

A Student Management System for Reed Music Studios

A Manuscript

Submitted to

the Department of Computer Science

and the Faculty of the

University of Wisconsin-La Crosse

La Crosse, Wisconsin

By

Ha Bui

in Partial Fulfillment of the

Requirements for the Degree of

Master of Software Engineering

April, 2010

A Student Management System for Reed Music Studios

By Ha Bui

We recommend acceptance of this manuscript in partial fulfillment of this candidate's requirements for the degree of Master of Software Engineering in Computer Science. The candidate has completed the oral examination requirement of the capstone project for the degree.

Dr. Kasi Periyasamy

Examination Committee Chairperson

Date

Dr. Kenny Hunt

Examination Committee Member

Date

Dr. Mao Zheng

Examination Committee Member

Date

ABSTRACT

BUI, HA, V., “A Student Management System for Reed Music Studios”, Master of Software Engineering, May 2010 (Dr. Kasi Periyasamy).

Reed Music Studios in Onalaska, Wisconsin provides private instruction in piano, strings, flute and guitar. Most of the management activities in the studio are manual paper-based operations which are time consuming and cause many difficulties in management. The purpose of the Student Management System for the Reed Music Studios is to provide a system that helps the studio managers conduct their daily business more easily and improve the studio’s business. The daily activities include student lessons, payment for lessons, book sales and more. In addition, information about special events such as recitals conducted by the studio and participation in various competitions will also be maintained by the system’s database. The system is developed as a web-based application that provides users flexibility and convenience. Previous data maintained by the Studio was converted from Microsoft Excel and Word files to Microsoft SQL databases and normalized. This manuscript describes the design and implementation of the Student Management System and also includes the challenges and problems faced during the development and deployment.

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to my project advisor Dr. Kasi Periyasamy for his valuable guidance and his patience to explain every difficult thing to me. I also would like to express my thanks to the project's customers, Rita Schuman and David Reedy, Reed Music Studios' managers, for providing helpful suggestion for my project. I would also like to express my thanks to the Computer Science Department, University of Wisconsin-La Crosse for providing me the good studying environment so I could have enough knowledge to do this project. I'm also grateful for the advice from my friends at the FPT software company, Hanoi Vietnam. Finally, I wish to thank my parents and friends for their encouragement during the duration of this project. I could not have finished it without their support.

TABLE OF CONTENT

ABSTRACT	i
ACKNOWLEDGEMENTS	ii
TABLE OF CONTENT	iii
LIST OF TABLES	iv
LIST OF FIGURES.....	v
GLOSSARY	vi
1. Background information.....	1
2. Design and Development of the Student Management System	4
2.1 Life Cycle Model for the Project.....	4
2.2 Development of the project	5
2.3 Collecting software requirements.....	5
2.4 Incrementing prototyping.....	8
3. The Student Management System Application	9
3.1 User classification and characteristics.....	9
3.2 High level architectural design.....	12
3.3 Detailed design architecture	13
3.4 Database design.....	17
3.5 Security Design	26
3.6 User Interface Design.....	27
3.7 Online help system	28
3.8 Testing.....	29
3.9 Deployment	29
4. Limitations.....	31
5. Continuing Work	32
6. Conclusion	33
BIBLIOGRAPHY	34
APPENDIX A: Selected screen shots	35

LIST OF TABLES

Table 1. User roles and their functionalities	10
Table 2. Core functionalities	14
Table 3. Functionalities provided by third party software	15
Table 4. User schema description in chapter 4.....	22
Table 5. Category schema description in chapter 4	22
Table 6. Sub-Category schema description in chapter 4.....	22
Table 7. Inventory schema description in chapter 4.....	23
Table 8. Order schema description in chapter 4.....	24
Table 9. Student Balance schema description in chapter 4	25
Table 10. Student Request Schedule schema description in chapter 4	25
Table 11. Final Schedule schema description in chapter 4	25

LIST OF FIGURES

Figure 1. Use case diagram for the student management system.....	11
Figure 2. High level architectural design	12
Figure 3. Class diagram for the student management system	16
Figure 4. Database design for the student management system.....	20
Figure 5. Login screen.....	35
Figure 6. User dashboard	36
Figure 7. Create user wizard	37
Figure 8. Search users	38
Figure 9. Add inventory	39
Figure 10. Search inventory	40
Figure 11. Add order	41
Figure 12. Search order	42
Figure 13. Inventory report	43
Figure 14. Inventory report (2).....	44
Figure 15. Personal schedule.....	45
Figure 16. Request schedule screen	46
Figure 17. Teacher’s available schedule	47
Figure 18. Create album and upload image.....	48
Figure 19. View album.....	49
Figure 20. Send a reminder	50
Figure 21. Change password	51
Figure 22. Online help system – common topics	52
Figure 23. Online help system – search feature	53

GLOSSARY

ASP.NET

A Web application framework developed and marketed by Microsoft to allow programmers build dynamic websites, web applications and web services.

HTML

Hypertext Markup Language. A predominant markup language designed for web pages.

JSP

Java Server Pages is a Java technology that helps software developers serve dynamically generated web pages based on HTML, XML, or other document types.

Microsoft SQL Server

A relational database management system (RDBMS) developed and marketed by Microsoft. Its primary query language is Transact-SQL, an implementation of the ANSI/ISO standard Structured Query Language (SQL) used by both Microsoft and Sybase. [2]

SRS

Software Requirements Specification, a complete description of the behavior of the system to be developed.

Visual Studio .NET

An Integrated Development Environment (IDE) from Microsoft. [3]

1. Background information

An organization that uses a reasonably large volume of data on a daily basis need to ensure the integrity of data it manipulates. When such an organization uses manual processes for handing the data, the processes not only take more time but also will be error-prone. Automation of those manual processes using software products appears to reduce unnecessary manual work hours, ensure consistency of data and hence improve the quality of work.

The Reed Music Studios in Onalaska, Wisconsin is one of the organizations addressed in the previous paragraph which happens to do a number of activities manually using paper and pencil. Founded in 1987, the studio provides music classes for a variety of instruments and vocal training. Currently the studio has 14 music teachers and nearly 250 students. The enrollment has grown steadily over the years and is expected to grow in the future as well. This significant increase in the number of teachers and students warrant the necessity of a computerized system for the studio in order to easily manage its day to day activities.

The studio currently manages most of its schedule and other activities (inventory management, order management, tuition management etc.) using spreadsheets and WORD documents. This process is very time consuming for the studio's managers, and also leads to inconsistency and incorrectness in data maintenance. The aim of the current project was to develop a web-based application that provides an effective student management system for the studio. The following functionalities were expected to be included:

- Maintain teachers' information (add/delete/modify).
- Maintain students' information (add/delete/modify).
- Assign/de-assign student to/from instructor; this is expected to be done manually and occasionally.

- Manage payments from parents. Each instructor is paid separately by the parents of kids taking lessons with that instructor. The software must generate automatic reminders for payments which will be emailed to the parents.
- Develop a schedule for lessons. This includes several constraints including the following:
 - The teachers enter their lesson time schedule into the system. This information will be available for parents and kids when they choose their schedule for lessons.
 - Parents and kids may prefer specific time slots, especially when more than one kid comes from the same family. The software was expected to support requests from parents which will later be honored by the instructors, if the requested time slots are available.
 - Kids from the same family are expected to be allocated contiguous time slots if they are taking lessons from the same instructor.
 - Continuing students of the studio must be given preferences over new students.
- Maintain information on various competitions and recitals. In addition to frequent recitals conducted by the studio, the kids from the studio participate in various competitions such as Honors recital, State competition from Wisconsin Music Teachers Association and so on. This information is a performance indicator which could be used for future improvement of the studio activities.
- Maintain all information related to inventories and orders in the studio such as instruments, accessories, books etc. This is the additional request that supports the daily tasks of the studio managers.

- Generate reports related to the inventories database. This feature is to help the studio managers get a quick overview of what the studio has currently in its inventories database.
- Manage messages and reminders sent by users to other users in the system. This feature is like a small messaging system in which the managers/teachers can send/receive/delete/reply to the reminders whereas the parents/students can only receive/reply to them.
- Security of information is another major concern since the product is web-based.

2. Design and Development of the Student Management System

2.1 Life Cycle Model for the Project

A software life cycle describes the development activities of a software product. It refers to a coherent sequence of well-defined software development processes, from conception to eventual retirement over the course of a software product's lifespan. The various stages and activities performed in each stage may vary depending on the life cycle model chosen. There are several software life cycle models described in the literature. However, there is no single life cycle model that is best suited for every project [1]. Different software life cycle models may complement each other on the same project during different stages of the development of a project. Choosing the most appropriate model for a particular software project is challenging. Many factors such as availability of resources, deadlines, developer's skill etc. may influence the choice of a life cycle.

For the current project, the customer was not sure of all the requirements of the system at the beginning of the project. Therefore, the developer initially used a rapid prototyping model to demonstrate the initial set of requirements by developing a throw-away prototype. This prototype helped the developer identify most of the requirements of the project. After discovering the most important as well as the set of anticipated requirements and having feedback from the customer, the developer primarily used the incremental prototyping model for the rest of the project. The gradual introduction of requirements in each iteration provided sufficient time for the customer to adjust to the system while also allowing for adjustment of expectations and responsiveness to changes [4]. This also helps the developer test and debugs the system easily and gets reality check from customer feedback.

2.2 Development of the project

The first phase of development used the rapid prototyping model collecting software requirements from the studio's managers and also investigating the document management of the studio. This was done to produce the SRS for the product. Requirements were then analyzed and a detailed design document was written. In the next phase, the product was developed using the incremental prototyping model.

2.3 Collecting software requirements

During the initial requirements gathering phase, the developer had a few meetings with the studio's managers and a few other meetings with the project adviser in order to understand the application domain of the project as well as the technology and tools to be used for the development. Some of the major requirements that were collected during this process are listed below:

- Maintain teachers' information (add/delete/modify).
- Maintain students' information (add/delete/modify).
- Assign/de-assign student to/from instructor; this is done manually and occasionally.
- Manage payments from parents.
- Develop a schedule for lessons.
- Maintain information on various competitions and recitals.

During the discussions, the following additional requirements were also added:

- Facilitate studio's inventory/order database management.
- All data must be validated in order to prevent inconsistency.
- A reminder sub-system must be introduced to notify users about important events like meeting, payment due date etc.
- The user with administrative rights (the studio managers) must be able to grant privileges to a set of users depending upon their roles in the studio (teachers, students).
- Security - Authentication: All users in the system need to login with a username and password. They should be able to change their own passwords. If they forget their passwords, there must be a password recovery wizard to create new password.
- Security – Authorization: The set of functionalities invoked by a user depends on the type of the user.
- Security – Confidentiality: The system must support encrypted passwords and the ability to encrypt data stored in database.

In addition to the above requirements, a careful study of the technical expertise of the various types of users in the system (mostly managers and teachers from the studio) revealed that the user interface of the system must be easy to use and user friendly because most of the users have little experience using computers. At this point, the project adviser suggested that web application would be the best solution for the users of the system. The advantages and disadvantages of a web application system are listed below:

Advantages of using web-based user interface:

- Deployment: eliminating client administration is possible to achieve if all processing is handled by the server. Though it may look like the server is overloaded, in this particular project, the volume of data is not significant that creates a performance issue.

- Portability is high because clients only need web browsers. Since most users of this system are not computer experts, installing and upgrading client side software should be minimum or null. It is therefore more convenient for a remote administrator to maintain the product without going to the studio.

Disadvantage of using web-based user interface:

- There are cross-browser issues since not all browsers render everything in the same way, especially older browser versions.
- There are no real standards that currently exist for web-based applications.
- There is no drag and drop facility in web interfaces without additional programming effort, intelligence, auto-completion and context-sensitive help.
- Responsiveness from servers can be an issue depending on bandwidth and graphics as well as the other factors. However, this does not seem to be an issue for the current project.
- The system will be unavailable if there are issues due to network problems.

At the beginning of the project, both the developer and the studio's managers had agreed that prototyping each subset of the project's functionalities would be of great value. It helped in refining requirements, and evaluating the feasibility of the product. The first prototype was completed in March 2009. Technologies including JSP/Servlet and ASP.NET were examined. Each technology is a server-side scripting language that embeds users' code, or logically separated for the application code and HTML for the presentation layer. Each of the languages is capable of creating a cross-platform, cross-browser web-based application as they return naïve HTML code to the clients after all the server-side code has been processed and replaced by the engines that process the server script. However, based on the developer's experience, Visual

C#.Net was chosen as the programming language and ASP.NET was chosen for developing the server-side scripting. Microsoft Visual Studio 2008 was used as a development environment for the project. Furthermore, Microsoft SQL Express 2008 was used as the database tool. It was easier to develop the user interface in ASP.NET and C# because of the developer's previous experience in that technology. LINQ was used to connect/update/retrieve data from SQL database.

The first prototype included user logins and a few record queries to establish communication between the C#/ASP.NET project files and the SQL Server. The prototype implemented the screen shorts for most of the functionalities. It also included some additional enhancements that the managers had asked for. The managers' feedback was positive and the next prototype was started.

2.4 Incrementing prototyping

After the first prototype, the actual development process was started. When each subset of functionalities was finished, the system was carefully unit-tested, and the developer performed integration and system testing as well. The developer then uploaded the system into his personal web server so that the studio managers can test the system and give a feedback to the developer. The product was then refined according to the customers' expectations. By using this prototype, the customer got more involvement in using the system, and it also helped the developer to closely match with the customers' expectations.

