QUESTIONNAIRE MANAGEMENT INTERFACE
Requirements Document

Mahir Hiro (s3561119)
Hleb Shmak (s3774244)
Robert Rey (s3309770)
Pal Poshyachinda (s2834839)
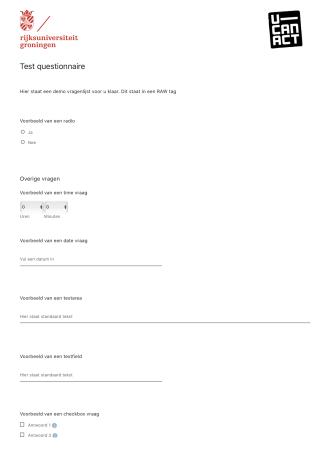Krishan Jokhan (s3790746)

research
able;

Contents

1. Introduction

A questionnaire is a research instrument consisting of a series of questions for the purpose of gathering information from respondents. The goal for this was project to build a front-end interface for the questionnaire engine, that enables users to define their questionnaires. Currently, to be able to build such a questionnaire the users of the application have to edit a JSON format text to create a questionnaire. Which then is rendered through a back-end system which is already implemented. A short example of how JSON text for a questionnaire may look depicted below:

Questionnaire JSON

[{"type":"raw","content":"<p class=\"flow-text\">Hier staat een demo vragenlijst voor u klaar. Dit staat in een RAW tag</p>"},
{"id":"v1","type":"radio","show_otherwise":false,"title":"Voorbeeld van een radio","options":[{"title":"Ja","shows_questions":["v2"]},
{"title":"Nee","shows_questions":["v2"]}]},{"id":"v2","hidden":true,"type":"range","title":"Voorbeeld met een range","labels":["heel weinig","heel veel"]},
{"id":"v3","type":"time","hours_from":0,"hours_to":11,"hours_step":1,"title":"Voorbeeld van een time vraag","section_start":"Overige vragen"},
{"id":"v4","type":"date","title":"Voorbeeld van een date vraag","labels":["helemaal intuïtief ","helemaal gepland"]},{"id":"v5","type":"textarea","placeholder":"Hier
staat standaard tekst","title":"Voorbeeld van een textarea"},{"id":"v6","type":"textfield","placeholder":"Hier staat standaard tekst","title":"Voorbeeld van een
textfield"},{"id":"v7","type":"checkbox","required":true,"title":"Voorbeeld van een checkbox vraag","options":[{"title":"Antwoord 1","tooltip":"Tooltip 1"},
{"title":"Antwoord 2","tooltip":"Tooltip 2"},{"title":"Antwoord 3","tooltip":"Tooltip 3"}]},{"id":"v8","type":"likert","title":"Voorbeeld van een
likertschaal","tooltip":"some tooltip","options":["helemaal oneens","oneens","neutraal","eens","helemaal eens"]},{"id":"v9","type":"number","title":"Voorbeeld van
een numeriek veld","tooltip":"some tooltip","maxlength":4,"placeholder":"1234","min":0,"max":9999,"required":true},
{"id":"v10","type":"textfield","placeholder":"Hier staat standaard tekst","title":"Voorbeeld van een klein vrij textveld"},{"id":"v11","title":"Voorbeeld van een
expandable","remove_button_label":"Verwijder","add_button_label":"Voeg toe","type":"expandable","default_expansions":1,"max_expansions":10,"content":
[{"id":"v11_1","type":"checkbox","title":"Met een checkbox vraag","options":["Antwoord A","Antwoord B","Antwoord C","Antwoord D","Antwoord E","Antwoord
F"]}]},{"id":"v12","type":"dropdown","title":"Waar hadden de belangrijkste gebeurtenissen mee te maken?","options":
["hobby/sport","werk","vriendschap","romantische relatie","thuis"]}]

SUBMIT

Clearly, this system was not feasible for in the long term since editing JSON text, whether a user of the system is or is not familiar with the basic programming knowledge is not user friendly. Henceforth, the goal of this project is too create a simple yet elegant interface to ease this process when creating questionnaires. Which in turns creates this JSON by through the new interface. Below is a picture to see how the JSON get's rendered in the system after the JSON is created.

rijksuniversiteit
groningen

U—CAN ACT

Test questionnaire

Hier staat een demo vragenlijst voor u klaar. Dit staat in een RAW tag

Voorbeeld van een radio

○ Ja

○ Nee

Overige vragen

Voorbeeld van een time vraag

| 0 | ⇕ | 0 | ⇕ |

Uren   Minuten

Voorbeeld van een date vraag

Vul een datum in

Voorbeeld van een textarea

Hier staat standaard tekst

Voorbeeld van een textfield

Hier staat standaard tekst

Voorbeeld van een checkbox vraag

☐ Antwoord 1 ⓘ
☐ Antwoord 2 ⓘ

2. Group meeting recap

2.1 27th February

- We created a separate board (Trello link is in general) for the authentication part.

