Evaluation Survey of a Resource-based Checklist Generation Tool

This survey is part of the design and implementation of a Resource-based Checklist Generation Tool, an MSc Project at the University of Edinburgh.

Principal investigator (PI): Petros Papapanagiotou (pe.p@ed.ac.uk)
Researcher: Aun Sirinaovakul (s2223223@ed.ac.uk)

* Required

Introduction

This project involves a checklist generation tool for WorkflowFM, a logic-based workflow management framework developed at the University of Edinburgh. A checklist generation tool is a tool to create checklists or forms (similarly to Google Forms, Microsoft Form, etc.) that allows people to perform tasks as part of a larger, collaborative workflow of individual processes.

This survey aims to collect feedback related to user performance, functionality, and usability. All personal or identifiable information will be fully anonymised, therefore, please enter your opinions truthfully.

More details are provided in the Participant Information Sheet Here

This survey has been approved by Informatics Research Ethics committee.
The Ethics application number is 486463.

In the following sections, you will be given scenarios with tasks to do in order to generate specific checklists. In the first two tasks, instructions will be provided to help you get used to the system. In the final task, you will be given only the scenario and a list of things you need to do. Between each task, you will be asked to do a functionality questionnaire.

After all the tasks are done, you will be asked to fill in a short questionnaire and an optional open-ended feedback question at the end.

Participant Consent Form

By participating in the study, you agree that: your opinions regarding the user interface and user experience will be used to evaluate the functionality and usability of the project.

- I have read and understood the Participant Information Sheet for this study, I have had the opportunity to ask questions, and any questions I had were answered to my satisfaction.
- My participation is voluntary, and I can withdraw at any time without giving a reason. Withdrawing will not affect any of my rights.
- I consent to my anonymised data being used in an MSc Dissertation as well as potential academic publications and presentations.
- I understand that my anonymised data will be stored for the duration outlined in the Participant Information Sheet.
- I allow my data to be used in future ethically approved research.
- I agree to take part in this study.

1. Agreement *

Check all that apply.

☐ I have read and agreed to both the Participant Information Sheet and the the Participant Consent Form above

Setup

- Please follow this link to start the web application: https://resource-based-checklist-generation.vercel.app/evaluation (the website may take a little while to finish loading)
- Check the evaluation id at the bottom of the screen.
- Enter the Evaluation ID characters in the input field below.
- Click the "*Start Evaluation*" button to start.

2. Evaluation ID *

3. Have you used WorkflowFM or any workflow management frameworks before? *

Mark only one oval.

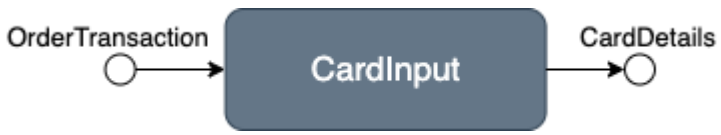
- Yes
- No

Task 1

Scenario: You are a checklist designer for a payment workflow. The owner wants you to create a checklist template for a *CardInput* process. In this process, a customer needs to enter the credit/debit card number, expire date, and security code to pay for some purchased items. The process contains *OrderTransaction* as the input information, with fields as described below. Additionally, the output form needs to be linked with the *CardDetails* output of the workflow process.

You will be given the instructions to create a checklist template based on the scenario above.

The *CardInput* Process



Instructions

You can also read instructions on the web application by clicking on the "Instructions" button on bottom left of the canvas screen.

These instructions are given in order to help users **create a template according to the Format Required**.

