

GRADUATION DATABASE SYSTEM

By

SRIKANTH GADDAM

B.E, OSMANIA UNIVERSITY, 2009

A REPORT

Submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Computing and Information Sciences
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2011

Approved by:

Major Professor
Dr. Gurdip Singh

ABSTRACT

Education plays a very important role in molding a person into a good individual by imparting knowledge in all aspects. Globalization has helped the students to go to different countries, attain knowledge and exposure on the culture of that country. Universities receive several hundreds of applications from all over the world to get admission for higher studies. By following the conventional method the faculty of the university and also the applicant will face lot of difficulties in communicating. In conventional method the acceptance or rejection would be generally intimated through post. The applicant will be in complete ignorance during the processing of application. There will be several scenarios where interaction between them is necessary to further process the application. This project is intended to overcome such problems and make the interaction between the staff and applicant smoother.

The main objective of the project is to develop a website for Computing and Information Sciences department which manages the data regarding the prospective students as well as students currently enrolled. It can be accessed by prospective students (who wish to join the university), faculty and administrators. Applicant will be notified constantly through email and they can always check their application status online.

The role of the administrators is to accept applications sent by prospective students and upload their details such as their GRE, TOEFL scores and their academic records into the database. Once these details are stored in the database, the students will be able to access their application status. Students are also given information about missing documents in the database.

There will be a committee to decide the application status whether it will be rejected or accepted. Once the applicant is accepted, the applicant would be added to the current students list. All this process is made easy through this website.

Table of Contents

List of Figures	iv
List of Tables	v
Acknowledgements.....	vi
Chapter 1 - INTRODUCTION	1
1.1 OBJECTIVE	1
1.2 MOTIVATION.....	1
1.3 PROJECT DESCRIPTION.....	1
1.4 INTENDED USERS:.....	2
Chapter 2 - BACKGROUND	3
2.1 PHP.....	3
2.2 MYSQL	3
2.3 PHP+MYSQL.....	3
2.4 Web Application Framework:	3
2.5 Model–View–Controller (MVC)	4
2.6 RELATED WORK:.....	5
Chapter 3 - IMPLEMENTATION.....	7
3.1 SYSTEM ARCHITECTURE: MVC	7
MVC (Model View Controller)	8
3.3 SYSTEM DESIGN.....	9
3.4 USE CASE DIAGRAM:	9
3.5 CLASS DIAGRAMS.....	11
3.6 Working	12
Chapter 4 - TESTING.....	23
4.1 Software Testing	23
Unit Testing	23
Performance Testing	24
Chapter 5 - CONCLUSIONS.....	30
Future Enhancements.....	30

List of Figures

Figure 2.1 Working of MVC	4
Figure 3.1 MVC System Architecture	7
Figure 3.2 3-Tier VS MVC Architecture	8
Figure 3.3 UseCase for Admin	10
Figure 3.4 Use Case for Applicant.....	10
Figure 3.5 Class Diagram from Applicants view.....	12
Figure 3.6 UML Diagram	13
Figure 3.7 Login Screen.....	18
Figure 3.8 Welcome Page	18
Figure 3.9 Create New Application Page	19
Figure 3.10 Search Application	19
Figure 3.11 List of Applications	20
Figure 3.12 Overview Page.....	20
Figure 3.13 Update Applicant.....	21
Figure 3.14 Generate Student Profile.....	21
Figure 3.15 Create New Student Page	22
Figure 3.16 Upload CSV File Page.....	22
Figure 4.1 JMeter Graph Results for Create New Application.....	25
Figure 4.2 JMeter Testing for Search Page.....	26
Figure 4.3 JMeter Testing for Student Profile Page	27
Figure 4.4 JMeter results for search students page	28

List of Tables

Table 2.1 Comparison of framework.....	6
--	---

Acknowledgements

I would also like to express my deep gratitude and wholehearted thanks to my advisor, Dr. Gurdip Singh, for his valuable suggestions and unconditional support given to complete the project successfully.

I would also like to thank my committee members Dr. Daniel Andresen and Dr. Mitchell L. Neilsen for their support throughout the project.

I would like to thank the administrative and technical staff of Computing and Information Sciences Department who have been helpful throughout my graduate study.

Chapter 1 - INTRODUCTION

1.1 OBJECTIVE

The current project involves the development of a website for Computing and Information Sciences. This website is used to manage the data regarding the applicants who apply for admissions in to the department and also to maintain the database for the current students about their grades, deficiencies etc.

1.2 MOTIVATION

One of the crucial tasks of every department in the university is to maintain a database for all the applications of admission. This includes the personal information, academic information, documents received, GRE and TOEFL scores. The main issue the applicant usually faces will be to know the status of the application throughout the application processing. The final status will be intimated through post after a thorough review by admission committee of the department. If sometimes documents go missing, it would become difficult for the applicant to track his/her status. To overcome such problems, an idea to develop a website is proposed. Through this site the applicants can know their status at every step of their application processing.

1.3 PROJECT DESCRIPTION

A unique web application has been developed to enable the prospective students to access all the required information in an easy to use interface. The main aim of the website is to enable an easy and communication between the department staff and the applicant via an online system. The status of an application is maintained in this website.

This website basically has two main modules: 1. Admin module which gives privileges to department staff to maintain the database of the applicants and the currently enrolled students and to enter/edit the data present in the database. 2. Applicant module which gives the applicants the privilege to check for their status.

1.4 INTENDED USERS:

There are two main users for this website: The Administrator and the Applicants.

Administrators: The administrators have complete access to the website. They can enter, modify or remove information of an applicant from the database. The administrator also has the authority to update the status of an applicant and make the decision made by the committee available to various applicants. They can notify the applicant about their status through the system. Administrators also have privilege to generate student profiles from the applicant information. The different kinds of student information include grades, deficiencies, GPA, etc.

Applicants: The applicants can login to the system and view their details such as their personal details, academics details, test scores and deficiency courses along with the overview of their status.

Chapter 2 - BACKGROUND

2.1 PHP

PHP stands for Hypertext Preprocessor. It is a scripting language that is used to develop dynamic and interactive web pages. PHP scripts run on the server side and are embedded into the HTML source document. The PHP processor module then interprets this HTML document and generates the web page. This language can also be used in standalone applications and can be deployed on various web servers. It is platform independent and supports several databases such as Oracle, SQL Server, MYSQL, etc.

2.2 MYSQL

It is very important to have a back end database to which the application is connected. It makes it easier to store all the information inside a database. I have connected this application to MYSQL which is an SQL based relational database management system (RDBMS) that is supported by numerous operating systems. It stores all its information in database objects called as Tables. Organizing data in tables makes data retrieval or modification efficient. It adds speed and flexibility to the application. Tables can be linked to other tables and data can be combined as per requirement.

2.3 PHP+MYSQL

I have used PHP in conjunction with a MYSQL database as the combination of both increases stability, lower server costs on servers and the availability of ready to use Open Source Software Applications on the Internet.

2.4 Web Application Framework:

A web application framework is a software framework that helps in designing dynamic websites, web applications and web services. It helps in reducing the overhead caused by common activities performed during Web Development. There are different web application frameworks

out of which I have chosen to use PHP as it is Database driven, enables efficient user management, session management, code reusability and URI Mapping, etc.

There are five major types of architectures: request-based, component-based, hybrid, meta, and RIA-based. This application follows MVC architecture.

2.5 Model–View–Controller (MVC)

This application utilizes the model–view–controller (MVC) architectural pattern to separate the data model with business logic from user interface. This is generally considered a good practice as it modularizes code, promotes code reuse, and allows multiple interfaces to be applied.

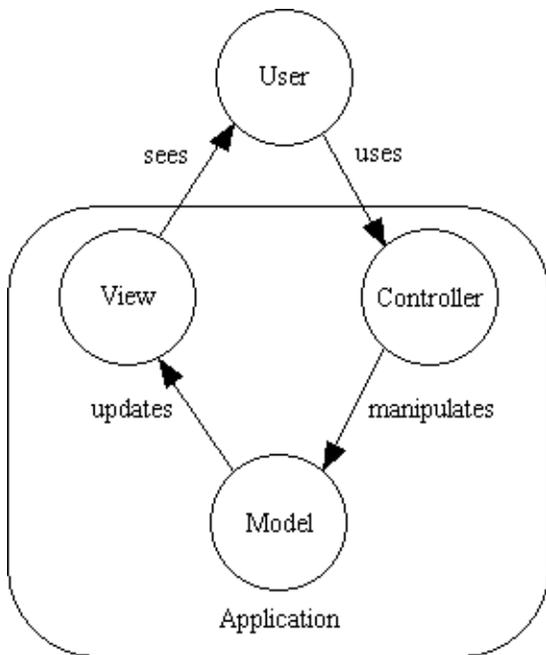


Figure 2.1 Working of MVC

Model: A model is an object representing data e.g. a database table or even some plant-floor production-machine process. The main work of the model is to manage the way data of the application domain works, how it responds to requests for information about its state and how it responds to instructions for a state change. The model represents part of the framework that consists of the data and rules that decide which data has to be accessed.

View: A view is renders the visual contents of model. It accesses data represented through the model and specifies how that data should be presented. The view is responsible for illustrating graphics into an application. It maintains a connection to the model, extracts information from it and displays it using various visual components of the framework.

Controller: A controller is part of framework that enables communication between the user and the application. A controller accepts input from the user and instructs the model and view to perform actions based on that input. It takes the response provided by the view and takes an appropriate action based on the requirement. It translates requests sent by the view to the model which in turn performs the action. Actions can be off different nature. They can be either button click or menu selection events or basic HTTP GET or POST requests. The actions performed by the model include invoking various processes or changing the state of the model. Based on the user interactions and the outcome of the model actions, the controller responds by selecting an appropriate view. It provides the mechanism by which changes are made to the state of the model.

2.6 RELATED WORK:

Many PHP frameworks have been evolved ever since PHP is under extreme usage. There is a great debate about which PHP framework to be used that is because of the simple fact that every framework its own advantages and disadvantages. These are the three frameworks considered as a base for the current project.

- 1) CAKE PHP: It is a good choice for beginners as well advanced PHP developers. It's based on principles that are similar to Ruby on Rails and this framework is heavily focused on rapid development. Its rapidly growing support system, simplicity, and scalability make CakePHP one of the popular PHP frameworks available today.
- 2) CODEIGNITER: It is well-known for its ease-of-use, performance and speed. This PHP framework is ideal for shared hosting accounts or for smaller projects. It is simple to use, and has a good library of video tutorials, forums, a user guide available for support.
- 3) Yii: Yii is a Web application development framework written in PHP5 that promotes clean, DRY design and encourages rapid development. It helps to ensure an extremely

efficient, extensible, and maintainable end product. Being extremely performance optimized, Yii is a perfect choice for any sized project with any sophisticated requirements.

Framework	PHP 5	MVC	Multiple DB's	ORM	DB Objects	Templates	Caching	Ajax	Auth Module	Modules
CakePHP	✓	✓	✓	✓	✓	-	✓	✓	✓	✓
CodeIgniter	✓	✓	✓	-	✓	✓	✓	-	-	-
Yii	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 2.1 Comparison of framework

- **MVC:** Indicates whether the framework comes with inbuilt support for a Model-View-Controller setup.
- **Multiple DB's:** Indicates whether the framework supports multiple databases without having to change anything.
- **ORM:** Indicates whether the framework supports an object-record mapper, usually an implementation of ActiveRecord.
- **DB Objects:** Indicates whether the framework includes other database objects, like a TableGateway.
- **Templates:** Indicates whether the framework has an inbuilt template engine.
- **Caching:** Indicates whether the framework includes a caching object or some way other way of caching.
- **Ajax:** Indicates whether the framework comes with inbuilt support for Ajax.
- **Auth Module:** Indicates whether the framework has an inbuilt module for handling user authentication.
- **Modules:** Indicates whether the framework has other modules, like an RSS feed parser, PDF module or anything else (useful).
- **EDP:** Event Driven Programming.

