



System Design Manual

for

KADA REGISTRATION SYSTEM

Prepared by

KADA Consulting Group

**Andrew Holder
Anthony Gamba
Deokie Seerattan
Kadeem Quallis**

Systems Design and Implementation

**CIS 5910
Professor N. Gupta
December 28, 2011**

TEAM MEMBERS

KADA CONSULTING SERVICES		
Name	E-mail Address	Phone Number
Andrew Holder	andrewholder101@gmail.com	(631) 627-1648
Anthony Gamba	bufflucky5000@comcast.net	(302) 344-7610
Deokie Seerattan	seerattand@hotmail.com	(347) 386-2748
Kadeem Quallis	scholar516@gmail.com	(347) 203-6772

MEETING LOG

DATE	ATTENDANCE (YES/NO)				LOCATION	TIME	NOTES
	ANDREW	ANTHONY	DEOKIE	KADEEM			
9/27/2011	N/A	Y	Y	N/A	Classroom	6:30-8:10	Class cancelled.
9/29/2011	Y	N	Y	Y	Lab	6:30-8:10	FRD outline.
10/4/2011	Y	Y	Y	Y	Lab	6:30-8:10	FRD review and discussion.
10/6/2011	Y	Y	Y	Y	Lab	6:30-8:10	FRD, ER diagram review and discussion.
10/11/2011	Y	Y	Y	Y	Lab	6:30-8:10	ER diagram revision.
10/13/2011	Y	Y	Y	Y	Lab	6:30-8:10	ER diagram revision.
10/18/2011	Y	Y	Y	Y	Lab	6:30-8:10	ER diagram revision/.submission. Relational schema.
10/20/2011	Y	Y	Y	Y	Lab	6:30-8:10	ER diagram comparison to FRD.
10/25/2011	Y	N	Y	Y	Lab	6:30-8:10	System environment setup.
10/27/2011	Y	Y	Y	Y	Lab	6:30-8:10	System environment setup.
11/1/2011	Y	N	Y	Y	Lab	6:30-8:10	System environment setup.
11/3/2011	Y	Y	Y	Y	Lab	6:30-8:10	System environment setup.
11/8/2011	Y	Y	Y	Y	Lab	6:30-8:10	UI review. Anthony working from home.
11/10/2011	Y	Y	Y	Y	Lab	6:30-8:10	User Interface
11/15/2011	Y	Y	Y	Y	Lab	6:30-8:10	User Interface; PHP Server setup
11/17/2011	Y	Y	Y	Y	Lab	6:30-8:10	User Interface; PHP Server setup
11/22/2011	Y	Y	Y	Y	Lab	6:30-8:10	User Interface; coding (login); Anthony working from home.
11/29/2011	Y	Y	N	Y	Lab	6:30-8:10	UI, coding, Demo
12/1/2011	Y	Y	Y	Y	Lab	6:30-8:10	UI, coding, database fix up.
12/6/2011	Y	Y	Y	Y	Lab	6:30-8:10	UI, coding, User Guide. Anthony working from home on User's Guide.
12/8/2011	Y	Y	Y	Y	Lab	6:30-8:10	Coding, testing, documentation
12/13/2011	Y	Y	Y	Y	Lab	6:30-8:10	Coding, testing, documentation

DATE	ATTENDANCE (YES/NO)				LOCATION	TIME	NOTES
	ANDREW	ANTHONY	DEOKIE	KADEEM			
12/15/2011	Y	Y	Y	Y	Lab	6:30-8:10	Coding, testing, documentation
12/20/2011	Y	Y	Y	Y	Lab	1:00-8:10	Coding, testing, documentation
12/21/2011	Y	N/A	Y	N/A	Lab	4:00-5:30	Review instructor and admin modules; gather system design info.
12/22/2011	Y	Y	Y	Y	Lab	3:00	Coding, testing, documentation, and final presentation.

DOCUMENT INFORMATION AND APPROVALS

VERSION HISTORY			
<u>Version #</u>	<u>Date</u>	<u>Author</u>	<u>Comments</u>
1.0	12/1/2011	D. Seerattan	Initial draft of document.
2.0	12/22/2011	D. Seerattan	Updates include current development specifications and details as they are done and obtained.
3.0	12/26/2011- 12/28/2011	D. Seerattan	Add report specifications, student confirmation screens, and system process/output details.

DOCUMENT APPROVALS			
<u>Approver Name</u>	<u>Project Role</u>	<u>Signature/Electronic Approval</u>	<u>Date</u>
Dr. N. Gupta	Client/Professor		

RELATED DOCUMENTATION AND MATERIAL

The following table lists the documents that are directly related to this project and referenced during the system design phase. These documents are stored in Google Docs online service.

NAME OF DOCUMENT	VERSION	DATE	DESCRIPTION
Project Proposal	2.1	9/27/2011	Functional Summary of Registration System.
Functional Requirements Specification	2.0	10/6/2011	User functional requirements system must be able to satisfy.
Entity Relationship Diagram		10/26/2011	
Relational Schema Document		10/27/2011	
Summary of Functional Requirement Questions		10/20/2011	Questions ER diagram model should be able to answer.
Change Requests		12/22/2011	Changes made to user functional requirements.
Project Task List		12/22/2011	Log of all tasks related to this project.

ACRONYMS

The table below lists the terms and acronyms used in this document.

TERMS/ACRONYMS	DEFINITION
CRQ	Change Request – new requirement; not in original “FRD” approved for development and implementation.
CSS	Cascading Style Sheets
FRD	Functional Requirement Document
FRDS	Functional Requirement Document Section
GPA	Grade Point Average
INC	Incomplete
PRI	Primary Key
RAD	Rapid Application Development
S	Satisfactory
SDS	System Design Specification
TBD	To be determined
U	Unsatisfactory
UAT	User Acceptance Testing

TABLE OF CONTENTS

1.0	Purpose	4
1.1	Methodology	4
1.2	Document Organization.....	4
1.2.1	Overview	4
1.2.2	Application Architecture	4
1.2.3	Detailed Application Design.....	4
1.2.4	System Infrastructure Design	5
1.2.5	Technology	5
1.2.6	Appendix	5
2.0	Overview	6
3.0	Application Architecture	6
3.1	Shared Services.....	6
3.2	User Interface Overview.....	6
3.3	General Layout & Design	6
3.4	Screens	7
3.5	File Structure	8
4.0	Detailed Application Design.....	9
4.1	USER LOGIN.....	9
4.1.1	Login Screen	9
4.1.2	Screen Specification	9
4.1.3	Login Process.....	11
4.1.4	System Process/Output	11
4.1.5	Code Snippet.....	13
4.2	STUDENT MODULE.....	16
4.2.1	Student Services Navigational Menu.....	16
4.2.2	Main Screen	17
4.2.2.1	Layout & Design.....	17
4.2.2.2	Screen Specification.....	18
4.2.2.3	Menu Structure Specification	19
4.2.3	COURSE CATALOG MENU	20
4.2.3.1	Basic Search Screen	20
4.2.3.1.1	Screen Specification	20
4.2.3.1.2	System Process/Output.....	21
4.2.3.1.3	Code Snippet.....	23
4.2.4.1	Advanced Search Screen	24
4.2.4.1.1	Screen Specification	24
4.2.4.1.2	System Process/Output.....	25
4.2.5	REGISTRATION MENU.....	27
4.2.5.1	Lookup Classes Screen.....	27
4.2.5.1.1	Screen Specification	27
4.2.5.1.2	System Process/Output.....	28
4.2.6.1	Add/Drop a Course Screen	29
4.2.6.1.1	Screen Specification	30
4.2.6.1.2	System Process/Output.....	31
4.2.7	MY PROFILE MENU.....	33

4.2.7.1 View Personal Information Screen.....	33
4.2.7.1.1 Screen Specification	34
4.2.7.1.2 <i>System Process/Output</i>	35
4.2.7.2 Security Question Screen.....	36
4.2.7.2.1 Screen Specification	37
4.2.7.2.2 <i>System Process/Output</i>	37
4.2.7.3 Emergency Contact Screen.....	38
4.2.7.3.1 Screen Specification	39
4.2.7.3.2 <i>System Process/Output</i>	40
4.2.8 MY RECORD MENU	41
4.2.8.1 Grades Screen	41
4.2.8.1.1 Screen Specification	41
4.2.8.1.2 <i>System Process/Output</i>	42
4.2.8.2 Transcript Screen	43
4.2.8.2.1 Screen Specification	43
4.2.8.2.2 <i>System Process/Output</i>	43
4.2.8.3 Class Schedule Screen.....	44
4.2.8.3.1 Screen Specification	44
4.2.8.3.2 <i>System Process/Output</i>	45
4.3 INSTRUCTOR MODULE	47
4.3.1 <i>Navigational Menu Structure</i>	47
4.3.2 Menu Structure Specification	48
4.3.3 ADVISEMENT MENU.....	49
4.3.3.1 Search Course Catalog Screen	49
4.3.3.2 Assign Student PIN Screen.....	49
4.3.3.2.1 Screen Specification	49
4.3.3.2.2 <i>System Process/Output</i>	50
4.3.3.2.3 <i>Code Snippet</i>	51
4.3.4 GRADE MENU	52
4.3.4.1 Mid-Term Grades Screen	52
4.3.4.1.1 Screen Specification	52
4.3.4.1.2 <i>System Process/Output</i>	53
4.3.4.2 Final Grades Screen	54
4.3.4.2.1 Screen Specification	54
4.3.4.2.2 <i>System Process/Output</i>	55
4.3.4.2.3 <i>Code Snippet</i>	56
4.3.4.3 REPORT MENU	57
4.3.4.3.1 Class Roster Screen	57
4.3.4.3.2 Screen Specification	57
4.3.4.3.3 <i>System Process/Output</i>	57
4.4 ADMINISTRATIVE MODULE.....	59
4.4.1 <i>Navigational Menu Structure</i>	59
4.4.2 Menu Structure Specification	59
4.4.3 MANAGE COURSE MENU.....	61
4.4.3.1 Add Course Screen.....	61
4.4.3.1.1 Screen Specification	61
4.4.3.1.2 <i>System Process/Output</i>	62
4.4.3.2 Update/Cancel Course Screen	63
4.4.3.2.1 Screen Specification	63
4.4.3.2.2 <i>System Process/Output</i>	64
4.4.3.3 Remove Course Screen.....	64
4.4.3.3.1 Screen Specification	65
4.4.3.3.2 <i>System Output/Process</i>	65
4.4.3.4 Add Section Screen	66
4.4.3.4.1 Screen Specification	66

4.4.3.4.2 System Process/Output.....	67
4.4.3.5 Update/Cancel Section Screen	68
4.4.3.5.1 Screen Specification	68
4.4.3.5.2 System Process/Output.....	69
4.4.3.6 Assign Instructor Screen.....	69
4.4.3.6.1 Screen Specification	70
4.4.3.6.2 System Process/Output.....	71
4.4.3.7 Assign Timeslot Screen	71
4.4.3.7.1 Screen Specification	71
4.4.3.7.2 System Process/Output.....	72
4.4.8 REPORT MENU	73
4.4.8.1 Grade Point Average	73
4.4.8.1.1 Screen Specification	73
4.4.8.1.2 System Process/Output.....	74
4.4.8.2 Student Transcript.....	76
4.4.8.2.1 Screen Specification	76
4.4.8.2.2 System Process/Output.....	76
5.0 System Infrastructure Design	78
5.1 Entity Relationship Diagram.....	78
5.2.1 Entity Sets	79
5.2.2 Relationship Sets	80
5.3 Database	80
5.3.1 Tables	80
5.3.2 Database Objects	81
6.0 Technology.....	87
6.1 Server Hardware/Software Requirements	87
6.2 Client Hardware/Software Requirements.....	87
7.0 Appendix	88
7.1 Appendix A – Script Files.....	88
7.2 Appendix B - Change Request Form.....	90
7.3 Appendix C – File Structure	91

1.0 PURPOSE

The purpose of this document is to describe the design and implementation of the KADA Registration System. This document translates the “User Functional Requirements” described in the FRD into the technical specifications that defines the solution of the system.

This document details the design of the functional requirements that are deemed in scope for development and implementation as described in the FRD. The design consists of three integrated modules (Student, Instructor, and Administrative) that facilitate the core user functions and business operations of the system.

1.1 Methodology

The RAD (Rapid Application Development) methodology is used to define the specifications of the system. Prototype of screens is used throughout this document to provide a visual depiction of the specifications and to provide clarity of the design.

1.2 Document Organization

This document is divided into the following main sections as described below. Some sections are further divided into sub-sections according to business functions or roles.

1.2.1 Overview

This section states the purpose or primary role of the KADA Registration System within the Institute’s organization, and the user and business functions it is expected to facilitate.

1.2.2 Application Architecture

This section provides a general overview of the application architecture. It includes information about shared or common services, and elements considered in the design phase to create the general layout and design of the system to ensure consistency across all modules.

1.2.3 Detailed Application Design

This section defines the technical specifications of the application. The details are organized by modules (Student, Instructor, and Administrative) to provide a clear and consistent flow of the design and operation of the module, each service within the module, and the operation of the service from the point of access or login and user service selection, to user input or operation, to the corresponding system processing of specified business rules and expected output.

Special design consideration has been implemented to reduce human errors and to maintain accuracy of data by using preset field values and dynamically building field selection values based on specific conditions being met.

In addition, actual prototypes or screens are used in this section to define the specifications related to the specific user functional requirements.

1.2.4 System Infrastructure Design

This section defines the layout and design of the backend data structure. It includes details about the database foundation that include data entities and their relationships. It provides a detailed view of the database objects (tables), fields, and attributes.

1.2.5 Technology

This section describes the technologies that make up the system environment and the components required to access and use the system.

1.2.6 Appendix

This section includes supplemental information.

2.0 OVERVIEW

The K.A.D.A Institute of Technology Registration System is an online web-based system. This is a self-service system that enables users to quickly, easily, and conveniently access and use the services of the Office of the Registrar in a safe and secure manner. This system addresses student registration needs, and administrative and business operational needs of the organization.

