

SoA ★ Assignment 4

Ranjit WAVHAL, Pierre QUEMARD

16 novembre 2007



1 Executive Summary

We realized that when we use our myub site we cannot access all the informations we needed as easily as we expected. For example it would be great to indicate when is the next buses for your favorite bus stop! We'd rather have a box that show mail informations, UB Learns assignment. It would also be great to have an agenda with the courses and the due tasks, informations regarding academics courses. And why cant we also shows external informations like gmail, facebook etc ... We are undertaking this project because we think this could really help UB students by simplifying the way they access to the critical informations like exams dates and homework submissions.

Our project will be a web browser based service that will gather informations for the user. Those informations will come from UB services available on myub and maybe from som other services like gmail, facebook, gcalendar and so on. We will first focus on getting informations from UB services and trying to put everything on one browser frame.

The web page may be used on any web browser so we may use html and php for the frontend. A few coding in Ajax and JavaScript maybe usefull too. We may gather informations from excel files for the buses schedule and xml from the UB services.

This web service is mainly for students but it also can be used by faculty staffs or teachers, on top of that it can be modified to fit with others university intranet.

2 Project objectives

PROJECT → YOUUB



- ▶ UB Learns Assignement and Tasks
- ▶ UB Webmail (Incomming mails should be noticed with infos about them)
- ▶ Bus Schedule (with preferred bus station)
- ▶ UB infos like event athletics news announcements
- ▶ Finances informations
- ▶ An alert if you need to return a book!
- ▷ Facebook/Other Social networking support
- ▷ Gmail support
- ▷ Gcalendar or Icalendar
- ▷ Bank account
- ▷ Phone bill, house rent, electricity bill

Primary Objectives (Mandatory) →▶
Second Objectives (Optional)→▷

3 Project Approach

We would like to develop a interface that would help its users to get easy access to the information which is required on a regular basis. This service will help in obtaining the data from different locations and application on a single front end.

We would mainly focus on the applications related to UB like UB learns, Webmail, Bus Schedule, Events and so on since these are the important application which almost all the students would be accessing. Secondly we would also like to integrate some personal information like Gmail / Yahoo, Social networking applications like Facebook or Orkut. We would also like to have a agenda on weekly or monthly basis which can show the academic course schedule, assignment dealines, examination dates, due dates for bill payment for credit card, gas, electricity, house rent and so on.

This service will also have a drag and drop feature. The front end can configured by picking the required application from the list and viewing it on the front end. This feature is made available to provide some flexibility to its users.

4 Project Design

4.1 Architectural diagram

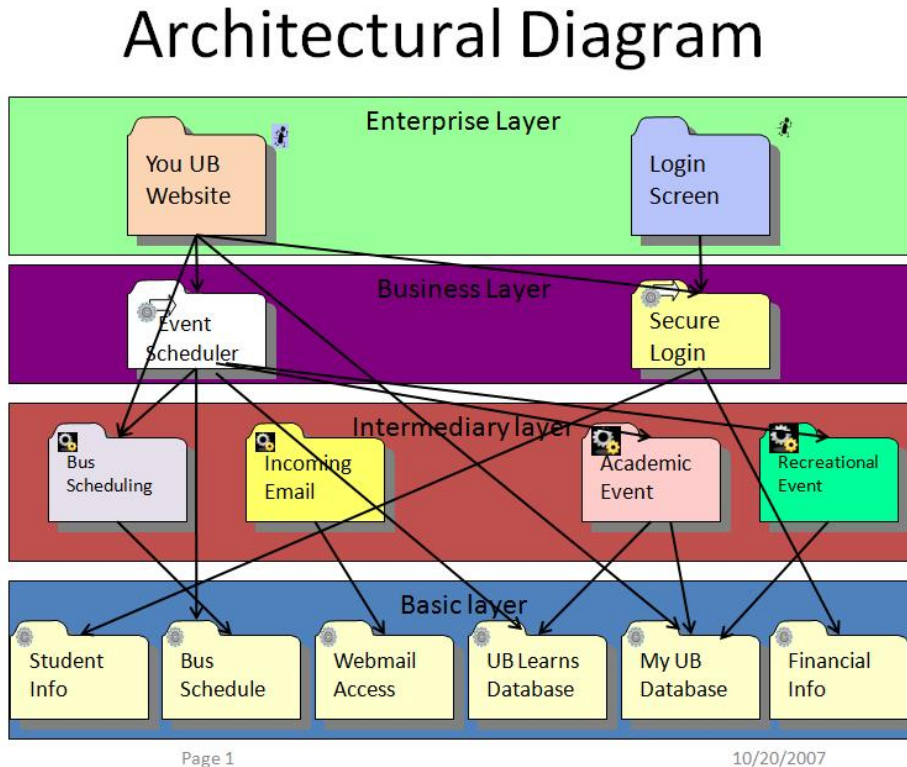


FIG. 1 – Architectural Diagram

4.2 Architecture explanation

You UB is a frontend service that would provide all the commonly used information in one go.

