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

Final Report

Team Members:

Josh Schmitt
Peter Benzoni
Colin Duffy
Nathan Steussy
Alex Yasa

Client:

East Los Angeles Community College

Advisor:

Diane Rover

Group May15-04

Table of Contents

- 1 [Problem Statement](#)
- 2 [Goals](#)
- 3 [Deliverables](#)
- 4 [Background](#)
 - 4.1 [Overall System Description](#)
 - 4.2 [System Requirements](#)
 - 4.2.1 [Functional Requirements](#)
 - 4.3 [Evaluations](#)
 - 4.3.1 [Risks & Feasibility Assessment](#)
 - 4.3.2 [Cost Considerations](#)
 - 4.3.3 [Market/Literature Survey](#)
 - 4.3.4 [Future Research](#)
 - 4.4.1 [ISU Development Schedule](#)
- 5 [Design Specifications](#)
 - 5.1 [Decomposition](#)
 - 5.2 [User Roles and Capabilities](#)
 - 5.2.1 [Student capabilities:](#)
 - 5.2.2 [Instructor capabilities:](#)
 - 5.2.3 [School Administrator capabilities:](#)
 - 5.3 [System Level Design](#)
 - 5.3.1 [Server - Code interacting with code](#)
 - 5.3.2 [Client - Humans interacting with code](#)
 - 5.4 [Page Descriptions](#)
 - 5.4.1 [Landing](#)
 - 5.4.2 [Register](#)
 - 5.4.3 [Register Administrator](#)
 - 5.4.4 [SuperUser Console](#)
 - 5.4.5 [Student Console](#)

- [5.4.6 Admin Console](#)
- [6 Implementation](#)
 - [6.1 Specifications](#)
 - [6.2 Investigation of the current environment](#)
 - [6.3 Process Details](#)
 - [6.3 Interface specifications](#)
 - [6.3.1 Models](#)
 - [6.3.2 Views](#)
 - [6.3.3 Controllers](#)
 - [6.4 Assessment of the Solution](#)
- [7 Testing](#)
 - [6.5 Validation and Acceptance Test](#)
 - [5.2 Test Plan](#)
 - [5.3 Test Coverage](#)
 - [5.4 Test Methods](#)
 - [5.5 Testing Tools](#)
 - [5.6 Test Responsibilities](#)
- [14 Implementation Issues and Challenges](#)
- [7 Conclusion](#)
 - [8.1 Operations Manual](#)
 - [8.2 Alternate Designs](#)
 - [8.3 Other Considerations](#)

1 Problem Statement

The goal of the PAT- ELAC group was to create a web based Pre-Assessment Tool (hereafter, the PAT) modeled after the placement tests currently in use in higher education, like the ALEKS. It 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 is 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 outputs 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 five 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 proposed four high level criteria surrounding the construction of the PAT, that can:

- 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, and additional features were added, such as support for file uploads. Because this project is multi-year in nature, the client will likely add further to our product.

Each identified 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 consists 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
 - Per-user and per school customizable
 - Presents vital statistics in a usable analyzable form
 - Clear and understandable Data Models
 - Well document and easily maintainable
 - 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
 - Implementation Details
 - Testing process and details
- Project Presentation
 - Walkthrough
 - Key Challenges
 - Important outcomes
 - Addresses all of our identified goals
- Project Poster

- Introduction
- Design Requirements
- Intended uses/users
- Design Approach
- Technical Details

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 Background

4.1 Overall System Description

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 is more particular in its requirements. It would necessarily be scaled up with greater utilization and deployment 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

How they chose to customize their development environment varied among developers, but the overall system used for development was:

- Visual Studio 2012
- Windows Server 2008 (Locally, Windows 7 & 8)

- MySQL

4.2 System Requirements

4.2.1 Functional Requirements

In addition to standard website functionality, especially registration and login functionality as well as multiplatform support, the PAT team identified and met the following set of functional requirements, as well as identifying how to test and and verify that functionality, which will be referred to further in the testing section:

Requirement	Testing
Must allow students to take math and english tests	We will use functional testing on multiple sets of Math and English tests. We will ensure that we test every possible placement.
Allow students to save incomplete tests that can be revisited some other time	We will start many tests and come back to them several days later to finish. We will then review results to ensure proper scoring.
Allow students to review their results	After taking tests as in the above criteria, we will ensure that results are shown. We will also ensure that results can be reviewed from the users profile page.
Allow students to view resources to study	Login as an admin and add resources. Run queries on the database to ensure they were added. Login as a student and ensure these resources load properly.
Give students proper placements	

Allow admins to create tests, topics, questions, and answers	Login as an admin and attempt to create tests. Run queries on the database to ensure the test was added.
Allow admins to modify and delete tests, topics, questions, and answers	Login as an admin and view list of tests. Select a test to modify. Update topics, description and/or type. Run queries on database to ensure update was made. Attempt test as a student.
Allow admins to review students results	Run queries to view certain student placements. Login as an admin, select same students and compare results.
Allow admins to provide resources for specific topics	Login as an admin. Attempt to add a resource. Query database to ensure resource was added.

Table 1: Functional Requirements and Testing

4.2.2 Non-Functional Requirements

These non-functional requirements, meant to govern PAT behavior, were identified and addressed:

- Scalable to accommodate spikes in users,
- Customizable to individual schools’ needs
- Well documented and easily maintainable
- Easily extensible
- Secure to protect personally identifiable information
- Prompt in returning students’ results and performing operations
- Usable and intuitive for students
- Simple but elegant UI design

4.3 Evaluations

Much of the evaluation for this project has already already been conducted in the project proposal written by Arenas, et al. which can be accessed via this link:

<http://goo.gl/GpLnBr>

However, the assessments below address specific areas relating to the senior design project portion of the PAT

4.3.1 Risks & Feasibility Assessment

During the first semester, the PAT team conducted feasibility and risks assessment, which by necessity, must consider five aspects of what makes the project possible and what impact those areas have, in the short and long term. These areas which are technical, economic, operational, and schedule feasibilities, are examined below:

Arguably the primary hurdle in determining the feasibility of the PAT is the technical feasibility of such a product. As students, the team lacked much of the technical expertise, technical resources, and technical tools typically present in the creation of such a project. However, these three areas of potential pitfall were mitigated. First, our lack of expertise, both in the framework and overall, is made up for by the vast technical knowledge base available online, especially for the C#/ASP.NET MVC Razor framework, one of the most used web frameworks. Any potential problems were largely able to be addressed by the large number of tutorials and resolved problems online. In regards to tools/resources, Iowa State University has provided much of both, by providing a server and hosting, as well as bandwidth, all adequate to its current needs. Moreover, its software license have been essential in getting access to proprietary software. Where Iowa State fell through, it was easily made up for by open source software and libraries. In the long term, the coalition of schools coming together on the PAT have world class technical resources and support, so it shouldn't become an issue.