3.0 APPLICATION ARCHITECTURE

The application design consists of three integrated modules, Student, Instructor, and Administrative to facilitate the functional requirements of each role. The design allows for low maintenance, enforcement of data integrity, and sharing of common services across multiple roles in a single system environment.

Access to each module is dependent upon the user login and authentication process described in the “[User Login](#)” section below.

3.1 *Shared Services*

The following services are shared among multiple roles.

- Search Course Catalog
- Class Roster
- Transcript

3.2 *User Interface Overview*

The user interface design provides a consistent look and feel throughout the entire application. The user interface consists of a combination of HTML and PHP format webpages, designed primarily in Dreamweaver CS5.

In addition, cascading style sheets are used to create and maintain a consistent look and feel throughout the system.

3.3 *General Layout & Design*

The following items were taken into consideration to create a user-friendly application experience and to provide a safe and secure system environment.

- Single login screen to allow users to access the system.
- Control access to user interface by “User Type” input.
- Check and validate user input and credentials (User Type, Email Address, and Password).

- Customize user interface according to business roles and functions (Student console, Instructor console, Administrative staff console).
- Menu structure for quick and easy navigation to user functions from any screen within the interface.
- Standard look and feel of all webpages using frame layout and external style sheets.
- Identify authenticated user by displaying user ID and email address at the top of each page.
- Allow for the display of static and dynamic content. Static menu structure allow for the menu to be visible from anywhere in the system.

3.4 Screens

The following table lists the user interface screens that are defined in the detailed application design section.

Screen Name	Filename	Module
Login	Index.php	Main screen or homepage.
Basic Search	Student_CourseCatalog_basic search.php	Student
Advanced Search	Student_CourseCatalog_advanced search.php	Student, Instructor
Lookup Classes	Student_Registration_LookupClasses.php	Student
Add/Drop Course	Student_Registration_AddDropCourse.php	Student
Personal Information	PersonalProfileForm.php	Student
Change Address	ChangeMyAddress.php	Student
Change Email	ChangeEmail.php	Student
Change Phone Number	ChangePhoneNumber.php	Student
Change Password	ChangePassword.php	Student
Change Security Question	ChangeSecurityQuestion.php	Student
Emergency Contact	Emergency Contact.php	Student
View Grades	Student_My Record_Viewgrade.php	Student
View Class Schedule	Student_My Record_Viewclasses.php	Student
View Transcript	Student_My Record_Viewtranscript.php	Student
Assign Student PIN	Pin.php	Instructor

Screen Name	Filename	Module
Enter Mid-Term Grade	EnterMidTermGrade.php	Instructor
Enter Final Grade	EnterFinalGrade.php	Instructor
View Class Roster	instructor_viewClassRoster.php	Instructor
Add Course	Admin_addCourse.php	Administrative
Update/Cancel Course	Admin_updateCourse.php	Administrative
Remove Course	deletecourse.php	Administrative
Add Section	Admin_addSection.php	Administrative
Cancel/Delete Section	Admin_canceldelete_section.php	Administrative
Assign Instructor	Admin_assigninstructor.php	Administrative
Assign Timeslot	Admin_assigntimeslot.php	Administrative
Course Grade Point Average	Admin_courseGPA.php	Administrative
Student GPA	Admin_StudentGPA.php	Administrative
Student Transcript	Admin_StudentTranscript.php	Administrative, Student

3.5 File Structure

Refer to [Appendix C](#) for a view of the KADA site root folder and sub folders that contain the screen files, scripts, images, and style sheets.

4.0 DETAILED APPLICATION DESIGN

This section defines the technical specifications of the front-end or user application. The details are organized by modules (Student, Instructor, and Administrative) to provide a clear and consistent flow of the design and operation of the module, each service within the module, and the operation of the service from user input, to system processing and output.

The details include screen information, screen specification, screen prototype, user input, system processing and output. It also include the details of the technical or programmatic approach described as “System Process/Output” that defines the constraints and validations that is implemented to meet the business rules associated with the specified functional requirement.

In addition, code snippets are included where applicable to demonstrate the process used to build and enforce specific business rules and user requirements.

4.1 USER LOGIN

This section supports functional requirement FRDS 2.3.1 and 2.3.2, user login and account security.

4.1.1 Login Screen

This is the main screen or entry point to the KADA Registration System that require users to input a user type, a valid email address, and password to access and use the services.

<i>File Name</i>	<i>Description</i>
<i>index.php</i>	KADA Registration system main screen. This screen is designed to allow all users to login and access the system.

4.1.2 Screen Specification

KADA Main Screen – Login						
Order	Object Name	Required/Input	Type	Size	List of Values	Comment
1	Logo	N	Image	700x200	N/A	KADA logo.
2	Registration System Login	N	Text	N/A	N/A	Header text.
3	User Type	Y	List	N/A	Administrative Assistant; Instructor; Student.	Input field.

KADA Main Screen – Login						
Order	Object Name	Required/Input	Type	Size	List of Values	Comment
4	Email Address	Y	Text	N/A	N/A	Input field.
5	Password	Y	Text	N/A	N/A	Input field.
6	Submit	N	Button	N/A	N/A	Submit form.
7	Reset	N	Button	N/A	N/A	Reset form field values.
8	Forgot Password	N	Text	N/A	N/A	Open form on click.
9	Help	N	Text	N/A	N/A	Open form on click.
10	Page Footer	N	Text	N/A	N/A	Kada Institute of Technology. All rights reserved 2001.

4.1.3 Login Process

The user login process specifications consist of input constraints and error checking and verification that are used to validate user credentials, and to control access to the system.

4.1.4 System Process/Output

The following events are executed during the login process.

1. Require users to input a “User Type” a valid “Email address” and “Password” to access the system, as shown in [figure 4.1](#) below.

Figure 4.1 – Login Screen



The screenshot displays the login interface for the KADA Institute of Technology. At the top, there is a banner image featuring the institute's logo (a green diamond with 'K.A.A.D.' inside) and a photograph of a large brick building. Below the banner, the text 'Institute of Technology' is visible. The main heading is 'REGISTRATION SYSTEM LOGIN'. The login form includes a 'User Type' dropdown menu with 'Select a User Type' as the placeholder, followed by input fields for 'Email Address' and 'Password'. Below these fields are 'Submit' and 'Reset' buttons. A link for 'Forgot Password | Help' is positioned below the buttons. The footer of the page reads 'Kada Institute of Technology. All Rights Reserved 2011'.

2. Encrypt password to enable safe and secure login.
3. Checks to ensure that all required fields are populated on submit before proceeding to the next step.
4. On failure, output user friendly alerts (messages) to inform user what value(s) are missing, as shown [in figure 4.1.1](#) below.

Figure 4.1.1- Password Alert

The screenshot shows the login interface of the KADA Institute of Technology. At the top is a banner with the institute's logo and name. Below the banner is the title "REGISTRATION SYSTEM LOGIN". The login form includes fields for "User Type" (a dropdown menu set to "Student"), "Email Address" (containing "scholar516@gmail.com"), and "Password". There are "Submit" and "Reset" buttons. A red-bordered box contains the message "* Please Enter Password". Below this is a link for "Forgot Password | Help". The footer states "Kada Institute of Technology. All Rights Reserved 2011".

Submit Action:

5. Check the “user type” to determine which database table to validate the user login credentials (email address and password) against.
6. Validate the user email address and password against the related database table (Student, Staff) based on the “User Type” that is entered.
7. On failure, error checking process displays a user alert (message); that is if the user credentials (email address or password, or both) do not match the user record in the database as shown in [figure 4.1.2](#) below.

Figure 4.1.2 - Login Error

The screenshot shows the login error page. It features the same banner as Figure 4.1.1. Below the banner, the title "LOGIN ERROR" is displayed. The message "The email address or password is invalid. Please try again." is shown, followed by a link "Click here to login again.". Below this is a link for "Forgot Password | Help". The footer states "Kada Institute of Technology. All Rights Reserved 2011".

8. On successful login, the system authentication process uses the “User Type” value to grant the user access to the correct application module. That is to load the user main interface screen in the web browser.

4.1.5 Code Snippet

The following JavaScript code checks and verifies that all required fields are populated during the login process.

Refer to [Appendix A](#) for a complete list of code files.

Filename: Login.js

JAVASCRIPT

```
//Checks if values have not been selected/entered in all fields
function validateSelections()
{
    var dropdown = document.getElementById("UserType");
    var error = document.getElementById("errorLine");
    error.style.color = "red";

    if(dropdown.value == "Select a User Type")
    {
        error.innerHTML = "* Please Select A User Type";
        return false;
    }

    if(document.getElementById("EmailAddress").value=="")
    {
        error.innerHTML = "* Please Enter Email Address";
        return false;
    }

    if(document.getElementById("Password").value=="")
    {
        error.innerHTML = "* Please Enter Password";
        return false;
    }
    return true;
}
```

The following PHP code check and verifies the “User Type” value, in this case “Student” that is used to determine which user interface or module to grant the user access to upon successful login.

Filename:Functions.php

PHP

```
$result = mysql_query("SELECT * FROM Student WHERE email='$email' ");

$num_results = mysql_num_rows($result);
if($num_results == 0)
{
    header('Location:LoginErrorPage.php'); exit();
}

while( $row = mysql_fetch_assoc($result) )
{
    $db_password = $row['password'];
    if($passwd == $db_password
    $found = TRUE;
    else
    $found = FALSE;

    if($found == TRUE)
    {
        $_SESSION['userType'] = $userType;
        $_SESSION['id'] = $row['stuId'];
        $_SESSION['fName'] = $row['fName'];
        $_SESSION['mName'] = $row['mName'];
        $_SESSION['lName'] = $row['lName'];
        $_SESSION['birthDate'] = $row['birthDate'];
        $_SESSION['sex'] = $row['sex'];
        $_SESSION['homePhone'] = $row['homePhone'];
        $_SESSION['cellPhone'] = $row['cellPhone'];
        $_SESSION['email'] = $row['email'];
        $_SESSION['password'] = $row['password'];
        $_SESSION['address'] = array( "street"=>$row['street'], "city"=>$row['city'],
        "state"=>$row['state'], "zip"=>$row['zip'] );
        $_SESSION['commonFees'] = $row['commonFees'];
        $_SESSION['securityQuestion'] = $row['securityQuestion'];
        $_SESSION['securityAnswer'] = $row['securityAnswer'];

        $result2 = mysql_query("SELECT * FROM emergencycontact WHERE stuId = '" . $row['stuId']
        . "'");
        while( $row2 = mysql_fetch_assoc($result2) )
        {
            $_SESSION['emContact'] = array("fName"=>$row2['fName'], "mName"=>$row2['mName'],
            "lName"=>$row2['lName'],
            "relationship"=>$row2['relationship'], "street"=>$row2['street'], "city"=>$row2['city'],
```

```
"state"=>$row2['state'], "street"=>$row2['street'], "zip"=>$row2['zip'], "email"=>$row2['email'],  
"primaryPhone"=>$row2['primaryPhone'], "secondaryPhone"=>$row2['secondaryPhone'] );  
  
}  
header('Location:StudentConsole.php');  
exit();  
}  
else  
{  
header('Location:LoginErrorPage.php'); exit();
```

4.2 STUDENT MODULE

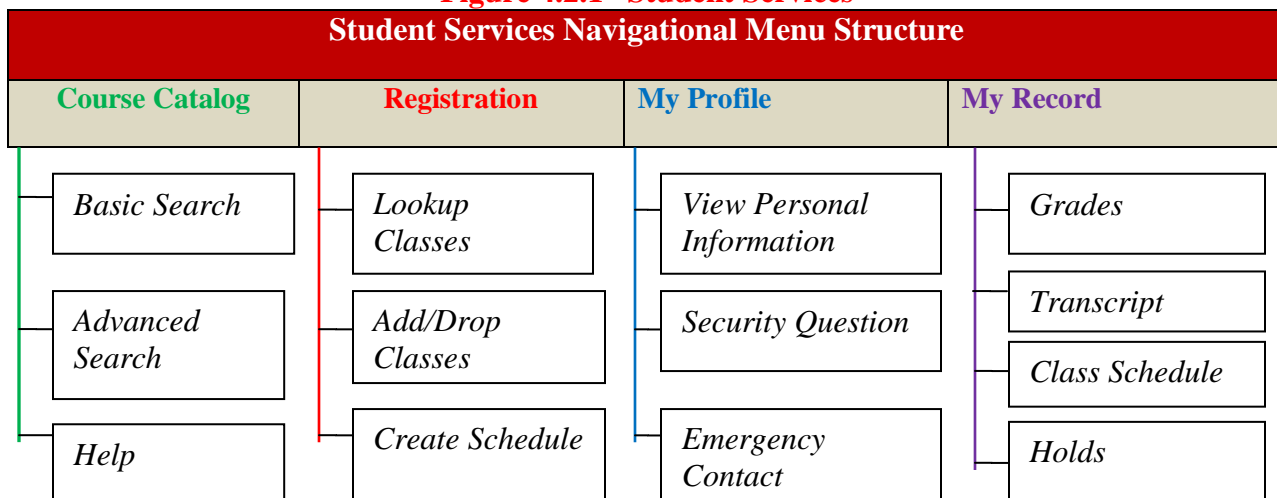
This module supports “Student Self-Services” functional requirements described in FRDS-2.1.

- FRDS 2.1.1 - Students will be able to quickly and easily access the student interface of the Registration System from the K.A.D.A Institute of Technology website to search and register for a course.
- FRDS-2.1.2 - Students will be able to view and make changes to their personal profile from the staff interface. Clicking the “My Profile” link will allow the student to view his/her information.
- FRDS – 2.1.3 - Students will be able to access and view their academic record from the staff interface. Clicking the “My Record” link will allow the student to access the following information. The student will be able to change his/her current class schedule only.

4.2.1 Student Services Navigational Menu

The following diagram defines the specification for the student services navigational menu structure. Note, menu items deemed out of scope due to time constraints or as project phase 3 may not be developed and implemented and is noted as such in the system process/output section.

Figure 4.2.1– Student Services



4.2.2 Main Screen

The student main screen design and layout consists of three sections. Frames are used to divide each section. Each section is represented by an individual screen.

