MODULE 11

DESIGN REPORT
SUSTAINABILITY SCORECARD

GROUP 4: AKMAL ALIKHUJAEV, BUGRA VEYSEL YILDIZ, SHASANK SEKHAR PANDEY, VLADIMIR KOBZEV

SUPERVISOR: LUIS FERREIRA PIRES

COMPANY: FOODCHAINID

19 TH APRIL ’23

UNIVERSITY OF TWENTE.

PREFACE

Sustainability goals are becoming more and more prominent in the commercial landscape, with most companies setting targets to reduce their carbon footprint or other environmental impacts. This is also important as most countries are setting up sustainability goals that are required to be followed by the companies and factories operating in them.

However, in most cases a company or a factory does not produce all components of their products in-house. Moreover, the raw materials used are usually sourced from other companies or suppliers and transported through other third-party companies, all of which makes calculating or predicting one’s sustainability harder.

The aim of our project was to tackle this exact problem by creating a singular platform that streamlines these complex problems, so that businesses can focus on making their practices more sustainable.

TABLE OF CONTENTS

Contents

1. INTRODUCTION

With the world moving towards preserving the climate, stop global warming and in general towards a sustainable future, not just people, but even businesses need to make their practices more sustainable.

For businesses, the reasoning for a move towards sustainability is two folds. Firstly, with the prices of raw materials increasing due to over-use, the cost of operating for businesses is expected to rise with time. Secondly, there is a pressure from governments that have already set up goals for sustainability (e.g., The Paris Agreement), making sustainability not just a choice, but a requirement.

However, moving towards sustainability is not an easy process, as it requires analysis of vast amounts of data to analyze unsustainable practices. The difficulty of this process is increased multifold, when you take into consideration the complexity of most supply chains, which might shift the source of unsustainability of one's practices away from their own factories and business to a third party.

The goal of our project is to develop a platform that helps businesses streamline, this data collection process from third parties, helping them analyze the sustainability of their partners and in turn their own.

In the following sections, we explain our vision for the project and how we realized it, explaining in great details how we performed requirement analysis, designed, implemented, and tested the application. Lastly as the application was developed for an external organization, we provide a detailed risk analysis and any improvements that can be made to the application if used in the future.

1.1 OUR VISION

The goal of the project was to create a platform for businesses to evaluate their sustainability, specifically by surveying their suppliers, to know more about their practices. To achieve this goal, we visioned a web application, that allows users to create questionnaires with custom grading schemes, which they can share with their suppliers. The users would also be able to share the same questionnaire with multiple suppliers, allowing them to compare results, with the aim being that this feature would allow them to choose between suppliers based on sustainability.

For suppliers, receiving a request to answer a questionnaire, we visioned a simple and straight-forward answering process, allowing them to login through custom links shared using email. The reason for custom links was to allow suppliers to answer in multiple sessions.

Lastly, the web application would also provide special access to administrators, allowing them to invite users onto the platform, and other administrator related functions that we have explored in the next sections.

2. REQUIREMENTS ENGINEERING

2.1 STAKEHOLDERS

As our product is a B2B solution, our major stakeholders are FoodChainID, the businesses that buy/use this project and the suppliers who are asked to answer the questionnaires created through our platform. For better understanding of the stakeholders and their impact on the project, we describe it below:

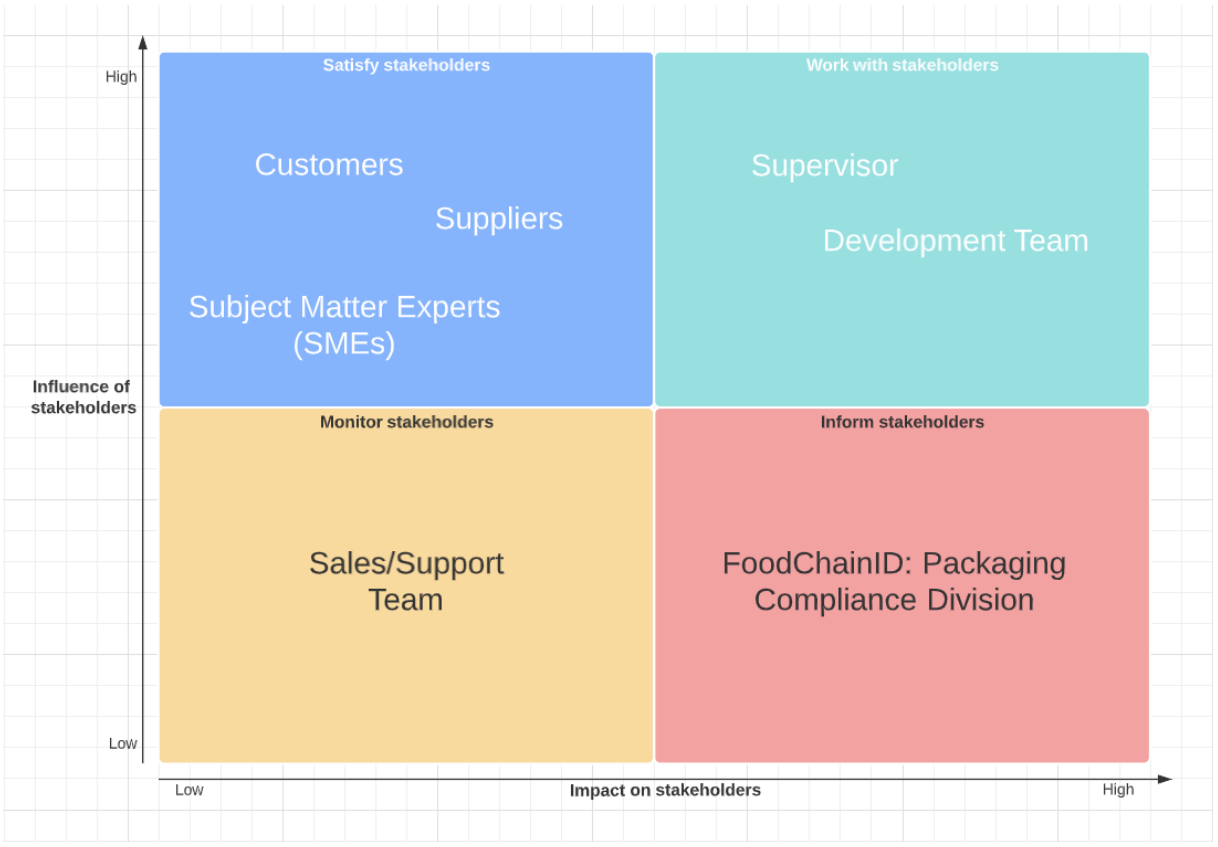


Figure 1: Stakeholder map

Figure 1 represents the stakeholders in the form of a map, showcasing the amount of influence (on the y-axis) and their impact/availability (on the x-axis). For example, Suppliers, Customers and Subject Matter experts have high influence on the design of the solution as they are the people who interact with the solution once it is developed, but have lower impact on the development process, as they are not part of the development process.

1. **Stakeholders to be Kept Satisfied**: These are the stakeholders that can be seen in the blue sticky notes, these are the prospective customers/users, and therefore, the developed solution must be designed with them in mind. This is why they have high influence. However, as these stakeholders are not part of

the development process, they have low impact on the process. We have included the Customers/businesses that will use the platform, the suppliers that will answer the questionnaires on the platform, and Subject Matter Experts employed by FoodChainID to create templates.

2. **Stakeholders that are actively engaged:** These stakeholders (in greenish-blue sticky notes) have both high impact and high influence in the final product as these are the stakeholders that we work with. For our project, this includes our supervisor, who we actively report our progress and ideas to, and the development team at FoodChainID as the product needs to maintain the design language and the Stack used by them for their other products.
3. **Stakeholders to be monitored:** These stakeholders (in yellow sticky notes) both have low impact and low influence on the project. For us, this includes the sales/support team of FoodChainID who would in turn be responsible for marketing the solution and helping the users in case of any difficulties. These stakeholders must also be kept in mind as they might require justification and knowledge about the developed solution.
4. **Stakeholders to be kept informed:** These stakeholders (in red sticky notes) have a high impact on the final solution. For us, this includes The Packaging compliance division of FoodChainID, as they are the product owners and therefore define the ideas and requirements that are to be implemented in the final solution.

2.2 USER STORIES

User stories help in assessing requirements for the project as they allow us to define the product features that will deliver value to the users of the system. The three types of users have different roles and therefore different needs and wishes for the system. We have established the following user stories to better understand these wishes.

2.2.1 FoodChainID (Administrator)

1. As an administrator, I want to be able to invite customers to the platform.
2. As an administrator, I want to be able to get an overview of all active customers.
3. As an administrator, I want to be able to create Sustainability templates for our customers.
4. As an administrator, I want to be able to delete a customer.

2.2.2 Customer (User)

1. As a user, I want to be able to get all my previously created templates.
2. As a user, I want to be able to create a new template.
3. As a user, I want to be able to create suppliers.

4. As a user, I want to be able to get an overview of all my suppliers.
5. As a user, I want to be able to share a template with multiple suppliers.
6. As a user, I want to be able to get all my shared templates for each supplier.
7. As a user, I want to be able to track the status of my shared templates.
8. As a user, I want to be informed when a supplier has finished answering a questionnaire.
9. As a user, I want to be able to check an answered questionnaire.

2.2.3 Supplier

1. As a supplier, I want to be informed when I am required to answer a questionnaire.
2. As a supplier, I want to be able to stop answering a questionnaire at any time and continue later without losing any of my answers.

2.3 REQUIREMENTS

Based on the user stories, and the basic requirements mentioned to us by FoodChainID we have organized them under functional and non-functional categories.

2.3.1 Functional Requirements

To ease our understanding of the requirements and to assign priority to each requirement, we use the MoSCoW[1] style of prioritization.

