

Homeroom SIS
A Student Information System for K-12 Schools

By

Kevin Ghantous, Michael Lucas, and Josh Maag

Submitted to
the Faculty of the Information Technology Program
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology

University of Cincinnati
College of Applied Science

June 2006

Homeroom SIS

A Student Information System for K-12 Schools

By

Kevin Ghantous, Michael Lucas, and Josh Maag

Submitted to
the Faculty of the Information Technology Program
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology

© Copyright 2006 Kevin Ghantous, Michael Lucas, and Josh Maag

The information in this document is proprietary and may not be reproduced or distributed in whole or in part without the permission of the owner.

_____ Kevin Ghantous	_____ Date
_____ Michael Lucas	_____ Date
_____ Joshua Maag	_____ Date
_____ Hazem Said, Faculty Advisor	_____ Date
_____ Patrick C. Kumpf, Ed.D. Interim Department Head	_____ Date

Acknowledgements

We would like to thank Professor Hazem Said; to whom without his continuous pushing, prodding and support would not have been able to create such an amazing and magnificent system. We would also like to thank our friends and family for putting up with our late nights and never ending ours of work.

Table of Contents

Acknowledgements	i
Table of Contents	ii
List of Figures	iv
Abstract	vii
1. Statement of the Problem	1
2. Description of the Solution	1
2.1 User Profiles	2
2.2 Design Protocols	2
2.2.1 Use Case Diagram	3
2.2.2 Architecture	4
2.2.2.1 Object/Data Mapping	4
2.2.2.2 Object/Controller Mapping	6
2.3 Data Source Design	7
2.4 User Interface	7
2.4.1 Interface Design/Navigation	8
2.4.2 Icons/Graphic Symbols	8
2.4.3 Color Scheme	9
2.4.4 Help	9
3. Objectives of the Project	9
4. Design and Development	11
4.1 Budget	11
4.2 Timeline	13
4.2.1 Senior Design I Accomplishments	13
4.2.2 Senior Design Project Management I Accomplishments	13
4.2.3 Senior Design Project Management II Accomplishments	14
4.3 Project Resources	14
5. Proof of Design	15
5.1 Splash Screen	15
5.2 Login	16
5.3 Page Layout	16
5.3.1 Header	17
5.3.2 System Search	17
5.4 Navigation Forms	21
5.4.1 Student Data Form	21
5.4.2 School Data Form	23
5.4.3 Administration Dashboard	23
5.4.3.1 User Management	24
5.4.3.2 Schedule Management	26
5.5 Student Privileges	27
5.6 Staff Privileges	28

5.6.1 Grade Book	29
5.6.2 Attendance	31
5.6.3 Courses	32
5.7 Administrator Privileges	32
5.7.1 Editing Courses	32
5.8 Error Handling	34
6 Testing	36
6.1 Prototype Debugging	36
6.2 Planned Testing	36
7. Appendices	39
7.1 Appendix A	39
7.2 Appendix B	40
7.3 Appendix C	42
8. Reference	45

List of Figures

Figure 1. Use Case Design	3
Figure 2. IDataAdapter Class Diagram	4
Figure 3. ObjectMappingBase Class Diagram	5
Figure 4. Unneeded Code	7
Figure 5. Database Diagram	7
Figure 6. Microsoft Project Budget	12
Figure 7. Hardware/Software Costs	12
Figure 8. Resource List	15
Figure 9. Initializing Core	15
Figure 10. Login Page	16
Figure 11. Header	17
Figure 12. System Help	17
Figure 13. Search and Navigation	17
Figure 14. Search Results	18
Figure 15. Student View	19
Figure 16. Course View	20
Figure 17. Staff View	20
Figure 18. Parent View	21
Figure 19. Add Student	21

Figure 20. SSN Validation	22
Figure 21. Calendar	22
Figure 22. Add School	23
Figure 23. Administration Dashboard	24
Figure 24. User Search	24
Figure 25. User Search Results	25
Figure 26. Add User	25
Figure 27. Course Editor	26
Figure 28. Period Editor	26
Figure 29. Student Login	27
Figure 30. Student Grades	28
Figure 31. Staff Login	28
Figure 32. Grade Book	29
Figure 33. Grade Assignments	30
Figure 34. Edit Assignments	30
Figure 35. Record Attendance	31
Figure 36. Course Definition	32
Figure 37. Section Editor	33
Figure 38. Teacher Search	33
Figure 39. Search Error Message	34
Figure 40. Error Screen	35

Figure 41. Student Lookup Error Message	35
Figure 42. Exception Catcher	35
Figure 43. Testing Results	37

Abstract

Homeroom SIS is an innovative, Web based student information system designed for K-12 school districts. *Homeroom SIS* is architected from the ground up to overcome the shortfalls of current student information systems; it provides the ability to add additional customized functionality including communication with existing data warehouses and the ability to create (or purchase) specialized add-on-solutions to ensure the district can meet the demands of government, educators, and parents. To provide a flexible system, *Homeroom SIS* has been designed utilizing the latest technologies and standards from Microsoft (including the .Net Framework and C#), World Wide Web Consortium (W3C), and the Schools Interoperability Framework Association (SIFA).

Homeroom SIS

A Student Information System for K-12 Schools

1. Statement of the Problem

Current student information systems (2) are inadequate to handle the needs of today's school districts because they do not provide the necessary tools and features to meet the demands of educators and students. These systems do not provide enough flexibility for a district's unique needs, such as supplemental data storage, custom business process handling, and remote user accessibility. The educator's decision making processes are often hindered by these issues, which can foster invalid student guidance and even unallocated district's funds. Educational institutions are mandated by government agencies to validate monetary needs based on pupil statistics.

2. Description of the Solution

The student information system, entitled *Homeroom SIS*, encompasses many necessities of K-12 schools including attendance, grading, student information, staff information, course management, scheduling, and reporting. The system uses an n-tier model with layers for presentation, object control mapping, business logic, relational data object mapping, and data storage. *Homeroom SIS* is an extensible, flexible, and accessible system that uses a Web browser as a thin-client for viewing and managing information. The modular architecture of the system allows a school district to add plug-ins to meet their specific needs, such as a discipline or cafeteria management system.

2.1 User Profiles

To provide a granular level of security while maintaining capability, *Homeroom SIS* implements multiple user roles and privileges to meet the different functions within a school district. Initial roles and privileges will be based on the following user profiles:

- **Student:** A student has read-only access to their current schedule and assigned grades.
- **Teacher:** Teachers are provided a number of tools accessible from within the district's intranet or remotely from their home computer to help them better manage their classroom. Teachers have direct access to all course sections which they currently instruct. They also have an online grade book for each section where they can manage a student's grades, or contact the student or parents. Classroom attendance is available online on a per section basis.
- **Staff:** This profile includes but is not limited to: secretaries, educational assistants, and nurses. These employees have delegated access to enroll a new student, and modify student information.
- **District/Administrative Staff:** Includes all features of Staff and Guidance Counselor roles. Administrative staff may also add new staff, modify existing staff information/accounts, and remove staff. Only administrative staff may be able to modify student grades. In addition, district staff can add and modify school buildings.

2.2 Design Protocols

The application's architecture is built using Microsoft's .NET framework, permitting the system to be run on Windows, and in the future, Unix-based server

systems through the use of the Mono project. The three-tier design incorporates an uncoupled data layer that allows virtually any data source (e.g. T-SQL compliant database) to be used for data storage. Additionally, with the module architecture, developers have access to a flexible platform to store additional pertinent information unique to a district’s needs, while maintaining consistent user interface and functionality.

Homeroom SIS also follows current

standards in student data management which

comply with recommendations from the

Department of Education, the Schools

Interoperability Framework committee

(SIF), and other local and state school

organizations. The base language used in

Homeroom SIS is C# through ASP.NET in

conjunction with T-SQL, JavaScript, XML,

and XHTML. This system operates as a

centralized portal for district staff (teachers

and administrators) and students to provide

instant access to information and resources.

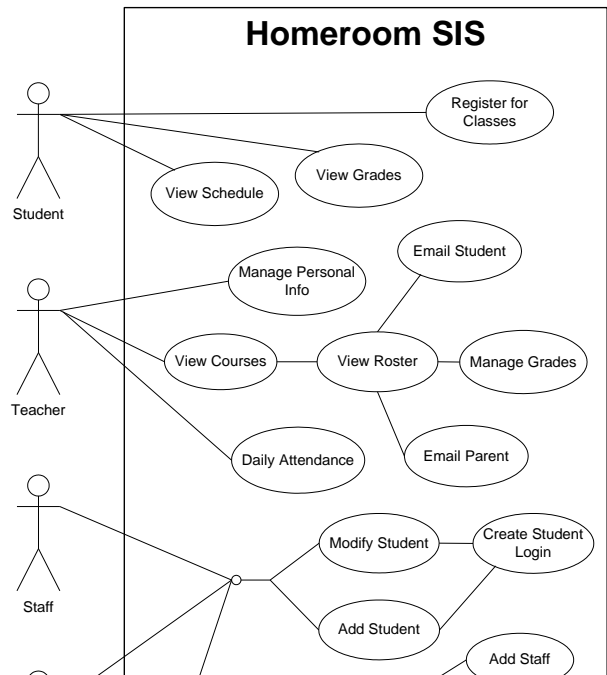


Figure 1
Use Case Diagram
 (See Appendix A for full diagram)

2.2.1 Use Case Diagram

To better analyze the functionality required by the system, as well as streamline tasks, Use Case Diagrams have been employed. Figure 1 depicts all of the users as well as additional, specific roles and the system features needed by that user in a UML diagram. UML stands for Unified Modeling Language. The diagram also shows which

system features each user has access to, and the system features shared between different actors.

2.2.2 Architecture

As with any student information system, Homeroom SIS must be capable of providing specific services and functionality to any user. These services may be internal systems or external systems, but all systems must have the capability to communicate and interact seamlessly. This is accomplished by designing the system like building blocks. Data access can be flexible and independent of core modules, and plug-in components are capable of attaching to the existing data structure, or spawning a separate data source and simply linking to existing core data; this allows for extreme flexibility. Additionally, tools have been built into the core of the system to ease the development and integration process.

