

# Implementation and Evaluation of an Open Source Student Information System at Meio University

Peter Arathoon

## Abstract

We complete a full implementation of an open source student information system in our medium-sized university and evaluate its suitability. We further evaluate features and user interfaces for three classes of user, students, instructors and office staff, and draw conclusions on the range of features and their ease of use.

Students can log in to view their lesson schedule on a daily, weekly or monthly basis. This shows the lessons they are registered for, and a view of the daily lesson-plans published by the instructors (through simple entry in a browser window). Calendar-views include university events, and students can add personal entries to their own calendars, sharing this with chosen friends. Students can securely view on one screen their daily class-attendance record and mid-term test results for all of their subjects, as soon as the instructor has entered the data.

Instructors can keep online attendance and test records, and have the ability to add or remove students from class lists, which will greatly alleviate both confusion and office paperwork during the start of term class-registration period. At the end of term, they have the choice of either manually entering final grades, or alternatively using an automatic grade calculation based on mid-term and final test results. This can even be over-ridden on a student-by-student basis, while keeping a permanent record of the reasons for that decision. Weekly public online lesson notes can be used to publish the class syllabus, while private class notes can also be attached to each lesson.

Office staff can be given the authority to add or remove students from class lists, and view individual student timetables, monthly attendance records and grades. Altogether this should result in a decrease in the office workload and an increase in efficiency.

## 名桜大学でのオープンソース学生情報管理システムの導入シミュレーションと評価

アラースン・ピーター

### 要約

オープンソース学生情報管理システムSchoolToolの導入シミュレーションを行う。2008年度の学生・先生・授業・授業登録リスト・時間割情報のデータをシステムに登録し、3種類のユーザ（学生、先生、教務課の職員）の視点からシステムの特徴やユーザインタフェース(UI)等のシステム適性、機能や使いやすさを調査し、評価する。調査の手法は、今回時間の都合でアンケートを含まない著者の解析に基づく。関連の評価方法はGUIや機能の欠点を一覧で示す形になる。

学生がログインすると日・週・月の単位で時間割が表示され、登録した授業の授業内容や連絡事項及び大学イベントが閲覧できる。個人入力も可能し、各自のスケジュールを友人と共有することができる。一目で自分の出席記録や中間テスト結果、レポート評価、最終評価などの確認ができる。

教員は出席や中間テストなどの評価をウェブ・ブラウザで管理ができる。教員自ら学生を授業一覧に追加したり削除したりすることが可能になるため、登録調整期間中の混乱が回避される。学期末の評価はブラウザで入力ができ、またそれまでの宿題や課題の評価に基づいて自動評価もできる。評価を変更した場合、備考欄にはその理由を正確に残すこともできる。授業シラバスや・日当たりのプランを公開しながら、個人用の教案やメモなども記載できる。

## 1. Introduction to student information management systems

Many academic institutions feel the need to create central Student Information Systems (SIS), or to upgrade those they may already have in place. An increasingly wide range of student-related data must be handled by academic and office staff on a daily basis. This data is typically held in differently constructed databases spanning separate office departments, and often with further details held by the individual instructors for the various classes. Under the initiative of data disclosure, students should be reasonably able to access and confirm some subset of this data (attendance records for classes, for example, or records of their home addresses, mobile phone numbers, and perhaps photographic data, etc.). At the same time, access to this and other data needs to be strictly restricted to only those people with the appropriate authority. Ideally there should also be a system of version control which would enable confirmation of exactly how the data has been modified since its first entry (who changed what, how and when).

The usual way to meet this need is to commission the creation of a tailored system, commonly at considerable expense. Whenever such decisions are made, they inevitably involve the hidden built-in cost of any future changes, however small, to that system. And however much money the system may cost, the terms of the commission will typically guarantee the ownership of this system remains with the subcontracted company.

However, with the rise in popularity of the open source model, and as part of a trend certain to be fuelled by the recent downturn in the global economy, many institutions are choosing such an open system which can be legally tweaked by their own staff to adjust to their changing requirements. They will also enjoy the considerable benefits of constant bug-fixes and updates from the developer community.

Against this background, we chose to implement such a solution in the form of the recent beta release in November 2008 of the freeware SchoolTool, the development of which has been followed with interest for some years by the author. This paper describes some of the necessary steps involved and the problems encountered. We also make an evaluation of the software (although not able to expose it yet to a realistic workload of multiple users) in terms of its range of features, its suitability to its task, and the design of its user interface, from the perspective of students, instructors and office staff.

This system enables instructors, students and office staff to input or view the lesson-related student data over the net using nothing but browser software. The data instantly becomes widely accessible, yet remains centrally and securely held.

## 2. Implementation

The implementation process involved:

### 2.1 Creation of a Linux virtual machine environment and installation of SchoolTool.

The presently available version is built for a Linux platform. To accommodate this on the present Windows server requires a virtual machine (VM) arrangement. We have specified an allowance of 640MB of dedicated memory for this virtual machine, but this can be increased later if necessary. One advantage of the VM is the security gained by the isolation of this application from the main server OS.

