

CASE-BASED SIMULATION

Requirements Document

Team Members:

Jacob Schmitt, Timothy Mai, Austyn Mancini, Kuan Liu, Nathaniel Hector, Sean Jerzewski

Sponsor:

Neil Toporski

Prepared by Jacob Schmitt on 2/25/2024

Requirements Document

PREFACE

The Requirements Document outlines the scope, and resource requirements for the implementation of the Case-Based Simulation. This report provides the information necessary to support the High-Level Scope and Sizing and defines the Case-Based Simulation project in sufficient detail to support the future Design Phase.

This Requirements Document was developed by the Pandas Team and is based on information gathered from discussions with the sponsor, Neil Toporski. It provides a summary of information gathered to date and is organized into the following chapters:

Introduction

Summarizes the Case-Based Simulation project and its business value and describes how this system accomplishes the goal of the sponsor.

Glossary

Defines the technical terms used in the document.

Functional Requirements

Describes the functional services provided by the application first in a broad natural language with user requirements, then in more specific technical terms with the system requirements. There are also activity diagrams included to visually demonstrate the requirements.

Nonfunctional Requirements

Presents a high-level overview of the anticipated system architecture.

Out Of Scope Requirements

Describes the requirements of the system that have been deemed to be out of scope for this prototype.

System Evolution

Details future application requirements of the system and areas for further development.

Requirements Document

TABLE OF CONTENTS

1	INTRODUCTION.....	ERROR! BOOKMARK NOT DEFINED.
1.1	SYSTEM SUMMARY.....	ERROR! BOOKMARK NOT DEFINED.
1.2	BUSINESS VALUE.....	ERROR! BOOKMARK NOT DEFINED.
1.3	DEFINITION OF DONE	ERROR! BOOKMARK NOT DEFINED.
2	GLOSSARY	ERROR! BOOKMARK NOT DEFINED.
3	FUNCTIONAL REQUIREMENTS	3
3.1	LOGIN	3
3.2	DEVELOPER MAIN MENU	6
3.3	DEVELOPER AVAILABLE SIMULATIONS	8
3.4	DEVELOPER DELETE SIMULATIONS	9
3.5	DEVELOPER SIMULATION RESULTS	9
3.6	USER MAIN MENU	10
3.7	DEVELOPER NODE CREATION.....	13
3.8	Create Simulation Menu	18
3.9	SIMULATION PLAYER.....	22
3.10	ACCOUNT STORAGE.....	24
3.11	FILE DISPLAY AND STORAGE	25
3.12	USER NODE VIEWS	27
4	NONFUNCTIONAL REQUIREMENTS	29
4.1	COMPATABILITY	29
4.2	RESPONSIVENESS.....	29
4.3	MEMORY PERFORMANCE.....	29
4.4	DOCUMENTATION	29
5	OUT OF SCOPE REQUIREMENTS	30
5.1	ACCOUNT MONETIZATION.....	30
5.2	INFORMATION SECURITY	30
5.3	SERVER BALANCING.....	30
6	SYSTEM EVOLUTION.....	30
6.1	ACCOUNTS.....	30
6.2	SIMULATION DEADLINES	30
6.3	TIME AS PART OF SCORE.....	30
6.4	INCREASED MEDIA INPUTS	30
6.5	SIZE OF TREE	30

Requirements Document

1 Introduction

1.1 System Summary

The Case-Based Simulation Developer is a Web application that will allow developers to design and construct authentic simulated environments where users can then access and interact with those simulations.

The Case-Base Simulation consists of four primary user interventions or “nodes:” scenario, information-gathering, decision-making, and end. The user starts with a problem set, or scenario node. Then, the simulation as a decision tree allows the user to gather data and to make decisions based on environmental or situational conditions, leading them through a series of multiple and possible branched pathways. The user completes the simulation at the end node, where their judgments (how effectively they interacted with the simulation) are evaluated.

The scope of the Case-Based Simulation is to create a Web application where developers can build, edit, save, or delete case-based simulations for users to access and take prepared simulations.

1.2 Business Value

This project has many possible applications in education and skill evaluation in various fields, saving lots of time creating and distributing assessments, and making individualized and accurate skill assessment much more efficient for evaluators.

1.3 Definition of Done

For any new/added functionality to be considered done it must abide by these requirements:

- The code is fully commented so that future developers of the system can understand the purpose and design of the current implementation.
- The code has passed a series of unit tests and regression tests to ensure that both the new functionality and the existing functionality works correctly.
- Diagrams should be created and delivered periodically to the sponsor representing added functionality in sufficient detail as it would appear in the User Manual document.
- The speed performance of the system remains consistent to that of a usual website (one second at most). If the loading time takes longer than this, then a loading bar of the progress needs to appear so that the user remains engaged with the site.
- The memory usage of the system remains minimal with optimizations put in place for storing videos and photos. These should include reducing the size for easier storage and modal pop-up windows being used when applicable as well.

Requirements Document

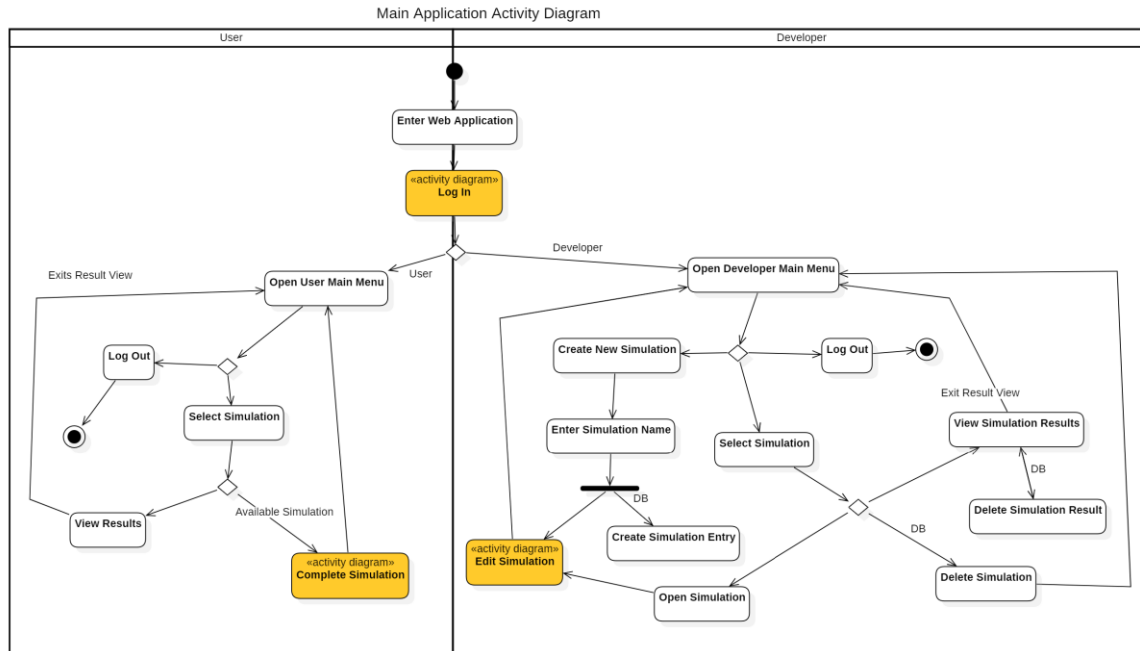
2 Glossary

Term	Description
Simulation	The entire set of nodes that the user can traverse in their decision making and information gathering process. These are made by the developer and completed by the user.
User	A person (a student/trainee) that uses the simulation to be evaluated on their performance of responses and actions within the simulation environment.
Developer	A person who creates/builds a simulation using the product.
Node	Each step in the simulation path. There are 4 types of nodes: 1. Scenario (the start of the simulation) 2. End (the end of the simulation) 3. Information Gathering (IG) 4. Decision-Making (DM)
Scenario Node	The problem set. What introduces the user to the environmental situation/case/problem they need to solve. The Scenario node is the first node. It introduces the problem set to the user. There is only one Scenario child node.
Information Gathering (IG) Node	The type of node that allows the user to check (appropriate or inappropriate) data to better understand the current conditions of the simulation. There is only one IG child node.
Decision Making (DM) Node	The type of node that allows the user to make choices or decisions There are between 1 to 5 child nodes.
End Node	Last node in the simulation that provides the user with a summary and evaluation of their decision-making and information gathering skills as it relates to the simulation. The end node has no children and all paths eventually lead to the end node.