The following screens make up the main Student Services screen.

Screen ID	Name	Description
<i>Studentconsole.php</i>	Student Console.	Main student interface screen that loads upon successful student login and system authentication.
<i>Stu_Console_header.php</i>	Student Console Header.	Header section of the Student Console (main) screen.
<i>Stu_Console_sidebar.php</i>	Student Console Side Bar.	Left column section of the Student Console (main) screen.
<i>Stu_Console_content.php</i>	Student Console Content.	Content section of the Student Console screen that displays current/active menu selection.

4.2.2.1 Layout & Design

- Screen provides a single point of access for student services.
- Access to this screen requires the student to successfully login to the registration system. That is to input his/her username and password at the login screen, and for the system to validate the student input and to authenticate the login.
- Screen is designed to provide a quick and easy navigational menu structure that is always visible from anywhere within the interface so that students can move from one screen or service to another with one click.
- Screen consists of three sections; a header, left column, and body or content section. Each section is maintained in a separate file.
- Screen consists of static and dynamic content. The header and left column sections are static.
- The content section is dynamic. The content of this section changes to reflect the menu option that is selected by the student.

4.2.2.2 Screen Specification

Student Interface Main Screen – Header Section						
Order	Object Name	Required/Input	Type	Size	List of Values	Comment
1	Welcome	Y	Text	40	N/A	Value retrieved from student record in the Student table upon login
2	Student ID	Y	Text	15	N/A	Value retrieved from student record in the Student table upon login.
3	Logout	N	Button	N/A	N/A	On click, closes the active screen and exit the program.
4	Home	N	Button	N/A	N/A	On click, loads the Student Services main page.
5	Menu Bar	N	Menu	N/A	N/A	Navigate the student services console options.
6	Student Services Logo	N	Image	700x200	N/A	

Figure 4.2 Main Screen – Header Section



4.2.2.3 Menu Structure Specification

Menu Name	Item	Action
Course Catalog	<i>Basic Search</i>	Open the basic search screen and allow students to search the course catalog.
Course Catalog	<i>Advanced Search</i>	Open the advanced search screen and allow students to search the course catalog.
Course Catalog	<i>Help</i>	Phase 3 – not developed.
Registration	<i>Lookup Classes</i>	Open the lookup class screen and allow students to lookup classes to register for in a given semester and year.
Registration	<i>Add/Drop a Course</i>	Open the add/drop course screen and allow students to register for a course or to drop an existing course.
Registration	<i>Create Schedule</i>	Phase 3 – out of scope.
My Profile	<i>Personal Information</i>	Open the personal information screen and display logged in user personal information. Also, allow students to update personal information
My Profile	<i>Security Question</i>	Open the security question screen and allow students to change their security question.
My Profile	<i>Emergency Contact</i>	Open the emergency contact screen and allow students to update their emergency contact information.
My Record	<i>Grades</i>	Open the grades screen and allow student to view their mid-term or final grades for a given class in a given semester and year.
My Record	<i>Transcript</i>	Allow students to generate their academic transcript.
My Record	<i>Class Schedule</i>	Open the class schedule screen and allow students to lookup their current schedule.
My Record	<i>Holds</i>	Time constraint.

Figure 4.2.1 Student Services Main Screen

4.2.3 COURSE CATALOG MENU

This menu defines the specifications for the functional requirements defined in the FRD to search the course catalog.

4.2.3.1 Basic Search Screen

Screen Name	File Name	Menu Access	Description
Basic Search	Student_CourseCatalog_basic search.php	Course Catalog>Basic Search	Search course catalog by semester, year, subject

4.2.3.1.1 Screen Specification

Basic Search Screen					
Object Name	Required	Type	Size	List of Values	Comment
Semester	Y	List	158	Fall Spring Summer Winter	Selection list.
Year	Y	List	158	2012 2011 2010	Selection List.

Basic Search Screen					
Object Name	Required	Type	Size	List of Values	Comment
Subject	Y	Button	158	All Business Computer Science Information Technology Management Information Systems Mathematics	Selection List.
Search	N	Button			Submit form.
Reset	N	Button			Reset form field values.

Figure 4.2.2 - Basic Search Screen

4.2.3.1.2 System Process/Output

The following events are executed during the search course catalog process.

1. Require users to input a “Semester” and checks to ensure that a valid entry was made.
2. Output user alert (message) on screen if semester value is not valid as shown in [code snippet](#) below.
3. On Search, dynamically builds and execute query against the related tables in the database.

4. If no match is found, return message “No classes were found that meet your search criteria”.
5. On success, output search results generated by [code snippet](#) below on screen; results shown in [figure 4.2.3](#).

Figure 4.2.3 Basic Search Output

Course ID	Course Name	Description
BS100	Intro. to Business Management	An introductory course to business management topics.
BS200	Accounting I	Accounting Principles I
BS300	Accounting II	Accounting Principles II
CS100	Java Programming I	An introduction to object oriented programming in Java.
CS110	Intro. to Computer Applications	Learn Microsoft Office Suite
CS200	Java Programming II	Advanced object oriented programming in Java.
CS400	Database Systems	Learn database principles and concepts such as ACID, Normalization
CS410	Network Administration	Learn to administer Windows and Linux based networks
MA100	Pre-Calc	Foundation principles before entering Calculus
MA200	Calculus I	Calculus I and Analytic Geometry
MA300	Calculus II	Calculus II and Analytic Geometry
MA330	Statistics	Course teaches you about statistics

4.2.3.1.3 Code Snippet

The following code checks to verify that the “Semester” field value is populated with a valid entry.

JAVASCRIPT

```
<script type="text/javascript">
<!--
function checkSemester()
{
var semester = document.getElementById("Semester");
var errorMsg = document.getElementById("errorText");
errorMsg.style.color = "red";
if(semester.value == "Select a Semester")
{
errorMsg.innerHTML = "* Please select a Semester";
return false;
}

}
-->
</script>
```

The following code performs a basic search against the course catalog and returns the matching results.

PHP

```
$semester = $_POST['Semester'];
$year = $_POST['Year'];
$subject = $_POST['Subject'];

if($subject == "All")
{
$query = "SELECT DISTINCT section.courseId, courseName, courseDescription FROM
course, section WHERE course.courseId = section.courseId AND semester='$semester'
AND year='$year' ORDER BY section.courseId";
}
else
{
$query = "SELECT DISTINCT section.courseId, courseName, courseDescription FROM
course, section WHERE course.courseId = section.courseId AND semester='$semester'
AND year='$year' AND course.subject='$subject' ORDER BY section.courseId";
}
```

4.2.4.1 Advanced Search Screen

Screen Name	File Name	Menu Access	Description
Advanced Search	Student_CourseCatalog_advanced search.php	Course Catalog>Advanced Search	Search course catalog by semester, year, title, course number.

4.2.4.1.1 Screen Specification

Advanced Search Screen					
Object Name	Required	Type	Size	List of Values	Comment
Semester	Y	List	338	Fall Spring Summer Winter	Selection list.
Year		List	338	2012 2011 2010	Selection List.
Subject		Button	338	All Business Computer Science Information Technology Management Information Systems Mathematics	Selection List.
Title	N	Text	338		User Input.
Course Number	N	Text	50		User Input.
Search	N	Button			Submit form.
Reset	N	Button			Reset form field values.

Figure 4.2.4 - Advanced Search Screen

Welcome: Student ID: [Logout](#) [Home](#)

[Course Catalog](#) [Registration](#) [My Profile](#) [My Record](#)

Student Services

COURSE CATALOG > Advanced Search

Semester:

Year:

Subject:

Title:

Course Number:

4.2.4.1.2 System Process/Output

The following events are executed during the advanced search course catalog process.

1. Require users to enter a “Semester”.
2. Optional field input “Title, Course Number”.
3. On Search, dynamically build and execute query against the related tables in the database.
4. On no match found, return message “No classes were found that meet your search criteria.”
5. On success, output search results as shown in [figure 4.2.5](#) below.

Figure 4.2.5 Advanced Search Output

Year:	2011 ▼
Subject:	Business ▼
Title:	Accounting
Course Number:	BS40
<input type="button" value="Search"/> <input type="button" value="Reset"/>	

Course ID	Course Name	Description
BS400	Accounting Information Systems	An advanced course that further develops an understanding of the elements, relationships, and issues associated with manual and computerized accounting information systems.

4.2.5 REGISTRATION MENU

This menu defines the specifications for functional requirements described in the FRDS 2.1.1. The design specifications include FRDS 2.1.1.2, 2.1.1.4, and 2.1.1.5 to allow students to look up and register for classes in a given semester and year, and to add or drop a course for a given semester and year.

4.2.5.1 Lookup Classes Screen

Screen Name	File Name	Menu Access	Description
Lookup Classes	Student_Registration_Lookupclasses.php	Registraion>Lookup Classes	Search the course catalog for classes to register for in a given semester and year.

4.2.5.1.1 Screen Specification

Lookup Classes					
Object Name	Required	Type	Size	List of Values	Comment
Semester	Y	List	275	Select a Semester Fall Spring Summer Winter	Selection list.
Year		List	158	2012 2011 2010	Selection List.
Subject		List	158	All Business Computer Science Information Technology Management Information Systems Mathematics	Selection List.
Title		Text			User Input
Course Number		Text			Reset form field values.
Instructor		Text			User Input

Lookup Classes					
Object Name	Required	Type	Size	List of Values	Comment
Session		List		Day Evening Night	Selection List.
Submit		Button			Submit form.
Reset		Button			Reset values in form.

Figure 4.26 – Lookup Classes Screen

The screenshot shows a web form titled "REGISTRATION > Lookup Classes". The form contains the following elements:

- Semester:** A dropdown menu with "Spring" selected.
- Year:** A dropdown menu with "2012" selected.
- Subject:** A dropdown menu with "Computer Science" selected.
- Title:** A text input field.
- Course Number:** A text input field.
- Instructor (Last Name):** A text input field.
- Session:** A dropdown menu with "All" selected.
- Buttons:** "Submit" and "Reset" buttons at the bottom right.

4.2.5.1.2 System Process/Output

The following events are executed during the “lookup classes” process.

1. Require users to enter semester.
2. Default the year field to the current year the student can register for classes.
3. Dynamically build and run query against related tables in the database on submit action.
4. Output results in “Register for Courses” screen shown in [figure 4.2.7](#) below.

Figure 4.2.7 – Lookup Classes Output

Register for Courses

Select	Title	ID	Sec	Credits	Subject	Days	Time	Cap	Act	Rem	Instructor	Room/Bldg
<input type="checkbox"/>	Java Programming II	CS200	001	4	Computer Science	Monday, Wednesday	01:00:00 PM - 02:30:00 PM	20	20	20	Diane Napolitano	A300/Academic Village
<input type="checkbox"/>	Software Engineering	CS450	001	4	Computer Science	Monday, Wednesday	02:30:00 PM - 04:20:00 PM	20	20	20	Lili Hai	A200/Academic Village
<input type="checkbox"/>	Java Programming I	CS100	001	4	Computer Science	Tuesday, Thursday	01:00:00 PM - 02:30:00 PM	20	20	20	Glenn Dodd	A130/Academic Village
<input type="checkbox"/>	Internet and Web Tech	CS300	001	4	Computer Science	Tuesday, Thursday	02:50:00 PM - 04:20:00 PM	20	20	20	Lili Hai	A200/Academic Village
<input type="checkbox"/>	System Design	CS500	001	4	Computer Science	Tuesday, Thursday	06:30:00 PM - 08:00:00 PM	15	15	15	Naresh Gupta	A120/Academic Village
<input type="checkbox"/>	Java Programming II	CS200	002	4	Computer Science	Tuesday, Thursday	06:30:00 PM - 08:00:00 PM	20	20	20	Carlos Charles	A140/Academic Village
<input type="checkbox"/>	Database Systems	CS400	001	4	Computer Science	Tuesday, Thursday	08:20:00 PM - 09:50:00 PM	20	20	20	Naresh Gupta	A100/Academic Village

4.2.6.1 Add/Drop a Course Screen

Screen Name	File Name	Menu Access	Description
Add/Drop a Course	Student_Registration_AddDropCourse.php	Registration>Add/Drop course	Allow students to add a course for a given semester and year and to drop an existing course for a given semester and year.

4.2.6.1.1 Screen Specification

Add/Drop a Course					
Object Name	Required	Type	Size	List of Values	Comment
Semester	Y	List	338	Fall Spring Summer Winter	Selection list.
Year	Y	List	338	2012 2011 2010	Selection List.
Subject	Y	Button	338	All Business Computer Science Information Technology Management Information Systems Mathematics	Selection List.
Title	N	Text	338		User Input.
Course Number	N	Text	50		User Input.
Add Course		Button			Submit form.
Drop Course		Button			Submit form.
Reset		Button			Reset field values.

Figure 4.2.8 – Add/Drop Course Screen

REGISTRATION > Add/Drop Course

Semester:

Year:

Subject:

Title:

Course Number:

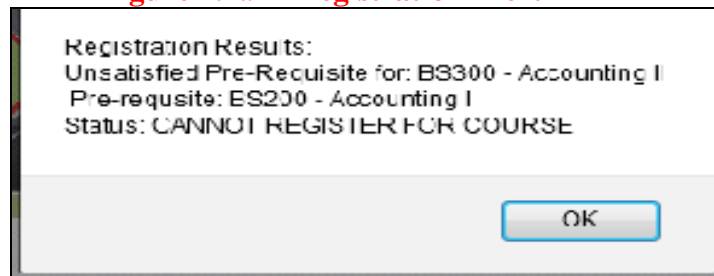
© 2011 KADA Institute of Technology
All Rights Reserved

4.2.6.1.2 System Process/Output

The following events are executed during the “Add/Drop Course” process.