1. Click on "*Start Task 1*" to start the task.
2. Create an *CardInput* process template **WITHOUT** auto-generation.
3. Change the name of the form to "*Card Details Form*".
4. Hide the *customer_id* field in the input information section (second section).
5. Then, query 2 new fields using *id* of the *OrderTransaction*.
6. Set the queried tables and queried fields to *transactions - id*, and *item_list - item_id*.
7. Change the 4 visible fields to "*Order ID*", "*Transaction ID*", "*Item ID*", and "*Total Price*".
8. Check the position of each input information field is in order as followed by the **Format Required** below.
9. Change the name of the *OrderTransaction* to *"Purchased Items*".
10. In the form adjustment section (last section), add a new field and name it "*Card Details*".
11. Set the field type of *Card Details* to HEADER.
12. Expand the header and add 3 new fields inside it.
13. For the adjustment of each field inside the header, refer to the **Format Required** below. (*Tips: Dependencies can be managed through the "Dependency Setup" button under each field OR the "Manage Dependencies" button at the bottom of the screen.*)
14. Preview the result to check the template. It should look like the figure given in the **Preview** below.
15. Click on the "*Create*" button to complete the task.

Format Required

Form Name: Card Details Form
Input Information

- Purchased Items (Former *OrderTransaction*)
    - Order ID (Former *id*)
    - Transaction ID (*transactions - id* queried by *id*)
    - Item ID (*item_list - item_id* queried by *id*)
    - Total Price (Former *total_price*)

Form

- Card Details (Field type **Header**)
    - Card Number (Field type **Input**)
        1. **Required:** Yes
        2. **Visibility:** Not Hidden
        3. **Input dependency:** Unlinked
        4. **Output dependency:** Linked to *CardDetails - card_no*
    - Expire Date (Field type **Date**)
        1. **Required:** Yes
        2. **Visibility:** Not Hidden
        3. **Input dependency:** Unlinked
        4. **Output dependency:** Linked to *CardDetails - expire*
    - Security Code (Field type **Input**)
        1. **Required:** Yes
        2. **Visibility:** Not Hidden
        3. **Input dependency:** Unlinked
        4. **Output dependency:** Linked to *CardDetails - code*

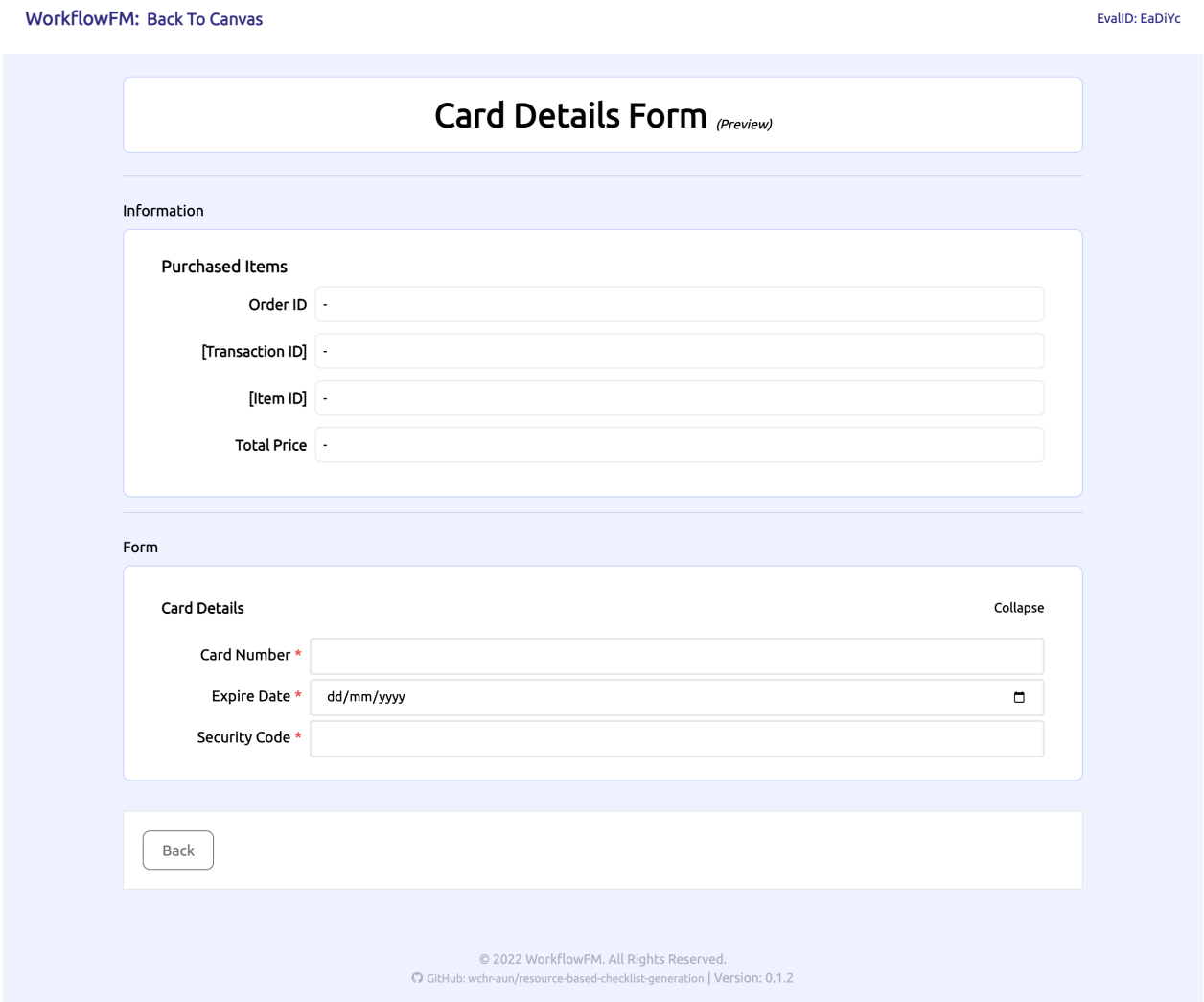
Required is a requirement if a specific field needs to be set as required.