Requirements Document

3 Functional Requirements

Overall Activity Diagram:



3.1 Login

Total Business Value: 91

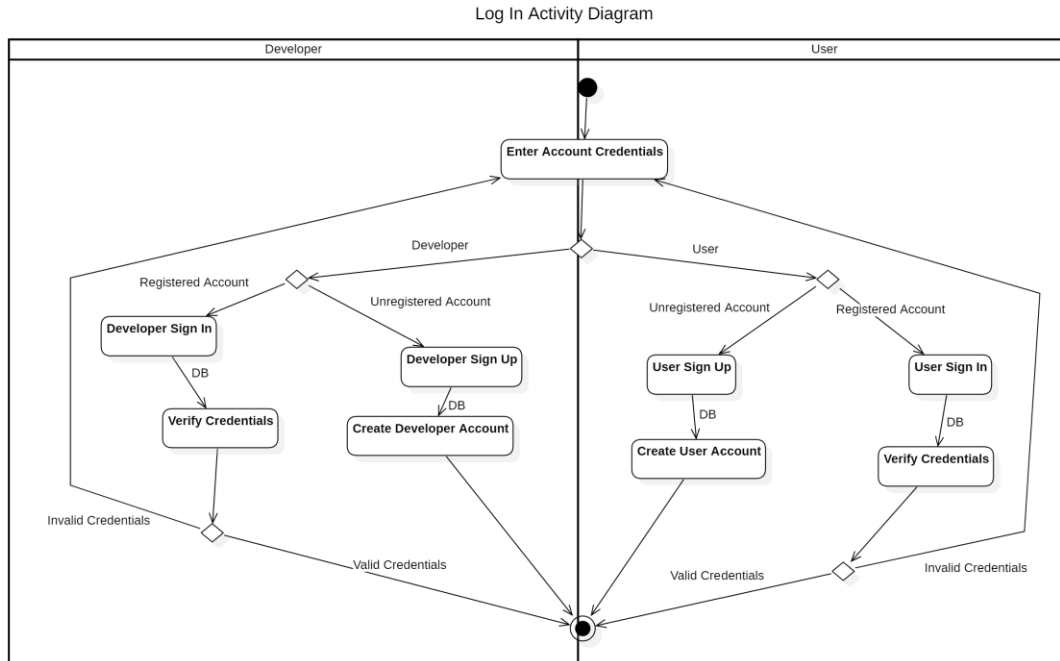
Total Difficulty: 66

User Description:

The Case-Based Simulation web application should allow a person accessing the site the ability to sign up for a user account in which they can take simulations or sign up as a developer so that they can create their own simulations. If the person has already created an account, they should be able to login to their respective developer or user account using the credentials they signed up with. These accounts should have simulations related to them. For developers it would be the simulations they have created and for users it would be the simulations that have been assigned to them.

Activity Diagram:

Requirements Document



System Description:

Req#	Requirement	Requirement Description	BV	Difficulty
1.1	The landing page of the application must display the application's title and provide a means for the user to log in.	Upon entering the software, you will be brought to a sign in page. On this page in the middle of the left part of the screen will be text in a title format saying, "Case Based Scenario". On the right side of the page will have text saying "Sign in" where fields will eventually be placed below it. The background of the page is just a plain white screen.	13	8
1.2	To log in there must be two areas for credentials to be entered which are the person's last name and email address.	Below the "Sign In" section of the general display, there should be two fillable boxes, one above the other, to aid in the sign-in process. The first box will have text above it stating, "Last Name" and then the box below it will be fillable, where the person can click then type in their last name. The second box will be immediately below this box. It will have text above it stating "Email Address" then the box below it will be fillable, where the person can click then type in their email address.	13	8
1.3	Below where the credentials are entered there must be	Below the sign in credentials button, where the person enters their last name and email address, there should be four radio buttons placed next to each other. One radio button should say "User Sign In", one radio button should say, "Developer Sign	13	8

Requirements Document

	four buttons to determine the type of logging in the person is completing.	In”, another radio button should say “User Sign Up”, and the last radio button should say “Developer Sign Up”. All of these buttons are clickable with the mouse and become highlighted when hovered over.		
1.4	After a user enters there credentials and tries to log in, the credentials must be checked to see if they match an existing account then act accordingly based on that.	Users will type their credentials (last name and email address) into the respective text boxes and then click the “User Sign In” button. If the last name and email are invalid an error message will appear in a modal window in the middle of the screen saying, “Error: Incorrect Credentials”. The credentials are considered invalid if the combination of last name and email does not already exist as a user account in the database. If both the last name and email address are valid and match, the user will be redirected to the user main menu. If the last name and/or email address text boxes are left blank then a different modal window will appear in the middle of the screen stating, “Please fill out both fields of last name/email”.	13	13
1.5	After a developer enters there credentials and tries to log in, the credentials must be checked to see if they match an existing account then act accordingly based on that.	Developers will type their credentials (last name and email address) into the respective text boxes and then click the “Developer Sign In” button. If the last name and email are invalid an error message will appear in a modal window on the screen saying, “Error: Incorrect Credentials”. The credentials are considered invalid if the combination of last name and email does not already exist as a developer account in the database. If both the last name and email address are valid and match, the developer will be redirected to the developer main menu. If the last name and/or email address text boxes are left blank then a different modal window will appear in the middle of the screen stating, “Please fill out both fields of last name/email”.	13	13
1.6	After a developer signs up an account must be created with their entered credentials if they are valid.	Developers will type their credentials (last name and email address) into the respective text boxes and then click the “Developer Sign Up” button. If the last name and/or email address text boxes are left blank then a modal window will appear in the middle of the screen stating, “Please fill out both fields of last name/email”. Otherwise, if both fields have been correctly completed, a developer account is created and saved in the database with the specified credentials. The developer is then redirected to the developer main menu in less than one second.	13	8

Requirements Document

1.7	After a user signs up an account must be created with their entered credentials if they are valid.	Users will type their credentials (last name and email address) into the respective text boxes and then click the “User Sign Up” button. If the last name and/or email address text boxes are left blank then a modal window will appear in the middle of the screen stating, “Please fill out both fields of last name/email”. Otherwise, if both fields have been correctly completed, a user account is created and saved in the database with the specified credentials. The user is then redirected to the user main menu in less than one second.	13	8
-----	--	---	----	---

3.2 Developer Main Menu

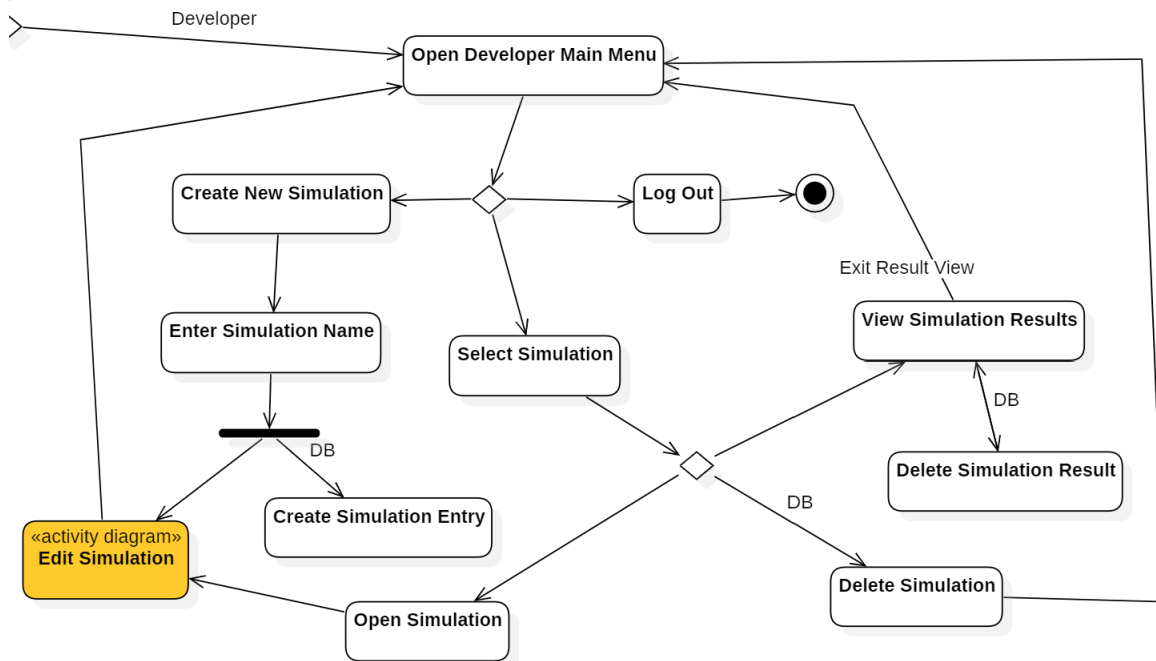
Total Business Value: 113

Total Difficulty: 37

User Description:

Once a person logs into a developer account from their main menu they can either start a new simulation, open an existing simulation they’ve previously worked on, delete a simulation, see the user results of their simulations or logout of their developer account.