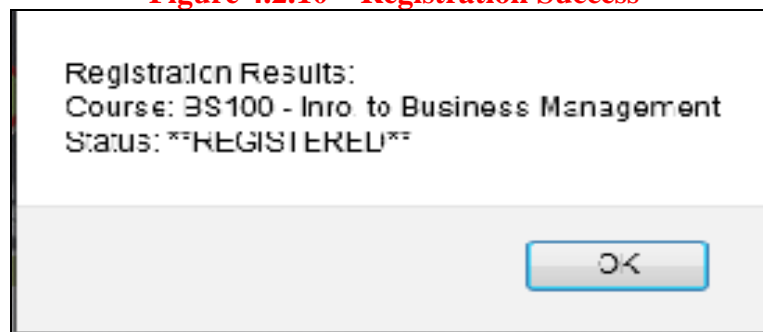
1. Check course prerequisite, course capacity, and maximum number of credits students can register for requirements.
2. Verify that student didn’t register for the same section in the same semester and year on “Register” action.
3. Output alert on screen if registration requirement is not met as shown in [figure 4.2.9](#) below.

Figure 4.2.9 – Registration Alert



4. Register student for course section and update student class schedule and transcript on success.
5. Output registration confirmation on screen as shown in [figure 4.2.10](#) below.

Figure 4.2.10 – Registration Success



6. Add/Drop course selection displays the student current class schedule as shown in [figure 4.2.11](#) below.

Figure 4.2.11 – Add/Drop Course

Select	Course	Course ID	Section	Semester	Year	Status
<input type="checkbox"/>	Java Programming II	CS200	001	Spring	2012	*Registered on Dec. 22, 2011*
<input type="checkbox"/>	Statistics	MA330	001	Spring	2012	*Registered on Dec. 22, 2011*

[COURSE SEARCH](#)

© 2011 KADA Institute of Technology
All Rights Reserved

7. Drop course action prompts the student to verify action, as shown in [figure 4.2.12](#) and removes the course from the student class schedule.

Figure 4.2.12 – Drop Course Confirmation

Are you sure you want to drop this course:
BS100 ?

Select	Course	Course ID	Section	Semester	Year	Status
<input type="checkbox"/>	Intro. to Business Management	BS100	002	Spring	2012	*Registered on Dec. 22, 2011*

4.2.7 MY PROFILE MENU

This menu defines the specifications for functional requirements as described in the FRS 2.1.2. The design specification enables students to view and edit their personal information.

4.2.7.1 View Personal Information Screen

The design specifications enable students to edit the following pieces of information by means of an “edit” link as shown [figure 4.2.13](#) below.

- Address
- Email
- Phone Number
- Password

Figure 4.2.13 - Edit Personal Information

Address:	141 Emerson Place, Valley Stream, NY 11580	Edit
Email:	scholar516@gmail.com	Edit
Phone Number:	Primary Phone: 516-599-2692 Secondary Phone: 347-203-6772	Edit
Password:	*****	Edit

Screen Name	File Name	Menu Access	Description
Personal Information	PersonalProfileForm.php	My Profile>View Personal Information	Allow students to view and update their personal information

4.2.7.1.1 Screen Specification

Personal Information					
Object Name	Required	Type	Size	List of Values	Comment
FirstName	Y	Text	50		User Input
MiddleName	Y	Text	50		User Input
LastName	Y	Text	50		User Input
Address	Y	Text	50		User Input.
Street	Y	Text	50		User Input.
City	Y	Text	50		User Input.
State	Y	Text	50		User Input.
Zip	Y	Text	50		User Input.
Email	Y	Text	50		User Input.
Primary Phone	Y	Text	50		User Input
Secondary Phone	N	Text	50		User Input.
Update		Button			Submit changes.
Reset		Button			Reset form field values.

Figure 4.2.12 - My Profile Menu

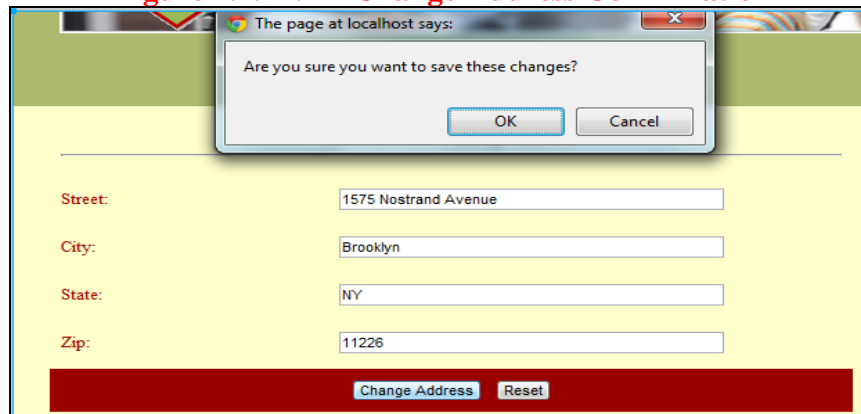


4.2.7.1.2 System Process/Output

The following events are executed during the edit personal profile process.

1. Restrict students from changing their name by setting the name fields attribute to “read only” on display.
2. Enable students to change their address, phone number, email address, and password.
3. The following forms are opened depending on edit action that is selected:
 - Change Address: **ChangeMyAddress.php**
 - Change Phone Number: **ChangePhoneNumber.php**
 - Change Email Address: **ChangeEmail.php**
 - Change Password: **ChangePhoneNumber.php**
4. Prompt student to confirm change before committing to the database as shown in figures [4.2.12:1](#), [4.2.12:2](#), [4.2.12:3](#) below.
5. On password change action, output alert on screen if password entries do not match as shown in [figure 4.2.12:4](#) below.

Figure 4.2.12:1 – Change Address Confirmation



The screenshot shows a web browser window displaying a form titled "The page at localhost says:". The form contains four input fields for address information: Street (1575 Nostrand Avenue), City (Brooklyn), State (NY), and Zip (11226). Below the fields are two buttons: "Change Address" and "Reset". A modal dialog box is open over the form, asking "Are you sure you want to save these changes?" with "OK" and "Cancel" buttons.

Figure 4.2.12:2 – Change Phone Number Confirmation

The page at localhost says:

Are you sure you want to save these changes?

OK Cancel

Primary Phone: 718-826-3758

Secondary Phone:

Change Phone Number Reset

Figure 4.2.12:3 – Change Email Address Confirmation

The page at localhost says:

Are you sure you want to save these changes?

OK Cancel

Current Email Address: scholar316@gmail.com

New Email Address: qualis_k@yahoo.com

Re-enter New Email Address: qualis_k@yahoo.com

Password: *****

Change Email Address Reset

Figure 4.2.12:4 – Change Password Error

MY PROFILE > Change Password

New Password: *****

Confirm New Password: *****

Current Password: *****

Change Password Reset

* Please check that your new password entries match and try again

4.2.7.2 Security Question Screen

Screen Name	File Name	Menu Access	Description
Security Question	ChangeSecurityQuestion.php	My Profile>Security Question.	Allow the logged in student to change his/her security question.

4.2.7.2.1 Screen Specification

Security Question					
Object Name	Required	Type	Size	List of Values	Comment
Current Question		Text			Retrieved from student record in the database on form open.
Answer		Text			Retrieved from student record in the database on form open.
Security Question	Y	List		What high school did you go to? What was the name of your first pet. Who is your favorite athlete.	Get list values.
Answer	Y	Text			User Input
Change Security Question		Button			Submit form.
Reset		Button			Reset field values.

Figure 4.2.13 – Change Security Question Screen

MY PROFILE > Change Security Question

Current Question: What was the name of your first pet?

Answer: Salmi

Security Question:

Answer:

4.2.7.2.2 System Process/Output

The following events are executed during the change security question process.

1. Display current security question and answer.

2. Require a valid security question value as shown in [figure 4.2.13:1](#) below.

Figure 4.2.13:1 – Change Security Question Error

MY PROFILE > Change Security Question

Current Question: What high school did you go to?

Answer: Leon M. Goldstein High School for the Sciences

Security Question: Select a question

Answer:

Change Security Question Reset

* Please select a security question

3. Require the answer field to be populated.
4. Prompt student to confirm change before committing to the database as shown in [figure 4.2.13:2](#) below.

Figure 4.2.13:2 – Change Security Question Confirmation

MY PROFILE > Change Security Question

Current Question: What high school did you go to?

Answer: Leon M. Goldstein High School for the Sciences

Security Question: What was the name of your first pet?

Answer: Salmi

Change Security Question Reset

4.2.7.3 Emergency Contact Screen

Screen Name	File Name	Menu Access	Description
Emergency Contact	Emergency Contact.php	My Profile>Emergency Contact	Allow students to view and update their emergency contact information.

4.2.7.3.1 Screen Specification

Emergency Contact					
Object Name	Required	Type	Size	List of Values	Comment
FirstName	Y	Text	50		User Input.
LastName	Y	Text	50		User Input.
MiddleName	Y	Text	50		User Input.
Relationship	Y	Text	50		User Input.
Address	Y	Text	50		User Input.
City	Y	Text	50		User Input.
State	Y	Text	50		User Input.
Zip	Y	Text	50		User Input.
Email	N	Text	50		User Input. Optional Field.
Primary Phone	Y	Text	50		
Secondary Phone	N	Text	50		User Input. Optional Field.
Update Emergency Contact		Button	50		Submit form.
Reset		Button	50		Reset form field values.

Figure 4.2.14 – Emergency Contact Screen

MY PROFILE > Emergency Contact

First Name:

Middle Name:

Last Name:

Relationship:

Address:

City:

State:

Email:

Zip:

Primary Phone:

Secondary Phone:

4.2.7.3.2 System Process/Output

The following events are executed during the change emergency contact process.

1. Require all fields except “Secondary Phone” number.
2. Output alert on screen if required fields are not populated as shown in [figure 4.2.14:1](#) below.

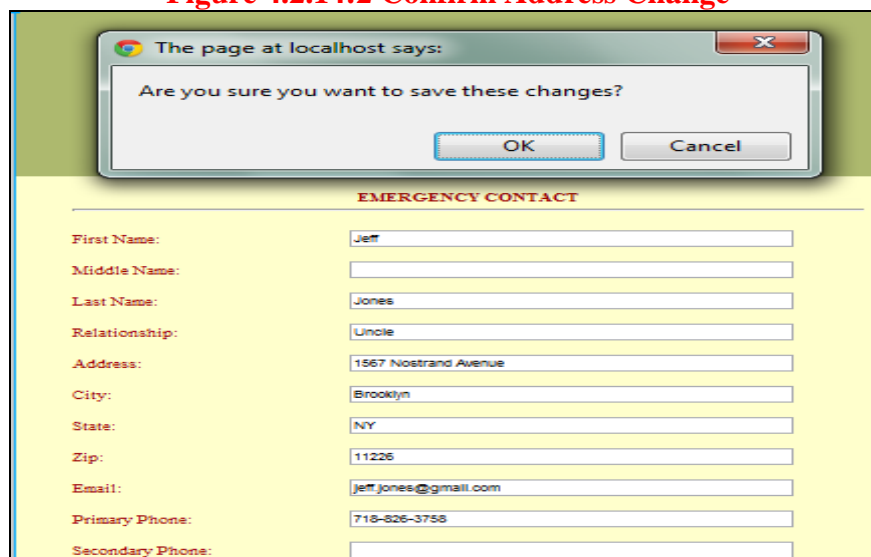
Figure 4.2.14:1 Change Address Error Message



The screenshot shows a web form titled "Change Address" with a yellow background. The form contains the following fields: Last Name (filled with "Jones"), Relationship (filled with "Uncle"), Address (empty), City (empty), State (empty), Zip (empty), Email (filled with "jeff.jones@yahoo.com"), Primary Phone (filled with "718-826-3758"), and Secondary Phone (empty). At the bottom, there are two buttons: "Update Emergency Contact" and "Reset". A red error message at the bottom of the form reads: "* Please enter address".

3. Prompt student to confirm change before committing to the database as shown in [figure 4.2.14:2](#) below.

Figure 4.2.14:2 Confirm Address Change



The screenshot shows a web browser window displaying a confirmation dialog box. The dialog box has a title bar that says "The page at localhost says:" and contains the text "Are you sure you want to save these changes?". There are two buttons: "OK" and "Cancel". Below the dialog box, the "EMERGENCY CONTACT" form is visible, showing the following fields: First Name (filled with "Jeff"), Middle Name (empty), Last Name (filled with "Jones"), Relationship (filled with "Uncle"), Address (filled with "1567 Nostrand Avenue"), City (filled with "Brooklyn"), State (filled with "NY"), Zip (filled with "11226"), Email (filled with "jeff.jones@gmail.com"), Primary Phone (filled with "718-826-3758"), and Secondary Phone (empty).

4.2.8 MY RECORD MENU

This menu design defines the specifications for functional requirements as described in FRDS 2.1.3. It provides students with the ability to view their mid-term and final grades in a given semester and year.

4.2.8.1 Grades Screen

Screen Name	File Name	Menu Access	Description
Grades	Student_MyRecord_Viewgrade.php	My Record>Grades	Allow students to view mid-term and final grades for a given semester and year.

4.2.7.1.1 Screen Specification

Grades					
Object Name	Required	Type	Size	List of Values	Comment
Semester	Y	List	N/A	Select a Semester Fall Spring Summer Winter	Selection List.
Year	Y	List	N/A	2010 2011 2012	Selection List.
GradeType	Y	Dialog	N/A	N/A	
Submit		Button	N/A	N/A	
Reset		Button	N/A	N/A	

Figure 4.2.15 – View Grades Screen

MY RECORD > View Grades

Semester:

Year:

Select Grade Type: ☐ Mid-term ☐ Final

© 2011 KADA Institute of Technology
All Rights Reserved

4.2.8.1.2 System Process/Output

The following events are executed during the view grade process.

1. Require semester, year, and grade type values.
2. Output alert on screen if all requirements are not met as shown in [figure 4.2.16](#).
3. Output result on screen as shown in [figure 4.2.17](#).

Figure 4.2.16 – View Grades Error Screen

Figure 4.2.17 – View Grades Output

Subject	Course Id	Course Name	Section	Final
Computer Science	CS100	Java Programming I	001	A
Mathematics	MA100	Pre-Calculus	001	B-

4.2.8.2 Transcript Screen

Screen Name	File Name	Menu Access	Description
View Transcript	Student_MyRecord_ViewStudentTranscript.php	My Record>Transcript	View academic transcript.