By comparing among different important attributes required. Yii has out taken the remaining frameworks. Hence YII framework has been chosen for the development of the project.

Chapter 3 - IMPLEMENTATION

3.1 SYSTEM ARCHITECTURE: MVC

Model–View–Controller^[3] (MVC) is a software architecture, currently considered an architectural pattern used in software engineering. The pattern isolates "domain logic" (the application logic for the user) from the user interface (input and presentation), permitting independent development, testing and maintenance of each (separation of concerns).

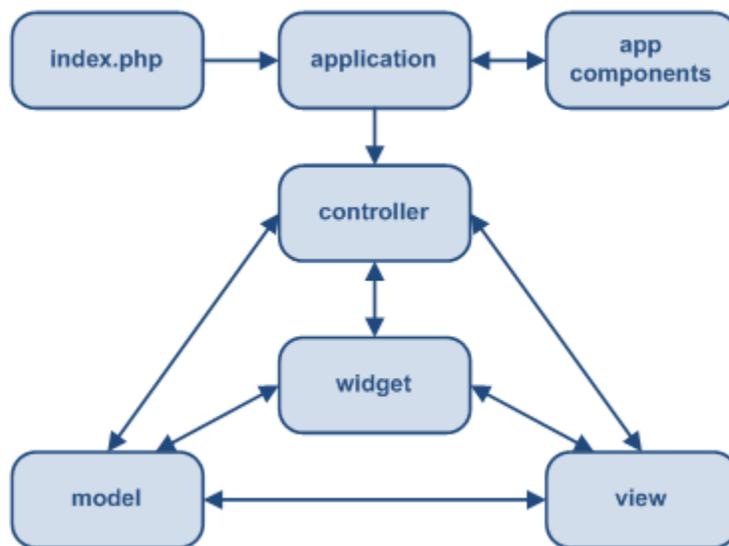


Figure 3.1 MVC System Architecture

Though MVC comes in different flavors, control flow is generally as follows:

- 1) The user interacts with the user interface
- 2) The controller handles the input event from the user interface, such as a registered handler or callback, and converts the event into an appropriate user action, understandable for the model.
- 3) The controller notifies the model of the user action, possibly resulting in a change in the model's state.
- 4) A view queries the model in order to generate an appropriate user interface. The view gets its own data from the model. In some implementations, the controller may issue a

general instruction to the view to render itself. In others, the view is automatically notified by the model of changes in state (Observer) that require a screen update.

- 5) The user interface waits for further user interactions, which restarts the control flow cycle.

3.2 3-Tier VS MVC Architecture



Figure 3.2 3-Tier VS MVC Architecture

A 3-tiered system^[4] really is made up of three layers. The UI Layer has access to the Business Logic Layer, and the Business Layer has access to the Data Layer. But the UI Layer cannot directly access the Data Layer. In order for the UI Layer to access data, it must go through the Business logic Layer via some kind of interface. If it helps, you could think of each layer as one big loosely coupled component with strict design rules of access between layers.

MVC (Model View Controller)

In contrast, the MVC pattern^[5] obviously does not keep a layered system. The Controller accesses the Model (a runtime data repository) and the View. The View then accesses the Model. Typically, the Controller will retrieve, build, or modify a Model base on some triggered action. The Controller then decides which View is appropriate via some internal logic. At that point the Controller will push the Model to View.

3.3 SYSTEM DESIGN

Based on the gathered requirements, the various components, modules, interfaces and interactions of the system have been designed including different components, interfaces and packages. These are the prototypes that are built and are reviewed by the users. Various modules have been designed to describe the relation between types of users and the different entities. Two diagrams have been discussed here from numerous diagrams that have been used to identify inputs and outputs of the system:

- Use Case Diagram
- Class Diagram

These diagrams are discussed in detail in the following sections.

3.4 USE CASE DIAGRAM:

Use Case diagram is a behavioral diagram that defines a use-case analysis of the complete system. It represents a graphical over view of the functionality of the system in terms of actors, which are persons, organizations or external system that plays a role in one or more interaction with the system. These are drawn as stick figures. The goals of these actors are represented as use cases, which describe a sequence of actions that provide something of measurable value to an actor and any dependencies between those use cases.

The associations between actors and use cases are indicated in use case diagrams by solid lines and this exists whenever an actor is involved with an interaction described by use case.

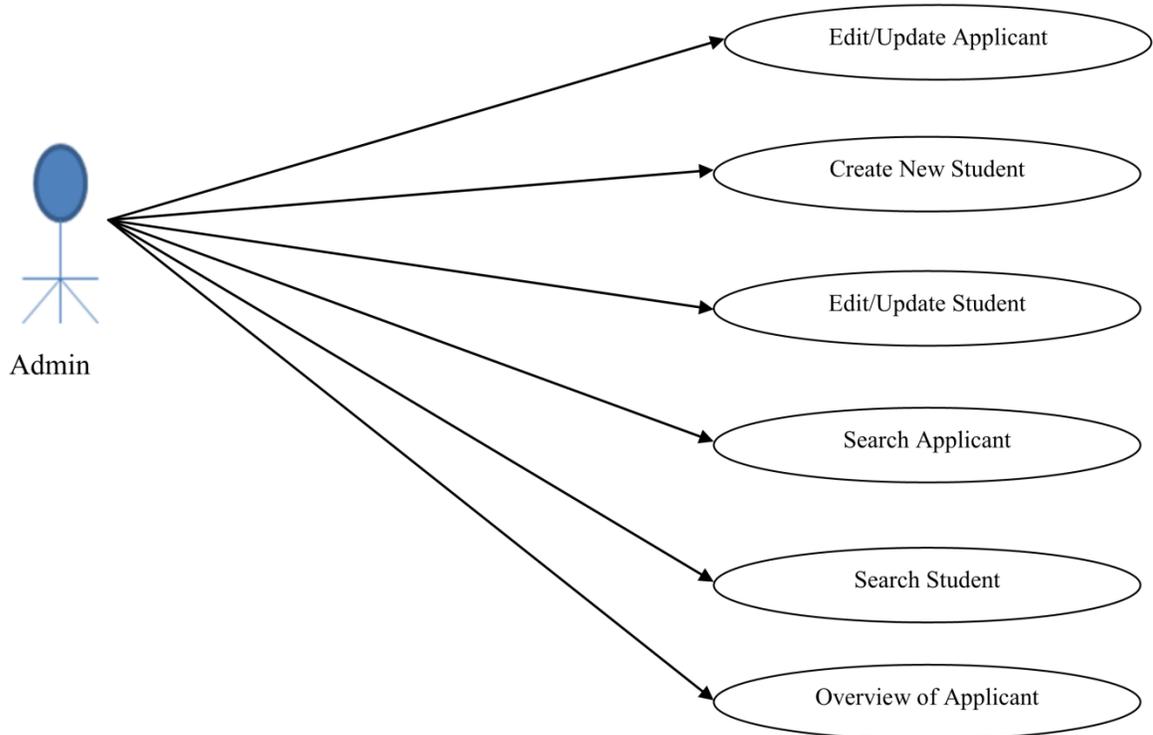


Figure 3.3 UseCase for Admin

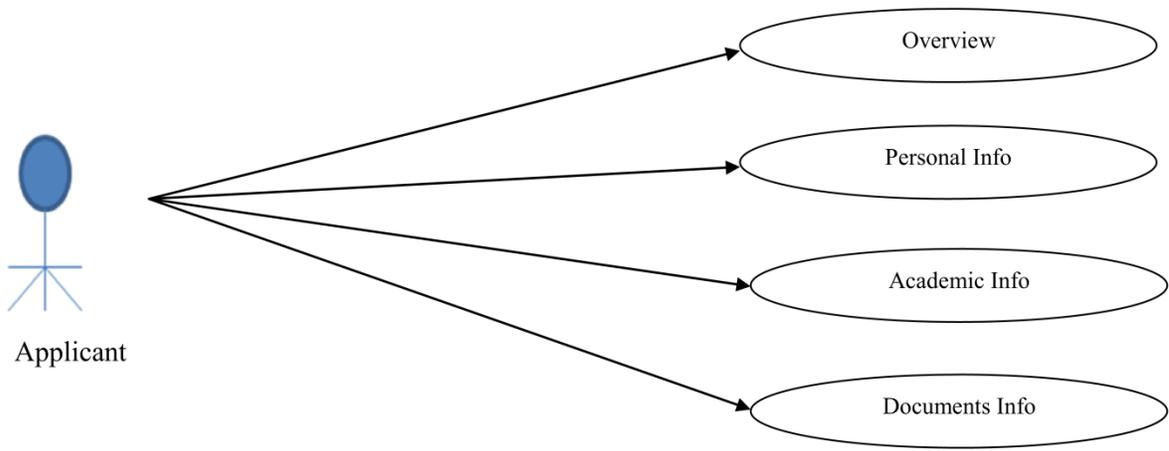


Figure 3.4 Use Case for Applicant

3.5 CLASS DIAGRAMS

In the UML, class diagram is a type of static structure diagram which describes the structure of a system by showing the classes, attributes and the relationships between the classes of a system. It is the main building block in object oriented modeling. The classes represent both the main objects and or interactions in the application. The class diagram consists of classes which are represented in boxes which contain three parts. The name of the class is contained in upper part, with the attributes of classes in middle part and the bottom part contains the methods or operations that the classes undertake. With detailed modeling, the classes of the conceptual design are split into number of subclasses.