Activity Diagram:



System Description:

Req#	Requirement	Requirement Description	BV	Difficulty
2.1	The main menu page	After logging in, the developer will be brought to the developer main menu. On this page in the top left will be the	34	3

Requirements Document

	for the developer must have the application's title and list of simulations associated with the developer. It must also give the options to create a new simulation and log out.	words "Case Based Simulation" in title format. Then at the top right of this page are two buttons. One is labeled "New" and the other is labeled "Log Out". These buttons are side by side. In the main body of the page will be a list of the developer's simulation (Req 3.X) and how to open (Req 2.3), delete (Req 2.4, 4.X) and view results (Req 2.5, 5.X) but that will be explained in later requirements.		
2.2	When the new button is hovered over it must show that it is interactable then once it is clicked a modal window must appear to name the simulation.	Developer will hover over the "New" icon where the icon will be highlighted. The users will then click the icon and a modal pop up will appear within one second. The modal pop up will have text stating "Enter a name for the simulation" at the top of the modal. Then below this is a text box that is fillable upon click.	21	3
2.3	When the open button is hovered over next to each simulation it must show that it is interactable then once it is clicked the simulation is opened.	In the body of the developer main menu, next to each simulation in the simulation list there should be a button labeled "Open". The button should be to the right of the simulation name it corresponds to. Developers will hover over the "Open" icon where the icon will be highlighted by changing color. The developer will then click the icon and be redirected to the create sim menu for the selected simulation.	21	5
2.4	When the delete button is hovered over next to each simulation it must show that it is interactable	In the body of the developer main menu, next to each simulation in the simulation list should be a button labeled "Delete". The button should be to the right of the "Open" button next to the simulation names. Developers will hover over the "Delete" icon where the icon will be highlighted by changing color. The users will then click the icon and a modal window will pop up (the contents of the modal window are described in Req 4.X).	8	8

Requirements Document

	then once it is clicked a modal window must appear.			
2.5	When the results button is hovered over next to each simulation it must show that it is interactable then once it is clicked a modal window will appear.	In the body of the developer main menu, next to each simulation in the simulation list should be a button labeled “Results”. The button should be to the right of the “Delete” button next to the simulation names. Developers will hover over the “Results” icon where the icon will be highlighted by changing color. The users will then click the icon and a modal window will pop up (the contents of the modal window are described in Req 3.X).	13	5
2.6	The modal window for new simulations must have a button to create the simulation and one to cancel which appear next to each other.	In the bottom right corner of the modal pop-up generated from clicking new will be two radio buttons. The buttons will appear next to each other and both be clickable. The left button will have text on it saying “Create” while the button on the right will have the text “Cancel” on it. When the create button is clicked the developer is redirected to the create sim menu. When the cancel button is clicked the modal window is closed. Both of these should happen in less than one second.	8	8
2.7	When the log out button is hovered over it must show that it is interactable then once it is clicked the developer is logged out and returned to the landing page of the application.	Developers will hover over the “Log Out” icon where the icon will be highlighted by changing color. The developers will then click the icon and be logged out of their current account. They will then be automatically returned to the main landing page of the application where they can enter credentials to log in or sign up. This will be done in less than one second.	8	5

Requirements Document

3.3 Developer Available Simulations

Total Business Value: 34

Total Difficulty: 16

User Description:

The developer will be able to access both their in-progress and published simulations through the main body of the developer's main menu page. The simulations will be separated into their respective sections in a vertical list with associated actions for each. From these lists of in-progress and published simulations, the developer can select the one they wish to open, and they are brought to the editing page for that simulation. They can also select a simulation to be deleted or a simulation to review the user results of.

System Description:

Req#	Requirement	Requirement Description	BV	Difficulty
3.1	The “In Progress” list of simulations must be a vertical list that is alphabetized.	There will be a section in the simulation list of the developer main menu with the header “In Progress Simulations”. In the body of the section, there is a vertical list of all simulations the developer has created that they have not published. This list will be displayed in alphabetical order.	8	8
3.2	The “In Progress Simulations” must have three interactable buttons next to each simulation for opening, deleting, and viewing the results.	For each simulation in the list of “In Progress Simulations” they will have three corresponding buttons next to the name of the simulation. Those buttons will be placed next to each other with the following labels from left to right: “Open”, “Delete”, and “Results”. These buttons will be highlighted by changing color when the cursor hovers over them. The functionality when the buttons are clicked is described in Req 2.3-2.5.	8	2
3.3	The list of simulations must be sectioned off in the developer main menu with two headers for “In Progress Simulations” and “Published Simulations”.	The simulation list in the body of the developer main menu will be sectioned off from the rest of the page with a thin black square encasing the entire list. The list should have two sections, “In Progress Simulations” and “Published Simulations” with contents as described in Req 3.1, 3.2, 3.4, 3.5.	5	2

Requirements Document

3.4	The “Published” list of simulations must be a vertical list that is alphabetized.	There will be a section in the simulations list of the developer with the header “Published Simulations”. In the body of the section, there is a vertical list of all simulations the developer has published. This list will be displayed in alphabetical order.	5	2
3.5	The “Published Simulations” must have three interactable buttons next to each simulation for opening, deleting, and viewing the results.	For each simulation in the list of “Published Simulations” they will have three corresponding buttons next to the name of the simulation. Those buttons will be placed next to each other with the following labels from left to right: “Open”, “Delete”, and “Results”. These buttons will be highlighted by changing color when the cursor hovers over them. The functionality when the buttons are clicked is described in Req 2.3-2.5.	8	2

3.4 Developer Delete Simulations

Total Business Value: 15

Total Difficulty: 8

User Description:

The developer can delete simulations by clicking the “Delete” button next to the desired simulation in the developer's main menu simulation list. The developer will then confirm this is the proper deletion and the simulation will be removed from the list of available simulations for the developer, and also the user if it is a published simulation.

System Description:

Req#	Requirement	Requirement Description	BV	Difficulty
4.1	After clicking to delete a simulation a modal must appear asking if the developer wants to delete the selected simulation which then has two buttons to	Once the simulation the developer wants to delete is clicked on, a modal pop up opens in less than a second that asks if you are sure you want to delete the selected file. It does this with the prompt “Are you sure you want to delete: SimName” at the top of the modal window. Then below this text are two buttons that are centered, one that says “Yes” and the other says “Cancel”.	5	2

Requirements Document

	confirm or cancel their decision.			
4.2	The “Yes” button must be interactable and when it is clicked it must delete the simulation from the database.	The “Yes” button is clickable and is surrounded by a box with the text inside. When the cursor hovers over the option it becomes highlighted by changing color. If the “Yes” button is selected, then the simulation is deleted from the database. The developer is then sent back to their main menu page with the list of simulations, however the one deleted is now gone. This process of deletion occurs within one second.	5	5
4.3	The “Cancel” button must be interactable and when it is clicked it must close the modal window without any changes.	The “Cancel” button is clickable and is surrounded by a box with the text inside. When the cursor hovers over this option it becomes highlighted by changing color. If the “Cancel” button is selected, then nothing happens to the simulation. The developer is then sent back to their main menu page with the list of simulations including the one they priorly selected. This return to the developer’s main menu should occur within one second.	5	1

3.5 Developer Simulation Results

Total Business Value: 28

Total Difficulty: 41

User Description:

The developer can view the user results of their simulation by clicking the “Results” button next to the interested simulation in the simulation list of the developer’s main menu. This will open a new modal pop up which provides a list of all the users that have completed the selected simulation and their respective score(s), time to complete, and date of completion.

System Description:

Req#	Requirement	Requirement Description	BV	Difficulty
5.1	Once the “Results” button is clicked there must be a modal window that has the simulation	When the “Results” button is clicked for a simulation in the developer main menu a modal pop up window appears within one second. At the top of this window will be the title “SimName Results” where SimName is replaced by the simulation’s actual name. There is also an exit button in the top right corner of the modal display. It is represented by an X and when it is clicked the developer will return to the developer’s main menu.	5	13

Requirements Document

	name and a button to exit the window.			
5.2	The results modal window must have headers for last name, score, time and date ordered from left to right next to each other.	Below the title and exit button of the simulation results modal, there will be headers in a horizontal list that will help divide the information of the results. The headers will be “Last Name”, “Score”, “Time”, and “Date” in that order from left to right. These headers should be spaced out enough to reasonably contain the associated field of the user’s simulation attempt.	5	2
5.3	The results modal window must have a vertical list of user scores with their associated information in alphabetical order by last name.	Beneath the headers of the simulation results modal window is the vertical list of user scores. Each score attempt will have the last name of the user, their score, the time it took them to complete the simulation, and the date they completed the simulation. These fields will be on the same line but separated so that they appear under their respective headers. This list of scores from the users will be in alphabetical order by last name. If no user has completed this simulation, then this section of the modal would be left blank as there are no scores to display.	13	13
5.4	There must be a button next to each simulation score that allows the developer to delete the score.	When a developer is viewing simulation scores in the simulation results modal, a remove button is present to the right of each score next to the “Date” column of the user’s attempt. This button should be highlighted by changing color when hovered on. When it is clicked the score is removed from the database and is removed from this window’s display. The removal process should take less than one second.	5	13