The system was beta-tested mostly by the studio's managers and teachers first. All the functionalities in the system reflect their manual daily tasks in the studio, and so they were able to comment on changes and improvement. After a finite number of iterations and positive feedback, the final software package was given to the customer. It will be deployed to the current studio's web server.

3. The Student Management System Application

This section explains the user classification, high-level architectural design, detailed design, database design, security design, and user interface design along with some detailed usability issues that were some of the factors considered in the development process.

The Student Management System is composed of three layers - a presentation layer, an application logic layer, and a database layer. By designing each layer independently, it is possible to change one layer without affecting (or reduce changes made to) the other layers. For example, the presentation layer (the user interface) can be modified without affecting the application layer. Likewise any change to the database design can be made with a little modification to the other two layers. The three-layered architecture thus supports easy maintenance.

3.1 User classification and characteristics

The application is a multi-user system. There are mainly three types of users: managers, teachers and parents/students (parents will be a part of the system if the students are still young).

- Managers are responsible for creating and maintaining user accounts, keeping track of inventories/orders, managing category (like instruments, books etc.) and sub-category of inventories (for example, in instrument category, there are sub-categories like violin, guitar etc.), and managing competition information and photos as well as generating reports.
- Teachers are responsible for managing their individual teaching schedules. They can also manage information regarding tuition payments made by students or parents every month.

- Parents/Students are able to view teachers' schedules and then request lesson schedule from a teacher.

The managers and teachers are able to create and send messages and reminders; they are also able to receive messages and reminders from other users in the system as well as able to reply back to the messages. Parents/students can only receive messages and reply to reminders. The managers and teachers could also manage their own daily schedule. Login authentication for all types of users is done by their username and password. Table 1 shows the user types and the functionalities that can be performed by each user type.

User type	Functionalities
Manager	<ul style="list-style-type: none"> • Add/Modify/Delete/Search users in the system. • Add parent/guardian information of a student. • Add/Modify/Delete/Search category/sub-category in the system. • Add/Modify/Delete/Search inventory in the system. • Add/Modify/Delete/Search order in the system. • Create albums and upload images. • Generate cost and inventory report. • Create/ Reply to/Delete reminders. • Personal schedule management.
Parents/Students	<ul style="list-style-type: none"> • Request for lesson schedule. • Reply to/Delete reminders. • Personal schedule management.
Teacher	<ul style="list-style-type: none"> • Payment management. • Create/ Reply to /Delete reminders.

Table 1. User roles and their functionalities

The use case diagram for the entire system is shown in the figure 1.

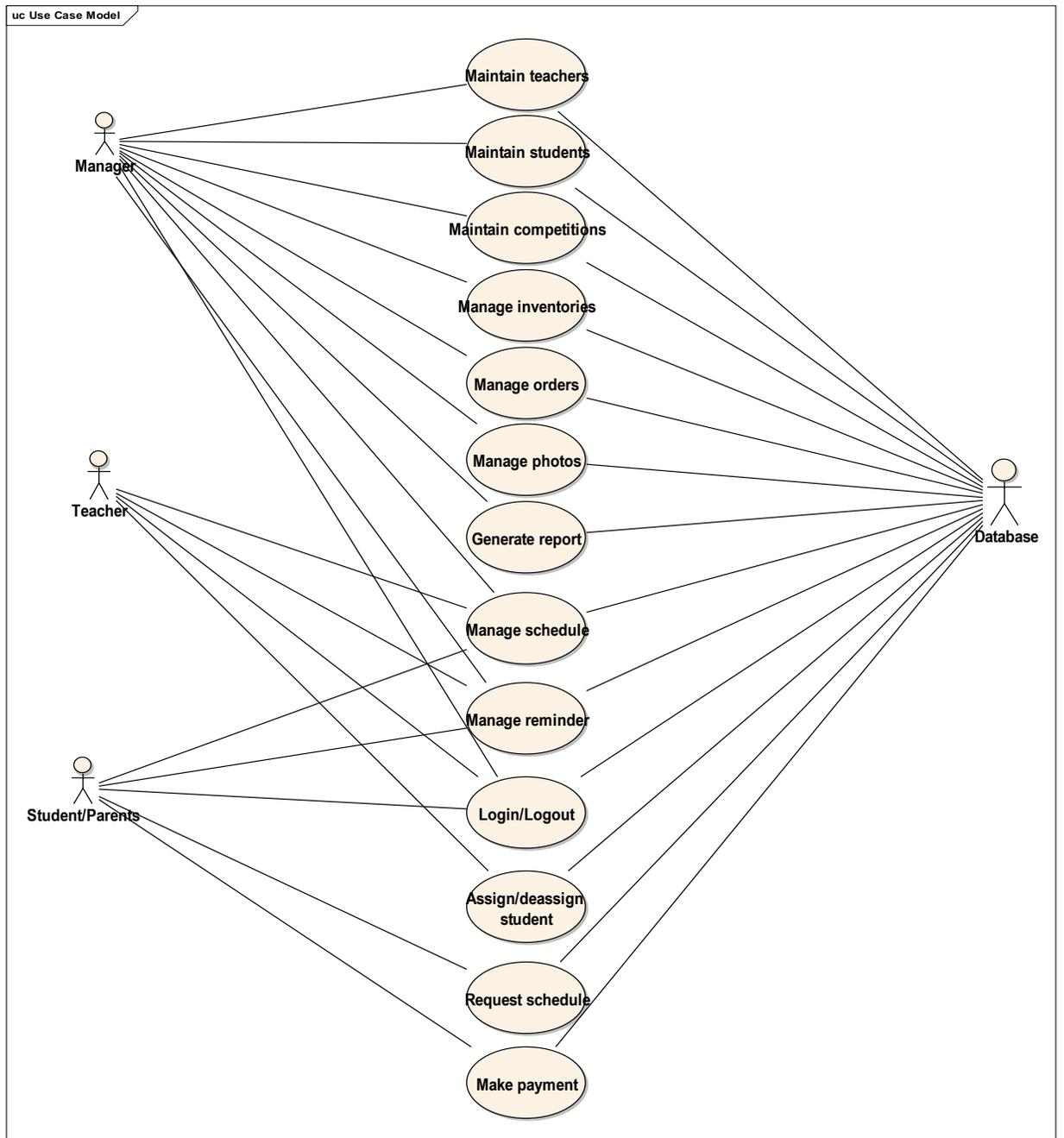


Figure 1. Use case diagram for the student management system

3.2 High level architectural design

The Student Management System was developed using the Microsoft .NET framework; the application layer is written in Visual C#. The user, from a client machine, will use his/her web browser to interact with the system. The browser will then send HTTP requests to the web server which in turn requests a service and passes parameters to the application tier. The application tier handles the request by making queries and updates to the database. The results will be passed back to the presentation tier which creates the user interface in HTML. Finally, this result is returned to the client browser.

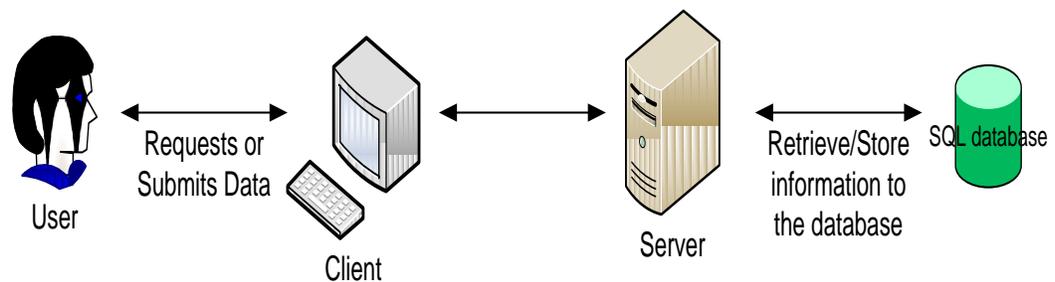


Figure 2. High level architectural design

Currently, the Student Management System is designed as a two-tiered application. The database and the web server will be on the same machine that hosts the current studio's website, and the clients are on separate machines. The main advantage of using this type of architecture is the flexibility and the separation between the logic and content of the web application. For this scope of the project, the database administrator's role is primarily setting up the database structure for the application to run. This could be done by the person who is currently maintaining the studio's website. In the future, when the studio's database is growing, it is anticipated that there will be more work for the database administrator in terms of concurrency, efficiency

and data integrity. The current version of the system does not include any concurrent features.

3.3 Detailed design architecture

As a web-based application, users have to interact with the system through web pages. Each page stands for a distinct functionality of the system. After discussing with the studio's managers and the project adviser, the functionalities were identified and grouped into two major categories: (1) core functionalities (shown in Table 2), and (2) functionalities provided by third party software (shown in table 3). Each category consists of several modules which are listed in the corresponding tables below:

No.	Module name	Functionalities
1	Account module	Allow every user in the system to perform login related functionalities: login, logout, change password, reset password, recover password.
2	User module	Allow studio's managers to manage a user's profile in the system: create new user, modify/delete/search existing users, add parent/guardian information for students.
3	Inventories module	Allow studio's managers to manage the inventory of the studio. There are three sub-modules that are included in this module which are: categories management, sub-categories management and inventories management.
4	Orders module	Allow studio's managers to manage the entire orders database in the studio: create new order, modify/delete/search existing orders, and search orders associated with existing inventories.

5	Teachers schedules module	This module allows the teacher to submit their availability schedule for teaching and this schedule can be viewed by students/parents. After the students/parents request specific time slots, it will allow the teachers to assign/de-assign the student to the time slot that he/she has requested. In this module, the teachers will also be able to prioritize students and the time slots based on the requests from students/parents.
6.	Parents/students requests module	Allow the parents/students to see teacher's teaching schedule and request time slots that they want (must be more than two so that the teachers have more options to decide the schedule in case the time slot chosen has been taken by another student).
6	Payment module	Allow a teacher to manage his/her students' payments: add/modify/search payments. This module also allows the teacher to keep track of all his/her students' balances.
7	Reminder module	This module is a small messaging subsystem. It allows users to send messages and reminders to each other depending on the roles of the user (discussed in the previous section). The senders can decide whether or not a copy of the reminder will be sent to sender's email address.
8	Report module	Allow the studio's managers to generate reports related to inventories, orders, and competitions information.
9	Images module	Allow the studio's managers to maintain images and photo albums regarding the studio. The other users (students, teachers) can only view these photos and albums.

Table 2. Core functionalities

No.	Module name	Functionalities
1	Personal schedule module	Allow every user in the system to manage their personal schedule by themselves: add/delete/modify daily events add/delete/modify event categories.

Table 3. Functionalities provided by third party software

Figure 3 shows the class diagram for the Student Management System.

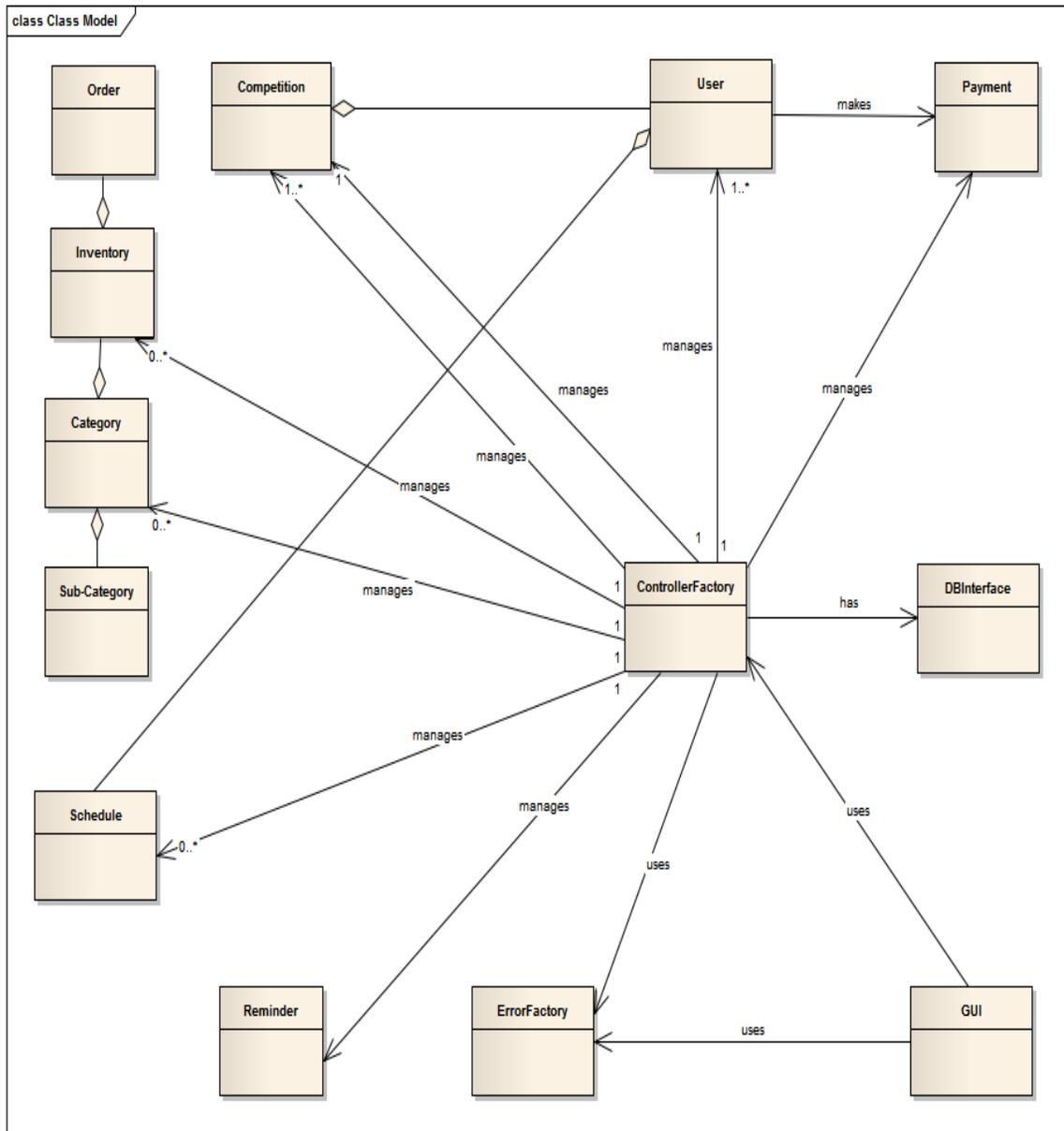


Figure 3. Class diagram for the student management system

The system was developed based on 12 main classes. Among these classes, there are 2 aggregate classes which are Category and Inventory.

