

MOBILE ONLINE ACADEMIC SERVICE PROTOTYPE USING .NET ARCHITECTURE

Moh. Alwi Hasan, Tutun Juhana
Electrical Engineering Department
Institut Teknologi Bandung
Jl. Ganesha 10 Bandung - Indonesia
Email : ahsan@s.ee.itb.ac.id, tutun@telecom.ee.itb.ac.id

Abstract— Mobile internet application depend on its device capabilities, data transport technologies, and the services offered to users. In the past days, mobile application developer must develop their applications based on specific device. So for each of device type, they must develop different ways of application. Nowadays, Microsoft .NET technology offered fast and easy mobile applications development, regardless of client device capability. This paper describes prototype of online academic service application development for mobile internet service using ASP.NET Mobile Control over Microsoft .NET architecture. The application includes core application based on WAP standard, for web page and phone call, and complementary application such as database and email.

Keywords -- Mobile Internet Application, Microsoft .NET , academic services, multiple devices

1. Introduction

Fast, easy, anytime and anywhere are advantages of applications using wireless network. However, the design of wireless internet service is not a simple task. The devices, with different capabilities, and the network reliability must be accounted in building mobile internet application. For example, the service designed for PDA or iMode device with CHTML support can be accessed by WAP-supported hand phone only.

Microsoft with their MMIT product, as the first version, and ASP NET Mobile Controls, as their new version, is a “Multi-Client Mobile Based Application” software. It promise easy mobile internet programming. In this research we build prototype of online academic services in ITB on mobile device, based on microsoft .NET architecture.

2. Overview of Microsoft .NET

2.1 Microsoft .NET^[1]

Microsoft® .NET is a set of software technology to connect information, people, system and device using web service over internet. .NET technology enables the development of applications, processes, and web sites, based on XML, as services that share their information and functionalities to build an enterprise service.

As a “*software platform*” which is not depend on any language programming, .NET offers easier process in building application and it promise service interoperability among all language programming that already supported by .NET. .NET uses XML and HTTP to support distributed computation application over internet.

2.1.1 Microsoft .NET Framework^[1]

DOT NET Framework is computation platform that facilitate application development in distributed environment over internet. There are three main components in .NET Framework Architecture, those are :

- ❖ CLR (Common Language Runtime)
- ❖ Framework classes
- ❖ ASP .NET

Figure 1 below shows the diagram of .NET Framework

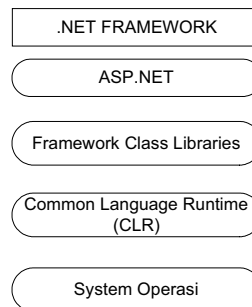


Figure 1. Microsoft Framework Architecture^[1]

2.1.2 ASP.NET

ASP.NET is the main component to build web based application on Microsoft .NET platform. ASP .NET is the new version of ASP 3.0 and has been developed based on OOP (*Object Oriented Programming*). It has server side characteristic, which is dynamically build the web application on server and presented at client as static page.

2.1.3 Microsoft ASP.NET Mobile Controls^[1]

ASP.NET Mobile Controls is an extension of Microsoft ASP.NET and Microsoft .NET Framework. It contains a set of server-side Web Form Control to build application for mobile devices. This component enables the output of the application result to be match to the mobile client device

capabilities. It is also able to produce output in WML1.1, HTML3.2, and compact HTML markup language.

3. Simulation of Online Academic Service Application

3.1 The Purpose of Simulation

The purpose of simulation is to evaluate the benefits of using Microsoft .NET to build online academic service application for mobile device. The application consists of web page application, phone call application, which is the standard WAP application, and also additional application such as email and database access.