3.6 User Main Menu

Total Business Value: 95

Total Difficulty: 90

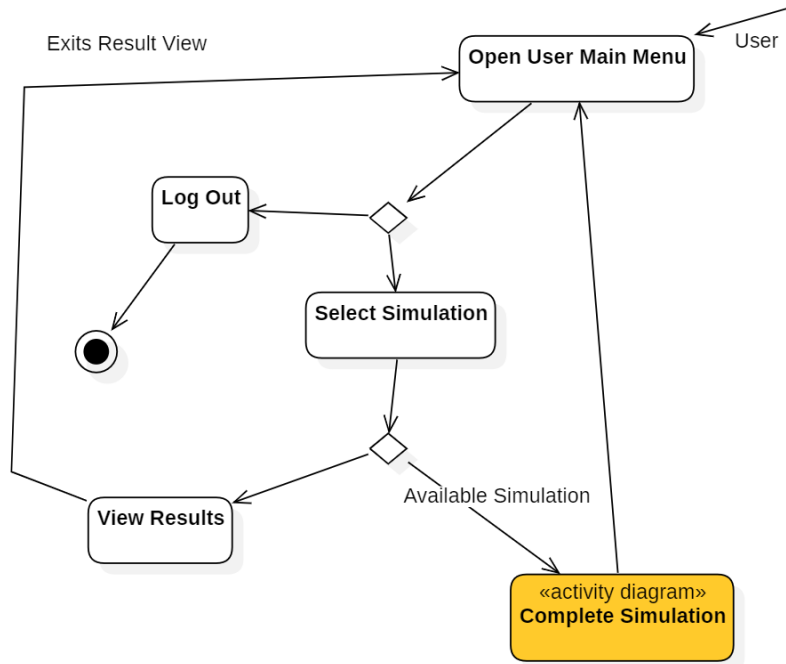
User Description:

Once a person logs into a user account from their main menu they can either select a new simulation to take, view the results of their completed simulations, or logout of their user account. When the user clicks to complete a simulation, they are brought to that simulation player. When the user views their results of a

Requirements Document

simulation, they are displayed their score, explanation of their score, time to complete the simulation, date of completion, and the path of the simulation tree they took.

Activity Diagram:



System Description:

Req#	Requirement	Requirement Description	BV	Difficulty
6.1	The user main menu must have a title specifying the page, headers for available and unavailable simulations, and a log out button.	After logging in, the user will be brought to a new page that has the title “User Main Menu” at the top of the page. In the body of the page there will be a header for “Available Simulations” and a header for “Unavailable Simulations”. These headers will appear one above the other with a list of simulations below them as described in Req 6.3. There is also a button labeled “Log Out” (Req 6.2) in the top right of the page even with the title of the page.	21	5
6.2	The user main menu must have a log out button that is interactable and logs out the user, returning	There is a button even with the title of the page in the top right of the user main menu labeled “Log Out”. When the user hovers over this button it highlights by changing color. When the user then clicks the button it should log them out of their account and return them to the main landing page of the web application that contains the user or developer login options.	5	2

Requirements Document

	them to the landing page of the application.			
6.3	Below both the available and unavailable headers there must be a vertical list of the simulations corresponding to that section in alphabetical order by name. Each simulation also needs to display the name and best score of the simulation.	In the body of the user main menu with the list of simulations there are two sections, one for available simulations and the other for unavailable simulations. Below the “Available Simulations” header there will be a vertical list of all the simulations the developer has created that he has published for the user to attempt. Below the “Unavailable Simulations” header there will be a vertical list of all the simulations that are past the deadline or the number of attempts for taking the simulation have been reached. For each simulation in both lists, the name of the simulation and the user’s best score out of the total score will be displayed on the same line (just a dash if the simulation is yet to be completed). The lists will also be in alphabetical order by simulation name.	8	5
6.4	Each simulation in the available and unavailable simulations must have an interactable button to view the user results of that simulation.	The list of available and unavailable simulations will have a corresponding button next to the name and score of the simulation that is labeled “View”. When the cursor hovers over the button it gets highlighted by changing colors. Once this button is clicked on by the cursor it shows either the available simulation display or the unavailable simulation display based on which header the simulation is under. The contents of these displays are described in Req 6.6-6.9.	5	13
6.5	There may be another section in the list of simulations that specifies the deadline for when a simulation must be completed by.	Stretch Goal: In the list of simulations for the user, there is another section of the simulation entries along with the name of the simulation and the user’s best score. This extra section would be the date of the deadline to complete the simulation by. It will be in the format “Deadline: MM/DD/YYYY” and appear to the right of the user’s best score and left of the “View” button.	1	8

Requirements Document

6.6	When the user views an available simulation the screen must be split in half vertically with the right half having the end node view if the simulation has been completed or blank otherwise. There must also be a button to start the simulation.	After the user clicks the “View” button of a simulation in the “Available Simulations” list, the user will be shown a divided screen, split in half vertically. The right side of the screen will show the end node view of the simulation if the user has already taken it. The contents of this are described in Req 12.4 but the main parts are the simulation name, the score, how long it took to complete, the date of completion, and a list of the incorrect choices with explanations of why they are incorrect. There is also a button at the top right corner of this display to start the simulation as described and implemented in Req 6.10. If the simulation has yet to be completed then the display is blank beside the button to start the simulation. This entire display process is completed within one second.	13	13
6.7	When the user views an available simulation, on the left side of the split screen there must be a tree display with a highlighted path the user traversed or blank if the user has not completed the simulation.	After the user clicks the “View” button of a simulation in the “Available Simulations” list, the user will be shown a divided screen, split in half vertically. The left side of the screen will utilize the “TreeDisplay” as seen in the create simulation menu of the developer. If the user has completed the simulation, then this tree will show the path that the user took to get their best score by highlighting the path in yellow on the tree. If the user has not completed the simulation at least once, then the tree is displayed with no highlighting.	8	13
6.8	When the user views an unavailable simulation the screen must be split in half vertically with the right half having the end node view except it displays the best decisions of the	After the user clicks the “View” button of a simulation in the “Unavailable Simulations” list, the user will be shown a divided screen, split in half vertically. The right side of the screen will show the end node view of the simulation if the user has already taken it. The contents of this are described in Req 12.4 but the main parts are the simulation name, the score, how long it took to complete, and the date of completion. However, instead of showing the incorrect choices and why they are wrong, the correct choices of the simulation are displayed with the reason why they are correct. This will be shown in the same format as the end node view but replacing	13	13

Requirements Document

	simulation with explanations for them instead of explanations for the users incorrect decisions.	the incorrect decisions and explanations with the correct ones. This entire display process is completed within one second.		
6.9	When the user views an unavailable simulation, on the left side of the split screen there must be a tree display with a highlighted path the user traversed and a highlighted path of the best choices for the simulation.	After the user clicks the “View” button of a simulation in the “Unavailable Simulations” list, the user will be shown a divided screen, split in half vertically. The left side of the screen will utilize the “TreeDisplay” as seen in the create simulation menu of the developer. It will then utilize the “Optimal Path” to highlight in blue the best path the user could have taken. Also highlighted, but in yellow, is the pathway that the user took in their attempt at the simulation. Any overlapping paths will be in blue. This entire display process is completed within one second.	8	13
6.10	In the available simulation view there must be an interactable start button at the top right of the window to begin a new attempt of the simulation.	At the top right corner of the available simulation display right there should be a button labeled “Start Simulation”. When this button is hovered over with a mouse it is highlighted by changing color. Once this button is clicked the user is then redirected to the user sim player in less than one second so they can step through the simulation and make a new attempt.	13	5

3.7 Developer Node Creation

Total Business Value: 564

Total Difficulty: 324

User Description:

When in the create simulation menu the developer can create information gathering nodes, decision making nodes, and an end node. The scenario node will be prepopulated, but the developer can still edit the

Requirements Document

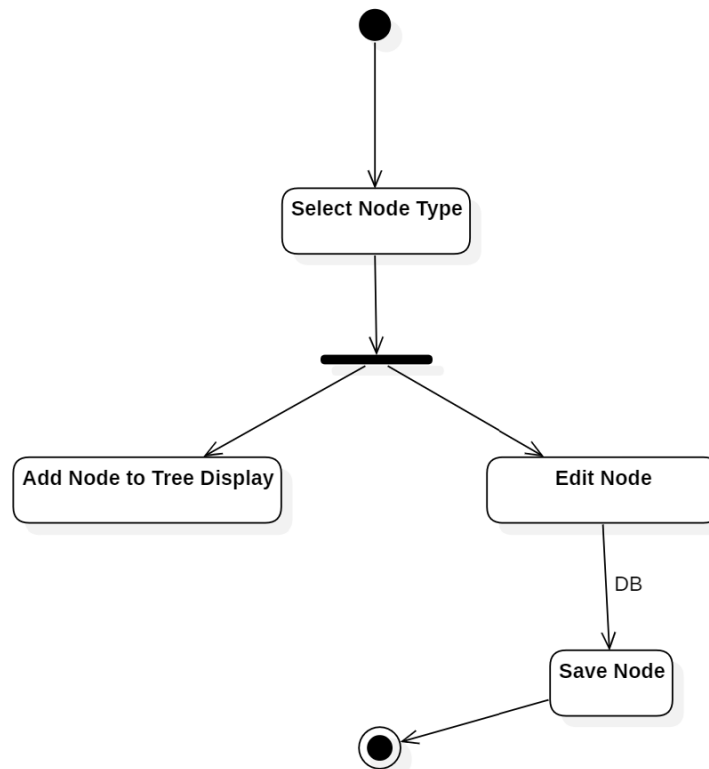
information associated with this node. When creating or editing nodes the developer will input information that is related to each node:

- For a scenario node the developer will input story text and an optional media upload
- For an IG node the developer will input story text, an optional media upload, information gathering options, and lastly context, score, and an explanation of the score for each of these options.
- For a DM node the developer will input story text, an optional media upload, decision options, and a score and explanation of the score for each of these options.
- For an end node the developer will input story text and an optional media upload.

