

Glossary:

- Professor: A user who administers course material to a class.
- Class: A group of users associated with a professor.
- Student: A user who is in a class.
- Contributor: A user who can add public content to the system.
- Simulation: An interactive animation of an equation.
- Content: Key terms, definitions, examples, equations, and topics.
- New user: a user with an account under three weeks old.

1. SRS

1. Introduction

1. Aim of the document

This document is intended to introduce PhysAssist, a comprehensive physical science learning tool, and its development processes.

2. Overview of the defined system

PhysAssist is an interactive learning experience intended to help students understand physical science. Users will be able to upload equations if they do not already exist in the system already. PhysAssist will provide supplementary quizzes, an equation solver, and definitions of key terms and constants. It is designed to help users with different learning styles develop stronger physics skills.

PhysAssist is a comprehensive physics learning tool. PhysAssist has definitions, constants, equations, and simulations categorised under their respective physics topics for easy navigation and access. For example, a user navigates to the topic of orbital motion and finds a simulation as well as relevant equations, definitions, and constants. PhysAssist is the ultimate tool a student can use to enhance their understanding of a particular topic.

Simulations will be predefined, but allow for tweaking of certain parameters to create unique problems. The simulator is intended to be used by professors as a teaching aid, as well as by visual learners looking to gain a foundation on a certain concept.

PhysAssist provides equations with relevant simulations, allowing users to see the relationship between the equations and the visual simulation.

3. Stakeholders

This program is of great value to students and teachers of physics. The purpose of this program is to help the user understand intricate lessons by minimizing time spent on browsing for explanations. The indexed equations also spare the user from having to repeatedly type in constants into their calculators which allows them to focus more time on understanding the concepts. The PhysAssist team is highly invested in the success of this program and will be working diligently to bring PhysAssist into fruition.

PhysAssist content (subject, simulations, equations, and vocabulary), is created and maintained by a community of students and professionals, herein known as contributors.

4. Operational settings

To make the application as accessible as possible, PhysAssist would be most beneficial to students as a web application. Web pages can run on many different platforms, which would help PhysAssist reach the largest audience. As a JavaScript program, PhysAssist would easily be available to mobile, desktop, tablet, steambox, chromebook, or even Nintendo DSi/3DS users with ease (and many more!).

5. Related systems

PhysAssist will be a competitor to Khan Academy, a learning tool that guides independent learners via video lectures and articles. PhysAssist has a broader scope because it is a tool not only for individuals, but also for teachers in designing classroom activities such as standardised worksheets and quizzes, and as a demonstration aid. PhysAssist will easily be able to accommodate users with different learning styles.

PhysAssist includes the features of a calculator program similar to Wolfram|Alpha. Wolfram|Alpha organizes possible calculations rather than equations by subject, displaying examples instead of formulas. PhysAssist allows users to enter values directly into an interactive equation. Our program will go beyond a calculation program by focusing on a streamlined user experience and by accommodating multiple learning styles with visual simulations.

PHET (see <https://phet.colorado.edu/en/simulations/category/physics>) has Java simulations and simulation videos that cover many branches of physics, as well as other science and engineering subjects. However, these are static, downloadable videos. PhysAssist will contain online resources to dynamically build, modify and run

simulations. This approach will allow users to benefit from an interactive physics experience.

2. User Stories

1. As a person who is colorblind, I want to choose alternate color schemes, so that I can see the information more clearly.
2. As a Google Chrome user, I want to run the program in Chrome, so that I do not have to download Mozilla Firefox.
3. As a student, I want to enter known variables into a physics equation and have it solved for a selected variable, so that I can check my homework.
4. As a physics student, I want to practice solving physics problems of a selected topic, so that I can practice material I struggle with.
5. As a physics student, I want a reference for physics constants and equations, so I can save time looking for them.
6. As a student, I want to receive notifications when my professor has assigned my class a quiz, so that I do not miss an assignment.
7. As a student, I want to see all my quizzes, so that I can study for exams.
8. As a student, I want to see the definition of a key term where it is used, so that I do not have to look for the definition.
9. As a visual learner, I want to see illustrated examples of physics problems and concepts, so that I can learn on my own time.
10. As a physics professor, I want to distribute digital equation sheets to my class, so that I can save paper.
11. As a physics professor, I want to create quizzes for my students that will be graded by the system, so that I receive feedback on students' performance.
12. As a physics professor, I want to restrict the number of times a student can attempt a quiz, so I receive a more accurate indication of their current level of knowledge.
13. As a physics professor, I want to show simulations to my classes, so that I can appeal to visual learners.
14. As an inquisitive student, I want to read about the history of a selected physics equation, so that I can broaden my knowledge.
15. As a contributor, I want to add new content, so that I can expand the information available on the site.