Must:

1. The application must allow the administrator to invite/create users.
2. The application must have a view for the administrator to see all the users of the platform.
3. The application must have a login for existing users.
4. The application must allow users to create and add suppliers.
5. The application must allow users to create questionnaire templates.
6. The application must allow users to use a global questionnaire template.
7. The application must allow users to share questionnaires with multiple suppliers.

[1] is a prioritization technique used in management, business analysis, project-management, and software development to reach a common understanding with stakeholders on the importance they place on the delivery of each requirement; it is also known as MoSCoW prioritization or MoSCoW analysis.

8. The application must have a view for the user to see/track all answered questionnaires.
9. The application must have a view for the user to see all templates.
10. The application must have a view for the user to see all its added suppliers.
11. The application must inform a supplier to answer a questionnaire through email, with a link to open the questionnaire.
12. The application must allow suppliers to submit their answers after completion.

Should:

1. The application should allow users to reset their passwords.

Could:

1. The application could have notifications regarding questionnaire status for users.
2. The application could allow the administrator to create global templates for all users.
3. The application could have analytics view for administrators to see user's usage patterns.
4. The application could allow administrators to create API keys.
5. The application could allow creation of questionnaires through API.

Won't:

1. The application won't have SSO integration for users.
2. The application won't have a dashboard for suppliers.

2.3.2 Non-functional (Quality) Requirements

These are the requirements that complement the functional requirements, defining a condition or capability of the application that can be used to judge its operation. Our quality requirements are as follows:

1. The application frontend should use FoodChainID defined colors.
2. When users or suppliers are invited to sign up or answer a questionnaire, they should be able to navigate to the correct page through a link in their invitation email.
3. The application should be capable of running on different browsers namely Microsoft Edge, Google Chrome, Mozilla Firefox, and Safari.
4. The system should be able to handle at least 100 simultaneous users.
5. The application should show results in less than 3 seconds.

3. PLANNING

This section has been divided into 3 sections, explaining different aspects of our planning namely our development workflow, our development branching model, and our use of CI/CD.

3.1 DEVELOPMENT WORKFLOW

For our project development we decided to use the AGILE methodology, organizing our development in sprints of 2 weeks, with predefined goals for each sprint. Each sprint involved two meetings with our supervisor to establish our progress and assess any issues that were encountered during the sprint.

Sprint 1: The first sprint involved getting to know each other and as we were doing an external project, finding a supervisor. We also used the first sprint to ideate and have a thorough requirement analysis, identifying our stakeholders and their needs. We also used this time to create Use case diagrams that can be found in section 4.1.1. Lastly, we created wireframes for the system which can be found in the section 4.3.

Sprint 2: The second sprint was followed with the creation of an Entity Relation Diagram (available in the Technical Specification) to map out our Models and start development on the project. We also developed Activity diagrams to help us define the flow of the main events, allowing us to create a better structured solution. These Activity Diagrams can be found in the section 4.1.2.

We also moved to actual project development under this sprint, where the backend and frontend were simultaneously developed in smaller teams.

Sprint 3: The third sprint was focused solely on developing functionalities, where we divided ourselves into two teams, one working on frontend and the other on backend. Our main goals for this sprint were to develop authentication, general functionalities, and a cohesive design.

We also worked on the report during this sprint, focusing on the Implementation and design sections, explaining our choices.

Sprint 4: After showing our progress to the supervisor and the company, and with some valued feedback, the fourth sprint comprised of writing tests, testing our application, and fixing and changing features as requested by our supervisor and client.

Sprint 5: The final sprint was dedicated to writing this detailed report, creating user manuals that can be found in Appendix A and a poster for the project.

3.2 DEVELOPMENT BRANCHING MODEL

Every agile team developing applications strives to have shorter release cycles, continuous integration of code, high degree of collaboration and lower time to market.

To enable this, developers must choose the development workflows that enables said agility. For this project, given the short period of time and a large number of requirements, the team has decided to embrace trunk-based development methodology.

As the name suggests, the trunk-based development model revolves around the idea of having 1 main branch called “Trunk” in a Git version control system. The methodology encourages developers to make small, concise, and frequent commits to the “trunk” branch so that the code frequently integrates with the rest of the codebase. Furthermore, the added advantage of this approach is that the likelihood of the code breaking after a change due to merge conflict introduced by working on multiple files is minimized. Secondly, it allows all team members to have a more holistic view of ongoing development. Thirdly, it was intended to shorten the software release cycles and allow deployment to production daily (Hammant, 2017).

However, this model adds an additional layer of complexity. In order to orchestrate such a workflow, the team has embraced Test Driven Development (TDD) which ensures that the code that is being deployed to production works and integrates with the rest of the system. Without automated tests, trunk-based development can only ensure that there are no structural breakages, but it cannot catch semantic ones.

Lastly, trunk-based development encourages developers to automate continuous integration and deployment to direct the focus on business logic and new functionality rather than frequent manual deployments.

3.3 CONTINUOUS INTEGRATION / DEPLOYMENT

According to one of the most prominent figures in the software engineering field, Martin Fowler (2006), Continuous integration can roughly be defined as a software development practice that encourages developers to frequently integrate their code, multiple times a day, and running automated tests and builds after code changes. Whereas, Continuous Delivery is simply an indicator of production readiness of our artefact, it's the whole overview of the pipeline that produces feedback for each stage. The piece that really glues everything together and enabled fast release cycles and trunk-based development (described in section 3.2) is Continuous Deployment, which is an automated process that deploys the packaged artefact in the CI stage to the production server (Newman, 2021).

CI/CD became de-facto a standard in the software engineering field and in most of the companies a hard requirement. Since the team has embraced trunk-based development, implementing CI/CD pipeline was a natural decision. The pipeline by itself has been implemented using GitHub Actions. GitHub Actions enables developers to write pipeline definitions using markup language YAML, which are triggered by the set of the defined actuators (events) and picked up as a job by a runner (Virtual Machine) and executed sequentially/parallelly.

For this project the pipeline has been split into 3 stages, namely, backend automated test and artefact build, frontend artefact build and deployment stage. The backend build and test stage, firstly, runs a set of unit tests, then integration tests and, finally, packages it as a Docker image and pushes it to the GitHub Container Registry. The frontend build stage entails a build of the Angular project that is then bundled together into a Docker image together with reverse proxy Nginx and again pushed to the GitHub Container Registry. Lastly, the deployment stage extracts encrypted secrets from the GitHub Secrets store and creates an environment file. The environment file along with the deployment scripts are sent to the staging server via SSH, where the build can be validated by the client and where the team performs User Acceptance Test. Lastly, if everything is in order the same set of scripts are used to deploy the artefacts to the production environment, all without any human interaction.

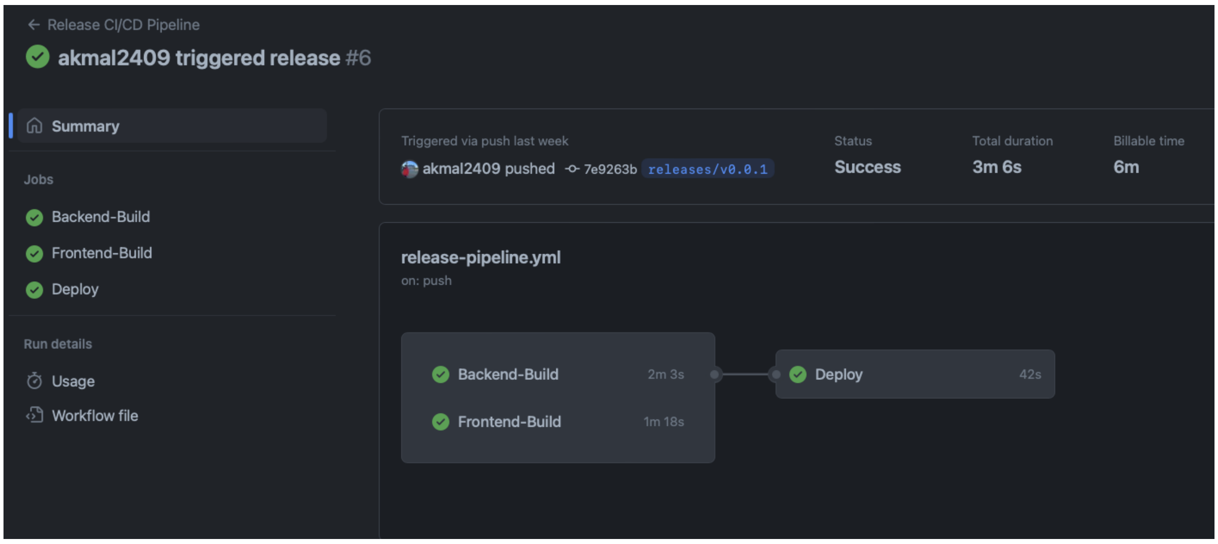


Figure 2: CI/CD pipeline

4. DESIGN

4.1 HIGH-LEVEL DESIGN

4.1.1 Use-case Diagrams

To better understand what each user's expectations are with the system we decided to create a use-case diagram, outlining all the different interactions different users will be making with the underlying system.

In the use case diagram, we have defined 3 types of users as mentioned in the Stakeholder section of the report. We have created one use case diagram per user, highlighting all their use cases:

1. User: These are the people that get access to the system through the Admin. The use cases mentioned here can be found in **Figure 3**.
    a. Logging into the system. This also includes being able to reset their password in case they forget it or want to change it.
    b. Adding a supplier. This also includes functionality like importing a list of suppliers from a CSV file.
    c. Creating Questionnaire Templates.
    d. Sharing a snapshot of a Template with suppliers.
    e. Viewing the answered questionnaires and tracking their status after sharing.
    f. Viewing all the suppliers that they added.