### 2.2 Specification of the academic year.

Dates and descriptions of fixed events and university-specific class-holidays have been entered into the calendar.

### 2.3 Specification of the term structure and daily timing and naming of class periods.

Two terms T1 and T2 are set up. Seven 90-minute periods (named numerically) divide time into allocated slots for lessons to be scheduled in.

### 2.4 Preparation and import of a structured list of all subjects taught in one year.

Japanese titles including the class (A,B,C,etc.) and their code-representation in the Academic Affairs database are prepared. Imported data must be in csv format:

["Course-name (Japanese)", "Course ID"]  
eg: ["コンピュータ・リテラシー A", "18IIC30"]

### 2.5 Setup of student and instructor accounts

Raw data obtained from the Academic Affairs department for in excess of 2000 undergraduate, postgraduate and auditory students is rearranged using spreadsheet software in the csv format below. Permanent and temporary teaching staff data were obtained separately, verified, and non-conflicting login names were assigned.

[Unique login name, First Name (Japanese kanji combined with the katakana reading of it), Last Name (student number combined with romaji name), initial password]

eg: [s1408\*\*\*, 名桜 太郎 メイオ タロ, 1408\*\*\* Meio Taro, password]

### 2.6 Preparation and import of 2008 class registration data for more than 2000 students.

This process will soon be possible by importing xls-formatted data, hopefully in the scheduled official release in April 2009. However at the time of this beta-evaluation

it still required considerable preparation of class lists in the csv format described below. This was done automatically with a Python programming script.

normal_timetable	[時間割の名前]
	[空行]
18IIC30, arathoon	[コース ID、担当のユーザ名]
Monday, 3	[曜日と時限目：なくて良い]
Wednesday, 1	[週に複数授業の場合]
***	[登録したメンバー一覧前の区切り]
s1407***	[一人一行のクラスメンバー]
s1408***	
s2108***	
	[次のクラスデータ前には空行]

読み込むクラスのデータ

説明

Fig.1 Csv-formatted data for importing multiple class lists with teacher and timetable data

### 3. Problems encountered in the implementation

3.1 Multiple course-codes were used in the raw data for a single class, depending on the department affiliation of the intended students. These were unified into a single code.

3.2 Course names were cleaned (trailing symbols erased, class-identifier letters concatenated for clarity).

3.3 Raw-data codes for teachers were converted to their SchoolTool identities.

3.4 Raw-data timetable data was often missing or meaningless (eg day of the week 0).

Some such lessons have no timetable slot allotted, and thus will not appear in the calendar views until the teacher or office enters this data.

3.5 Raw-data had some teachers double-booked for the same time-slot.

One such data in the complete 17,000 line import data prevented the import of any data until it was found and rectified by hand.

3.6 Raw-data represented team-teaching slots in inconsistent ways.

Sometimes it only listed one teacher, and sometimes listed them separately as if separate synchronous classes existed. These cases included lessons held in multiple sessions per week. This was rectified manually, but will need to be handled by the

conversion programme in the next data import iteration.

### 3.7 SchoolTool documentation errors:

Through trial and error the author found that the documented format for importing data erroneously specified that the term name must be included. Import succeeded only when this was ignored. Secondly, importing duplicate data is documented to be handled without error, but actually inexplicably truncated the import process each time until this was discovered.

## 4. Use Cases

### 4.(1) Use Case(1): Students

#### 4.(1).1 Calendar and weekly class information

Upon login, students see today's **calendar** in the main window, and a top menu choice of:

Home **Gradebook** **Calendar** 2008 **Gradebook** (see Fig. 4)

The most common interaction might be to mouseover the events visible, and see an expanded description of each (Fig. 3):