2.2.2.1 Object/Data Mapping

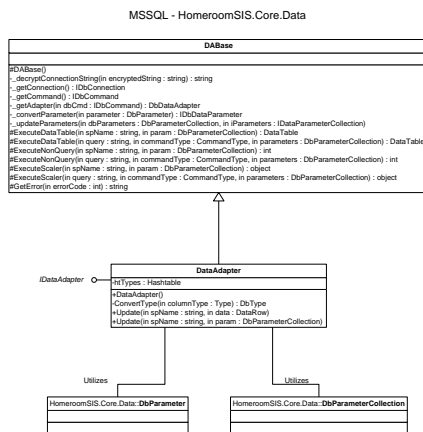


Figure 2
IDataAdapter Class Diagram
 (view Appendix B for full diagram)

While keeping with the concept of modular design and flexibility, an abstract core data adapter has been constructed to provide common data access object for all core modules and plug-in to use. The data access layer is abstracted by the `IDataAdapter` interface. This interface provides a simple, yet comprehensive collection of methods and properties to access any structure within the data source. For core development under Microsoft SQL

Server, the `IDataAdapter` interface is implemented utilizing a base adapter class, `DABase`. The `DABase` is a smart class, with the ability to initialize and load the necessary database driver without having the developer perform connection management. The `DABase` is coupled with two additional data classes, `DbParameter` and `DbParameterCollection`. These two classes provide generic implementations of the `IDbParameter`, `IDataParameter`, and `IDataParameterCollection` interfaces. The `DABase` is aware of these universal database parameter classes, and knows how to utilize and convert these classes into the appropriate `IDbParameter` data types for the available relational database management system.

All data points are object oriented within the Homeroom SIS core. These objects utilize methods such as inheritance, polymorphism, and reflection to accomplish their given tasks while not coupling objects to a data source. To streamline data object data mapping, the current implementation utilizes an `ObjectMappingBase` class depicted in Figure 3.

This class is designed to act as a wrapper for the `.NET Framework System.Data.DataRow` class,

and provides appropriate, strongly data-typed methods to access all field data-types. The `ObjectMappingBase` make use of an object implementing the `IDataAdapter` interface (Figure 2) for data access. An *Update* method is virtualized and made available from the `ObjectMappingBase` to provide consistent data updating. Therefore, all

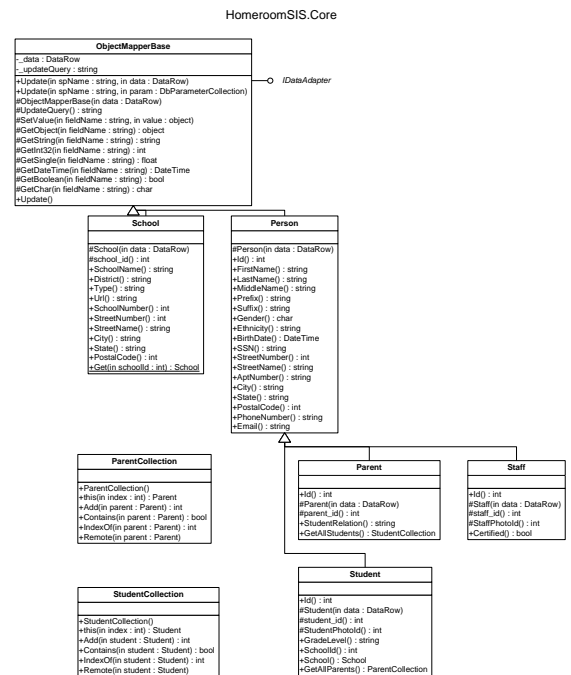


Figure 3
ObjectMappingBase Class Diagram
(view Appendix B for full diagram)

objects that represent database tables or views will inherit the `ObjectMappingBase` class.

To bridge between objects and data sources, all modules have pre-defined module controllers. These controllers are explicitly defined by a specialized interface stored within the core, and are made accessible by use of a `ModuleControllerUtility` class. By using reflection, this class will examine all loaded assemblies and locate the object which implements the Controller Interface for that module. If no implementation is found, a null value is returned. By developing the module controller, it is possible to construct a specialized controller for each module if it accessed a different data source.

2.2.2.2 Object/Controller Mapping

Providing a consistent mechanism of populating and storing form information is essential when developing a system involving data-entry. To assist in this task, the Object/Controller Mapping Interface is designed to ensure that object properties and fields are efficiently retrieved with minimal code. All controls that perform Object/Controller Mapping (OCM) must implement the `IOCMappable` interface. The functionality of the OCM infrastructure is deeply rooted in the .NET framework `System.Reflection` libraries. By utilizing reflection, all Web controls perform property retrieval and assignment by simply obtaining the object and the name of the property it is associated with. If the property contains the necessary `ToString()` method and string parsing constructor, the control is capable of performing the appropriate OCM regardless of data type. Compared to standard form development, this technology releases the direct association of a single control (e.g. `TextBox`) to a single

object property or database field (Figure 4). For example:

```
/*The following code can be eliminated from a textbox control*/  
Student stu = Student.GetStudent(studentId);  
txtStudent.Text = stu.FirstName; //Property Assignment  
//
```

Figure 4
Unneeded Code

2.3 Data Source Design

The *Homeroom SIS* database was created from analyzing the SIF framework XML documentation (3), and extrapolating a database which follows the third Normal Form. The database's relationship diagram (Figure 5) shows its ability to extend the data

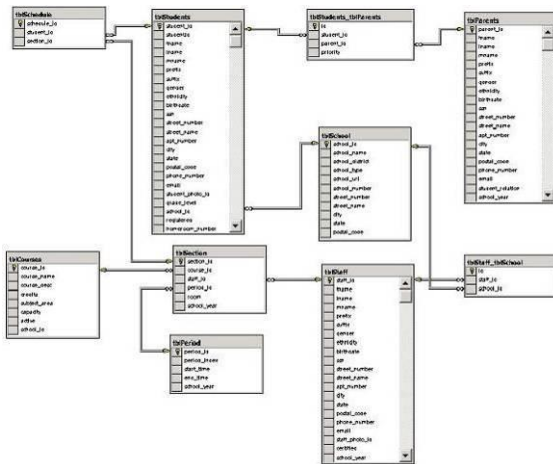


Figure 5
Database Diagram
(view Appendix C for Full Diagram)

structure to provide storage for the courses, sections, and periods. Data queries are performed using stored procedures to provide a higher level of security so SQL injection attacks cannot be effectively implemented. All of the database management systems in the data layer have the capability of supporting views and stored procedures in T-SQL, ensuring the

goal of an environment independent application.

2.4 User Interface

The following sections will review what was used to design the project's interface, graphics, color scheme, and help section.

2.4.1 Interface Design/Navigation

The user interface encompasses W3C standards, including XHTML 1.0, CSS 2.0, and JavaScript 1.2. This simplistic design allows the user interface to adapt to different hardware platforms such as a workstation, notebook, tablet, mobile device, blackberry, and even mobile phones. For higher-end browsers, the user interface utilizes Asynchronous JavaScript and XML (AJAX) which provides the user with a more responsive “desktop application” experience. This includes anticipating the user’s actions, and providing real-time suggestions to intelligently adapt to assist users with their work-flow. All components of the user interface are modular, allowing for specific, independent modifications by the *Homeroom SIS* development team or via the plug-in architecture.

A search box provides a single access point for system-wide searching. Content covered within the server is dependent upon module implementation, but can provide such abilities as student, staff, and course search. Current searching implementation utilizes Full-Text Indexing. This helps speed up searching, and return partial results if the user is not fully aware of how to spell their criteria.

2.4.2 Icons/Graphic Symbols

Homeroom SIS contains a minimal amount of icons and graphics. This helps decrease loading time for low speed or remote connections. The first image users see is an interchangeable logo of the school’s choosing on a splash screen. This logo is scaled and placed in the upper left corner of the screen, or defaulted to a general *Homeroom SIS* logo.

2.4.3 Color Scheme

The color scheme for *Homeroom SIS* will default to a blue, white, and black theme. These colors can be configured using CSS files which can be customized by a *Homeroom SIS* developer, or other capable web developer.

2.4.4 Help

The help section dynamically loads onto the page mounted on the right side of the screen. Utilizing floating layers, the help system does not obscure any currently displayed content, and will display help questions based on the context and modules of the current page.

3. Objectives of the Project

1. Student information system which will track students and their related information in K-12 school districts. The application will:
 - a. Follows the Schools Interoperability Framework
 - b. Provide security through logins with role-based permissions
 - c. Allow information related to schools to be archived and stored
2. Use C# with ASP.NET for the core and allow use for any .NET language to develop a plug-in
3. Accessible through standard Web browsers that support JavaScript and allow secure access outside the school's intranet
4. Modular architecture consisting of core system and modules
 - a. Core modules that perform all needed actions to store and use basic student information

- b. Grading and attendance modules that implement core interfaces and dynamically load to the system
5. Data layer for relational databases that support SQL, Views and Stored Procedures, and written using Microsoft SQL 2000
6. Application architecture with an n-tier approach for logic separation
7. Template layout for every page.
8. Intelligent user interface allowing for real-time interaction with information to allow ease-of-use for inexperienced users through the use of AJAX
9. Allow for teachers to perform tasks related to students
 - a. View their courses, view class roster, manage grades, email student and email parent
 - b. Record and modify student attendance
 - c. Manage personal information (address, phone number, etc.)
10. Allow for students to view course schedule and grades
11. Staff are able to add and modify students
12. In addition, administrative and district staff are able to:
 - a. Add, manage and remove staff
 - b. Create course definitions and schedules
 - c. Manage student schedules
 - d. Modify student grades
 - e. Manage course definitions

4. Design and Development

The next sections will cover the budget, timeline, and hardware/software costs of the project.

4.1 Budget

The cost to develop *Homeroom SIS* at the University of Cincinnati is minimal. Since all of the essential software has been acquired, an example of “real world” costs has been outlined in Figure 6. If this project was being developed by a startup company, the costs would come up to \$37,320.72. The actual cost is what the project would cost to develop, and the remaining costs are hardware and software costs.

Due to the amount of capital this project would take to develop in a business environment, it would potentially require an investor. Once an investor is acquired, it can be assumed they would want *Homeroom SIS* to be worthwhile and turn a profit.