When nodes are created, they are generated on the tree display of the simulation so that the developer can see the structure of the simulation.

Activity Diagram:

Create Node Activity Diagram



System Description:

Req#	Requirement	Requirement Description	BV	Difficulty
7.1	When a new simulation is created the scenario modal must display with the title of the	Once a new simulation is created from the developer main menu, the developer is brought to the sim main menu where a modal window automatically displays on the screen. This modal window will have a title at the top middle stating "Scenario Node". This modal will appear in less than one second upon creation of the simulation.	34	13

Requirements Document

	node at the top.			
7.2	When a developer clicks to create an IG node a modal must appear displaying the title of the node.	In the simulation main menu the developer will click the create IG Node button in the toolbar near the top of the screen. Once this button is clicked a modal window pop-up will display in less than one second. This modal window will have a title at the top middle stating "Information Gathering Node".	34	13
7.3	When a developer clicks to create a DM node a modal must appear displaying the title of the node.	In the simulation main menu the developer will click the create DM Node button in the toolbar near the top of the screen. Once this button is clicked a modal window pop-up will display in less than one second. This modal window will have a title at the top middle stating "Decision Making Node".	34	13
7.4	When a developer clicks to create an end node a modal must appear displaying the title of the node.	In the simulation main menu the developer will click the create End Node button in the toolbar near the top of the screen. Once this button is clicked a modal window pop-up will display in less than one second. This modal window will have a title at the top middle stating "End Node".	34	8
7.5	The scenario node modal must have labeled areas to insert files and a text description.	The modal pop-up window of the scenario node should include a file input button, a header and a text box to input information about the scenario. This will be done with a clickable button at the top of the display stating "Upload File". Below this is the header "Description" that has a large accompanying box where the developer can insert text.	21	13
7.6	The scenario node modal must have clickable save and cancel buttons to either save the inputted data or close	At the bottom of the scenario modal page should be a clickable "Save" button and a clickable "Cancel" button. The save button will be to the left of the cancel button at the bottom of the window. When the save button is clicked the process of saving the node in the database is initiated and the modal window is closed. When the cancel button is clicked the modal window is closed and none of the information inserted into the fields is saved. The process of saving and closing the window should take less than one second.	21	8

Requirements Document

	the window without saving.			
7.7	When a scenario node is created a blue square must appear on the tree display that when clicked opens a modal with the node information.	Once a new simulation is created a blue square should be displayed at the top and middle of the tree display. This square will be clickable by the developer and when it is clicked the modal pop-up window for that scenario will appear in less than one second. If the scenario node has been saved with filled fields those areas will be auto-populated with the last save. If any of the fields have not been saved then they just appear as empty. This should appear in less than one second.	21	21
7.8	The inputted information into a scenario node modal must be saved as a JSON and stored in the database when the save button is clicked.	When the developer clicks save in the scenario node's modal pop up window the information the developer inputted into the fields are saved in a JSON object and stored in the database so that it relates to the current simulation.	34	21
7.9	The IG node modal must have clickable save and cancel buttons to either save the inputted data or close the window without saving.	At the bottom of the information gathering node modal page should be a clickable save button and a clickable cancel button. The save button will be to the left of the cancel button at the bottom of the window. When the save button is clicked the process of saving the node in the database is initiated and the modal window is closed. When the cancel button is clicked the modal window is closed and none of the information inserted into the fields are saved. The process of saving and closing the window should take less than one second	21	8
7.10	When an IG node is created an orange rhombus must appear on the tree display that when	Once a new information gathering node is created an orange rhombus should be displayed on the tree display connected to the associated node it was created from. This rhombus will be clickable by the developer and when it is clicked the modal pop up window for that information gathering node will appear. If the information gathering node has been saved with filled fields those areas will be auto-populated with the last	21	21

Requirements Document

	clicked opens a modal with the node information.	save. If any of the fields have not been saved then it just appears as empty. This should appear in less than one second.		
7.11	The inputted information into an IG node modal must be saved as a JSON and stored in the database when the save button is clicked.	When the developer clicks save in the IG node's modal pop up window the information the developer inputted into the fields is saved in a JSON object and stored in the database so that it relates to the current simulation.	34	21
7.12	The DM node modal must have clickable save and cancel buttons to either save the inputted data or close the window without saving.	At the bottom of the decision making modal page should be a clickable save button and a clickable cancel button. The save button will be to the left of the cancel button at the bottom of the window. When the save button is clicked the process of saving the node in the database is initiated and the modal window is closed. When the cancel button is clicked the modal window is closed and none of the information inserted into the fields are saved. The process of saving and closing the window should take less than one second.	21	8
7.13	When a DM node is created a green pentagon must appear on the tree display that when clicked opens a modal with the node information.	Once a new decision making node is created a green pentagon should be displayed on the tree display connected to the associated node it was created from. This pentagon will be clickable by the developer and when it is clicked the modal pop up window for that DM node will appear. If the DM node has been saved with filled fields those areas will be auto-populated with the last save. If any of the fields have not been saved then it just appears as empty. This should appear in less than one second.	21	21
7.14	The inputted information into a DM node modal must be saved as a JSON and stored in	When the developer clicks save in the DM node's modal pop up window the information the developer inputted into the fields are saved in a JSON object and stored in the database so that it relates to the current simulation.	34	21

Requirements Document

	the database when the save button is clicked.			
7.15	The end node modal must have clickable save and cancel buttons to either save the inputted data or close the window without saving.	At the bottom of the end node modal page should be a clickable save button and a clickable cancel button. The save button will be to the left of the cancel button at the bottom of the window. When the save button is clicked the process of saving the node in the database is initiated and the modal window is closed. When the cancel button is clicked the modal window is closed and none of the information inserted into the fields are saved. The process of saving and closing the window should take less than one second.	21	8
7.16	When an end node is created a grey square must appear on the tree display that when clicked opens a modal with the node information.	Once a new end node is created a grey square should be displayed on the tree display connected to the associated node it was created from. This square will be clickable by the developer and when it is clicked the modal pop up window for that end node will appear. If the end node has been saved with filled fields those areas will be auto-populated with the last save. If any of the fields have not been saved then it just appears as empty. This should appear in less than one second.	21	21
7.17	The inputted information into an end node modal must be saved as a JSON and stored in the database when the save button is clicked.	When the developer clicks save in the end node's modal pop up window the information the developer inputted into the fields are saved in a JSON object and stored in the database so that it relates to the current simulation.	34	21
7.18	The IG node modal must have labeled areas to insert files, a text description, and a list of choices with their	The modal pop up window of the IG node should include a file input button, a description area, and a list of choices with their respective scores and explanations. This will be done with a clickable button at the top of the display stating "Upload File". Below this is the header "Description" that has a large accompanying box where the developer can insert text. Finally below that is a header stating "Choices". Within this section is a list of 8 information gathering options. These options will include 4 boxes per listing. Each listing will have the "Option	21	13

Requirements Document

	respective scores, information, and explanations.	#” as the first text fillable box all the way to the left, then the “Provided Information” text fillable box to the right, then the “Score” text fillable box next to that, and lastly the “Explanation” box all the way to the right. It should look similar to the picture provided below but with the extra input box. All of these boxes can be clicked on by the developer where they can then type the related information.		
7.19	The DM node modal must have labeled areas to insert files, a text description, and a list of choices with their respective scores and explanations.	The modal pop up window of the DM node should include a file input button, a description area, and a list of choices with their respective scores and explanations. This will be done with a clickable button at the top of the display stating “Upload File”. Below this is the header “Description” that has a large accompanying box where the developer can insert text. Finally below that is a header stating “Choices”. Within this section is a list of 4 decision making options. These options will include 3 boxes per listing. Each listing will have the “Option #” as the first text fillable box all the way to the left, then the “Score” text fillable box next to that, and lastly the “Explanation” box all the way to the right. All of these boxes can be clicked on by the developer where they can then type the related information.	21	13
7.20	The end node modal must have labeled areas to insert files and a text description.	The modal pop up window of the end node should include a file input button, a header and a text box to input information about the scenario. This will be done with a clickable button at the top of the display stating “Upload File”. Below this is the header “Description” that has a large accompanying box where the developer can insert text.	21	13
7.21	There must be a general node class that has fields for child nodes, description, media type, and media and is inherited by all the specific node classes.	There should be a general node class which stores information relating to the child nodes, a description for the node, the media type and the media contained within the node. These data fields should have respective getters and setters for each. They should also be able to be set when the node is created. This class will then be inherited by the specific node classes which are scenario, IG, DM, and end.	8	5
7.22	There must be an IG node class that inherits from the general	There should be an information gathering node class which stores information unique to the IG node. This would include the choices of the node, their respective scores, the score explanations, the response of the choice, and all the information from the general node class as that should be	8	5