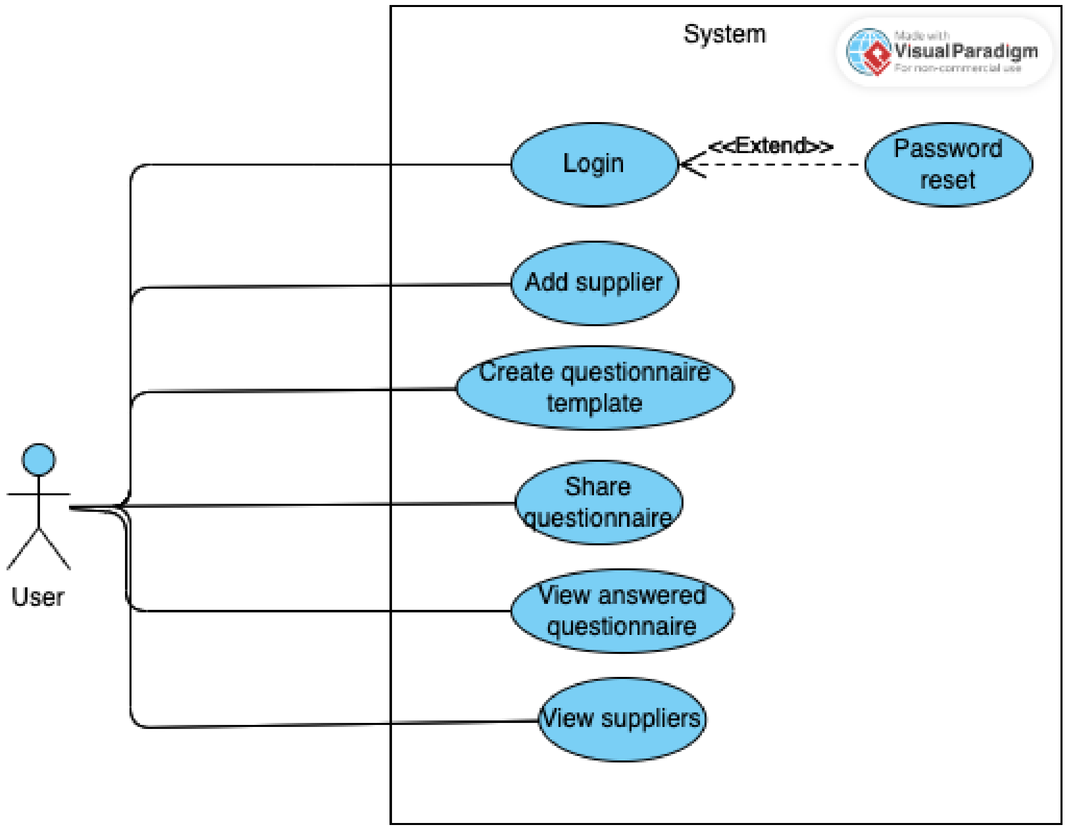

System
Made with Visual Paradigm For non-commercial use
Login
<<Extend>>
Password reset
Add supplier
Create questionnaire template
Share questionnaire
View answered questionnaire
View suppliers
User

Figure 3: Use case diagram for user

2. Admin: They are users from FoodChainID that manage the system. The use cases described here can be found in **Figure 4**.
    a. Logging into the system. This also includes being able to reset their password in case they forget it or want to change it.
    b. Adding and inviting Users to create an account.
    c. Adding Suppliers. As an admin is basically an user with extended privileges, they can also use the system as a normal user would.
    d. Creating Questionnaire Templates.
    e. Sharing a snapshot of a Template with suppliers.
    f. Creating API keys that allow for creation of
    g. Creation of Questionnaires through API keys.
    h. View Analytics of Users and System.

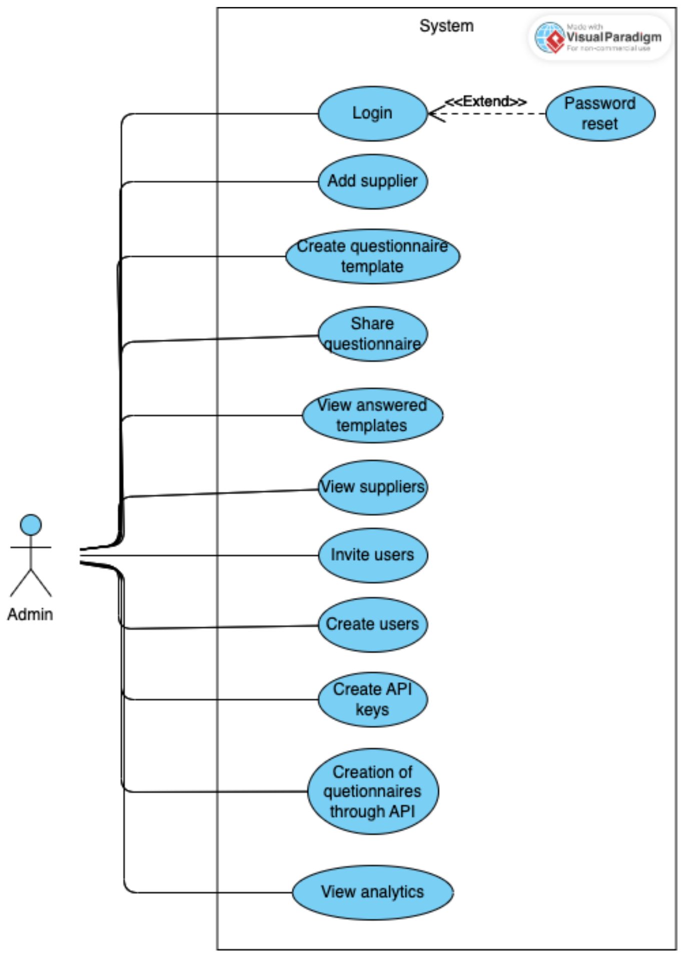


Figure 4: Use case diagram for admin

3. Supplier: These are the users that get invited to answer the questionnaires created by admins and users above. Their use cases are defined in **Figure 5**.
    a. Answer the questionnaire, he/she is invited to answer. (Includes the system notifying the supplier about answering a questionnaire)
    b. Submit a questionnaire when it's completed. (Includes the system notifying the user who shared the questionnaire, about its completion).

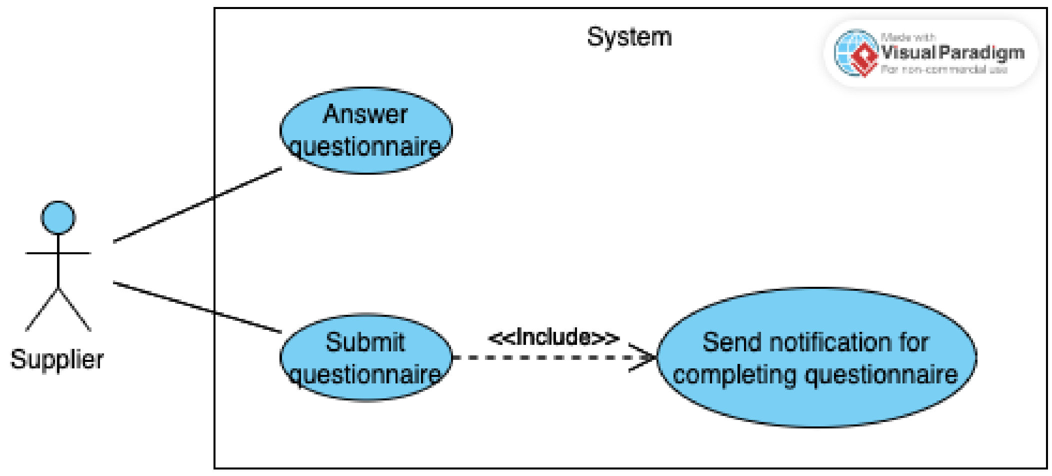

System
Made with Visual Paradigm For non-commercial use
Answer questionnaire
Supplier
Submit questionnaire
<<Include>>
Send notification for completing questionnaire

Figure 5: Use case diagram for supplier

4.1.2 Swimlane/Activity Diagram

We created UML activity diagrams, showcasing the different activities all user roles make and the systems response to those activities. The different activities and their diagrams are as follows:

Logging In

A flow starting from the user navigating to the website. The login page will be the first page that they will see. If the user has an account, they can enter their credentials and then the system validates this data. If it is valid, they will be redirected to the Dashboard page. If the credentials are not valid, the system will throw an error message and the user can re-try to login. In case the user does not remember the password, they are able to reset their password.

If there is a new user, they can fill the sign-up form with the invitation code that they got as email. Once they get registered into the system, they follow the process of logging in.