4.2.8.2.1 Screen Specification

Transcript					
Object Name	Required	Type	Size	List of Values	Comment
Transcripttype	Y	List		Select Transcript Type Student Copy	Selection List.
Generate Transcript		Button			
Reset		Button			

Figure 4.2.16 – View Transcript Screen

MY RECORD > View Transcript

Transcript Type:

© 2011 KADA Institute of Technology
All Rights Reserved

4.2.8.2.2 System Process/Output

The following events are executed during the view grade process.

1. Require students to select a transcript type.
2. Output alert on screen if requirement is not met as shown in [figure 4.2.16:1 below](#).

3. Output student transcript on screen as shown in [figure 4.2.16:2](#) below.

Figure 4.2.16:1 – View Transcript Error

Figure 4.2.16:2 – Student Transcript Output

Student I.D.	First Name	Last Name	Birth Date	
700544268	Kadeem	Quallis	1990-03-30	
Major	GPA			
Computer Information Science	3.35			
Official KADA Institute of Technology Transcript				
Term: Fall 2010				
Subject	Credits	Course	Title	Grade
Computer Science	4	CS100	Java Programming I	A
Mathematics	4	MA100	Pre-Calculus	B-

4.2.8.3 Class Schedule Screen

Screen Name	File Name	Menu Access	Description
View Class Schedule	Student_MyRecord_ViewClassSchedule.php	My Record>Class Schedule	View current class schedule.

4.2.8.3.1 Screen Specification

Class Schedule					
Object Name	Required	Type	Size	List of Values	Comment
Semester	Y	List		Select a Semester Fall Spring Summer Winter	Selection list.
Year	Y	List		2010 2011 2012	Selection List.

Class Schedule					
Object Name	Required	Type	Size	List of Values	Comment
Lookup Class Schedule		Button			
Reset		Button			

Figure 4.2.17 – Class Schedule Screen

MY RECORD > View Class Schedule

Semester:

Year:

© 2011 KADA Institute of Technology
All Rights Reserved

4.2.8.3.2 System Process/Output

The following events are executed during the view class schedule process.

1. Require semester and year values.
2. Output alert on screen if all requirements are not met as shown in [figure 4.2.17:1](#) below.
3. Output result on screen as shown in [figure 4.2.17:2](#) below.

Figure 4.2.17:1 – View Class Schedule Error

MY RECORD > View Class Schedule

Semester:

Year:

*** Please select a semester**

Figure 4.2.17:2 – Class Schedule Output

Semester:

Year:

Schedule - Spring 2012

Student ID: 700544268

Title	Course ID	Sec	Credits	Subject	Days	Time	Instructor	Room/Bldg
Intro. to Business Management	BS100	001	4.00	Business	Monday, Wednesday	06:30:00 PM - 08:00:00 PM	Elena Smirnova	C120/Campus Center
Statistics	MA330	001	4.00	Mathematics	Saturday	02:00:00 PM - 06:00:00 PM	Carlos Charles	S400/Natural Science
Java Programming II	CS200	002	4.00	Computer Science	Tuesday, Thursday	06:30:00 PM - 08:00:00 PM	Lili Hai	A140/Academic Village
Calculus I	MA200	002	4.00	Mathematics	Tuesday, Thursday	10:20:00 AM - 11:50:00 AM	Ricardo Valdez	A305/Academic Village

© 2011 KADA Institute of Technology

4.3 INSTRUCTOR MODULE

This module supports the business function requirements described in FRDS-2.2.

- Assign Student PIN
- FRDS 2.1.1.5 – Add mid-term and final grades for students that completed a course section.
- CRQ1 - Search course catalog.
- CRQ 2 - Generate class roster report.

4.3.1 Navigational Menu Structure

The following diagram represents the instructor services navigational menu structure.

Figure 4.3:1 – Instructor Services User Interface

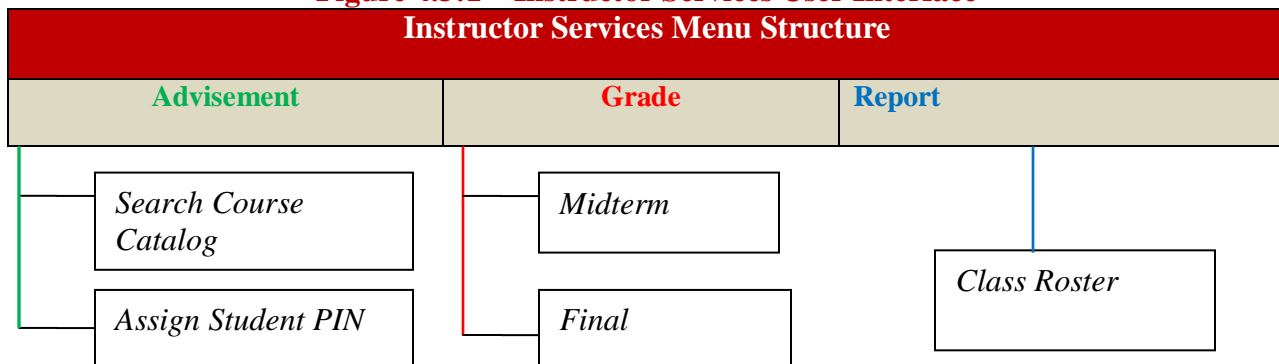


Figure 4.3:2 – Instructor Services Main Screen

4.3.2 Menu Structure Specification

Menu Name	Item	Action
Advisement	<i>Search Course Catalog</i>	Open the search course catalog form and allow instructors to search the course catalog.
Advisement	<i>Assign Student PIN</i>	Open the assign student pin form and allow instructors to assign a PIN to a student.
Grade	<i>Mid-term</i>	Open the mid-term grade form and allow instructors to enter student grade.
Grade	<i>Final</i>	Open the final grade form and allow instructors to enter student grade.
Report	<i>Class Roster</i>	Open the class roster form and allow instructor to generate report for a given section.

4.3.3 ADVISEMENT MENU

This menu option defines the specifications for the functional requirements as described in the FRDS 2.2.2.

4.3.3.1 Search Course Catalog Screen

This is a shared service that is accessible by the Student and Instructor roles, defined as “[Advanced Search](#)” in this document. Please refer to that section for specification details.

4.3.3.2 Assign Student PIN Screen

Screen Name	File Name	Menu Access	Description
Generate/View Student PIN	Pin.php	Advisement>Assign Student PIN	Enter student PIN.

4.3.3.2.1 Screen Specification

Assign Student PIN					
Object Name	Required	Type	Size	List of Values	Comment
Semester	Y	List	338	Select a Semester Fall Spring Summer Winter	Selection list; preset value.
Year	Y	List	338	Select a Year 2012	Selection List; preset value.
Student ID					Dynamic List.
PIN					Auto generated.
Submit	N	Button			Submit form.
Reset	N	Button			Reset form field values.

Figure 4.3:3 Student PIN Screen

GENERATE/VIEW STUDENT PIN

Semester:

Year:

Student ID:

PIN:

4.3.3.2.2 System Process/Output

The following events are executed during the assign student PIN process.

1. Require users to select a semester, year, and student ID.
2. Dynamically generates student ID field selection values based on the semester and year selected.
3. On Submit, search for student PIN; if PIN already exists for the given semester and year, display in PIN field.
4. If PIN doesn't exist, auto-generate new Pin and display on the screen as shown in [figure 4.3:4 below](#).

Figure 4.3:4 - Generate New PIN

GENERATE/VIEW STUDENT PIN

Semester:

Year:

Student ID:

PIN:

4.3.3.2.3 Code Snippet

Filename: getPin.php

```
<?php
/*
 * This script has the function of Getting a students pin number for a given semester. In the event that
 there is no pin
 for a student in a semester, the script will auto generate a student pin, which will then be stored in the
 db and displayed
 for the Instructor/Admin_Assistant to view.
 */
include './functions.php';

$q=$_GET["q"];
$q = explode("-", $q);
$semester = $q[0];
$year = $q[1];
$id = $q[2];

$sql="SELECT pin FROM studentpins WHERE stuId = ".$id." AND year = ".$year." AND
semester = ".$semester."";

$result = mysql_query($sql);
$num_results = mysql_num_rows($result);
if($num_results == 0){
    // Generate a new pin; This process can eventually be refined in the future
    $sql = "INSERT INTO studentpins (stuId, semester, year)
VALUES( '$id','$semester', '$year')";

    if (!mysql_query($sql,$con))
    {
        echo "No Pin - Unable to Generate Pin Error -" .mysql_error();
        //die('Error: ' . mysql_error());
    }
}
else{ // Display Generated Pin
    $sql="SELECT pin FROM studentpins WHERE stuId = ".$id." AND year = ".$year."
AND semester = ".$semester."";
    $result = mysql_query($sql);
    while($row = mysql_fetch_array($result))
    {
        echo $row[0];
    }
}
```

```

    }
} // End of Handling Generating a new pin
else{ // Display Pin
    while($row = mysql_fetch_array($result))
    {
        echo $row[0];
    }
}
?>

```

4.3.4 GRADE MENU

This menu option defines the specifications for the functional requirements as described in the FRDS 2.2.15.

4.3.4.1 Mid-Term Grades Screen

Screen Name	File Name	Menu Access	Description
Mid-term grade.	EnterMidTermGrade.php	Grade>Mid-term	Enter student mid-term grade.

4.3.4.1.1 Screen Specification

MidTerm					
Object Name	Required	Type	Size	List of Values	Comment
Semester	Y	List	338	Fall Spring Summer Winter	Selection list.
Year	Y	List		Select a Year 2010 2011 2012	
Course/Section Number	Y	List			Dynamic List.
Student ID	Y				Dynamic List.
Mid-Term Grade		List		Select a Grade S U INC	
Submit		Button			
Reset		Button			

Figure 4.3.4:1 – Enter Mid-Term Grade

ENTER MID-TERM GRADE

Semester:

Year:

Course/Section Number:

Student ID:

Midterm Grade:

4.3.4.1.2 System Process/Output

The following events are executed during the enter grade process.

1. Require semester, year, course/section number, and student ID selection values.
2. Dynamically builds Course/Section Number selection list values based on logged in instructor ID, semester and year selection values. That is code identifies what courses the instructor teaches in a given semester and year.
3. Dynamically builds “Student ID” selection list values based on the “Course/Section Number” selection value; that is code identifies what students registered for a given course in a given semester.
4. Preset selection values for Mid-term grades.
5. Check to ensure that grade is not already entered for that student in the given semester and year.
6. Confirm that grade was submitted successfully as shown in [figure 4.3.4:2](#).
7. Update student record.

Figure 4.3.4:2 – Enter Mid-Term Confirmation

Successfully entered the Mid-Term Grade for Student 700000000

[Click Here To Return to 'Enter Mid-Term Grade' Page](#)

4.3.4.2 Final Grades Screen

Screen Name	File Name	Menu Access	Description
Final Grades.	EnterFinalGrade.php	Grade>Final Grade	Enter student final grade.

4.3.4.2.1 Screen Specification

Grade-Final					
Object Name	Required	Type	Size	List of Values	Comment
Semester	Y	List	338	Select a Semester Fall Spring Summer Winter	Selection list.
Year	Y	List		Select a Year 2010 2011 2012	
Course/Section Number	Y	List			Dynamic List
Student ID	Y				Dynamic List
Final Grade		List		Select a Grade	Preset values; see figure 4.3.4:4
Submit		Button			
Reset		Button			

Figure 4.3.4:3 – Enter Final Grade

ENTER FINAL GRADE

Semester:

Year:

Course/Section Number:

Student ID:

Final Grade:

4.3.4.2.2 System Process/Output

The following events are executed during the enter grade process.

1. Requires semester, year, course/section number, and student ID selection values.
2. Dynamically builds Course/Section Number selection list values based on logged in instructor ID, semester and year selection values. That is code identifies what courses the instructor teaches in a given semester and year as shown in the code snippet below.
3. Dynamically builds “Student ID” selection list values based on the “Course/Section Number” selection value; that is code identifies what students registered for a given course in a given semester.
4. Preset selection values for final grades as shown in [figure 4.3.4:4](#).
5. Check to ensure that grade is not already entered for that student in the given semester and year.
6. Confirms that grade was submitted successfully.
7. Updates student record.

Figure 4.3.4:4 – Final Grade List

The screenshot displays a web form titled "ENTER GRADE" on a yellow background. The form contains several input fields and a dropdown menu. The fields are labeled "Semester:", "Year:", "Course/Section Number:", "Student ID:", and "Final Grade:". A dropdown menu is open, showing a list of grades: A, A+, A-, B, B+, B-, C, C+, C-, D, D+, D-, and F. The "Final Grade:" field has a small dropdown arrow next to it, and the "Final Grade:" label is in red.

4.3.4.2.3 Code Snippet

The following code dynamically sets the “Course/Section Number” select list values.

```
<?php
    include './functions.php';
    if($_SESSION['userType'] == "Student"){
        die("Unauthorized access");
    }
?>

<?php
$q=$_GET["q"];
$array = explode("-", $q); // id - semester - year

// Variables
$semester = $array[0];
$year = $array[1];

// $con = mysql_connect("localhost","root","Yahweh") or die();
if (!$con)
{
    die('Could not connect: ' . mysql_error());
}

mysql_select_db("registrationsystem", $con) or die();

//Need CourseName and Section Number
$sql="SELECT courseId, sectNum FROM section WHERE staffId = '".$_SESSION['id']."' . "" . " AND semester = '".$_semester.'" . " AND year = '".$_year.'"";

$result = mysql_query($sql);
echo '<option>Select a Course</option>';
while($row = mysql_fetch_assoc($result))
{
    echo '<option>' . $row['courseId'] . '-' . $row['sectNum'] . '</option>';
}

?>
```

4.3.4.3 REPORT MENU

4.3.4.3.1 Class Roster Screen

Screen Name	File Name	Menu Access	Description
View Class Roster	Instructor_ViewClassRoster.php	Report>Class Roster	Generate class roster for a given section.