Many of the economic considerations are covered in the cost analysis section, so this section will be general and brief. The primary economic benefit here is primarily to the student, who can by using the PAT prepare for tests that cost between \$25-75 per student. At universities where this bill is footed by the institution, this savings can be transferred to the institution. Moreover, much of the costs associated with the site are to be offset by a series of grants.

Operational feasibility, too, is of concern. The PAT must not only solve the problem presented, but solve it well. Namely, at its minimum, it must meet all previously identified project goals, as well as the identified functional and nonfunctional requirements.

As the solution and specifications are fundamentally centered around these assumptions and how to meet said needs, this is perhaps the area it is most suited for. Finally, the project team has put particular emphasis on reliability, usability, and maintainability and have therefore integrated them into aspect of the project.

The problem of schedule is primarily addressed in the forthcoming schedule section. The PAT team is part of a larger implementation plan and has broken its section down into smaller, manageable parts. Given the level of technical experience and total period for development, the developers were able to meet their portion of the larger PAT implementation plan

The financial feasibility of the Pre-Assessment Tool is expanded in considerable detail in the Cost Considerations section below. The gist of the conclusion, though is simple: low costs in the initial period of the site, though these will rise somewhat over time

Table 2 represents the risks that the development team faced and how the strategies used to address those concerns.

Risks	Prob	Criticality	Risk Factor	Mitigation Strategy
Not being able to effectively communicate with client due to distance	0.9	High	Low	Email makes it easy to discuss things with our client. Also, our meeting schedule is fairly flexible so we're able to be efficient with our time.
Unfamiliarity with C# and MVC.net framework	0.8	Medium	Low	Use online tutorials and resources to ensure that all members of the team are able to help develop the project effectively.
Creating adaptive testing structure	0.95	High	High	Collaborate as a team to help best design the adaptive testing algorithm.
Client decides to use different language or framework	0.05	Very High	Very Low	Develop our code in a way that we can reuse it on a different platform easily, if need be.
Client decides to change project requirements	0.25	Very High	Medium	Keep track of all designs and plans to make any changes easier to deal with.
Not having enough time to finish due to busy schedules	0.1	Very Low	Low	Plan our time effectively to help mitigate any project halts due to schedule constraints.

Lack of user base for field testing	0.95	Medium	Low	Keep our client updated with all progress, and test it with others as we go to ensure that they are satisfied with the project.
Website optimization may be difficult	0.8	Low	Medium	Continuously test and improve optimization of the project.

Table 2: Risks & Mitigation Strategies

4.3.2 Cost Considerations

Costs, while not of paramount concern in this of the project, must still be considered, especially as this project is expected to scale to the national level within the next 5 years, and could experience relatively high levels of use as it is adopted among universities across the nation, as envisaged in the 45 page proposal narrative provided by the client. Such a vision can be executed in one of two ways - single source hosting (likely Iowa State University) or virtual web hosting. (e.g. Amazon AWS) Both possibilities are addressed herein, along with initial costs (those created during development) and an evaluation of costs of ongoing support upon completion of the Senior Design Project phase of this project.

Initial design costs are extremely low. As this is a senior design project, the developers are unpaid, and due to the hosting being kept in house, the two primary costs normally associated with web development, have been effectively and entirely mitigated. What costs that remain are simply small incidentals, such as software licenses and pizza.

Looking into the slightly longer term, upon completion of the Senior Design Project, the university consortium will likely have to hire someone to maintain and update the product. This will cost >\$50000 for a single developer. It is suggested that the

university consortium fold this into one of their web development departments, such as the eminently qualified Iowa State WebDev team.

With this approach, then, it also makes sense to host in house (at Iowa State) with its privileged bandwidth and large number of servers. Even scaled, the University should be able to manage the relatively high number of requests that would come through (particularly in the late summer) This would allow them to keep annual costs in a range affordable to the entire consortium of schools, ~\$100,000 based on the cost of server maintenance and bandwidth

Though, in terms of just-in-time scalability, Amazon AWS reigns, so it remains to be seen which would be the most affordable in the long run.

4.3.3 Market/Literature Survey

Arguably the most essential part of an evaluation, this marketing and literature analysis is split into its two logical parts: Marketing, which will consider the potential demand for the services the PAT provides and includes a long term plan for adoption. It will then consider literature - that is, existing alternatives as well as research on adaptive testing and practice tests

Data collection will be a very important aspect of the PAT. A mechanism that will record student information as it's imputed into the system. As students log in, and they begin taking and finishing exams, their data is collected automatically.

Table 3 (provided by ELAC) below represents the data that the system collects as users are logged in and accessing the system.

PAT	Transcript data	Demographics (College Data)
Student ID	Student ID	Gender

PAT Scores - Mathematics and English	Actual Placement - Mathematics and English	Race
Predicted Placement	Enrollments in Mathematics and/or English	Age
# minutes observing resources	English/Mathematics grades and course withdrawal information	Pell eligibility
# times Repeat PAT	Retention and Completion	
	Semester and Cumulative GPA	

Table 3: Types of User Data to be Collected

Through this information, ELAC will be able to determine specifics as to who is using their system, and what changes may be necessary. In terms of marketing, this allows for flexibility in the system, and invaluable instantaneous feedback.