- User class:

This class stores the information of the users in the system including their username, password and user type.

- Inventory class:

This class represents the inventory in the system. One inventory object can have multiple order objects associated with and belongs to one category only.

- Order class:

This class stores the order's detail information. One order object can only associate to one inventory object.

- Category class:

This class represents the category in the system. One category object can have multiple sub-category objects as well as multiple inventories associated with.

- Sub-category class:

This class represents the sub-category in the system. One sub-category object can only associate to one category.

- Payment class:

This class represents the information of the monthly payment from the parents as well as the balance of the student.

- Schedule class:

This class stores the teaching/learning schedule in the system.

- Reminder class:

This class stores the information of the reminder. It is basically the email message which has the sender, recipients, content of the reminder and date/time that the recipients need to reply.

- Competition class:
This class represents the competition or recitals conducted by the studio.
- Controller Factory class:
This class is the intermediary between the GUI and the database. It is responsible for receiving the request from a client. Once a request is received, it executes the appropriate business logic and then produces the output/result to the GUI.
- DBInterface class:
This class is to retrieve/update/insert/delete information from the database.
- ErrorFactory class:
This class is to return friendly notifications/guidelines to users if there is any errors happened during the run time.

3.4 Database design

The database design was the most complex process in this project. Since all the current data for the studio was kept in spreadsheets (mostly Microsoft Excel files), there were a lot of inconsistencies and redundancies that need to be eliminated. A lot of time was spent by the developer in checking the consistency and integrity of the data. To solve this problem, the developer used the feature of SQL Management Studio 2008 to convert the Excel documents to SQL tables. Then, by adding the relationships between tables manually as well as automatically checking, several inconsistencies in the previous data were eliminated.

Another major issue was invalid data entry in the previous studio's documents. Most errors were found to be in data corresponding to Date/Time, currency and number type. For example, the field with Date/Time type was entered in both short format and long format, the field with currency type was entered in both text and number format etc. This issue led to a problem for the conversion mentioned above. To solve this issue, the developer had to manually change invalid data to valid one.

One of the requirements was to keep track of all pending orders that the studio ordered. These pending orders will be moved to the ORDER table only when they are received by the studio. Otherwise, they cannot be inserted into this table. Moreover, the managers also want to check whether or not they received the items. Therefore, to address this problem, a table named "PENDING ORDER" was created to keep track of the pending orders. When pending orders are marked as received, they will be moved to the ORDER table. Figure 4 shows the database design for the student management system.

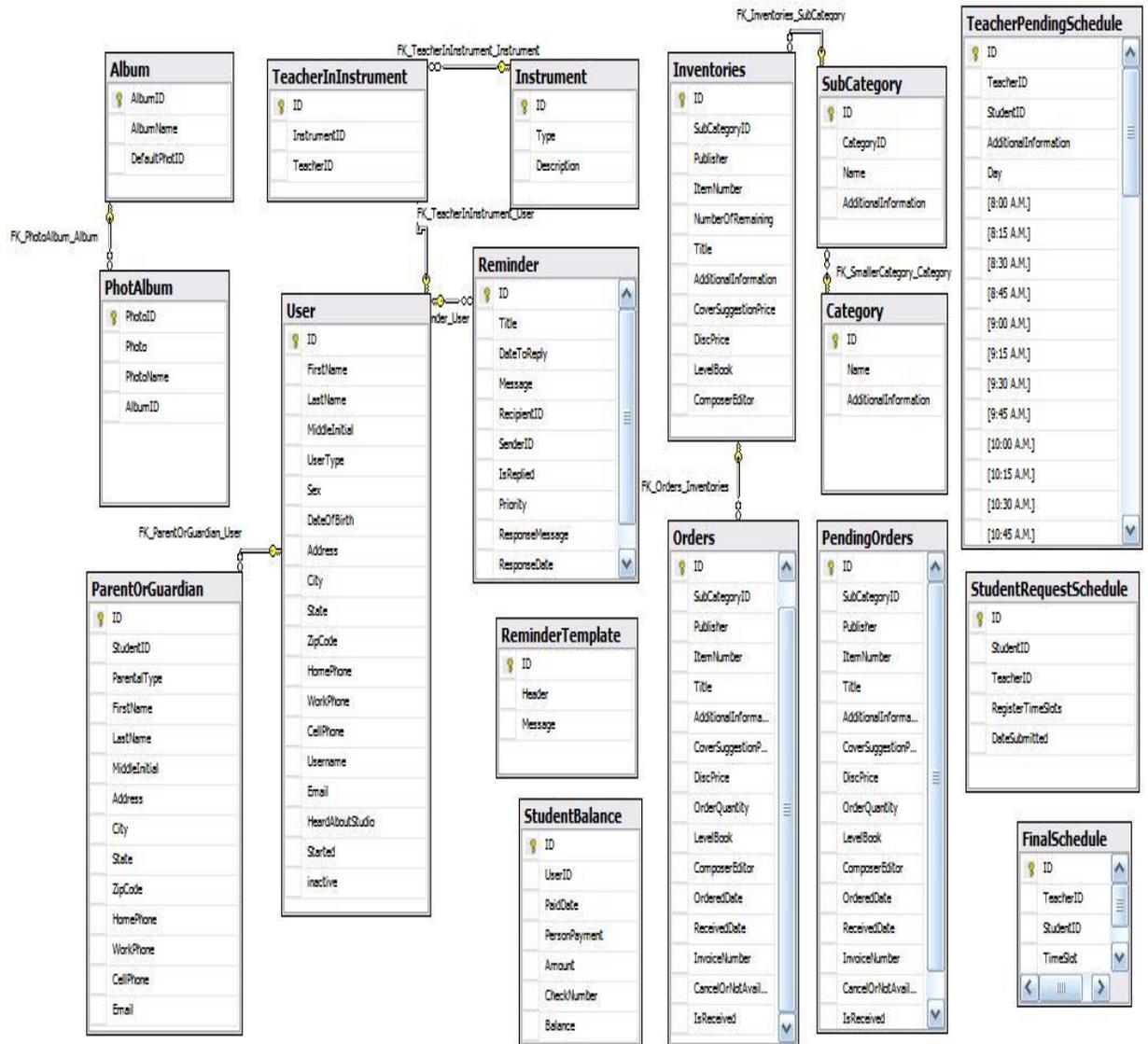


Figure 4. Database design for the student management system

The following tables describe only the most important schemas in the database which play major roles in the system:

Column name	Description
ID	Store value of record's index
Firstname	Store user's first name
Lastname	Store user's last name
MiddleInitial	Store user's middle initial
UserType	Store user type. It could be teacher, manager or student
Sex	Store user's sex
DateOfBirth	Store user's date of birth
Address	Store user's address
City	Store user's city
State	Store user's state
ZipCode	Store user's Zip Code
HomePhone	Store user's home phone
WorkPhone	Store user's work phone
CellPhone	Store user's cell phone
Username	Store user's username to login to the system
Email	Store user's email address

HeardAboutStudio	Store information how the user heard about the studio
Started	Store information when the user joined the studio
Inactive	Store user's status: active or inactive

Table 4. User schema description in chapter 4

Column name	Description
ID	Store value of record's index
Name	Store category name
Additional Information	Store additional information of the category

Table 5. Category schema description in chapter 4

Column name	Description
ID	Store value of record's index
CategoryID	Store category ID which this sub-category belongs to
Additional Information	Store additional information of the sub-category

Table 6. Sub-Category schema description in chapter 4

Column name	Description
ID	Store value of record's index
SubCategoryID	Store sub-category's id that the inventory belongs to
Publisher	Store inventory's publisher
ItemNumber	Store inventory's item number
NumberOfRemaining	Store number of the inventory remaining in the studio
Title	Store inventory's title
AdditionalInformation	Store inventory's additional information
CoverSuggestionPrice	Store inventory's cover/suggestion price
DiscPrice	Store inventory's discount price
LevelBook	Store inventory book level
ComposerEditor	Store inventory's composer/editor

Table 7. Inventory schema description in chapter 4

Column name	Description
ID	Store value of record's index
InventoryID	Store the inventoryID of which this order associates to
Publisher	Store order's publisher
ItemNumber	Store order's item number

Title	Store order's title
AdditionalInformation	Store order's additional information
CoverSuggestionPrice	Store order's cover/suggestion price
DiscPrice	Store order's discount price
OrderQuantity	Store number of order quantity
LevelBook	Store order book level
ComposerEditor	Store inventory's composer/editor
OrderedDate	Store date of order
ReceivedDate	Store date of receiving the order
InvoiceNumber	Store the invoice number of the order
CancelOrNotAvailable	Store the value indicate whether or not the order is canceled or not available
IsReceived	Store the value indicate whether or not the order is received

Table 8. Order schema description in chapter 4

Column name	Description
ID	Store value of record's index
UserID	Store parent/student's id
PaidDate	Store the date the parent/student paid the tuition

Amount	Store the amount the parent/student paid
CheckNumber	Store the check number
Balance	Store the balance of the parent/student

Table 9. Student Balance schema description in chapter 4

Column name	Description
ID	Store value of record's index
StudentID	Store the id of the parent/student who make the request
TeacherID	Store the id of the teacher being requested
RegisterTimeSlots	Store the value of the time slots being requested
DateSubmitted	Store the date that the request was made

Table 10. Student Request Schedule schema description in chapter 4

Column name	Description
ID	Store value of record's index
StudentID	Store the id of the student
TeacherID	Store the id of the teacher
Timeslot	Store the value of the scheduled time slot

Table 11. Final Schedule schema description in chapter 4

3.5 Security Design

The Student Management System is a multi-user web based application which stores and processes some confidential information such as personal information, financial information of the studio etc. Therefore, security was one of the major concerns throughout the development process of the project.

All authorization and authentication processes were automatically supported and done by Microsoft .NET framework. Each user has his/her own username and password, and they can only perform a specific set of functionalities based on their user roles. For example, student and teacher users have limited privileges and so would not be able to view the financial information of the studio such as inventories/orders information.

Another feature that provided to support security is that the user's password is only stored in the database and is encrypted. This kind of encryption will ensure that even the database administrator cannot guess or retrieve the password because it is a one way encryption. The user can access the system only if he/she enters a password and this password is hashed and matched to the one stored in the database.

During the design phase, the developer took care of data validation and verification on both client-side and server-side of the application. It is really apparent that client-side validation only is not secure, because some malicious users can see the Java Script code and change the script to try different values until something unauthorized is revealed. The developer has combined both Java Script and the client-side validation supported by Microsoft .NET framework to verify the data input by the users. Users are also provided tooltips for every input field and so they have the guidance when they do not know whether or not the data is valid. The information from the client side will be passed to the application layer and then be checked one more time to verify the correctness. These steps minimize the risks to the system created by the users.

3.6 User Interface Design

The Student Management System contains a web-based graphical user interface, also known in some literature as a WUI (Web User Interface). This interface consists of many ASP.NET web pages through which the users interact. ASP.NET page is basically a HTML page with associated code that is to be processed by the server before it is returned to the client. The code is specified in C#. Microsoft Visual Studio 2008 takes care of associating the code with the associated HTML used for the actual presentation layer. Code may also be embedded directly into HTML. The code-behind methodology was selected by the developer instead, since it allows for better encapsulation, easier maintenance, and better readability at the price of slow performance.

Regardless of the coding method used for web pages, all ASP.NET code is processed on the web server and is replaced with HTML before the response is sent back to the requesting client. This allows multiple browsers such as Internet Explorer, Mozilla Firefox, Google Chrome, and Safari etc. interact with the system, even though the client browsers may be on non-Windows platforms or older versions of the Windows operating system.

3.7 Online help system

The Student Management System provides the most useful features for the studio's managers as well as to the students and teachers so that they can operate more efficiently. However, most of the users in the system have little experience using computers and knowledge about technology. Therefore, producing invalid data inputs or having troubles to operate a function is unavoidable. To support the users, besides the tooltips provided for every data fields, the developer has integrated an online help system inside the web application.

The online help system is a sub-system which contains HTML web pages. It consists two parts:

- The first part contains all common topics that users may want to see or know when they access the system. Those topics help users manage user, schedule, report, reminder, photo, payment, order and inventory. Each help page will show which type of user can perform the operation and guide the user step by step to successfully finish the operation. This help page also lists all valid inputs for every data field so the users will have the document to reference when they do not know how to enter the data.
- The second part is the search feature which allows users to enter any keyword and the system will return the topic containing that keyword. The search feature provides another convenient way so the users can quickly find the topic they want.

