General Description

This is an educational game about learning new concepts by entering their descriptions, repetitive studying of these descriptions, and taking subsequent tests to verify acquired knowledge. The player enters the name of the concept in one textbox and its description in another, possibly multiple concepts and descriptions in a single session, and saves the session’s record. Then, the player can bring up this session (or any other session previously recorded) and verify the knowledge by taking the test (playing the game) to see if the concepts are understood. The tests (games) may include a variety of techniques, for example, matching questions, multiple choice questions, dynamic multiple choice questions, fill-in-the-blanks, etc.

Definitions and Acronyms

SLS – Self-Learning Software (tentative name)
Main User Interface General Requirements:

Requirement #001
Registration Screen

The interface shall require new users to register.

Source: Scott LaForge
History:

Requirement #002
Login Screen: The interface shall require users to login to access the Main Page of the site.

Source: Scott LaForge
History:

Requirement #003
Main Page Screen: The screen shall provide links, allowing the user to:

- choose to continue an active game
- create a new game
- select an available game
- review and provide feedback on previously played games
- logout
- view rules regarding the different games on the site
- view details on how to use the site

Source: Scott LaForge
History:

Requirement #004
Create New Game screen: This screen shall allow the user to create and submit word/description pairs with the functionality illustrated in Fig. 1 of Draft #0, where:

- The PLUS “+” button in the left column, when pressed, shall cause additional row to appear underneath the current row.
- The MINUS “-” button in the rightmost column, when pressed, shall cause the current row
disappear.

• Left textbox shall allow entering by a user a word or a phrase of up to TBD characters long
• Right textbox shall allow entering by a user the description (definition) of the word or phrase, which appears in the left textbox.
• The END button at the bottom, when pressed by a user, shall cause: (1) the completion of the data entering process, and (2) recording of the data.

Source: Scott LaForge

History:

**Requirement #005**  
**Create New Game screen:** The "Review Submission" screen shall be displayed when the END button is pressed.

Source: Scott LaForge

History:

**Requirement #006**  
**Create New Game screen:** The word/description pair and submission date shall be inserted into the user's dataset when the END button is pressed.

Source: Scott LaForge

History:

**Requirement #007**  
**Review Submission screen:** This screen shall display only those word/description pairs just added to the user's dataset.

Source: Scott LaForge

History:

**Requirement #008**  
**Review Submission screen:** The screen shall allow the user to modify the word/description pairs.

Clarification: How will the program allow the user to do this? What can/can't be done in this screen?

Source: Scott LaForge

History:

**Requirement #009**
**Review Submission screen:**  The screen shall allow the user to submit the word/description pairs to the public dataset.

Source: Scott LaForge

History:

**Requirement #010**

**Review Submission screen:** The "Create New Game" screen shall be displayed with the boxes pre-populated with the current pairs when the modify button is selected.

Clarification: Replace "current pairs" with "most recent entries".

Source: Scott LaForge

History:

**Requirement #011**

**Review Submission screen:** The data shall be inserted into the public data set when the submit button is selected.

Source: Scott LaForge

History:

**Requirement #012**

**Review Submission screen:** The user's screen name shall be added to the "Available Games" queue, accessed in the "Available Games" screen, when the submit button is selected.

Source: Scott LaForge

History:

**Requirement #013**

**Review Submission screen:** The "Main Menu" screen shall be displayed when the submit button is selected.

Source: Scott LaForge

History:

**Requirement #014**

**Active Games screen:** This screen shall provide links to all of the user's existing games on separate lines.

Source: Scott LaForge
History:

**Requirement #015**

**Active Games screen:** Each line shall display: "<user who instantiated game>'s game with <other user>: <which user's> turn."