Visibility is a requirement if a specific field needs to be set as hidden.

Input dependency is the source of a field's value in the form. For example, if the input dependency of *Card Number* is linked to *OrderTransaction - id*, the value of this field will always depend on *OrderTransaction - id.* Meanwhile, being unlinked means the field will solely depend on the user input.

Output dependency is the destination of values for a component in the form. For example, the output dependency of *Card Number* is linked to *CardDetails - card_no*. That means the value of *Card Number* will be directly saved to *CardDetails - card_no*. Unlike input dependency, output dependency cannot be unlinked. Being unlinked means that field does not have anywhere to store.

Preview



Task 2

In this task, the scenario is still the **same as Task 1**. However, instead of manually adding information fields and form fields, you will be using **help tools** provided by the system.

The *CardInput* Process

Instructions

You can also read instructions on the web application by clicking on the "Instructions" button on bottom left of the canvas screen.

These instructions are given in order to help users **create a template according to the Format Required**.

1. Click on "*Start Task 2*" to start the task.
2. Create an *CardInput* process template **WITH** auto-generation.
3. Change the name of the form to "*Card Details Form*".
4. Hide the *customer_id* field in the input information section (second section).
5. Then, query 2 new fields using *id* of the *OrderTransaction* **using the suggestion option (is now enabled!)**.
   (Hint: You may need to delete unused fields from the suggestion.)
6. Change the 4 remaining field names to "*Order ID*", "*Transaction ID*", "*Item ID*", and "*Total Price*".
7. Check the position of each input information field is in order as followed by the **Format Required** below.
8. Change the name of the *OrderTransaction* to *"Purchased Items*".
9. In the form adjustment section (last section), check each field in the section if they all follow the **Format Required** below (names, types, dependencies, etc.).
10. Preview the result to check the template. It should look like the figure given in the **Preview** above.
11. Click on the "*Create*" button to complete the task.