Requirements Document

	node class and has fields for choices, scores, score explanations, and responses with getters and setters for each of these fields.	inherited from for this class. All these data fields should also have respective getters and setters but they can also be set through the creation of the class.		
7.23	There must be a DM node class that inherits from the general node class and has fields for choices, scores, and score explanations with getters and setters for each of these fields.	There should be a decision making node class which stores information unique to the DM node. This would include the choices of the node, their respective scores, the score explanations, and all the information from the general node class as that should be inherited from for this class. All these data fields should also have respective getters and setters but they can also be set through the creation of the class.	8	5
7.24	There must be an end node class that inherits from the general node class and sets the values in this class as it relates to an end node.	There should be an end node class which stores information unique to the end node. This is done by inheriting all the information from the general node class and setting those values as it relates to the end. There are no new data fields for this node from the general node and the children and media fields should be set to null. All the data fields should also have respective getters and setters but they can also be set through the creation of the class.	8	5
7.25	There must be a scenario node class that inherits from the general node class and sets the values in this class as it relates to a scenario node.	There should be a scenario node class which stores information unique to the scenario node. This is done by inheriting all the information from the general node class and setting those values as it relates to the scenario. There are no new data fields for this node from the general node. All the data fields should also have respective getters and setters but they can also be set through the creation of the class.	8	5

Requirements Document

3.8 Create Simulation Menu

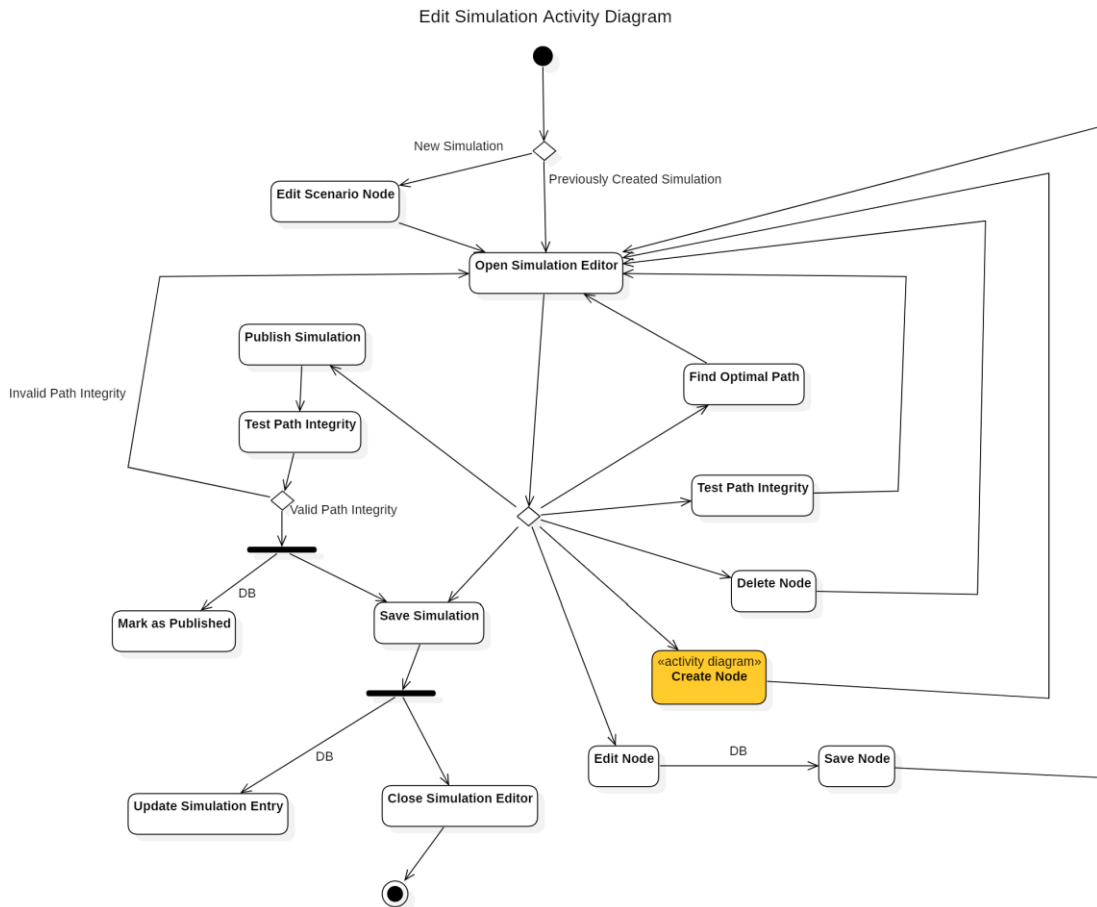
Total Business Value: 261

Total Difficulty: 195

User Description:

The developer can edit their simulation and gather information about it within the create simulation menu. The options the developer has in this menu are create/edit/delete scenario, IG, DM and end nodes, check the optimal path of the simulation, check that the simulation is complete, test the simulation as a user, save the simulation and publish the simulation. Checking the optimal path shows the developer the best path, in terms of highest score, within the simulation. Checking simulation path ensures that all nodes are connected and eventually lead from the scenario node to the end node. Publishing the simulation allows users to complete the simulation.

Activity Diagram:



System Description:

Req#	Requirement	Requirement Description	BV	Difficulty
8.1	The create sim menu must have interactable	Upon entering the “create sim,” developers will be prompted with options: Delete Node, Test Sim, Check Sim Path, Check Optimal Path, Scores, Finalize/Package at the top of the page as buttons. These buttons should be placed next to each other	34	8

Requirements Document

	options to delete nodes, test the simulation, check simulation integrity, check optimal path, view score, and finalize the simulation.	in a row. As a user hovers over an option, the label will be highlighted by changing color and be clickable.		
8.2	The main section of the create sim menu must be a tree display showing the connection between all the nodes in the simulation.	In the “CreateSimMenu”, the tree of all the nodes will be displayed on the center of the screen below the buttons to delete nodes, test the sim, and so on. The Tree should show all nodes with their names and how each node is connected with clear lines.	34	13
8.3	The tree display in the create sim menu must be interactable so that a developer can click on a node and drag it to wherever they want on the display.	Developers will be able to drag the nodes around freely with their connected branches following them wherever they go. This is done by clicking and holding the node then moving the cursor to the desired location and releasing.	21	13
8.4	Nodes on the tree display must have a plus and negative sign attached to them that adds or removes a branch from the node respectively.	Nodes will have a plus sign or negative sign beside them. The plus and negative signs will be clickable. The plus will add a branch directly connected from the node, while the negative will remove an unconnected branch. If there are no unconnected branches to be removed, an error message will appear saying “There are no applicable branches to be removed”.	21	13

Requirements Document

8.5	The TestSim button must be interactable and when clicked it redirects the developer to the start of the simulation.	Developers will hover over the “TestSim” icon where the icon will be highlighted by changing color. The developers will then click the icon and be redirected to the current SimPlayerStart.	8	8
8.6	When a node is selected the delete node icon must be interactable.	The “Delete Node” icon will be “faded out” when a node is not selected by making the text for the button gray and not highlighting when the developer hovers over it. After selecting a node, the “Delete Node” icon will be more visible with black text and the highlighting feature when hovered over.	8	8
8.7	When the developer deletes a node the node must be removed from the tree display and removed from the database.	After selecting a node and clicking the DeleteNode icon, the selected node will be removed from the tree display and removed from the database. The child nodes will not be deleted as a result but will be without a parent.	13	21
8.8	The check sim path icon must be interactable and when it is clicked it must start the process of checking the path integrity.	Developers will hover over the “CheckSimPath” icon and the icon will be highlighted by changing colors. Developers will click the “CheckSimPath” icon and the functionality will be called.	8	8
8.9	As the process of checking the path integrity begins a modal window must appear with the status of the check and	After developers click the “CheckSimPath” icon, a modal window will pop-up preventing users from editing with the simulation. The window will give the message “Checking Path Integrity...”. After “Checking”, the window will either give the message “Path Integrity Passed”, if the “Checking” is true, or “Path Integrity Failed”, if the “Checking” is false. A “Close” will appear at the bottom right corner of the window. Developers can click the “Close” icon to close the window. The modal should appear in less than one second.	8	8

Requirements Document

	update as it progresses.			
8.10	Checking the path integrity must perform a search of the entire tree to ensure all branches are connected to an end node eventually.	Code will perform a breadth first checking if each node has a child. If a node other than the “End Node” is found to not have a child, the node will be highlighted red and the “Checking” is set to be false. If the last node is “End Node”, “Checking” is set to true.	21	34
8.11	The check optimal path button must be interactable and start the process of finding the optimal path in the tree.	Developers will hover over the “CheckOptimalPath” icon and the icon will be highlighted by changing colors. Developers will click the “CheckOptimalPath” icon and the functionality will be called.	8	8
8.12	As the process of finding the optimal path begins a modal window must appear with the status of the check and update as it progresses.	After developers click the “CheckOptimalPath” icon, a modal pop up window will appear preventing users from editing the simulation. A message will be given inside the window saying “Checking optimal path...”. Once “CheckOptimal” is complete, the window will close. The modal should appear in less than one second.	8	8
8.13	Checking the optimal path must perform a search of the tree to find the path that results in the highest point value once the end node is reached.	Code will begin at the “Start node.” The code will look at each child node and choose the greatest point option during the DM node and traverse there. The code will continue until it reaches the “End Node.” The path will be highlighted along the way in the tree display to visualize the “OptimalPath.”	13	21