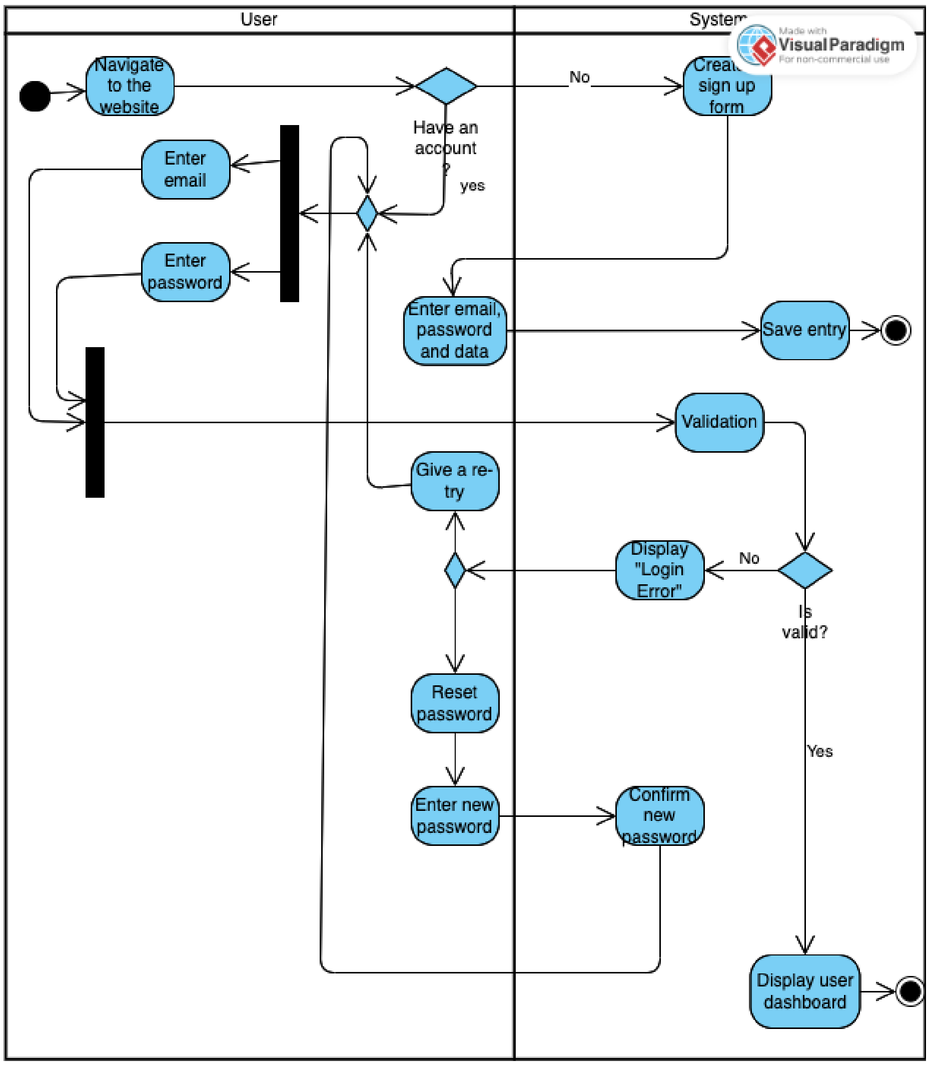


Figure 6: Activity diagram for logging in

Sharing Questionnaires with Suppliers

Once the questionnaires have been created, the user opens the template and uses the share button to share the template with the suppliers. In that dialog, users can either choose suppliers from the existing suppliers or add/create new suppliers. Once the suppliers have been selected, the system delivers the questionnaires to them.

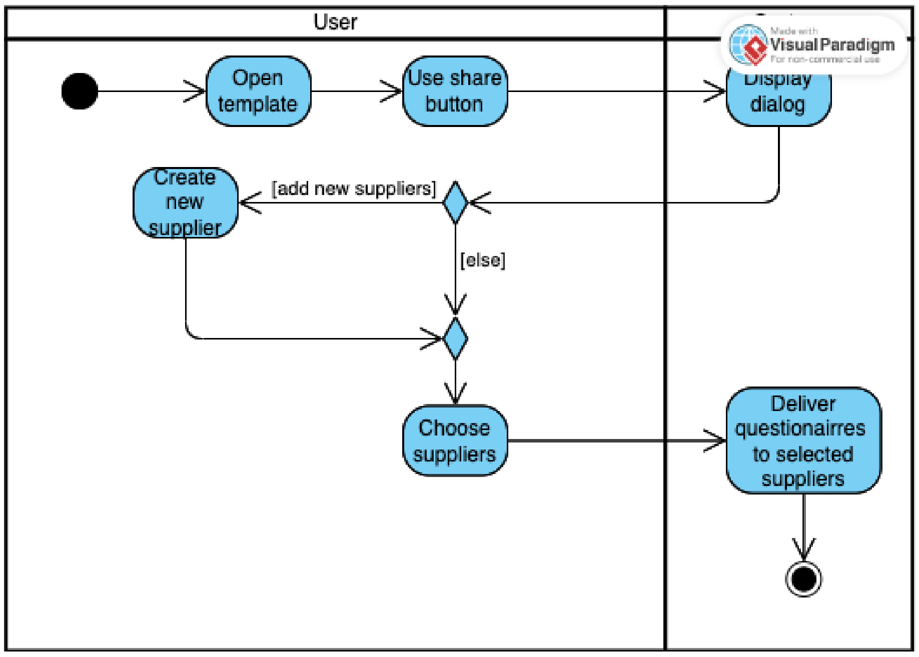


Figure 7: Activity diagram for sharing questionnaires with suppliers

Creation of Custom Template

To be able to create a questionnaire template, users need to use the “New Template” button. The system will open the page where the user can fill in the details of the template. In that page, users can either create a stage or import an existing stage. Once they import the stages, they need to assign points for each option and add questions within the stage. Before they upload and save the stage, they can edit it and delete or update some of the questions if needed.

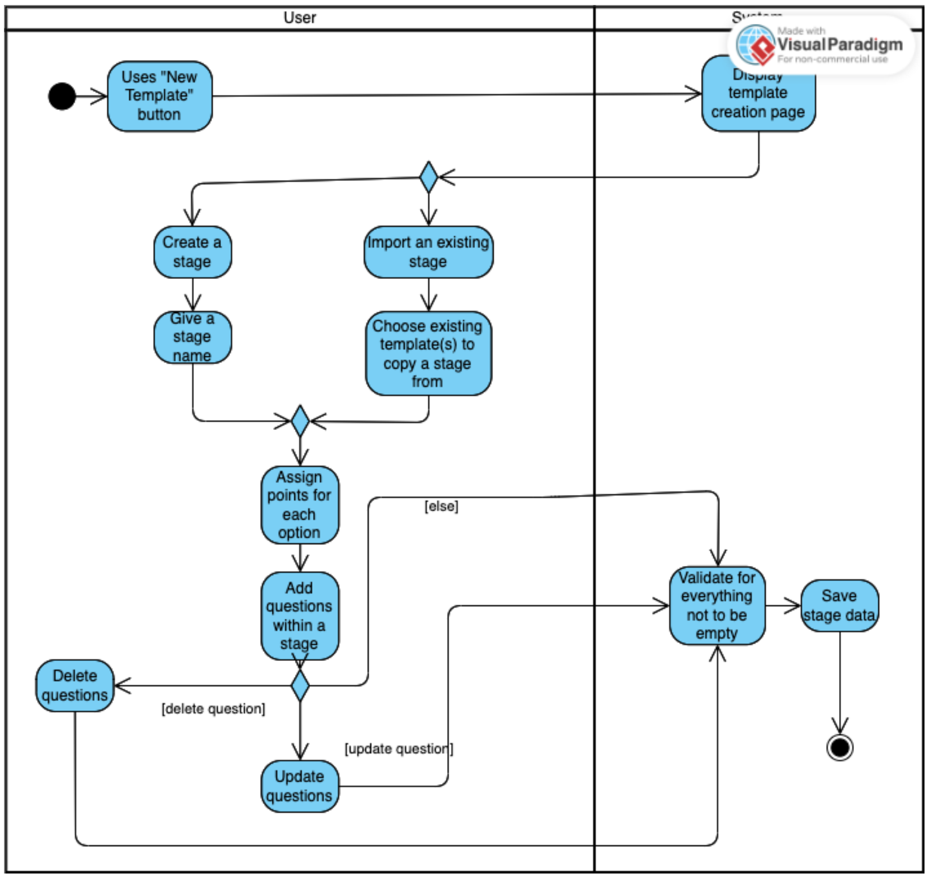

User
System
Uses "New Template" button
Display template creation page
Create a stage
Import an existing stage
Give a stage name
Choose existing template(s) to copy a stage from
Assign points for each option
Add questions within a stage
[else]
Validate for everything not to be empty
Save stage data
Delete questions
[delete question]
[update question]
Update questions

Figure 8: Activity diagram for creation of custom template

Managing the Suppliers

Users can edit and manage their suppliers. They need to visit the supplier’s page. In that page, there are options to create, delete, view, and update the suppliers. If they want to create suppliers, they need to trigger the “+” icon and within the following page, they enter the supplier details such as salutation, name, organization name and email address. The updates that they have made are then going to be saved in the system.

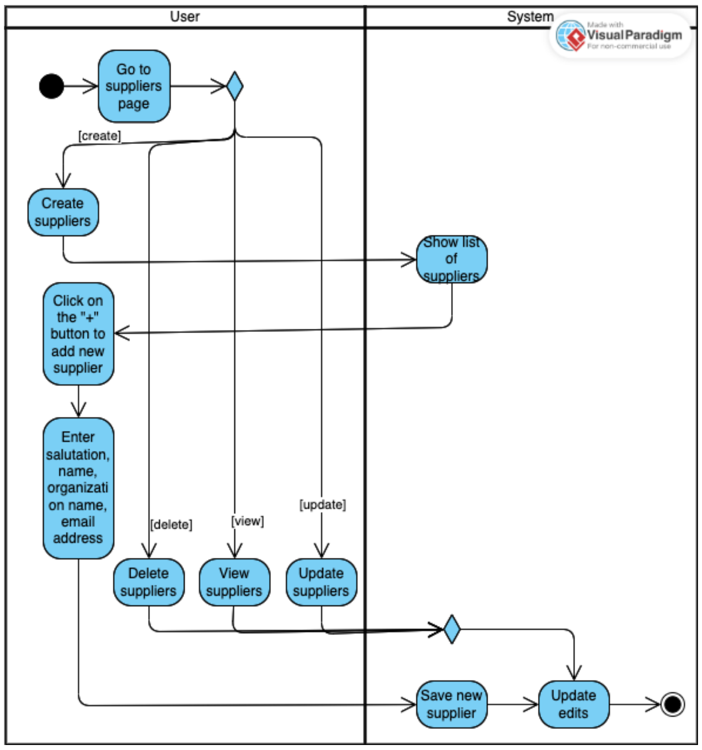


Figure 9: Activity diagram for managing suppliers.

Answering a Questionnaire

To be able to answer a questionnaire, a supplier is supposed to receive an invite. This invite will forward them to the questionnaire where they can answer. A questionnaire can only be submitted if all the questions are answered. Otherwise, the system sends a reminder message to the supplier to answer all the questions.