Homeroom SIS currently has a basic set of features, but has the ability to expand to handle more features than other student information systems, which justifies the cost of \$14 per student (Figure 7). Potential clients may need to purchase hardware and software to run *Homeroom SIS* properly. A Windows environment is recommended with Microsoft SQL Server 2000 for a school district with 1,000 students, although the project can handle other database types by modifying the database adapter and module controllers. Two servers are recommended for redundancy in case one of the servers should crash, both of them hosting internet services and the database management. Even though the client expenses total \$35,098 (Figure 7), this may change depending on the school’s current infrastructure and the number of students. For instance, *Homeroom SIS*

could cost much less if the school uses a Linux operating system with an open source database system running on the Mono framework.

Task Name	Total Cost	Actual Cost	Remaining Cost
SIF Data Collection	\$840.00	\$840.00	\$0.00
Research Application Pools	840.00	840.00	0.00
Revisit Problem Statement	840.00	840.00	0.00
Create Database	840.00	840.00	0.00
Use Case Diagram	840.00	840.00	0.00
Develop Timeline	840.00	840.00	0.00
Revisit Solution	3,960.00	3,960.00	0.00
Narrow Scope	3,960.00	3,960.00	0.00
Add Technical Detail	3,960.00	3,960.00	0.00
O/R Mapping	2,182.32	2,016.00	166.32
Solidify Information Retrieval	1,176.00	672.00	504.00
Required Methods	1,680.00	1,680.00	0.00
Properties/Fields	554.40	0.00	554.40
Events	554.40	0.00	554.40
Develop Stored Procedures	840.00	840.00	0.00
Begin Intelligent User Interface	840.00	126.00	714.00
Create Student Module – Core	840.00	0.00	840.00
Insert Timeline into MS Project	840.00	0.00	840.00
View Courses/Roster/Students - Core	1,920.00	0.00	1,920.00
Manage Grades – Plug-in	1,920.00	0.00	1,920.00
Project Risk Plan	1,920.00	0.00	1,920.00
Security Policy Interactions	840.00	0.00	840.00
Finish Module Design	840.00	0.00	840.00
Project Unit Testing	840.00	0.00	840.00
Create Administrative Functions	1,782.00	0.00	1,782.00
Plan Upcoming Quarter Design Changes	831.60	0.00	831.60
Totals	\$37,320.72	\$22,254.00	\$15,066.72

Figure 6
Microsoft Project Budget

Software	Qty	Cost	Total
Homeroom SIS License (per Student)	1000	\$14.00	\$14,000.00
Microsoft Windows Server 2003 - Standard Edition	2	799.00	1,598.00
Microsoft SQL Server 2000 (per Processor)	2	5,000.00	10,000.00
Hardware (Dell Ohio Educational Pricing)			
Implementation Server (PowerEdge SC1425)	2	3,000.00	6,000.00
Personal Computer	5	700.00	3,500.00
Totals		9,513.00	\$35,098.00

Figure 7
Hardware/Software Costs

4.2 Timeline

The timeline for this project is broken up through three quarters. Each section below will detail through our achievements during each quarter.

4.2.1 Senior Design I Accomplishments

During Senior Design I, we accomplished the following:

- Gathered information on student information systems
- Researched School Interoperability Framework (SIF)
- Examined different angles to develop our project
- Planned timeline for future quarters
- Created UML diagrams tracing out programming paths
- Developed proposal and presentation

While looking for examples of other student information systems, we gathered lots of information on what they should and should not do. SIF is the current standard for our field of study, so we mostly focused on how different programs implement SIF. From this data, we began to create UML diagrams on how to design the project.

4.2.2 Senior Design Project Management I Accomplishments

During Senior Design Project Management I, we accomplished the following:

- Created virtual server to store our project
- Designed and developed database and stored procedures
- Outlined Web layout
- Defined project core, modules, and plug-ins
- Developed project core

- Designed data classes
- Developed student module
- Prepared Design Freeze prototype and presentation

The second quarter consisted of designing a prototype of our project. We began this section by setting up our server using Microsoft Virtual Server 2003 and later switched our project to VMWare GSX virtual server to emulate an instance of Windows Server 2003. Visual Studio .NET 2003 and SQL Server 2000 were installed to begin the process of creating our project beginning with the database. While working on the project, we removed the parents' functionality from the project as we noticed the project's scope was beginning to exceed our time constraints.

4.2.3 Senior Design Project Management II Accomplishments

During Senior Design Project Management II, we accomplished the following:

- Defined roles for students, staff, and administrators
- Developed course, grades, and attendance module
- Generated webpage template
- Modified user login to reflect required tasks

The third and final quarter consisted of creating the final few modules and linking the information together. A web template was created to streamline the form making process.

4.3 Project Resources

The resources for this project, as defined in Figure 8, were each selected for a specific reason. The server this project resides on runs VMWare's GSX Server which

emulates an instance of Windows Server 2003. The project was developed under Visual Studio .NET 2003 and SQL Server 2000.

Name	Type	Base Calendar
Kevin Ghantous	Developer	24 Hours
Michael Lucas	Developer	24 Hours
Josh Maag	Developer	24 Hours
Hazem Said (1)	Advisor	24 Hours
Hosting Server	Hardware	
Personal Computer	Hardware	
Windows XP	Software	
Visual Studio .NET 2003	Software	
VMWare GSX Server	Software	
Microsoft SQL Server 2000	Software	

Figure 8
Resource List

5. Proof of Design

The following sections will detail the design of our project, and all of deliverables users will interact with.

5.1 Splash Screen

The project begins with a splash screen (Figure 9). This initial screen has our project's title and logo which



Figure 9
Initializing Core

represents the project. The logo reflects the project because *Homeroom* is made of individual modules merged together to form the core, and a puzzle symbolizes this analogy. The Web page is programmed using the JavaScript XMLHttpRequest object (or alternate Microsoft.XMLHTTP ActiveX object). This object calls upon a Web service

to retrieve the status of the initializing system. The status bar is the information region which is being used to display the status received from the Web service. The first stage of startup is to initialize the core.

During this stage, ASP.NET is performing the pre-compile of the project and is initializing the core assemblies known as DLLs (Dynamic Linked Libraries). The next phase of loading consists of the program's modules. These modules contain any extra features a school adds beyond managing their students and faculty's data. Once the plug-in modules finish loading, the progress bar is set to 100%, and the user is informed the load is complete. The program then redirects the user to the login page.

5.2 Login

The initial page a user will see is the login page (Figure 10). The system now checks to see if a user is logged in or not, and will redirect the user back to this page if they are not logged in. Once the user logs in, they are redirected to either a student or staff page depending on their roles.

5.3 Page Layout

This page is made up of several sections consisting of the header, system search, navigation, and main context section.



Figure 10
Login Page

5.3.1 Header

The header is divided into two sections (Figure 11). The left side contains the *Homeroom SIS* logo, while the right hand side includes the name of the currently logged



Figure 11
Header

in user, home page link, help access, and login/logout system. The help section of the

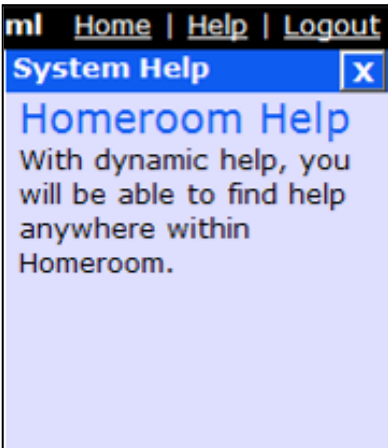


Figure 12
System Help

project loads from the right side of the screen onto the user's page (Figure 12). It has a title in a banner on the top next to a close button, and a welcome message in the body. Eventually, this section will load dynamic content containing information pertinent to the user, based on their current page. There will also be an index the user can select in order to pick a topic of their choosing.

5.3.2 System Search

The system search is the first section of the project where the user can view internal labels, which contain meta data (Figure 13). These internal labels, which are a custom property of *Homeroom SIS*'s controls, explain to the user what they need to enter into a text box. The search works

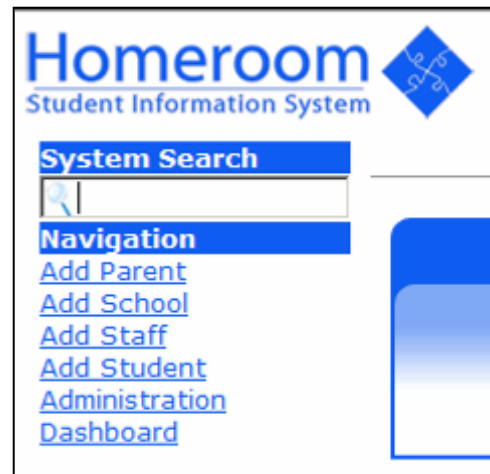


Figure 13
Search and Navigation

with partial results, so a user can enter a portion of a name, and the database will be searched for any relevant matches. The search box looks for students by last name, instructors, and courses. In this example, a search is performed using the letter “s” as search criteria. There are six items returned as a result (Figure 14). A number showing the amount of results found is listed at the top of the page, and each result is returned below grouped according to type. The type with the greatest amount of results is automatically moved to the top of the

list. If pertinent, a descriptive element is linked with each result, such as a student has their grade listed below their name. Once the results of the user’s search are displayed on the screen, the user is able to click on the result, and be redirected to another page which will display the appropriate information. Each result

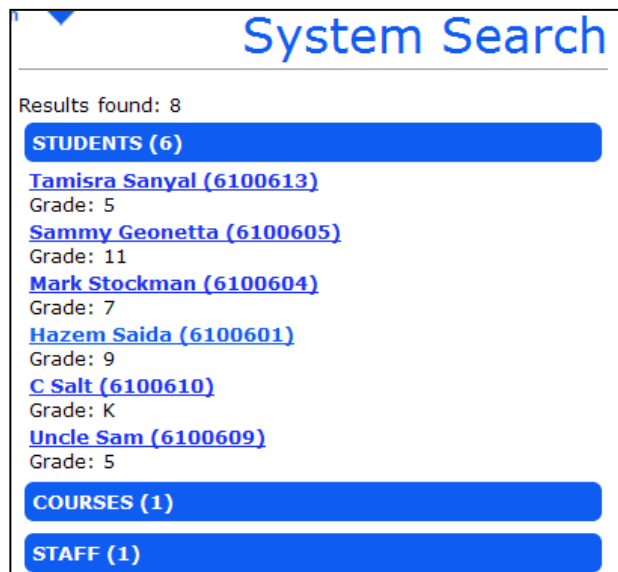


Figure 14
Search Results

set shows all of the items returned but can be collapsed as courses shows in Figure 16. The results will link to a new page which is defaulted to view mode. The other mode edit will be discussed in a later section. The link for parents, staff, and students are similar, so here we will show an example of the student view form (Figure 15). This page displays the student’s name, address, email, phone number, picture, and courses with grades if the student is currently taking any classes. If no picture is available, a default images is used in its place. The course view form displays the course name, description, credits, subject

area, capacity, whether it is active or not, and any course sections (Figure 16). Figure 17 and Figure 18 are the staff and parent views respectively. Depending on the user's role, they will see different information in the blue bar at the top of the page, which will be covered later.