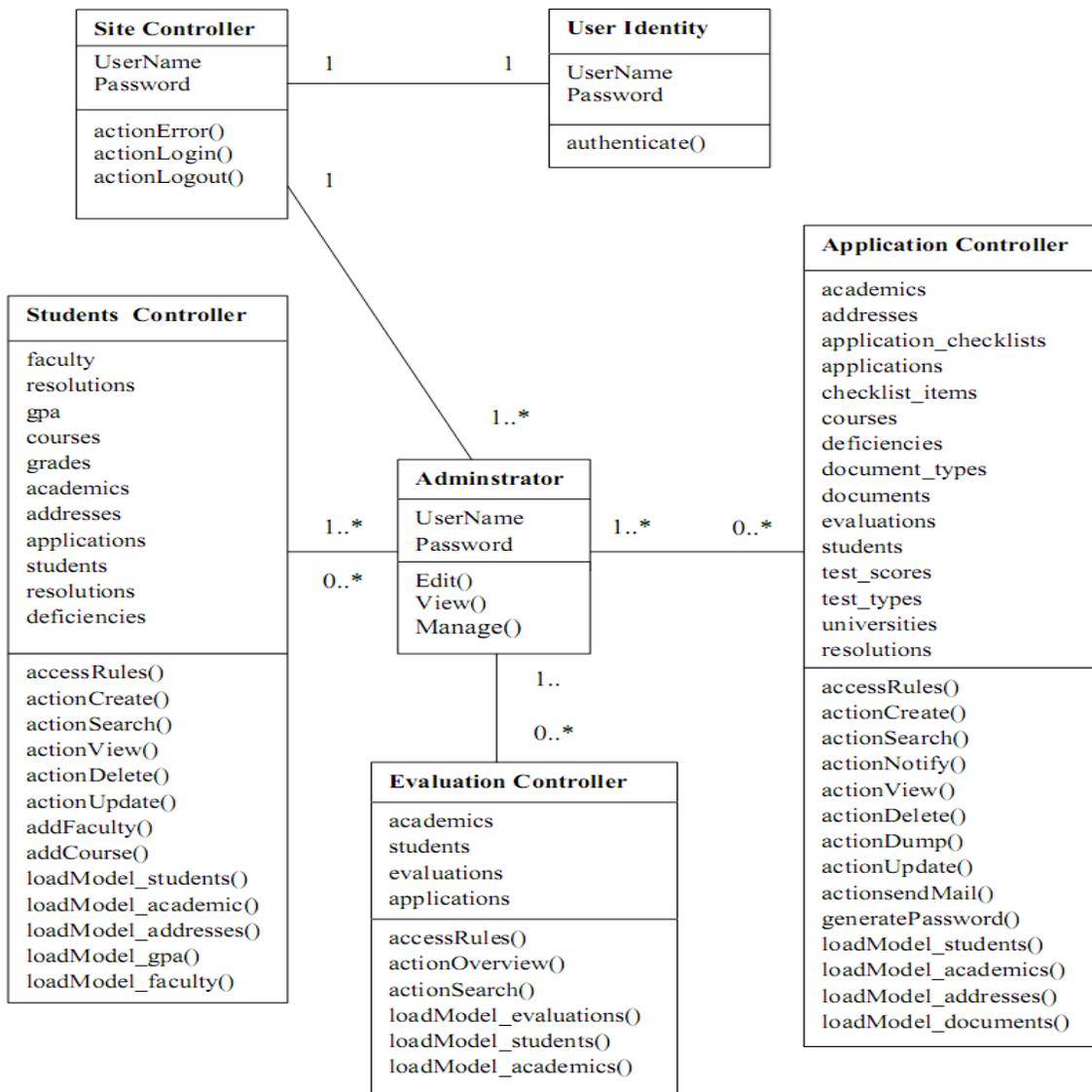


Figure 3.4 Class Diagram from Admin View

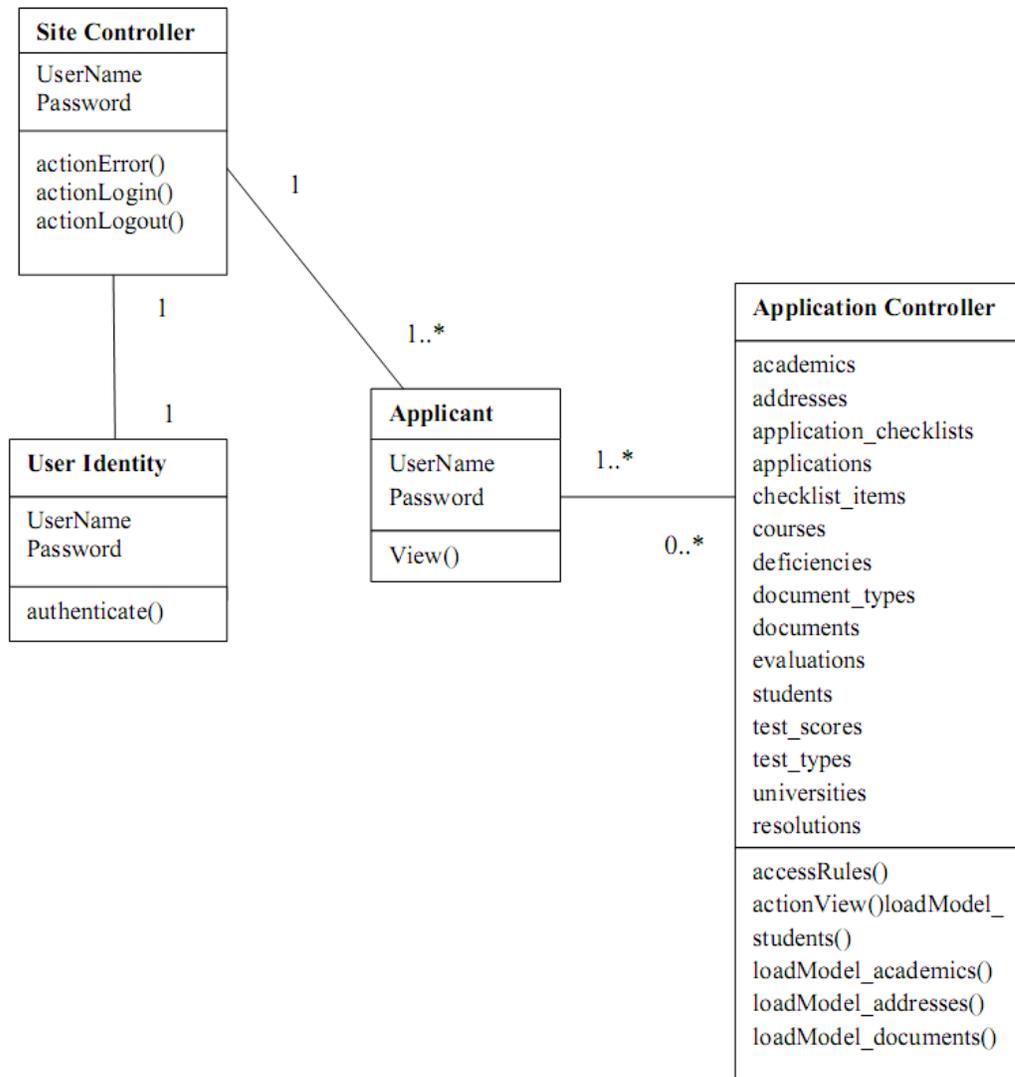


Figure 3.5 Class Diagram from Applicants view

3.6 Working

The application is designed using PHP framework called YII. MySQL is been used for storing information. Advanced technologies such as AJAX, JQuery has been extensively used to make the interaction with the interface much easier. CSS has been used to format the web pages and also creation of layout of the website. This website is compatible with browsers such as Internet Explorer, Mozilla Firefox, Google Chrome, Safari etc. Controls such as datagrids, dropdownlists

have been used to display the information in an organized format and maintain input data consistency.

Database Schema: The database consists of 21 tables. Each table has a primary key and one or more foreign keys.

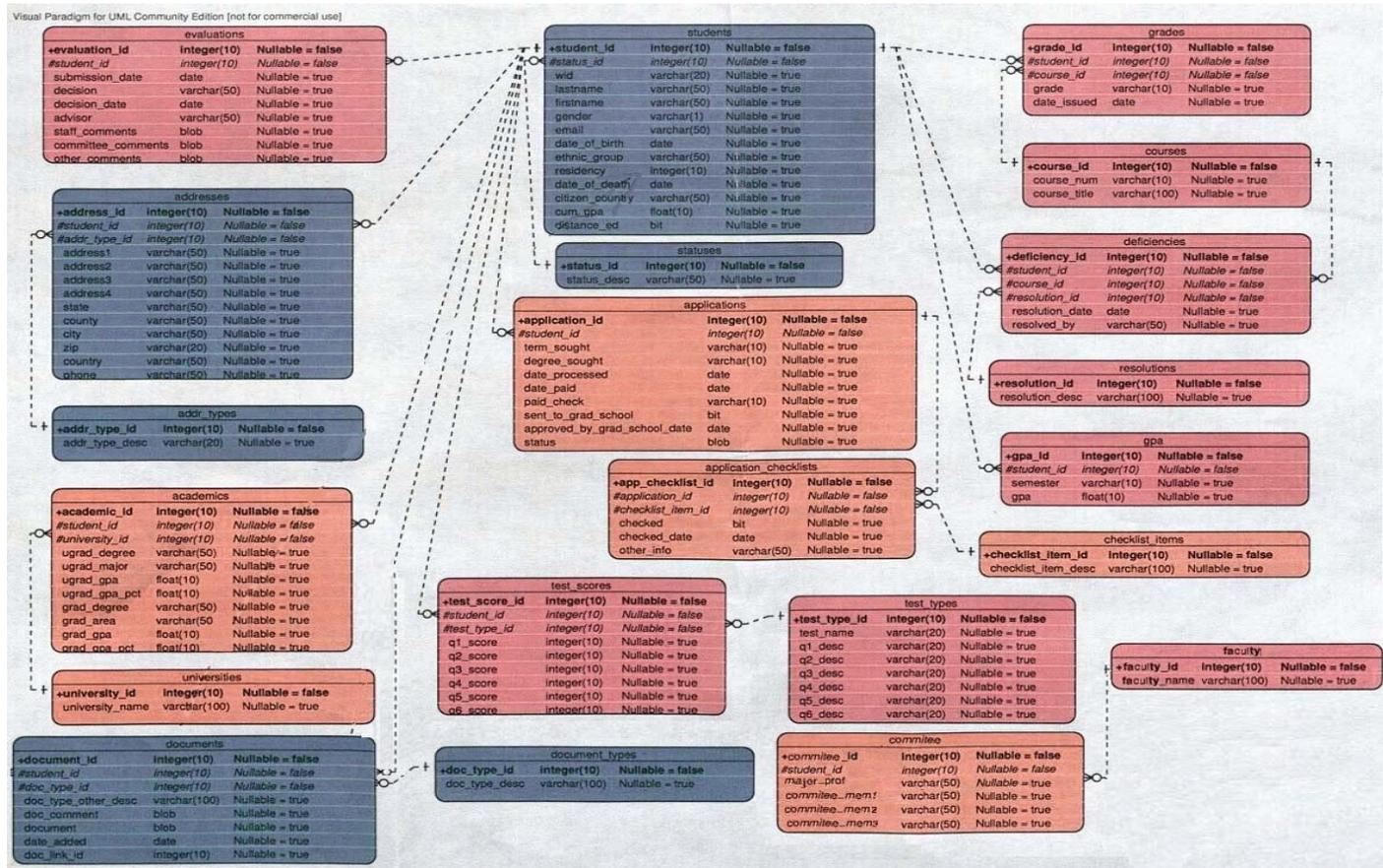


Figure 3.6 UML Diagram

Students table: Students table is the main table which serves as the pivot point to connect to other tables. Primary key of this table is student_id. General information such as first name, last name, date of birth etc are the fields in the table.

Addresses table: This table consists of the details of address such as street name, state, county, zip with primary key as address_id and two foreign keys student_id and addr_type_id.

Applications table: This table consists of the details of the application such as application Id, term sought, degree sought, paid method with application_id as primary key and student_id as foreign key.

Checklist Items: This table is used to store the checklist item id and its description. This table has been created in order to avoid redundancy of data.

Application Checklists table: This table utilizes the data that is present in the Checklist Items table and identifies which item is checked and stores its details such as the checked date, application id, etc. app_checklist_id is the primary key of this particular table. The application id is a foreign key and is used to relate with Applications table. The checklist_item_id is another foreign key that relates with Checklist Items table.

Academics table: This table stores information about the academics of prospective students or graduate students or PhD students such as the student's major, degree, university etc. The primary key for this table includes academic_id. Foreign Keys for this table are student_id and university_id.

Evaluation table: This table is mainly used to store information about a student whether his application for enrollment has been approved or not. It also stores data about a student's advisor, staff comments, committee comments and decision date. The primary key for this table is evaluation_id and the foreign key for this table is Student_id.

There are several other tables inside the schema. There are tables that store the information about the different tests taken such as GRE, TOEFL etc. (Test_Scores, Test_Types), information about different documents (Document_Types, Documents), information about the GPA of currently enrolled students (gpa), information about courses (Courses), information about deficiency courses (Deficiencies), information about resolutions that resolve deficiencies (Resolutions), information about grades for different courses (grades), information about faculty(faculty) and about the committee members(committee) etc.

This project is handled in a modular approach. Different modules used in the project are:

Applicant module: In this module there is creation of applicant record, updating applicant record, search applicants, overview of applicant, List of applicants.

a) Creation of Applicant record: There are different sections such as personal information, academic information, addresses, evaluations, deficiencies, application details, documents uploaded etc. JQuery is used to apply the feature of updating mailing address with the permanent address. Under academics section AJAX is used for adding new university name to the database without reloading the page. Here, a particular university can be added dynamically to the dropdownlist of Universities. Multiple uploading of documents has been implemented where the admin can upload upto 15 documents at a time. I have used Toggle function in jQuery to make the university fields visible on clicking the “Add University” button. When the request is submitted, a series of ajax functions such as createurl are executed and the database values are inserted into the respective tables. The dropdownmenu is invoked using CHtml::encode and CHtml::tag functions. All this is done in the backend without the whole page being reloaded. This whole functionality enables postback of a page without disturbing the values already entered.

