**Math Department Scheduler**

**Software Requirement**

**Pasadena City College**

**Spring 2014**

Version: 1.0

DEVELOPERS:

Saul Castro

Shin Kang

Ving Trung

Yekun Yang

1. **Introduction**

The program is mainly designed:

1. To create a Course Request form for instructors to send out request to select the desired courses. The request information will be stored in database.
2. To assist the Dean of Math Department to import the collected request information from database, organize and determine the final course schedule, providing several friendly options for the Dean, including add, remove, swap, and filter functions.
3. To assist IT technicians to view, edit, and manage the database.

1. **Specific Requirements**

1. Gather faculties’ informations.
2. Instructor’s name
3. Faculty ID (8 digits)
4. Requests

- Courses desired? (up to four courses)

- Time block for each course

- Additional note

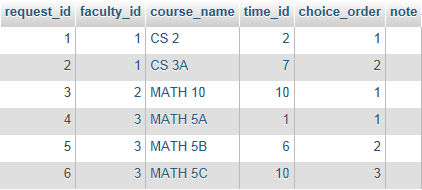
1. Display Instructor’s name, course name, time requested, and full-time or adjunct determinant on an excel sheet sorted by course name.
2. Determine if faculty is full time or adjunct.

a) Sort excel document to display a list of full time instructor and a list of adjunct.

1. **Class and Objects**

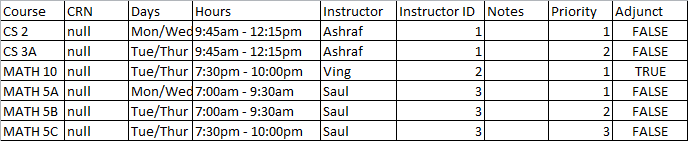
**All the input information will be stored in the database and can be imported by the 3b.jar file.**

**Input File** (example)



**Output:**

CSV file



1. CourseRequest Java Servlet Classes

**package com.admin**

**AdminBean**

Constructor Summary

public AdminBean()

Method Summary

private String read()

Call the XMLwriter object method connectParse() to parse settings XML into the object, then set AdminBean variables port, user, pass base on settings

private String write()

Write XML file by passing port, user, pass string variable into XMLwriter object method connectXML

public String getPort()

Return String for value of port variable

public void setPort(String port)

Set port string variable

public String getUser()

Return String for value of user variable

public void setUser(String user)

Set port string variable

public String getPass()

Return String for value of pass variable

public void setPass(String pass)

Set pass string variable

public String build()

Call write() and read() method to build the variables based on XML file settings. Use BuildSQL object method build() to build starting database

**BuildSQL**

Constructor Summary

public BuildSQL(){}

Method Summary

public static void build(String port, String user, String pass)

Build the starting database required to run program. Method requires the sql server access parameters to build

public static void addCourseData(Statement stmt)

Insert courses into database from predefined list

public static void addTimeData(Statement stmt)

Parse timeblocks into format compatible to database from predefined list

public static String insertTime(String day, String begin, String end, String cs)

Insert timeblock into database from predefined list

public boolean checkSQL()

Check if connection to database was made

public Connection getConStatement()

Connects to the database being used

**DataConnect**

SQL data connection controlling class used to interact between servlet and database

Constructor Summary

public DataConnect()

Method Summary

*public static Connection connect()*

Reads the connection requirements from an XML using XMLwriter class and build the connection to database.

*public static void runUpdate(String query)*

Runs mySQL update query that is passed through argument to database

*public static ResultSet runQuery(String query)*

Runs mySQL request query that is passed through argument to database. Returns the result in a ResultSet object

*public static void insertInstructor(String id, String name, String fulltime)*

Insert the Instructor data passed to function into instructors table in database

*public static ResultSet getTimeList()*

Run request query for all in timeblocks table

*public static int getTimeID(String time)*

Run request for time id based on timeblock sent

*public static ResultSet getTime(String timeid)*

Run request query to return results from timeblock table with timeid

*public static ResultSet getInstructor(String facultyid)*

Run request query to return results from instructor table with faculty id

*public static ResultSet getRequestList()*

Run request query to return all results from request table interjoined with instructor and timeblock data