The literature survey of the market shall be conducted by ELAC. This very comprehensive, extensive study will analyze the ways that commercial test systems, such as ACCUPLACER, COMPASS, and ALEKS function in terms of accuracy placement. The PAT system will require such analysis to be done in order to ensure that the system is placing students in the correct spot. A small amount of informal research was conducted by individual members of the senior design team to help gain an understanding of other projects on the market well before ELAC has time to finish their study. This aided in both the design and the development of the product, and the way that it function in terms of accuracy and overall student placement. The main goal here

is to provide ELAC with a system that will be easily expanded upon in the following years.

4.3.4 Future Research

Also worth noting are a number of avenues for future research as identified by Arenas Et Al. 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)?

4.4 Schedule

Below, courtesy of Arena et al, is a high level view of the PAT project as a whole. The highlighted area represents the contribution of the ISU senior design team.

4.4.1 ISU Development Schedule

The following tables show, by semester, the schedule that was used for the project. This is divided first by the tasks that need to be completed during the given periods, and then into who was responsible for what part of the project (or tasks listed alone for group collaborations).

This schedule, developed in the fall, was not followed entirely strictly, but provides a good overview of the project as a whole.

Fall Semester 2014								
Area	To 9/15	To 10/1	To 10/15	To 11/1	To 11/15	To 12/1	To 12/15	To 1/1
Project Prep	Research project requirements and start using Visual Studio for code development			All further tasks are outlined in the subsequent rows, but is related to the actions done in the project prep.				
Website Framework			Josh: Build website	Josh: Maintain website and add new documents to the site as required.				
Build Database			Colin: Find and secure server needed to store database	Colin: Continue updating database tables Alex, Nathan: Integrate test taking, results and other program parts into the server				
Implement Basic Admin Functions				Alex: Create basic framework for admin page Peter: Flesh out admin page and create spinoff functions for page Colin: Create <u>superadmin</u> and user roles	Nathan: Create student info page for admins All: Debug problem areas within the prototype			
Develop Testing	Colin: Create prototype model for project.			Alex: Create test taking and results pages Peter: Create page for question creation as well as storage for answers		Peter: Edit question creation so that it writes to database Colin: Update test model for database table		
Implement Basic User Functions				Colin: Create and update user login, other user roles Josh: Create basic framework for student page			All: Debug problem areas within the prototype	
Implement MVP							(Carries over to late Jan. 2015) Move project over to dedicated servers, open it up to client	
General Project Items	Biweekly meetings with advisor, biweekly communication with client, weekly meetings to review tasks and documentation, weekly status reports							

White boxes are initializations

Gray boxes are maintenance

Table 4: Fall 2014 Schedule and Duties

Spring Semester 2015								
Area	To 1/15	To 2/1	To 2/15	To 3/1	To 3/15	To 4/1	To 4/15	To end of term
Add Extra Features		Evaluate need to add non-essential features (references to course material, videos, etc.)						
Testing/ Troubleshooting	Preliminary viability testing will be done beforehand.		Begin running prototype, fix any bugs that come up, add essential components as necessary					
Mobile Development				Adapt prototype to work on Android (will add xml programming at that time).				
Verification, Documentation, etc.	Most documentation tasks will be done throughout the semester and are listed under "General Project Items" in the Fall 2014 table.						Complete documentation, present at end of semester	

Table 5: Spring 2015 Schedule and Duties

4.4.2 Overall Schedule

Additionally, the tables on the following pages demonstrate further work to be done on the PAT after the completion of the team's installment. By necessity, the categories on these tables are different from ours, to reflect involvement from outside groups and more general project needs at the relevant times. It is important for understanding ISU's contribution in a broader implementation context and the needs identified in the non-functional requirements

Year 1				
Area	Summer 2015	Fall 2015	Winter 2016	Spring 2016
Human subjects	Obtain final IRB approval at all sites			
Mathematics and English/Reading PAT questions	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			
Promotion and Marketing	Design promotions and marketing strategies		Print promotion and marketing materials	Train staff to use materials
Mathematics and English/Reading videos	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.		
Site Visits		Visit DMACC	Visit JCTC	
Software development	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
Research and Evaluation	Design meetings	Map of the PAT	Design of the data collection system	Create databases (College level data)
Advisory board	Finalize advisory board membership	Initial advisory board meeting		Advisory board meeting
Dissemination		Present prototype PAT at student success conference		Presentation at a state and national conference

Table 6: Plan for Summer 2015 to Spring 2016

Year 2				
Area	Summer 2016	Fall 2016	Winter 2017	Spring 2017
Human subjects	Renew IRB documentation			
Mathematics and English/Reading PAT questions	Develop PAT questions for topics			Testing of questions
	Develop practice problems tied to topics			
Promotion and Marketing	Continue development of materials	Web-based promotions		
Data Collection	Collect college level data including levels of specified enrollments (Math and English)			
YouTube	Upload videos to YouTube			
Mathematics and English/Reading videos	Rehire and train actors as necessary			
	Pre-production/ Production/ Post-production (repeat for each topic)			
Software development	Review of design with PI and co-PI's	Presentation of software to test sites	Software tests	Installation of PAT at test sites
	Programming of the PAT as a web-based system		Complete programming of web-based PAT	initial field based testing
	Design of PAT app	Programming of PAT app		Complete programming of PAT app
Site Visits		Visit JCTC		Visit DMACC
Research and Evaluation	Pilot studies at ELAC begin	Descriptive analyses - gender, Pell, ethnicity, etc.		PAT goes live and begins collecting student data
Advisory board		Advisory board meeting		Advisory board meeting
Dissemination		Present on PAT - state level organizations in CA, IA, and KY		State and National conference presentations

Table 7: Plan for Summer 2016 to Spring 2017

Year 3				
Area	Summer 2017	Fall 2017	Winter 2018	Spring 2018
Human subjects	Renew human subjects approvals			
Mathematics and English/Reading PAT questions	Refine PAT items based on statistical data			
	Add to practice problem database			
		Field testing of questions		
Data Collection	Collect college level data including levels of specified enrollments (Math and English)			
Promotion and Marketing	Monitor promotion plans and modify as necessary			
Mathematics and English/Reading videos	Reshoot, refine, and edit videos as needed			
Software development	Software tests of PAT and modifications as needed			
YouTube	Testing of YouTube videos			
Research and Evaluation	Tests of Fidelity of the instrument - Analyses- and reports		Analyses - Use of materials (videos and other)	Evaluation of PAT use
Advisory board		Advisory board meeting		Advisory board meeting
Dissemination	Writing articles for publication in national journals		DREAM conference	Presentation at AERA