Requirements Document

8.14	Clicking the publish button must open a modal window where the developer then enters a short description of the simulation before finishing it.	<p>Developers who have finalized their simulation to be published and clicked on the “Publish” button are presented with a modal window for project finalization which contains a text field and button labelled “Finish” and appears in less than one second.</p> <p>A text field is headed with the title “Simulation Description.” Developers are able to input a concise text description of the simulation in 100 characters or less. This description will be visible to Users before starting a simulation.</p>	13	3
8.15	There must be a finish button in the publish modal window that completes the publishing process of the simulation by moving the simulation into its appropriate section so users can access it.	When a Developer is finished typing a description for their simulation in the Publish modal window, they can press a Finish button located below the text field, which will publish the simulation by moving it to the published simulations list in the database. This will then give Users the ability to complete the simulation.	21	8
8.16	In the publish modal there may be a deadline section that the developer sets for the users to complete the simulation by.	STRETCH GOAL: Developers who have clicked on Finalize/Package are taken to a modal window. Located between the text field and “Finish” button is another text field with the header: “Deadline (leave blank if none)” with a line beneath it that says “[MONTH]/[DAY]/[YEAR] [HOURS IN MILITARY TIME]:[MINUTES].” The Developer must enter a deadline for the simulation to the specifications of the second line. If the deadline is in the incorrect form, when the Developer clicks “Finish” a red error message will appear beneath this field which states “Incorrect Deadline Formatting.”	1	5
8.17	There must be a save button in the create sim menu for the developer to save the current state of the simulation	Developers who have started creating a simulation but have not yet finalized it or published it are able to click Save as a menu option in the CreateSimMenu. Clicking on this button will save the work currently done on the simulation, so the Developer can continue to work on it at a later session.	21	8

Requirements Document

	they are working on.			
--	----------------------	--	--	--

3.9 Simulation Player

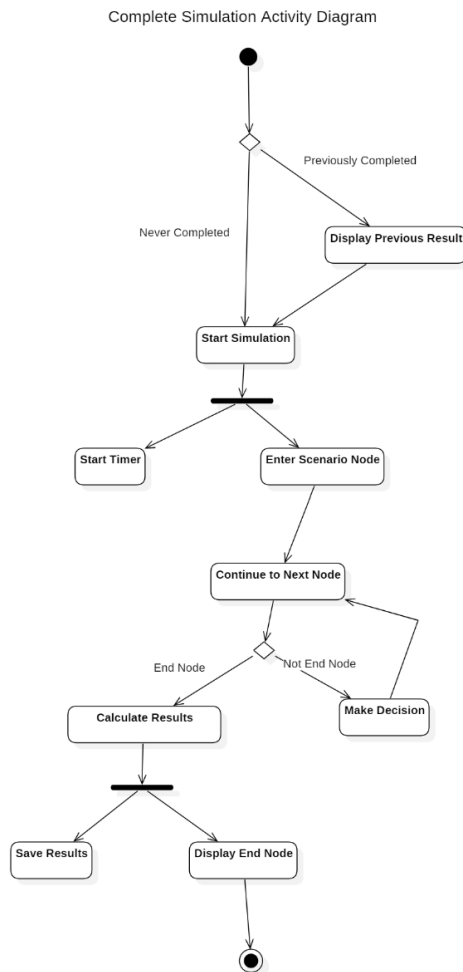
Total Business Value: 79

Total Difficulty: 61

User Description:

When the user selects a simulation to complete, they are brought into an immersive environment where they begin traversing through the simulation based off their decisions. This immersive environment includes a timer to track how long the user takes to complete the simulation, a dropdown menu where they can review past answers, the name of the simulation and a button to go to the next node. A visual tree should also be populated displaying the path the user has taken thus far in the simulation. At the end of the simulation the score of the user is automatically calculated and saved so that both the developer and user can reference it. The developer can also access this simulation player by testing their simulation in the create simulation menu, but when they do this, they will have an additional option in the environment to end the simulation early.

Activity Diagram:



Requirements Document

System Description:

Req#	Requirement	Requirement Description	BV	Difficulty
9.1	On every node in the user view, there must be a drop-down menu that vertically lists the previous nodes so that the user can view previous information gathered in the simulation.	Users should have a drop-down menu at the top right of every node that contains a vertical list of previous nodes they already explored. Users should be able to select a node and be redirected to that node in the simulation. The drop-down menu with also include a “Current” node to be redirected back to the furthest node. When going back to previous nodes the user can view information gathered from them but cannot change any decision that has been made.	8	21
9.2	When the user starts a simulation, they must be redirected to a new page that has the title of the simulation and begins at the scenario node of the simulation.	When the user starts a simulation through the user main menu they are brought to a new page. This new page will have the title of the simulation at the top middle. Then in the upper left corner but below the title will say “Scenario”. Beneath this will appear the “Scenario” description with any uploaded media being presented above the description. At the bottom right corner, a “Next” icon can be clicked to move on to the next node in the simulation.	8	3
9.3	There must be a timer that tracks how long a user takes to complete a simulation which appears in the upper left corner of each page of the simulation.	When a simulation starts by the user, a timer incrementing by one every second starting from zero appears at the top left corner of the page. This timer will continue to each node to show the time it takes the user to complete the entire simulation.	3	3
9.4	The final score of the simulation must be calculated by	As the user traverses through a path within the simulation, all the decision’s scores they make are summed in a variable that will then be presented to the user at the end of the simulation.	34	13

Requirements Document

	adding all the points gathered from the nodes and decisions the user traversed.			
9.5	Once a score is calculated it must be saved in relation to the user and simulation in the database.	Once the simulation has been completed by the user, the score and the decision explanations are stored in database as a JSON with a relation to the current simulation and the user.	13	8
9.6	When a developer tests a simulation there must be an interactable end button throughout the simulation that allows them to end the simulation prematurely.	When a developer tests a simulation they should not only have the same functionality as the user but with an additional icon at the top right of the page named "End". The icon will be clickable and allow developers to end the simulation prematurely. This will redirect developers back to the "CreateSimMenu".	13	13

3.10 Account Storage

Total Business Value: 68

Total Difficulty: 68

User Description:

The developer and user accounts need to be saved so that returning users of the application can access their accounts and the simulations associated with them.

System Description:

Req#	Requirement	Requirement Description	BV	Difficulty
10.1	When a new user signs up the inputted credentials must be saved	When a new user clicks the user sign up button on the login screen of the application, the inputted fields are saved as a new user account in the database. The user can then use the same inputs as their credentials to log in to their account.	21	21

Requirements Document

	as a user in the database.			
10.2	When a new developer signs up the inputted credentials must be saved as a developer in the database.	When a new developer clicks the developer sign up button on the login screen of the application, the inputted fields are saved as a new developer account in the database. The developer can then use the same inputs as their credentials to log in to their account.	21	21
10.3	Newly created accounts through the developer sign up must be stored in the developers' section of the database.	Newly created accounts should be marked as a developer account when using the developer signup button on the home screen. These accounts should be stored separate so they are able to access certain creation features.	13	13
10.4	Newly created accounts through the user sign up must be stored in the users' section of the database.	Newly created accounts should be marked as a user account when using the user signup button on the home screen. These accounts should be stored separate so they restricted from developer features, but have access to simulation functions.	13	13

3.11 File Display and Storage

Total Business Value: 129

Total Difficulty: 81

User Description:

Media files are an integral part of nodes as they can provide further context for each node. Therefore, when nodes are created they must have the functionality to store files and links to media and then display them when the simulation is being taken.