**Request**

Constructor

*public Request(String instructorname, String instructorid, String fulltime, String course, String day, String begin, String end)*

Methods

*public String getDay()*

Get day variable

*public void setDay(String day)*

Set day variable

*public String getBegin()*

Get Beginning time variable

*public void setBegin(String begin)*

Set Begin time variable

*public String getEnd()*

Get Ending time variable

*public void setEnd(String end)*

Set End time variable

*public String getInstructorname()*

Get instructor name variable

*public void setInstructorname(String instructorname)*

Set Instructor name variable

*public String getInstructorid()*

Get instructor id variable

*public void setInstructorid(String instructorid)*

Set Instructor id variable

*public String getFulltime()*

Get fulltime variable

*public void setFulltime(String fulltime)*

Set fulltime variable

*public String getCourse()*

Get Course variable

*public void setCourse(String course)*

Set course name variable

**XMLwriter**

Use to build and parse XML files to store information

Constructor

public XMLwriter()

Methods

*public String getPort() {*

Get port variable

*public void setPort(String port) {*

Set port variable

*public String getUser() {*

Get user variable

*public void setUser(String user) {*

Set user variable

*public String getPass() {*

Get password variable

*public void setPass(String pass) {*

Set password variable

*public void connectXML(String port, String user, String pass)*

Create “connect.xml” file that will be saved with the input port, user, and password strings

*public void print(String filename, String text)*

Use printwriter to print out file with filename and text

*public void connectParse()*

Parse “connect.xml” for the port, user, and password string and set them to the class variable to be extracted

**package com.request**

**CourseListBean**

Constructor

*public CourseListBean()*

Use DataConnect application to run query to return all courses in course table of the schedule database into an arraylist

Methods

*public ArrayList<String> getList()*

Get arraylist of course names

*public void setList(ArrayList<String> list)*

Set arraylist of course names

**LoginBean**

Constructor

public LoginBean()

Methods

*public String buildsql()*

Builds starting database using BuildSQL application

*public String getName ()*

Get name variable

*public void setName (final String name)*

Set name variable

*public String getPassword ()*

Get password variable

*public void setPassword (final String password)*

Set password variable

**RequestBean**

Constructor

public RequestBean()

Methods

*public String getFulltime()*

Get fulltime variable

*public void setFulltime(String fulltime)*

Set fulltime variable

*public String getInstructorname()*

Get instructorname variable

*public void setInstructorname(String instructorname)*

Set instructorname variable

*public String getInstructorid()*

Get instructor id variable

*public void setInstructorid(String instructorid)*

Set instructor id variable

*public String getCourse1()*

Get course1 variable

*public void setCourse1(String course1)*

Set course 1 variable

*public String getCourse2()*

Get course2 variable

*public void setCourse2(String course2)*

Set course2 variable

*public String getCourse3()*

Get course3 variable

*public void setCourse3(String course3)*

Set course3 variable

*public void setCourse4(String course4)*

Set fulltime variable

*public String getCourse4() {*

Get course4 variable

*public String getTime1() {*

Get time1 variable

*public void setTime1(String time1) {*

Set time2 variable

*public String getTime2() {*

Get time2 variable

*public void setTime2(String time2) {*

Set time3 variable

*public String getTime3() {*

Get time3 variable

*public void setTime3(String time3) {*

Set time3 variable

*public String getTime4() {*

Get time4 variable

*public void setTime4(String time4) {*

Set time4 variable

*public void convertTime(){*

Converts time variable into time id

*public String build(){*

Call DataConnect application runQuery method to insert data in this java bean into the sql database

**RequestListBean**

Constructor

public RequestListBean()

Retrieve all requests from request table in database inner joined with instructor and timeblock data

Methods

*public Request getListRequest(int index)*

Set the index request from variable

*public ArrayList<Request> getList()*

Get Request arraylist

*public void setList(ArrayList<Request> list)*

Set Request arraylist variable

*public String getTest()*

Get Test variable

*public ArrayList<String> getStringList()*

Set Request String arraylist variable

*public void setStringList(ArrayList<String> stringList)*

Set Request String arraylist variable

**TimeListBean**

Constructor

*public TimeListBean()*

Use application DataConnect method getTimeList() to retrieve all times in timeblock table on database and fill an arraylist to be used on servlet