Table 8: Plan for Summer 2017 to Spring 2018

Year 4				
Area	Summer 2018	Fall 2018	Winter 2019	Spring 2019
Human subjects	Renew IRB approvals			
Mathematics and English/Reading PAT questions	Refine PAT items based on statistical data			
	Add to practice problem database			
Data Collection	Collect college level data including levels of specified enrollments (Math and English)			
Mathematics and English/Reading videos	Reshoot and refine videos as needed			
Promotion and Marketing	Monitor promotion plans and modify as necessary			
Site Visits		Visit JCTC		Visit DMACC
Software development	Review functioning of PAT software and modify as needed			
Research and Evaluation	Survival analysis methods & event history analysis			
Advisory board		Advisory board meeting		Advisory board meeting
Dissemination	Writing articles for publication and presentation	Presentations at state level		National conferences

Table 9: Plan for Summer 2018 to Spring 2019

5 Design Specifications

5.1 Decomposition

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

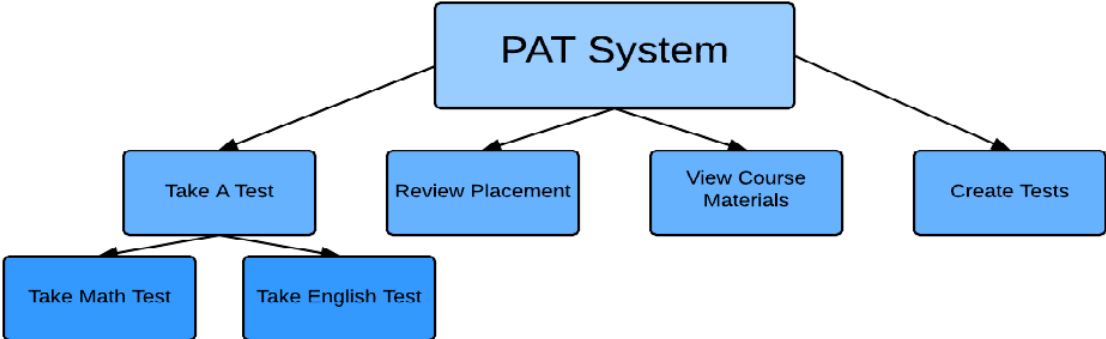


Figure 1: Functional Decomposition of the PAT

Expanding this idea to include a general flow through the program and various forthcoming specifications, we can create a flowchart outlining the minimum viable

version of the PAT.

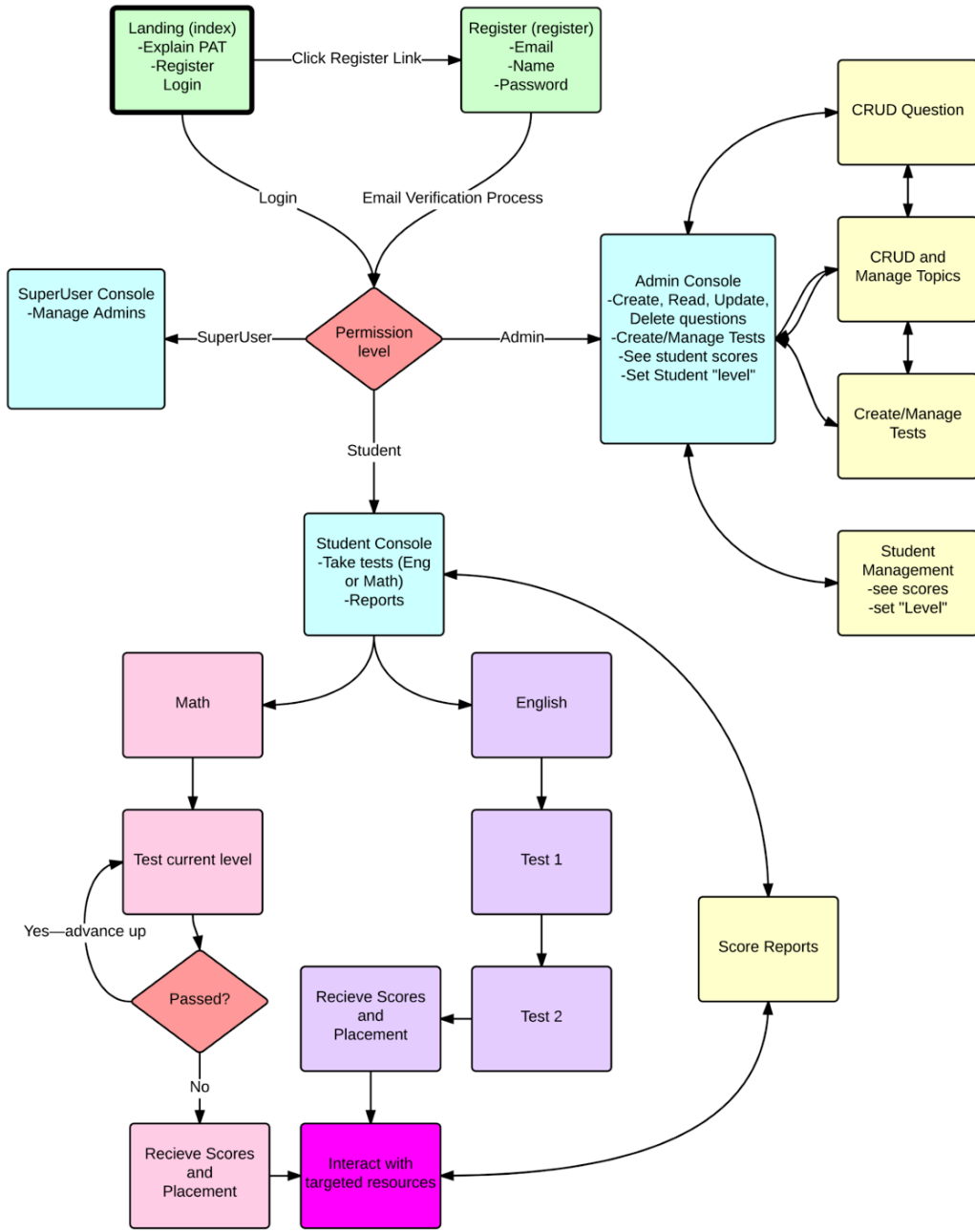


Figure 2: PAT Flowchart

5.2 User Roles and Capabilities

With the previous section in mind, Pre-Assessment Tool will have three main categories of users: students, instructors, and school administrators. The capabilities of each type of user are described below.