4.3.4.3.2 Screen Specification

Class Roster					
Object Name	Required	Type	Size	List of Values	Comment
Semester	Y	List	338	Fall Spring Summer Winter	Selection list.
Year	Y	List			
SectionID	Y	List			Dynamic list.
Search		Button			
Reset		Button			

Figure 4.3.4:5 – Class Roster

VIEW CLASS ROSTER

Semester:

Year:

Course/Section:

4.3.4.3.3 System Process/Output

The following events are executed during the view class roster process.

1. Requires semester, year, and course section selection values.

2. Dynamically builds “Course/Section” selection list values based on the semester and year entered.
3. Output report on screen that include Student ID, First Name, and Last Name.

Figure 4.3.4:6 – Class Roster Output

VIEW CLASS ROSTER

Semester:

Year:

Course/Section:

Class Roster for Course MA100 Section 1 (Spring 2012)

Student ID	First Name	Last Name
700446690	Deokie	Seerattan
700596948	Andrew	Holder

© 2011 KADA Institute of Technology
All Rights Reserved

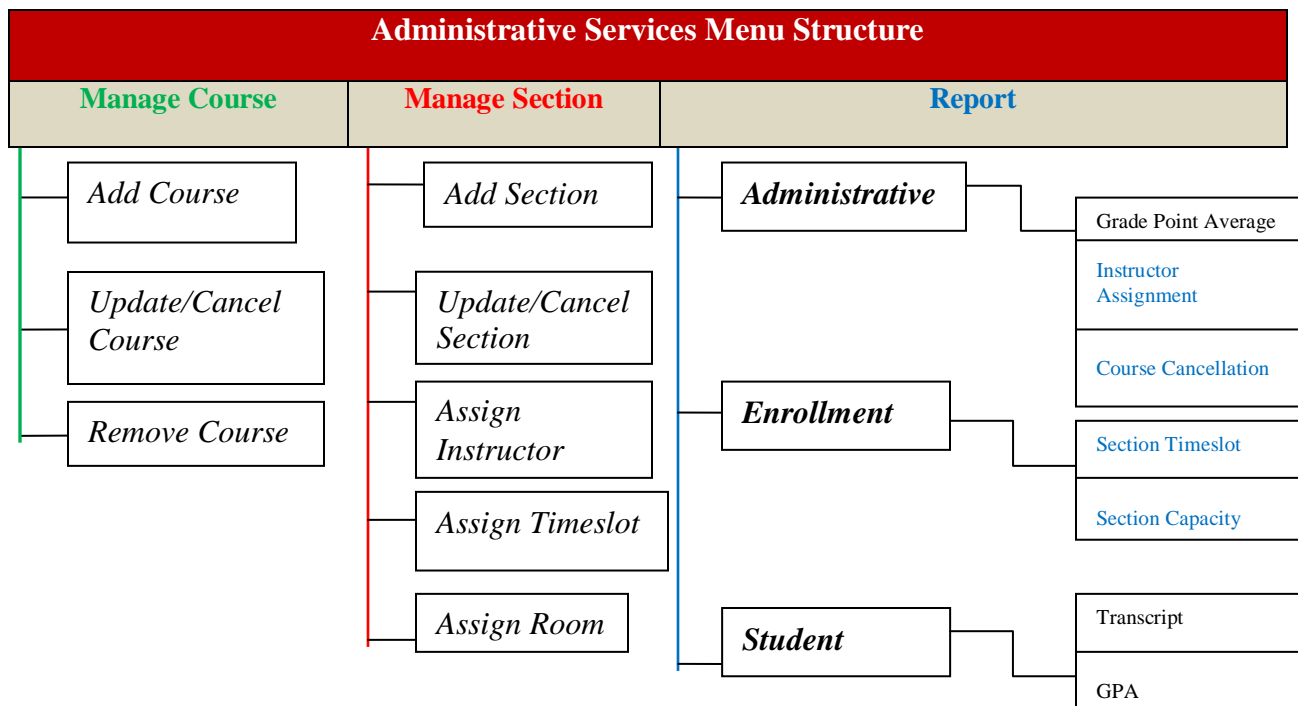
4.4 ADMINISTRATIVE MODULE

This module supports business function requirements described in FRDS-2.2. The requirements in “blue” font in the menu table below are deemed out of scope for delivery.

- FRDS 2.2.1 - manage courses; add and update course offerings for a given semester.
- FRDS 2.2.1.4 - Manage Section – add and update course sections for a given semester.
- CRQ3 - Generate Administrative reports.
- CRQ4 - Generate Enrollment reports.
- CRQ5 - Generate Student reports.

4.4.1 Navigational Menu Structure

The following diagram represents the Administrative Services navigational menu structure. Note, menu items deemed out of scope due to time constraints or as project phase 3 may not be developed and implemented. These reports are noted in blue font in the [menu structure specification table](#) below.



4.4.2 Menu Structure Specification

Menu Name	Item	Action
Manage Course	<i>Add course</i>	Add a new course to the catalog.
Manage Course	<i>Update/Cancel Course</i>	Update or cancel an existing course for a given semester or year.
Manage Course	<i>Remove Course</i>	Update the status of a course that is no longer offered.
Manage Section	<i>Add Section</i>	Enter a new section in the course catalog.
Manage Section	<i>Update/Cancel Section</i>	Cancel or delete an existing section in the course catalog.
Manage Section	<i>Assign Instructor</i>	Assign an instructor to a course section for a given semester and year.
Manage Section	<i>Assign Timeslot</i>	Assign a timeslot of a course section in a given semester and year.
Manage Section	<i>Assign Room</i>	Assign a room to a course section in a given semester and year.
Report	<i>Administrative>Grade Point Average</i>	Generate report of a course/section grade point average.
Report	<i>Administrative>Instructor Assignment</i>	Generate report of course sections assigned to an instructor in a given semester and year.
Report	<i>Administrative>Course Cancellation</i>	Generate report of cancelled courses in a given semester and year.
Report	<i>Enrollment>Section Timeslot</i>	Generate report of section timeslot in a given semester and year.
Report	<i>Enrollment>Section Capacity</i>	Generate report of how many students are enrolled in a section in a given semester and year.
Report	<i>Student>Transcript</i>	Generate a detailed report of a student academic transcript.
Report	<i>Student>GPA</i>	Generate a report of a student

		grade point average.
--	--	----------------------

4.4.3 MANAGE COURSE MENU

This section defines the specifications for functional requirements as described in the FRDS 2.21. That is administrative staff will be able to add and update course offerings for a given semester.

4.4.3.1 Add Course Screen

Screen Name	File Name	Menu Access	Description
Add Course	Admin_addcourse.php	Manage Course>Add Course	Add a new course that is offered in the course catalog.

4.4.3.1.1 Screen Specification

Add Course					
Object Name	Required	Type	Size	List of Values	Comment
Department	Y	List		Select a Department Business Computer Science Information Technology Management Information Systems Mathematics	Selection list.
Course Number	Y	Text			User input.
Title	Y	Text			User input.
Subject		List		Business Computer Science Information Technology Management Information Systems Mathematics	Selection list.
Number of Credits		List		Select Credit Value 3 4	Selection list.
Course Description		Text			User input.
Add Course		Button			
Reset		Button			

Figure 4.4.3:1 – Add Course Screen

MANAGE COURSE > Add Course

Department:

Course Number:

Title:

Subject:

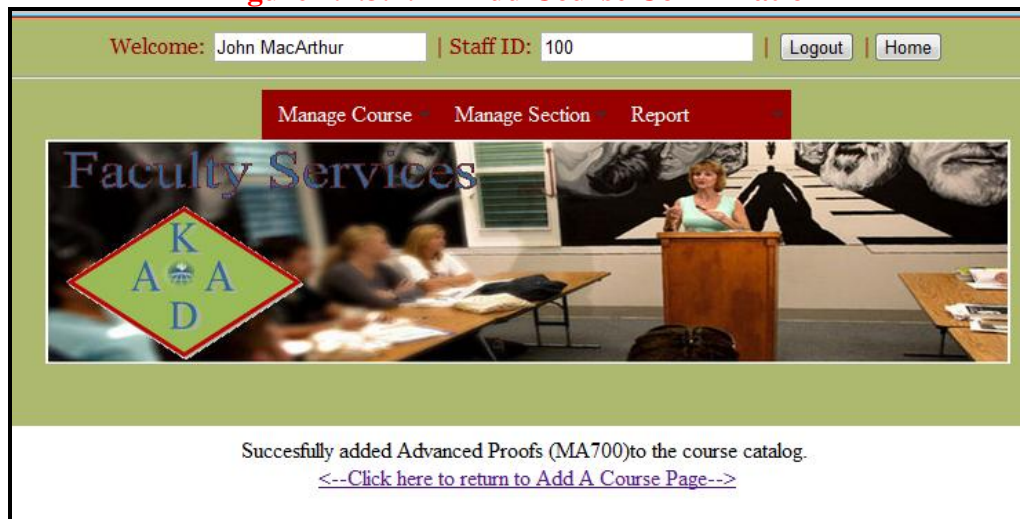
Number of Credits:

Course Description:

4.4.3.1.2 System Process/Output

The following events are executed during the add course process.

1. Require department value selected from preset list.
2. Require course number and title entries.
3. Dynamically builds subject selection list based on department value.
4. Require number of credits and course description entries.
5. On add course action, output confirmation message on screen as shown in [figure 4.4.3.1:1 below](#).
6. Add and display the course record in the course catalog.

Figure 4.4.3.1:1 – Add Course Confirmation

4.4.3.2 Update/Cancel Course Screen

Screen Name	File Name	Menu Access	Description
Update/Cancel Course	Admin_updatecourse.php	Manage Course>Update/Cancel Course	Update an existing course title or description, or cancel the course in a given semester and year.

4.4.3.2.1 Screen Specification

Update/Cancel Course					
Object Name	Required	Type	Size	List of Values	Comment
Semester	Y	List	338	Select a semester Fall Spring Summer Winter	Selection list.
Year	Y	List		Select a Year 2010 2011 2012	
SectionID	Y	List			Dynamic list.
Update Course		Button			
Reset		Button			

Figure 4.4.3:2 Update Course Screen

MANAGE COURSE > Update Course

Department:

Course Id:

Title:

Course Description:

4.4.3.2.2 System Process/Output

The following events are executed during the update course process.

1. Requires semester, year, and section ID values.
2. Dynamically builds section selection list based on semester and year values.
3. Enter new course title or description.
4. Updates field values in the course table.
5. On Cancel action, check for section registration; if found, remove section from student class schedule.
6. Update section status to “Not Available” and hide from course catalog.

4.4.3.3 Remove Course Screen

Screen Name	File Name	Menu Access	Description
Remove Course	deletecourse.php	Manage Course>Remove Course	Remove course that is no longer offered.

4.4.3.3.1 Screen Specification

Remove Course					
Object Name	Required	Type	Size	List of Values	Comment
Department	Y	List	N/A	Select a Department All Business Computer Science Information Technology Management Information Systems Mathematics	
Course	Y	List	N/A		Dynamic list.
Remove Course		Button			
Reset		Button			

Figure 4.4.3:3 Remove Course

4.4.3.3.2 System Output/Process

The following events are executed during the remove course process.

1. Require department value.
2. Dynamically build course selection list based on department value entered.
3. On remove course action, update the status of the course record in the database to “not available” and hide the course from the course catalog.

4.4.3.4 Add Section Screen

Screen Name	File Name	Menu Access	Description
Add Section	Admin_addsection.php	Manage Section>Add Section	Add a new section.

4.4.3.4.1 Screen Specification

Add Section					
Object Name	Required	Type	Size	List of Values	Comment
Semester	Y	List	N/A	Select a Semester Fall Spring Summer Winter	Selection list.
Year	Y	List	N/A	2010 2011 2012	Selection list.
Department	Y	List	N/A	Select a Department All Business Computer Science Information Technology Management Information Systems Mathematics	
Course	Y	List	N/A		Dynamic list.
Section Number	Y	List	N/A		Dynamic list.
Instructor	Y	List	N/A	Select an Instructor	Dynamic list.
Section Capacity	Y	Text	N/A		User Input.
Building	Y	List	N/A	Select a Building	Dynamic list.
Room Number		List	N/A		Dynamic list.
Session		List	N/A	Select a Session	Dynamic list.
Days		List	N/A	Select Days	Dynamic list.
Timeslot		List	N/A		Dynamic list.
Start Time		List	N/A	Select Start Time	Dynamic list.
End Time		List	N/A	Select End Time	Dynamic list.

Figure 4.4.3:4 Add Section Screen
MANAGE SECTION > Add Section

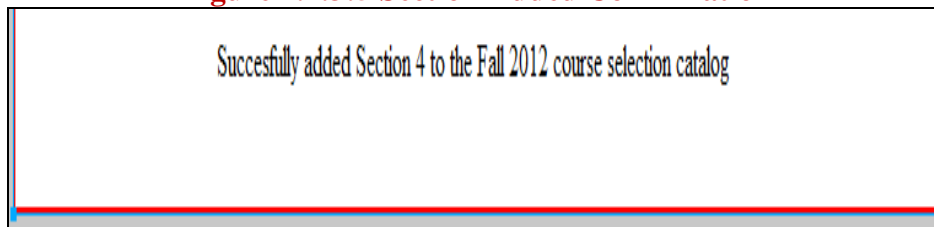
The screenshot shows a web form titled "MANAGE SECTION > Add Section". The form is set against a light yellow background. It contains the following fields and controls:

- Semester:** A dropdown menu with the text "Select a Semester".
- Year:** A dropdown menu with the text "Select a Year".
- Department:** A dropdown menu with the text "Select a Department".
- Course:** A dropdown menu with the text "Select a Course".
- Section Number:** A text input field.
- Instructor:** A dropdown menu with the text "Select a Instructor".
- Section Capacity:** A text input field.
- Building:** A dropdown menu with the text "Select a Building".
- Room Number:** A dropdown menu with the text "Select a Room".
- Session:** A dropdown menu with the text "Select a Session".
- Days:** A dropdown menu with the text "Select Days".
- Time Slot:** This field is split into two parts: "Start:" followed by a dropdown menu "Select Start Time", and "End:" followed by a dropdown menu "Select End Time".