3.8 Testing

Testing was started right away once the requirement document was completed and after getting feedback on the rapid prototype. Review and inspection of the requirement document was done in several iterations and by the end of the requirement phase there were about 10 defects found. After those defects were fixed, the developer kept the requirement document fixed and the incremental software development life cycle was followed to engineer the project. On completion of each module, unit testing as well as integration testing were performed.

After the entire software was developed, the system testing was thoroughly conducted both by the developer and the studio's users for about a month (from the middle of Nov 2009 to middle of Dec 2009). Currently, the product is left with the customer for beta testing. If they find any issues, they will contact the developer. It is agreed that the developer will maintain the system through offline contacts for some time until the product is stable.

3.9 Deployment

Since the Reed Music Studio already has the website which has the advertisement information of the studio, the managers want to integrate the new system into their current web host. Therefore, the developer divided the deployment into three phases:

- In the first phase, the developer deployed the system on his own machine. The database server was SQL Server Express and the web application was compiled and run directly on a local host. There were no problems reported in this phase.
- In the second phase, the developer started to deploy the system on his own web server. The web server is running Microsoft SQL Server 2005 and allows the .NET framework 3.5 to be deployed. This was the first

time the developer deployed a .NET web application to an unfamiliar web hosting service, and so there were a lot of troubles encountered. Mostly it was due to the restriction of the web hosting company to the deployed assemblies (a combination of single file or multiple files in ASP.NET). They only allow the assemblies with the Medium Trust level to be deployed. However, in the system, there is more than one important assembly that needs the Full/High Trust level in order to work correctly. For example, the assembly for Microsoft Chart Control or the one for Scheduler control is one of them. It took about two weeks for the developer to discuss and negotiate with the web hosting company so they could increase the trust level for the project. The other troubles were mostly due to the lack of the developer's experience on hosting web-based applications.

- In the last phase, after the project works stably on the developer's web server, the developer will give all the source code as well as the documents to the person who is currently maintaining the studio's website. He/she will follow the developer's guidance to integrate the new web application the studio's current web hosting. This phase will take a lot of time to communicate technical issues back and forth and is not done yet.

4. Limitations

Development of the Student Management System is the first attempt to automate the entire managing workflow inside the studio as well as to eliminate a series of paper work needed and reduce manual work time for studio's managers. While the system has many capabilities and advantages for the studio, it still has some limitations:

- The GUI is easy and simple to use, but it is not quite attractive to the users.
- After every school year, the managers have to delete the teaching/learning schedule of the previous year manually.
- The user's ability to create his/her configurable profile is not supported in this version. For example, a user may want to change the current font in the application to his/her favorite, or change the layout and appearance of the web pages.
- Reports generated by the system are not printable. If the managers want to print, they must use the print feature integrated in the web browsers.

5. Continuing Work

Since the system still has some limitation listed on section 5, there are still few requirements and enhancements need to be implemented in future versions:

- Generate reports in printable formats.
- Modify the system to be a portal so that the users can easily change the appearance of the web application.
- Automatically delete the teaching/learning schedule after every school year, and then send emails to the teachers as well as the students to renew the schedule.

Further modifications to the user interface may also be performed to achieve the desired look and feel which satisfies most of the users in the system.

6. Conclusion

The Student Management System for Reed Music Studio is a web application which is capable of storing and maintaining different types of user accounts, keeping track of the inventories and orders database, managing the teaching and learning schedules as well as the payments made by students. It also allows studio's managers to generate reports about all the information related to inventories and orders database.

The system was designed to capture all vital information needed to improve the management of workflow inside the studio. The system helps the managers reduce most of the manual steps and organize data more efficiently. Many of the data entry tasks are automated and save a significant amount of time for users. It also provides a consistent interface to the application data that is enforced by referential integrity constraints specified in the database.

In conclusion, the Student Management System for Reed Music Studio promises to be a fast, easy, cost efficient and effective tool to address most of the studio's issues as well as being a really good alternative to replace the current manual workflow in the studio.

BIBLIOGRAPHY

1. A.M. Davis, *Software Requirements Analysis and Specification*, Prentice Hall, 1990.
2. J. Duckett, *Beginning Web Programming with HTML, XHTML, and CSS (Wrox Programmer to Programmer)*, Wrox, 2008.
3. M.J. Hernandez, *Database Design for Mere Mortals(R): A Hands-On Guide to Relational Database Design*, Addison Wesley Professional, 2008.
4. P. Kimmel, *LINQ Unleashed: for C#*, Sams, 2008.
5. I. Sommerville, *Software Engineering 6th Ed.*, Addison Wesley, 2001.
6. A. Troelsen, *Pro C# 2008 and the .NET 3.5 Platform*, Apress, 2008.
7. S. Walther, *ASP.NET 3.5 Unleashed*, Sams, 2008.
8. <http://msdn.microsoft.com/en-us/library/h8w79z10.aspx>, *Application Development in Visual Studio*, Microsoft Corporation, 2007.
9. [http://msdn.microsoft.com/en-us/library/wxt2cwcc\(VS.80\).aspx](http://msdn.microsoft.com/en-us/library/wxt2cwcc(VS.80).aspx), *Connecting to Data in Visual Studio Overview*, Microsoft Corporation, 2007.