- The goal for this block is to guarantee to satisfy the three user stories on overleaf or Trello.

- The first sprint will officially end on the 12th of march where the goal is to be able to create an account and connect the details to the database

- All of us will be working on this to see how we can assess everyone's current react skills and see how accurate the first planning poker number were according to how we rated it

- Mahir, Hleb and Robert (3/5 members) will be on a holiday from the 4th - 9th March

- This weekend everyone will practice and then meet up to discuss questions on the 3rd of March, and maybe send a question or two to be cleared out

2.2 12th March

This meeting was after the meeting with Frank Blauuw.

- We decided, that we should elaborate more on Requirements Document.

- We divided the workload:

  - **Toolbox** with all question types: Robert, Hleb, Mahir.
  - Dragging types to the **Edit Area**: Robert, Hleb, Mahir, Ou.
  - Rendering a template of the type in the editor upon release: Krishan and Ou.
  - Editing questions: Ou.

2.3 23rd March

This is a small meeting done online to get updates from each other regarding the current sprint

- Due to most of us heading home because of the Corona virus, some time was needed to adjust to the new workplace. This resulted in most of the group not being able to make much progress. Because of this, an extension of one week was asked to Frank, which was accepted.

- Krishan and Ou were both refactoring parts of the code unknowingly from each other. As a whole group we decided to keep Trello more up to date from now on to prevent such events from happening.

- Research in a more preferred drag and drop framework will be done.

- It is decided to use icons in the toolbox representing each question type, and when they're (in progress of being) dragged to the editor view, the icon will transition into a template representing the question type. This template is the one also eventually rendered in the editor view.

Also, on this day, we updated with Frank the current progress via Slack.

2.4 4th April

This is a small meeting done online to get updates from each other regarding the progress since the demo session. During this meeting the team also discuss the points of improvement by watching other presentations which the team found intriguing!

2.5 21th April

The team was scheduled to meet in a week with the client. Thus, it was a notice to accelerate certain tasks which may have been prolonged. Additionally, we had to implement a number of additional features such as:

- Duplicating questions
- Double tap on a title to edit it
- Theme selector
- Hidden questions
- Raw data type of question type

2.6 30th April

This meeting was after the meeting with Frank Blauuw. We decided upon 2 last sprints.

- We formed the tasks on Trello.
- We decided to add two new question types: Toggled and Expandable questions.
- After implementing these questions we decided to implement the authentication part of the project.
- The final sprint will be fully dedicated to testing, refactoring and creating the documentation for the project.

Also, we discussed the possibility of deploying the server to test our project.

2.7 10th May

The team formed the. following goals for the next sprint.

- Create demo version of the application and gather some feedback from the users (students/professors).
- The sprint is mainly dedicated to refactoring.
- We decided to add the tutorial video for the application on the Editor page of the application.

2.8 20th May

This sprint was mainly dedicated to the documentation. We decided that we should elaborate on the following documents: Readmi, Architectural and Requirements documents.

3. User Stories

- **The Professor**:
  The Professor represents an ordinary user of the application. The Professor is already given the credentials and needs to be logged to the **System** in order to create/modify questionnaires.

- **The Registered User**:
  The Registered User has an account within the **System**. The User can create multiple questionnaires using the **System**, as well to delete them. The User is able to see the rendered questionnaire at any stage of the development of the questionnaire.

3.1 Must Have Functional Requirements

- As a `Professor`, I want to

    1. Log in (authenticate)/ log out.

- As a `Registered User`, I want to

    1. Create new questionnaire. [CRUD]
    2. Delete existing questionnaire. [CRUD]
    3. Modify the questionnaire. [CRUD]
    4. Render the questionnaire.
    5. Clone existing questionnaire.
    6. Drag the question types to the discard area.