Source: Scott LaForge

History:

**Requirement #016**

**Active Games screen:** Each link shall be activated and clickable when it is the logged in user's turn.

Source: Scott LaForge

History:

**Requirement #017**

**Active Games screen:** Each link shall lead to one and only one active game.

Source: Scott LaForge

History:

**Requirement #018**

**Available Games screen:** This screen shall access a queue of names derived from user's who have added word/description pairs in the "Review Submission" screen.

Clarification: Only users who have entered word/description pairs are allowed to play games? Can the public data set be used when the user has an empty data set?

Source: Scott LaForge

History:

**Requirement #019**

**Available Games screen:** This screen shall display links, listed in order from oldest to newest date, displaying "Join crossword game with <username>.'

Source: Scott LaForge

History:

**Requirement #020**

**Available Games screen:** The specific "Game" Screen shall be displayed according to the particular link
Requirement #021
Available Games screen: The <username> shall be removed from the "Available Games" queue when a specific game link is selected.
Source: Scott LaForge
History:

Requirement #022
Available Games screen: Both users shall be added to an "Active Game" queue stored in the site's database when a specific game link is selected.
Source: Scott LaForge
History:

Requirement #023
Available Games screen: Each time a user pair is added to "Active Game" queue, the program shall randomly select word/description pairs from the public data set and associate them with the specific game.
Source: Scott LaForge
History:

Requirement #024
Review Games screen: The program shall display a list of links to completed games, from most recent to oldest.
Source: Scott LaForge
History:

Requirement #025
Review Games screen: The game screen shall be displayed, populated with the entire crossword game filled in, when a specific completed games link is selected.
Source: Scott LaForge
History:
Requirement #026

Review Games screen: A pop up window shall be displayed where the user can provide feedback when the user selects a description corresponding to a word filled in on the crossword puzzle.

Source: Scott LaForge

Requirement #027

Review Games screen: The pop-up shall have both a submit and cancel button.

Source: Scott LaForge

Requirement #028

Review Games screen: The pop up screen shall disappear, losing all written data when the cancel button is selected.

Source: Scott LaForge

Requirement #029

Review Games screen: The feedback shall be added to the word/description pair in the database when the submit button is selected.

Source: Scott LaForge

Requirement #030

"Backend" Screen: This screen shall allow the administrator to set the maximum number of word/description pairs per game.

Source: Scott LaForge

History:

Conflict: With the requirements listed in the crossword puzzle game, the program allows the user to specify the number of word/description pairs to use in the current game.

Requirement #031

"Backend" Screen
This screen shall allow the administrator to add/remove word/description pairs. 
Source: Scott LaForge 
History:

Requirement #032
"Backend" Screen
This screen shall allow the administrator to review the subsets of a particular user's dataset. 
Source: Scott LaForge 
History:

Requirement #033
"Backend" Screen
This screen shall allow the administrator to review feedback provided by all users. 
Source: Scott LaForge 
History:

Requirement #034
"Backend" Screen: 
This screen shall allow the administrator to modify the length of time for each player's turn. 
Source: Scott LaForge 
History:

Requirements by Andon Coleman
Requirement #101. 
Shall use an HTTP 1.1-compliant server with server-side hypertext processing to generate dynamic multiple-choice quizzes. 
Source: Andon Coleman 
History:
Description of defect: The environment shouldn’t be so specific, the software may be implemented in various environments and still meet the requirement. The second half should be a separate, more detailed requirement in and of itself. The software is not specifically referred to when stating that it shall use.
Reason (criterion applied): clarity, correctness

**Requirement #102.**
Shall be playable on an HTML 4.0-compliant web browser, relying on HTML’s form/submit functionality.
Source: Andon Coleman
History:
Description of defect: The same first reason as the defect from requirement 1. Who/what is the source of the requirement is not identified.
Reason: clarity, correctness

**Requirement #103.**
Shall be capable of hosting multiple simultaneous players.
Source: Andon Coleman
History:

**Requirement #104.**
Shall facilitate user-created questions, and a set of multiple correct answers, such that “All of the above” and “None of the above” can be implemented.
Source: Andon Coleman
History:
Description of defect: Facilitate user-created questions is too ambiguous. This can also be split into separate more detailed requirements. Source also not identified.
Reason: clarity, correctness, completeness