Student Information


[Edit Information](#)

Marky Stockman

Address
42 Stockman Rad 4
Cincinnati OH 45908

Phone Number: (513) 555-8329

Email: mark.stockman@uc.edu



Courses	
Course	Grade
Writing (601-3)	90.91 %

Figure 15
Student View

Courses

[Create New Course](#) | [Edit Course Information](#)

Course Name: Biology
Course Description: test
Credits: 3
Subject Area: Science
Capacity: 25
Active: Yes

[Add Section](#)

Period	Section	Room	Teacher
05	Biology (201-6)	104	Jolie

Figure 16
Course View

Staff Information

[Edit Information](#)

Carly Simpson

Address:
 1
 45220

Phone Number: (404) 555-1234
Email:
Certified: Yes




Figure 17
Staff View

**Figure 18
Parent View**

5.4 Navigation Forms

Below the system search box is the main navigation for the project (Figure 13). These links are loaded for all users with an administrator role, and allow the user to add a parent, student, staff, school, or user to the project. Since the add parent, staff, and student pages are almost identical, we will go over just the student form.

5.4.1 Student Data Form

The student data form is one example of our data entry forms. (Figure 19). This form displays the full potential of our internal labels. Each text box

**Figure 19
Add Student**

which may not be completely clear to a user is automatically filled in with a gray text describing to the user what should be typed in. As the user selects a text box, the gray text clears allowing the user to enter in text. If the user moves onto the next text box without entering any information, the internal label returns.

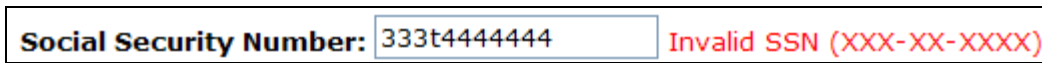


Figure 20
SSN Validation

There are certain validations attached to the system to make sure the user is entering in the correct information when the system cannot dictate what they are entering in through a drop down list. One of these validations is on the Social Security Number.

This validation allows for the user to enter the Social Security Number with or without dashes. If the user does not enter the right amount of numbers, or enters a letter, a red warning appears next to the text box alerting the user to their error as in Figure 20.

As soon as the number is corrected, the error label goes away. This same type of validation appears on the birth date and phone number text boxes.

Next to the birth date text box is a calendar icon (Figure 21).

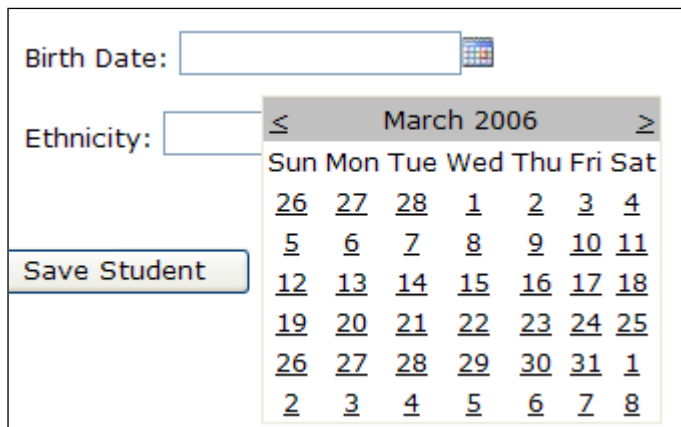
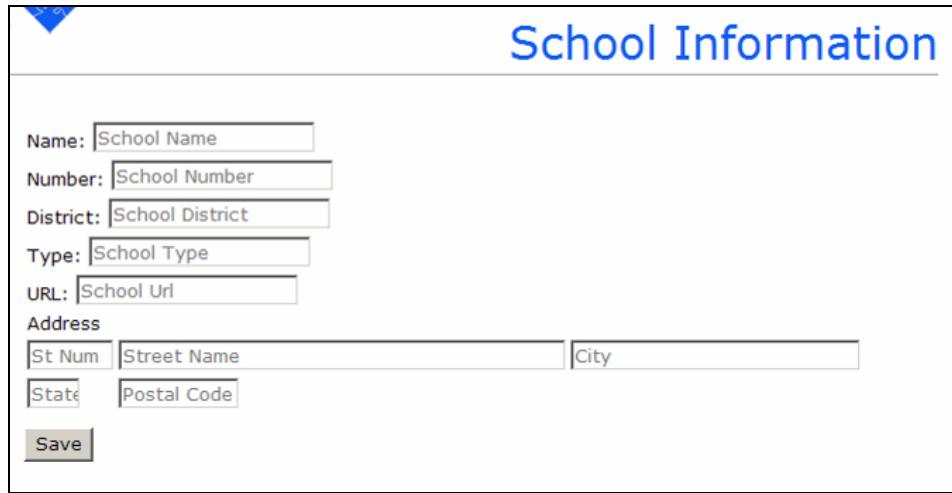


Figure 21
Calendar

Clicking on this icon will bring up a calendar containing the current month, and the option to move forward or backwards between the months. The user can click on a date from the calendar, and the form will automatically close the calendar and enter the selected date into the birth date text box.

5.4.2 School Data Form

Another form in our navigation allows schools to be added to the database (Figure 22). This functionality will be used when the application is first installed, and later on if buildings are added to the school district. The school's name, building number, district, type (elementary, middle, high), url, and address are added in this form.



School Information

Name:

Number:

District:

Type:

URL:

Address

Figure 22
Add School

5.4.3 Administration Dashboard

The final element of the navigation is the administration dashboard. This page is broken into three sections including school, user, and schedule management (Figure 23). The school management section has redundant links to add a new student, staff member, or parent to the database which were previously covered.



Figure 23
Administration Database

5.4.3.1 User Management

The user management redirects the user to a Web form which searches the database for students, parents, and staff login names based on partial search criteria performing the same as our system search. The search can be narrowed down to look for just students, parents or staff to minimize results from a drop down list (Figure 24).

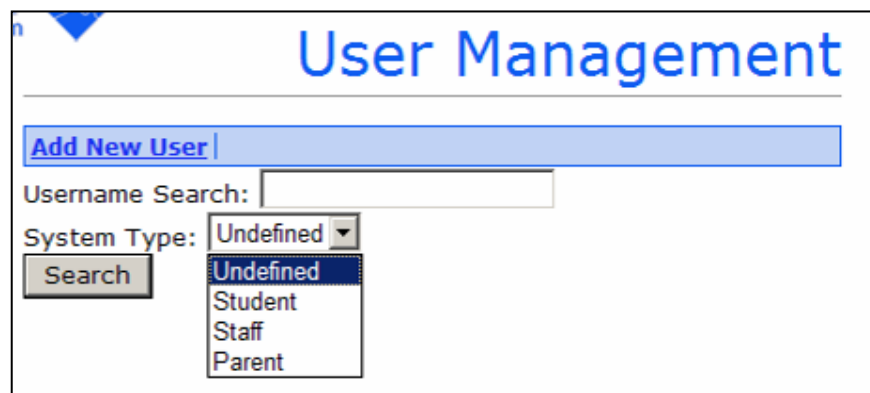


Figure 24
User Search

The results are returned in a table below the search form (Figure 25). New users can be added on the same page by clicking on the Add New User link. The page redirects itself

to a new form where the username, password, system id, and system type are entered (Figure 26).

The screenshot shows the 'User Management' interface. At the top, there is a blue header with the text 'User Management'. Below the header is a blue button labeled 'Add New User'. Underneath, there is a search form with the following elements: 'Username Search:' followed by a text input field containing the letter 'S'; 'System Type:' followed by a dropdown menu showing 'Undefined'; and a 'Search' button. Below the search form is a table with three columns: 'Username', 'SystemType', and 'System ID'. The table contains two rows of data: one for 'saidhm' (Student, System ID 11) and one for 'sheetby' (Staff, System ID 12).

Username	SystemType	System ID
saidhm	Student	11
sheetby	Staff	12

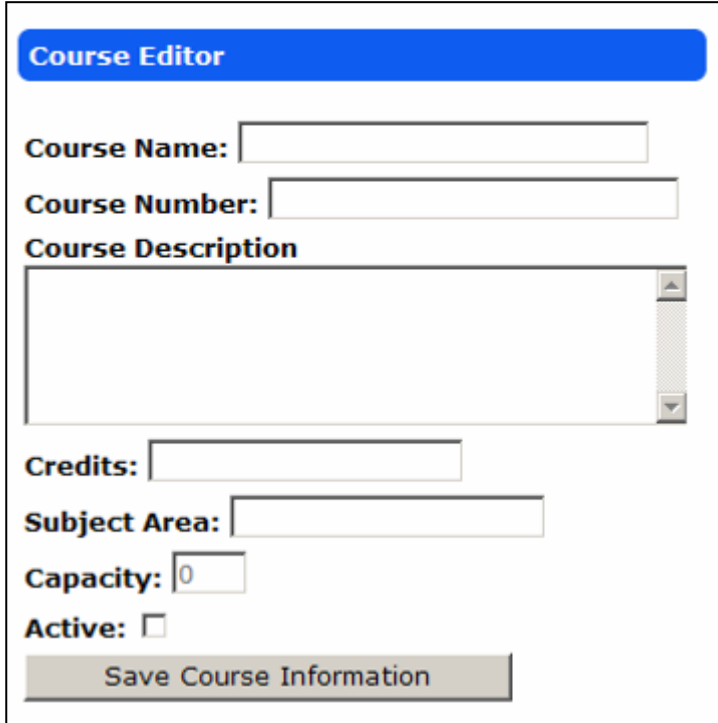
Figure 25
User Search Results

The screenshot shows the 'User Management' interface for adding a new user. At the top, there is a blue header with the text 'User Management'. Below the header is a blue button labeled 'Add New User'. Underneath, there is a section titled 'User Account' with the following form fields: 'Username:' followed by a text input field; 'Password:' followed by a text input field; 'Verify Pwd:' followed by a text input field; 'System Id:' followed by a text input field; and 'System Type:' followed by a dropdown menu showing 'Undefined'. At the bottom of the form is a 'Save' button.