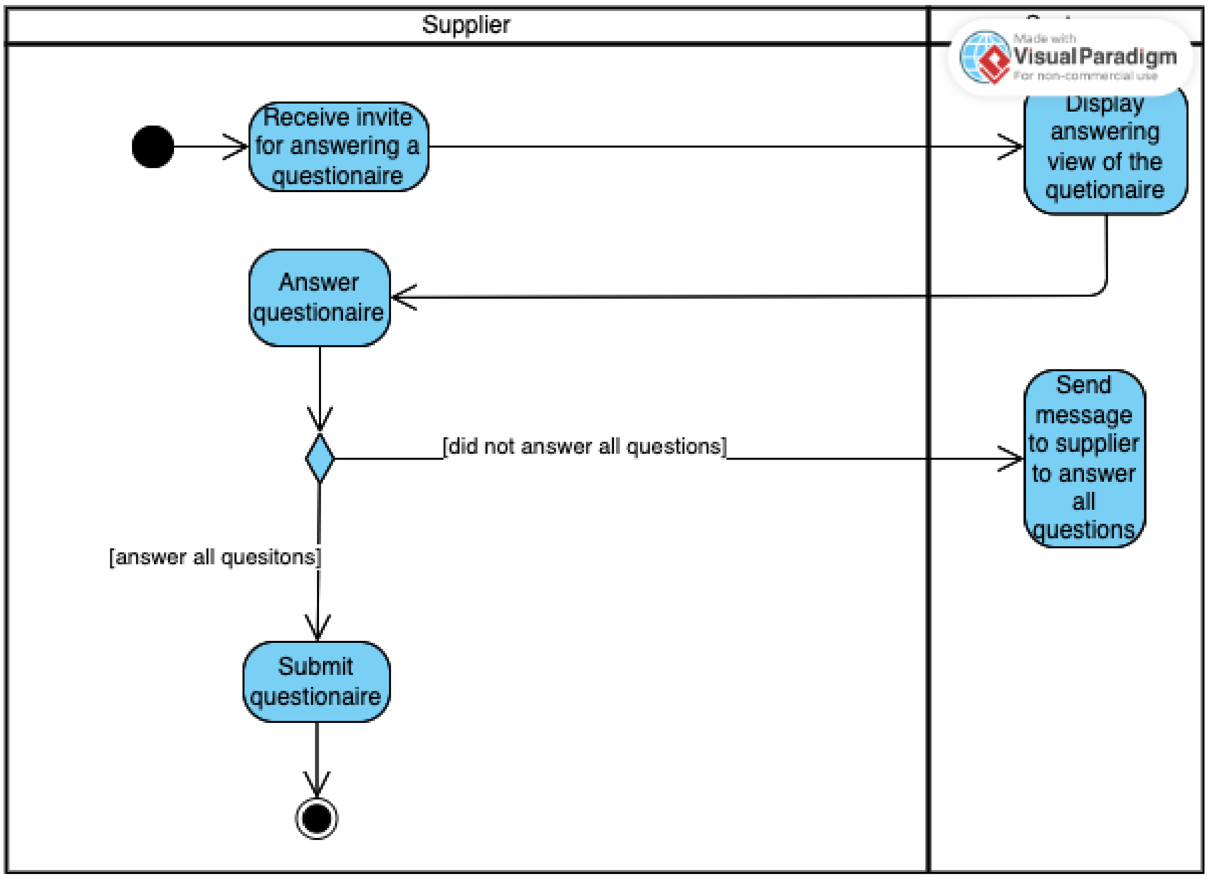

Supplier
Made with Visual Paradigm For non-commercial use
Receive invite for answering a questionaire
Display answering view of the quetionaire
Answer questionaire
[did not answer all questions]
Send message to supplier to answer all questions
[answer all quesitons]
Submit questionaire

Figure 10: Activity diagram for answering a questionnaire.

Inviting Users

The process of inviting the users is similar to adding suppliers. However, only administrators can invite users. They first need to navigate to the “Users” page. The list of existing users is shown. This page can be used for viewing the existing users or inviting new users. Invite users button supposed to be clicked. Then, the admin can enter the user email addresses that they want to add. Once all the email addresses are filled, pressing the “Invite” button sends invitations to those addresses.

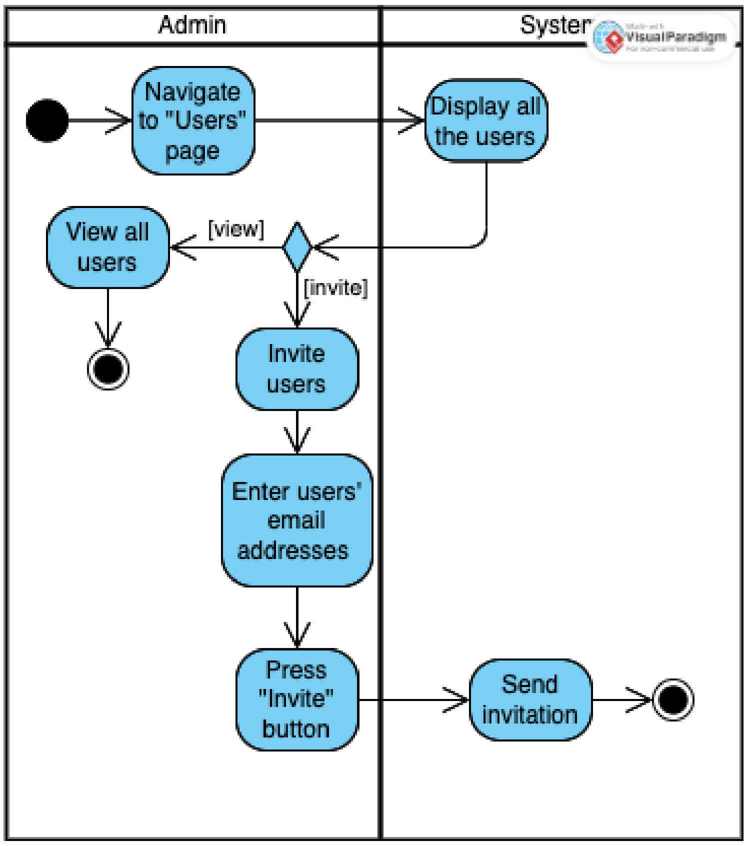


Figure 11: Activity diagram for inviting users.

4.2 BACKEND DESIGN

4.2.1 Database Schema

As we chose to go with a relational database, the design of the schema was very important to the success of the project. Which is why we chose to map out all our entities, and the relationships between them, before moving on towards the development of the server. This was done using an Entity Relationship Diagram, which is shown in Figure 12.

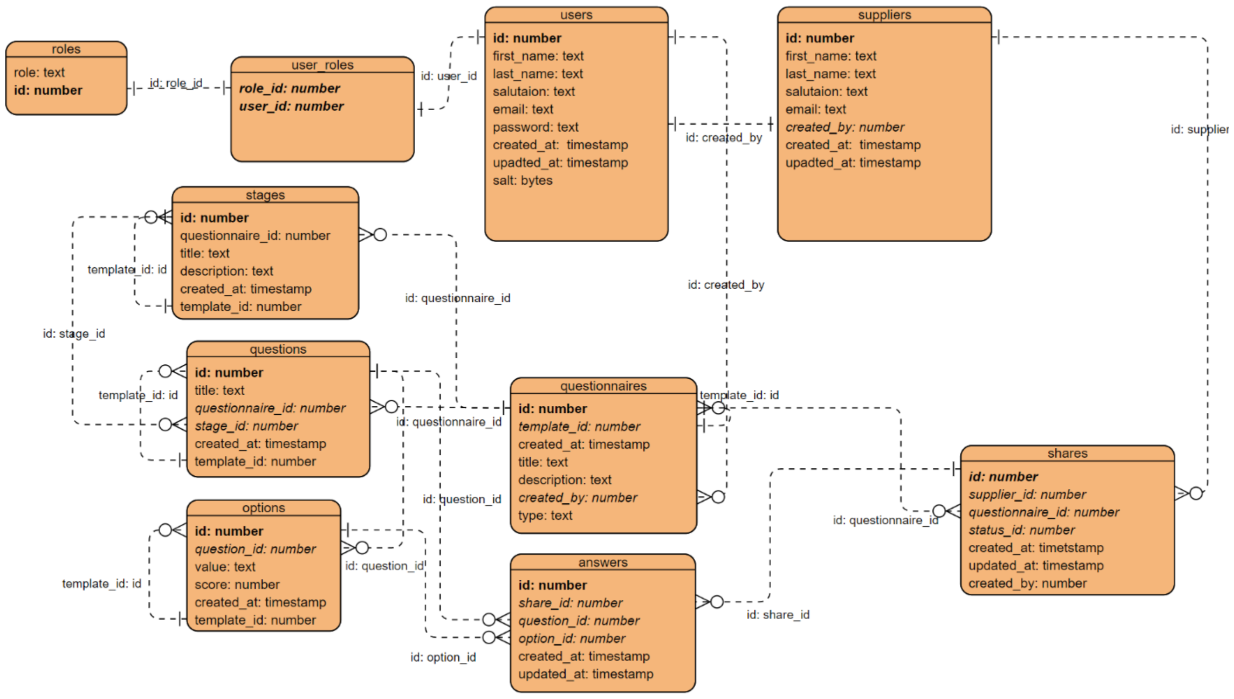


Figure 12: Entity Relation Diagram

Before we dive into the different relationships shown in the ERD, we must first understand the reasoning behind certain key aspects of our design. These are as follows:

1. A Questionnaire can have two types, one is “Template” and the other is a “Snapshot”. The purpose of these types is to define the use of the questionnaire.
    a. A Template Questionnaire is the one that a User interacts with, adding questions, stages, basically, it denotes the design of a questionnaire.
    b. A Snapshot Questionnaire is a copy of a Template Questionnaire shared with suppliers. When a user decides to share a Template, he/she can do so for multiple suppliers, creating a Share entity per supplier. However, all the Share entities created point to a single snapshot questionnaire, which is a copy of the template, with its template_id pointing to the id of the Template Questionnaire.
    The reason for this approach is that it allows us to decrease the number of duplicates in the database, As once a template is shared, nothing about it can be changed, we keep one copy of the template for all the suppliers it was shared with, allowing the user to still make changes to the template and reshare it at a later instance if he/she wants to.