Application software used in the design is as follows:

1. IDE, Microsoft Visual Studio as a design tools.
2. Etheral version 0.10 for data and client server communication analysis
3. EditPlus 2.10 for data byte counting and text editor

3.2 Online Academic Service Application Design

The design process is made as close as possible to data process in manual academic systems (including the adoption of the real database on manual service in simpler version) as follow:

- Online FRS (*Formulir Rencana Studi (Plan of Study Formi)*)-Form. Students have to fill this form prior the commencement of each semester. This online form is connected to database of latest students status, database of student advisors (counselor), database of tuition fee administration, database of courses and database of FRS itself. The flow is shown in *Figure 2*.
- Online PRS (*Perubahan Rencana Studi- Plan of Study Modification*)-Form. It connected to FRS database, database of student advisors (counselor), database of tuition fee administration, database of courses and database of FRS itself. The flow diagram is shown in *Figure 2*.
- Academic service information systems. It consist of academic calendar, information searching, academic record, and examination schedule (test schedule). See *Figure 3* for detail.
- Help system as a guidance to use the application
- Login application is equipped with authentication system to check whether the user has already made an agreement with Students Academic Administration Bureau (BAAM) or not. This is to make sure that only authorized students who can use the applications.

4. Simulation Results and Analysis

To analyze the system, we were performing following test and measurement:

- Measurement of information size transferred from client to server and vice versa. The result of this measurement is shown in *Table 1* and *Table 2*.
- Multi devices test capability. The result of this testing is shown in *Figure 5*.