3. Functional Requirements

1. The system shall allow contributors to create simulations.
2. The system shall display a list of topics.

3. The system shall provide a dictionary of physics constants.
4. The system shall define constants where they appear in simulations and equations.
5. The system shall display all parameters, constants, and variations of a selected equation.
6. The system shall highlight terms that are in the system's dictionary, wherever they appear, and provide the term's definition upon user request.
7. The system shall allow the user to browse all content by topics.
8. The system shall display illustrated examples of problems for a selected topic upon user request.
9. The system shall solve a physics equation for a selected parameter using user-entered values.
10. The system shall display a simulation of a selected physics equation upon user request.
11. The system shall update a simulation to represent a user's changes to the equation's parameters.
12. The system shall display the name, branch, and history of a selected equation.
13. The system shall allow professors to create online quizzes that they can assign to their students.
14. The system shall allow professors to set how many times a student can attempt a quiz.
15. The system shall display to the student a quiz attempt grade within 30 minutes of submission.
16. The system shall send an email to the professor containing all students' grades for a quiz within one day after the quiz deadline.
17. The system shall provide professors with a list of all their students.
18. The system shall allow professors to create classes.
19. The system shall allow students to join a professor's class.
20. The system shall send a notification to each student in a class within five minutes after a new quiz is assigned.
21. The system shall provide students with a list of all quizzes that have been assigned to them.
22. The system shall allow contributors to add content for all users to see.
23. The system shall allow contributors to edit existing content.
24. The system shall allow professors to distribute digital equation sheets to their classes.
25. The system shall provide practice problems for a selected physics topic.

4. Non-Functional Requirements

1. The system shall take a new user less than 2 minutes to learn how to navigate to a physics equation.
2. The system shall take a software developer with at least one year of experience no more than 2 days to fix any UI bug.

3. The system shall allow a user to input all variable fields of an equation in less than 2 minutes.
4. The system shall offer navigational hints to new users.
5. The system shall run on Android phones and tablets running KitKat (API 4.4) and above.
6. The system shall run on Windows 7, 8, and 10 and Mac OS X El Capitan and above.
7. The system shall keep a history of revisions and be able to restore them within one hour of user request.
8. The system shall allow users to report inaccurate edits.
9. The system shall retrieve selected content within 3 seconds of a user's request.
10. The system shall organize equations such that it will require no more than five clicks to reach any equation.
11. The system shall notify the user when they have entered incorrect data types into an equation.
12. The system shall allow contributors to submit translations for content.
13. The system shall display an error message and return control to the user when a simulation encounters an error.
14. The system shall have an online quiz functionality with a mean time between failures of more than ten weeks.
15. The system shall save the user's page on exit and reload it on start.

5. Use Cases



Internal Use Cases

Name: Submit Assignment

1. The student requests to view an assignment.
2. The system shows the student the prompt and attempt details for the assignment.
3. The student uploads a file.
4. The system processes the student's file and saves it to internal storage.

5. The student signifies completion.
6. The system shows the student the assignment prompt and details about the submission.

Extensions

- 1a. *The student is not logged in:* The system redirects the user to [log in](#).
- 1b. *The student does not have permission to view the course:* The system shows the user an error message prompting them to enroll or choose a new course.
- 3a. *The file type the student submitted is incorrect:* The system shows an error message to the student and prevents the user from submitting the assignment.
- 3b. *The file size is too large:* The system shows an error message to the student and prevents the user from submitting the assignment.
- 3c. *The student uploads the wrong file:* The student can delete or replace any file before submitting.
- 3d. *The current time is past the deadline for submissions:* The system shows the deadline, indicates it is in the past, and prevents further submissions.

Name: Take Quiz

1. The student requests to take a quiz.
2. The system shows the quiz details, topic, attempt limit, and time limit to the student.
3. The student requests to begin the quiz.
4. The system shows one question to the student at a time.
5. The student answers each question in the quiz.
6. The system shows the student a summary of the questions the student has answered.
7. The student submits the quiz.
8. The system shows the student the graded quiz, and records an attempt.
9. The student reviews their graded answers.

Extensions

- 2a. *The quiz has ended:* The system hides the button to take the quiz and indicates the deadline has passed.

3a. *The student attempts to take the quiz more times than allowed:* The system shows a message.

to the the student notifying them of the attempt limit, and prevents the quiz questions from being shown again.

5a. *The student returns to a previously answered question:* The system saves all the student's answers and shows the student the requested question.

5b. *The quiz has a time limit and the student takes too long:* The system shows the student an error message and ends the quiz.

6a. *The student left at least one question blank:* The system shows the student all the questions and indicates which questions are missing answers.