Format Required

Form Name: Card Details Form
Input Information

- Purchased Items (Former *OrderTransaction*)
    - Order ID (Former *id*)
    - Transaction ID (*transactions - id* queried by *id*)
    - Item ID (*item_list - item_id* queried by *id*)
    - Total Price (Former *total_price*)

Form

- Card Details (Field type **Header**)
    - Card Number (Field type **Input**)
        1. **Required:** Yes
        2. **Visibility:** Not Hidden
        3. **Input dependency:** Unlinked
        4. **Output dependency:** Linked to *CardDetails - card_no*
    - Expire Date (Field type **Date**)
        1. **Required:** Yes
        2. **Visibility:** Not Hidden
        3. **Input dependency:** Unlinked
        4. **Output dependency:** Linked to *CardDetails - expire*
    - Security Code (Field type **Input**)
        1. **Required:** Yes
        2. **Visibility:** Not Hidden
        3. **Input dependency:** Unlinked
        4. **Output dependency:** Linked to *CardDetails - code*

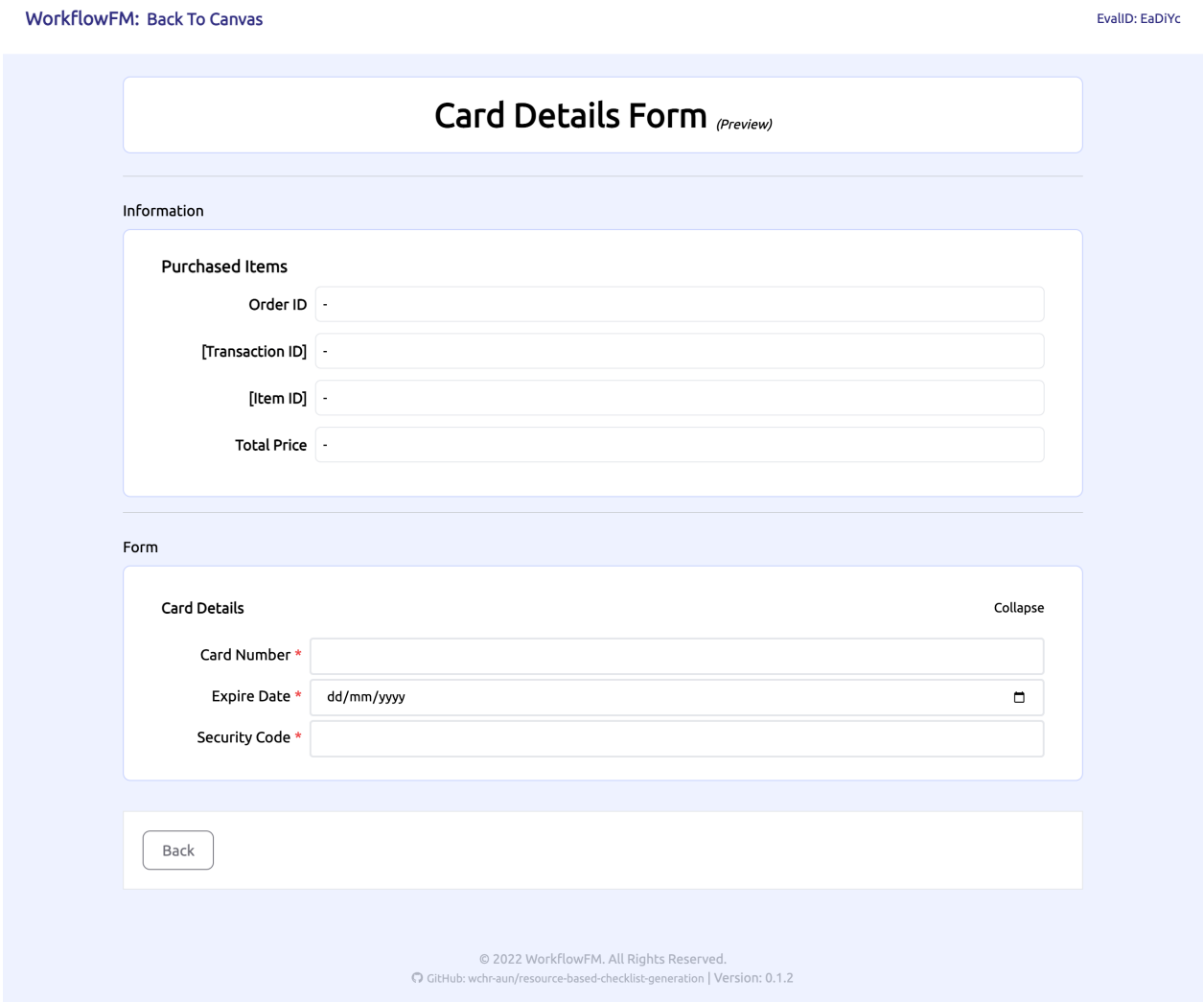
Required is a requirement if a specific field needs to be set as required.

