

---

# **Community College Pre-Assessment and Placement Tools: Development of the PAT**

---

**Design Document**

**Group May15-04**

# Table of Contents

- 1 Project Definition
- 2 Goals
- 3 Deliverables
- 4 System Level Design
- 5 System Requirements
- 6 Functional Decomposition
- 7 Feasibility Study
  - 7.1 Specifications
  - 7.2 Investigation of the current environment
- 8 Detailed Page Descriptions
  - 8.1 Landing
  - 8.2 Register
  - 8.3 Register Administrator
  - 8.4 SuperUser Console
  - 8.5 Student Console
  - 8.6 Admin Console
- 9 Interface specifications
  - 9.1 Models
  - 9.2 Views
  - 9.3 Controllers
- 10 Server Specifications
- 11 Client Specifications
- 12 Software specifications
  - 12.1 Student capabilities:
  - 12.2 Instructor capabilities:
  - 12.3 School Administrator capabilities:
- 13 Simulations and modeling
- 14 Implementation Issues and Challenges
- 15 Testing, procedures and specifications
  - 15.1 Test Coverage
  - 15.2 Test Methods
  - 15.3 Testing Tools
  - 15.4 Test Responsibilities
- 16 Use Case Diagrams, Use Cases, Module Diagrams
- 17 Application Screenshots
- 18 Conclusion



# 1 Project Definition

---

The goal of the PAT- ELAC group is to create a web based Pre-Assessment Tool (PAT) that will allow students to determine their projected class placement in English and Mathematics by taking one or more pre-tests in each area. The Mathematics test will be adaptive by level, beginning with basic pre-algebra and proceeding through calculus. The English test will be two separate tests designed to test students' aptitudes in different areas of English knowledge.

Based on the result of the tests, the tool must output a placement level and a variety of resources focused on their areas of weakness to assist students in preparation for the final assessment.

For research purposes, and among other information the PAT will collect student information (ID number), overall pre-assessment quiz scores, and scores for individual items on the quiz, to be used to determine PAT question, topic, and test efficacy.

The primary stakeholder in this project is East Los Angeles College, (ELAC) with secondary stakeholders of Jefferson Community and Technical College, (JCTC) Iowa State University, (ISU) Des Moines Area Community College, (DMACC) and the Iowa Board of Regents.

The primary participants in the PAT-ELAC project are we, the 5 undergraduate software developers; Ruben Arenas, the point of contact for ELAC; and our advisor, Diane Rover.

## 2 Goals

---

Breaking down the definition of the PAT-ELAC project, we propose four high level criteria surrounding the construction of a web-based tool, the PAT, that will:

- Predict a community college applicant's placement in both English and Mathematics regardless of the required placement exam (ACCUPLACER, COMPASS, or ALEKS)
- Be easily calibrated to college-specific cut-scores aligned with course and remediation levels
- Recommend instructional videos and other materials specifically aligned with the applicant's deficit areas (in both English and Mathematics)
- Be accessible and repeatable by students without cost
- Collect data that can be used by ELAC to aid in knowledge of their students, and in future development of product updates.

The goals constitute a minimum viable product that will fulfill the needs of a client. Additional features will likely be added as time allows. Each criterion may be further broken down in to a number of subgoals surrounding particular challenges in algorithm design,

data modeling, user experience, and other technical issues. However, we feel that these four goals provide an adequate level of abstraction without losing any definition in the overall project

### 3 Deliverables

---

The overall products of our development will consist of four primary deliverables and the required sub-criteria thereof:

- Fully Functional Website
  - Well Documented Code
  - Built in a Model/View/Controller framework
  - Consistent, Appealing, and Simple User Experience
  - Scalable
  - As per-user customizable as reasonably possible
  - Presents vital statistics in a usable analyzable form
  - Logging
  - Clear and understandable Data Models
  - Entirely (or nearly) bug free
- Project Report
  - Reflection
  - Guide to PAT use and, particularly, administration of the PAT
  - Highlights of key features
  - Avenues for expansion
  - Documentation of the design and creation process
- Project Presentation
  - Walkthrough
  - Key Challenges
  - Important outcomes
  - Addresses all of our identified goals

Additional deliverables may arise in the development of the overall product, but we consider these to be the three most important items for us to produce.

### 4 System Level Design

---

This section outlines a high level understanding of the Pre-Assessment Tool's basic design. Given that we are designing a web server, this primarily consists of communication between a Client and a Server. Because this Project implements a Model, View, Controller framework, this relationship can be further abstracted:

Server - Code interacting with code

- Models/Database

- Provides abstractions for use in object processing in the Controller execution and View pre-processing
- Provides queryable data storage, typically for processing in Controllers. It stores, retrieves, and updates.
- Models/Database Tables:
  - Tests
  - Topics (Question groups surrounding a single subject)
  - Questions (Individual Multiple Choice questions)
  - Answers
  - Resources (To provide to users looking to improve in a given area)
  - Users
  - Results (Test scores and related metadata)
- Controllers
  - Controllers do pre and post processing of web pages as well as database connections. This is where the vast majority of the processing on the server takes place
  - Verifies, validates, and processes form data
  - Performs Create, Report, Update and Delete operations on the above listed Models/Databases
  - On a high level, they are used in this project primarily in:
    - Verify users identity and privilege level
    - Create, store and take tests
    - Report student placements and suggest areas for additional study

