Requirements for a scenario-based collaborative learning platform

The system verbs in this document loosely follow the [MoSCoW](https://en.wikipedia.org/wiki/MoSCoW_method) method:

* MUST denotes a requirement, without which the system cannot be used to perform core use cases.
* SHOULD denotes a requirement which is not needed to perform core use cases, but without which the system is strongly limited.
* COULD denotes a requirement which is not necessary for the system to function, but which might bring considerable additional value.
* WILL NOT denotes a requirement which is understood to be relevant for the system, but not currently in scope of implementation.

The behavior verbs in this document describe which actions the system supports:

* **ENFORCE**: This action must be performed by the user. The system validates the action.
* **PROHIBIT**: This action must not be performed by the user.
* **ALLOW**: The user can choose to perform this action or not. The system validates the input.
* **SHOW**: The user must do nothing, there is no input.

**Player Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requirement** | **Priority** | **AIT or MT or both?[[1]](#footnote-1)** |
|  | The platform COULD ALLOW participants to play a game alone. |  |  |
|  | The platform MUST ALLOW participants to collaboratively[[2]](#footnote-2) play a game as part of a group. |  |  |
|  | The platform MUST ALLOW a participant to access a game without having to log in. |  |  |
|  | The platform SHOULD ALLOW participants to describe themselves using keywords.  Keywords can be used to describe:   * The industry of the participant, * The prior experience of the participant, * The position of the participant within their organization. |  |  |
|  | IF the participants have used keywords to describe themselves,  THEN the platform SHOULD SHOW additional information to these participants. |  |  |
|  | The platform MUST ALLOW participants to view the variables of the game that are set to “visible”. |  |  |
|  | The platform MUST ALLOW players to solve injects.  An inject solution may be:   * Selecting one of the transitions of the inject. * Inserting a textual value. |  |  |
|  | The platform MUST SHOW the same sequence of injects to all participants of a game. |  |  |
|  | IF participants choose different transitions when solving one inject,  THEN the system MUST determine one transition which all participants follow. |  |  |
|  | IF a participant was previously disconnected from a game,  THEN the platform MUST ALLOW players to rejoin. |  |  |
|  | The platform MUST PROHIBIT participants from joining closed games. |  |  |
|  | The platform SHOULD SHOW participants the personal history of a previously played game. The personal history of a game consists of:   * All of the solutions for injects which this participant has submitted in the course of this game. * Timestamps for the aforementioned solutions. * Timestamps denoting the start and end time of the game. |  |  |

**Trainer Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requirement** | **Priority** | **AIT or MT or both?** |
|  | The platform MUST ALLOW trainers to create games. |  |  |
|  | The platform SHOULD ALLOW trainers to see how many participants have joined an open game. |  |  |
|  | The platform MUST ALLOW trainers to start games. |  |  |
|  | The platform MUST ALLOW trainers to see the relevant information of a started game:   * Which inject participants are currently working on. * Which variables exist in the game. * What value each of the variables has. * Any other injects and stories. |  |  |
|  | The platform SHOULD ALLOW trainers to change variable values for games in progress. |  |  |
|  | The platform MUST ALLOW trainers to close games. |  |  |
|  | The platform SHOULD SHOW trainers the general history of a game which they have previously closed. The general history of a game consists of:   * A count of the number of players of this game. * All of the solutions for injects which have been submitted in the course of this game. * Timestamps for the aforementioned solutions. * Timestamps denoting the start and end time of the game. |  |  |

**Observer Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requirement** | **Priority** | **AIT or MT or both?** |
|  | The platform SHOULD ALLOW observers to see how many participants have joined an open game. |  |  |
|  | The platform SHOULD ALLOW observers to see the relevant information of the started game:   * Which inject participants are currently working on. * Which variables exist in the game. * What value each of the variables has. * Any other injects and stories. |  |  |
|  | IF an observer has observed a game in the past,  THEN the platform SHOULD ALLOW the observer to view the general history of this games any time in the future. |  |  |

**Scenario Designer Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requirement** | **Priority** | **AIT or MT or both?** |
|  | The platform MUST ALLOW the creation of scenarios. |  |  |
|  | The platform MUST ENFORCE that a scenario has all required meta-information:   * Title, * Description, * Target Group. |  |  |
|  | The platform MUST ENFORCE that a scenario has one or more *stories*. |  |  |
|  | The platform MUST ENFORCE that the stories of a scenario are in sequential order. |  |  |
|  | The platform MUST ENFORCE that a story has the following properties:   * Title. |  |  |
|  | The platform MUST ENFORCE that a story has one or more *injects*. |  |  |
|  | The platform MUST ENFORCE that a story has exactly one inject as an *entry point*. |  |  |
|  | The platform MUST ENFORCE that an inject has the following properties:   * Title, * Textual Description. |  |  |
|  | The platform SHOULD ALLOW an inject to have the following properties:   * Image, * Informational snippets.[[3]](#footnote-3) |  |  |
|  | The platform MUST ALLOW an inject to have one or more transitions. |  |  |
|  | The platform MUST ensure that all injects within a story are connected by transitions. |  |  |
|  | The platform MUST ENFORCE that a transition references a target inject. |  |  |
|  | The platform MUST ALLOW a transition to have the following properties:   * Change Action, * Conditions, * Title. |  |  |
|  | IF an inject has more than one transition, the system MUST ENFORCE that each transition has a title. |  |  |
|  | The platform MUST ALLOW a transition *change action* to change the variable values of a game. |  |  |
|  | The platform MUST ALLOW a transition *condition* to evaluate the value of a game variable against a predefined threshold. |  |  |
|  | The platform MUST ENFORCE that a transition condition references an alternative inject. |  |  |
|  | The platform MUST ALLOW the editing of scenarios. |  |  |
|  | IF a scenario has been edited,  THEN the platform MUST save the new version as a separate entity in the database and create a reference to the old version.[[4]](#footnote-4) |  |  |
|  | The platform MUST ALLOW the deletion of scenarios. |  |  |
|  | The platform MIGHT NOT allow the deletion of historical versions of a scenario. |  |  |

**Administrator Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requirement** | **Priority** | **AIT or MT or both?** |
|  | The platform SHOULD ALLOW the creation of user accounts. |  |  |
|  | The platform SHOULD ALLOW a user account to have multiple roles. |  |  |
|  | The platform SHOULD ALLOW an admin to add to and remove roles from user accounts. |  |  |
|  | The platform SHOULD ALLOW each user to delete their own account. |  |  |
|  | The platform SHOULD ENFORCE each account to have adequate authentication measures, such as, but not limited to:   * Password strength, * 2-factor-authentication, * Biometrical authentication. |  |  |

1. Whether this requirement is relevant for the AIT, for the master’s thesis or relevant for both the master’s thesis and the AIT. [↑](#footnote-ref-1)
2. Collaborative game play: All participants have the same goal. In practice, they play through exactly the same scenario. [↑](#footnote-ref-2)
3. An informational snippet is shown to a user, IF they meet certain criteria (i.e. pre-defined keywords) and can provide additional information or an additional challenge.

   An example for an additional snippet would be to show a HR-professional an explanation of why a certain attack vector can be easily exploited, but not show ing this information to an IT specialist, who is expected to know this already. [↑](#footnote-ref-3)
4. This way, it is always clear which version the usage statistics of a scenario refer to. This approach also allows backwards compatibility of scenario statistics. [↑](#footnote-ref-4)