APPENDIX A: Selected screen shots

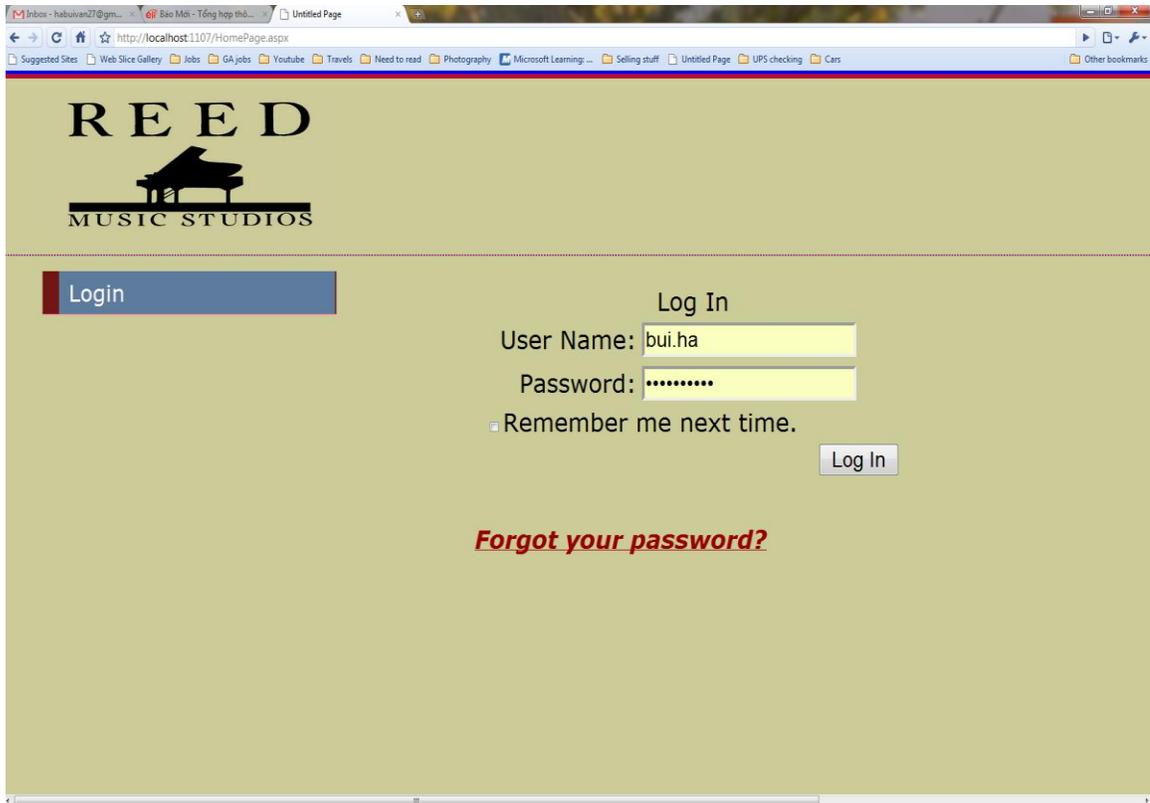


Figure 5. Login screen

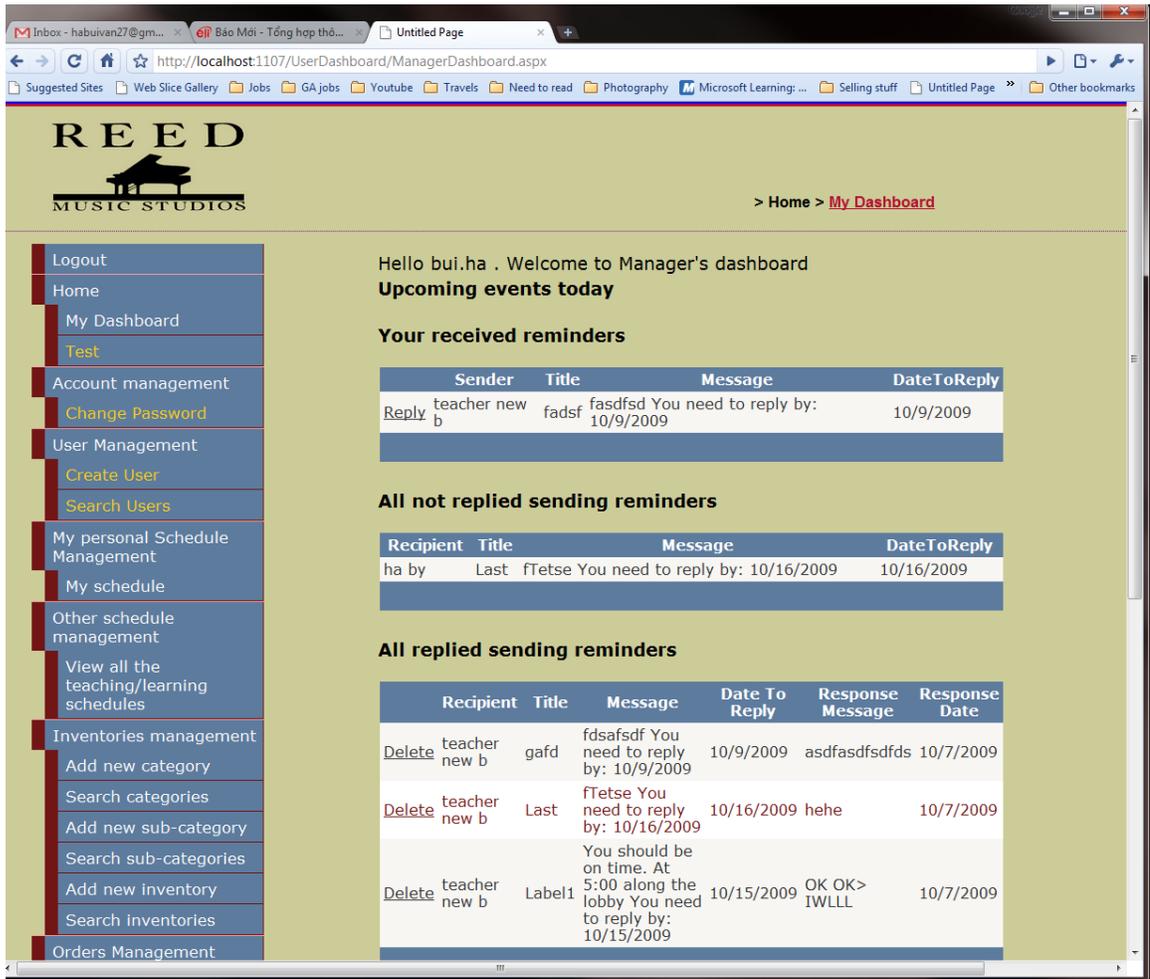


Figure 6. User dashboard

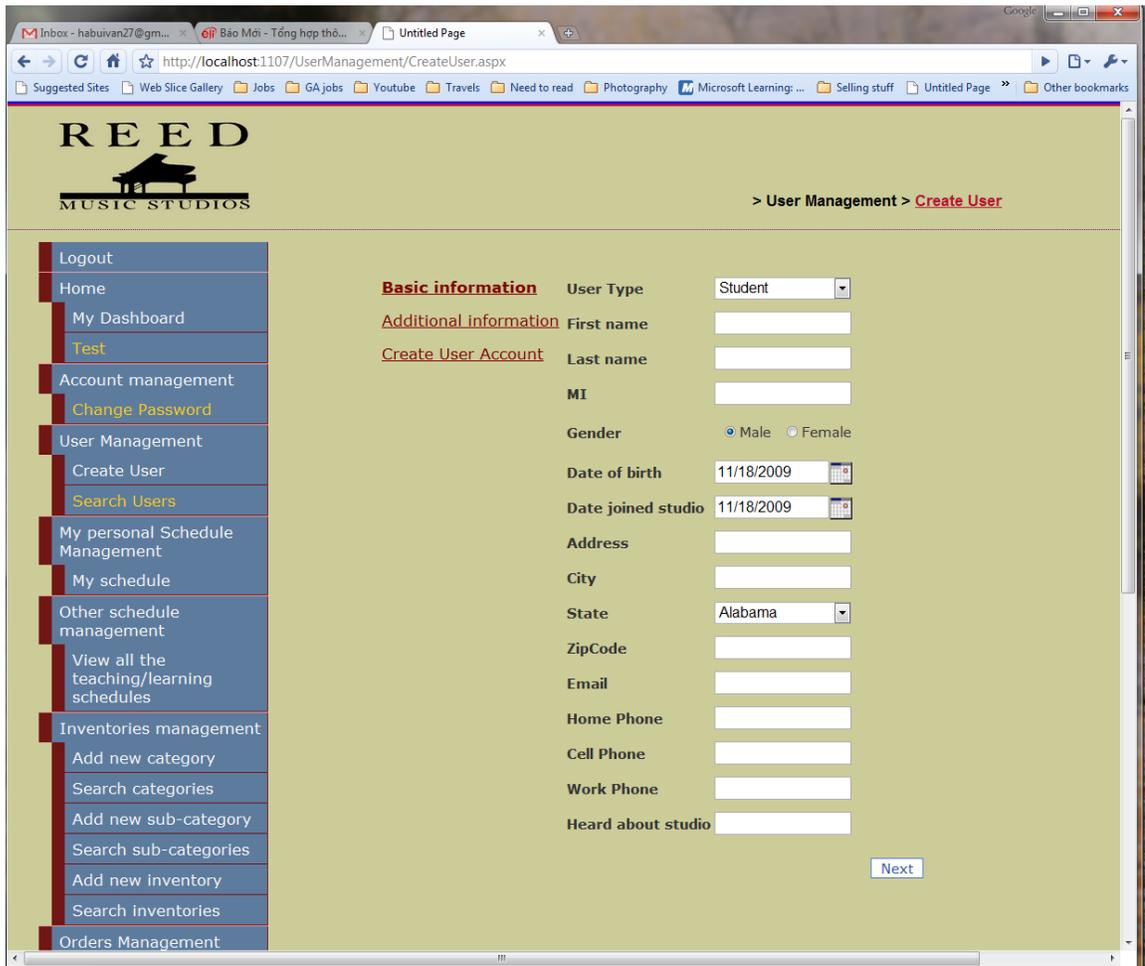


Figure 7. Create user wizard

Search user

Search for keywords:

Search by user type:

Search by name:

Search by city:

Search by state:

Search continue/discontinue students: Continue Discontinue

Lastname	FirstName	M.I	User Type	Sex	Date Of Birth	Address	City	State	Zip Code	Home Phone	Work Phone	Cell Phone	Username	Email	Heard about student	Started	Inactive
Edit	Delete	Parents info	Clair s	Alex	Student	Male	10/29/1996	128 South 200 St.	Le Crosse	WI	54601-796-1901	792-9746	clclair	clclair@charter.net	Camie's student	4/4/2009	<input type="checkbox"/>
Edit	Delete	Parents info	Sjahi	Hannah s	Student	Male	2/8/1996	N3305 CTH M. lot 11.6	West Salem	WI	54669-786-3174	791-498-6537	stie.hann	pumkinthekitty@yahoo.com	Camie's student	4/4/2009	<input type="checkbox"/>
Edit	Delete	Parents info	Devine s	Julia	Student	Male	1/10/1901	W7421 Cty Hwy Z	Onalaska	WI	54650-781-7792		devi.juli			2/1/2005	<input type="checkbox"/>
Edit	Delete	Parents info	Rabihah	Brianna	Student	Male	7/31/2002	W6990 Reef Road	Onalaska	WI	54650-519-0950	796-3714-9484	rebh.bria	vbrebhah@charter.net	Aredia naive and Haureen	1/26/2009	<input type="checkbox"/>
Edit	Delete	Parents info	Simmons	Grant	Student	Male	1/10/1901	5222 Bradenwood Court	Le Crosse	WI	54601-781-3698	775-2102-8551	simm.gran	lonnimm@mac.com		1/1/2006	<input type="checkbox"/>
Edit	Delete	Parents info	Brockman	Patrick	Student	Male	1/10/1901	N5550 Moos Road	Onalaska	WI	54650-779-5142	782-2760-0113	bmc.pat	patrickbrockman@charter.net		1/1/1900	<input type="checkbox"/>
Edit	Delete	Parents info	Dummer	Isaac	Student	Male	1/10/1901	N6669 CTH XX	Holmen	WI	54634-526-4384	781-386-2710	dumm.isaa		previous student	4/7/2009	<input type="checkbox"/>
Edit	Delete	Parents info	Hess	Julia	Student	Male	1/10/1901	1895 Wood Run Place	Onalaska	WI	54650-779-9370	385-2630	hess.juli	tmishael@centurytel.net		1/1/1900	<input type="checkbox"/>
Edit	Delete	Parents info	Iyer	Jai	Student	Male	4/25/1991	518 Stonebridge Ave.	Onalaska	WI	54650-781-3786		iyer.jai	mayura_iyer@hotmail.com	Friend	7/1/2002	<input type="checkbox"/>
Edit	Delete	Parents info	Jumper	Corbin	Student	Male	12/22/1996	113727 Schubert Road	Trempealeau	WI	54661-608-7463	715-985-2337	jump.corb	ddjumper@yahoo.com	Drive by	1/1/2009	<input type="checkbox"/>

1 2 3 4 5 6 7 8 9 10 ...

Figure 8. Search users

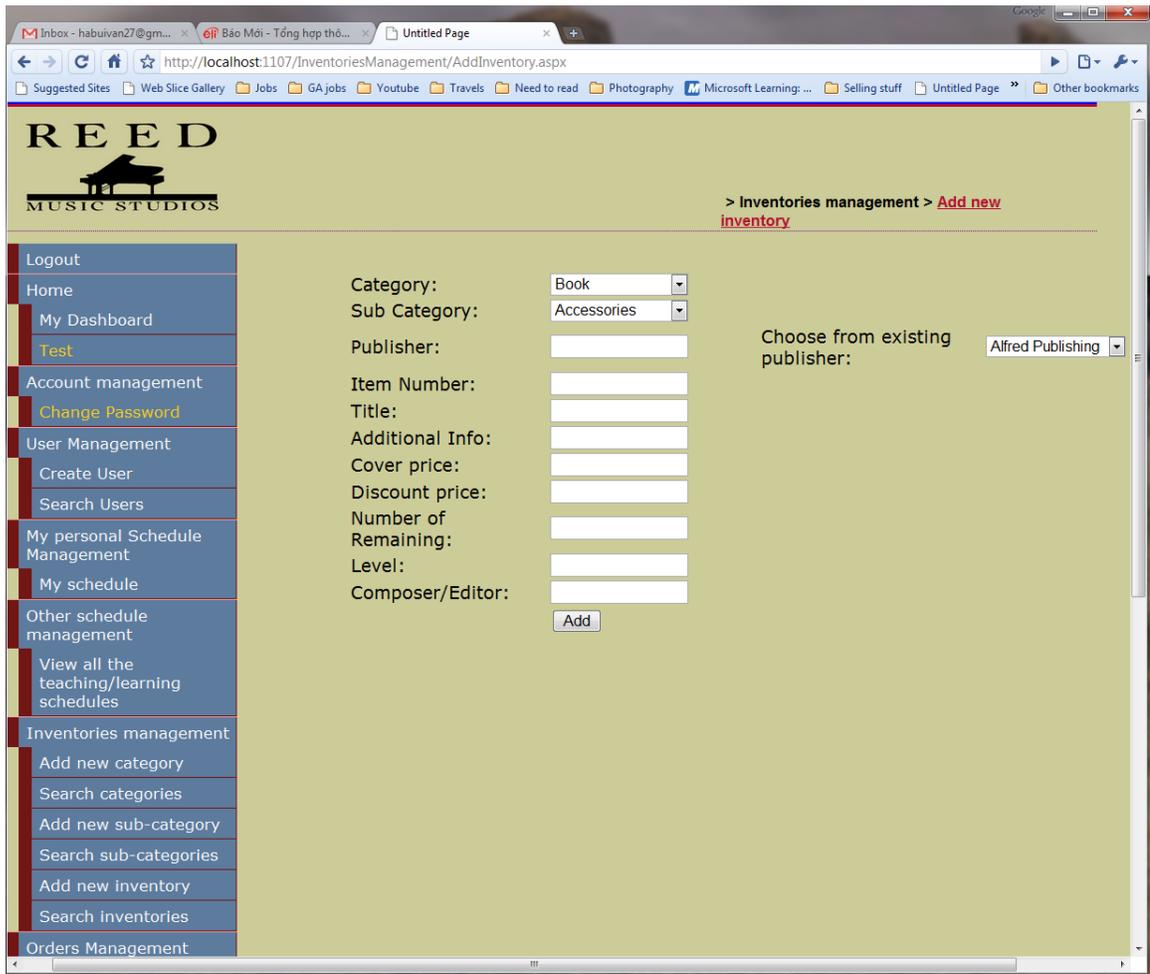


Figure 9. Add inventory

REED MUSIC STUDIOS

> Inventories management > [Search inventories](#)

Logout

Home

My Dashboard

Test

Account management

Change Password

User Management

Create User

Search Users

My personal Schedule Management

My schedule

Other schedule management

View all the teaching/learning schedules

Inventories management

Add new category

Search categories

Add new sub-category

Search sub-categories

Add new inventory

Search inventories

Orders Management

Add new Order

Search orders

Competition

Add participant

Update participants

Add competition

Report

Competition Report

Inventory Report

Cost Report

Reminder

Create new Reminder

Search Inventory

Category: All Specific

Book

Sub Category: All Specific

Accessories

Publisher:

Item Number:

Title:

Level:

Composer/Editor:

Number of Remaining:

	Publisher	Item Number	No. Of remaining	Title	Additional Info.	Cover/Suggestion Price	Discount Price	Level	Composer/Editor
Edit Delete Associated orders	Alfred Publishing	16636	2	Essential Dictionary of music	Pocket size reference	5.95	3.57		Alfred
Edit Delete Associated orders	Alfred Publishing	22531	1	Celebrated Piano Duets	Piano Duets	6.95	4.17	1	VANDALL, R
Edit Delete Associated orders	Alfred Publishing	22532	0	Celebrated Piano Duets	Piano Duets	7.5	4.5	2	VANDALL, R
Edit Delete Associated orders	Alfred Publishing	22533	0	Celebrated Piano Duets	Piano Duets	7.95	4.77	3	VANDALL, R
Edit Delete Associated orders	Alfred Publishing	24551	1	Celebrated Piano Duets	Piano Duets	7.95	4.77	4	VANDALL, R
Edit Delete Associated orders	Alfred Publishing	24552	2	Celebrated Piano Duets	Piano Duets	7.95	4.77	5	VANDALL, R
Edit Delete Associated orders	Alfred Publishing	6049	0	HYMNS WITH STYLE	SUPPLEMENTAL	6.95	4.17		JERRY RAY
Edit Delete Associated orders	Alfred Publishing	23255	0	Play Praise Worship songs		8.95	5.37	Early Int	Gerou and Labenske
Edit Delete Associated orders	Alfred Publishing	23253	0	Play Praise Worship songs		8.95	5.37	Elem	Gerou and Labenske
Edit Delete Associated orders	Alfred Publishing	23254	0	Play Praise Worship songs		8.95	5.37	LATE ELEM	Gerou and Labenske

1 2 3 4 5 6 7 8 9 10 ...

