# REQUIREMENTS DOCUMENT: CASE-BASED SIMULATION [BUILDER]

A Requirements document provides an in-depth and comprehensive understanding of what the product specifications and user requirements are and how the software would accomplish it.

1. PROJECT INFORMATION
   1. Project Name
   2. Date of document submission
   3. Version of document (if any changes)
   4. Client (name of client company)

Also state which team member is responsible for writing which section in this document.

1. PROJECT SCOPE AND DEFINITION

Outline the strategic vision, goals & objectives and ideally include a high-level mission statement. This will help align the team on the approach and keep these goals in mind during solutioning. It will also help in defining additional work and potential project enhancements as the team keep this context in mind.

This section gives a high-level overview of the software application to be built, sets the tone for the project, defines what the long-term objectives and goals of the project are and gives all the team members working on the project absolute clarity.

Key items to be addressed:

1. Define the project. What is the project about?
2. What is the scope of the project?
3. What is the problem to be solved?
4. What specific outcomes will be achieved? How will they be measured?
5. What are the deliverables?
6. What are the milestones for the deliverables?
   1. If possible, identify projected increments by Sprint
7. Are there any constraints that may influence your deliverables and schedule?
8. Describe why the software (or upgrade) is being developed.
9. List the most important features and capabilities.

COMMON DEFINITIONS

1. CLIENT: The owner of the product.
2. DEVELOPER: A person that creates/builds a simulation using the product.
3. USER: A person (a student/trainee) that uses the simulation to be evaluated on their performance of responses and actions within the simulation environment.
4. 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) IG (information gathering), and 4) DM (decision making).
5. SCENARIO: The problem set. What introduces the USER to the environmental situation/case/problem they need to solve.
6. NODE LINK: The directional connection between 2 nodes.

PROJECT DEFINITION AND SCOPE

The Case-Based Simulation is a Web application that allows 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 Developers

* Allow developers to build simulations based on authentic scenarios using a diversity of media such as text, images, audio, and video.
* Allow developers to create simulations through branched pathways or decision trees.
  + The simulation starts with a scenario that describes the problem to be solved.
  + The body of the simulation is designed through a series of links to and from IG (information-gathering) nodes and DM (decision-making) nodes where users can select and review data and make decisions of what to do next respectively.
    - Multiple IG and DM nodes can be linked to provide different “choices” or pathways in the simulation.
  + The simulation ends with an automated tally of user results based on their interactions and choices in the simulation.
* Allow developers to save, edit, and delete simulations.
* Allow developers to view user results.

For Users

* Allow users to access and use developed and archived scenarios.
* Evaluate users on their judgements in selecting appropriate data (information gathering) and acting on that data or environmental situation (decision making).
* Allow users to save their simulation results.

Deliverables

* A fully functional Web application that allows developers to create, edit, save, and delete case-based simulations and allows users to access and use those simulations.
* Future deliverables will provide “user” licensing and subscription for provisioning discrete application accounts.

Case-Based Simulation Decision Tree

<simulation>

<scenario>

<data> </data>

<link> </link>

</scenario>

<ig>

<data> </data>

<link> </link>

</ig>

<dm>

<data> </data>

<links> </links>

</dm>

<end> </end>

</simulation>

Scenario

Node (IG)

Node (DM)

End

1. SYSTEM REQUIREMENTS AND FUNCTIONAL REQUIREMENTS

The functional requirements or the overall description documents include the product perspective and features, operating system and operating environment, graphics requirements, design constraints and user documentation.

The appropriation of requirements and implementation constraints gives the general overview of the project in-regards-to what the areas of strength and deficit are and how to tackle them.

Key items to be addressed:

1. Describe system functionality and features. Describe what the system must do and how it does it.
2. Describe system workflows.

Any Requirement Which Specifies What The System Should Do.

Describes the system.

Diagram

Description automatically generated

|  |  |  |
| --- | --- | --- |
| Functional Requirement | Constraints | System Environment |
| Web Application | -TLS/SSL  -WAF | Secure Web Server |
| XML Data Tree | -Schema Constraints  -Complex Data Tree Structure  -Viable Tree Pathways  -Editing XML Tree Structure  -Mapping XML Tree Structure to Graphic Representation | Secure Web Server or Database |
| Database Type | -Relational  -MySQL | mySQL Server |
| Media (text) | -XML Encapsulated  -External Files | Simulation Application |
| Media (images) | -File Type  -Orientation  -Size  -Scale to App | Simulation Application |
| Media (audio) | -File Type  -Size | Simulation Application |
| Media (video) | -Streaming Server Types | -YouTube  -Vimeo |
| Inputs | -Media  -Node Type  -Links  -Points (selection values) | Simulation Application |
| Outputs | -Simulation (dev)  -XML  -Media  -User Data |  |
| Tree Structure | -Define Optimal Path  -Check “Dead Ends”  -Edit, Delete Nodes and Links | Simulation Application |

The Case-Base Simulation Builder system consists of 4 discrete parts: 1) the Web-based simulation where the user interacts with the simulation; 2) a Web Server where the website, images, and XML files are logically stored; 3) an XML data tree where the simulation data tree structure is stored; and 4) a relational database where user and simulation data is stored.