5.2.1 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”.

5.2.2 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.

5.2.3 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.

5.3 System Level Design

It is now possible to initiate a high level understanding of the Pre-Assessment Tool's basic design. Given that the PAT is 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:

5.3.1 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

5.3.2 Client - Humans interacting with code

- Views

- Views create and 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)

5.3.3 Block diagram

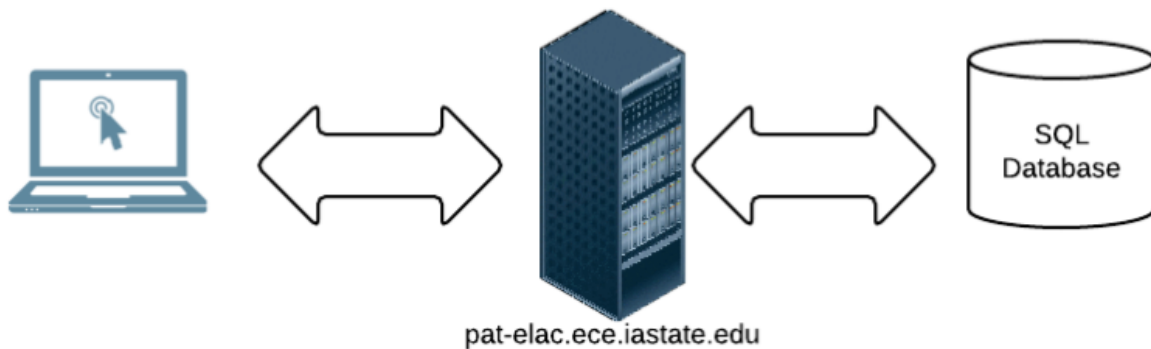


Figure 3: PAT Block Diagram

5.4 Page Descriptions

5.4.1 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.

5.4.2 Register

New users can create an account on the Register page by providing a name, email address, and password.

5.4.3 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.

5.4.4 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.

5.4.5 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.

5.4.6 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.

6 Implementation

6.1 Specifications

The 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 the application installed on a Windows server allowed for the database to be stored on the same server. This approach allowed optimal speed, flexibility, portability, and ease of use for both users and developers.

Pursuant to this goal, the PAT team gained access to a Windows server, and used personal computers as a local area for debugging. The server was updated periodically with new stable versions of the PAT.

6.2 Investigation of the current environment

The PAT team was 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, they were able to design the flow the PAT, as can be seen in the flowchart in the previous section. Additionally, there are a number of testing models already in existence that were consulted in the construction of the project, such as ALEKS or Blackboard's internal testing system, which were used for guidance and flow.

6.3 Process Details

The structure of the project is modular. The application was built using the ASP .NET MVC framework, so the code is separated into models, views, and controllers. The team has implemented basic login functionality for users, using CSHTML to generate the view

and C# for the model and controller. Using the Entity framework, user information is stored in the database. The modular design of the project allowed the team to add additional functionality by building on the existing MVC structure, as seen below:

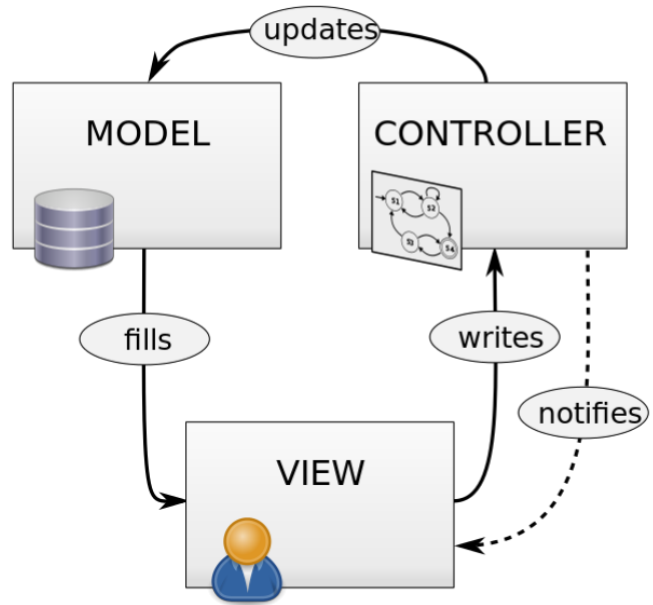


Figure 4: Model, View, Controller Framework

6.3 Interface specifications

6.3.1 Models

Table 10 describes the models that will be used to update states within the system.

Name	Description
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 10: Models

6.3.2 Views

Table 11 describes the areas where users will be able to see visual output.

Name	Description
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 11: Views

6.3.3 Controllers

Table 12 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 12

6.4 Assessment of the Solution

The solution provides a very scalable design, mostly due to the database which is designed in way to handle any type of test, any topic areas for a test, any number of questions for a test and any number of possible answers for a question. Drawbacks to this solution are the number of queries required. This will not hurt performance significantly though, as SQL is extremely fast.

Along with the database design, ASP.NET MVC framework is being used for development, which provides a modular structure to the source code. Separating the models, views and controllers in a web application allows for easy optimization. The ability to use C# and standard object oriented practices for back end code allows a large amount of flexibility. While the front end is developed using Razor, which blends html and C# to provide simple design with advanced capabilities. Two drawbacks here are that the application must run on a Windows server and the developing team has little experience with the framework.