Visibility is a requirement if a specific field needs to be set as hidden.

Input dependency is the source of a field's value in the form. For example, if the input dependency of *Card Number* is linked to *OrderTransaction - id*, the value of this field will always depend on *OrderTransaction - id.* Meanwhile, being unlinked means the field will solely depend on the user input.

Output dependency is the destination of values for a component in the form. For example, the output dependency of *Card Number* is linked to *CardDetails - card_no*. That means the value of *Card Number* will be directly saved to *CardDetails - card_no*. Unlike input dependency, output dependency cannot be unlinked. Being unlinked means that field does not have anywhere to store.

Preview



Task 1 and Task 2 Questionnaire

4. Functionality Questionnaire *

Mark only one oval per row.

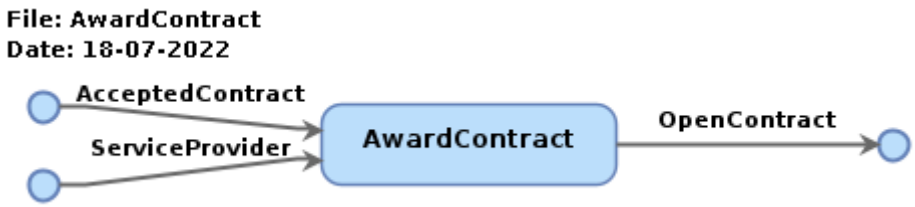
	Disagree	Partly Disagree	Neutral	Partly Agree	Agree
I was able to perform task 1 and task 2 successfully.					
I could easily query new input information for the fields.					
The form adjustment section was easy to use.					
I could easily add and adjust new form fields in the form adjustment section.					
I could easily link dependencies of each form field.					
The help tools (the query suggestion and the auto-generation) were easy to use.					
The help tools were helpful and made the whole process easier.					

Task 3

Scenario: You are a checklist designer for a healthcare workflow. A doctor asks you to create a checklist template for an *AwardContract* process. In this process, a clinician agrees to another member of clinical staff, (the *ServiceProvider)* to provide a medical service on their behalf (i.e. delegating the task to someone else). The process contains two useful pieces of input information: the *ServiceProvider* that contains information on the doctor who is assigned to the contract, and *AcceptedContract* that contains information about the contract itself (i.e. the nature of the clinical task, details about the patient, etc).

You need to create a checklist template based on the scenario above with fully linked dependencies followed the **Format Required**. You are also **allowed to use any help tools** introduced in Task 2 to perform this task; however, there are **no instructions** in this task.

The *AwardContract* Process



Format Required

You can also look at Format Required on the web application by clicking on the "Format Required" button on bottom left of the canvas screen.