3.2 Should Have Functional Requirements

- As a `Professor`, I want to

    1. View the profile.

- As a `Registered User`, I want to

    1. Modify the question type.
    2. Toggle the question.
    3. Duplicate the question.
    4. Erase questionnaire.

3.3 Could Have Functional Requirements

- As a `Registered User`, I want to

    1. Change the theme.
    2. Get the JSON of the questionnaire.
    3. Watch short tutorial video, explaining how to create questionnaires.

4.   Non functional Requirements:

A system-wide constraint is called a non-functional requirement.

The FURPS model organizes all requirements into five categories:

F = **Functional**

U = **Usability**

R = **Reliability**

P = **Performance**

S = **Supportability**

The FURPS+ model adds a few more categories, which we will see in the following section.

Usability:

Usability is yet another classical nonfunctional requirement that addresses a simple question we should ask ourselves: How hard is it to use the product? Keeping this question in mind the team went on to develop the application to make it as user friend as possible, a few examples of adding extra features to maximize user usability is:

- Option to delete all questionnaires

- Theme picker to best match the users preferred preference.

- A "scroll top" button to make it easier to navigate to the top of the page

1. **Learnability**: Users can quickly complete the main actions require once they see the interface due to the simplistic layout implemented

2. **Efficiency**: In terms of efficiency to to reach the end goal this is specific to each user, which means if a user wants to create a questionnaire in a short duration of time it is possible!

3. **Memorability**: We implemented the front-end so users can return to the interface and start working with it right away.

4. **Errors**: Users can barely make mistakes using this system since there is validation checks across the the platform

5. **Satisfaction**: The client and his team at `Researchable` found the design very pleasant to use!

Reliability:

This quality attribute specifies how likely the system or its element would run without a failure for a given period of time under predefined conditions. However, for obvious reasons this cannot be measured until the application is in production, henceforth, this section will be updated in the near future.

Performance:

This quality attribute specifies how likely the system or its element would run without a failure for a given period of time under predefined conditions. However

Supportablility:

There are 2 main questions which need to be answered. How will system be extended and who maintains the system? To answer the first question the team and the client have been in very close contact, instead of just portraying the demo at the end of every block, the client and his team suggested it would be better to have a sprint review every 2 weeks. This was a great idea because the team at `Researchable` has been active in observing what the team has created, this makes extending the project easier in the near future. The Readme is also document if at any point a developer is confused about a function/ design. Finally, to answer the latter question, the team has implemented 2 software technologies to keep the code as up to date as possible whilst the team at `Researchable` takes over.

- **Dependabot:** Dependabot creates pull requests to keep your dependencies secure and up-to-date.

- **Snyk:** Cybersecurity platform that helps developers find vulnerabilities in their open source applications, it also creates pull requests to update the existing code automatically.

These following sections are not part of the regular FURPS model.

Security:

For the authentication section, the security aspect lies in the **JSON Web Token** (compact and self-contained way for securely transmitting information between parties as a JSON object) is used to securely authenticate users into the system. Additionally, the team used Snyk (cybersecurity platform that helps developers find vulnerabilities in their open source applications) to check for security vulnerabilities in our dependencies for the current and existing code base. The great part about Snyk is that once setup correctly, it will keep running and always check the current dependencies (whenever the master branch is updated). This keeps the code maintainable and extensible for the future!

Availability:

The **System** should be fully functioning on the following browsers: Chrome 80+, Firefox 60+, Safari 10.0+.

5. Planning Poker

We decided to use the Planning Poker estimate the complexity and classify the user stories. We did **Planing Poker** without the **Product Owner**, instead we had a **Scrum Master** responsible for preparing and asking the questions. Additionally, we also integrated this into our `Slack` to ease the process of rating user stories. The planning poker points are in between 1 and 10.

Must Have Functional Requirements

1. Authentication ≡ 7 pts.

    • Authentication requires the knowledge of the Auth0 technology.

    • Authentication requires the **Landing page** (page where you appear after the authentication).

2. Log out ≡ 4 pts.

    • Log out requires the knowledge of the Auth0 technology.

3. Create new questionnaire ≡ 2 pts.

4. Delete existing questionnaire ≡ 2 pts.

5. Modify the questionnaire ≡ 8 pts.

    • Requires the definition of all properties of all questions.