4.4.3.4.2 System Process/Output

The following events are executed during the add course section process.

1. Require semester, year, department, and course values to be entered.
2. Dynamically builds the section number selection list based on values entered in step 1.
3. Dynamically builds the instructor selection list based on valued entered in step 2.
4. Require section capacity value.
5. Dynamically builds the building selection list based on value entered in step 4.
6. Dynamically builds the room number selection list based on building value.
7. Dynamically builds the session and days selection list values.
8. Dynamically builds start and end time based on days value.
9. On submit add course section into the related database tables as shown in [figure 4.4:3:5](#) below.

Figure 4.4.3:5 Section Added Confirmation

4.4.3.5 Update/Cancel Section Screen

Screen Name	File Name	Menu Access	Description
Update/Cancel Section	Admin_Cancel_Deletesection.php	Manage Section>Update/Cancel Section	Update or cancel an existing section.

4.4.3.5.1 Screen Specification

Cancel Section					
Object Name	Required	Type	Size	List of Values	Comment
Semester	Y	List	N/A	Select a Semester	Selection list.
Year				2010 2011 2012	
Department	Y	List	N/A	Select a Department All Business Computer Science Information Technology Management Information Systems Mathematics	
Course	Y	List	N/A		Dynamic list.
Section Number	Y	List	N/A		Dynamic list.

Figure 4.4.3:6 Cancel Section Screen

MANAGE SECTION > Cancel/Delete Section

Semester:

Year:

Department:

Course:

Section Number:

4.4.3.5.2 System Process/Output

The following events are executed during the cancel section process.

1. Require semester, year, and department values to be entered.
2. Dynamically builds the course number selection list based on values entered in step 1.
3. Dynamically builds the section number selection list based on course selection value.
4. On cancel, update the status to the course section in the related database tables, do not display the section in the course catalog, and remove the course from the student schedule.

4.4.3.6 Assign Instructor Screen

Screen Name	File Name	Menu Access	Description
Assign Instructor	UpdateSectionInstructor.php	Manage Section>Assign Instructor	Assign an instructor to a course section.

4.4.3.6.1 Screen Specification

Assign Instructor					
Object Name	Required	Type	Size	List of Values	Comment
Semester	Y	List	N/A	Select a Semester	Selection list.
Year				2010 2011 2012	
Department	Y	List	N/A	Select a Department All Business Computer Science Information Technology Management Information Systems Mathematics	
Course	Y	List	N/A		Dynamic list.
Section Number	Y	List	N/A		Dynamic list.
Instructor		List	N/A		Dynamic list.
Save Change		Button			
Reset		Button			

Figure 4.4.3:7 Assign Instructor Screen

MANAGE SECTION > Assign Instructor

Semester:

Year:

Department:

Course:

Section Number:

Instructor:

4.4.3.6.2 System Process/Output

The following events are executed during the assign instructor process.

1. Require semester, year, and department.
2. Dynamically builds course selection list value based on department entered.
3. Dynamically builds section number selection list based on course selection, semester, and year values entered.
4. Dynamically builds instructor selection list based on department entered.
5. On save changes action, updates course section in the related database tables.

4.4.3.7 Assign Timeslot Screen

Screen Name	File Name	Menu Access	Description
Assign Timeslot	Admin_updatesectiontimeslot.php	Manage Section>Assign Timeslot	Assign a timeslot to a section.

4.4.3.7.1 Screen Specification

Assign Timeslot					
Object Name	Required	Type	Size	List of Values	Comment
Semester	Y	List	N/A	Select a Semester	Selection list.
Year				2010 2011 2012	
Department	Y	List	N/A	Select a Department All Business Computer Science Information Technology Management Information Systems Mathematics	
Course	Y	List	N/A		Dynamic list.
Section Number	Y	List	N/A		Dynamic list.
Days		List	N/A		Dynamic list.

Assign Timeslot					
Object Name	Required	Type	Size	List of Values	Comment
Start		List			Selection list.
End		List			Selection list.
Assign Time		Button			
Reset		Button			

Figure 4.4.3:8 Assign Timeslot Screen

MANAGE SECTION > Assign Time

Semester:

Year:

Department:

Course:

Section:

Session:

Days:

Time Slot: Start: End:

4.4.3.7.2 System Process/Output

The following events are executed during the assign timeslot process.

1. Require semester, year, and department.
2. Dynamically builds course selection list value based on department entered.
3. Dynamically builds section number selection list based on course selection, semester, and year values entered.
4. Require session and days.
5. Dynamically builds timeslot selection list based on session value.
6. Dynamically builds start time selection list based on session value.

7. On assign time slot action, updates course section in the related database tables.

4.4.8 REPORT MENU

This section defines the specifications for functional requirements as described in the change request document, included here as ([Appendix B](#)). This is considered a change in requirement that was not originally included in the FRD.

4.4.8.1 Grade Point Average

Screen Name	File Name	Menu Access	Description
Course GPA	Admin_courseGPA.php	Administrative>Grade Point Average	Calculate course GPA for a given semester and year.

4.4.8.1.1 Screen Specification

Grade Point Average					
Object Name	Required	Type	Size	List of Values	Comment
Department	Y	List	N/A	Select a Department All Business Computer Science Information Technology Management Information Systems Mathematics	
Course	Y	List	N/A		Dynamic list.
GPA					
Semester	Y	List	N/A	Select a Semester Fall Spring Summer Winter	Selection list.
Year		List	N/A	2010 2011 2012	Selection list.
GPA for Semester					
Submit		Button			
Reset		Button			

Figure 4.4.8:1 – Course GPA

Report > Course GPA

Department:

Course:

GPA:

Semester:

Year:

GPA for Semester:

4.4.8.1.2 System Process/Output

The following events are executed during the GPA report process.

1. Require admin staff to select a department and a course.
2. Dynamically builds the course selection list based on the department value that is selected.
3. On course selection, displays GPA for selected course in the GPA field as shown in [figure 4.4.8:2](#) below.

Figure 4.4.8:2 – Course GPA Output

The screenshot shows a web form titled "Report > Course GPA". It contains the following fields:

- Department: Mathematics (dropdown menu)
- Course: Pre-Calculus (dropdown menu)
- GPA: 2.7 (text input)
- Semester: Select a Semester (dropdown menu)
- Year: Select a Year (dropdown menu)
- GPA for Semester: (empty text input)

A red bar at the bottom contains a "Reset" button.

4. Optionally, to generate the GPA of the selected course for a given semester, require admin staff to select a year and semester.
5. Displays the GPA for the semester in the GPA for semester field as shown in [figure 4.4.8:3](#) below.

Figure 4.4.8:2 – Course GPA for Semester Output

The screenshot shows the same web form as Figure 4.4.8:2, but with the following values:

- Department: Mathematics (dropdown menu)
- Course: Pre-Calculus (dropdown menu)
- GPA: 2.7 (text input)
- Semester: Fall (dropdown menu)
- Year: 2011 (dropdown menu)
- GPA for Semester: 2.7 (text input)

A red bar at the bottom contains a "Reset" button.

4.4.8.2 Student Transcript

Screen Name	File Name	Menu Access	Description
Student Transcript	Admin_studenttranscript.php	Student>Transcript	Generate a student transcript report.

4.4.8.2.1 Screen Specification

Student Transcript					
Object Name	Required	Type	Size	List of Values	Comment
StudentId	Y	Text	N/A		User Input.
Submit		Button			
Reset		Button			

Figure 4.4.8:3 – Generate Student Transcript Screen

4.4.8.2.2 System Process/Output

The following events are executed during the generate student transcript report process.

1. Require admin staff to enter the student ID.
2. On submit, generate and display the student transcript as shown in [figure 4.4.8:4](#) below.

Figure 4.4.8:4 Student Transcript Report Output

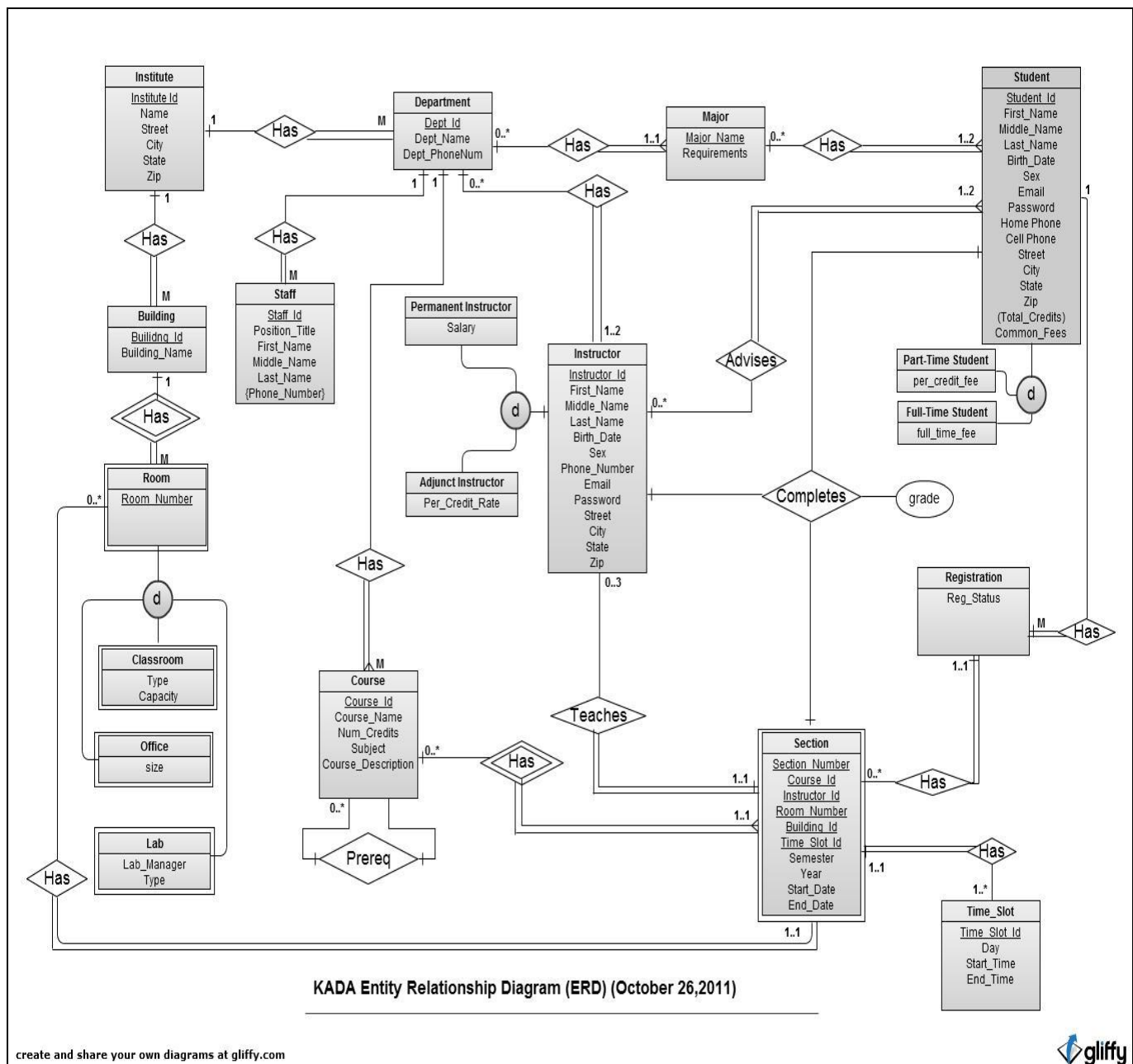
	Student I.D.	First Name	Last Name	Birth Date
	700544268	Kadeem	Quallis	1990-03-30
	Major	GPA		
	Computer Information Science	3.425		
Official KADA Institute of Technology Transcript				
Term: Fall 2010				
Subject	Credits	Course	Title	Grade
Business	4	BS100	Inro. to Business Management	B
Term: Spring 2011				
Subject	Credits	Course	Title	Grade
Business	4	BS200	Accounting I	A
Term: Fall 2011				
Subject	Credits	Course	Title	Grade
Computer Science	4	CS100	Java Programming I	A
Mathematics	4	MA100	Pre-Calculus	B-

5.0 SYSTEM INFRASTRUCTURE DESIGN

This section defines the technical specifications of the backend or server side system requirements. It provides the database foundation details in the form of an ER diagram and relational database schema used to create the KADA registration system database.

5.1 Entity Relationship Diagram

The following diagram illustrates the entities and relationships of the database objects.



5.2 Relational Schema

The following section defines the entity sets and relationship set between database objects.

5.2.1 Entity Sets

1. **Building**(buildingId, buildingName)
2. **Room**((roomNum, buildingId) *references* Building)
3. **Classroom**((roomNum, BuildingId) *references* Room, capacity, type)
4. **Office**((roomNum, buildingId) *references* Room, size)
5. **Lab**((roomNum, buildingId) *references* Room, type)
6. **Department**(deptId, deptName, deptPhoneNum)
7. **Staff**(staffId, fName, mName, lName, birthDate, sex, phoneNum, email, password, street, city, state, zip, deptId *references* Department)
8. **Major**(majorName, requirements, deptId *references* Department)
9. **Course**(courseId, deptId *references* Department, coursName, numCredits, subject, courseDescription)
10. **Section**(secNum, courseId *references* Course, (semester, year) *references* Semester, staffId *references* Instructor, (roomId, buildingId) *references* Room, timeslotId *references* Timeslot)
11. **TimeSlot**(timeslotId, day, startTime, endTime)
12. **Instructor**(staffId *references* Staff, rank)
13. **PermanentInstructor**(staffId *references* Instructor, salary)
14. **AdjunctInstructor**(staffId *references* Instructor, perCreditRate)
15. **Student**(studentId, fName, mName, lName, birthDate, sex, street, city, state, zip, birthDate, email, password, homePhone, cellPhone, commonFees, status, securityQuestion, securityAnswer, majorName *references* Major)
16. **Registration**((secNum, courseId, semester, year) *references* Section, studentId *references* Student, regStatus)
17. **EmergencyContact**(firstName, lName, primaryPhone, secondaryPhone, studentID *references* Student)
18. **Semester**(semester, year, startDate, endDate)

5.2.2 Relationship Sets

1. **StaffDept**(deptId, staffId, percentage)
2. **StudentMajor**(studentId, majorID)
3. **Prereq**(courseId, prereqId)
4. **Advises**(studentId, instructorId)
5. **Completes**(studentId, (sectionNum, courseId, semester, year), staffId, grade)

5.3 Database

The KADA Registration system database design consists of the following objects. These objects are listed in alphabetical order by Table ID (name) below.