**Requirement #105.**
Shall randomize order of questions, multiple-choice options, and randomly insert one or more incorrect responses from other questions.
Source: Andon Coleman
History:
Description of defect: Contains several requirements in one sentence, should be split. Source not identified.
Reason: clarity, correctness, completeness
**Requirement #106.**
Shall keep track of time spent per-question as well as total time, and support a maximum allotment of time before failing a question, or the entire quiz.

Source: Andon Coleman

**History:**

**Requirement #107.**
Shall generate, display (client-side) and record statistics (server-side) for each completed session.

Source: Andon Coleman

**History:**

Description of defect: Very ambiguous, should be spelled out as a user interface set of requirements.

Source not identified.

Reason: clarity, correctness, completeness.

The choice of HTTP server, server-side hypertext processor, and preferred web browser are not included in the initial requirements specification. These requirements will be made more specific when a survey of available hardware / software is conducted prior to detailed design.

**Notes. Potential Design Constraints**

Requirements 2, 4 and 7 will require designing a user interface of some sort using HTML, and server-side hypertext processing.

Description of defect: very ambiguous, needs to be clarified and specified.

Reason: clarity, correctness, completeness.

Requirement 3 can be satisfied by using the concept of HTTP sessions.

Requirement 6 may be handled on the server-side and/or client-side using JavaScript.

Functional requirements 4-7 are, in my opinion, the most important part of the proposed game. They are intended to prevent users from memorizing answers to a given quiz based largely on the order of the questions and the possible responses. Focusing on generating dynamic quizzes and statistical feedback will improve the game’s educational value significantly.
Requirement #201

The game shall use flash cards that were created by the user during the current session or a previous session.

Source: Jorge Ocañas

History:

Description of defect: When the flash cards were created is unimportant.

Reason (criterion applied): Clarity

Suggested fix: “The game shall use user-created flash cards.”

Requirement #202

“The flash cards will consist of two parts, the concept, and the description.”

Source: Jorge Ocañas

History:

Description of defect: The software should be the subject…

Reason (criterion applied): Clarity

Suggested fix: “The game will use flash cards that consist of two parts, …”

Requirement #203

“The software will give the user an option to end his/her session at any time.”

Source: Jorge Ocañas

History:

Requirement #204

“The game will consist of three rounds of interactive questions.”

Source: Jorge Ocañas

History:

Requirement #205

“For questions in the first round, the game shall provide the user with two choices on screen; one correct description and one incorrect description of the concept displayed on screen.”
Source: Jorge Ocañas

History:

Description of defect: May contradict requirement #3, because this requirement implies there are a total of two user choices on the screen.

Reason (criterion applied): Consistency and Correctness

Suggested fix: “… the game shall include two question responses …”

Requirement #206

“For questions in the second round, the game shall provide the user with up to five choices on screen; one will be a correct description, while the other four will be incorrect.”

Source: Jorge Ocañas

History:

Description of defect: May contradict requirement #3, because this requirement implies there are a total of no more than five user choices on the screen.

Reason (criterion applied): Consistency and Correctness

Suggested fix: “… the game shall include up to five question responses …”

Requirement #207

“For questions in the third round, the game will only provide the user with the concept and a location to input his/her response.”

Source: Jorge Ocañas

History:

Requirement #208

“For each question, the game shall notify the user if he/she is correct in their choice.”

Source: Jorge Ocañas

History:

Description of defect: When this requirement takes place is unclear.

Reason (criterion applied): Clarity

Suggested fix: “The software shall notify the user when he/she correctly answers a question.”

Requirement #209

“For each question, the game will also update the user’s current session score.”