6. Render the questionnaire ≡ 7 pts.

    • Requires the routing to the u-can-act API.

7. Clone existing questionnaire ≡ 2 pts.

8. Drag the question types to the discard are ≡ 5 pts.

    • Requires drag-and-drop interface.

6. Testing

Login page:

☐ User enters the correct credentials

☐ User enters the correct username but wrong password

☐ User enters the correct password but wrong username

☐ User forgets the credentials.

Landing page:

☐ User clicks the questionnaire.

☐ User creates new questionnaire.

☐ User deletes the questionnaire.

☐ Amount of questionnaires equals to the number of icons.

Questionnaire creator page:

Edit Dialog

☐ Test the properties for each question

☐ Changing the title of a question alone

☐ Changing the options of a question alone

☐ Changing the specific options of certain questions

Preview

☐ JSON matches the number of questions below

☐ Clicking on a question open's it

☐ Clicking on a raw question does not open

☐ Delete question functionality.

☐ Duplicate question functionality.

☐ Erase question functionality.

☐ Light/Dark mode change.

☐ Render questionnaire functionality

Navigation leftside bar

☐ When you click on the 3 lines, the navigation bar opens.

7. Contact

7.1 Frank Blauuw

- Office Location: BB 564

- Email `f.j.blaauw@rug.nl`

7.2 Mahir Hiro

- Email `m.hiro@student.rug.nl`

7.3 Alexander Fyodorov

- Email `a.fyodorov@student.rug.nl`

7.4 Hleb Shmak

- Email `h.shmak@student.rug.nl`

7.5 Robert Rey

- Email `r.rey@student.rug.nl`

7.6 Krishan Jokhan

- Email `k.p.jokhan@student.rug.nl`

7.7 Pal Poshyachinda

- Email `p.poshyachinda@student.rug.nl`

8. Abbreviations

1. **JSON**: JavaScript Object Notation. It is a lightweight format for storing and transporting data. Often used when data is sent from a server to a web page.

2. **Deployment**: The process of setting the netlify server in order to launch the React App.

3. **JWT**: Web technology used for secure authentication.

9. Changelog

Who	When	Which section	What
M. Hiro	February 20, 2020	The document	Created the document.
M. Hiro & H. Shmak.	February 24, 2020	User Stories section.	Completed the entire section
M. Hiro	February 24, 2020	Meeting # 1 Recap	Completed the entire section
M. Hiro & H. Shmak.	February 27, 2020	Functional & Non-functional requirements	Completed the entire section
H. Shmak	February 25, 2020	User Stories	Added planning poker points for sprint
M. Hiro	February 27, 2020	Group Meeting recap.	Completed the entire section
H. Shmak	March 11, 2020	Meeting # 2 Recap	Completed the entire section
M. Hiro	March 12, 2020	Group Meeting recap.	Completed the entire section
M. Hiro	March 23, 2020	Group Meeting recap.	Completed the entire section
M. Hiro	March 26, 2020	Contact Section.	Completed the entire section
H. Shmak	April 2, 2020	Useful Links	Completed the entire section
M. Hiro	April 16, 2020	Introduction	Rewrote the section
M. Hiro	April 16, 2020	Requirements	Edited the section based on feedback from TA
H. Shmak	April 30, 2020	Group Meeting recap.	30th April meeting
M. Hiro	May 1, 2020	Group Meeting recap.	4th April meeting & 21th April meeting
M. Hiro & H. Shmak	May 8, 2020	Introduction	Edited the section
M. Hiro	May 9, 2020	User Stories	Re wrote the non functional requirements (FURPS Model)
H. Shmak	May 9, 2020	User Stories	Added extra non functional requirements after the FURPS Model
H. Shmak	May 9, 2020	User Stories	Planning Poker
H. Shmak & Mahir	May 11, 2020	Testing requirements	Created section and added some points
M. Hiro	May 11, 2020	Testing requirements	added some points
M. Hiro & H. Shmak	May 17, 2020	Abbreviations	Started the section
H. Shmak	May 17, 2020	Whole document	Changed the document structure & fixed the hyperlinks
M. Hiro	May 24, 2020	Introduction	added extra information
H. Shmak	May 28, 2020	Group Meeting recap.	10th April meeting & 20th April meeting