Figure 26
Add User

5.4.3.2 Schedule Management

In this section of the dashboard, administrators can create new courses and their



Course Editor

Course Name:

Course Number:

Course Description

Credits:

Subject Area:

Capacity:

Active:

Save Course Information

Figure 27
Course Editor

end time can be edited by selecting the Edit link on the left hand side. The application checks the current times against these times and then displays specific information to the user based on the period (Figure 28).

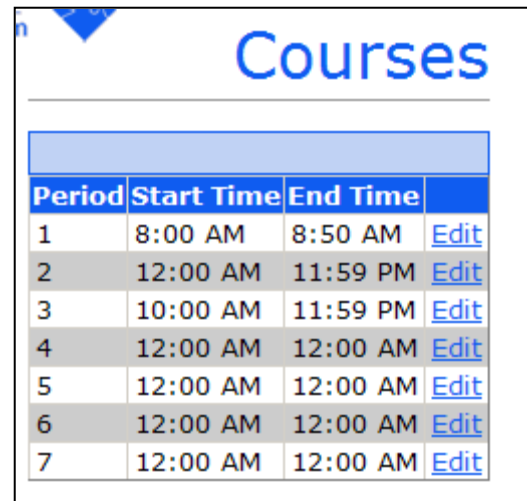
sections on one page and edit period definitions on another.

On the manage courses editor page, the name, number,

description, credits, subject, and class capacity are entered.

There is also a check box to activate the class. If the class is not activated, staff will not be able to view it (Figure 27).

On the edit period page, the period number, start time, and



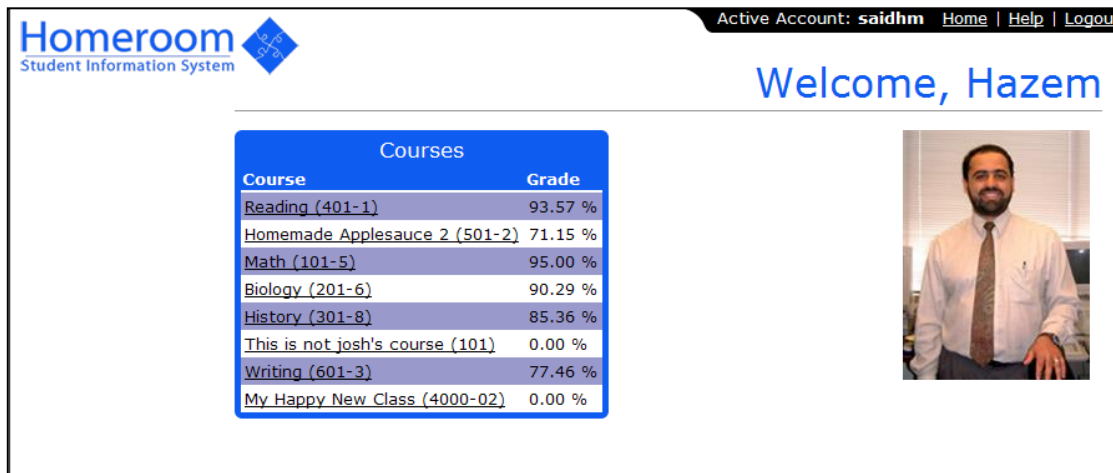
Courses

Period	Start Time	End Time	
1	8:00 AM	8:50 AM	Edit
2	12:00 AM	11:59 PM	Edit
3	10:00 AM	11:59 PM	Edit
4	12:00 AM	12:00 AM	Edit
5	12:00 AM	12:00 AM	Edit
6	12:00 AM	12:00 AM	Edit
7	12:00 AM	12:00 AM	Edit

Figure 28
Period Editor

5.5 Student Privileges

When a student logs in, they are redirected to the studentstart.aspx page, and greeted by the application (Figure 29). Since a student's role is much more limited than a staff or administrator, they are unable to see the side navigation or system search. The student's picture and all of the student's courses and grades are shown.



The screenshot shows the Homeroom Student Information System interface. At the top left is the logo for Homeroom Student Information System. At the top right, it says "Active Account: saidhm" with links for Home, Help, and Logou. The main heading says "Welcome, Hazem". Below this is a table of courses and grades, and a profile picture of a man in a white shirt and tie.

Course	Grade
Reading (401-1)	93.57 %
Homemade Applesauce 2 (501-2)	71.15 %
Math (101-5)	95.00 %
Biology (201-6)	90.29 %
History (301-8)	85.36 %
This is not josh's course (101)	0.00 %
Writing (601-3)	77.46 %
My Happy New Class (4000-02)	0.00 %

Figure 29
Student Login

A student can select one of their courses on the previous page to view the section information and grades for individual assignments. This page displays the section id, room number, period, course name, capacity, credits, subject area, teacher profile, teacher's email, if the teacher is certified, and if the section is active (Figure 30). From here, the student can view the course definition, teacher profile, or email the teacher. Below this information are the student's grades for each assignment and points possible.

Section ID: 401-1	Room Number: 100	Period: 03
Course Name: Reading	Active: Yes	Capacity: 30
Credits: 3	Subject Area: English	
Teacher Name: Michael Lucas	Email Address: lucasml@homeroom.com	Certified: Yes

Student Information

Name: Hazem Saida

Assignment	Earned	Points Possible
Homework1	11	10
Homework2	10	10
Homework3	8.5	10
Homework4	9	10
Test	92.5	100

Figure 30
Student Grades

5.6 Staff Privileges

Staff users are provided more access into the application. When a staff member logs in, they are presented with the system search, current period course information, and their course list. This information consists of the period, room number, class start and stop times, and the current courses grade book and attendance. If the teacher logs in before or after school hours, they are just presented with their courses (Figure 31).

Homeroom Student Information System

Active Account: **carly** [Home](#) | [Help](#) | [Logout](#)

System Search

Search Terms

Welcome, Carly

Math

Period: 4 **Room: 103** **6:00 PM - 7:00 PM**

[Gradebook](#) [Attendance](#)

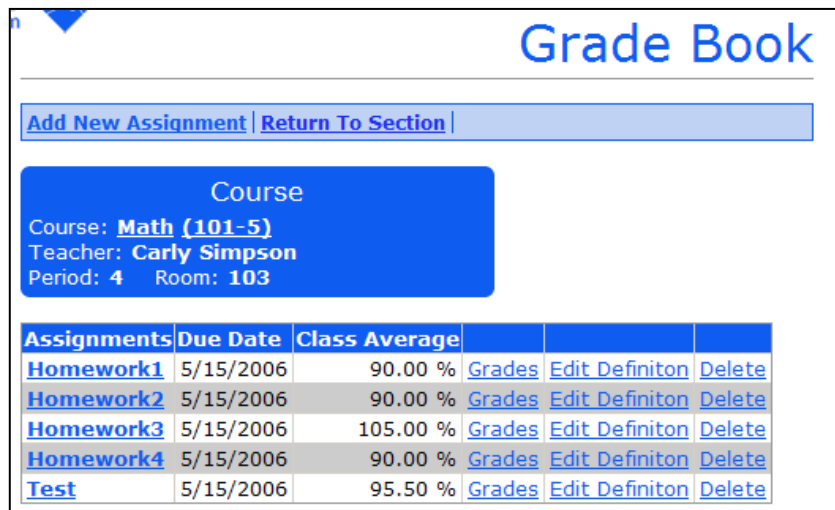
Registered Courses

[Math \(101-5\)](#)

Figure 31
Staff Login

5.6.1 Grade Book

The grade book shows all assignments for the designated course including the name, date due, and current class average (Figure 32). For repetition, the grade book also states the course name, teacher, period, and room number. The class average is dynamically built based on the students' grades. By clicking on the assignment name or "Grades" link, a teacher can assign grades for a specific assignment (Figure 33). Teachers can also edit the assignment to change its name, type, point value, or the assigned/due date (Figure 34). Finally, teachers can delete the assignment all together.



The screenshot shows a web interface titled "Grade Book". At the top right, the title "Grade Book" is displayed in blue. Below the title, there are two links: "Add New Assignment" and "Return To Section". A blue box contains course information: "Course: Math (101-5)", "Teacher: Carly Simpson", "Period: 4", and "Room: 103". Below this is a table with columns for "Assignments", "Due Date", "Class Average", and three action links: "Grades", "Edit Definiton", and "Delete".

Assignments	Due Date	Class Average			
Homework1	5/15/2006	90.00 %	Grades	Edit Definiton	Delete
Homework2	5/15/2006	90.00 %	Grades	Edit Definiton	Delete
Homework3	5/15/2006	105.00 %	Grades	Edit Definiton	Delete
Homework4	5/15/2006	90.00 %	Grades	Edit Definiton	Delete
Test	5/15/2006	95.50 %	Grades	Edit Definiton	Delete

Figure 32
Grade Book

Grade Book

Course

Course: **Writing (601-3)**
 Teacher: **Kevin Ghantous**
 Period: 3 Room: 102

Assignment

Assignment: **Homework1**
 Points Possible: **11**
 Assigned On: **4/30/2006** Date Due: **5/15/2006**

Student	Points Earned
John Nyland	8
Annu Prabakhar	11
Sam Geonetta	10
H Salt	
Uncle Sam	9
Mark Stockman	10
Hazem Said	10
Russel McMahon	
Tom Wulf	
Bob Marley	

Figure 33
Grade Assignments

Grade Book

Course:
Writing (601-3)

Assignment

Assignment Name:

Assignment Type:

Points Possible:

Date Assigned:











Date Due:

Figure 34
Edit Assignment

5.6.2 Attendance

By clicking on the attendance link from the grade book, a teacher can record the attendance for the day. Each student's picture and name will appear on the screen with three buttons below their name (Figure 35). The buttons have a P, T, or A on them which stand for present, tardy, and absent respectively. To make this process more efficient, the teacher can click on each student's photo to enter the student as present. Once all students have been recorded, the teacher selects the save attendance button to enter the information into the database. At the top of the page, there is a calendar control where a teacher can view previous dates attendance was taken.