Source: Jorge Ocañas
History:
Description of defects:
- When this requirement takes place is unclear.
- “also” implies that this requirement has multiple parts.
Reason (criterion applied): Clarity
Suggested fix: “The game shall update the session score when a question is answered.”

Requirement #210
“The score will be an average of the user’s overall accuracy in answering the questions.”
Source: Jorge Ocañas

Requirement #211
“A comfort level shall be between 80-100 percent accuracy in answering the last n questions. n is defined by stack size * 0.3. For example, a stack of 10 would have an n of 3.”
Source: Jorge Ocañas

Requirement #212
“Each individual round will continue until 30 percent of the stack has been used as questions and a certain comfort level is achieved. After these two requirements are met, the round will end.”
Source: Jorge Ocañas
Description of defects:
- The software should be the subject…
- This requirement is two sentences
Reason (criterion applied): Clarity
Suggested fix: “The game will end a round when …”

Requirement #213
The user will only be given one chance to answer each question correctly before moving on to another question.
Source: Jorge Ocañas

History:
Description of defect: The software should be the subject…
Reason (criterion applied): Clarity
Suggested fix: “The game will give the user …”

Requirement #214
“The game will end when the user decides to exit, or when the third round ends as defined by requirement 12.”
Source: Jorge Ocañas

History:
Description of defect: This requirement restates Requirements #3 and #4.
Reason (criterion applied): Consistency
Suggested fix: Remove this requirement – it is completely redundant.

Requirement #215
“At the end of the game, a user’s accuracy is displayed to the user as a score.”
Source: Jorge Ocañas

History:
Description of defect: This is not a “shall/should” statement.
Reason (criterion applied): Clarity
Suggested fix: “The game will display a user’s accuracy in the form of a score <work in the definition from Requirements #10 and #16> at the end of a game.”

Requirement #216
“The score must at least be the average accuracy of answering all questions across all rounds.”
Source: Jorge Ocañas

History:
Description of defect: This is not a software requirement, it is a fragment, and appears to do nothing more than restate Requirement #10.
Reason (criterion applied): Clarity and Consistency
Suggested fix: Figure out how this requirement is supposed to be different from #10, and restate it more clearly.
Or remove the requirement if it is truly redundant.

Requirement #217
“The game may include more comprehensive scoring but not be limited to ideas such as: individual round accuracy, problem cards (cards with low comprehension), individual card accuracies, and time.”

=================================================================

Jeopardy-Like Game, Justin Hodnett
By: Josh Yanai

General Description
This is a game to test concept knowledge by utilizing a game similar in style to Jeopardy with concepts and descriptions being entered previously into the system in the manner described in the Self Learning game document prepared by Dr. Janusz Zalewski. The user has a number of tiles to choose from ranging in points. The user chooses a tile and the description is shown to the user. The system then displays a choice of 4 concepts for the user to choose from. Points are awarded to the user when a correct choice is met.

Requirement #301
The game shall have the primary user interface with the functionality illustrated in Fig. 1 where the user selects a block with a certain point value.
Source: Justin Hodnett
History:
Description of defect: One cannot place a requirement on the user. The requirement should be placed on the software.
Reason (criterion applied): Clarity and correctness
Correction: Requirement #1
Upon startup, the gaming software shall display the primary user interface illustrated in Fig. 1

Requirement #302.
When a button with a points value is selected, the user shall be presented with a description and 4 possible concepts (one is correct).
Source: Justin Hodnett
History:

The study of the theoretical foundations of computation using logic, mathematics, and formal languages to realize a practical implementation.
Description of defect: One cannot place a requirement on the user. The requirement should be placed on the software. Does not state how a button is selected.
Reason (criterion applied): Clarity and correctness
Correction: Requirement #2
When a button in Fig. 1 is selected via mouse click, the gaming software shall display a question along with 4 possible answer buttons as shown is Fig. 2
Note: 1 of the 4 concepts is the correct answer.