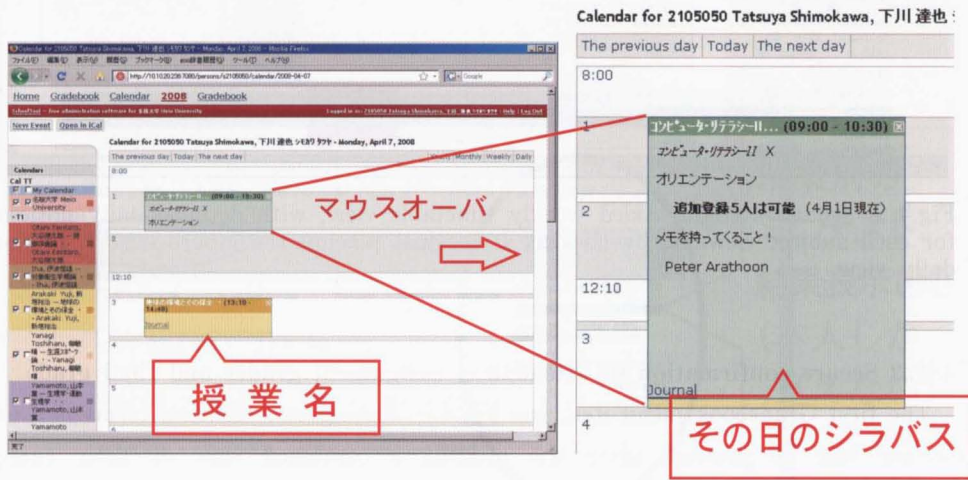


Fig.2 Daily timetable view seen on login

Fig.3 Mouseover an event for details

If the event is a lesson, they see any description the instructor has written, perhaps a summary of that day's lesson content, a reminder of work to be completed, or even a warning that the day's lesson has been cancelled, which may save a wasted journey to the university that day. If no event is scheduled on the same day, there is a list of the next day's events to the left. This calendar view can be expanded to a weekly timetable (Fig. 4), or further zoomed out to that whole month or year.

Events are colour-coded by lesson, with university-wide events also highlighted. Each group of events can be filtered out of the view by deselecting from the list at the left of the page, and clicking Apply. The student's own personal calendar is



created by clicking once on a time-slot, and entering the details. This group of events can then even be shared with friends if desired, (set in the Edit Preferences submenu under the Home menu).

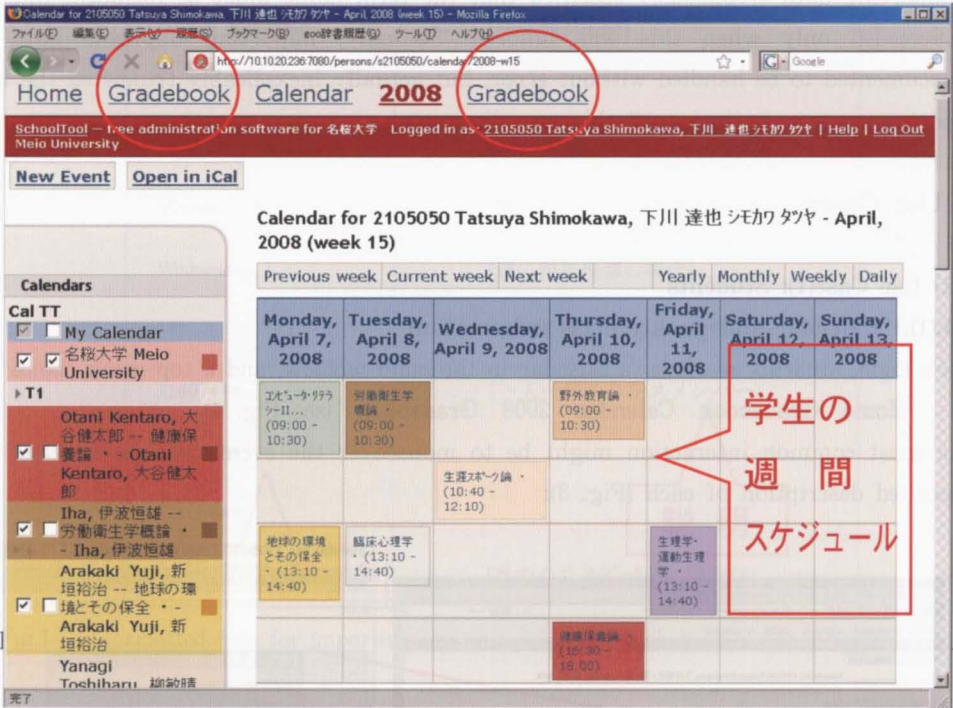


Fig.4 A student's colour-coded weekly timetable view, with detailed daily notices for each subject viewable by clicking it, or just placing the mouse over it in the daily view.

#### 4.1.2 Secure confirmation of attendance, assignment grades and final grade

The first **Gradebook** menu item (see Fig.4 — having two identical terms is a user interface bug at present) gives the student a combined view of their current attendance record across all the subjects they are enrolled for, day by day in monthly views (Fig. 5). This gives feedback of their present status, which can be very important towards the end of term, since 6 absences signal an automatic Fail grade. And after grading, the same view shows the student their own grades as soon as they have been entered by the teacher. This speedy feedback allows the student to choose relevant courses in their end-of-term automatic online registration for the next term, avoiding unnecessary reregistration for lessons they have actually passed. This behaviour would up to now have deprived another student the chance to register on a popular course.