6.5 Screenshots

6.5.1 Concept Sketch

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

6.5.2 Screenshots

Respectively:

1. Login Page
2. Student Landing Page
3. Test Taking Page
4. Resources Page
5. Admin Landing Page
6. Test Creation Page
7. Test Create, Read, Update, and Delete Page

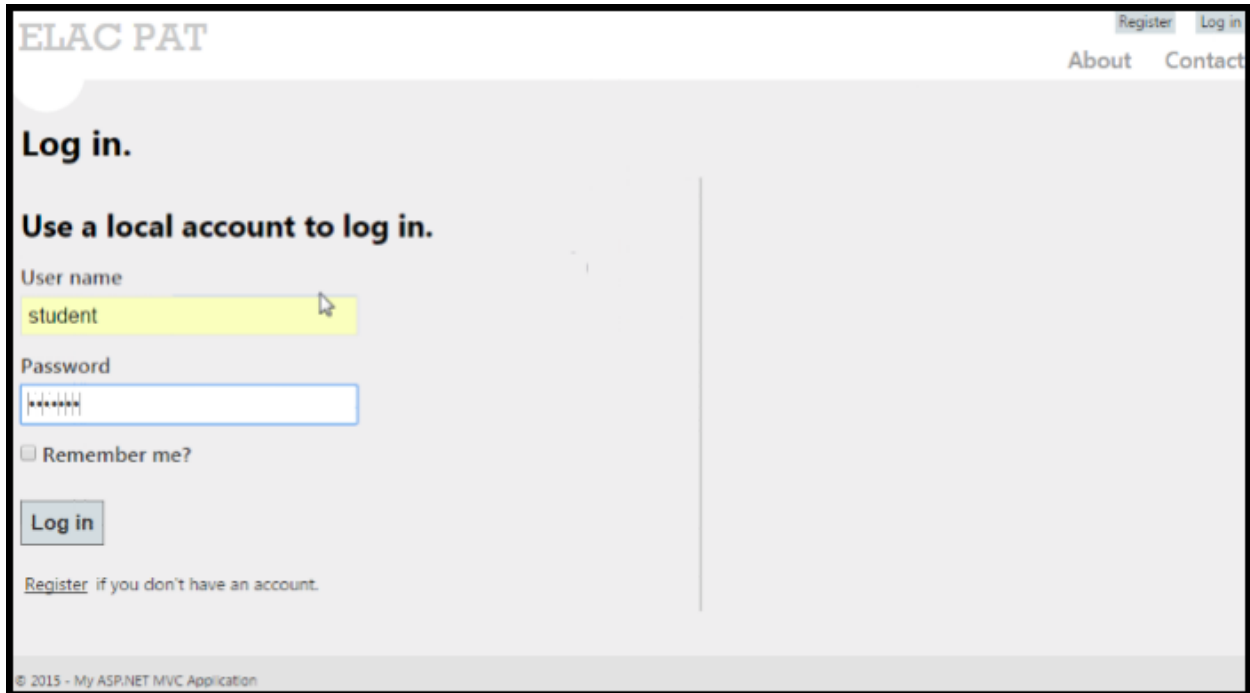


Figure 5: Login Page

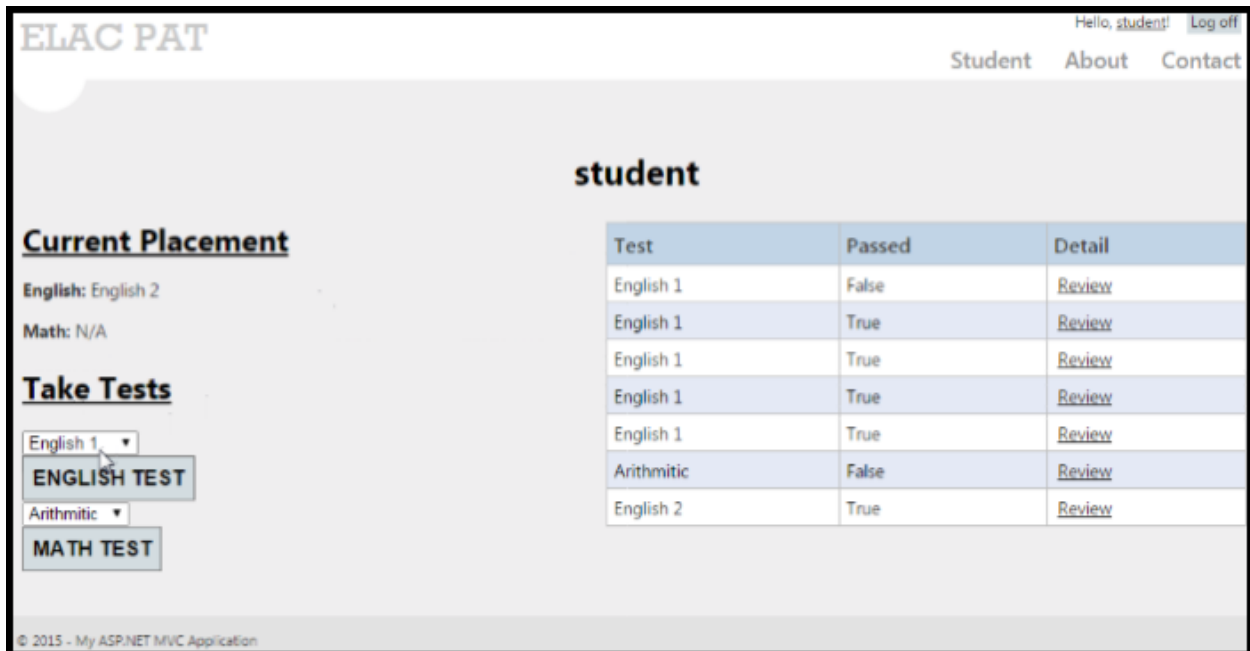


Figure 6: Student Landing Page



Figure 7: Test Taking Page

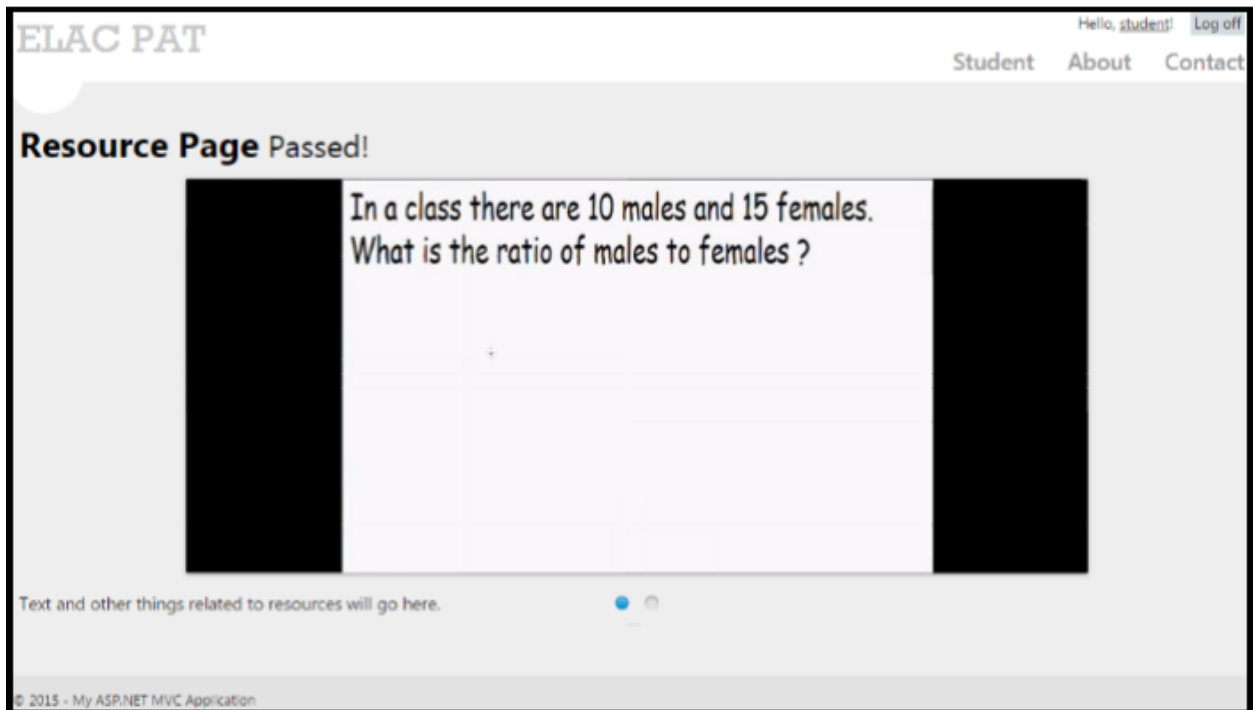


Figure 8: Resources Page

ELAC PAT Hello, s_admin! [Log off](#)

[Student](#) [Admin](#) [About](#) [Contact](#)

s_admin

Manage

[Upload File](#) [Manage Topics](#) [Manage Questions](#)
[Manage Tests](#) [Manage Resources](#) [Register Admin](#)

Student Management

[Student Info](#)

Question	Topic	Percent	Correct	Total
10:2 = ?	Ratios	100	2	2
20 white 10 asian 10 hispanic. white:everyone?	Ratios	50	1	2
Ratio 3	Ratios	50	1	2
Add 5	Addition	50	1	2
Add 2	Addition	50	1	2
Add 4	Ratios	100	1	1
Sub 1	Subtraction	100	2	2

1 2 3 4 >

© 2015 - My ASP.NET MVC Application

Figure 9: Admin Landing Page

ELAC PAT Hello, s_admin! [Log off](#)

[Student](#) [Admin](#) [About](#) [Contact](#)

Create

Level:

Threshold:

SubjectId:

Description:

Topic	Quantity	Value/Question
<input type="text" value="Ratios"/>	<input type="text" value="0"/>	<input type="text" value="1"/>
<input type="text" value="Ratios"/>	<input type="text" value="0"/>	<input type="text" value="1"/>
<input type="text" value="Ratios"/>	<input type="text" value="0"/>	<input type="text" value="1"/>
<input type="text" value="Ratios"/>	<input type="text" value="0"/>	<input type="text" value="1"/>

Figure 10: Test Creation Page

Index

[Create New](#)

Level	Threshold	SubjectId	Description	
0	70	0	Arithmitic	Edit Details Delete
0	60	2	English 1	Edit Details Delete
0	100	2	English 2	Edit Details Delete
1	4	1	Test	Edit Details Delete
1	1	1	r	Edit Details Delete
1	1	1	test	Edit Details Delete
1	1	1	dd	Edit Details Delete
1	1	2	EnglishID?	Edit Details Delete
1	2	1	Lots	Edit Details Delete
2	90	1	New Test	Edit Details Delete
1	74	1	new test	Edit Details Delete

Figure 11: Test CRUD Page

7 Testing

7.1 Test Plan

In the course of the prototype's development, several tests were conducted to determine and enhance the viability of the model. Each round of testing matured the solution in terms of sophistication and purpose. However, the majority of testing in development was done informally or built into regression tests.