Figure 10. Search inventory

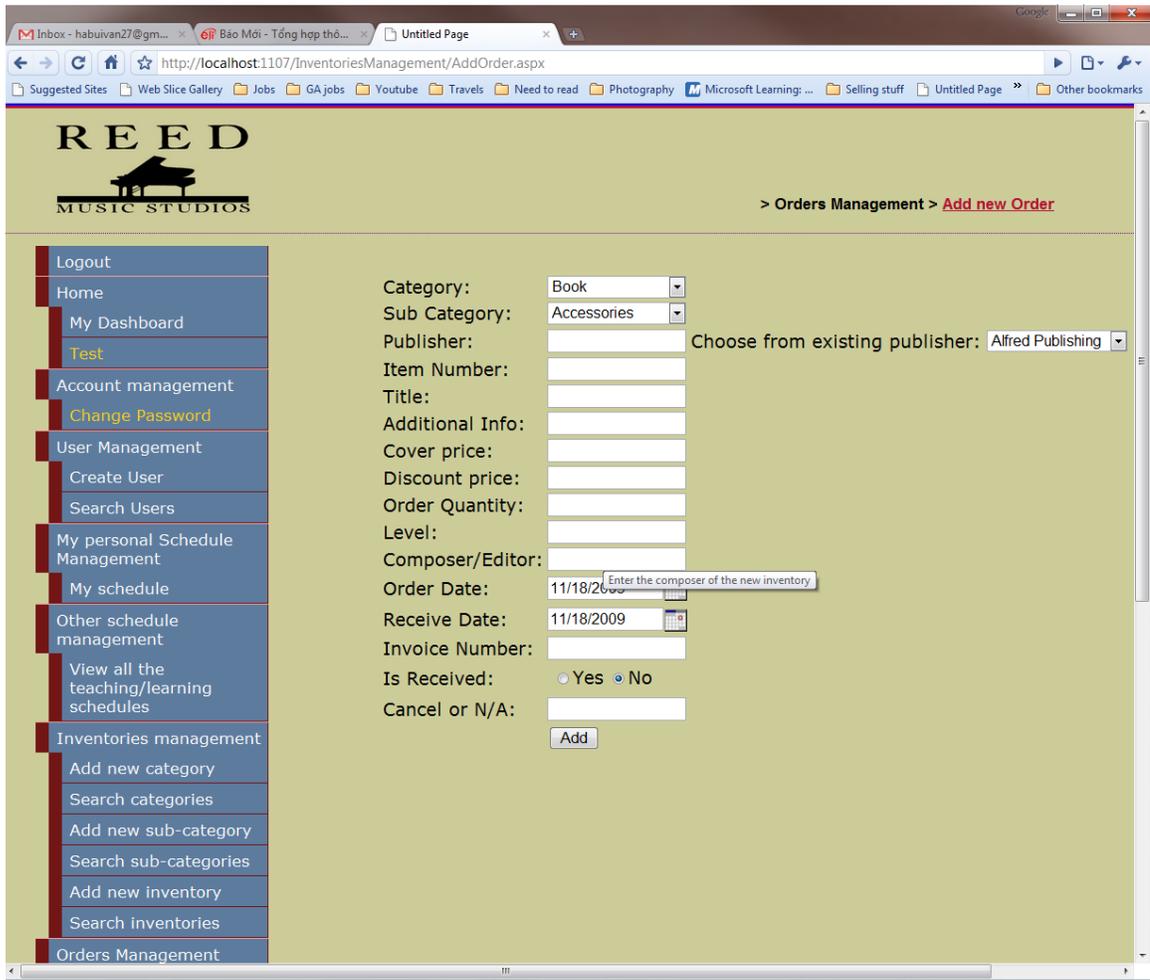


Figure 11. Add order

REED MUSIC STUDIOS

Logout
Home
My Dashboard
Test
Account management
Change Password
User Management
Create User
Search Users
My personal Schedule Management
My schedule
Other schedule management
View all the teaching/learning schedules
Inventories management
Add new category
Search categories
Add new sub-category
Search sub-categories
Add new inventory
Search inventories
Orders Management
Add new Order
Search orders
Competition
Add participant
Update participants
Add competition
Report
Competition Report
Inventory Report
Cost Report
Reminder
Create new Reminder
Create reminder template
Photo albums
View all albums
Create new album
Image Upload
Help Section

Orders Management > Search orders

Search Received Order

Category: All Specific
Book

Sub Category: All Specific
Accessories

Publisher: _____
Item Number: _____
Title: _____
Level: _____
Composer/Editor: _____

Start Order Date: 11/18/2009 End Order Date: 11/18/2009
Start Receive Date: 11/18/2009 End Receive Date: 11/18/2009
Invoice Number: _____
Cancel or N/A: Is Canceled or N/A?
Search

	Publisher	Item Number	Title	Additional Info.	Cover/Suggestion Price	Discount Price	Order Quantity	Level	Composer/Editor	Ordered Date	Received Date	Invoice	Cancel or N/A	Is Received
Edit Delete	Alfred Publishing	2482C	NOCTURNES COMPLETE		0	0	0	ADV	CHOPIN, ED. PALMER	6/25/2002	6/27/2002	2476092		True
Edit Delete	Alfred Publishing	19-29009119	BEST LOVED CLASSICAL MELODIES		0	0	0	BIG NOTE	MIXED	6/25/2002	6/27/2002	2476092		True
Edit Delete	Alfred Publishing	16611	BEST OF MARTHA MIER LATE ELEM/E. INT		0	0	0	2	MIER, M	6/25/2002	6/27/2002	2476092		True
Edit Delete	Alfred Publishing	16612	BEST OF MARTHA MIER INT		0	0	0	3	MIER, M	6/25/2002	6/27/2002	2476092		True
Edit Delete	Alfred Publishing	18099	BEST OF CATHERINE ROLLIN E. INT/ INT		0	0	0	1	ROLLIN, C	6/25/2002	6/27/2002	2476092		True
Edit Delete	Alfred Publishing	18100	BEST OF CATHERINE ROLLIN INT/ L. INT		0	0	0	2	ROLLIN, C	6/25/2002	6/27/2002	2476092		True
Edit Delete	Alfred Publishing	6643	JAZZ RAGS, BLUES E. INT/ INT		0	0	0	2	MIER, M	6/25/2002	6/27/2002	2476092		True
Edit Delete	Alfred Publishing	16871	JAZZ RAGS, BLUES INT/ L. INT		0	0	0	3	MIER, M	6/25/2002	6/27/2002	2476092		True
Edit Delete	Alfred Publishing	6642	JAZZ RAGS, BLUES LATE ELEM/E. INT		0	0	0	1	MIER, M	6/25/2002	6/27/2002	2476092		True
Edit Delete	Alfred Publishing	ELM02024	JAZZ MONTAGE INT		0	0	0	4	MINSKY, L	6/25/2002	6/27/2002	2476092		True

1 2 3 4 5 6 7 8 9 10 ...

Figure 12. Search order

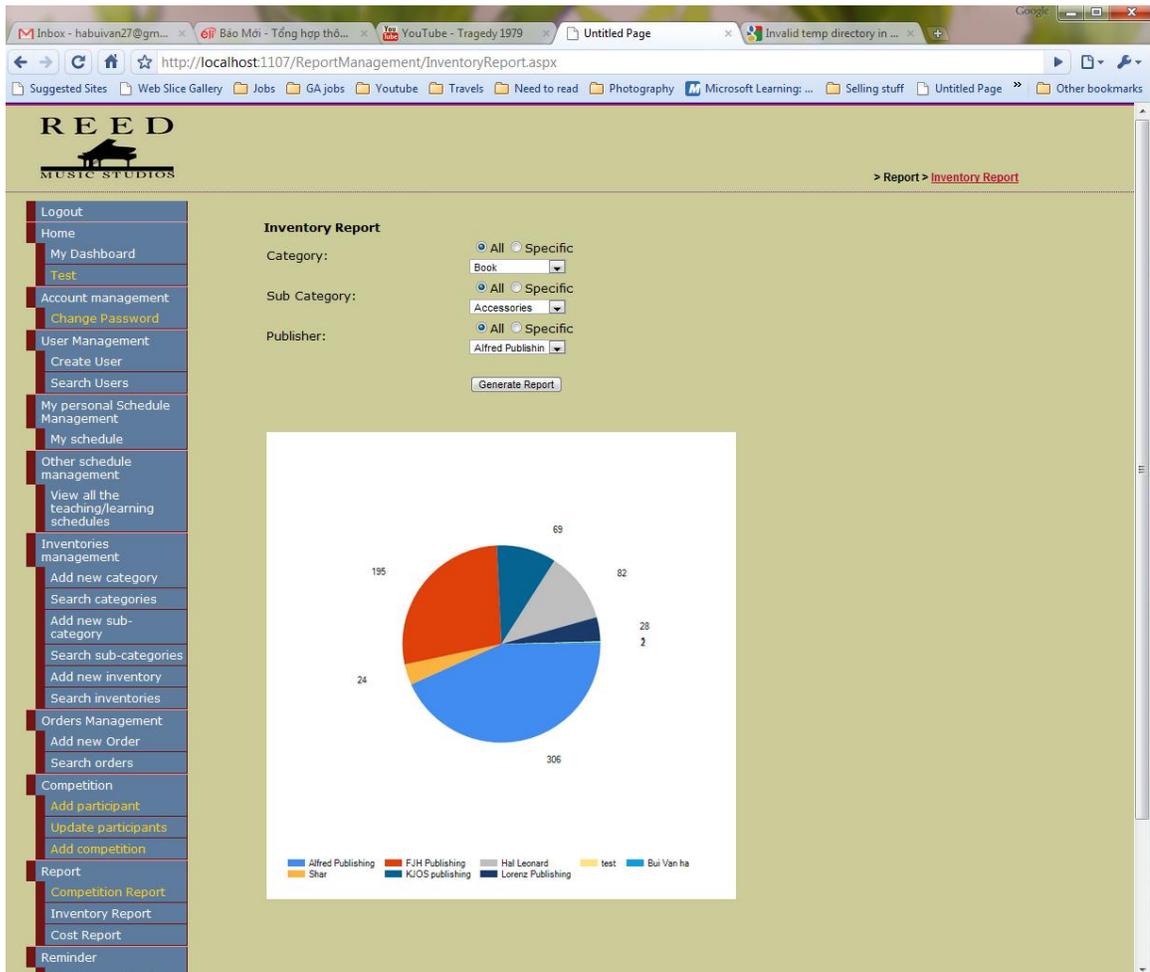


Figure 13. Inventory report

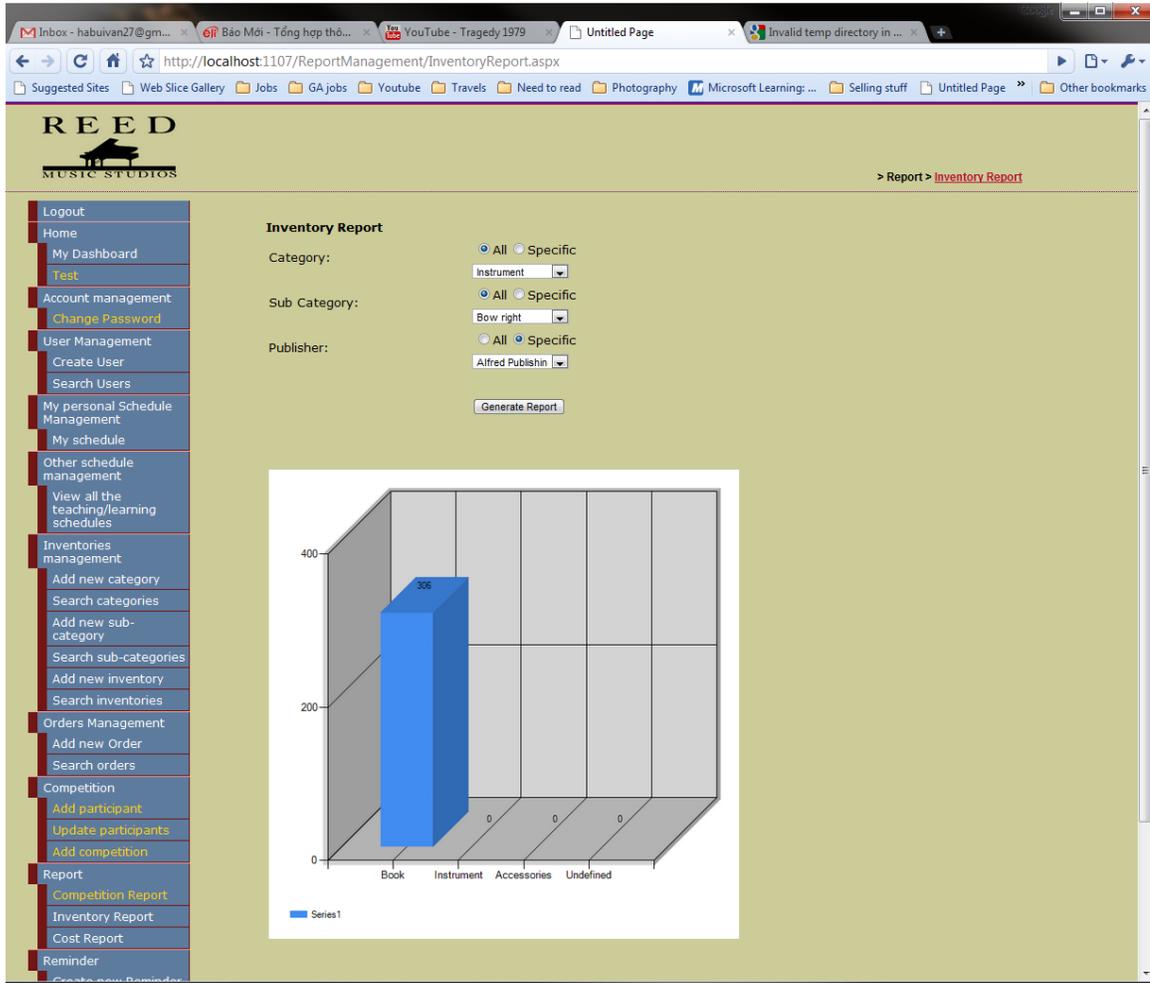


Figure 14. Inventory report (2)

The screenshot displays a web browser window with the URL `http://localhost:1107/PersonalScheduleManagement/MySchedule.aspx`. The page header includes the logo for "REED MUSIC STUDIOS" and a breadcrumb trail: `> My personal Schedule Management > My schedule`.

On the left side, there is a vertical navigation menu with the following items:

- Logout
- Home
- My Dashboard
- Test
- Account management
 - Change Password
- User Management
 - Create User
 - Search Users
- My personal Schedule Management
 - My schedule
- Other schedule management
 - View all the teaching/learning schedules
- Inventories management
 - Add new category
 - Search categories
 - Add new sub-category
 - Search sub-categories
 - Add new inventory
 - Search inventories
- Orders Management
 - Add new Order
 - Search orders
- Competition
 - Add participant
 - Update participants
 - Add competition

The main content area shows a calendar for the week of Nov 16, 2009, to Nov 22, 2009. The calendar grid has columns for each day and rows for hourly slots from 0:00 AM to 9:00 AM. Events are represented by colored blocks:

- Monday, Nov 16:** 0:30am (blue, subject: adf), 4:00am (purple, subject: 21), 7:00am (green, subject: No Subject).
- Tuesday, Nov 17:** 2:00am (blue, subject: No Subject), 5:30am (green, subject: No Subject).
- Thursday, Nov 19:** 1:00am (orange, subject: fdacf).
- Friday, Nov 20:** 4:00am (green, subject: No Subject).
- Saturday, Nov 21:** 1:30am (green, subject: fads).
- Sunday, Nov 22:** 0:30am (purple, subject: ba), 7:00am (orange, subject: No Subject).