Client - Humans interacting with code

- Views
  - Views create populate the web pages seen by users, and send data back to the server upon interaction
  - Responsible for all interactions with users, including
    - Displaying user options
    - Displaying interface for tests
    - Displaying placements
    - Create, Report, Update and Delete Interfaces for nearly every corresponding model
  - Some preprocessing also takes place here (for example, basic form validation with javascript)

## 5 System Requirements

---

Given the relative simplicity of the system level design, this project has only two required systems, a server and a client. The client has, of course, extremely basic requirements necessary to access the internet:

Personal Computer or Mobile Device with a Web Browser

- Chrome, Internet Explorer, Firefox, Opera or Safari (Including Mobile Versions)
- Internet Connection

It is worth noting, however that the server, conversely, is more particular in its requirements. It would necessarily be scaled up with greater use of the project, however the listed server is based only on current needs. These requirements, are primarily based on the current volume of requests the chosen development environment and supporting programs:

#### Windows Server

- Windows Server 2008 or newer
- 512 MB RAM
- SQL database either on server or in a nearby data storage unit (Depending on size)
- C# with Razor MVC framework, ASP .NET

## 6 Functional Decomposition

---

Figure 1 displays the functional decomposition of our system, which represents high level layers of our system.

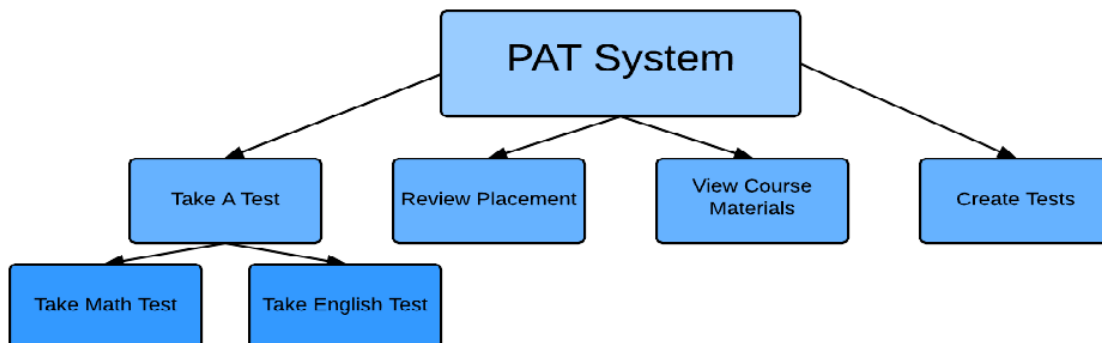


Figure 1

## 7 Feasibility Study

---

Much of the feasibility study for this project has already already been conducted in the process of the project proposal written by Arenas, et al. that can be accessed via this link: <https://drive.google.com/a/iastate.edu/file/d/0BwSkbYvbQthkKbJRRVdjVlB2d0dZVXlKcjN1b3FubUNJdnlV/view?usp=sharing>

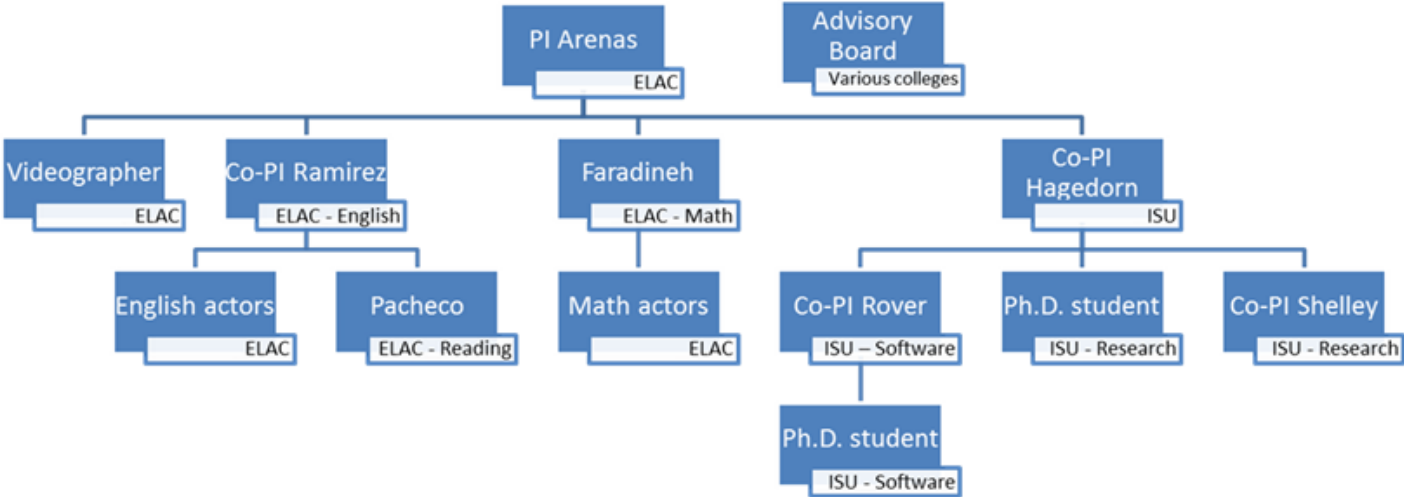
It can be found in the second half of the paper, which was not attached due to its length. However, included below are a number of notable items from the above:

<b>Year 1</b>				
<b>Area</b>	<b>Summer 2015</b>	<b>Fall 2015</b>	<b>Winter 2016</b>	<b>Spring 2016</b>
<b>Human subjects</b>	Obtain final IRB approval at all sites			
<b>Mathematics and English/Reading PAT questions</b>	Generate list of topics covered by assessment test packages that are universal		Develop PAT questions for topics	Iterative Process of testing the questions
		Develop practice problems tied to topics		
<b>Promotion and Marketing</b>	Design promotions and marketing strategies		Print promotion and marketing materials	Train staff to use materials
<b>Mathematics and English/Reading videos</b>	Choose actors from existing student tutors.	Train actors.	Pre-production/ Production/ Post-production (repeat for each topic)	
	Contract videographer	Inventory existing videography equipment. Purchase as necessary.		
<b>Site Visits</b>		Visit DMACC	Visit JCTC	
<b>Software development</b>	Assessment of prototype PAT strengths/weakness.	Needs assessment for PAT software at 3 test sites	Present software design to PI and co-PI's	



		Initial PAT design	Refine design	Programming of the PAT as a web-based system
<b>Research and Evaluation</b>	Design meetings	Map of the PAT	Design of the data collection system	Create databases (College level data)
<b>Advisory board</b>	Finalize advisory board membership	Initial advisory board meeting		Advisory board meeting
<b>Dissemination</b>		Present prototype PAT at student success conference		Presentation at a state and national conference

**Item 1.** A timeline of a proposed development and ramp-up timeline (There are 3 additional tables of this nature) (Page 17)



**Item 2.** The structure of the proposed management team for the PAT, including participants from ELAC and Iowa State University

**Research questions**

The specific research questions are:

1. Is the PAT a reliable instrument for predicting placement by the commercial tests (ACCUPLACER, COMPASS, and ALEKS). Is the prediction stable across all levels in both English and Mathematics? In other words, does the predicted placement match the actual placement? This question involves accuracy rates of the PAT.
2. Provided with a free resource, what proportion of students will use the PAT? Among those who do, what proportion will view the additional resources (the tool can record this). What proportion of PAT users will choose to re-use the PAT? Thus the study will include data on repetition and frequency of attempts.
3. For those who advance in level (after viewing the resources and retaking the placement test) what is the reliability coefficient of the instrument? In other words, were the videos and other aids responsible for an advancement in level?
4. For those who advance, are they successful in the math /English courses in which they were placed?
5. What is the student satisfaction level and ease of use associated with the PAT?
6. Do any of the above differences vary with demographics (gender, age and racial data)?

**Item 3.** Research Questions addressed in both the feasibility study and to be addressed in (planned) future research (Page 13)

## **7.1 Specifications**

The most basic concept of the PAT is very straight forward, create a system that allows students to take tests via a web interface. As such, technical studies led to a fairly simple solution. In order to fully fit the required criteria, Microsoft's .NET framework was chosen which allows use of C# mixed with standard web encoding to develop web based applications. Having our app installed on a Windows server will allow for the database to be stored on the same server. This approach allows us optimal speed, flexibility, portability, and ease of use for both users and developer

Pursuant to this goal, we have already gained access to a Windows server, and our computers provide a local area for debugging. we plan to update the server with the periodic versions of our development.

## **7.2 Investigation of the current environment**

We were provided with a simple prototype that was developed in PHP by Ruben Arenas. It implemented a number of the requested features, albeit not fully. From this, as well as the provided document, we were able to design the flow for our application, as can be seen in the flowchart below. Additionally, there are a number of testing models already in existence that we can consult in the construction of our project, such as ALEKS or Blackboard's internal testing system, which we hope to use for guidance and flow.

## 8 Detailed Page Descriptions

---

### 8.1 Landing

The web page users will be taken to when they click on the link given to them in an email about the pre-assessment test or arrive at from clicking a link on their institution's website. There will be a brief description of the PAT (possibly a video). Users can either login as an existing user or register a new account.

### 8.2 Register

New users can create an account on the Register page by providing a name, email address, and password. The new user must go through an email verification process before their account is active.

### 8.3 Register Administrator

Super admin's have the ability to register new admin's and new super admin's. This page requests the user being registered email, name, phone number and type of admin. It will then send the new user an email containing their password.

### 8.4 SuperUser Console

The SuperUser Console page is the home page for the administrator for the entire system. Super users will be redirected to this page when they login. They will have the ability to view, create, edit, or delete school administrators.

### 8.5 Student Console

The Student Console is the home page that students will be redirected to when they log in. Students will have the ability to view previous test scores, interact with resources based off of incorrect question responses, or take a Math or English practice placement exam.