Requirement #303
If the user answers the question correctly, the game shall award the appropriate level of points to a total. If the user answers incorrectly, the system shall display the correct answer in an obvious way and award no points and return to Fig. 1.
Source: Justin Hodnett
History:
Description of defect: One cannot place a requirement on the user. The requirement should be placed on the software. Does not state how an answer is selected. A requirement cannot contain multiple sentences and requirements. The requirement does not specify how to return to Fig. 1.
Reason (criterion applied): Clarity, correctness and completeness
Correction: Requirement #3
If the correct answer button is selected via mouse click, the gaming software shall add the number of points, equal to the point value of the selected button in requirement #1, to the user’s total score.
Correction: Requirement #4
If the incorrect answer button is selected via mouse click, the gaming software shall display the correct answer and no points shall be added to the user’s total score.
Correction: Requirement #5
Whether the answer selected is correct or incorrect, a button that says “continue” shall be displayed on the screen.
Correction: Requirement #6
When the continue button is selected via mouse click, the gaming software shall display Fig. 1 with the previously selected Fig.1 buttons disabled.

Requirement #304.
After all questions have been answered, the system shall display the overall score and a grade. The system then shall return to the selection screen.
Source: Justin Hodnett

History:
Description of defect: Does not state how score and grade are derived. A requirement cannot contain multiple sentences and requirements. The requirement does not specify how to return to the selection screen. The requirement does not define what the selection screen is.
Reason (criterion applied): Clarity, correctness and completeness
Correction: Requirement #7
Once all buttons in Fig. 1 have been disabled, the game shall calculate and display the user’s overall score and a grade.
Correction: Requirement #8
The game shall also display a button that says “exit”.
Correction: Requirement #9
When the exit button is selected via mouse click, the gaming software shall terminate.

================================

Original Document: Josh Yanai
Reviewed by: Justin Hodnett

**Requirement #401**
Once the game program is running, it shall require the user to select the test that they would like to use during game play.
Source: Josh Yanai

History:
Description of defect: We should assume that the game is running when making requirements of it.
Reason (criterion applied): Correctness.
Suggested Correction: The software shall require the user to select the test that they would like to use during game play.

**Requirement #402**
After a test has been selected, the program shall require the user to press the enter key (keyboard) to start game.
Source: Josh Yanai

History:
Requirement #403
When game play starts, the program shall start a timer.
Source: Josh Yanai
History:

Requirement #404
The program shall display the timer in the upper-right corner of the screen.
Source: Josh Yanai
History:

Requirement #405
The program shall also display a vehicle on the screen.
Source: Josh Yanai
History:

Requirement #406
The vehicle shall be controlled via keyboard input.
Source: Josh Yanai
History:

Requirement #407
There shall be a number of check points along the race path.
Source: Josh Yanai
History:
   Description of defect: Race path is undefined.
   Reason (criterion applied): Clarity.
   Suggested Correction: Define the “race path” in the SRS in a definitions section.)

Requirement #408
The number of check points shall equal the number of questions in the selected test.
Source: Josh Yanai
History:
   Description of defect: Redundancy from [Requirement 7] to [Requirement 8].
Reason (criterion applied): Consistency.
Suggested Correction: (Combining [Reg. 7] and [Req. 8] into) There shall be a number of checkpoints throughout the game equal to the number of questions in the selected test.)

**Requirement #409**
When a check point is reached, the game program shall display a question from the selected test.
Source: Josh Yanai
History:

**Requirement #410**
The program shall select a random question from question set.
Source: Josh Yanai
History:
Description of defect: Redundancy from [Requirement 9] to [Requirement 10]. Reason (criterion applied): Consistency. Suggested Correction: (Combining [Req. 9] and [Req. 10] into) When a checkpoint is reached, the game shall display a randomly selected question from the current test.