The screenshot shows a web interface titled "Attendance" for the course "Writing (601-3)". The current date is 5/30/2006. The interface displays a grid of student records, each consisting of a photo, a name, and three buttons labeled P, T, and A. The students listed are Johnny Nyland, Annie Prabakhar, Sammy Geonetta, C Salt, Uncle Sam, Marky Stockman, Hazem Saida, Russy McMahon, Tommy Wulf, and Bobby Marley. A "Save Attendance" button is located at the bottom left of the grid.

Writing (601-3)					Current Date: 5/30/2006				
 Johnny Nyland P T A	 Annie Prabakhar P T A	 Sammy Geonetta P T A	 C Salt P T A	 Uncle Sam P T A					
 Marky Stockman P T A	 Hazem Saida P T A	 Russy McMahon P T A	 Tommy Wulf P T A	 Bobby Marley P T A					

[Save Attendance](#)

Figure 35
Record Attendance

5.6.3 Courses

When a teacher selects on of their courses, they are brought to the courses page. This page displays the same information as the student's view, along with a few more options. The teacher has direct access to the class's attendance, grade book, and student roster complete

with the student's

current grade

(Figure 36).

The screenshot shows a web interface titled 'Courses'. At the top right, there are links for 'Open Grade Book' and 'Open Attendance'. Below these are several rows of course information: Section ID: 101-5, Room Number: 103, Period: 04; Course Name: Math, Active: No, Capacity: 25; Credits: 3, Subject Area: Mathematics; Teacher Name: Carly Simpson, Email Address: carly.simpson@email.uc.edu, Certified: Yes. Below this is a section titled 'Student Roster' with a table containing two students: Hazem Said with a 95.00% grade, and John Nyland with N/A.

Section ID:	101-5	Room Number:	103	Period:	04
Course Name:	Math	Active:	No	Capacity:	25
Credits:	3	Subject Area:	Mathematics		
Teacher Name:	Carly Simpson	Email Address:	carly.simpson@email.uc.edu	Certified:	Yes

Student Name	Current Grade
Hazem Said	95.00 %
John Nyland	N/A

Figure 36
Course Definition

5.7 Administrator Privileges

As stated before, administrators are the only users who are able to access the navigation panel on the left side of the screen. In addition to this, they are able to edit grades, courses, and sections after they have been added.

5.7.1 Editing Courses

When an administrator views a course, they are given the opportunity to edit the course, which uses the same form as Figure 18, or add a section (Figure 37). The section editor allows the user to choose the teacher name (Figure 38), and enter in the section number, room number, and period. At this point, the teacher can add students to the course through a partial name search.

Courses

Section Editor

Course: Math

Teacher:

Section Number:

Room Number:

Period:

Student Roster

Student Name:

Figure 37
Section Editor

Courses

Section Editor

Teacher Search

Teacher Name:

Teacher

- [Crystal Mahottie](#)
- [Josh Maag](#)
- [Michael Lucas](#)

Figure 38
Teacher Search

5.8 Error Handling

Error handling techniques have been implemented throughout our system to ensure a pleasant user (and developer) experience. For example, when performing a system search, certain error handling techniques have been implemented in order to ease the users and help them find the information they seek. When using the system search, if the user enters in a value which does not return any results, a warning message box is displayed (Figure 39). To make sure that a user acknowledges the message presented, all other forms and tools displayed to the user are disabled and grayed out. This effect is performed by using a <DIV> tag stretched across the screen so the user cannot click on any other part of the system until they acknowledge the error message (Figure 40). Once the OK button has been pressed, the <DIV> is removed, and the page is now useable again. The search on the student form page has an error message which operates the same way, and states the student id entered is invalid (Figure 41).

If the project is running in debug mode, an exception catcher (Figure 42) also appears on the screen in a separate <DIV> tag. This way, developers can see feedback on which exception occurred, and where it appears in the code.

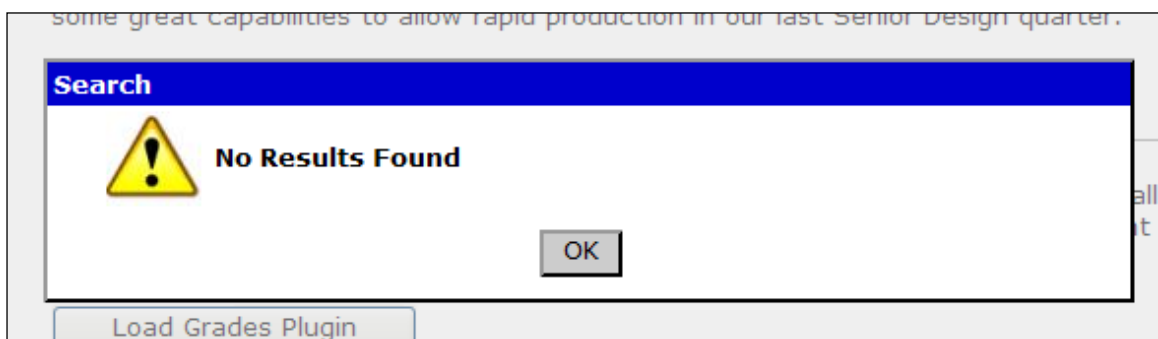


Figure 39
Search Error Message

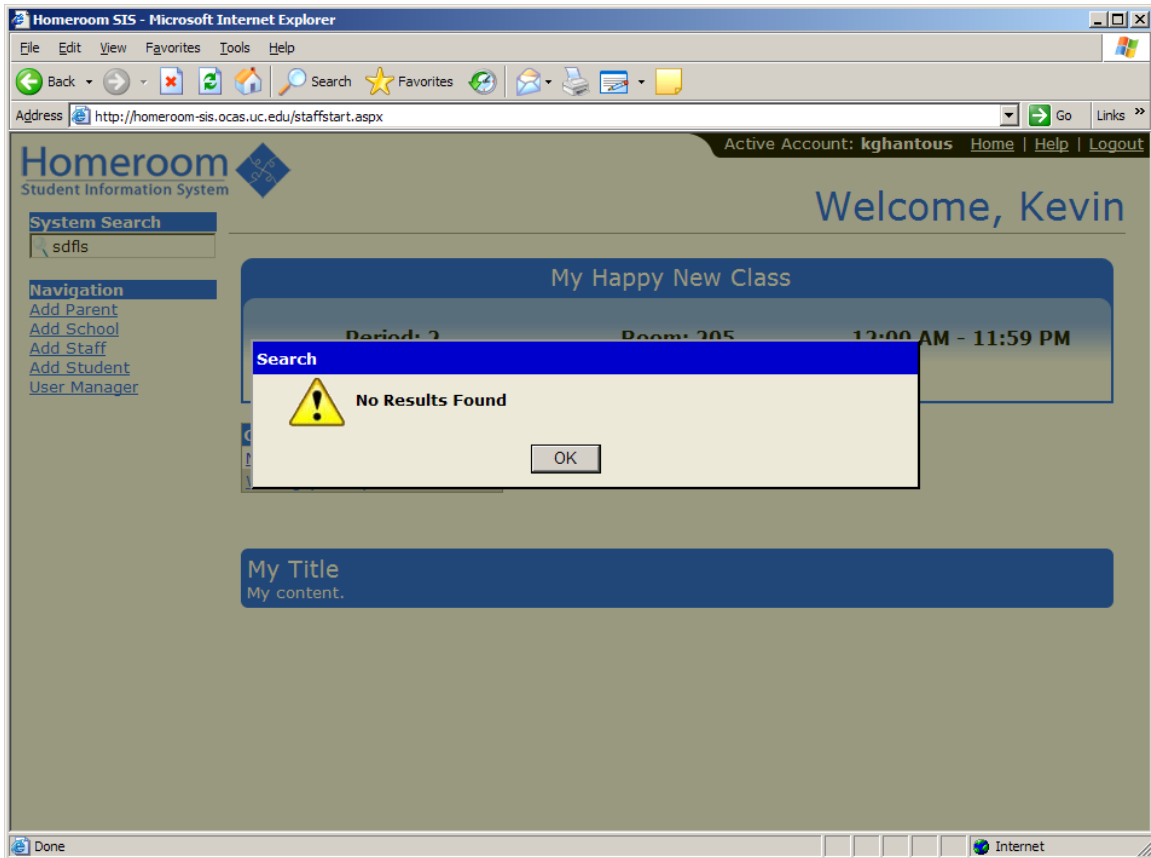


Figure 40
Error Screen

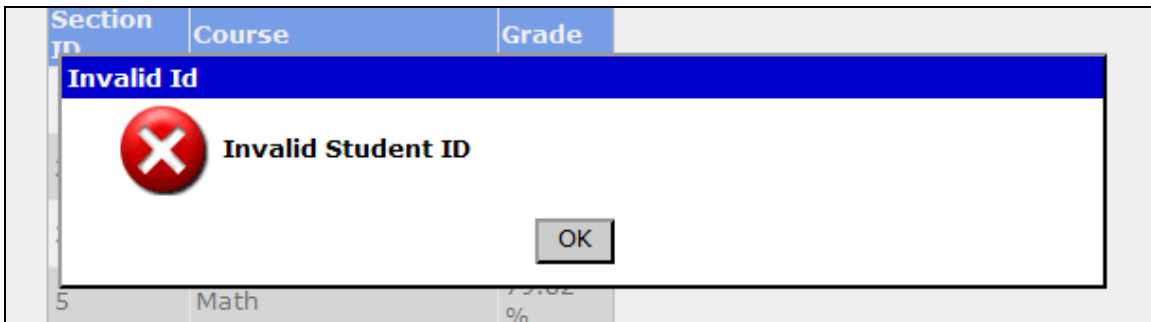


Figure 41
Student Lookup Error Message

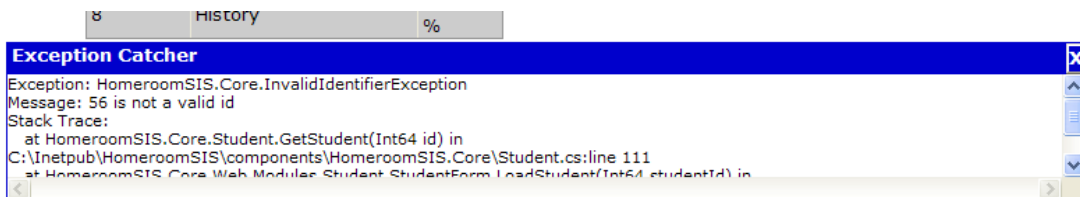


Figure 42
Exception Catcher

6 Testing

This section reviews our testing procedures during the project's construction.

