

**May15-04 Community College
Pre-Assessment and Placement
Tools: Development of the PAT**

Project Plan

Team Members:

Josh Schmitt

Peter Benzoni

Colin Duffy

Nathan Steussy

Alex Yasa

Client:

East Los Angeles Community College

Advisor:

Diane Rover

Table of Contents:

[Problem Statement](#)

[Deliverables](#)

[Specifications](#)

[Students must have the ability to:](#)

[Instructors must have the ability to:](#)

[School Administrators must have the ability to:](#)

[Concept Sketch/MockUp](#)

[Software](#)

[Functional Requirements](#)

[Non-Functional Requirements](#)

[Work Breakdown](#)

[Resource Requirements](#)

[Project Schedule](#)

[Risks](#)

[Market/Literature Survey](#)

Problem Statement

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.

Deliverables

The overall products of our development will consist of 4 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 3 most important items for us to produce.

Specifications

Students must have the ability to:

- 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 10 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”.

Instructors must have the ability to:

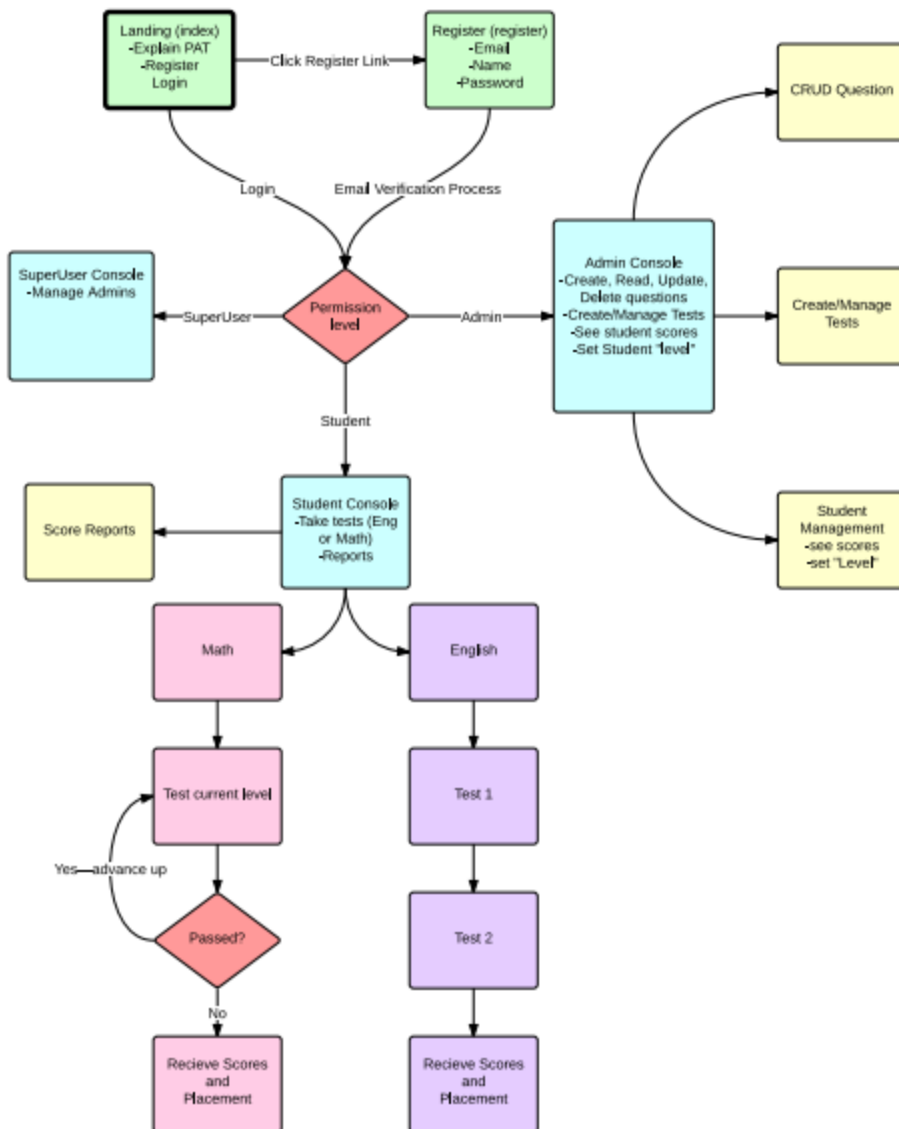
- View exams, topics, and questions that they have already created. They are displayed on the instructor home page.
- Create an exam by clicking on the appropriate button on the instructor home page. They will be prompted to select which topics to include on the exam and give the number of questions 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.

School Administrators must have the ability to:

- 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.

Concept Sketch/MockUp



Software

Production Environment:

Microsoft Visual Studio, Windows Server 2008 or Newer

Language:

C#; ASP .NET with Razor syntax.

Structure

Model, View, Controller framework

Other tools

JQuery, Twitter Bootstrap, HTML, SVN

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).

Functional Requirements

- Must allow students to take math and english tests
- Allow students to save incomplete tests that can be revisited some other time
- Allow students to review their results
- Allow students to view resources to study
- Give students proper placements
- Allow admins to create tests
- Allow admins to modify tests
- Allow admins to review students results
- Allow admins to provide resources for specific topics

Non-Functional Requirements

- Scalable to allow many users at any time
- Give students placements in a timely manner
- Interface that is simplistic enough for minimal training
- Students should be able to use the tool with zero training
- Scalable for more test types in the future

Work Breakdown

Members and component worked on

Alex

Colin

- Created initial project
- Implemented database
- Implemented user registration functionality
 - Student, Instructor and Admin
- Implemented user login functionality
 - Redirects user based on their role

Josh

- Created student profile page
- Working on page formatting
- Keeping project website updated

Peter

- Created create question page
- Lead meetings and set agenda
- Edited documentation

Nathan

- Created Student info page
- Recording meeting minutes
- Submitting weekly reports

Resource Requirements

- Windows Web Server
- SQL database

Project Schedule

	1-Sep	15-Sep	1-Oct	15-Oct	1-Nov	15-Nov	1-Dec	15-Dec	to 2015	
Project Prep	█									
Website Framework				█						
Build Database				22-Oct	7-Nov					
Implement Basic Admin Func										
Develop Testing							7-Dec			
Implement Basic User Func										
Implement MVP							7-Dec cont.			
	1-Jan	15-Jan	1-Feb	15-Feb	1-Mar	15-Mar	1-Apr	15-Apr	1-May	15-May
Implement MVP	█									
Add Extra Features		█								
Testing/Troubleshooting			█							
Mobile Development				█						
Verification, Documentation, etc.							█			

Part starts on first bold space, and ends at first empty space (or date in last bold space).

Risks

Risks	Probability	Criticality	Risk Factor	Mitigation Strategy
Not being able to effectively communicate with client due to distance	0.9	10	5	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.
Unfamiliar with C# and MVC.net framework	0.8	50	35	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	70	80	Collaborate as a team to help best design the adaptive testing algorithm.
Client decides to use different language or framework	0.05	90	10	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	90	35	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	10	20	Plan our time effectively to help mitigate any project halts due to schedule constraints.
Lack of user base for field testing	0.95	40	20	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	30	50	Continuously test and improve optimization of the project.

Market/Literature Survey

Our product is designed for potential community college students. East Los Angeles Community College is maintaining intellectual property of the pre-assessment test. Since we are signing over intellectual property rights to the client, ELAC will be handling any literature surveys and analysis.