The website will have a login screen which would provide access to the user to go ahead and use this service. This login would be connected to secure login which would contain the student information and other relevant details.

The website will contain an event scheduler which would give easy availability of the list of activities to be taken care of. This would mainly contain the important academic as well as recreational activities such as sport, dance, performances and so on. We plan to obtain the academic and recreational information from UB Learns and My UB as a source. We also plan to integrate UB webmail service along with YouUB. This would help in being updated about the latest incoming email without login into the Webmail service.

We also plan to focus on the bus schedules and its availability at different times. And finally apart from the above mentioned feature would like to integrate some personal information such as Mobile bills, Electricity and Gas bill due dates and if possible bank details as well.

5 Implementation Details

5.1 Details

Starting with the Inter Campus Bus Schedule, we have implemented a web service which would give an option of selecting a bus stop to the customer. For example, if Flint Loop is selected, then the output of the web service will give timings in terms of hours and minutes for next 3 buses scheduled to arrive on Flint Loop from the particular time request was made. This service will help its consumers to plan and schedule their work accordingly before they wish to take the bus.

Whenever a request is made by a consumer, the web service gets the information from the UB database where the bus schedules are available. Hence whenever there is a change in the schedule in the UB database, the web service would always return the latest information.

The structure of the web service is such that it initially registers the entry of the bus stop made. It then has a checker which scan through the bus schedule database after considering the time at which request was made and by default returns the next 3 timing of buses scheduled on that particular bus stop. It is also possible to make a entry of number of bus timings required along with the bus stop. This type of request would generate the timings of the buses as per the number specified. There also would be situation if it is late in the night that only 3 more buses are not available, then in that case the output generated will have only 2 or 1 timings. If no buses are available then output will show no bus. It would also be possible enter a particular bus stop and a particular time and the web service would generate the next 3 bus timings for the criteria specified.

5.2 Code

5.2.1 Main java service

```
/*
 * BusonTime.java
 *
 * Created on 16 novembre 2007, 19:21
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */

package org.piran.ubservices;
import org.piran.library.checker;

import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.jws.WebService;

/**
 *
 * @author pierrequemard
 */
@WebService()
public class BusonTime {
    checker check = new checker();
    /**
     * Web service operation
     */
}
```

```

    */
    @WebMethod
    public int [] getNextBus (@WebParam(name = "busstop") String busstop, @WebParam(name = "hour"
        int [] res = check.nextBus (busstop ,hour ,min ,3);
        return res ;
    }
    @WebMethod
    public int [] getNBus (@WebParam(name = "busstop") String busstop, @WebParam(name = "hou
        int [] res = check.nextBus (busstop ,hour ,min ,ite );
        return res ;
    }
}

```

5.2.2 Libraries

```

/*
 * checker.java
 *
 * Created on 16 novembre 2007, 19:33
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */

```

```

package org.piran.library;
import org.apache.tools.ant.taskdefs.Java;
import org.piran.library.busStopStruct;

```

```

/**
 *
 * @author pierrequemard
 */

```

```

public class checker {

    private busStopStruct [] BusSchedule;

    public checker (){

        //Initialisation of all the BusStop
        /*
         * @TODO Generate those table using the xls file on internet !
         */
        int [][] tempo = new int [24][5];
        int [] temp2 = new int [5];
        int [] temp1 = new int [1];
        temp1[0]=-1;
        int ite=5;
        for (int i=0;i<5;i++){
            temp2[i]=ite;
            ite=ite+10;
        }
        for (int i =0;i<7;i++){
            tempo[i]=temp1;
        }
        for (int i =7;i<24;i++){
            tempo[i]=temp2;
        }
        BusSchedule = new busStopStruct [2];
        BusSchedule[0] = new busStopStruct (" Flintloop" ,tempo);
    }
}