CHtml::activeFileField is used to pop up the choose file button to upload the desired file. Functionality to upload multiple files has been implemented in this page using CUploadedFile::getInstance inside a foreach loop running over an array of documents to be uploaded.

A feature to send email dynamically to the person when his application is created has also been implemented successfully using mail() function in PHP.

b) Update Applicant record: This also consists of sections similar to applicant creation page where the Admin can edit the fields and save it to database. More documents can be uploaded and the documents uploaded in the Application creation page can be viewed here. There are buttons for sending notification to the applicant and generate student record from the current applicant record. Top and bottom hyperlinks have been provided for easy navigation.

c) Search Applicant record: Gridview have been used to display the list the application with buttons view, update and delete. Admin can search for the applicant using the Application ID, Last Name, First Name, Date of Birth fields. By clicking on the buttons view the applicant’s view can be seen. Update button will redirect to the particular applicant update page. Delete button is used to delete the record from the database.

- d) Overview of Applicant: In the overview of applicants the review form of the applicant is generated dynamically in Microsoft word. The applicants can be filtered using the Application ID and Date to Committee fields.
- e) List of applicants: List of all applicants is been displayed in the ascending order of Application ID. Filtering of the applicants is done by using Application ID, Term Sought and Degree Sought fields.

Student module: In this module there are features such as creation of student, search students, upload CSV.

- a) New Student: Student record can be generated from the update page or by creating an entirely new record. This consists of different sections such as personal information, grades obtained, courses taken, deficiencies present, list of Committee members and Current GPA. Ability to add new course name, new faculty name has been provided. This is achieved using JQuery and AJAX.
- b) Searching Student: Student records are searched by WID ID, First Name, Last Name fields.
- c) Upload CSV: CSV files can be uploaded where the grades, courses of all the students in the current semester are present, Where the parsing of CSV files are done and grades, courses are been added to the database for all the students listed in the document. I have used CUploadedFile::getInstance method to upload documents to the server. Then the file is opened and later fgetcsv() is used to parse data present in each cell in the excel sheet. Only the required information such as grade and course is dumped into the database.

Applicant's view:

Applicant can login to the system by using the credentials sent to him through the mail. After login the applicant can see the list of received documents and list of missing documents. Tabs are provided to view personal information, academic information and deficiency information. Different layout has been used for the applicant's view. Personal information will provide his first name, last name, gender, email, date of birth. Address info will give the street number, city, country, zip, phone number. Academics Info will give the Undergrad Degree, Undergrad Major, Undergrad GPA. Deficiency information will give the list of deficiency courses.

Ajax Validation has been used for validating the fields in the login page. By using these validation controls the user will feel the dynamic interaction with the application. JQuery have been used to give sliding effects.

Using Yii:

Installing Yii involves download and unpacking the Yii release file to a web-accessible directory. To ensure the proper installation, verify that server satisfies Yii's requirements by accessing the requirement checker script by entering the URL in a web browser:

<http://hostname/path/to/yii/requirements/index.php>

Establishing the connection with the database: To establish the connection with the database change the configuration file with required connectionString.

```
return array(  
    .....  
    'components' => array(  
        .....  
        'db' => array(  
            'connectionString' => 'mysql:protected/data/testdrive.sql',  
        ),  
    ),  
    .....  
);
```

Yii is a powerful framework which gives a quick start to develop the application to avoid the start from the scratch. Yii has built in yiic a command line tool which generates the basic controllers, views and models depending on the tables defined in the database. This gives an instant start from where adding new functions to the controllers, models and changing the views according to the requirements. Functions to generate new applicant, to update an existing applicant, to delete, search have been defined. Similarly for the currently enrolled students similar functions have been defined. Below are the figures of the built application:

