai, P. T., Malik, N. A., & Dar, K. A. (2013). Smartphone: Android Vs IOS. *The SIJ Transactions on Computer Science Engineering & its Applications (CSEA)*, *1*(4), 141-148.

Wagh, K., & Thool, R. (2014). Mobile Web Service and Performance Evaluation of Mobile Host. *International Journal on Web Service Computing (IJWSC), 5(2), 1-10.*