```

```

    BusSchedule [1] = new busStopStruct (" Gouvernor" ,tempo);
}
public int [] nextBus (String name, int hour, int min, int ite){
    int val = 0;
    busStopStruct actualbus=BusSchedule [0];
    int actualhour=hour;
    int actualmin=min;

    while (this .BusSchedule [ val ].getName ()!=name && val<this .BusSchedule .length){
        actualbus=this .BusSchedule [ val ];
        val++;
    }

    if (val>=this .BusSchedule .length){
        // if the bus stop wasnt found or doesnt exist
        int [] res = new int [1];
        res [0]=-1;
        return res;
    }else{
        //The Bus stop exist
        int [] res = new int [ite*2+1];
        res [0]=0; //Let's assume no value where found
        for (int i=0;i<ite;i++){
            int [] tempres = this .horSui vant (actualbus ,actualhour ,actualmin );
            if (tempres [0]==-1){
                //No BUS at this HOUR
            }else{
                res [res [0]*2+1]=tempres [0];
                res [res [0]*2+2]=tempres [1];
                res [0]++;
                actualhour=tempres [0];
                actualmin=tempres [1]+1;
            }
        }
        return res;
    }
}

}
private int [] horSui vant (busStopStruct bus, int hour, int min){
    int [] minuten = bus .getMin (hour);
    if (minuten [0]==-1){
        //No bus at this hour
        hour=hour+1 %24;
        min=0;
        return horSui vant (bus ,hour ,min );
    }else{
        int iteration=0;
        int resultat=minuten [0];
        while (minuten [iteration]<min && iteration<minuten .length){
            iteration++;
            if (iteration<minuten .length){
                resultat=minuten [iteration ];
            }
        }
        if (iteration>=minuten .length){
            //No bus on this hour for those min
            hour=hour+1 %24;

```

```

        min=0;
        return horSuivant (bus , hour , min );
    }else{
        int [] res= new int [2];
        res [0]=hour ;
        res [1]=resultat ;
        return res ;
    }
}
}
}

/*
 * busStopStruct.java
 *
 * Created on 16 novembre 2007, 20:42
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */

package org.piran.library;

/**
 *
 * @author pierrequemard
 */
public class busStopStruct {

    private String name;
    private int [][] hours;

    /** Creates a new instance of busStopStruct */
    public busStopStruct (String name, int [][] hours) {
        this.name=name;
        this.hours=hours;
    }

    public int [] getMin (int hour){
        try{
            return this.hours [hour];
        }catch (java.lang.Exception e){
            System.out.println ("###Error while reading hours");
            int [] res = new int [1];
            res [0]=-1;
            return res;
        }
    }

    public String getName(){
        return name;
    }
}

```

5.2.3 JUnit tests

```

/*
 * checkerTest.java
 * JUnit based test

```

```

*
* Created on 16 novembre 2007, 22:03
*/

package org.piran.library;

import junit.framework.*;
import org.apache.tools.ant.taskdefs.Java;
import org.piran.library.busStopStruct;

/**
 *
 * @author pierrequemard
 */
public class checkerTest extends TestCase {

    public checkerTest(String testName) {
        super(testName);
    }

    protected void setUp() throws Exception {
    }

    protected void tearDown() throws Exception {
    }

    /**
     * Test of nextBus method, of class org.piran.library.checker.
     */
    public void testNextBus() {
        System.out.println("nextBus");

        String name = "Flintloop";
        int hour = 11;
        int min = 22;
        int ite = 3;
        checker instance = new checker();

        int [] expResult = {3,11,25,11,35,11,45};
        int [] result = instance.nextBus(name, hour, min, ite);
        for (int i=0;i<result.length;i++){
            assertEquals(expResult[i], result[i]);
        }
        /*
        name = "Flintloop";
        hour = 11;
        min = 34;
        ite = 3;
        instance = new checker();

        int [] expRes = {3,11,35,11,45,7,5};
        result = instance.nextBus(name, hour, min, ite);
        for (int i=0;i<result.length;i++){
            assertEquals(expRes[i], result[i]);
        }
        */
    }
}

```

}

}

6 Detailed plan

6.1 Implementation plan

We plan to :

- ▶ Integrate UB webmail service, information from UB learns, \Rightarrow Meet the person in charge of myub (UB IT) to see how they can use our web services on the myub web page.
- ▶ Develop the others services such as mail gathering and so on.

6.2 Presentation plan

We plan to finalize and integrate all the services to b implemented in our web service by 11/27/07. This would give us time to give the final touches required and prepare the final documentation for the presentation on 12/03/07