2. To facilitate the formation of Snapshot Questionnaires and always allow the user to know the source of each Questionnaire, Stage, Question and option, each of these entities reference to their Template counterparts.

The main relationships in this diagram are as follows:

1. User to Supplier: A user can create many suppliers, but a supplier can only be created by one user, therefore, there is a Many-To-One relationship between supplier and user, denoted by the “id: created_by” (User → Supplier) link.
2. User to Questionnaire: A user can create many questionnaires, but a questionnaire can only be created by one user, therefore, there is a Many-to-One relationship between questionnaire and user denoted by “id: created_by” (User → Questionnaire) link.
3. Questionnaire to Stage: As a questionnaire can have multiple stages, there is a One-to-Many relationship between a questionnaire and stage, denoted by the link “id: questionnaire_id” (Questionnaire → Stage).
4. Questionnaire to Question: A Questionnaire is composed of multiple questions therefore, there is One-to-Many relationship between Questionnaire and Question, denoted by “id → questionnaire_id” (Questionnaire to Question).
5. Stage to Question: A Stage is nothing but a group of questions, therefore there is a One-to-Many relationship between a Stage and Questions, denoted by the relationship “id: stage_id” (Stage → Question)
6. Question to Option: A Question can have multiple options for the user to select from, therefore there is a One-to-Many relationship between a Question and Options, denoted by the relationship “id: question_id” (Question → Option)
7. Questionnaire to Share: A Share for us, is a request that a User makes, to share a Snapshot of a Questionnaire Template, to be answered by Suppliers. As the User can request to share a Questionnaire with multiple Suppliers, there is a Many-to-One relationship between a Share and a Questionnaire, this is denoted by the relation “id: questionnaire_id” (Questionnaire → Share)
8. Share to Supplier: A Share is meant for exactly one Supplier, defining the questionnaire that the supplier needs to answer, and the status of this answering. As multiple questionnaires can be shared with the same supplier, there is a Many-to-One relationship between Share and Supplier, denoted by “id: supplier_id” (Supplier → Share).
9. Share to Answer: An answer refers to a response by a supplier to a question in a questionnaire shared with them. As a Share is related to a single Questionnaire, but a Questionnaire can have multiple Questions, there is a One-to-Many relationship between a Share and Answer, denoted by “id: share_id” (Share → Answer).
10. Answer to Question: As mentioned above, the same snapshot of a questionnaire can be shared with multiple Suppliers, Therefore, multiple

answers with different share_id’s can point to the same Question. Resulting in a Many-to-One relationship between Answer and Question, denoted by “id:question_id” (Question → Answer).

This Schema was finalized after careful consideration from each person involved in the process. Our goal was to minimize the duplication of data, which is why we made certain design choices.

As our data requirements had a lot of child entities, we chose to go with a relational schema and a relational Database, in our case PostgreSQL.

4.2.2 Class Diagram

Based on our Entity Relationship Diagram, shown in Figure 13, we created our class diagram. In our design of the class diagram, we decided to maintain Bi-directional mapping between all our classes. This was required by Hibernate, our choice of an ORM, for easier mapping of our models to the database.

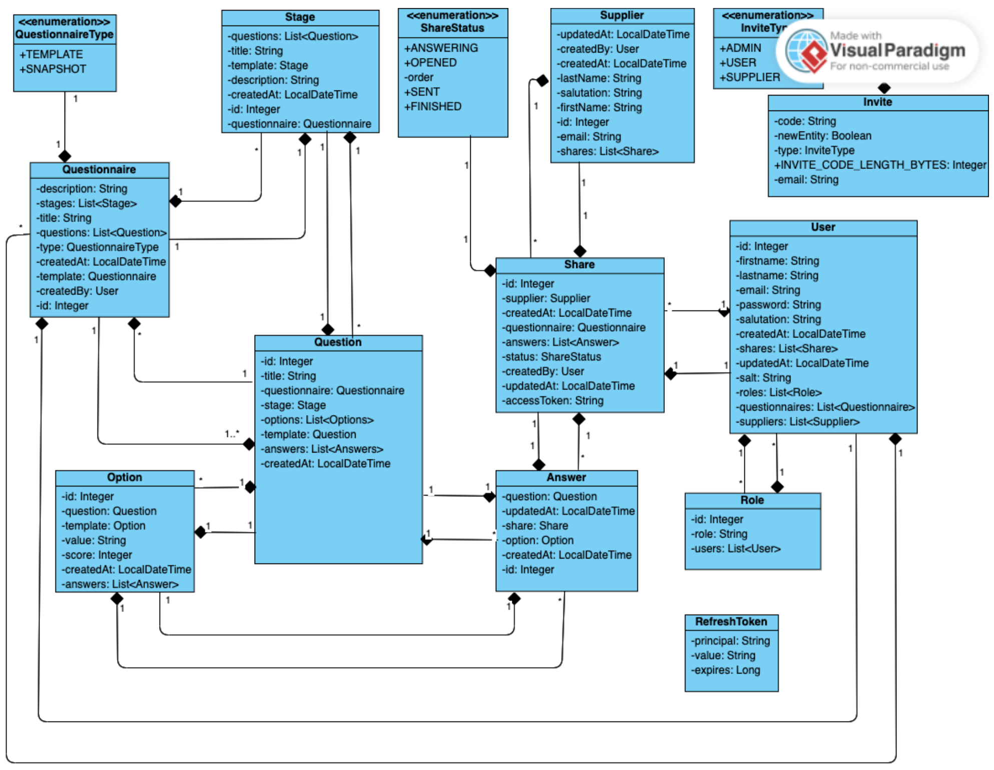


Figure 13: Class Diagram

4.2.3 Authentication/Security

Security is a cross-cutting concern in any application regardless of whether it's critical or not. Therefore, it must be tackled with utmost care and its design must satisfy the strict heuristics. Nowadays, there are numerous ways of implementing authentication.

That includes using specialized identity providers like Okta, Auth0, Keycloak, AWS Cognito and others, as well as implementing your own.

To make the best choice and tradeoffs we came up with the following requirements for our authentication system. It must be (1) secure, (2) modern, (3) scalable, (4) stateless and (5) have existing rich tooling.

For any monolithic backend application often, a natural choice is a session-based authentication. It is simple to implement and there is a lot of tooling supporting that in the Spring Framework ecosystem. However, session-based authentication does not meet requirement 1 (secure)., long lived sessions in case of session hijacking can cause significant harm to the application. Secondly, session-based authentication is not scalable. In the event when the organization decides to scale the application to multiple instances, they will have to implement some sort of shared session store, which would introduce additional database calls to each user request. Lastly, having sessions would mean that the application is not anymore stateless, which introduces potential problems to scalability.

That being said, we started exploring the possibility to outsource the authentication to a 3rd party Identity Provider like Okta. However, after discussing with the client, they have told us that they are currently in the process of implementing an SSO solution, but it is not available to us at the moment of development.

Therefore, we decided to explore lightweight JWT tokens as the means of passing the user authentication state between the requests. It seemed like a natural choice because they are secure if they are short-lived. In case the token is intercepted by a malicious party, the lifetime of the token is usually minimal and therefore, the attacker will not be able to cause significant damage. However, using raw JWT tokens hinders the UX of the application because it means that the user will have to sign in every 2-5 minutes into the application, which is not acceptable. That is why we have introduced the concept of the refresh tokens that have generally a longer validity and upon JWT token expiry, the refresh token can be exchanged for a new JWT access token. This solution is modern (2) and employed by all major applications. It is also scalable (3) because the tokens are encoded as a Base64 string and are passed by the client on each and every request meaning that they do not need to be stored and because they are cryptographically signed, the application can be certain that the contents of the JWT were not tempered with. Lastly, JWTs have great tooling both in the Spring Framework ecosystem as well as in the Java community, therefore, its implementation is trivial.

Once the authentication scheme had been designed, the question of the means of obtaining the JWT access token were still unanswered. Our team has decided to choose email and password-based authentication, where the user presents his/her credentials and if they match, the server responds with an access token and optionally with a refresh token. The passwords, however, must be stored hashed and salted to prevent brute force rainbow table attacks in the event of a database breach.

Last piece of the puzzle, however, was designing the authentication flow for the suppliers that are invited to answer the questionnaire and lifecycle management of their credentials. Of course, we could enforce the registration of the suppliers as regular users on the platform before being able to answer the questionnaire. However, that might repel some and prevent them from even approaching the questionnaire, harming the overall UX. That is why we have opted for generating an access code for each questionnaire that supplier was shared with and sending it via email embedded in the link. The only thing that the supplier has to do afterwards is to click on the link, get signed-in to the questionnaire, answer it and then his/her credentials will be revoked.

4.3 FRONTEND DESIGN

To understand how the web application should look like, and what features each page should have, we decided to design Wireframes. The purpose of the wireframes was to be a guide for each team member once we started development.

Below we have all the wireframes, with a description of the functionalities each wireframe encompasses.