Below the calendar grid, a message box states: "Your event was created."

Figure 15. Personal schedule

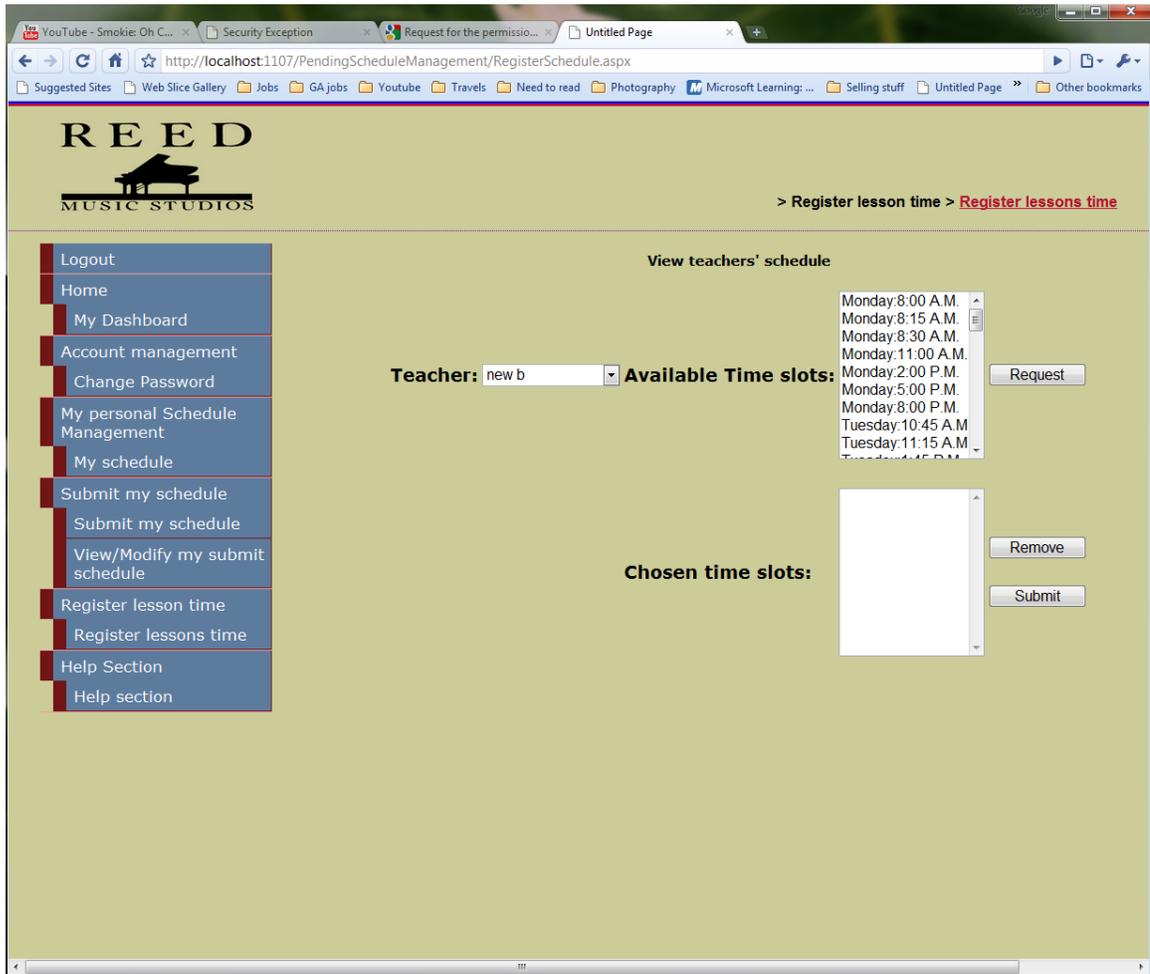


Figure 16. Request schedule screen

The screenshot displays a web browser window with the URL `http://localhost:1107/PersonalScheduleManagement/ViewPendingSchedule.aspx`. The page header features the logo for REED MUSIC STUDIOS and navigation links: [Submit my schedule](#), [View/Modify my submit schedule](#), and [submit schedule](#).

The main content area is titled "View teachers' schedule" and includes a "Modify" button. Below this is a grid representing the teacher's available schedule. The grid has columns for days of the week (Monday through Sunday) and rows for time slots from 8:00 A.M. to 1:15 P.M. in 15-minute increments. Checkmarks in the grid indicate available time slots.

Day	8:00 A.M.	8:15 A.M.	8:30 A.M.	8:45 A.M.	9:00 A.M.	9:15 A.M.	9:30 A.M.	9:45 A.M.	10:00 A.M.	10:15 A.M.	10:30 A.M.	10:45 A.M.	11:00 A.M.	11:15 A.M.	11:30 A.M.	11:45 A.M.	12:00 P.M.	12:15 P.M.	12:30 P.M.	12:45 P.M.	1:00 P.M.	1:15 P.M.
Monday	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Tuesday	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Wednesday	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Thursday	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Friday	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Saturday	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Sunday	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>										