### 8.6 Admin Console

The Admin Console is the home page that school administrators will be redirected to when they log in. Administrators will have the ability to view, create, edit, and delete questions, topics, and tests. When administrators create questions, they indicate what topic(s) that question falls into. Administrators can create an exam by picking out specific topics to include on the exam and how many questions to include for each topic. Administrators also have the ability to view student scores and set placement level benchmarks.

## 9 Interface specifications

---

### 9.1 Models

Table 1 describes the models that will be used to update states within the system.

<b>Name</b>	<b>Description</b>
Student	Defines a student using our system
Admin	Defines an administrator using our systems
Test	Represents a test/set of questions
Question	Represents a test used for quizzes
Topics	Represents a topic related to questions
Resource	Represents the set of resources available to users.
Topic	Represents a topic used for tests

**Table 1**

### 9.2 Views

Table 2 describes the areas where users will be able to see visual output.

<b>Name</b>	<b>Description</b>
Login	Defines the login page
Register	Defines the page that allows registration
EditUser	Defines the page that allows users to change their information
StudentAccount	Defines the student account page
AdminAccount	Defines the admin account page
Test	Defines the page for tests
About	Defines the page describing the project

**Table 2**

### 9.3 Controllers

Table 3 describes the controllers of the system that will send commands in the system.

Name	Description
Home	Controls flow between accounts and test
Account	Controls login and registration of a user
Test	Controls flow of test data
Question	Controls flow of question data

**Table 3**

## 10 Server Specifications

Our system is being developed with Microsoft's .NET framework, meaning it will be hosted on a Windows server. This server will also host a SQL database which will consist of tables that mimic our models. It will need to run Windows Server 2008 or newer.

Table 4 below represents the actions that the system will take.

Action	Interface	Methods Used
Store questions	Question model	createQuestion()
Store topics	Topic model	createTopic()
Store tests	Test model	createTest()
Send data to the client	All controllers	ReceiveData()
Receive data from client	All controllers and Models	On Page Submit

**Table 4**

## 11 Client Specifications

Table 5 below represents the specifications and actions/functionality that the client has requested the system to have.

Action	Interface	Methods Used
Register account	Registration web page (Click on link)	Register

Take Test	Student Profile	TakeTest
Review Placement	Placement Page (page forwarded to after test completion)	DisplayPlacement
View Course Materials	Student profile	ViewProfile
Create Test	Admin Page (login req.)	CreateTest

**Table 5**

## 12 Software specifications

Since our final product is a web application, we have no need for hardware specifications. Our product should function properly on any major web browser (Chrome, Firefox, Internet Explorer, Safari). For our project, we are using Microsoft’s ASP.NET MVC framework with Razor syntax. The ASP.NET MVC framework is an alternative ASP.NET Web Forms for creating web applications. The Razor syntax allows us to smoothly integrate C# and HTML into our project. We are using a Visual Studio as our integrated development environment .

Looking at the Pre-Assessment Tool itself, it will have three main categories of users: students, instructors, and school administrators. The capabilities of each type of user are described below.

### 12.1 Student capabilities:

- Register an account by navigating to the registration page and providing a username, email address, and password. After validating their email address, their account will be activated.
- Log in to their account by navigating to the login page and providing their username and password. After authenticating the student, the system will redirect the student to the student homepage.
- Take a math exam by selecting the appropriate button on the student home page. In the math exam, students will answer a series of ten questions at a time from a test created by an Instructor. If a student answers enough questions correctly (exact number determined by the school admin), they will move on to more difficult questions. If a student misses too many questions, the test will end and they will be given their math placement score.
- Take an English exam by selecting the appropriate button on the student home page. In the English exam, students will take a two-part exam consisting of reading comprehension and sentence completion. After completing both sections, students will receive their English placement score.

- View their math and English placement scores by navigating to the student home page.
- View resources, such as YouTube videos and worksheets, that specifically cover the topics that they lost points on during the exam. The resources can be found by navigating to the student home page and selecting “View Resources”.

## **12.2 Instructor capabilities:**

- View exams, topics, and questions that they have already created. They are displayed on the instructor home page.
- Create an exam by clicking on the appropriate button on the instructor home page. They will be prompted to select which topics to include on the exam and give the number of questions over each topic. Then the instructors can enter the different courses that students can test into and indicate how many correct responses are required to test into each course.
- Edit an exam by selecting an existing exam on the instructor home page and clicking the edit button. Instructors can select what part of the exam they wish to edit, such as the number of questions, topics covered, or adjust the placement courses.
- Delete an exam by selecting an existing exam on the instructor home page and clicking the delete button. Instructors will be prompted to confirm the deletion before the exam is deleted. Topics and questions are unaffected by deleting exams.
- Create a new question by clicking on the appropriate button on the instructor home page. Instructors will be prompted for the question, the correct answer, and 1-4 incorrect answers. They will also have the option of labeling the question under an existing topic.
- Edit a question by selecting an existing question on the instructor home page and clicking the edit button. They can edit the question, correct answer, or incorrect answers.
- Delete a question by selecting an existing question on the instructor home page and clicking the delete button. Instructors will be prompted to confirm the deletion before the question is deleted.
- Create a topic by clicking the appropriate button on the instructor home page. Instructors will be prompted to name the topic and then a list of all questions will be displayed. They can select which questions should be added into the newly created topic. They can also upload worksheets or videos to be used as resources by students who struggle with the topic.
- Edit a topic by selecting a topic on the instructor home page and clicking the edit button. Instructors can edit the name of a topic or add/remove

questions from the topic. They can also add/remove resources associated with the topic.

