

Task List

Task List Tool

1. Introduction

1.1 Purpose and Scope

The purpose of this tool is to list a number of tasks (or checklist) that might require to be completed by user.

The instructor sets the task list with a number of appropriate conditions for the user to check and complete when required.

1.2 What needs to be designed

Author

In the author interface the instructor enters the check list with its conditions.

Basic Tab

The basic tab displays the following fields:

- Title
- Description (as fckeditor)

There's a button underneath description to "Add Task", when pressed a new layer displays:

- Task
- Description/Instructions: (as fckeditor)

Options:

- Is this task required to complete this activity (checkbox)
- Allow participants to add comments/files for this task (checkbox)
- This task is a child of [this other] parent task
 - Don't display until parent has been completed [off by default] (checkbox)

Important considerations and constraints:

- There must be one or more tasks for this activity to be saved.
- We need the ability to delete, edit tasks
- We need to be able to move activities up or down on the list

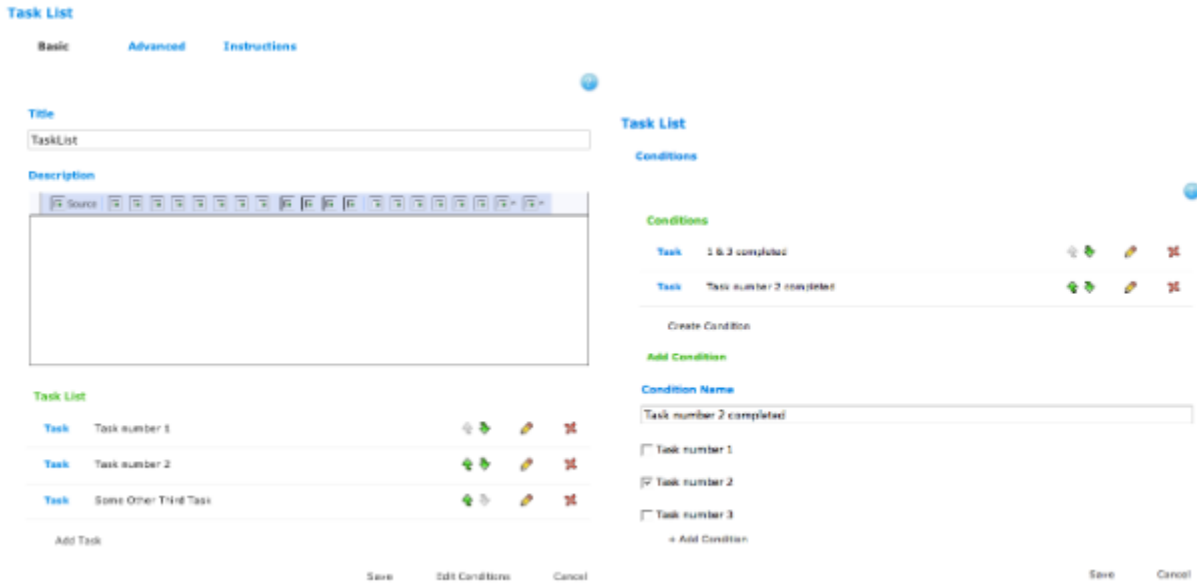
Advanced Tab

- Lock when finished [off by default] (checkbox)
- Tasks have to be completed in sequential order [off by default] (checkbox)
- Add noticeboard after task [off by default]
[and fckeditor to enter reflection]
- Monitor need to verify these tasks before letting participants to complete activity [off by default] (checkbox)
- Participants can contribute tasks

Instructions Tab

All fields and behavior as usual

Conditions Tab



Task List is the first tool to have a Conditions page. This will follow the normal authoring page, accessed by a new button. When the "Edit Conditions" button is clicked, the tool should do the normal save (persisting all the data in the database) and then forward to a new tab with a single Conditions tab. It should still have the help button.

The tab should list any existing condition relating to this content, with the normal up and down arrow keys. Order is very important for conditions as the conditions will be tested in the order shown on the screen.

Then when the user edits a condition, or chooses to create a new condition, a new panel will appear. This panel will have an entry field for the condition name (just plain text box - fckeditor not required). Then all the tasks will be listed with check boxes. The user should select zero or more tasks. Then it will have the normal "+ Add" button to save the condition.

To store in the database, we will need a new Object TaskListCondition, with the following fields for

1. taskList (pointer back to the main Task List content object)
2. uid
3. name
4. orderId
5. set<TaskListItem> taskListItems

The set of taskListItems will probably need to be done with a join table as TaskListCondition to TaskListItem is a many-to-many relationship. If that is too hard, then you could just store the uid's of the TaskListItem's in a set (the way that the LearnerProgress object stores the activity ids for completed and attempted tasks) but that makes the copy tool content harder.

The name field needs to be unique to this TaskList content. ie for any one TaskList content, the name of each condition must be unique. This is needed for defining the conditions in branching.

When copyToolContent is called, ensure that the TaskListConditions are duplicated - in case someone changes the original design after a lesson has started.

Monitor

Summary Tab

This tab will display:

Overall Summary

User	Task1	Task2	Task3
User 1	✓	✓	✓
User 2	✗	✗	✗
User 3	✓	✓	✗

If the Advanced option "Monitor need to verify these tasks before letting participants to complete activity" is ON

Overall Summary

User	Task1	Task2	Task3	Complete Activity?
User 1	✓	✓	✓	[Complete for User 1]
User 2	✗	✗	✗	[Complete for User 2]
User 3	✓	✓	✗	[Complete for User 3]

[Complete for User X] is a button that prompts the monitor whether he's sure about completing this activity for this user or not. If yes, then the activity is marked as completed for the specified user.

Task summary

Task 1

User	Completed?	Time and Date
User 1	✓	June 23, 2008 13:00
User 2	✗	
User 3	✓	June 24, 2008 14:01

Task 2 *(this task is required to finish this activity)*

User	Completed?	Time and Date
User 1	✓	June 24, 2008 16:10
User 2	✗	
User 3	✓	June 25, 2008 18:31

Task 3

- *(Parent activity: Task 1)*
- *Comments/files enabled*

User	Completed?	Time and Date	Comments/Files
User 1	✓	June 25, 2008 06:10	I have completed this activity and the results are 123123. Find more information in the attached file <i>report.doc</i>
User 2	✗		
User 3	✗		

If Advanced option: "" is ON

Task X

- *Added by User2*

User	Completed?	Time and Date
User 1	✗	
User 2	✗	
User 3	✗	

Learner

(to be completed)

Note: Tasks added by users do not have the optionals as in Author

Administration

1.3 Constraints

2. Functional Model

2.1 General Functional Description

2.2 Functional Requirements

2.3 Use cases and User scenarios

3. Data Model

3.1 Description

3.2 ER Diagram

4. Presentation Tier

4.1 General GUI Layout

4.2 Authoring interface and flow

4.2.1 Object stores in the session

4.3 Monitor interface and flow

4.3.1 Object stores in the session

4.4 Learner interface and flow

4.4.1 Object stores in the session

4.5 Admin interface and flow

4.5.1 Object stores in the session

4.6 Export Portfolio interface and flow

4.6.1 Object stores in the session

5. Business Model

If required/needed

6. Data Model

6.1 Database schema

6.2 Data Definitions

7. Sequence Diagram

If required/needed

8. Testing considerations

9. Other technical considerations