Starting a Simulation

Diagram

Description automatically generated with low confidence

Building the Simulation

Diagram

Description automatically generated

Add Media

Diagram

Description automatically generated with low confidence

1. SYSTEM ENVIRONMENT

Include a Use-Case diagram here with a brief explanation of all the interactions.

Diagram, engineering drawing

Description automatically generated

1. EXTERNAL INTERFACE REQUIREMENTS

Interface requirements consist of the hardware and the software interfaces along with user and communication interfaces.

1. User interfaces consist of the style guides, screen layout, buttons, functions.
2. The software interfaces consist of the platform, database system, front end and the backend framework, operating systems, tools and libraries.
3. Hardware interfaces includes details of the hardware components like the list of supported devices, nature of data and the hardware-software interactions.
4. Communications interfaces are the network server communications protocols. The requirements determine the communication standards to be utilized.
5. NON-FUNCTIONAL REQUIREMENTS

The non-functional requirements constitute the following:

1. Performance/Capacity requirements. How quickly does the system respond to users’ actions, or how long does a user wait for a specific operation to happen? Examples might include:
   * Response time
   * Transaction performance
2. Capacity. What are your system’s storage requirements, current and future? How will your system scale up for increasing volume demands?
3. Resources such as memory, storage, communications
4. Centralized or distributed processing.
5. Security requirements. Does your product store or transmit sensitive information? Data or drive encryption? Authentication or authorization requirements? Asset security? (laptops, physical equipment) External partner requirements, especially those around data access and data security. Include policies, standards as well as methods for controlling access like Citrix, Firewalls, VPN
6. Compatibility. What are the minimum hardware requirements? What operating systems and their versions must be supported?
7. Environmental. What types of environments will the system be expected to perform within?

A non-functional requirement will describe how a system should

behave and what limits there are on its functionality.

Non-functional requirements affect the user experience as they

define a system’s behavior, features, and general characteristics.

|  |  |  |
| --- | --- | --- |
| Non-Functional Requirement | Limitations | Description |
| Web Security | -TLS/SSL  -WAF | Secure Web Site |
| Web Access | -Internet  -Computer  -Tablet | User Access |
| Interface | -Visual  -Graphical Sim Build Tree Structure  -Inputs (see Inputs) | Displays visual nodes during development. Edit nodes by clicking on visuals in tree structure. |
| Inputs | -Dialog Box  -Node data (IG, DM)  --Points (selection values)  -Node link(s)  -Node Selection | Pop-up dialog box for node input and edits. |
| Graphics | -Size | Size restrictions. >< X px |
| Video | -Length, Quality | Length restrictions < X min  Quality >= 720 <= 1080 |
| Buttons | -Save(As)  -Edit  -Close  -Delete | Delete  -Node(s) [links]  -Simulation |
| Capacity | -Media Storage | What are the limitations for storage? |
| Performance/Speed | -Internet | Responsive load times |

1. DATA MANAGEMENT

Describe the data management requirements for the system, including the primary data sources and repositories. Also describe the data retention, archival, and warehousing.

|  |  |  |
| --- | --- | --- |
| Data Requirement | System Environment | Data |
| Web Application | Secure Web Server | HTML, PHP, CSS, JSON, JS |
| XML Data Tree | Secure Web Server or Database | XML Data Structure |
| Database | MySQL | User Accounts  -User Account Info  -PW, Login  -Sim Data |
| Text | XML Structure | Node Stories (text) |
| Images | -XML Structure Links to Image Repository | -JPG  -PNG |
| Audio | -XML Structure Links to Audio Repository | -MP3 |
| Video | -XML Structure Links to Video Repository | -YouTube  -Vimeo |

1. CONSTRAINTS

Constraints are conditions outside the control of the project that limit the design alternatives. Describe any high-level items that limit the developer's options for designing the software such as:

1. Standards (including hardware and software) Imposed on the Solution
2. Schedule
3. Budget
4. Preferred Software Programming Language(s)
5. Required Development Tools
6. Handling of Security Requirements (if any)
7. Potential Risks
8. Policy and Regulation
9. CHANGE CONTROL

Describe how changes to the project scope are controlled, and who approves these.