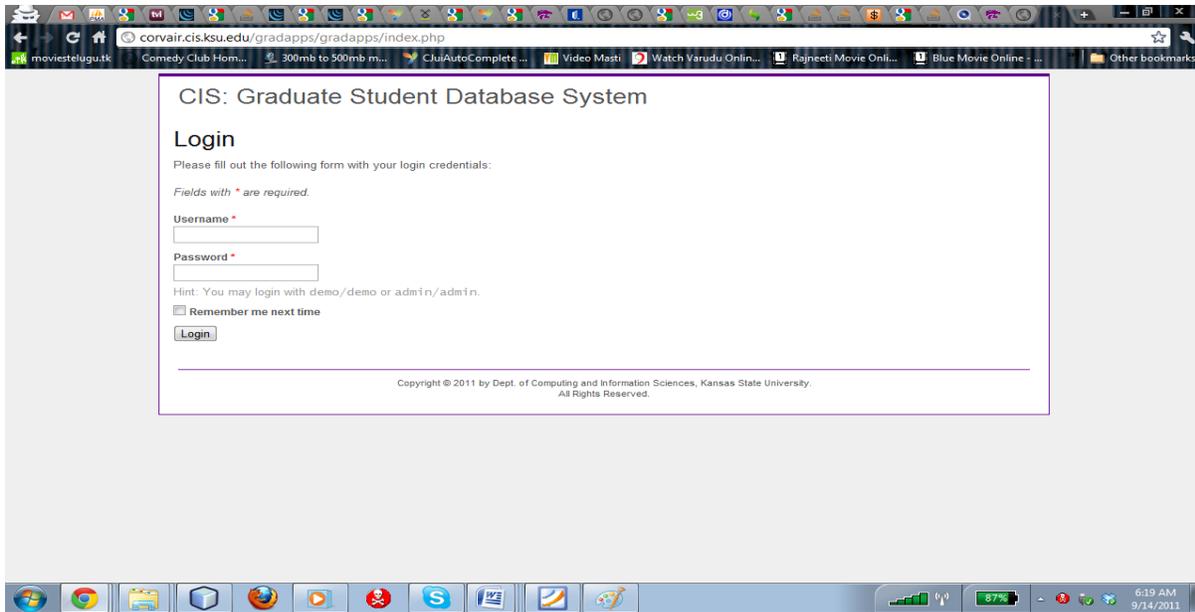


Figure 3.7 Login Screen

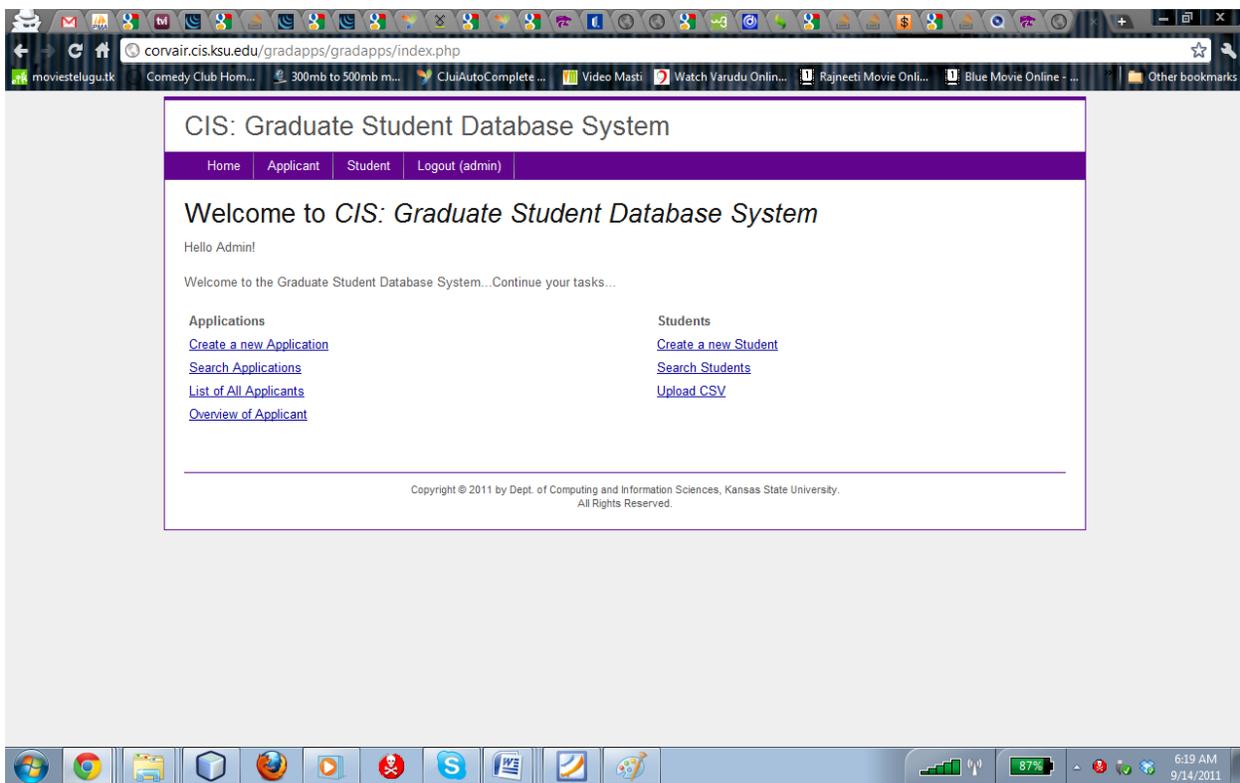


Figure 3.8 Welcome Page

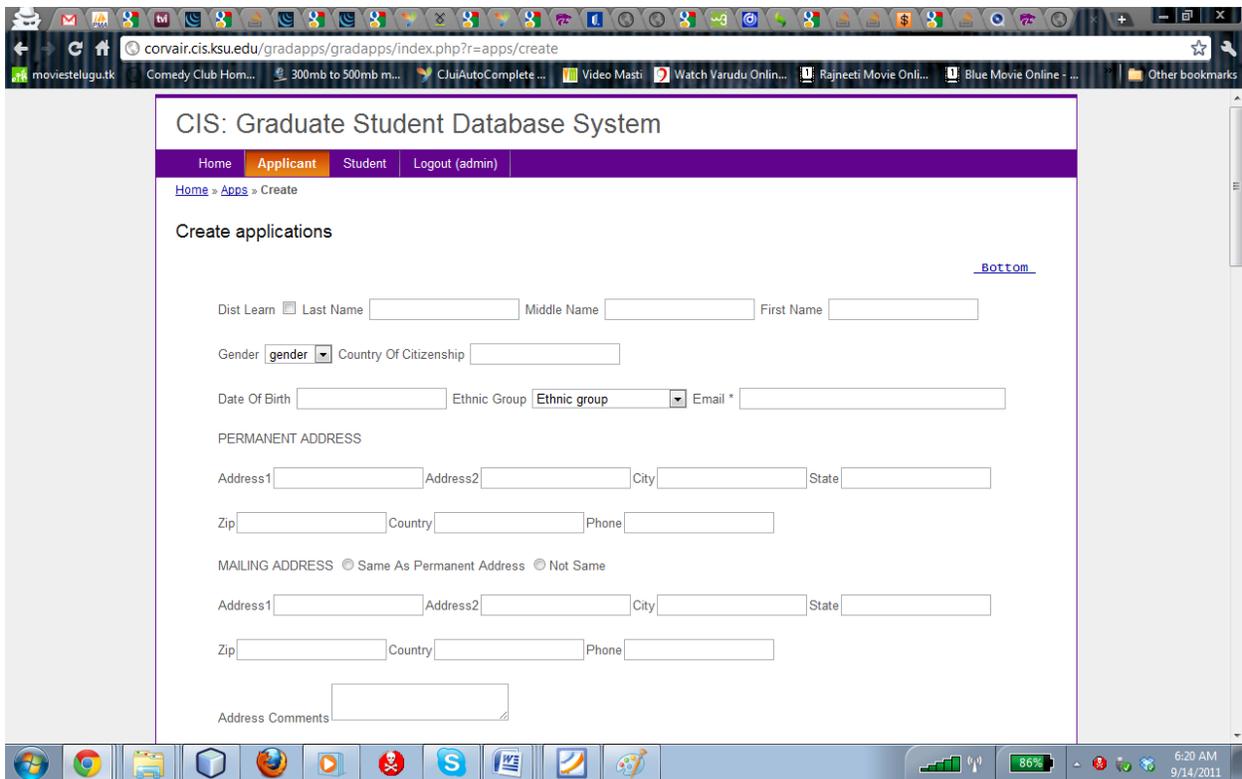


Figure 3.9 Create New Application Page

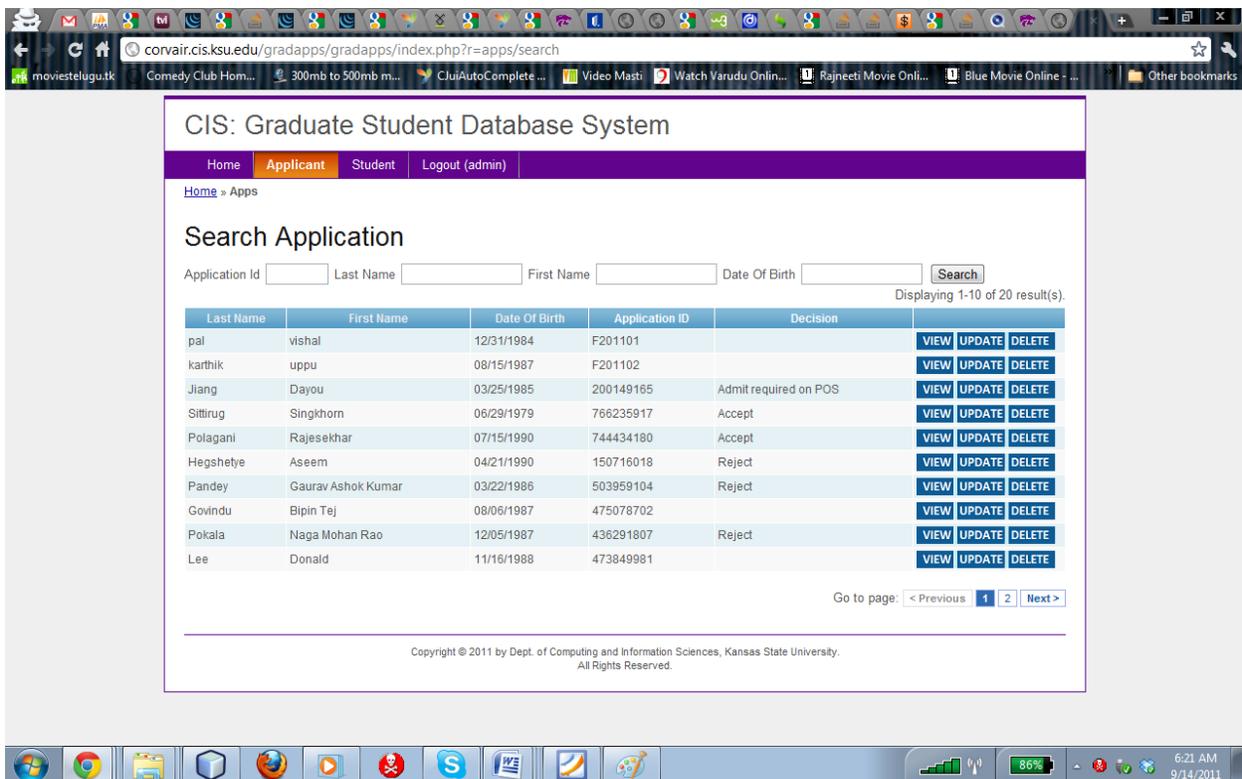


Figure 3.10 Search Application

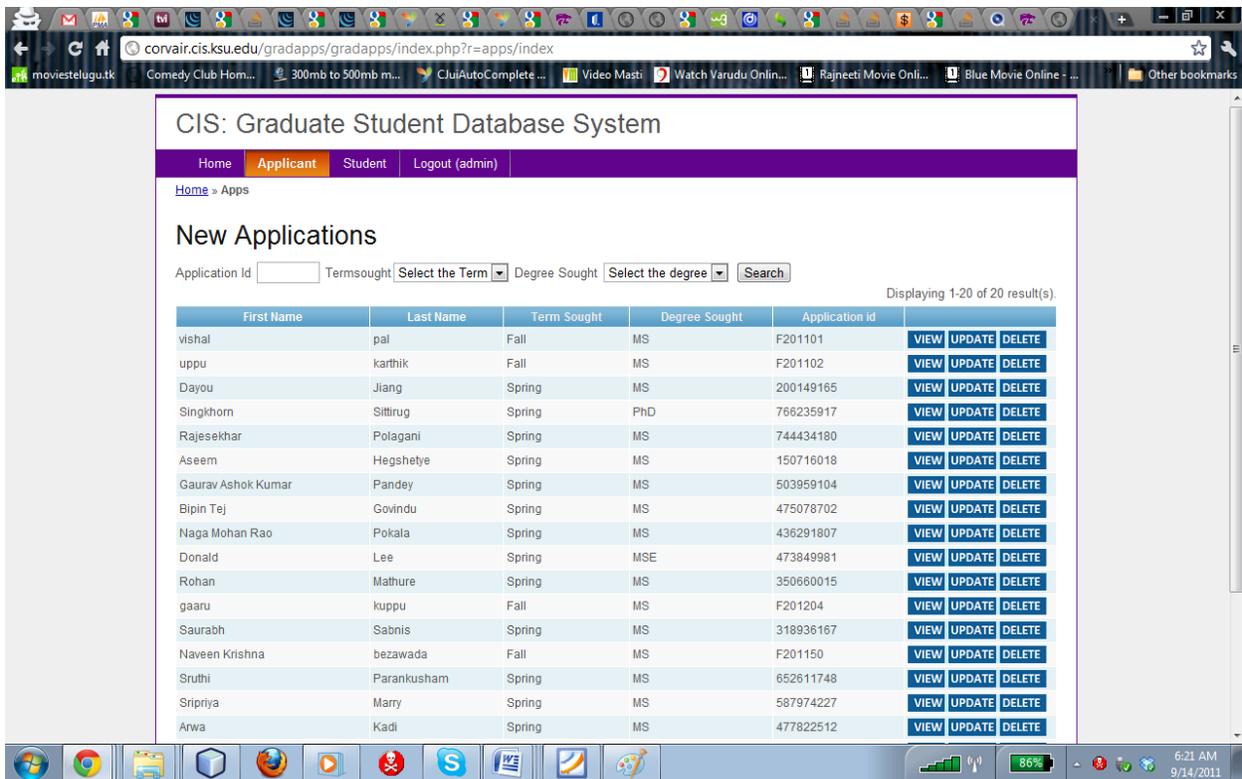


Figure 3.11 List of Applications

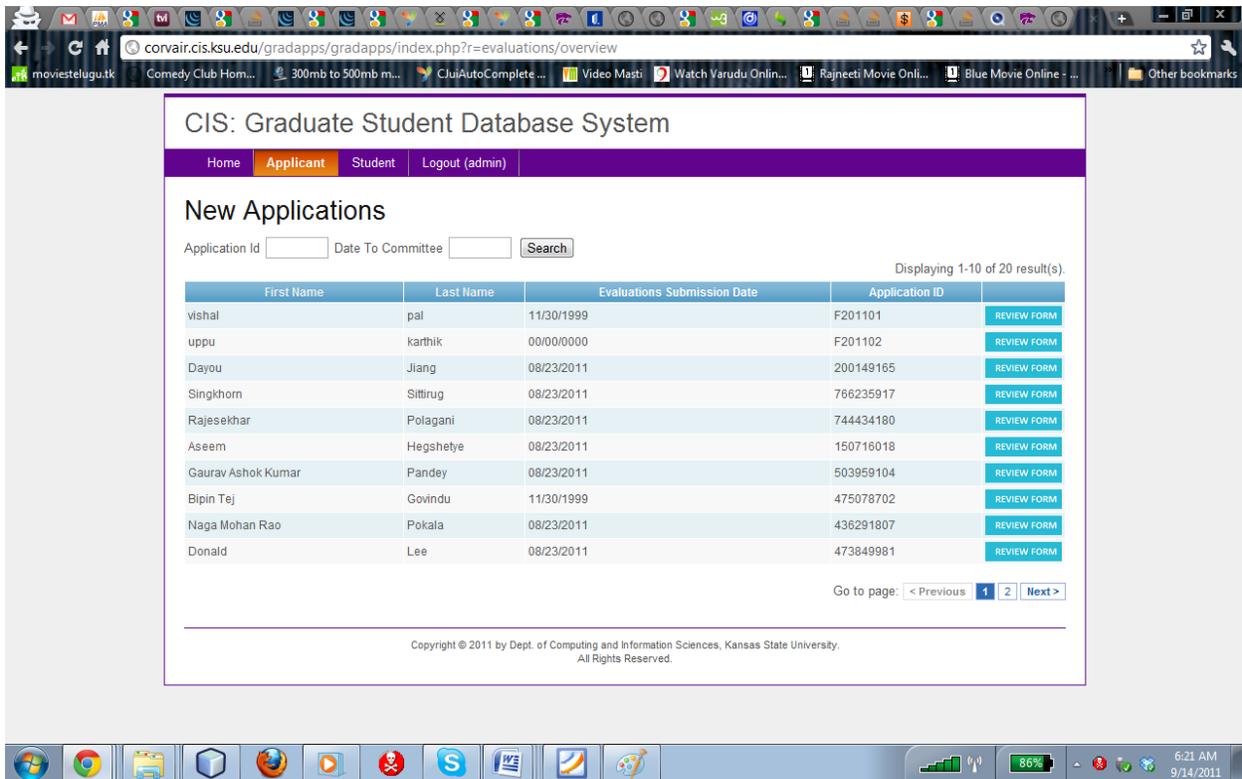


Figure 3.12 Overview Page

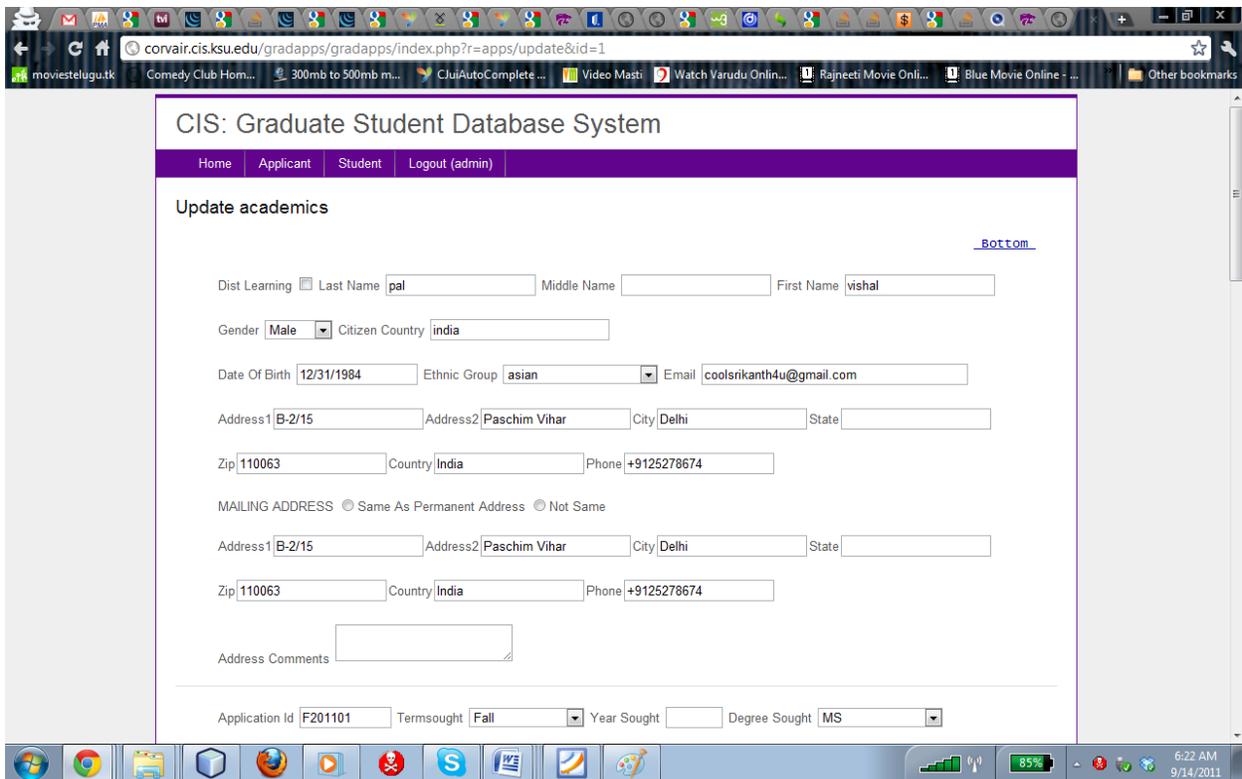


Figure 3.13 Update Applicant

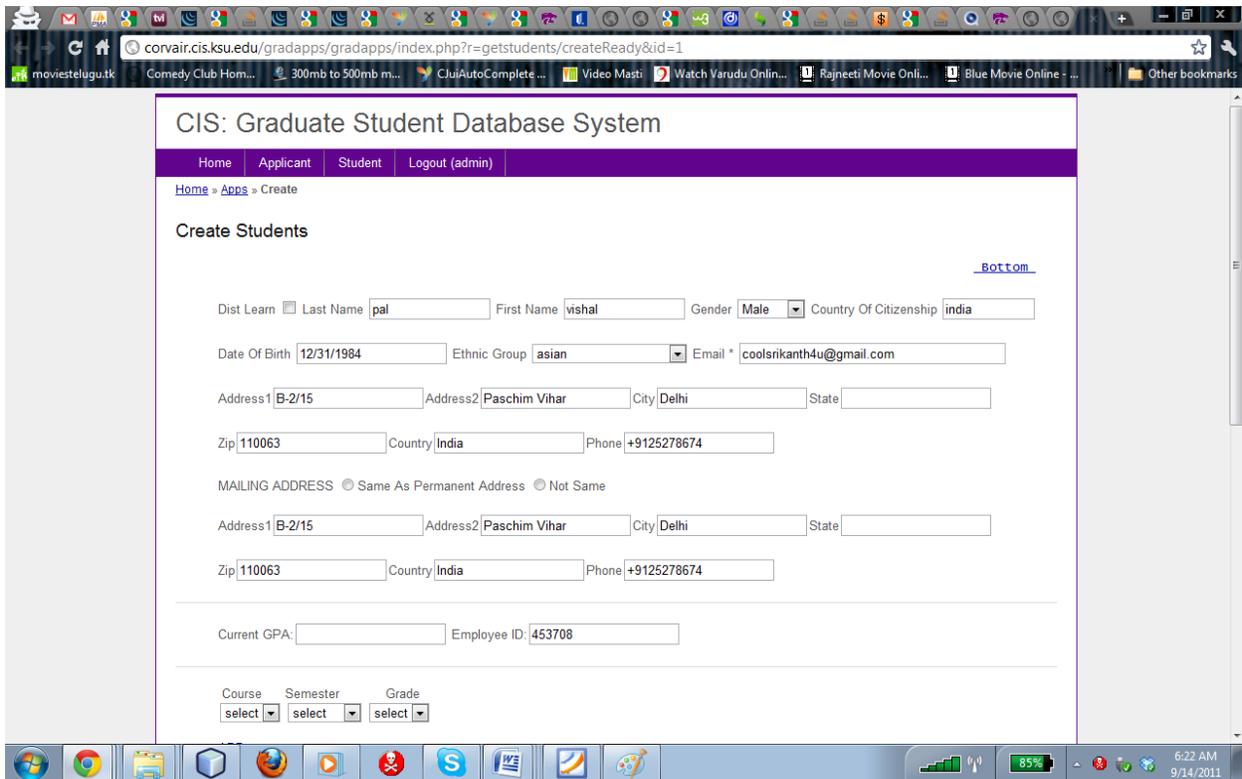


Figure 3.14 Generate Student Profile

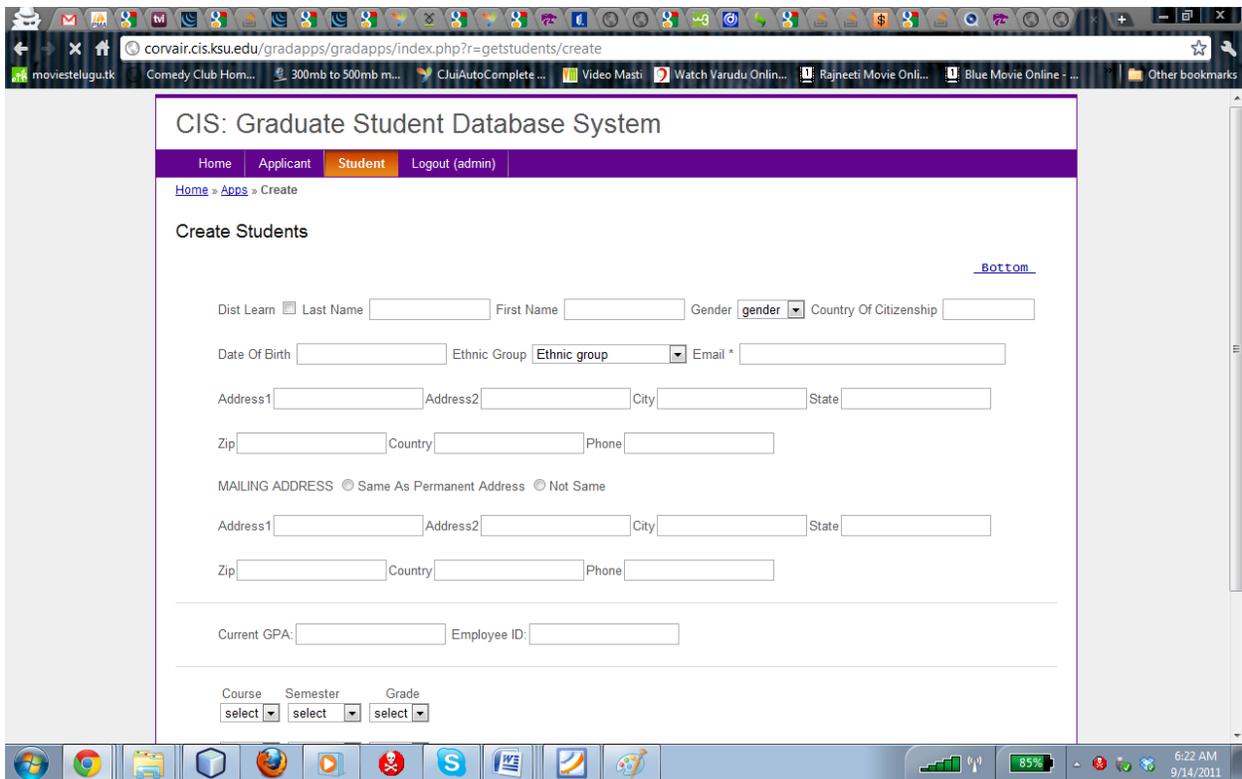


Figure 3.15 Create New Student Page

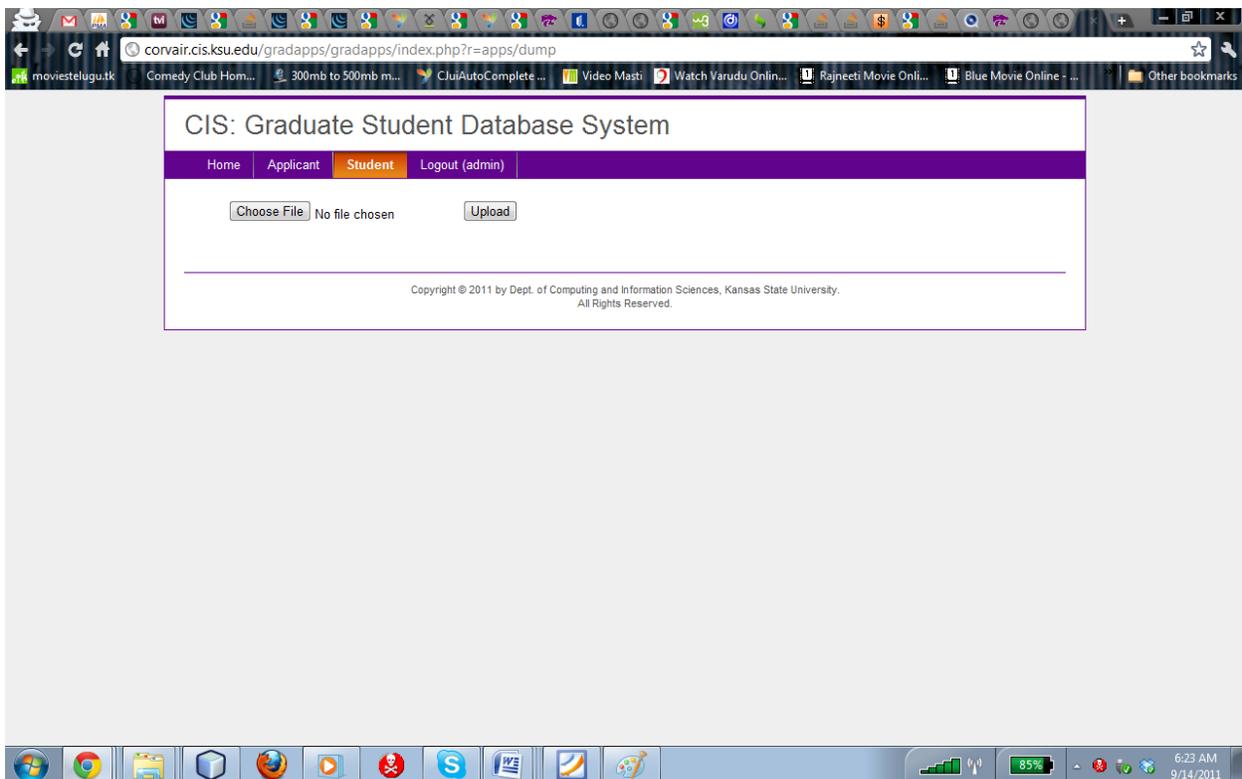


Figure 3.16 Upload CSV File Page

Chapter 4 - TESTING

Software Testing

Software testing is performed to verify that the completed software package functions according to the expectations defined by the requirements/specifications. The overall objective is not to find every software bug that exists, but to uncover situations that could negatively impact the customer, usability and/or maintainability. Testing allows developers to deliver software that meet expectations, prevents unexpected results, and improves the long term maintenance of the application. Depending upon the purpose of testing and the software requirements, the appropriate methodologies have been applied.

4.1 Unit Testing

Unit testing is a method by which individual units of source code are tested to determine if they are fit for use. A unit is the smallest testable part of an application. It is a development procedure where programmers create tests as they develop software. The tests are simple short tests that test functionality of a particular unit or module of their code, such as a class or function.