- Delete a topic by selecting a topic on the instructor home page and clicking the delete button. Instructors will be prompted to confirm the deletion before the topic is deleted.
- View student reports by clicking on the appropriate button on the instructor home page. Instructors view students math and English placement scores as well as more detailed information about their exams. Students can be sorted by name, score, or ID number.

### **12.3 School Administrator capabilities:**

- Add an instructor to the system by navigating to the admin home page and clicking the appropriate button. School administrators will supply an email address and name of the instructor and the instructor will be emailed with instructions on setting up their account.
- Remove an instructor from the system by selecting the instructor on the admin home page and clicking the delete button. School administrators will be prompted to confirm the deletion before the instructor is deleted.
- View aggregate data on students' results by navigating to the admin home page and clicking the appropriate button.
- Use all of the functionality added for instructors.

## **13 Simulations and modeling**

---

The client designed a prototype website. His site was designed to show how a student would take a placement test and view scores. The original prototype can be found at <http://www.rubenarenas.net/ELAC/>

## **14 Implementation Issues and Challenges**

---

Challenge: Maintaining a constant flow throughout the entire website.

Challenge: Guaranteeing that all of our designs will match up with the client's specifications.

Challenge: Developing an accurate class placement algorithm.

Challenge: Understanding the structure of tests



## 15 Testing, procedures and specifications

---

In the course of the prototype's development, several tests will be conducted to determine and enhance the viability of the model, with each test changing in sophistication and purpose as the project nears completion.

### 15.1 Test Coverage

Testing will begin as rudimentary checks to determine if each program part is compiling and committing in a timely manner, with later checks for bugs that can be removed and for successful communication between parts of the program.

As the project nears completion, further testing will take place using various case examples to see how the prototype reacts in each case. Ultimately, a full test of the prototype will be conducted to determine if the prototype is ready for delivery to the client.

An added feature to this process, in the case of the current project, will be the entry of the client in the later stages of testing, both for quality assurance and as an opportunity to test cases that may not have come up in earlier development.

### 15.2 Test Methods

There are four test methods that will be used for this project. The vast majority of the tests that will be used will be usability tests and regression tests. The usability tests will be used to test components and features and determine if they fit the benchmarks set by the client at the time of handover.

To that end, regression tests will be conducted beforehand during the development process to ensure proper function and coverage, and each developer has a hand in ensuring the success of the parts before further tests.

These tests will consist of a specific set of tests for roles within each part, such as build acceptance (which consists of both the successful compiling of the part as well as successful subversion), a related smoke test (determining if further testing is possible with the current level that the part is at), bug regression (in the simplest sense, finding errors and troubleshooting them), and critical path testing (assuring

that each part of the project is flowing to where it is supposed to end up and focusing on features that will be part of everyday use).

After it is reasonably certain that all major testing problems have been solved, unit testing will take place, allowing each developer to determine if proper functionality and flow has been accomplished in the course of development.

Closer the end of the project, before complete handover to the client, there will be either one or a series of final release tests to determine that the prototype is ready to be used in the field. At this point, any minor and non-threatening bugs will be evaluated and final troubleshooting will take place.

### **15.3 Testing Tools**

In the course of testing the prototype, the team has found that common programming and subversion platforms work best for the needs of development, testing and troubleshooting.

For regression testing, Visual Studio is being used for most evaluation needs, such as bugs and other errors, as well as to measure successful compilations and program execution. As well, TortoiseSVN is being used not just for subversion, but to measure the health of each new submission, determining if a new addition can be brought in without damage to the rest of the project.

For further testing involving end users, the project is being released to the client by way of movement to a Windows server, so that the client can conduct their own tests on their own equipment for how much the project matches their specifications.

When the project reaches the mobile development phase, the regression and usability cycle will restart for testing the new components. Because of its application for Android and other XML development, Eclipse is the current favorite for creation and testing at this late phase.

### **15.4 Test Responsibilities**

The project team will be conducting the rudimentary regression tests (build acceptance, bug regression, critical paths, etc.) during the early phase of the project, as these are relatively simple to check and easy to fix. The client will be introduced

when the time comes for unit testing and the final release test. As well, other end users may be introduced for later testing stages if the need arises. These end users would be other students, professors, administrators or programmers, and could be used to evaluate their respective parts or niches within those parts.

Data to be collected from the tests that is not part of the regular course of building the program will include test questions and answers, student scoring and course information, information and acceptance levels for the courses themselves, and (potentially) resources that can be used to preview courses and as reference material for the subject matter in the tests.

# 16 Use Case Diagrams, Use Cases, Module Diagrams

Figure 2 represents the use case diagram for our system.