Form Name: Award Contract Checklist

Input Information

- Provider (Former *ServiceProvider*)
  - Provider's Name (*staff - name* queried by *actorid*)
  - Provider's Surname (*staff - surname* queried by *actorid*)
  - Service Name (*services - name* queried by *serviceid*)
  - Service Description (*services - description* queried by *serviceid*)
- Patient (Former *AcceptedContract*)
  - Patient ID (*requestedservices - patientid* queried by *reqservid*)

Form Adjustment

- Open Contract (Field type **Header**)
  - idcontract (Field type **Input**)
    1. **Required:** Any
    2. **Visibility:** Hidden
    3. **Input dependency:** Linked to *AcceptedContract - idcontract*
    4. **Output dependency:** Linked to *OpenedContract - idcontract*
  - reqservid (Field type **Input**)
    1. **Required:** Any
    2. **Visibility:** Hidden
    3. **Input dependency:** Linked to *AcceptedContract - reqservid*
    4. **Output dependency:** Linked to *OpenedContract - reqservid*
  - providerid (Field type **Input**)
    1. **Required:** Any
    2. **Visibility:** Hidden
    3. **Input dependency:** Linked to *AcceptedContract - providerid*
    4. **Output dependency:** Linked to *OpenedContract - providerid*
  - time_requested (Field type **Input**)
    1. **Required:** Any
    2. **Visibility:** Hidden
    3. **Input dependency:** Linked to *AcceptedContract - time_requested*
    4. **Output dependency:** Linked to *OpenedContract - time_requested*
  - Date Opened (Field type **Date**)
    1. **Required:** Yes
    2. **Visibility:** Not Hidden
    3. **Input dependency:** Unlinked
    4. **Output dependency:** Linked to *OpenedContract - time_opened*
  - Opened State (Field type **Constant**)
    1. **Value*:** 3
    2. **Input dependency:** Unlinked
    3. **Output dependency:** Linked to *OpenedContract - stateid*

Value* is the specific value of a constant field.

Required is a requirement if a specific field needs to be set as required.

Visibility is a requirement if a specific field needs to be set as hidden.

Input dependency is the source of a field's value in the form. For example, if the input dependency of *Card Number* is linked to *OrderTransaction - id*, the value of this field will always depend on *OrderTransaction - id.* Meanwhile, being unlinked means the field will solely depend on the user input.

Output dependency is the destination of values for a component in the form. For example, the output dependency of *Card Number* is linked to *CardDetails - card_no*. That means the value of *Card Number* will be directly saved to *CardDetails - card_no*. Unlike input dependency, output dependency cannot be unlinked. Being unlinked means that field does not have anywhere to store.

Task 3 Questionnaire

5. Functionality Questionnaire *

Mark only one oval per row.

	Disagree	Partly Disagree	Neutral	Partly Agree	Agree
I had no problem performing this task without any instructions.					
I could easily query new fields in the information input.					
Adjusting the form to the requirement was troublesome in this task.					
I found the help tools (query suggestion and auto-generation) provided in the system helpful and helped me to perform this task.					

You have done all the tasks!

***** Don't forget to submit the last task on the website! *****

Next is the usability questionnaire. After performing all the tasks, how do you feel about this system?

(Optional) You can also go to https://resource-based-checklist-generation.vercel.app/?env=healthcare (healthcare scenario) and https://resource-based-checklist-generation.vercel.app/?env=payment (payment scenario) to get a better idea of the overall prototype.

6. Usability Questionnaire *

Mark only one oval per row.

	Disagree	Partly Disagree	Neutral	Partly Agree	Agree
I think that I would like to use this system frequently.					
I found the system unnecessarily complex.					
I thought the system was easy to use.					
I think that I would need the support of a technical person to be able to use this system.					
I found the various functions in this system were well integrated.					
I thought there was too much inconsistency in this system.					
I would imagine that most people would learn to use this system very quickly.					
I found the system very cumbersome to use.					
I felt very confident using the system.					
I needed to learn a lot of things before I					

could get going with this system.					
After learning through the instructions, I would be able to create a checklist template by myself.	⬭	⬭	⬭	⬭	⬭
The help tools (the query suggestion and the auto-generation) made it easier to create a template.	⬭	⬭	⬭	⬭	⬭

7. Feedback (optional)

Is there anything you want to mention or explain more about the system?
(For example: bugs found, overwhelming of the system, improper positioning of forms and buttons, etc.)

If no, you can leave the feedback empty.