Figure 17. Teacher's available schedule

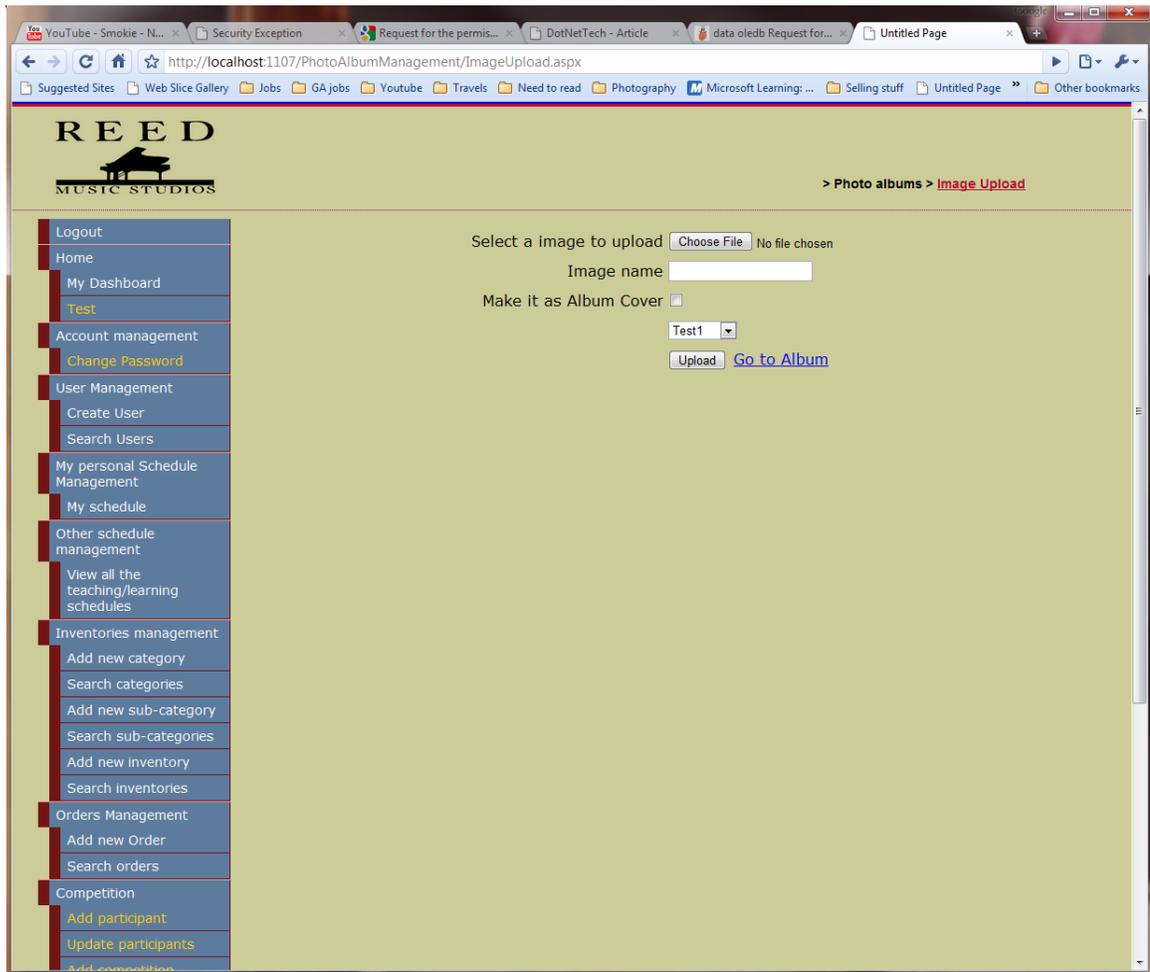


Figure 18. Create album and upload image

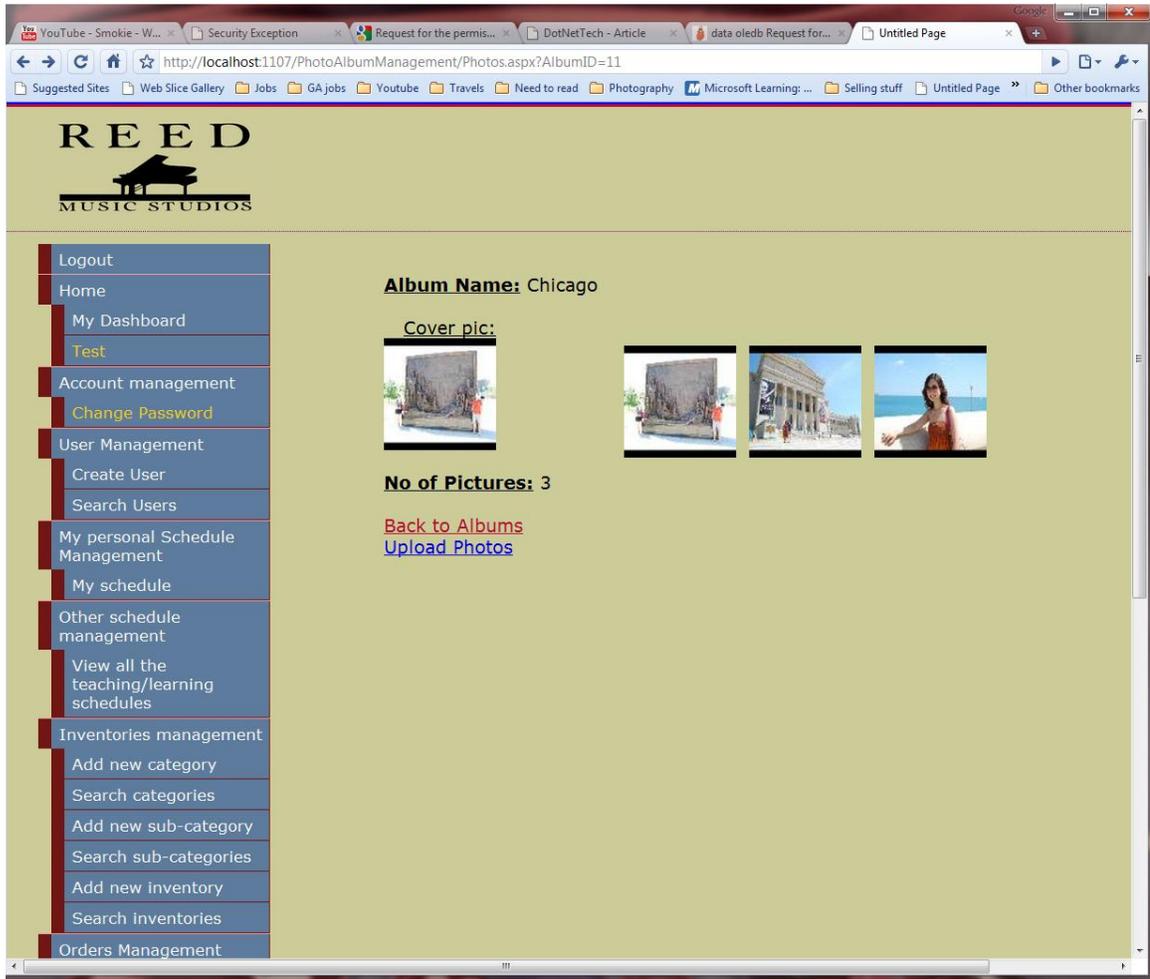


Figure 19. View album

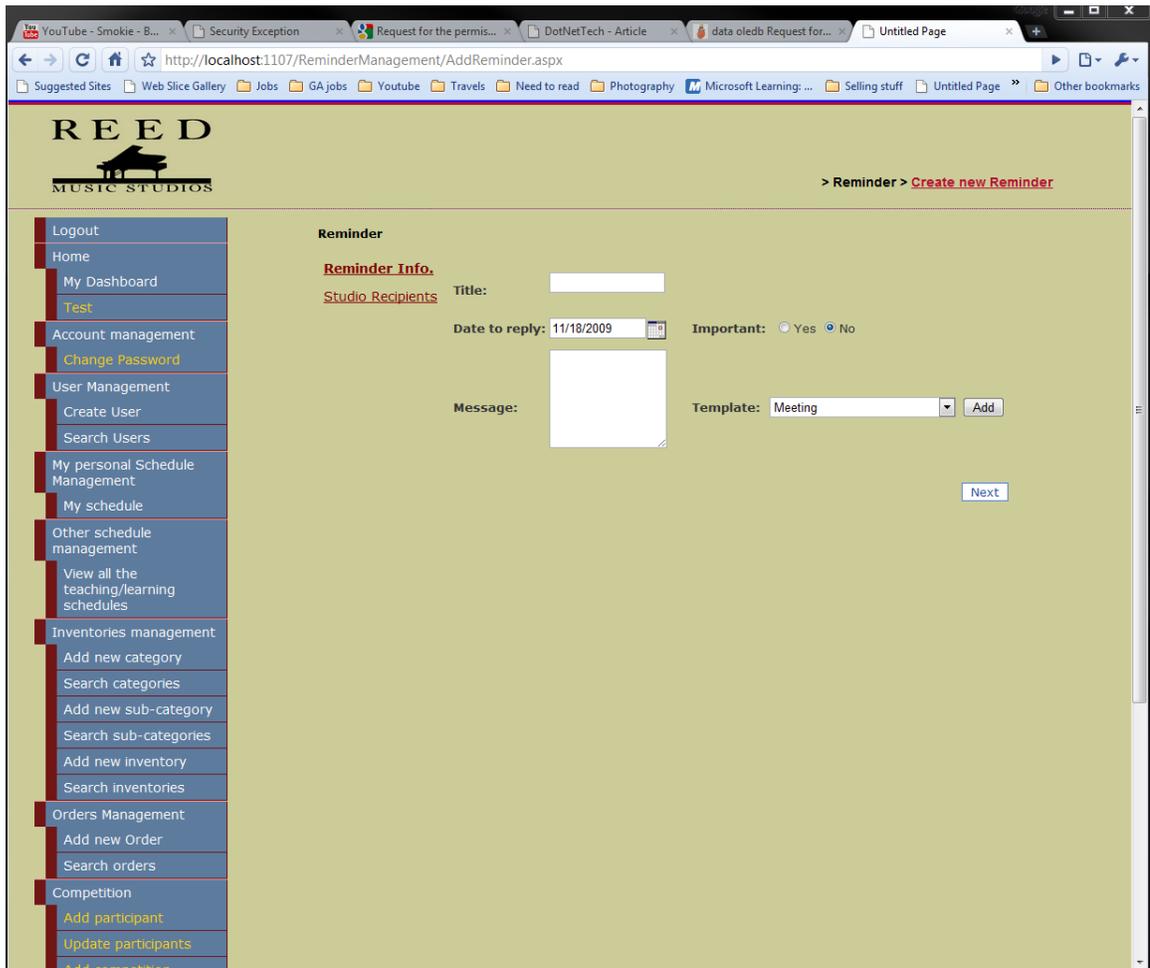


Figure 20. Send a reminder

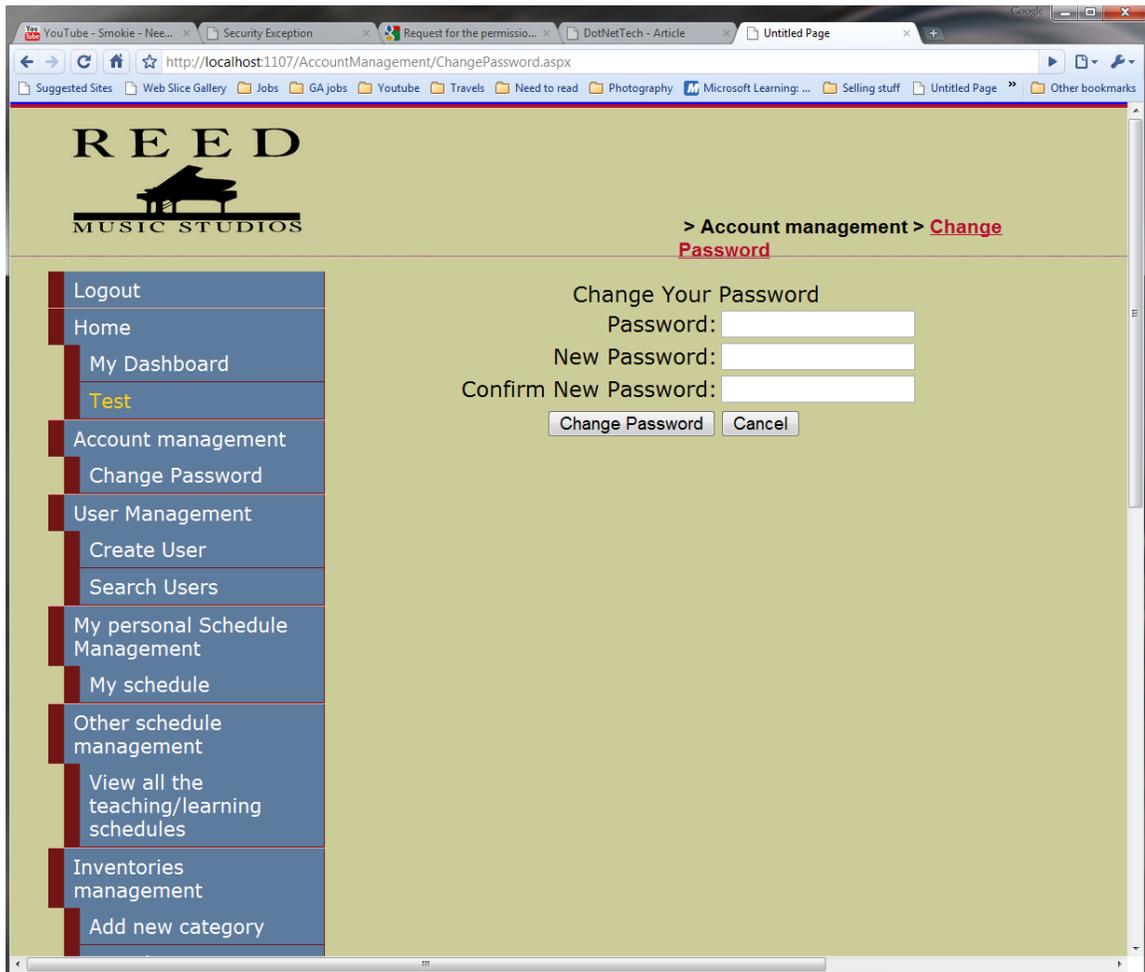


Figure 21. Change password

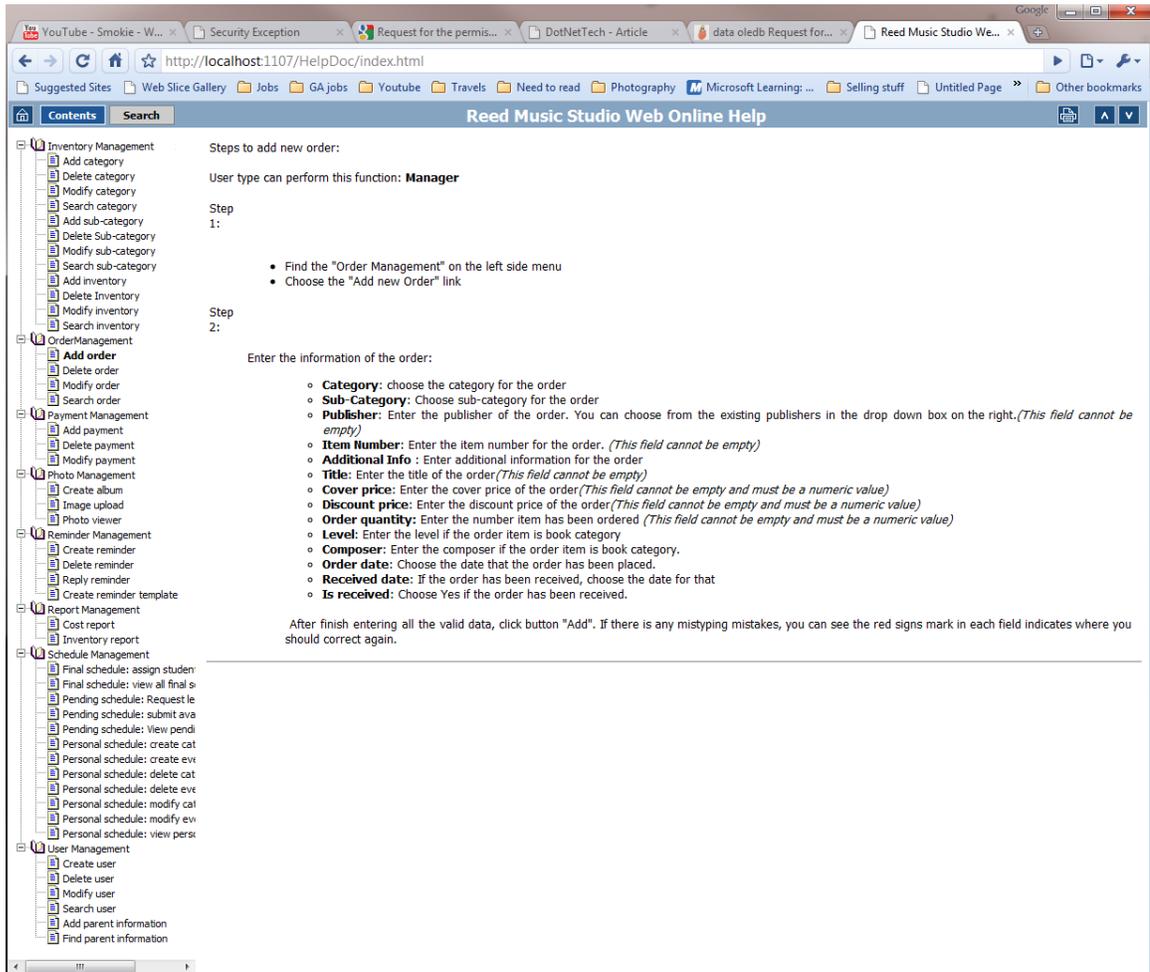


Figure 22. Online help system – common topics

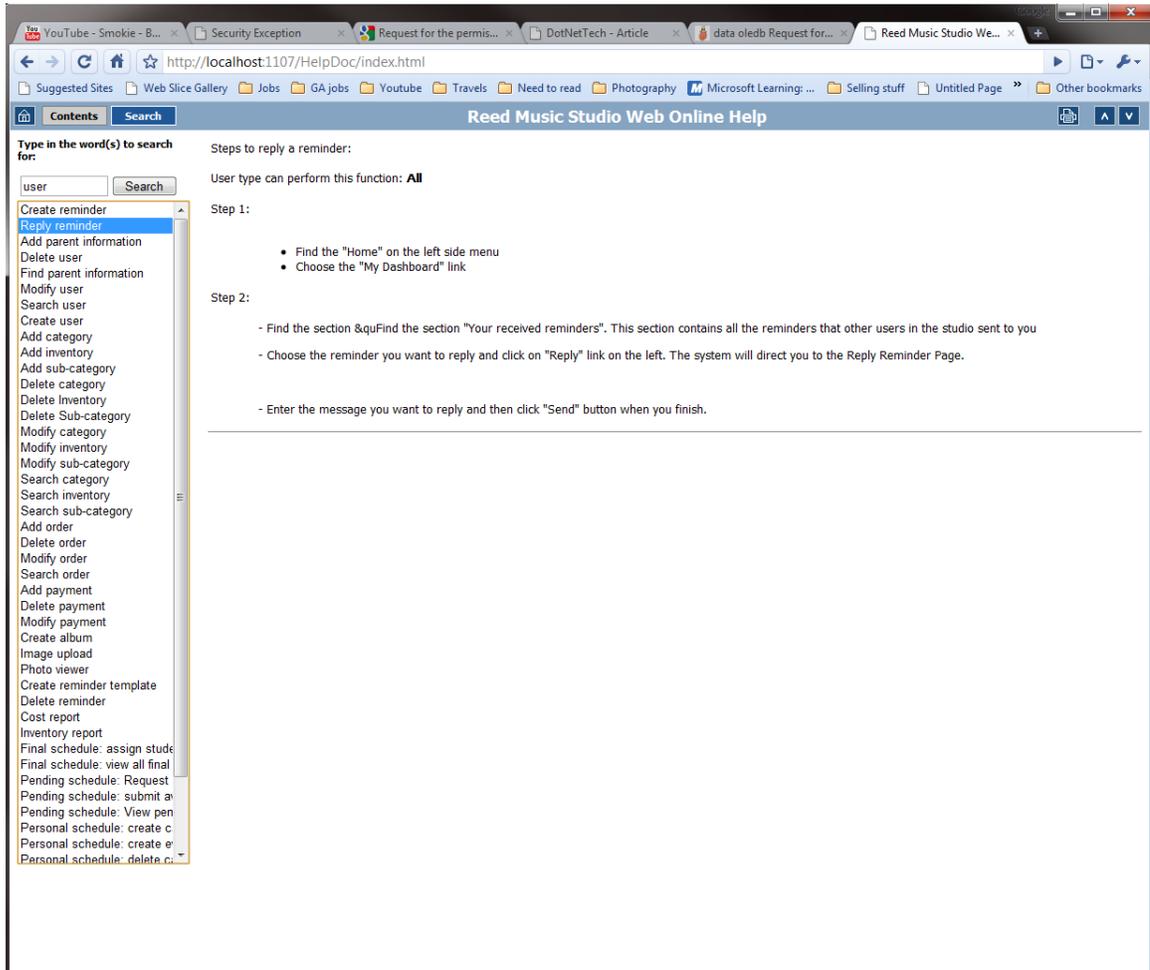


Figure 23. Online help system – search feature