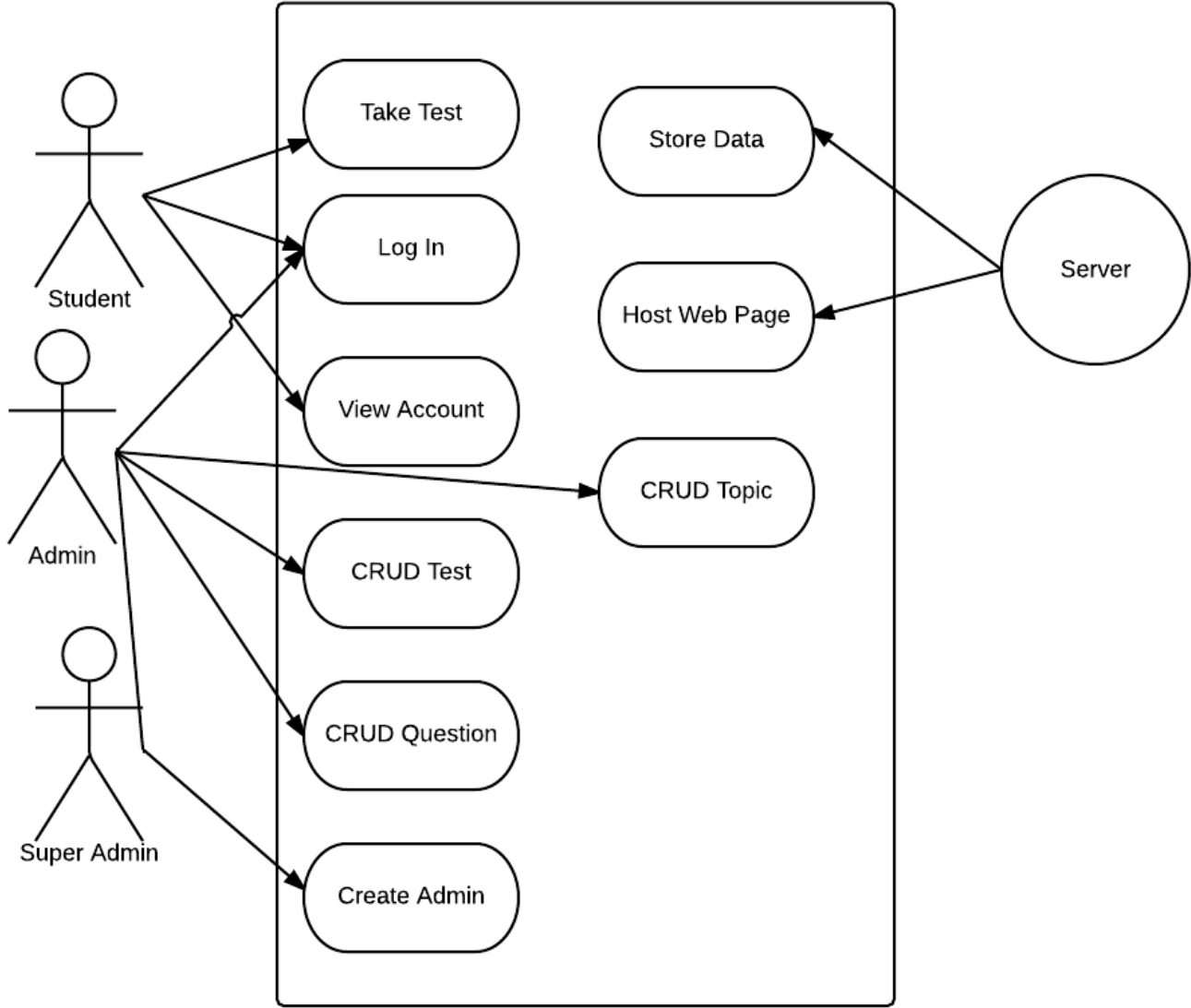


Figure 2

## 17 Application Screenshots

---

This section provides screenshots from the systems initial prototype.

### Login UI Screen

your logo here

Student Admin Test Home About Contact

Register Log in

**Log in.**

**Use a local account to log in.**

User name  
colinduffy

Password  
\*\*\*\*\*

Remember me?