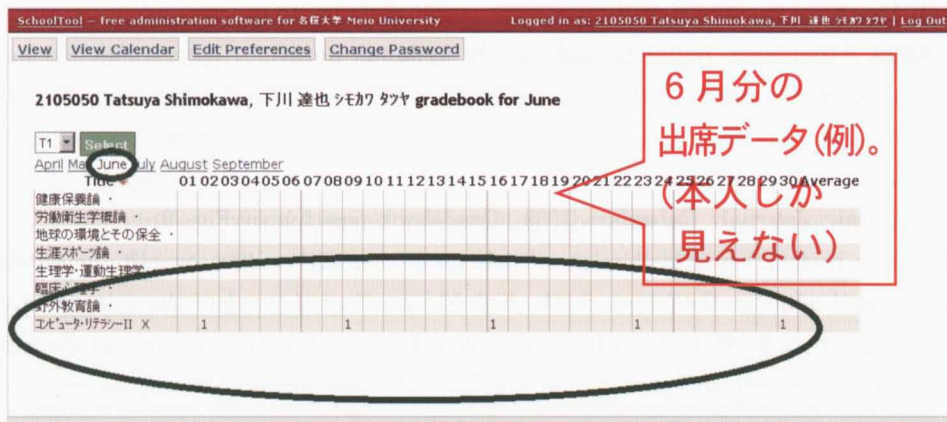


Fig.5 **Gradebook**① shows a combined view of a student's current attendance record across all the subjects they are enrolled for (only one subject data seen here, コンピュータ・リテラシー II X)

The second **Gradebook** menu item (Fig.4) is confusingly named at present, but again gives private feedback of essential lesson data. This time it is a one-page summary of realtime results of mid-term mini-tests or assignments such as reports. Clicking on each registered subject shows detailed individual results (Fig. 6).

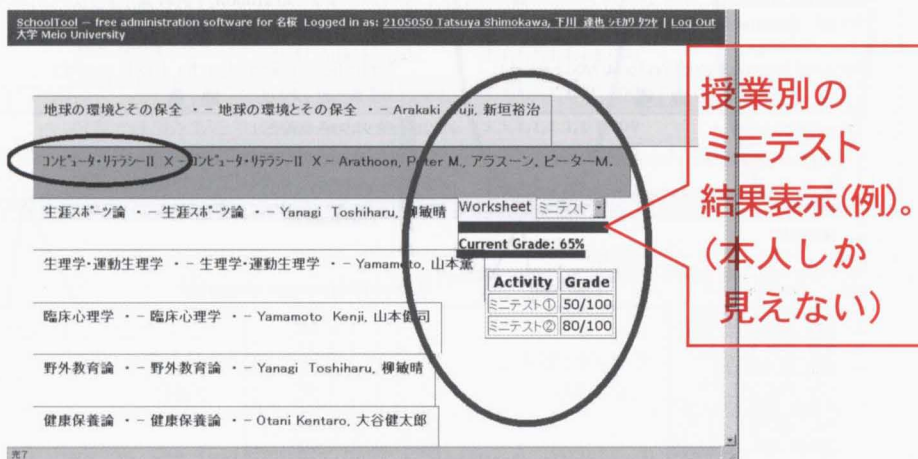


Fig.6 **Gradebook**② shows detailed personal results for each assignment ("Activity") within each group of assignments ("Worksheet"), for all of that student's registered subjects.

The **Home** button on the header menu shows a printed list of registered lessons. Clicking on each gives a list of fellow students in that class, information which today's students find increasingly difficult to find out for themselves, in often insular surroundings.

## 4.(2) Use Case(2): Instructors

### 4.(2).1 Attendance registers and weekly syllabus notes

Just as for students, the most common tasks can be performed from the first screen at login, the daily Calendar view. The top menu is almost identical to that for students, with the exception of the Journal link:

Home Journal Calendar 2008 Gradebook (see top of Fig. 9)

Instructors click either a visible lesson title, (and then either view or edit the publicly visible lesson note for that day), or the link "Journal" at the end of the calendar event (see Fig. 3). This leads to the attendance record for that lesson, as well as the same editable public lesson noticeboard and a private (teacher visible only) lesson note area (Fig. 7). One use for the lesson note area would be to paste a copy of the syllabus into the first lesson of term's notes.

(NB The same-named link **Journal** in the header menu leads through a list of the teacher's lessons to a similar attendance record view, except without these two note areas. This difference in views depending on the route taken to reach it is confusing and needs improvement.)

SchoolTool - free administration software for 名城大学 Meio University      Logged in as: Arathoon, Peter M., アラスーン, ピーター・M. | Log Out

**Arathoon, Peter M., アラスーン, ピーター・M. -- コンピュータ・リテラシー II X - Arathoon, Peter M., アラスーン, ピーター・M. journal for April 2008**

Personal lesson description for 4/7/08 period 1      Public lesson description for 4/7/08 period 1

学生に質問すること:  
 学生がUSB持っている?  
 How many students have their own computers and internet connections?  
 検定取得?

オリエンテーション  
 追加登録5人は可能 (4月1日現在)  
 メモを持ってくること!  
 Peter Arathoon

April May June July August September Term ending

Nr.	First Name	Last Name	07	14	21	28	Average	Absences
1	甘利 明大	アマリ アキヒロ	:	:	:	:	1.000	1
2	池原 正将	イハラ マサノブ	:	:	:	:	1.000	1
3	金良 宗幸	キムラ ムネユキ	:	:	:	:	1.000	2
4	魚住 海	ウヅミカイ	:	:	:	:	1.000	1
5	内間 勇太	ウチマ ユウタ	:	:	:	:	1.000	1
6	喜屋武 由紀乃	キヤウキ ユキノ	:	:	:	:	1.000	4
7	新垣 優梨恵	シンガキ ユリエ	:	:	:	:	1.000	1
8	仲宗根 亮太	ナカネノ 亮太	:	:	:	:	1.000	1

先生の授業メモ (非公開)

学生が見える、授業内容などを書く領域 (公開)

Fig.7 The "Journal" link on the lesson event in the calendar leads to a comprehensive lesson attendance entry screen, complete with publicly visible noticeboard and private note areas.