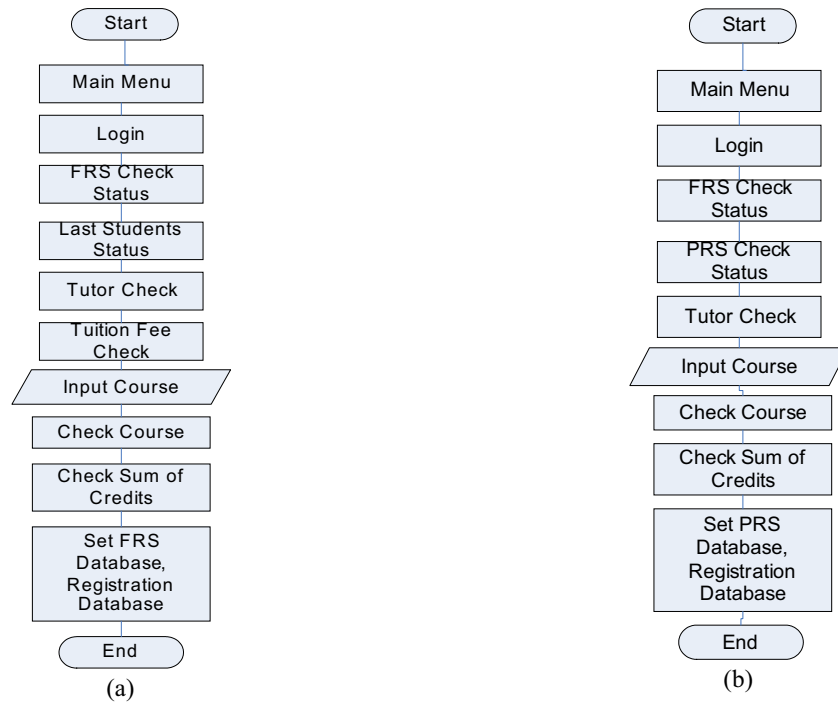


Figure 2 (a) FRS flow diagram (b) PRS flow diagram

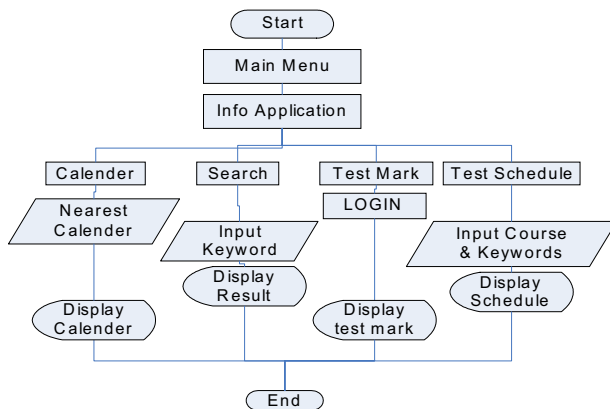


Figure 3 Academic Information Service

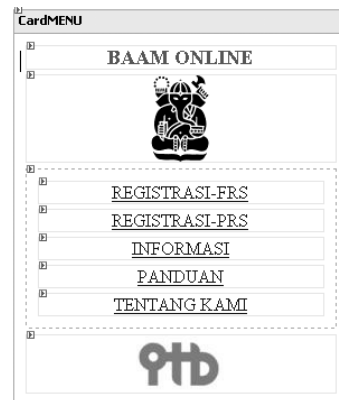


Figure 4. Welcome page

4.1 Analysis

From the result above and data log at IIS Server, it can be concluded that we only have to create one design application for diverse devices. This is the advantage of using “*multi client mobile based application*”. For client that is unable to accept the image, the image file will never be sent by the server.

.NET is able to determine device capability during GET request session. *Figure 7* show application result received by client that has no WTAI support (in our case, user will be notified with phone number that can be called).

Table 1 Data size send to client from log file Server Web IIS 5.1

Client Browser Type	Data Format	Data Size
Internet Explorer 6.0	text/html	1273 bytes
Up.3.2 HDML	text/vnd.wap.wml	1382 bytes
Up 4.1 WML	text/vnd.wap.wml	1406 bytes
UP 6.2 XHTML		1401 bytes
Handphone M50	text/vnd.wap.wml	1382 bytes

Table 2. Data size receive by client device

Client Browser Type	Data Format	Data Size
Internet Explorer 6.0	text/html	1011 Bytes
Up.3.2 HDML	text/vnd.wap.wmlc	576 Bytes
Up 4.1 WML	text/vnd.wap.wmlc	661 Bytes
UP 6.2 XHTML	text/vnd.wap.wbxml	1233 Bytes
NOKIA WAP	text/vnd.wap.wmlc	390 bytes

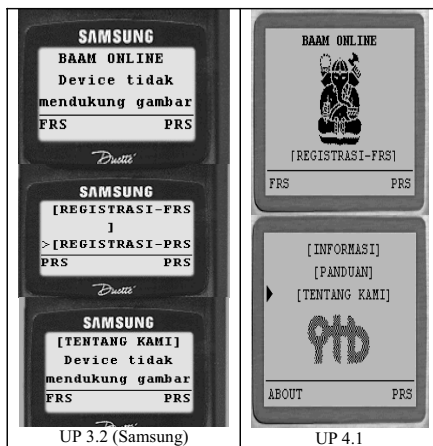


Figure 5. Application appearance at different device

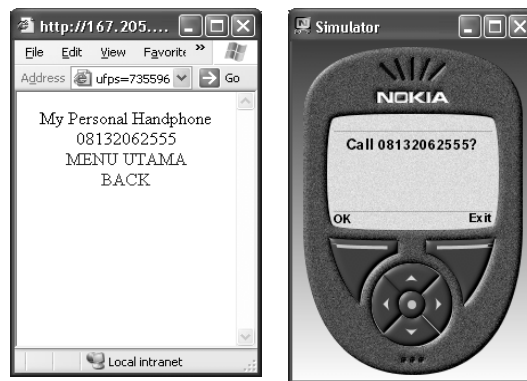


Figure 6. Result for no WTAI support client

5. Conclusion

1. Online academic service prototype has successfully built by .NET technology.
2. The prototype can be accessed by various different client device capabilities.
3. The design for online academic service application can be built faster and easier using IDE Visual Studio and ASP.NET WYSIWYG.

6. References

- [1] MSDN April 2003
- [2] *GPRS White Paper*, http://www.cisco.com/warp/public/cc/so/neso/gprs/gprs_wp.htm , 2000
- [3] Vineet Rajosi Sharma , *Wireless Application Protocol*, <http://www.iitd.ac.in/~rajosi/wap>, June 2000.
- [4] *ASP.NET Quickstart* , <http://docs.aspnet.com/quickstart/aspplus/>