**Requirement #411**
The program shall require the user to press a, b, c or d on the keyboard to answer question.
Source: Josh Yanai
History:
Description of defect: “The program shall require the user to press a,b,c or d on the keyboard to answer question.”
Reason (criterion applied): Clarity.
Suggested Correction: [New Requirement] The game shall display 4 answers: 1 correct answer and 3 incorrect answers. [Requirement 11] The program shall require the user to press a, b, c or d …”

**Requirement #412**
If the selected key matches the correct answer the game program shall allow the vehicle to continue on with the race.
Source: Josh Yanai
History:

**Requirement #413**
If an incorrect answer is selected the game program shall force the vehicle back to the previous check point.
Source: Josh Yanai
History:

**Requirement #414**
If an incorrect answer is selected and the vehicle is at the first check point, the vehicle shall be forced back to the starting line.
Source: Josh Yanai
History:
Description of defect: Starting line is undefined.
Reason (criterion applied): Clarity.
Suggested Correction: Define the “starting line” in the SRS in a definitions section.

**Requirement #415**
When the vehicle has reached the finish line, the game program shall stop the timer.
Source: Josh Yanai
History:
Description of defect: Finish line is undefined.
Reason (criterion applied): Clarity.
Suggested Correction: Define the “finish line” in the SRS in a definitions section.

**Requirement #416**
The game program shall record the players score.
Source: Josh Yanai
History:
Description of defect: There is no mention of how the player is scored or if it displayed to the user, only that it is recorded.
Reason (criterion applied): Completeness.
Suggested Correction: [New Requirement] The software shall score the player using... [New Requirement] The software shall display the players score...

**Requirement #417**
The game shall require the user to press enter or esc on the keyboard to return to the main menu.
Requirement #418
Thou shall not kill…
Source: Josh Yanai
History:
Description of defect: Unnecessary to the SRS. Reason (criterion applied): Correctness. Suggested Correction: Remove from SRS.

Nicholas Alteen & Scott LaForge, Web-Based Crossword Puzzle

The proposed project idea is a web-based crossword puzzle. The following are requirements and suggestions of how the program will function

Definitions

Grid: Refers to the actual crossword playing space, where the user is able to enter letters to solve the puzzle.

Requirement #501

The crossword puzzle module’s menu must have a field for the number of words and two radio buttons for selecting the option to use either the current user’s session or all sessions in the game.
Source: Nicholas Alteen
History:
Description of Defect: Refer to Figure 1 in the requirement.
Reason (criterion applied): Clarity.
Correction: Simply add at the beginning of requirement, "As shown in figure 1 below, the crossword
puzzle module's..."

Requirement #502

The program must not allow both radio buttons to be selected at the same time.

Source: Nicholas Alteen

History:

![Crossword Menu UI](image)

**Figure 1: Crossword Menu UI**

Requirement #503

When the user attempts to start a crossword puzzle, the program must check to see that all fields have correct values.

Source: Nicholas Alteen

History:

Description of defect: Requirement placed on user.
Reason (criterion applied): Correctness.
Correction: Start with, "The program must check to see that all fields have correct values when the user attempts to start a crossword puzzle."

Requirement #504
If the number of words field contains illegal characters (any characters other than ‘0’ through ‘9’), the program must display an error message and prompt the user for a new value.

Source: Nicholas Alteen

History:

Requirement 5

When the user starts the game, the program must dynamically create a crossword puzzle (selecting words and definitions stored on another computer).

Source: Nicholas Alteen

History:

Description of defect: Requirement placed on user.
Reason (criterion applied): Correctness.
Correction: Start with, "The program must dynamically create a crossword puzzle (selecting words and definitions stored on another computer) When the user starts the game."

Requirement #506

The program should attempt to intersect words wherever possible.

Source: Nicholas Alteen

History:

Requirement #507

The program must expand the crossword grid as needed to accommodate all words.

Source: Nicholas Alteen

History:

Requirement #508

The program must not use duplicate words.

Source: Nicholas Alteen

History:

Requirement #509

The program will number the beginning of each word in the puzzle, from 1 to the value entered in the number of words field.