Attendance records can be numbers 1 to 10, or "a" for absent and "t" for tardy (late) to class. These are entered directly in the displayed webpage as easily as in a spreadsheet. Automatic highlighting of new data until confirmation with the Update button, and automatic rejection of other numbers or letters, further improve the user experience here in direct, meaningful ways. Running averages are calculated, and a tally of absences displayed too. "Today's" data entry is vertical, unless a student is



selected, in which case data input becomes horizontal, for that person only, again a small but appreciated touch.

#### 4.(2).2 Grading Schemes

##### 2.1 Simple grading scheme

For the simplest record-keeping and grading, the Journal view described above is more than adequate. The link to "Term Grading" gives an overview of the term's attendance record, and an area for manual input of end-of-term grades(Fig. 8). It is this data which is viewable to logged in students on their Gradebook pages.

Nr.	Class	First Name	Last Name	Grades	Average Absences	Term Grade
1		甘利 明大 アツアキヒロ	1105003 Akihiro Amari	1,1,1,1,1,1,1,1	1.000	A
2		池原 正将 イケハラ マサノブ	1105005 Masanobu Ikehara	1,1,1,1,1,1,1,1	1.000	A
3		金良 宗幸 キナラ ムネユキ	1105040 Muneyuki Kinra	1,1,1,1,1	1.000 2	B
4		魚住 海 ウオズミ カイ	1106016 Kai Uozumi	1,1,1,1,1,1	1.000 1	C
5		内間 勇太 ウチマ ユウタ	1106018 Yuta Uchima	1,1,1,1,1,1	1.000	B
6		喜屋武 由紀乃 キンユキノ	1106030 Yukino Kyan	1,1,1,1,1	1.000 1	A
7		新垣 優梨恵 シンガキ ユリエ	1106047 Yurie Shigaki	1,1,1,1,1,1,1,1	1.000	A
8		仲宗根 亮太 ナカネ リョウタ	1106072 Ryota Nakasone	1,1,1,1,1,1,1,1	1.000	B
9		又吉 一斗 マタヨシ カズト	1106088 Kazuto Matayoshi	1,1,1,1,1,1,1,1	1.000	
10		村越 豪 ムラコシ ゴウ	1106097 Go Murakoshi	1,1,1,1,1,1	1.000 1	
11		西原 正吾 ニシハラ ショウゴ	1204085 Shogo Nishihara	1,1,1,1,1,1,1,1	1.000	

Fig.8 The "Term Grading" page shows term attendance figures, and allows manual entry of grades. An individual's data is visible to them on their Gradebook page.

##### 2.2 Comprehensive grading scheme

An alternative, fully comprehensive grading scheme can be accessed via the Gradebook link on the Instructor's page (Fig. 9). A variety of assessment data from mid-term tests to homework assignments can be recorded. The instructor can switch views between different subjects via a pull-down selection list. Gradeable assignments, called Activities, are grouped in one or more Worksheets of related activities, again displayed by a pull-down menu (Fig. 9).



Fig.9 The instructor's Gradebook: The comprehensive Activity view for data-intensive lessons.

With this alternative grading scheme, the final grade entry page is accessed through the "View Final Grades" button at the top of the page (Fig. 10). Grades are calculated automatically, but may be adjusted afterwards, with space to explain the reason for this.