6.1 Prototype Debugging

Homeroom SIS's testing during construction consisted of ensuring the program compiles at the end of the day. If a bug occurred which cannot be solved, the code was commented out or excluded from the project as a resource until it could be properly removed. Also, features were frequently checked to make certain that they did not interfere with the functionality of the system. Now that the project has entered beta and is ready for release, candidate testing is being done on a more thorough basis to verify use cases are flowing properly on more than just "happy" scenarios.

6.2 Planned Testing

Testing *Homeroom SIS* was done by people who will actually use the system, so several teachers and an administrator from different school districts were recruited to follow scenarios through the system. They were asked to go through a list of tasks to make sure that the application not only functions as it is supposed to, but that each task wasn't difficult to accomplish. The testers were asked to put a check, x, and/or exclamation point next to each task. The check verified they performed the task with no problems, the x stated they were unable to complete the task, and the exclamation point was used if the user had any comments to add to the task. The tasks cover student, staff, and administrator functions (Figure 43).

	Tester 1	Tester 2	Tester 3	Tester 4
Log in with given username and password	✓	✓	✓	✓
Create a student account and login	✓	✓	✓	✓
Create a course and section with tester as teacher	✗	✓	✓	✗
Use system search to find student and course	✓	✓	✗ !	✓
Assign the student and several other students to the previously made section	✗	✓	✓	✓
Insert assignments into course	✓	✓	✓	✓
Add grades to student account	✓ !	✓	✓	✓
Take attendance	✓ !	✓	✓	✗
Log out and log in as student	✓	✓	✗ !	✓
View class and grades	✓	✓	✓	✓

Figure 43
Testing Results

The table above shows the success rate of our project compared to each tester following the same scenarios. This testing provided valuable feedback which caused us to modify the project in several ways. Tester 1 fulfilled all of the tasks, but commented on a few. When taking attendance, Tester 1 noticed we used three colors for the present, tardy, and absent buttons which did not follow our color scheme. To keep the same visual appeal, we changed the three buttons to the same color which tied in with our color scheme. One problem Tester 1 found was when they tried to remove a grade from an assignment. They set the wrong grade to a student, and were unable to remove the grade. This was suppose to be a safety feature so a teacher could not remove a grade, only change it, but it proved to be detrimental to the system so it was removed.

Tester 3 was unable to view their new course after creating it. We discovered the full-text indexing happened once every minute, and changed the indexing on our tables. Now the database constantly scans for data changes and updates the index within ten seconds. Tester 3 was also unable to log in as their student account, but it was due to them not remembering their student's username after creating it. We asked them to use the User Management to find the username, and they were able to complete the task then.