7.2 Validation and Acceptance Tests

In section 4.2, functional testing practices were mentioned alongside each requirement. This practice was chosen opposed to unit testing for simplicity. Developers did all functional testing until December when the application was deployed to the server, which allowed our client to start testing as well. Once the client had more visibility, the developers had customized feedback which developers then responded to fit the concerns and requests of the client

7.3 Test Coverage

Testing began 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 neared completion, further testing took place using various case examples to see how the prototype reacts in each case. Ultimately, a full test of the prototype is scheduled to be conducted to determine prior to the prototype being delivered to the client.

The entry of the client in the later stages of testing ensured, both quality assurance and an opportunity to test cases that didn't come up in earlier development.

7.4 Test Methods

There are four test methods that 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 were conducted 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 consisted 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 later developers to determine if proper functionality and flow has been accomplished in the course of their development.

Before complete handover to the client, there will be 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, documented, and final troubleshooting will take place.

7.5 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 was 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 was released to the client by way of movement to a Windows server, so that the client could conduct their own tests on their own equipment to evaluate how well the project matches their specifications.

7.6 Test Responsibilities

The project team conducted 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 was introduced when the time comes for unit testing and the final release test. As development is passed on, other end users may be introduced for later testing stages if the need arises. These end users would likely be other students, professors, administrators or programmers, and could be used to evaluate their respective parts or niches within those parts.