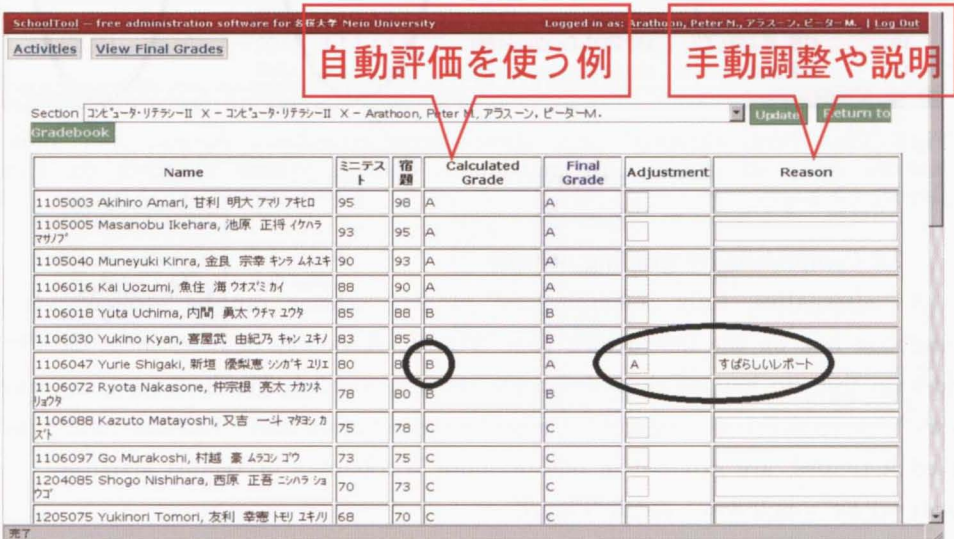


Fig.10 The instructor's Gradebook: The Final Grades view shows average scores by worksheet, and an automatically assigned grade from A to E based on the average of these again. Borderline criteria for these categories are the same as Meio university's guidelines (90%, 80%, 70%, 60%, less than 60%). Instructors can override grades however, and record an explanation (see ringed data showing calculated and adjusted grade).

Thus two alternative grading interfaces are provided, a simple Journal-based one and a detailed, automated-grading Gradebook. However although grades from the former scheme are visible to the individual student, the latter scheme only presents the detailed marks from all activities, and not the actual final grade itself. This may be addressed in later upgraded versions of the software.

#### 4.(2).3 Class-list management

Another useful function of this software is the teacher's ability to add or remove students from the class list. The present registration system has a two-week adjustment period in which students can change their choice of classes. In this time, newly registering students hand a registration slip to the teacher, which they must then forward to the office staff promptly for them to change the main database. In some cases the office accepts these new registrations directly, for example on behalf of part-time teachers. Then they must print out the updated list by the morning of the next lesson to inform the teacher of the changes. Inevitably this relay system sometimes cannot make the changes in time, and there is some confusion as to the integrity of the printed list.

With this system however, the teacher makes the changes to this separate database themselves, and all the office has to do is to wait till the end of the two-week period and output this constantly updated database in one operation to renew the central database. (If this export function exists, it is not at present visible from the manager's navigation pages. If it does not yet exist, then it needs to be added, either by request to the creators or by writing the necessary program internally.)

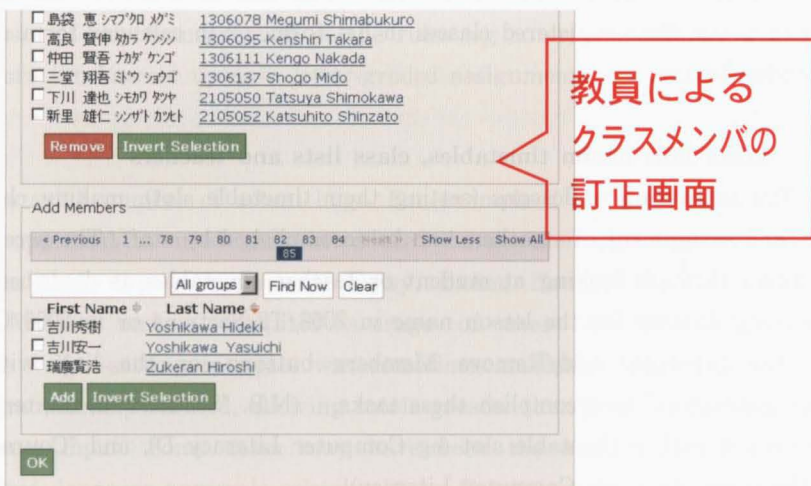


Fig.11 Changes to class lists are possible from Home/(class)/Add/Remove Members.

The **Add/Remove Members** link for instructors to change class lists is found in the class view when selected from the **Home** menu. The top half lists present members, who can be deleted by selecting the check box and clicking the **Remove** button. To add students requires first searching for them by either Japanese name, English name or student number, and then similarly selecting and clicking the **Add** button (Fig. 11).

#### **4.(2).4 View registered student's class calendars**

Another very powerful function of this class view is the ability to view a student's weekly class schedule, for example to find a convenient time to contact them. However, this involves privacy considerations, and this ability is automatically restricted to those instructors who teach that student. Accessing this feature is by clicking on the student name in the class list, which first shows a printed list of all classes registered. Clicking on the now-visible **View Calendar** button enables a calendar view of this data, customizable to weekly or monthly views as usual.

### **4.(3) Use Case(3): Office Staff**

#### **4.(3).1 Check student timetables and grades**

Office Staff accounts can be made members of the "clerk" group, which gives them appropriate authority to view student data which other students and instructors are prevented from seeing by the security system. They can then view students' (or teachers') timetables.

Since Office Staff teach no lessons, they cannot access student timetables through the **Home** link as teachers can. Instead, access is through the new **Manage** link in the header menu. From there, students can be searched by name or student number, and clicked to see their registered classes in list form, or in calendar form using the **View Calendar** button.