7a. *The system cannot save the quiz data:* The system shows the student an error message and requests that they try again.

9a. *The student has unused attempts for the quiz:* The student is given the option of retaking the quiz for a better score.

Name: View Equation

1. The student requests to view equation.
2. The system provides the list of topics.
3. The student selects a topic from the provided list.
4. The system provides the list of equations for the selected topic.
5. The student selects the equation.
6. The system provides the equation.

Extensions:

2a. *No topics to choose from:* The system notifies the user and terminates the use case.

4a. *No equations available under topic:* The system notifies the user and returns the student to list of topics (Step 2).

Name: Add Topic

1. The contributor requests to add a topic.
2. The system displays a new topic template.
3. The contributor enters a name for the topic.

4. The contributor signifies completion.
5. The system saves the new topic with the given title and directs the contributor to the new topic page.

Extensions:

1a. *The contributor is not logged in:* The system redirects the contributor to [log in](#).

4a. *The contributor leaves the topic name field empty:* The system notifies the contributor that they left the name field empty and prevents them from progressing.

Name: Add Quiz

1. The contributor creates a new quiz.
2. The system displays a new quiz in edit mode with zero questions.
3. The contributor adds questions and answer input(s).
4. The contributor selects which answer input(s) should be marked as correct.
5. The contributor gives questions point values.
6. The contributor saves quiz.
7. The system stores quiz data.

Extensions:

1a. *The contributor is not logged in:* The system redirects the contributor to [log in](#).

2a. *The system is unable to load a new quiz:* The system displays a message to try again and redirects the user to the main page.

7a. *The system is unable to save quiz data:* The system returns the contributor to previous page and displays an error message.

6. Bugs

FR - Constants Dictionary

- Original - The system shall show constants and values in a dictionary that can be accessed at any time, separate of any equation.
- Revised - The system shall provide a dictionary of physics constants.

FR - Plaintext

- Original - The system shall show physics equations in plaintext.
- Revised - The system shall allow physics equations to be copied from the application in plaintext.

FR - Constants with simulations/equations

- Original - The system shall define constants alongside simulations and equations.
- Revised - The system shall define constants where they appear in simulations and equations.

FR - Email Students

- Original - The system shall send an email to each student in a professor's list, notifying them that a new quiz has been posted.
- Revised - The system shall send a notification to each student in a class within five minutes after a new quiz is assigned.

FR - Email Professor

- Original - The system shall send an email containing all grades to the quiz creator within one day after the quiz due date passes, or after all the attempts for the quiz have been made.
- Revised - The system shall send an email to the professor containing all students' grades for a quiz within one day after the quiz deadline.

NFR - Navigability of Equations Listings

- Original - The system shall organize equations such that a user will have to click at most 5 times to reach any equation in the system.
- Revised - The system shall organize equations such that it will require no more than five clicks to reach any equation.

NFR - System Learn Time

- Original - The system shall take a user no more than 2 minutes to learn how to navigate to a physics equation.
- The system shall take a new user less than 2 minutes to learn how to navigate to a physics equation.

NFR - OS Support

- Original - The system shall run on Windows 7, 8, and 10; and Mac OS captain and above, so that it will be accessible to most students.
- Revised - The system shall run on Windows 7, 8, and 10 and Mac OS X El Capitan and above.

NFR - UI Structure

- Original - The system shall take a software developer with 1 year of experience less than 2 days to correct any UI defect.
- Revised - The system shall take a software developer with at least one year of experience no more than 2 days to fix any UI bug.

NFR - No Unhandled Exceptions

- Original - The system shall have no unhandled exceptions from incorrect user input.
- Revised - Removed

NFR - Quiz Learn Time

- Original - The system shall take a general user no longer than 5 minutes to learn how to create a quiz.
- Revised - Removed

NFR - Hints

- Original - The system shall offer navigational hints for users.
- Revised - The system shall offer navigational hints to new users.

NFR - MTBT for quiz functionality

- Original - The system shall have a mean time between failures greater than 10 weeks for the online quiz functionality.
- Revised - The system shall have an online quiz functionality with a mean time between failures of more than ten weeks.

NFR - Error Control

- Original - The system shall close itself if it experiences a freeze that lasts longer than 30 seconds.
- Revised - The system shall display an error message and return control to the user when a simulation encounters an error.

NFR - UI Structure

- Original - The system shall take a software developer with 1 year of experience less than 2 days to correct any UI defect.
- Revised - The system shall take a software developer with at least one year of experience no more than 2 days to fix any UI bug.

2. Storyboard

<http://users.csc.calpoly.edu/~dsabsay/cpe308/storyboards.html>

3. Design Decision Making Rationale Documentation