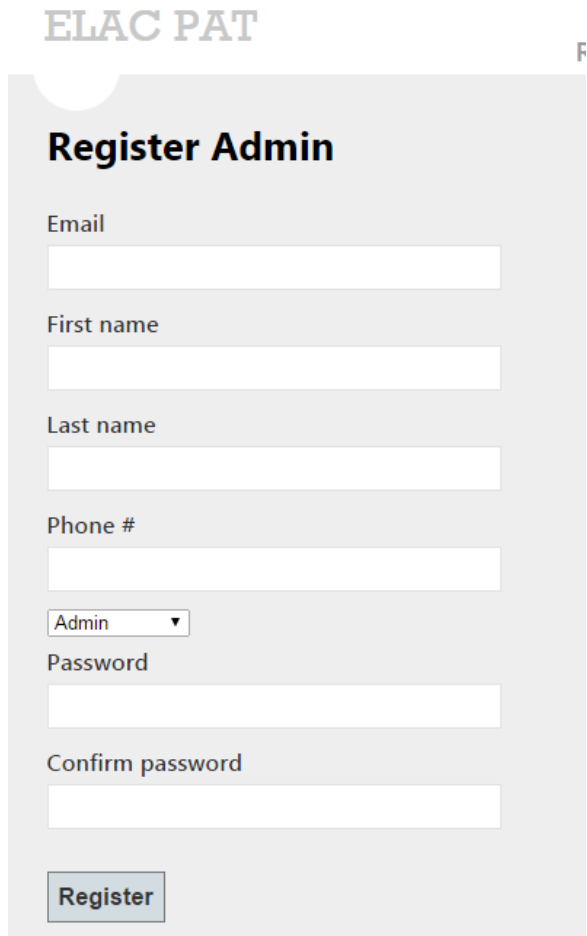
Log in

[Register](#) if you don't have an account.