#### **4.(3).2 Edit lesson timetables, class lists and teachers**

The scheduling of lessons (setting their timetable slot), making changes to class lists and assignment of teachers can be accomplished by staff. The necessary view can be found through looking at student or teacher timetables as described above, or by searching directly for the lesson name in **2008/T2/Sections** or in **2008/Courses**. Then use the **Schedule**, **Add/Remove Members** buttons, or the link within the page "edit instructors" to accomplish these tasks. (N.B. "Sections" is the terminology used for classes with a timetable slot (eg Computer Literacy D), and "Courses" for groups of the same class, (eg Computer Literacy).



#### 4.(3).3 Check a student's attendance record

A student's total attendance record is available at the URL ..../persons/(student id)/gradebook.html. No formal link to this view exists at present. This view is automatically restricted to the "clerk" group, to the students themselves and to teachers whose class they are taking.

### 5. Evaluation of Features and GUI

#### 5.1 General comments

Overall this software package is a fully-featured plug-in answer to the growing needs of many institutions beyond the educational arena, filling the need for scheduling software in some cases, or for marksheet capability in others. It has been in development for several years, and is being experimentally employed in several large educational institutions around the world. The two main features, Calendar and Gradebook, are well-developed pieces of software, and most of the problems have been ironed out. They have for the most part very refined interfaces. Calendar has colour-coded daily calendar entries for all subjects and potentially for all registered users of the system, with a high-level mouseover effect which displays a large amount of information within a single screen using pop-up expanded views of event entries. Gradebook has a superbly easy-to-use web interface for attendance or marks input, with restricted areas of input (for example that day's attendance cells) traversed by Tab or Enter keys, without overflow into unselected parts of the interface. At the same time colour is used effectively to highlight regions of newly input data which are volatile (unsaved) until the Update button is selected. Data entry has detailed checks on the data-type and value-ranges of entries, and focused error messages in red specify the causes of any transgression of the rules (eg entering a number grade in an area the instructor has set up as a letter-graded assignment).

#### 5.2 Bugs

There are a small number of actual bugs. For example if the most detailed form of marksheet appraisal is used (not introduced here), then assignments can be categorized into their type (Assignment, Essay, Exam, Homework, Journal, Lab, Presentation, Project) and a weighted average of these separate assessments performed. The weight-setting interface was sometimes unstable, forgetting assigned weights when immediately verified, allowing totals greater than 100%, or not accepting seemingly correct input. Some cases may have been the author's fault, perhaps entering full-sized Japanese numerals mistakenly, or forgetting the confirmatory Add button, but either way the interface does not seem so polished, and I would recommend not using this feature until reported bugs have been solved. Once such

weight settings have been used, even if subsequently reset, the calculated averages become suspect. If this bug is not fixed by April, then we should tweak the code to hide this functionality from users.

Instructor input of "a" (absent) in the attendance journal is viewed as "n" in the student view, and "t" (tardy) as "p". This is not a serious bug, and will no doubt be resolved quickly.

The most serious bug found was the ability of students to delete a term's data. While this may have been a one-off case of a browser cache history effect remaining from an immediately prior Administrator login, it is one case of several GUI niggles which need attention. This is by far the biggest area which this user feels needs attention from the developer community before the April release. A few such instances are discussed below.

### 5.3 Suggestions for improving the GUI

- Most sub-menu links in the teacher's view of a class obtained from Home/(class name) navigation should be removed, as they either duplicate header-menu entries (Gradebook, Journal), have very limited use-cases (Timetable, View (class-only) Calendar), are more appropriately found elsewhere (Activities, found in header-menu Gradebook view), or should not perhaps be available to individual teachers (Schedule, for changing the regular day and period for holding the class). This would prevent many frustrating mis-navigations, and leave only the "Add / Remove Members" link. Then the view can be made identical to that seen on clicking Journal in the calendar.
- Pages need a clearer indication of which feature is being viewed. In many views, the links used to get to the present page remain unchanged at the top left of the page, suggesting that the navigation move has failed. Better practice would be to have clear page headers as confirming feedback. (eg Gradebook / View Final Grades: the page has no indication that after clicking the link, this is indeed the Final Grades page.)
- Clicking on a student name has different effects depending on the environment. If the Journal is entered from the "Journal" link in a calendar entry, clicking a student name highlights a row for easy horizontal data entry instead of the (usually appropriate) vertical entry. But if Journal is entered from the Home button first, then clicking a student instead shows a list of all their registered subjects, which can be useful but is at first disorienting. Within Gradebook, the same action pulls up a screen with just that student's data, but fully editable; this can be useful for confirming marks entered for one student while looking at the screen with the student, but without exposing the whole class's data. Thus the wide range of capability of the software entails getting used to the interface, unless some features are to be sacrificed for instant interface usability.
- Teacher notes are visible in Journal when entered via calendar, but are not when

entered via the Home link. (Similar to the above case. The solution is that teachers should enter Journal from calendar view, and that this view be the slim-sub-menu version proposed above.)

—Clicking on a subject from the teacher's weekly calendar view allows easy entry of the student-visible daily class notes, without entering the full Journal view. However either confirming with Apply or cancelling with Cancel will take the viewer to a calendar view *for that class only*. The teacher's other lessons appear to have disappeared from the calendar until this difference is realized, and the browser Back button clicked many times, or Calendar view reinvoked. Better navigation would be to return the viewer to the previous (teacher's) calendar view automatically.