5.3.1 Tables

Object #	Table ID
1	<i>AdjunctInstructor</i>
2	<i>Advises</i>
3	<i>Building</i>
4	<i>Completes</i>
5	<i>Course</i>
6	<i>Department</i>
7	<i>Emergency Contact</i>
8	<i>Grade Points</i>
9	<i>Instructor</i>
10	<i>Major</i>
11	<i>PermanentInstructor</i>
12	<i>Prereq</i>
13	<i>Registration</i>
14	<i>Room</i>
15	<i>Section</i>
16	<i>Semester</i>
17	<i>Staff</i>
18	<i>Staffdept</i>
19	<i>Student</i>
20	<i>Studentmajor</i>
21	<i>Studentpins</i>
22	<i>Timeslots</i>

5.3.2 Database Objects

The following section defines the specifications of each table as shown in the objects below. Details include the table name, fields, and attributes.

AdjunctInstructor

Field	Type	Null	Key	Default	Extra
staffId	int(11)	NO	PRI	NULL	
perCreditFee	double	NO		NULL	

Advises

Field	Type	Null	Key	Default	Extra
staffId	int(11)	NO	PRI	NULL	
stuId	int(11)	NO	PRI	NULL	

Building

Field	Type	Null	Key	Default	Extra
buildingName	varchar(45)	NO	PRI	NULL	

Classroom

Field	Type	Null	Key	Default	Extra
roomNum	int(11)	NO	PRI	NULL	
buildingName	varchar(45)	NO	PRI	NULL	
capacity	int(11)	NO		NULL	
type	varchar(45)	NO		NULL	

Completes

Field	Type	Null	Key	Default	Extra
stuId	int(11)	NO	PRI	NULL	
staffId	int(11)	NO	PRI	NULL	
sectNum	varchar(3)	NO	PRI	NULL	
courseId	varchar(45)	NO	PRI	NULL	
semester	varchar(45)	NO	PRI	NULL	
year	year(4)	NO	PRI	NULL	
mGrade	varchar(3)	NO		NULL	
fGrade	varchar(3)	NO		NULL	

Course

Field	Type	Null	Key	Default	Extra
courseId	varchar(45)	NO	PRI	NULL	
courseName	varchar(45)	NO		NULL	
numCredits	double	NO		NULL	
subject	varchar(45)	NO		NULL	
courseDescription	varchar(1000)	NO		NULL	
deptName	varchar(45)	NO		NULL	
active	tinyint(1)	NO		NULL	

Department

Field	Type	Null	Key	Default	Extra
deptName	varchar(45)	NO	PRI	NULL	
deptPhoneNum	varchar(45)	NO		NULL	

Grade Points

Field	Type	Null	Key	Default	Extra
grade	varchar(2)	NO	PRI	NULL	
point	double	NO		NULL	

EmergencyContact

Field	Type	Null	Key	Default	Extra
stuId	int(11)	NO	PRI	NULL	
fName	varchar(45)	NO		NULL	
mName	varchar(45)	YES		NULL	
lName	varchar(45)	NO		NULL	
relationship	varchar(45)	YES		NULL	
street	varchar(45)	YES		NULL	
city	varchar(45)	YES		NULL	
state	varchar(45)	YES		NULL	
zip	varchar(45)	YES		NULL	
email	varchar(45)	YES		NULL	
primaryPhone	varchar(45)	NO		NULL	

secondaryPhone	varchar(45)	NO		NULL	
----------------	-------------	----	--	------	--

Instructor

Field	Type	Null	Key	Default	Extra
staffId	int(11)	NO	PRI	NULL	
rank	varchar(45)	NO		NULL	

Lab

Field	Type	Null	Key	Default	Extra
roomNum	int(11)	NO	PRI	NULL	
buildingName	varchar(45)	NO	PRI	NULL	
labManager	varchar(45)	NO		NULL	
type	varchar(45)	NO		NULL	

Major

Field	Type	Null	Key	Default	Extra
majorName	varchar(45)	NO	PRI	NULL	
requirements	varchar(1000)	NO		NULL	
deptName	varchar(45)	NO	MUL	NULL	

Office

Field	Type	Null	Key	Default	Extra
roomNum	int(11)	NO	PRI	NULL	
buildingName	varchar(45)	NO	PRI	NULL	
size	varchar(45)	NO		NULL	

PermanentInstructor

Field	Type	Null	Key	Default	Extra
staffId	int(11)	NO	PRI	NULL	
salary	double	NO		NULL	

Prereq

Field	Type	Null	Key	Default	Extra
courseId	int(11)	NO	PRI	NULL	
prereqId	int(11)	NO	PRI	NULL	

Registration

Field	Type	Null	Key	Default	Extra
stuId	int(11)	NO	PRI	NULL	
sectNum	int(11)	NO	PRI	NULL	
courseId	int(11)	NO	PRI	NULL	
Semester	varchar(45)	NO	PRI	NULL	
year	year(4)	NO	PRI	NULL	
Reg_Status	varchar(45)	NO		NULL	

Room

Field	Type	Null	Key	Default	Extra
roomNum	varchar(4)	NO	PRI	NULL	
buildingName	varchar(45)	NO	PRI	NULL	
capacity	int(11)	NO		NULL	
type	varchar(45)	NO		NULL	

Section

Field	Type	Null	Key	Default	Extra
sectNum	varchar(3)	NO	PRI	NULL	
courseId	varchar(45)	NO	PRI	NULL	
semester	varchar(45)	NO	PRI	NULL	
year	year(4)	NO	PRI	NULL	
timeSlotId	int(11)	NO	MUL	NULL	
roomNum	varchar(4)	NO	MUL	NULL	
buildingName	varchar(45)	NO		NULL	
staffId	int(11)	NO	MUL	NULL	
capacity	int(11)	NO		NULL	

Semester

Field	Type	Null	Key	Default	Extra
semester	varchar(45)	NO	PRI	NULL	
year	year(4)	NO	PRI	NULL	
startDate	date	NO		NULL	
endDate	date	NO		NULL	

Staff

Field	Type	Null	Key	Default	Extra
staffId	int(11)	NO	PRI	NULL	
positionTitle	varchar(45)	NO		NULL	
fName	varchar(45)	NO		NULL	
mName	varchar(45)	YES		NULL	
lName	varchar(45)	NO		NULL	
birthDate	varchar(45)	NO		NULL	
sex	varchar(7)	NO		NULL	
phoneNum	varchar(45)	NO		NULL	
email	varchar(45)	NO		NULL	
password	varchar(45)	NO		NULL	
street	varchar(45)	NO		NULL	
city	varchar(45)	NO		NULL	
state	varchar(45)	NO		NULL	
zip	varchar(45)	NO		NULL	
securityQuestion	varchar(45)	YES		NULL	
securityAnswer	varchar(45)	YES		NULL	

StaffDept

Field	Type	Null	Key	Default	Extra
deptName	varchar(45)	NO	PRI	NULL	
staffId	int(11)	NO	PRI	NULL	

Student

Field	Type	Null	Key	Default	Extra
stuId	int(11)	NO	PRI	NULL	
fName	varchar(45)	NO		NULL	
mName	varchar(45)	YES		NULL	
lName	varchar(45)	NO		NULL	
status	varchar(45)	NO		NULL	
birthDate	date	NO		NULL	
sex	varchar(45)	NO		NULL	
email	varchar(45)	NO		NULL	
password	varchar(45)	NO		NULL	
homePhone	varchar(45)	NO		NULL	

cellPhone	varchar(45)	NO		NULL	
street	varchar(45)	NO		NULL	
city	varchar(45)	NO		NULL	
state	varchar(45)	NO		NULL	
zip	varchar(45)	NO		NULL	
commonFees	double	YES		NULL	
securityQuestion	varchar(1000)	NO		NULL	
securityAnswer	varchar(1000)	NO		NULL	

Studentmajor

Field	Type	Null	Key	Default	Extra
majorName	varchar(45)	NO	PRI	NULL	
studId	int(11)	NO	PRI	NULL	

Studentpins

Field	Type	Null	Key	Default	Extra
stuId	int(11)	NO	PRI	NULL	
semseter	varchar(45)	NO	PRI	NULL	
year	year(4)	NO	PRI	NULL	
pin	varchar(45)	NO	PRI	NULL	

Timeslots

Field	Type	Null	Key	Default	Extra
Time_Slot_Id	int(11)	NO	PRI	NULL	
Day	varchar(45)	NO		NULL	
Start_Time	time	NO		NULL	
End_Time	time	NO		NULL	

6.0 TECHNOLOGY

The KADA Registration system environment consists of MYSQL Server, PHP and HTML format webpages, and PHP, JavaScript, and Ajax codes.

6.1 Server Hardware/Software Requirements

The following software and hardware requirements are recommended for the server environment.

Hardware:

- Minimum of 1GZ processor
- Minimum of 8 GB of memory
- Minimum of 40 GB of available disk space

Software:

- Apache Server – latest version
- Minimum - Windows 2000 Server, Service Pack 2
- MY SQL Workbench

6.2 Client Hardware/Software Requirements

The following software and hardware requirements are recommended for the client or user environment.

Hardware:

- Minimum of 1GZ processor
- Minimum of 4 GB of memory
- Minimum of 5 GB of available disk space













Software:

- Windows XP or latest Windows Operating System version
- Internet Explorer version 7 or higher
- Latest version of Google Chrome or Safari Firefox browser


















7.0 APPENDIX

This section provides additional details and information directly related to the system design.

7.1 Appendix A – Script Files

Name	Type	Compressed size	Password ...	Size	Ratio	Date modified
 addCourseForm.php	PHP Script	1 KB	No	2 KB	47%	12/5/2011 9:53 AM
 addSectionForm.php	PHP Script	2 KB	No	4 KB	63%	12/22/2011 2:41 PM
 cancelSectionForm.php	PHP Script	1 KB	No	2 KB	54%	12/22/2011 12:13 PM
 enterFinalGradeForm.php	PHP Script	1 KB	No	2 KB	48%	12/21/2011 1:01 PM
 enterMidTermGradeForm.php	PHP Script	1 KB	No	2 KB	49%	12/21/2011 12:22 PM
 LoginForm.php	PHP Script	2 KB	No	6 KB	76%	11/22/2011 8:26 PM
 Logout.php	PHP Script	1 KB	No	2 KB	45%	11/22/2011 9:36 PM
 removeCourseForm.php	PHP Script	1 KB	No	1 KB	36%	12/20/2011 9:32 PM
 studentPinForm.php	PHP Script	1 KB	No	2 KB	41%	12/13/2011 6:44 PM
 updateCourseForm.php	PHP Script	1 KB	No	1 KB	41%	12/21/2011 7:11 PM
 updateSectionInstructorForm.php	PHP Script	1 KB	No	2 KB	47%	12/22/2011 12:54 AM
 updateSectionTimeForm.php	PHP Script	1 KB	No	2 KB	53%	12/22/2011 9:16 AM

7.1 Appendix A – Script Files

Name	Type	Compressed size	Password ...	Size	Ratio	Date modified
 functions.php	PHP Script	1 KB	No	1 KB	33%	12/6/2011 6:19 AM
 getCourseDescription.php	PHP Script	1 KB	No	1 KB	34%	12/21/2011 6:41 PM
 getCourseName.php	PHP Script	1 KB	No	1 KB	33%	12/21/2011 6:42 PM
 getGPA.php	PHP Script	1 KB	No	1 KB	31%	12/22/2011 1:30 PM
 getPin.php	PHP Script	1 KB	No	2 KB	53%	12/20/2011 5:38 PM
 loadAdvisementList.php	PHP Script	1 KB	No	1 KB	34%	12/13/2011 7:32 PM
 Login.js	JScript Script File	1 KB	No	1 KB	65%	12/6/2011 6:29 AM
 updateCourseSelection.php	PHP Script	1 KB	No	1 KB	37%	12/22/2011 8:57 AM
 updateCourseSelectionByld.php	PHP Script	1 KB	No	1 KB	37%	12/21/2011 6:31 PM
 updateInstructorCourseSelection.p...	PHP Script	1 KB	No	1 KB	42%	12/14/2011 11:07 PM
 updateInstructorSelection.php	PHP Script	1 KB	No	1 KB	45%	12/10/2011 10:42 AM
 updateInstructorStudentSelection....	PHP Script	1 KB	No	1 KB	43%	12/22/2011 10:52 AM
 updateRoomSelection.php	PHP Script	1 KB	No	1 KB	43%	12/17/2011 7:02 AM
 updateSectionSelection.php	PHP Script	1 KB	No	1 KB	50%	12/22/2011 12:23 AM
 updateSectionSelection2.php	PHP Script	1 KB	No	1 KB	50%	12/22/2011 11:24 AM
 utilities.js	JScript Script File	2 KB	No	8 KB	84%	12/21/2011 6:09 PM
 Validations.js	JScript Script File	1 KB	No	5 KB	83%	12/22/2011 2:40 PM

7.2 Appendix B - Change Request Form

This document serves as an addendum to the Functional Requirements Specification executed between K.A.D.A Institute of Technology and the KADA Consulting Group dated 10/26/2011.

K.A.D.A. Institute of Technology has requested additional services/change from KADA Consulting Group that was not included in the Functional Requirements Specification, but is closely associated with the project. Therefore, the parties have agreed to amend the Functional Requirements Specification to change/expand the scope of the project. The following is a description of the requested services/change

Date	Requested By	Change Description
12/22/2011	N. Gupta	Grade Point Average Report
12/22/2011	N. Gupta	Student Transcript

7.3 Appendix C – File Structure

The following is a snapshot of KADA site root folder and subfolders.