Yii testing framework is built on top of PHPUnit, basic principles of writing a unit test in Yii are:

- A unit test is written in terms of a class `XyzTest` which extends from `CTestCase` or `CDbTestCase`, where `Xyz` stands for the class being tested. For example, to test the `Post` class, we would name the corresponding unit test as `PostTest` by convention. The base class `CTestCase` is meant for generic unit tests, while `CDbTestCase` is suitable for testing active record model classes.
- The unit test class is saved in a PHP file named as `XyzTest.php`. By convention, the unit test file may be stored under the directory `protected/tests/unit`.
- The test class mainly contains a set of test methods named as `testAbc`, where `Abc` is often the name of the class method to be tested.
- A test method usually contains a sequence of assertion statements (e.g. `assertTrue`, `assertEquals`) which serve as checkpoints on validating the behavior of the target class.

4.2 Performance Testing

In software engineering, performance testing is the testing that is performed to determine how fast some aspect of a system performs under a particular workload. Performance Testing is used to verify performance behaviors for business functions under the normal and heavy work conditions. The success criteria of this test are completion of all the test cases without any failures and within acceptable time allocation. Performance Testing is aimed at assessing the speed at which the product addresses different events under different conditions. In this application I have used Apache JMeter as a tool to enable performance testing. I have run performance testing in JMeter for 3 cases: 500 users, 1000 users and 2000 users. Below are the Figures of sample JMeter test-cases and analysis of their results:

a) Jmeter results for creation of application:

The creation of application page is tested to analyze its behavior and to get the idea about the performance of the application. The analyses would show the throughput as well as response times of the web page when number of users is accessing it. This test plan is also used to get an approximate estimation of maximum numbers of users who can access it at the same time.

In order to test the page, total of 500 threads, simulating 500 simultaneous users, with a ramp period of 10 seconds, with a total of 500 requests. Since the strength including applicants and faculty all together will not cross more than 500 that is the reason for choosing 500 threads.

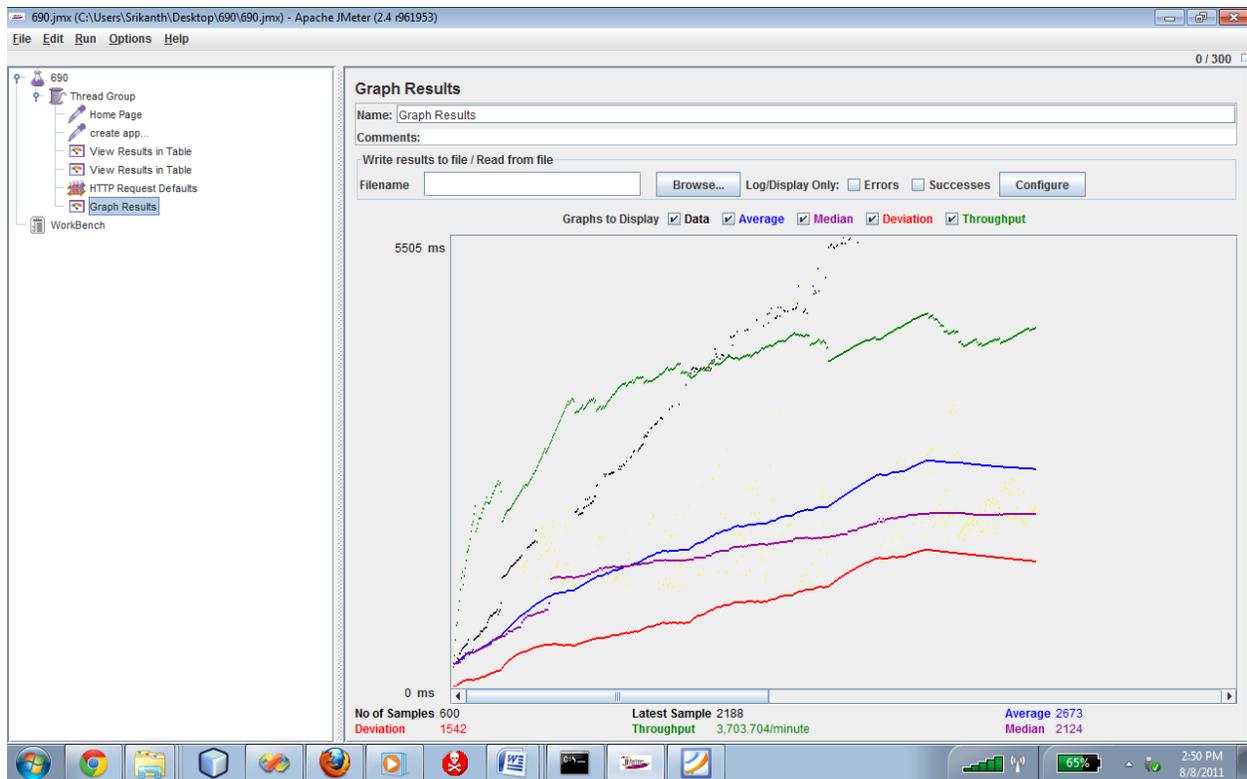


Figure 4.1 JMeter Graph Results for Create New Application

It is clear from the graph that the website performs consistently under heavy load generated by many simultaneous users. The throughput was around 3700, which indicates that the website can handle 3500 requests per minute.

Bottle Neck for create applicant page.

- Using JQuery plug-in can be considered a hindrance for lesser throughput.
- Due to big form with several divisions is one hindrance for the less number of requests per minute and more average time.
- CPU utilization can be considered as a bottle neck when more than 400 requests per minute.

b) JMeter results of search applicants:-

This page is also tested with similar test case specifications that are 500 threads in total, simulating 500 simultaneous users, with a total of 500 requests. This has got a throughput approximately equal to 2263 request per minute and which is less than creation of applicant page.

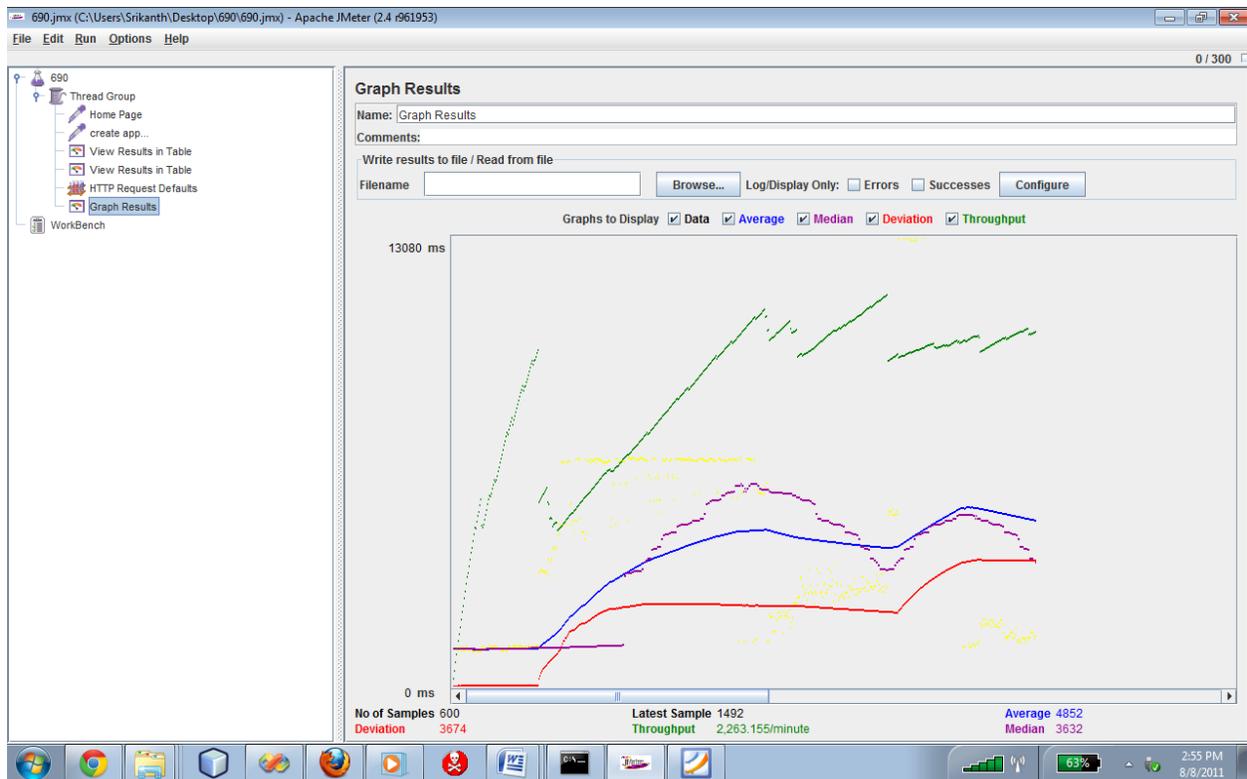


Figure 4.2 JMeter Testing for Search Page

Bottle Neck for search applicants:

- Due to loading of all the records in the gridview which has to deal with the database the throughput is considerably less.
- JQuery plug-in has been added which can be considered as one of the factors.

c) Results for student profile creation:

This page is tested by running test case with following test specifications 500 threads in total, simulating 500 simultaneous users, with a ramp period of 10 seconds, with a total of 500 requests. This has got a throughput approximately equal to 2287 request per minute. The average time 48 which is the least compare to the creation of application page because there are less number of divisions when compared.

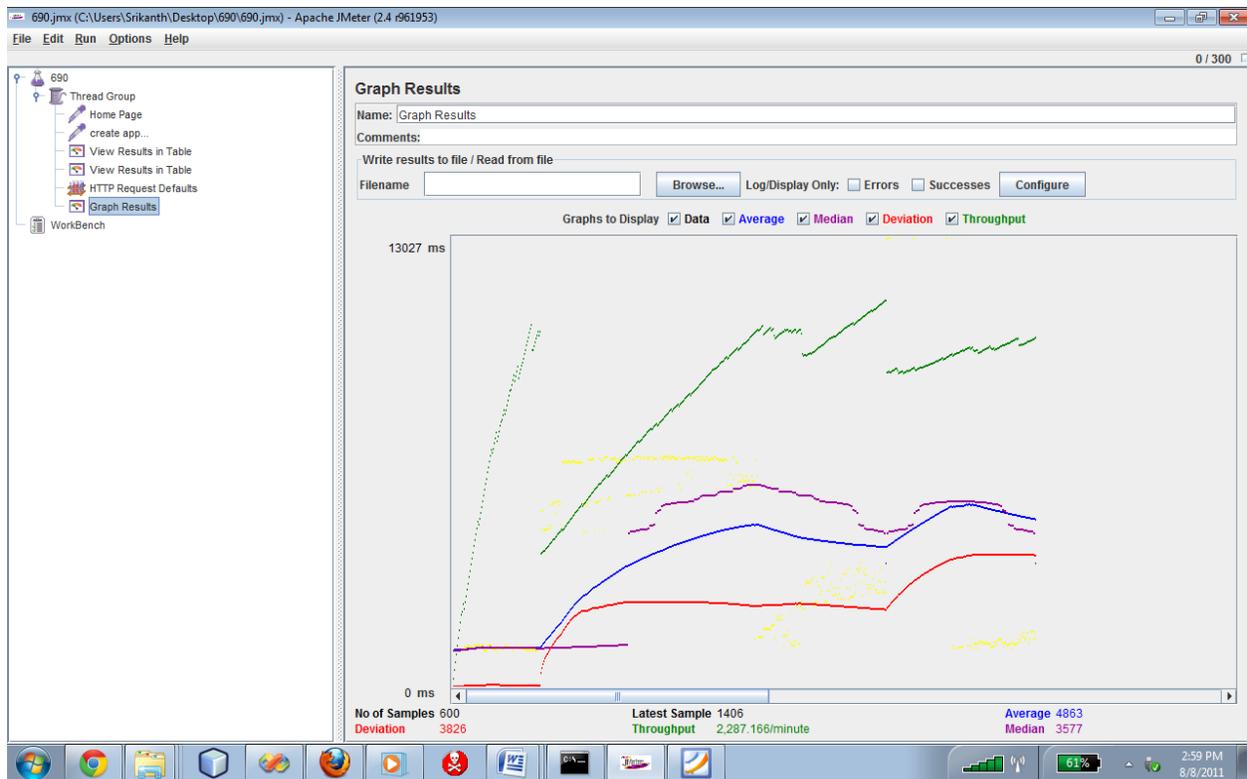


Figure 4.3 JMeter Testing for Student Profile Page

Bottleneck for creation of student profile:

- Due to usage of JQuery plug-in in the form the throughput might have been effected.
- This page consists of several divisions which might be considered as one of the factors for lesser throughput.

d) Results for Search Student:

Testing on this page shows similar results to the search applicants. Same number of test threads have been used.