System Description:

Req#	Requirement	Requirement Description	BV	Difficulty
------	-------------	-------------------------	----	------------

Requirements Document

11.1	When a file is uploaded it must be stored in the database with a relation to the node it was added to.	When a file is uploaded to a node through the image, video or audio option it is stored in the backend database with a relation to the node it has been uploaded to.	21	8
11.2	There must be an image upload option at the top of every node, except end nodes, that opens a file explorer and accepts images in jpg or png formats.	Developers will be able to upload an image file to any Scenario, Information Gathering, and Decision Making node in their respective modal windows. There will be a button at the top of the modal for image upload which uses a base image icon as the button. The image button will open a file explorer when clicked where the user can choose the file they wish to upload. The file has to be either in jpg or png format and only one image file can be uploaded.	8	5
11.3	All modals must be maneuverable by grabbing the edge of the window and dragging it to the desired location.	Once any modal pop up window appears that modal must be draggable. This is done by clicking and holding at the top border of the modal window then moving the mouse to the desired location and releasing. The modal is then set in this new location and stays there until it is moved again or closed. This functionality must be added to all node modals.	8	8
11.4	When the user steps through a simulation any uploaded media files must be shown above the descriptions of each node.	Users accessing a simulation will be able to view any files for the simulation available to them that were inputted by Developers. These files will be viewable above the description of every node in the simulation.	21	8
11.5	All uploaded media should be embedded in the nodes	Pictures, audio and videos should be embedded in a node, so when users are in a simulation, they can see the media without a popup window. For images the picture appear automatically	21	8

Requirements Document

	in the simulation so that videos and audio can be played directly and images appear on the screen.	while for videos and audio there is a player so that the user can pause/play the video or audio.		
11.6	All modals must have a vertical scroll bar on the right side of the modal to move the contents of the modal up and down.	When a modal pop-up window is opened it should have a vertical scroll bar on the right hand side of the modal. This scroll bar can be clicked and drag up and down which also moves the contents of the modal up and down in relation to this movement. This allows the full information in the modal to be viewed no matter the size of it. This functionality should be added to all modals.	5	5
11.7	Developers must be able to upload youtube or vimeo links through a button on the node modal windows.	Developers will be able to upload a video link to youtube or vimeo to any Scenario, Information Gathering, and Decision Making node in their respective modal windows. There will be a button at the top of the modal for video link upload which uses a base video icon as the button. The video button will open another modal window which prompts the developer for a link to the video: "Enter link to Youtube or Vimeo video:". There is then a text box below this prompt where the user can enter the link. Lastly, there is a "Cancel" and "Save" button at the bottom of this modal. The cancel button closes out of the upload without saving any information in the text field. The save button saves the link for that specific node. Only one video link can be uploaded per node.	8	5
11.8	Developers must be able to upload audio files in mp3 format through a button on the node modal windows.	Developers will be able to upload an audio file to any Scenario, Information Gathering, and Decision Making node in their respective modal windows. There will be a button at the top of the modal for video upload which uses a base audio icon as the button. The audio button will open a file explorer when clicked where the user can choose the file they wish to upload. The file has to be either in mp3 format and only one audio file can be uploaded.	8	5
11.9	Once an image is uploaded a preview of the selected image must	When a developer uploads an image a preview of that media is shown in the modal window. For an image upload the selected photo is shown below the image icon. This preview should appear within one second of the file being uploaded.	8	5

Requirements Document

	appear in the modal window.			
11.10	Once a video is uploaded a preview of the selected video must appear with a player in the modal window.	When a developer uploads an audio file a preview of that media is shown in the modal window. For an audio upload a player appears below the audio icon which will play the selected audio when clicked. This player will have a play button on the left and a bar with a dot that shows how far into the audio you are. when the play button is clicked the audio starts and if it is clicked again the audio pauses. The dot on the progress bar is draggable so that the developer can skip ahead or go backwards in the audio. The preview should appear within one second of the file being uploaded.	8	8
11.11	Once an audio is uploaded a preview of the selected audio must appear with a player in the modal window.	When a developer uploads a video a preview of that video is shown in the modal window. For a video link upload the selected video is shown below the video icon in an embedded format. This embedded format should have a pause/play button and a draggable progress bar where the developer can control where they are in the video. The preview should appear within one second of the file being uploaded.	8	8
11.12	When an image larger than 250 KB is uploaded to a node its dimensions must be reduced to less than this number.	When an image is uploaded ensure that the image size is less than 250 KB. If it is less than this then the image can be uploaded with no changes. If the image is larger than this, reduce the dimensions of the image to fit the constraint and save that edited image.	5	8

3.12 User Node Views

Total Business Value: 84

Total Difficulty: 84

User Description:

Along with the immersive aspects of the simulation player, the information associated with the nodes needs to be displayed to the user when they are completing the simulation.

- The scenario node should include the story text and any uploaded media.
- The IG node should include the story text, any uploaded media, and the information-gathering options in a checklist format.

Requirements Document

- The DM node should include the story text, any uploaded media, and the decision choice in a multiple-choice format.
- The end node should include the story text, the final score, the decision summary, and an end button to complete the simulation.

System Description:

Req#	Requirement	Requirement Description	BV	Difficulty
12.1	The simulation view of the scenario node must display the uploaded media and the text description of the node in an immersive way.	The scenario node will be viewed by the user once the simulation is opened. This will have the uploaded media at the top of the page beneath the title. The media will appear next to each other for each format that has been uploaded. Beneath the media is then the text description for the scenario which has a header before it stating "Description".	21	21
12.2	The simulation view of the IG node must display the uploaded media, the text description, the choices with respective check boxes, and the choice responses for each selected option in an immersive way.	In the user view of the IG node, in the top left below the simulation title it says "Information Gathering". Below this header is the uploaded media from the developer. These are embedded in the page and appear in a row. Beneath that is a header "Description" with text below it providing the description for the IG node. Then the available information gathering choices appear next with square checkboxes next to each one of them. These boxes become filled once clicked. Once the user selects all the options they want they then click a button beneath all the choices that say "Submit" which reveals the information for each selected choice. There is then a next button in the bottom right of the page to continue to the next node in the simulation.	21	21
12.3	The simulation view of the DM node must display the uploaded media, the text description, of the node, and the choices with	In the user view of the DM node, in the top left below the simulation title, it says "Decision Making". Below this header is the uploaded media from the developer. These are embedded in the page and appear in a row. Beneath that is a header "Description" with text below it providing the description for the DM node. Then the available decision making choices appear next with empty circles next to each one of them. The circle becomes filled once that choice is clicked but only one choice can be clicked. There is then a next button in the bottom right of the page to continue to the	21	21

Requirements Document

	respective buttons in an immersive way.	next node in the simulation once they select the choice they want.		
12.4	The simulation view of the end node must display the description of the node, the score the user received and a decision summary specifying the choices in which the user made a mistake.	In the user view of the End node, in the top left below the simulation title it says "End". Below this in the center of the page is the score the user received out of the maximum possible for the simulation. Below the score is another header that says "Description" with text beneath it providing the final description for the end node and simulation. Then below that is a header labeled "Decision Summary" which lists all the incorrect decisions made by the user and the developer provided explanation to why they aren't the right choice. Lastly there is an exit button in the bottom right corner of the page which returns the user to the user main menu.	21	21

4 Nonfunctional Requirements

4.1 Compatibility

As there will be many different types of people accessing this system, it must be highly compatible to different environments. The users of the system must be able to access the web application from any type of web browser. All displays within the application need to be resizable or scrollable so that they can be viewed on any screen.

4.2 Responsiveness

Since this system is highly interactive and user-based, responsiveness is a key component that needs to be considered in development. All pages, displays, and functions should load within one second of their initialization. If this is not possible due to heavy computation, then a loading screen showing the progress of the function must be displayed. This will ensure the user of the application is still engaged and not tempted to click away from the website.

4.3 Memory Performance

With files being uploaded to the system, there needs to be a way to ensure these files are not too large to overwhelm the database and website. To do this all file uploads need to have their sizes checked and possibly compress them if they are too large. Other types of data should also be stored in JSON structures to ensure small data sizes.

4.4 Documentation

This system is considered to be a prototype or version one of the final application. Due to this, there will be future developers working on the same code and adding new functionality. As a result, the current implementation needs to be well documented so that future developers can easily understand what is happening in the system. This should be done through explanatory comments throughout the code base and detailed documentation of all facets of the system and how they interact.

5 Out of Scope Requirements

As this system is expected to be a prototype and the main functionality prioritized, the following requirements detailed in this section have been discussed between the team and sponsor to be out of scope.

5.1 Account Monetization

The ability to charge for new developer and user accounts to access the application and put them on a payment plan is needed to ensure a return on investment for the sponsor.

5.2 Information Security

When people log in and create accounts in the system there should be security measures in place to protect their account information and related data.

5.3 Server Balancing

As many people access the server at one time it may create a heavy load and slow down everyone's experience with the application. Thus it may be useful to add multiple servers interacting with each other and balancing the users across them to ensure reliable speeds.

6 System Evolution

Since this system is considered to be a prototype there are many future versions of the application planned. The following are possible areas of exploration for these future versions.

6.1 Accounts

Account Monetization

- When developers and/or users create an account they have to set up a billing plan to use the application. This will increase the return on investment for the sponsor as people will have to pay to use the system.

Account Security

- Add password and username protection on accounts so that their related data and information remain safe within the system.

Account Retrieval

- If a user or developer forgets part of their credentials there is a way for them to reset their username or password from their email.

6.2 Simulation Deadlines

Attach deadlines onto simulations so that users only have a certain number of days to complete the simulation.

6.3 Time as Part of Score

Since the time it takes for the user to complete a simulation is already being tracked, it may be helpful to have this effect the final score somehow. There could be weights introduced that value the speed of decision making and increase the user score the faster they complete the simulation.

6.4 Increased Media Inputs

The acceptable media input formats could be increased from the ones already supported. New media types could also be introduced such as 360 images, animations and gifs.

6.5 Size of Tree

Currently, the size of the tree is limited to only 10 levels to reduce complexity. Further variations of the system may increase the size to create deeper simulation trees. The number of IG node options and DM nodes choices could also be increased thus increasing the breadth of the tree.