Data collected from the tests that is not part of the regular course of building the program 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.

8 Challenges

8.1 Implementation challenges

The development team face a number of challenges, most notably the following:

Server/Database Interface

- Inexperienced with deploying and configuring Windows Servers
- Limited database-application interfacing experience
- Unreliable local host

Meeting Technical Requirements

- Complex technical and non-technical requirements
- Based on ~60 page document
- Difficult to meet within given specifications

Test Integration Hierarchy

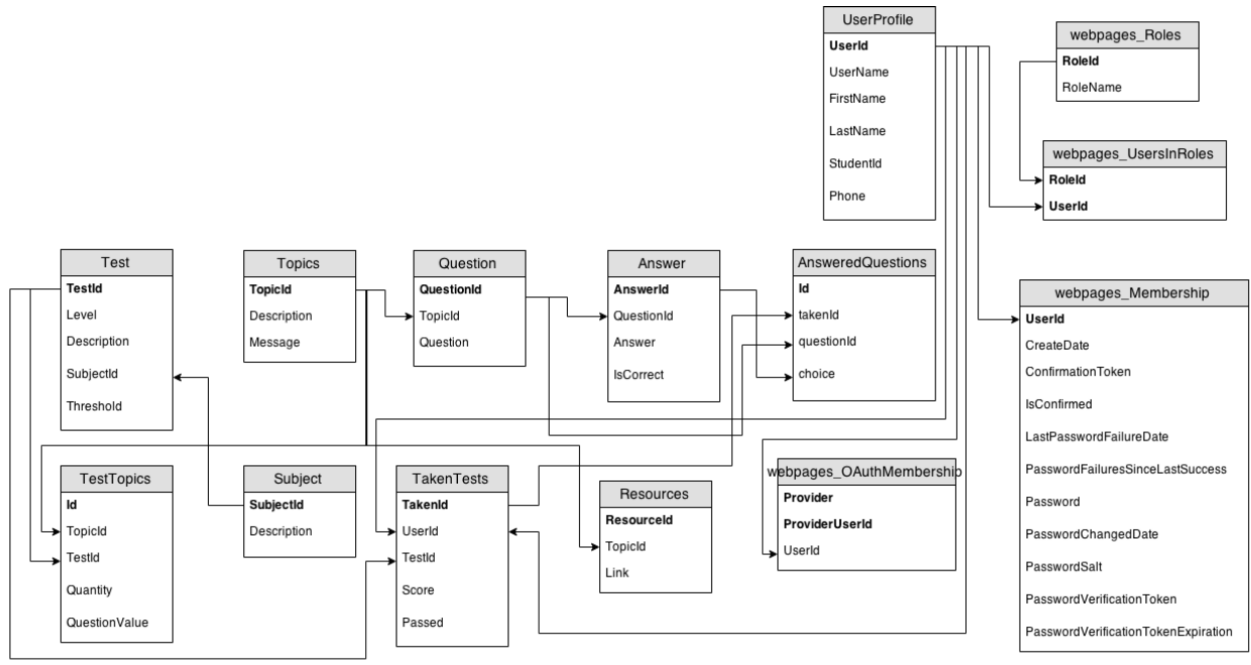
- Complex data retrieval structure
- Total model interdependency
- Adapting to New Framework

Team inexperienced with development environment

- New tools and languages
- ASP.NET Razor MVC
- C#, SQL, Visual Studio
- CSHTML

8.2 Data Retrieval Structure

In order to implement these requirements, the Pre-Assessment Tool uses a complex data retrieval structure, as shown below. This database is based on the Model portion of our Model, View, Controller framework, which is described previously.



9 Conclusion

The Pre-Assessment Test system was developed for East Los Angeles Community college to aid incoming students in class placement, and to assist in the collection of data that will allow the college to effectively tune the system to their needs. It will ultimately become a product that will be marketed to other colleges and universities.

Functionally, the PAT allows students to determine their projected English and Mathematics class placements by taking one or more pre-tests in each area. The Mathematics test is adaptive by level tested. The English test is split to evaluate students' English aptitudes. After placement, the PAT outputs a placement level and resources focused on the student's weak areas to assist students in preparation for the final assessment. It is customizable to individual institutions' needs—All levels of abstraction are open to customization to adapt to different testing standards and institutional requirements. Most importantly, the PAT will be completely free to end users.

Thank you for reading.

8 Appendices

8.1 Operations Manual

The client has worked closely with the team the entire class, so a user guide is largely unnecessary. However, for the sake of adequate documentation, a brief manual is included herein. Moreover, a visual guide is included here:

<http://goo.gl/T1pihW>

Upon first launch, the PAT will automatically create a superadmin, s_admin with a password not included for security reasons. This superadmin, upon logging in, will have to change his or her password immediately. He or she may then create (or later, remove) administrators by accessing the requisite form on his or landing page-Register Admin. They also have full access to the data contained in the database, accessible by using MySQL.

The administrator, once created, now has access to a variety of resources, mostly intuitive. Uploading a file in a csv format will automatically populate a test with the questions in the file. An Administrator can then manage--create, read, update, or delete--the test at any level of abstraction (Tests, Topics, Questions, Resources) by accessing the requisite "manage" link. This page provides a simple form to create or change a test that maps almost directly onto the database. They also have the ability to access any particular students info for further evaluation by clicking the relevant button on their landing page. Finally, administrators are granted access to view statistics on their tests and questions from students by accessing the provided table on the right side of the page.

Finally, the student has the easiest job. They must simply register, receive and input their password, then select a test and level to take--Math or English. From there, it's simply a matter of taking a test, getting and reviewing their results, and accessing the

provided resources. They may also review previously taken tests and their pass/fail status on them.

Refer to the provided video or the rest of this report for any other concerns with the operation of the PAT.

8.2 Alternate Designs

The client fairly strict design requirement that constrained the design of the PAT, which meant there are essentially no alternate designs worthy of consideration.

However, 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/>

Otherwise, the only alternative designs were a few alternative frameworks, such as Google App Engine and Python with MVC. However, none were sufficiently evaluated to the point where they would be useful for consideration here.