7. Appendices

7.1 Appendix A

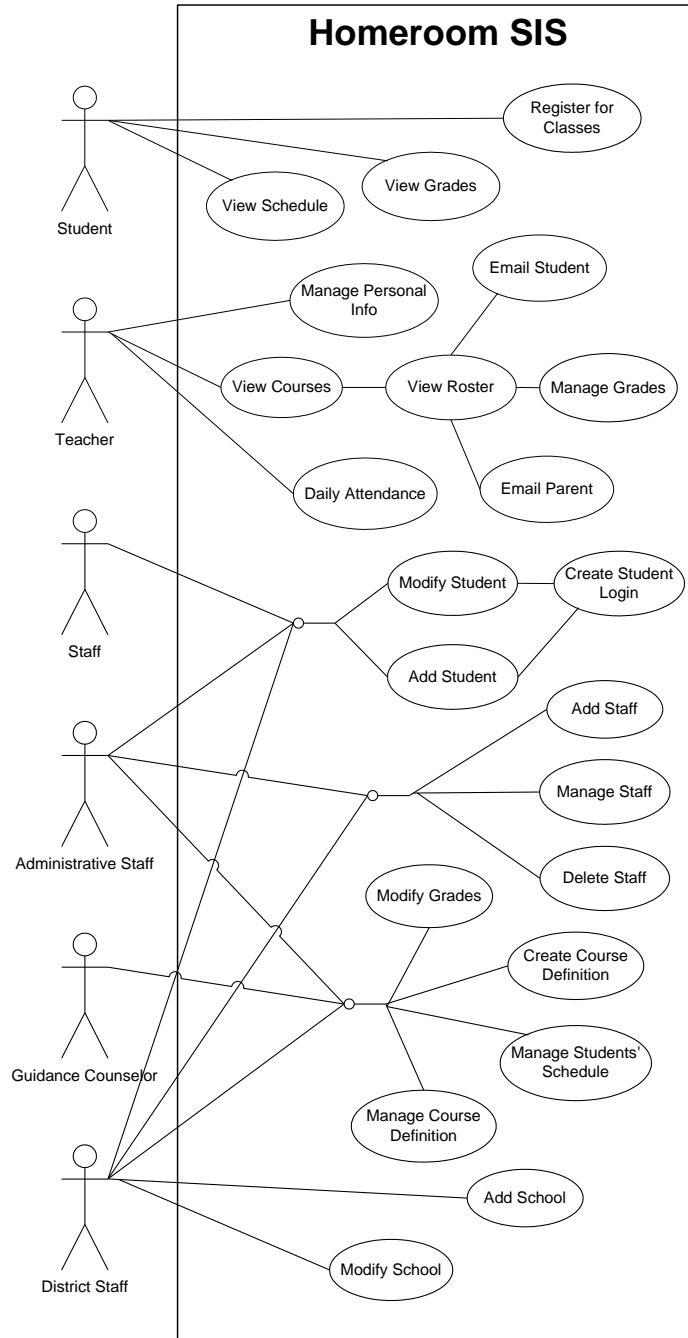


Figure 1
Use Case Diagram

7.2 Appendix B

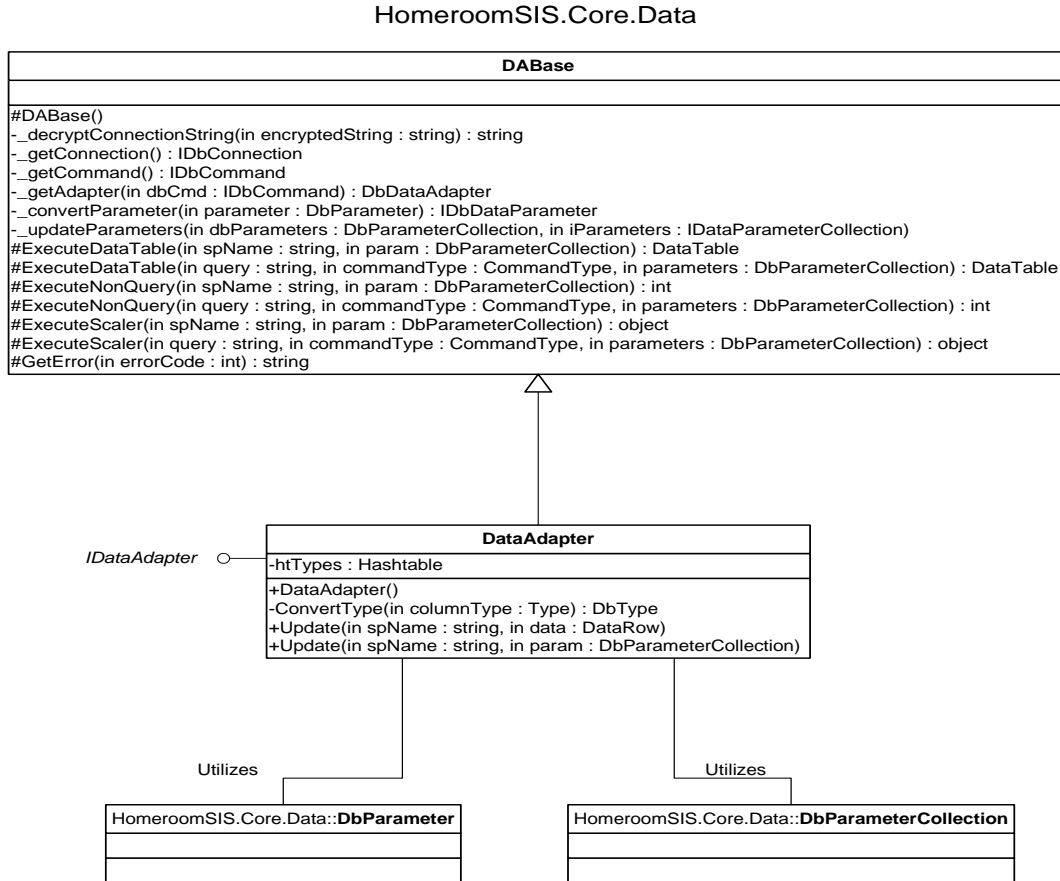


Figure 2
IDataAdapter Class Diagram

HomeroomSIS.Core

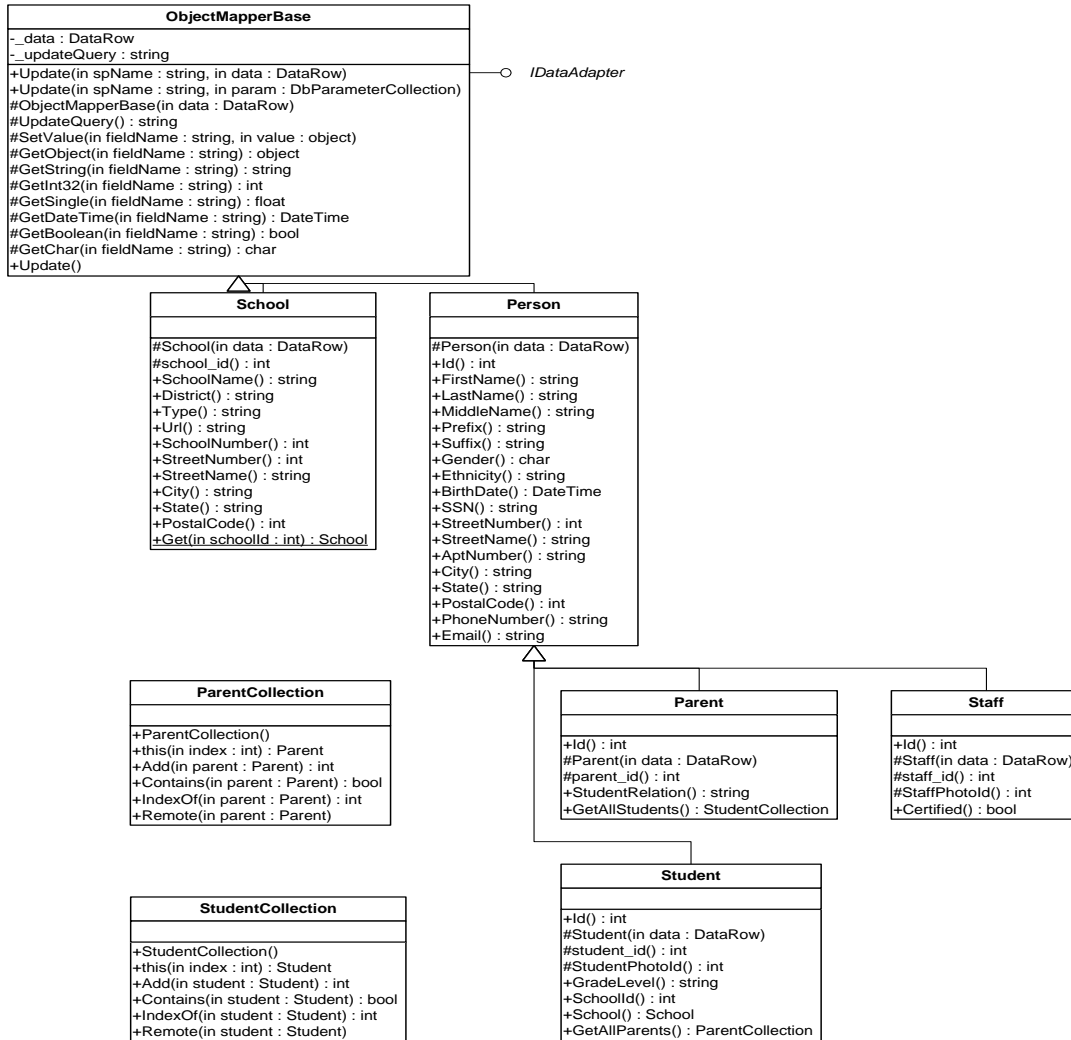


Figure 3
ObjectMappingBase Class

7.3 Appendix C

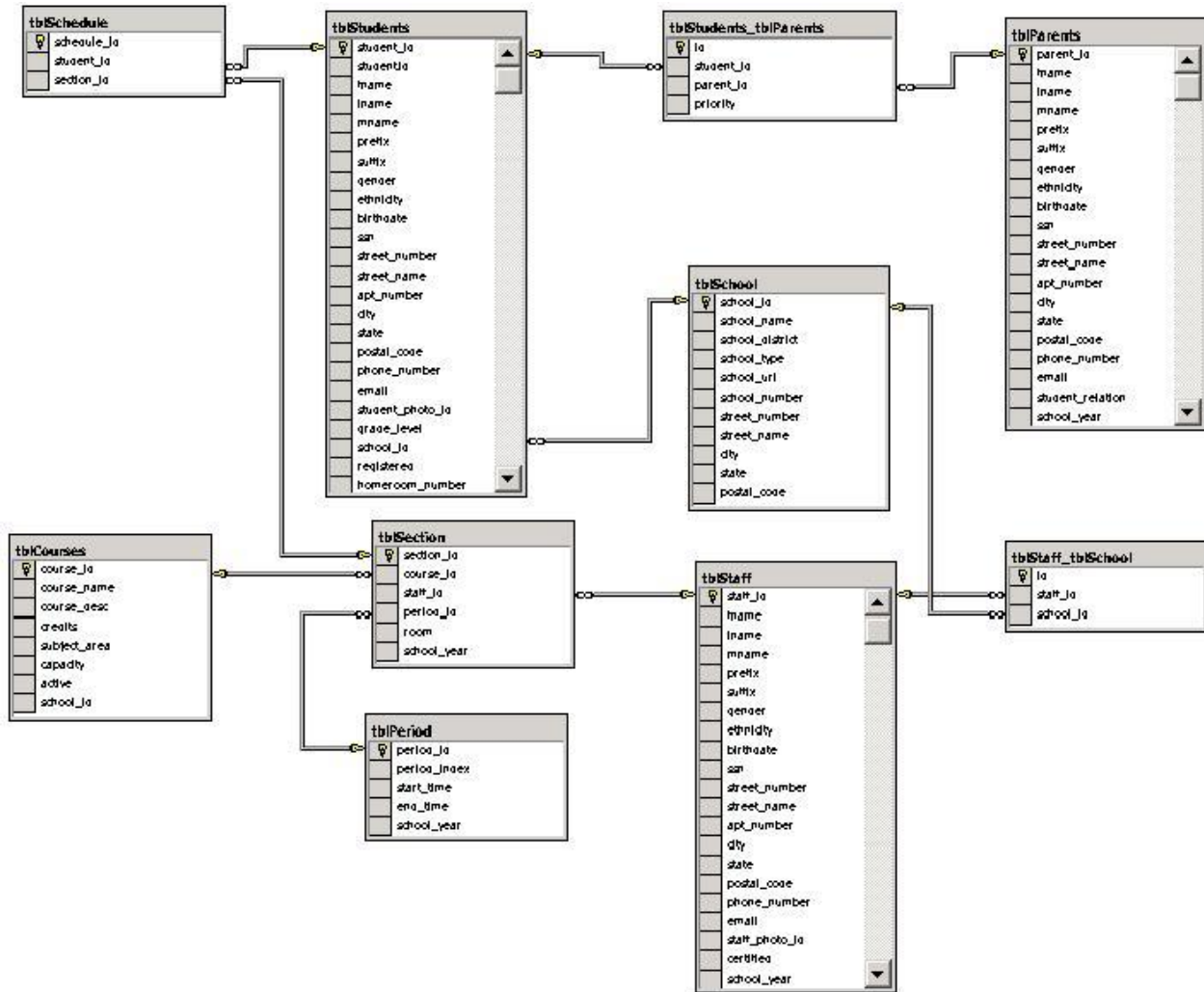


Figure 4
Database Diagram

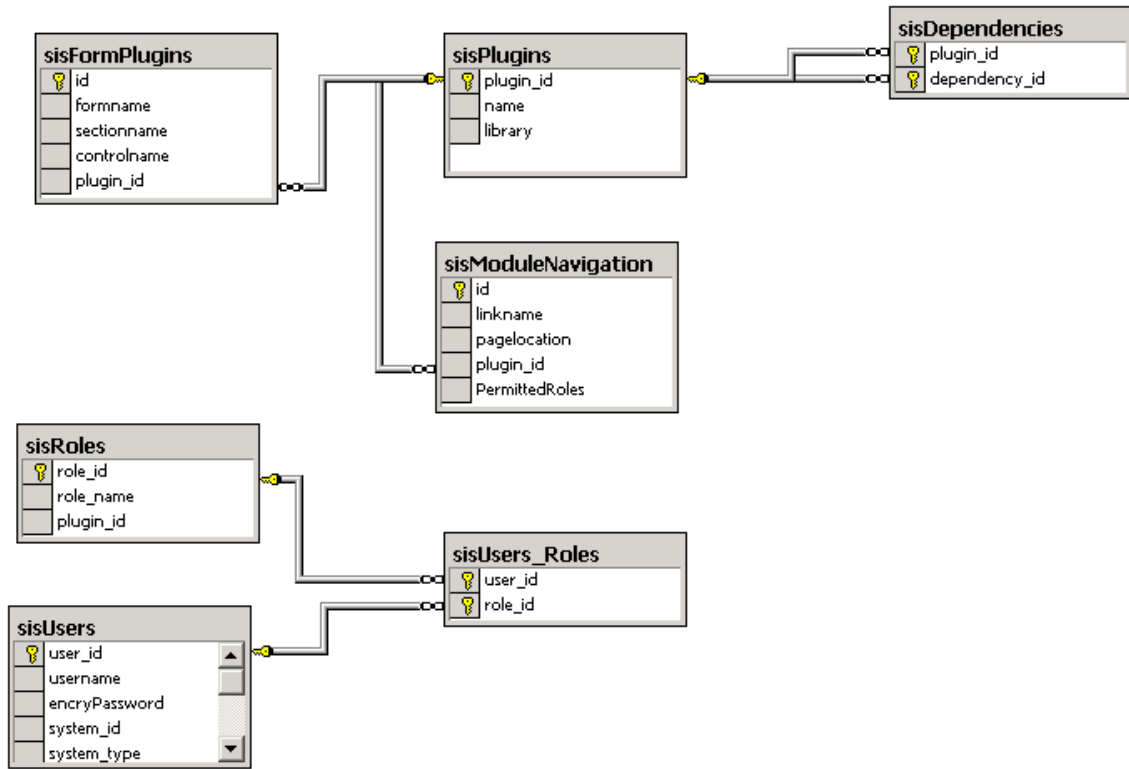


Figure 4.1
Core Database Architecture Diagram

tbl_plgGrades_Grades	
grades_id	
assignment_id	
student_id	
points_earned	

tbl_plgGrades_Assignments	
assignment_id	
section_id	
assignment_name	
points_possible	
assignment_type	
date_assigned	
date_due	

tbl_plgAttendance	
attendance_id	
section_id	
student_id	
status	
date_taken	

Figure 4.2
Grades/Assignments/Attendance Table Diagram

8. Reference

1. Said, Hazem. Personal Interviews. January 5, 2006 – March 2, 2006.
2. SASI. 2005. Student Information Series Jan. 2006.
<<http://www.pearsonschools.com/products/studentInfo/sasi.html>>.
3. Schools Interoperability Framework Association. 2006. Schools Interoperability Framework. Jan. 2005. <<http://www.sifinfo.org>>.