© 2014 - My ASP.NET MVC Application

**Figure 3** shows the first screen a user will be brought to, which allows them to login

## Register Admin UI Screen



ELAC PAT

### Register Admin

Email

First name

Last name

Phone #

Admin ▾

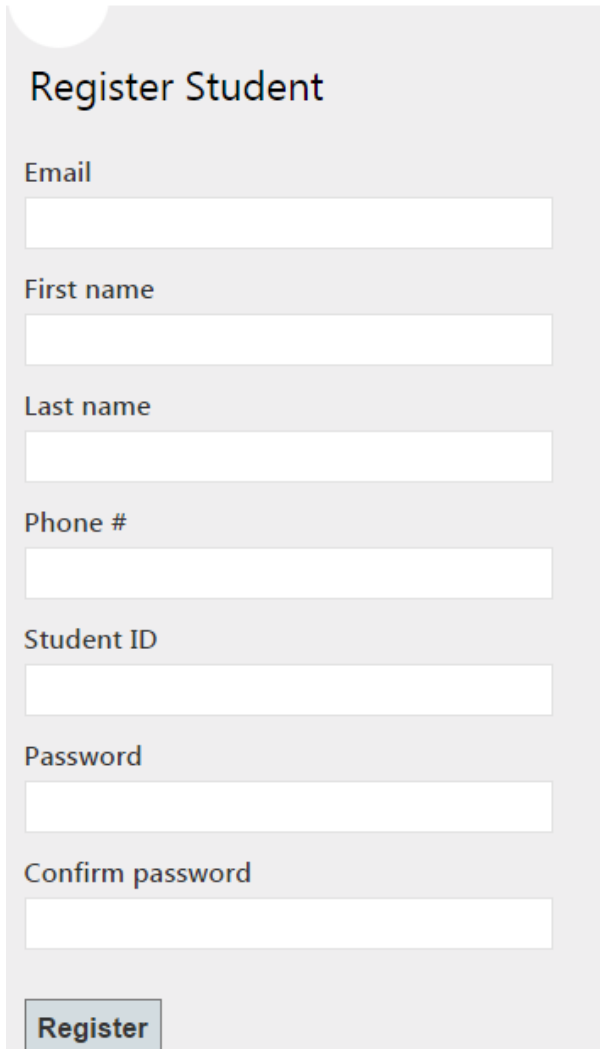
Password

Confirm password

**Figure 4** admin registration screen

## Register UI Screen

ELAC PAT



### Register Student

Email

First name

Last name

Phone #

Student ID

Password

Confirm password

**Figure 5** student registration screen

## Student Account UI Screens

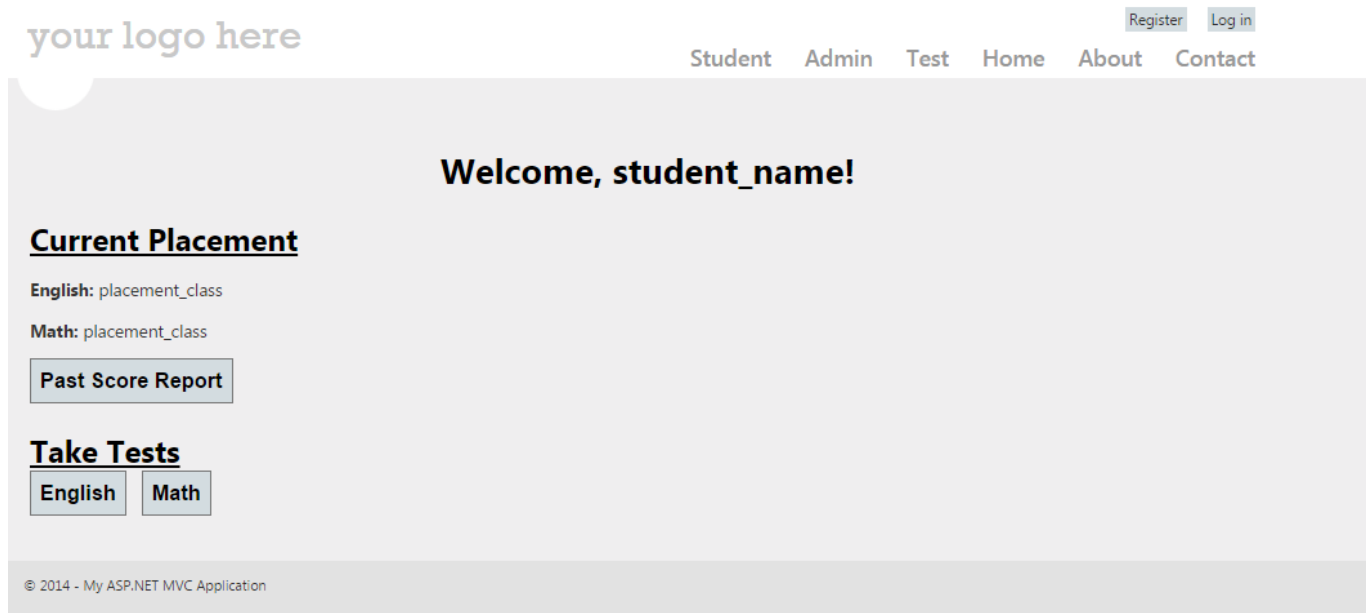


Figure 6 Student account screen

## Admin Account UI Screen

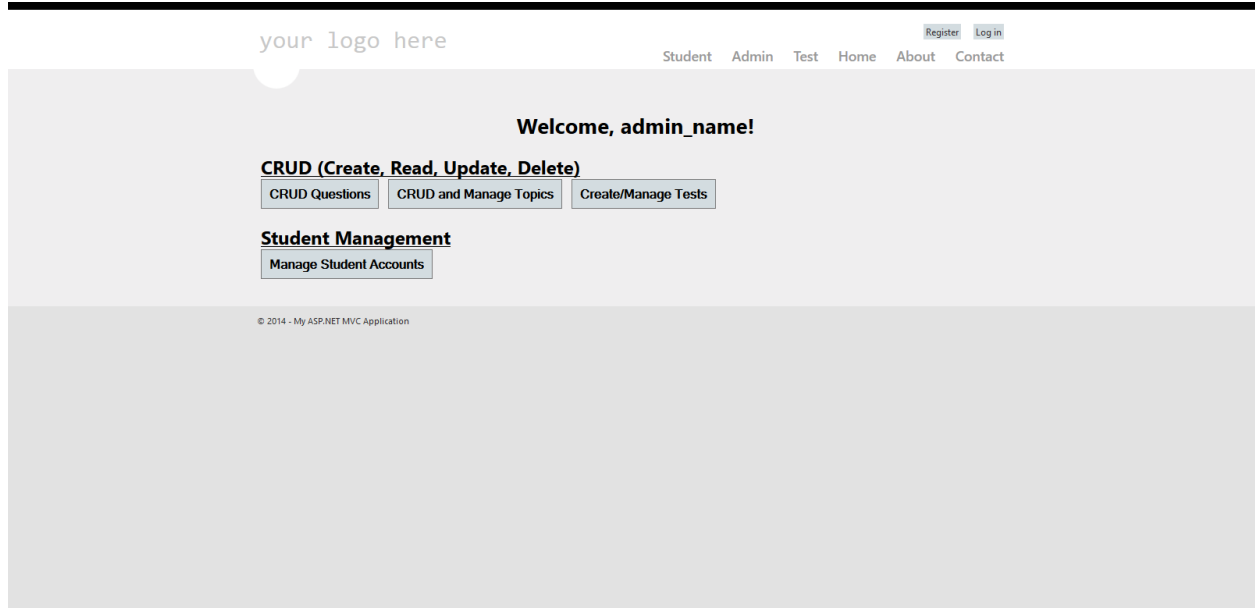


Figure 7 Admin account screen

## Quiz UI Screen

### Pre-Assessment Test

East Los Angeles College | DMACC

[Student](#) | [Admin](#) | [Test](#) | [Home](#) | [About](#) | [Contact](#)

Student: John Doe

First English Exam - MUST FINISH

This test is 10 questions long, and will take 30 minutes to complete.  
Score for this test will be measured with the Second English Test to determine English class placement

1. The proper word to use in the sentence "I \_\_\_ a good student" is:

- is
- am
- are
- be gonna

2. I before E, except after:

- C
- D
- X
- C, or when sounding like A, as in neighbor or weigh

Figure 8: Quiz User Interface Projected Screen



## Create Question UI Screen

### Pre-Assessment Test

East Los Angeles College | DMACC

[Student](#) | [Admin](#) | [Test](#) | [Home](#) | [About](#) | [Contact](#)

Professor: Max Power

QUESTION CREATION

Submit your question here:

Submit your answers here (make sure to add a comma after each answer to separate):

Answers

Designate class:

Figure 9: Question Creation User Interface Projected Screen

## 18 Conclusion

The PAT system has been designed in a manner to provide a scalable, maintainable, and reliable piece of software. Scalability comes from the database design which allows for any number of test topics and tests with any number of questions. Maintainability comes from the simplicity provided by the .NET Framework and specifically the MVC architecture used for implementation.

PAT will provide an intuitive interface that requires no training for students and minimal training for administrators. Students will be able to review material relating to the next test they will take and take tests which will help to understand placements in certain subject areas. Administrators will have several capabilities including creating tests and reviewing students placements.