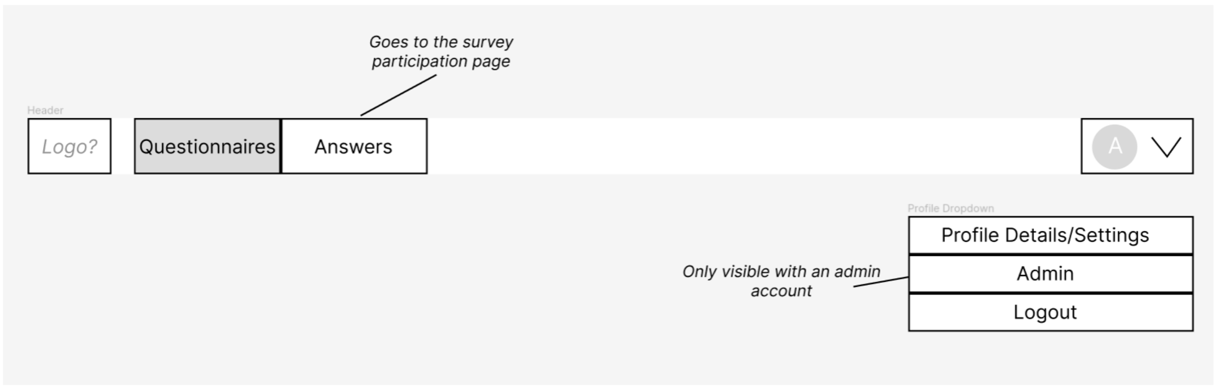


Figure 14: Initial Wireframe Header

The main application header on Figure 14 was originally designed to be a horizontal bar at the top of the screen. The main navigation buttons on the bar would always be visible and would take the user to the most frequently used sections of the application. Features used less frequently were designed to be hidden in the dropdown that appears when the user profile icon is clicked. For the actual application, the header was moved to the side of the screen to preserve vertical space on desktop monitors.

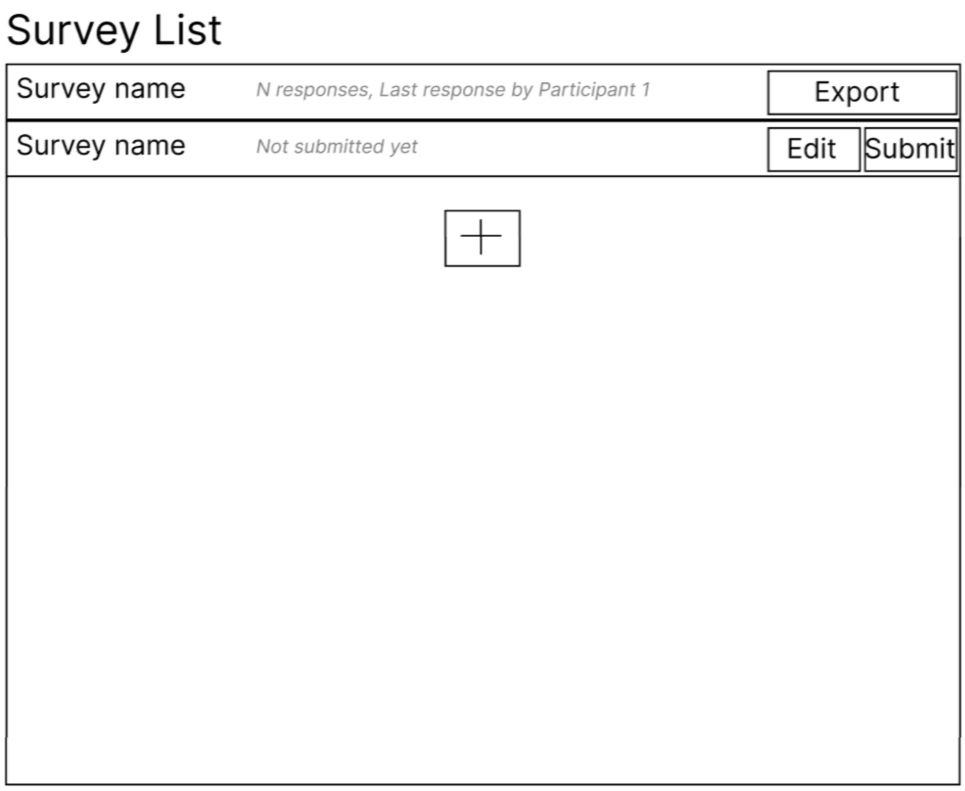


Figure 15: Questionnaire list wireframe

Figure 15 shows the wireframe for the questionnaire list page that user would see. The original wireframe design evidently did not make a distinction between templates and snapshots, and assumed that questionnaire templates irreversibly become snapshots, so both types are present in a single list. The questionnaire snapshot on this figure is displayed first, with additional response information in italics and an option to export response information to a spreadsheet format such as CSV. The template is displayed second, with an option to edit the template and turn it into a snapshot by submitting it. Lastly, a large button with a plus sign is always displayed at the bottom of the list to give the user the ability to design more questionnaire templates.

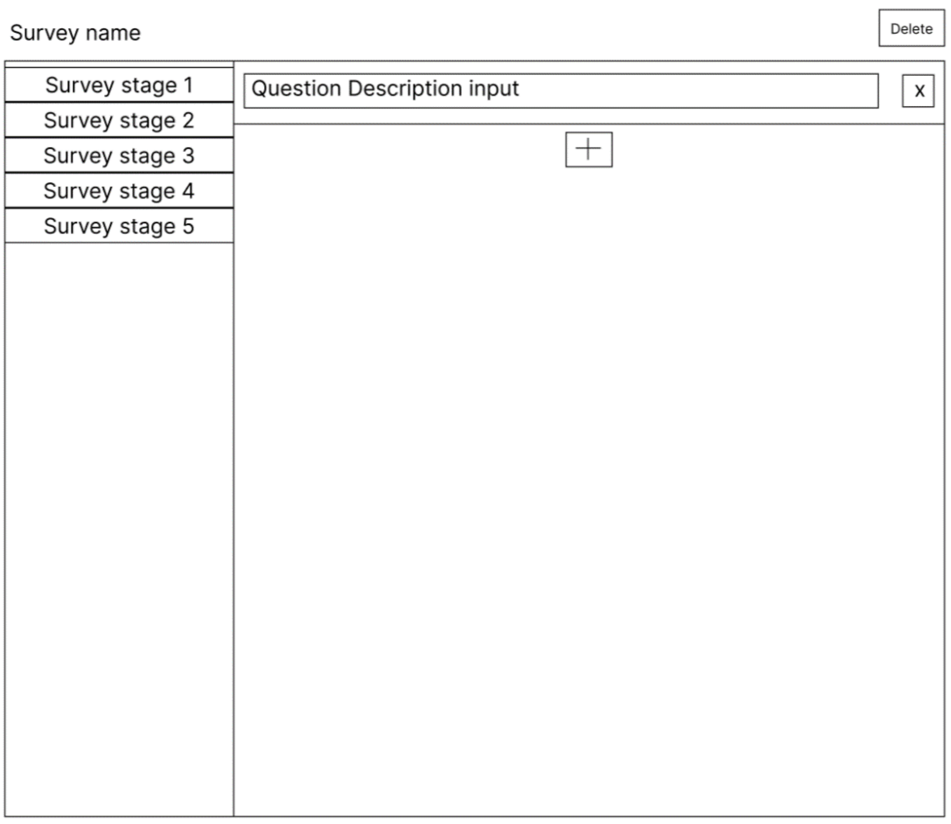


Figure 16: Template builder wireframe

The wireframe in Figure 16 displays the template builder that a user would utilize to create questionnaire templates. The stages are displayed on the left, akin to the final design used in the actual application, but the question list assumed that all questions are Likert scale questions, where a participant is presented the same predefined numerical scale. As such, the original wireframe design did not provide UI elements for creating an answer list.