Source: Nicholas Alteen
**Requirement #510**

The program must place the definitions of the words used in the crossword puzzle in the “Across” and “Down” columns, corresponding to the placement and orientation of the words on the grid.

Source: Nicholas Alteen

**Requirement #511**

The program must number the definitions in the “Across” and “Down” columns.

Source: Nicholas Alteen

**Requirement #512**

The numbers given words on the grid must correspond to the correct definitions listed in the “Across” or “Down” tables.

Source: Nicholas Alteen

History:

Description of defect: May be redundant as it seems to already be addressed in Requirement #10.

Reason (criterion applied): Consistency.

Correction: Check with Professor Zalewski to determine whether this is considered redundant.

**Requirement #513**

The crossword game must have buttons for the user to restart or quit.

Source: Nicholas Alteen

**Requirement #514**

The game menu should have buttons that will give the user hints.

Source: Nicholas Alteen

**Requirement #515**
The hints provided should include highlighting correctly placed letters/words, or randomly inserting a correct letter into the grid.
Source: Nicholas Alteen

History:

**Requirement #516**
When the user selects a particular description, a pop up window must be displayed, where the program will display a number of spaces equal to the number of characters of the answer.
Source: Scott LaForge

History:
Description of defect: Requirement placed on user. Reason (criterion applied): Correctness.
Correction: "The program shall display a pop up window that displays the number of spaces equal to the number of characters in the grid when a particular description is selected."

**Requirement #517**
If any letters were previously populated from another question which intersects the current selected question, these letters shall populate the appropriate position in the pop-up window.
Source: Scott LaForge

History:
Description of defect: Some vagueness.
Reason (criterion applied): Clarity.
Correction: The previously answered question should intersect the current one.

**Requirement #518**
The pop up window shall display a ‘submit’ and ‘cancel’ button.
Source: Scott LaForge

History:

**Requirement #519**
If cancel is selected, the pop up window shall be removed and the player shall be able to select a different description to answer.
Source: Scott LaForge

History:
(**Description of defect:** Requirement placed on user.  **Reason (criterion applied):** Correctness.  
**Correction:** "The pop up window shall include a cancel button, allowing the player to select a different description to answer.")

**Requirement #520**

A score shall be displayed above the crossword puzzle board where each user is listed with the score next to each username.

Source: Scott LaForge

History:

**Requirement #521**

When a player submits the correct answer, their score shall be increased by one point for every letter they typed correctly in the pop up window.

Source: Nicholas Alteen

History:

Description of defect: Requirement placed on user.
Reason (criterion applied): Correctness.
Correction: "The program shall increase the score by one point for every letter typed correctly in the pop up window.")

**Requirement #522**

The game screen shall have a timer located in the upper right hand corner that starts upon entering the active game screen. The timer shall count down starting from the length of time as chosen by the administrator.

Source: Scott LaForge

History:

Description of defect: More than one sentence.
Reason (criterion applied): Correctness.
Correction: Should be split into two requirements.

**Requirement #523**

Upon completion of the timer, the game screen shall automatically be closed.

Source: Scott LaForge

History:
Requirement #524
When a correct word is entered, the definition will be crossed out of the “Across” or “Down” columns.
Source: Nicholas Alteen
History:
Description of defect: Requirement placed on user.
Reason (criterion applied): Correctness.
Correction: "The program shall cross out the "Across" or "Down" columns when it detects that a correct word was entered.")

Requirement #525
When all words have been entered correctly, the user will receive a congratulatory message, and the program will return to the starting crossword menu.
Source: Nicholas Alteen
History:
Description of defect: (1) Requirement placed on user. (2) more than one requirement.
Reason (criterion applied): (1) Correctness, (2) Correctness.
Correction: (1) "The program shall display a congratulatory message when the entire board is completed."
(2) New Requirement: "The program will return to the starting crossword menu when the game is completed.")