Methods

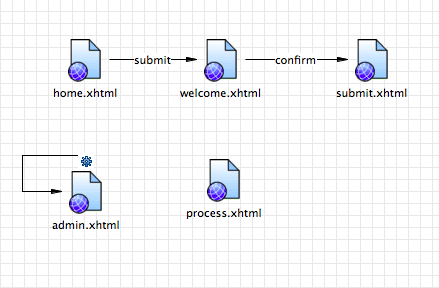
*public ArrayList<String> getTime()*

Get times arraylist

*public void setTime(ArrayList<String> time)*

Set times arraylist

1. CourseRequest Java Servlet Web Navigation



**home.xhtml**

starting webpage for instructors to configure their request

**welcome.xhtml**

display submission of request

**submit.xhtml**

confirms submission has been made

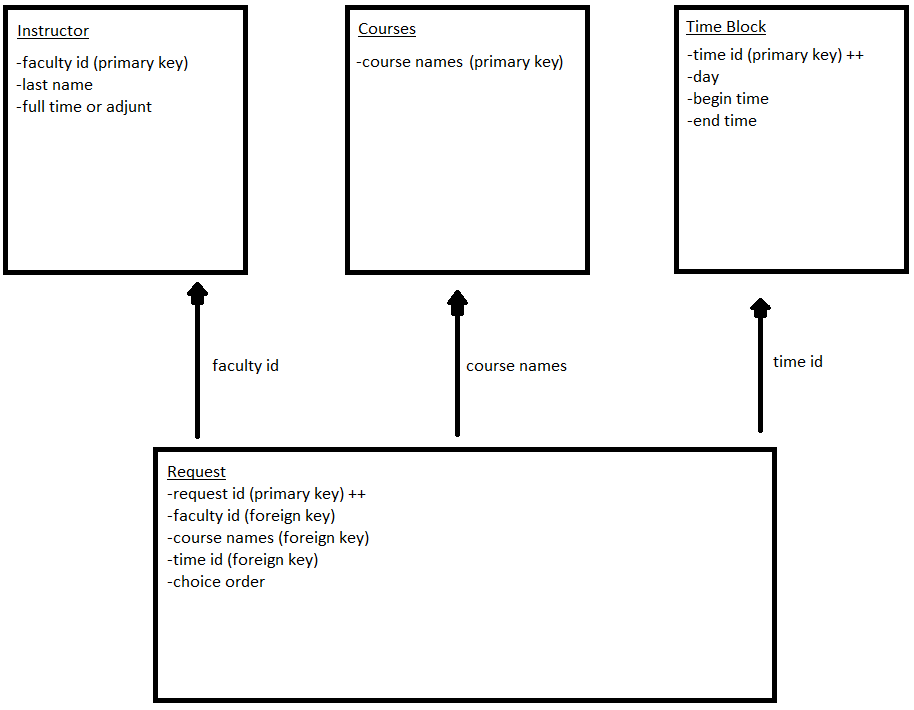
**admin.xhtml**

page for administrators to enter connection requirement for software to connect to database

**process.xhtml**

display all requests currently active

1. Database



The database in this project contains four tables. The instructor, courses, time block tables all have primary keys which is connected to a foreign key in the request table. A primary key is a unique value in the table meaning it will not repeat ever in that table. The Instructor Table will have the faculty’s id (primary key), faculty’s last name, and determining full time or adjunct as its data. The Course Table will only have one data, course names (primary key). The Time Block Table will have time id (primary key which will automatically increment), day, begin time, and end time. The Request Table will have request id (primary key which will automatically increment), faculty id (foreign key), course names (foreign key), time id (foreign key), and choice order which determines instructor’s first choice, second choice, and third choice for their class. Whenever the Request Table gets populated it will reference faculty id from the Instructor Table, reference course name from the Courses Table, and reference time id from the Time Block Table. The Courses and Time Block Table will be pre populated. When an instructor fills in their dream sheet, the Instructor and Request Table will be populated. Only an admin with the proper access will be able to access the Request Table.