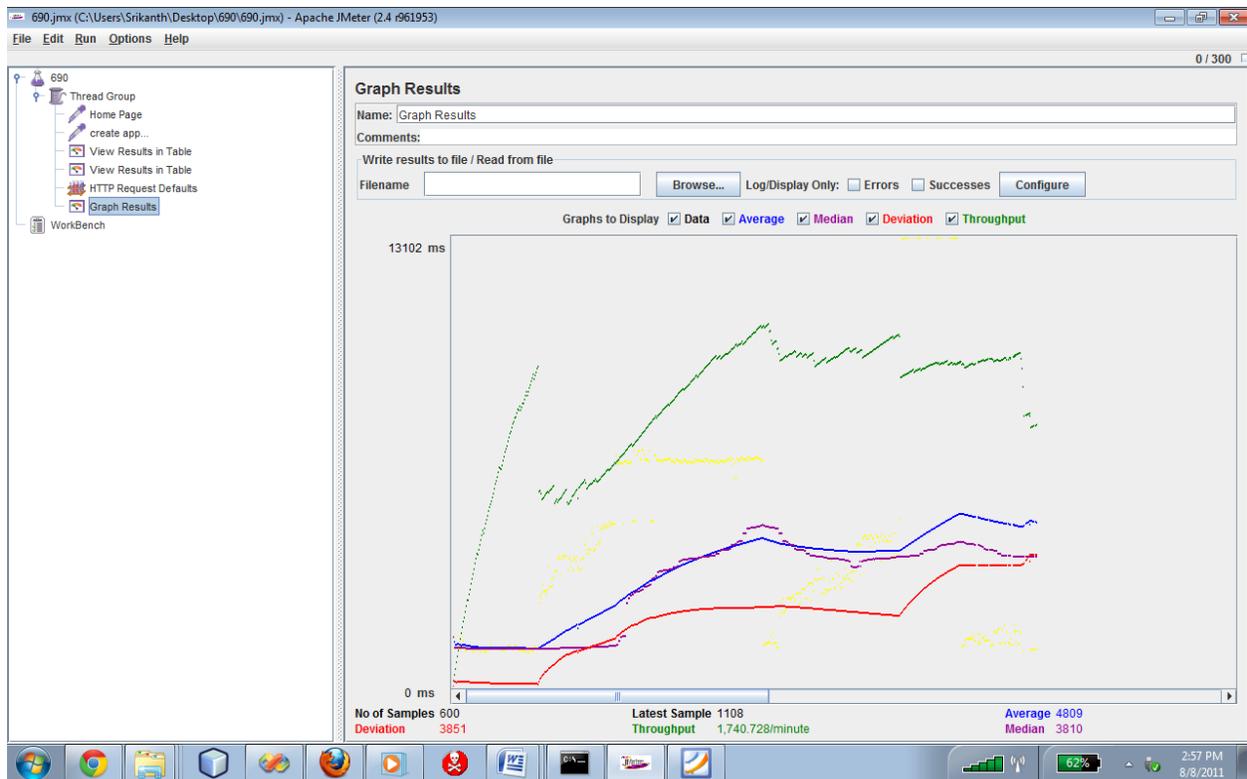


Figure 4.4 JMeter results for search students page

Database Performance:

Database duplication has been done and the performance of the pages have been tested. Records have been inserted into the database using php code which creates a file with insert statements and values which is then used to insert values into the database. 1000 Records have been entered for testing.

The student update page and application search pages have been tested with 500 samples using JMeter for its performance. There was increase in the average response time by 1300 milli seconds. The throughput is decreased from 1643.32/minute to 1332.504/minute which notifies that 1332 users are only served at a time.

There was increase in the average response time by 1458 milli seconds. The throughput is decreased from 2263/minute to 1707/minute which notifies that 500 less users are served at a time.

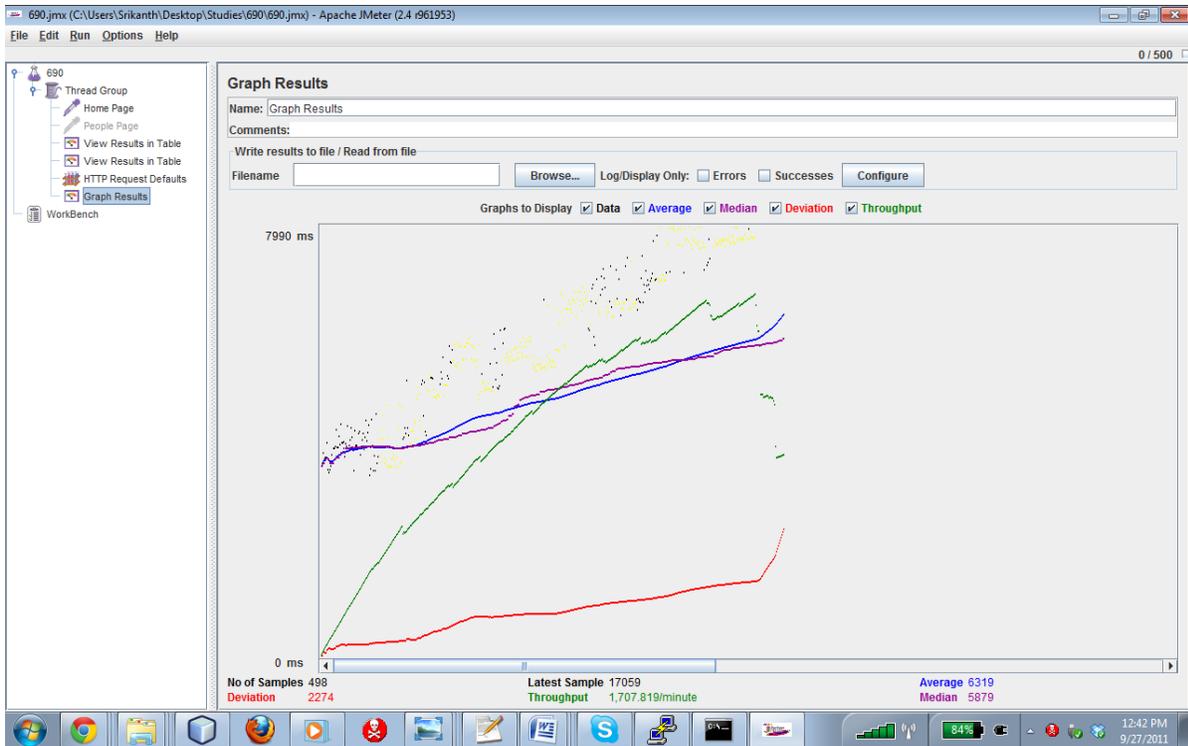


Figure 4.5 JMeter results of Applicant Search Page with large data

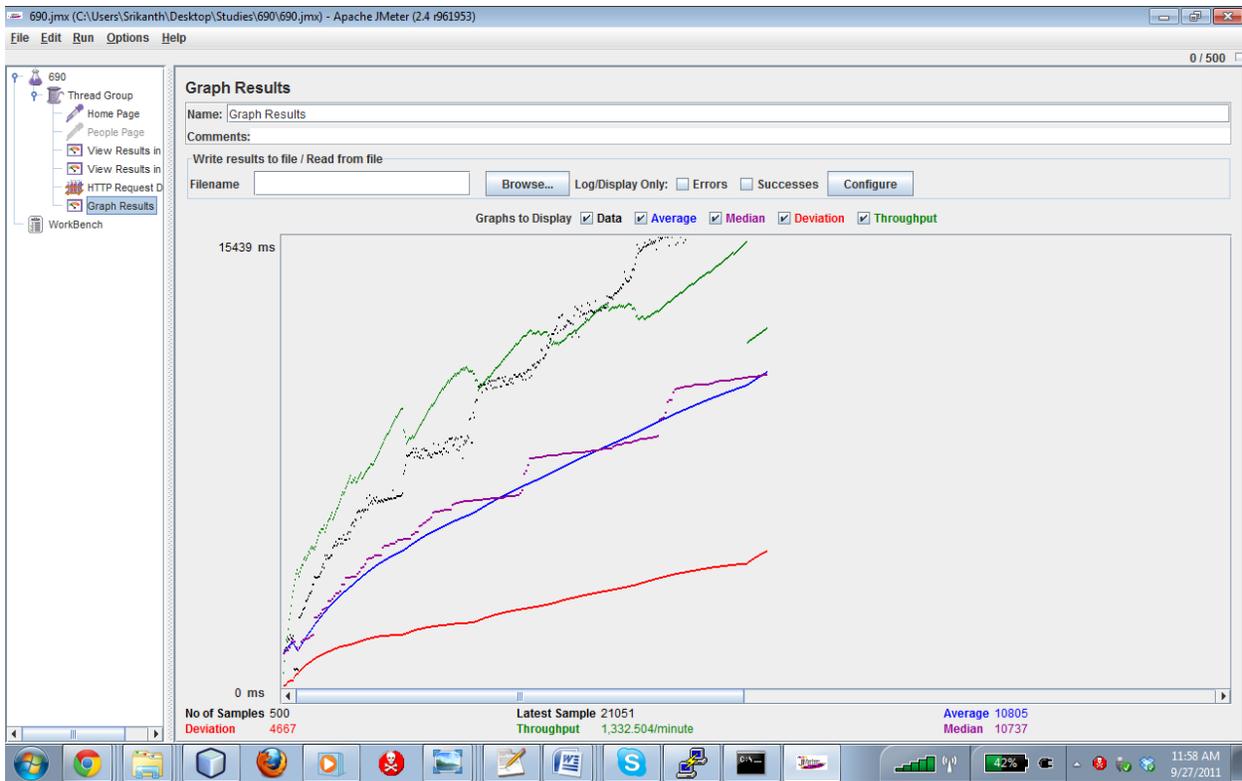


Figure 4.6 JMeter results of Applicant Update page with large data

Chapter 5 - CONCLUSIONS

The application has been designed successfully to meet all the user requirements. The Staff/Admin have been given the privilege to access; edit the information of the applicants and current students. The Applicants are given access to application to check their application status along with their information.

Future Enhancements

Current students can be given access to application so that they can check grades, gpa's and also submit their advising forms, POS forms electronically.

Data analysis can be done on grades for a particular course and generate report accordingly. Separate profiles can be created for the professors where they can keep track of the students, whom they are advising and whose committee they belong to.

CHAPTER 6: References

- [1] <http://www.tonymarston.net/php-mysql/model-view-controller.html#model>

- [2] <http://www.phpframeworks.com/php-framework-comparison/comparion.php?id=1-19>

- [3] <http://www.noupe.com/php/discussing-php-frameworks.html>

- [4] <http://www.phpframeworks.com/>

- [5] <http://ist.berkeley.edu/as-ag/pub/pdf/mvc-seminar.pdf>

- [6] <http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>

- [7] <http://allthingscs.blogspot.com/2011/03/mvc-vs-3-tier-pattern.html>