—(Logged in as a student): The links to Journal in calendared classes should be removed, since students do not have authority to view it.

—(Logged in as a student): The duplicate-named but functionally different header-menu "Gradebook" links should be renamed, to "Attendance" and "Results" perhaps, since they link to class-wide attendance and assignment results feedback pages respectively.

—(Logged in as a student): The sub-menu on clicking "Home" should not have "View" or "View Calendar" links, as the first has no effect and the second duplicates the main menu Calendar button. Neither should there be "edit name" and "edit groups" links within the page, as the student does not have the authority to do so anyway, and is only faced with a login screen demanding a higher authority login.

—(Logged in as a student): The sub-menu on clicking "Home" and then a registered subject should not have "Activities", "Gradebook" or "Timetable" links. They are all either not relevant, not navigable deeper, or both. The remaining "View Calendar" link needs renaming to emphasize that it will only be the calendar for the chosen subject, which is of perhaps limited value.

—(Logged in as a student): Clicking academic year "2008" should not give students the chance to delete whole terms of data. This portion of the page must be removed. The "Groups" link should also be removed as students have no authority to view the linked page anyway.

—The weekly calendar view does not display the period-of-the-day numerical indicators at the left as in the daily view, which can be confusing in a sparse schedule because of vertical compression of intervening periods.

—There should be a clear choice forced on instructors as to which scheme they will use for final grades. This should be linked to the student Gradebook view, to show whichever choice was made. At present only Term Grading page (manually input) grades are visible for student confirmation, and not the View Final Grades version.

—The Clerk role of user has too much authority, and can see not only attendance records and final grades (desirable behaviour) but also a teacher's personal lesson

description area (which ought to be private).

—The Clerk role can still edit student final grades even after this authority has been explicitly removed from the administrator's access settings page.

#### **5.4 Data import/export functionality**

—The import facility for simultaneous import of multiple class registration/timetable/teacher datasets needs to accept assignment of multiple instructors to each class (at present it does not). It should also handle cases of repeat import of the same class gracefully (it does not, failing without a helpful message indicating the problem). Until this is improved, these cases need to be handled programmatically in the data preparation stage. This same program should catch instances of double-booking a teacher, which is another use-case which fails in the data import stage.

—The Sections view (administrator view only) needs a Data Export functionality to capture the teacher-edited class lists at the end of the two-week registration period. This data can then be reimported into the separate database which the Academic Affairs department uses for everyday tasks.

## **6. Conclusions**

The opensource SchoolTool software is a multi-featured student information management system incorporating a first class feature set comparable to any commercial offering available. This report shows that it is well capable of handling one year's data, consisting of 2 semesters, 2000 students, 100 academic staff and 575 classes. The response of the server under the light load conditions tested was very fast. The next necessary step is a full load test involving simultaneous access by a large number of students. Given correctly formatted data, importing a semester of registration data is routine. The remaining logistical problems are in removing as many examples of bad data as possible from the original source of raw csv format data, and in further improving the author's program for converting this into the required import format despite any remaining inconsistencies.

The software certainly has very valuable features for all three user groups considered, students, instructors and office staff, and deserves immediate implementation for the coming academic year. A more conventional evaluation of the GUI and features based on user questionnaires can then be made, which was neither the professed aim of this paper nor a realistic possibility, given the timing of the software release two months ago.

The present beta version is let down somewhat by the GUI. However, a set of improvements has been listed, implementation of the majority of which would in the author's opinion enable the system to be fairly evaluated as user-friendly.



## Bibliography and Links

SchoolTool OpenSource software: <http://schooltool.org>

Schaffer, E., *Institutionalization of Usability: A Step-by-Step Guide*, Addison-Wesley, 2004.

Del Galdo, E.M. and Nielsen, J. (Eds.), *International User Interfaces*, Wiley, NY, 2001.

Holmes, M., *Web Usability and Navigation: A Beginner's Guide*, McGraw-Hill, 2002.

Chandler, K. and Hyatt, K., *Customer-Centered Design: A New Approach to Web Usability*, Prentice-Hall, 2002.

Fleming, J., and Koman, R., *Web Navigation: Designing the User Experience*, O'Reilly & Associates, Sebastopol, CA, 1998.

Flanders, V. and Peters, D., *Son of Web Pages that Suck: Learn Good Design by Looking at Bad Design*, Sybex, San Francisco, CA, 2002.

Wroblewski, L., *Site-Seeing: A Visual Approach to Web Usability Hungry Minds*, New York, NY, 2002.

Snyder, C., Morgan Kaufmann, *Paper Prototyping: The Fast and Easy Way to Define and Refine User Interfaces*, San Francisco, CA, 2003.

Krug, S., *Don't Make Me Think: A Common Sense Approach to Web Usability*, QUE, Indianapolis, IN, 2000.