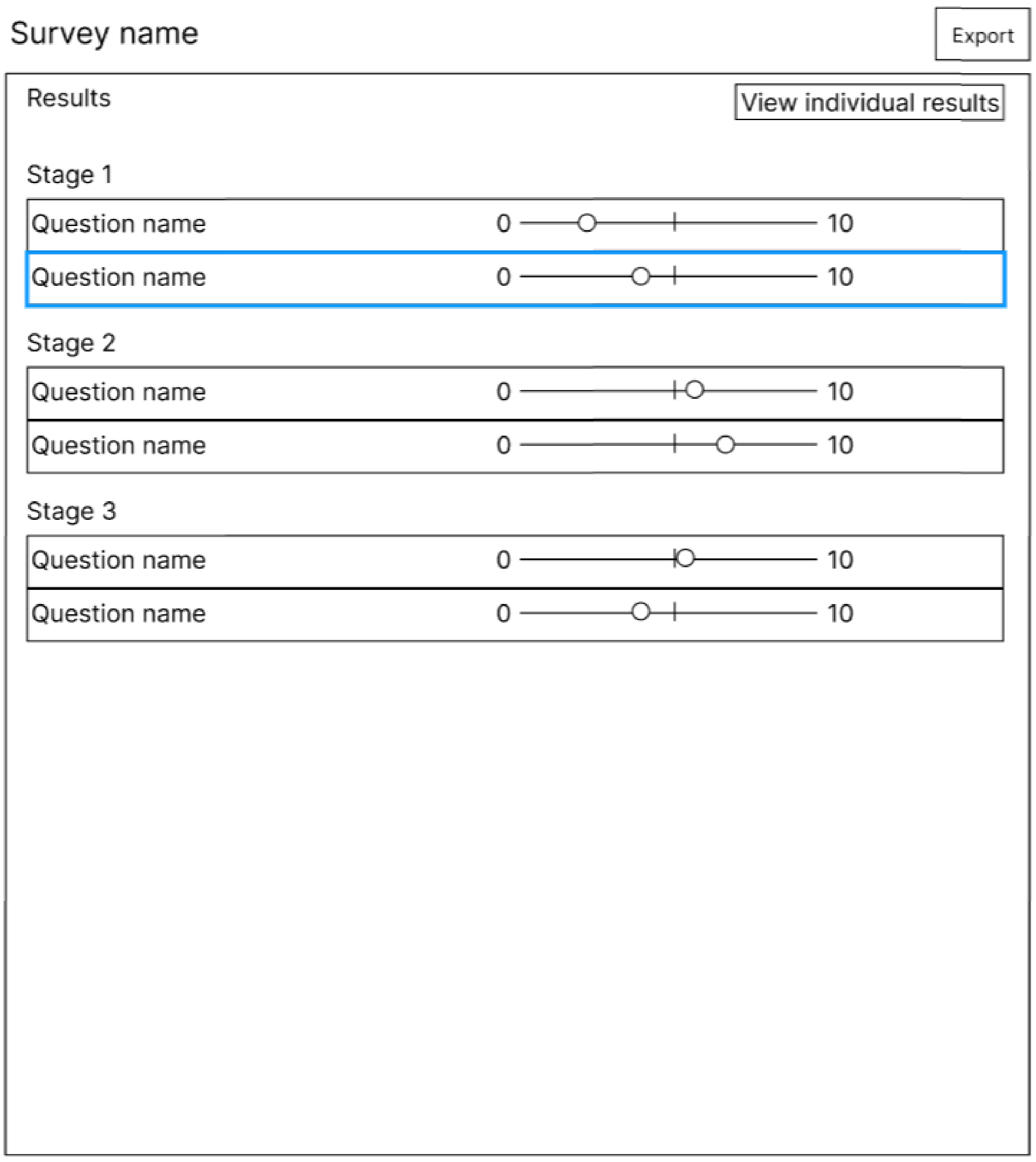


Figure 17: Survey Result View Wireframe

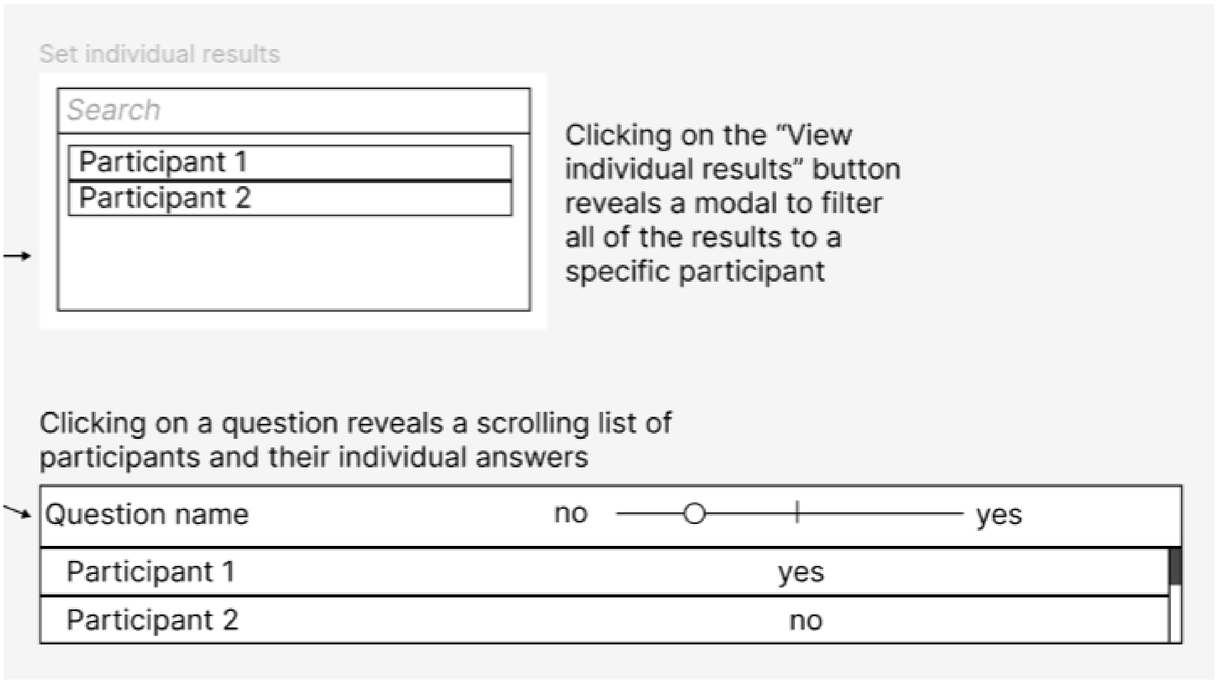

Set individual results
Search
Participant 1
Participant 2
Clicking on the "View individual results" button reveals a modal to filter all of the results to a specific participant
Clicking on a question reveals a scrolling list of participants and their individual answers
Question name
no
yes
Participant 1
yes
Participant 2
no

Figure 18: Individual Result Views

Clicking on a submitted questionnaire in Figure 15 directs a user to the questionnaire results page, shown in Figure 17. On this page the user is mainly presented with the average score of all participants per question. Clicking on a question reveals a submenu below with individual participant answers. Clicking on a "View Individual Results" button reveals a modal window, where a user can select an individual participant, after which all the questions on the main view display the individual answers of the selected participant. This design was significantly modified in favor of a more informative "Insights" view in the final application.

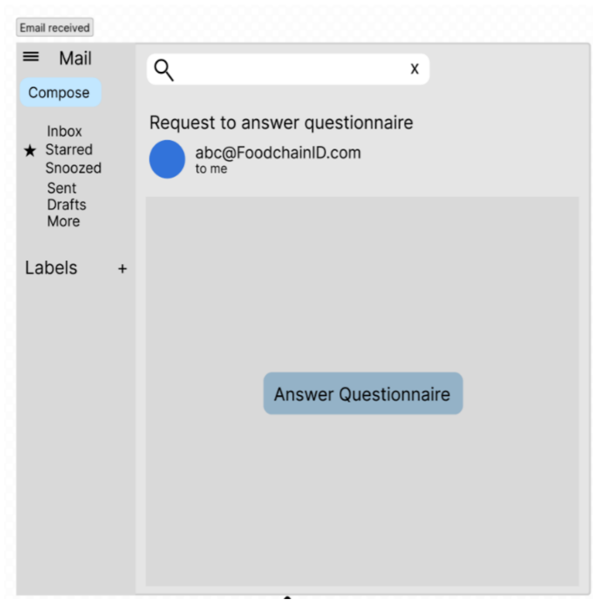


Figure 18: Email Participation Invitation Wireframe

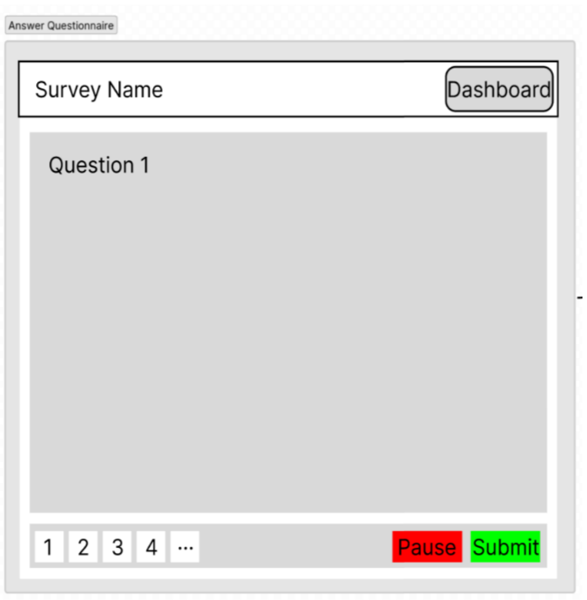


Figure 19: Survey Participation

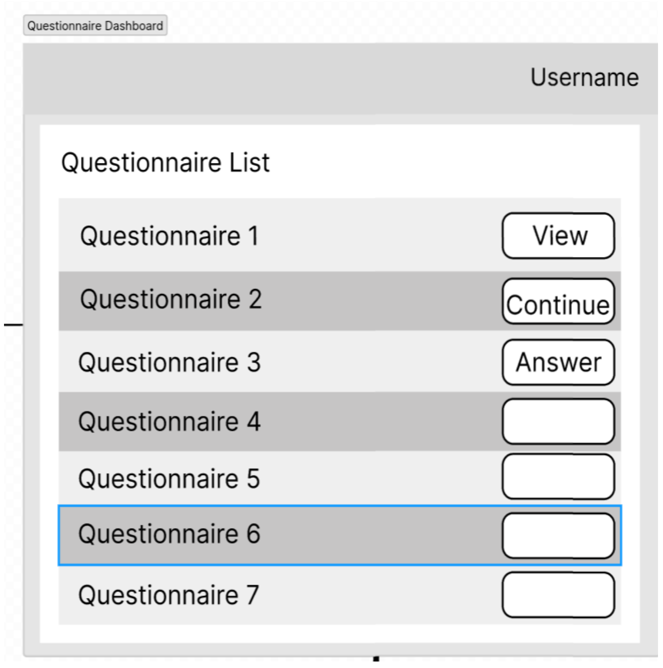


Figure 20: Supplier Questionnaire List

Figures 18-20 show a flow for the questionnaire participant. The participant is first notified of the ongoing questionnaire by email, shown in Figure 18, in which a link can be clicked to be redirected to the application to answer the questionnaire, shown in Figure 19. Finally, the participant is presented with a list of all questionnaires to which the supplier is invited to, as seen in Figure 20. This list is not implemented in the final application.

5. IMPLEMENTATION

5.1 TOOLS AND FRAMEWORKS

As our project was made in collaboration with an external company, we were required to work with their existing stack, allowing FoodChainID to continue development of the project.

Therefore, we designed and implemented the Backend using the Spring framework, with Java as the Programming language. Spring brings with it a lot of functionality like its own testing framework, support for Hibernate as the ORM and is also developed with the MVC framework in mind, allowing for faster development.

For our database, we chose to go with PostgreSQL, as it is a relational database, which is what we needed for our requirements. The reason we went with a relational database is because of identified relations between Questionnaires, Stages/Groups, Questions, and options. Furthermore, we planned on reducing data duplication by relating multiple answers from different suppliers to a single questionnaire, all of which led us to conclude that a relational database would be the best for our use case.

Lastly, we went with the Angular Framework with TypeScript for our front-end development. Again, this was something the company is using and therefore, using it would maintain consistency with their current stack.

5.2 SECURITY

Implementation of the authentication and authorisation was a trivial task because the initial designs were well thought and properly evaluated. Since we are using Spring Framework for building our application, we have extensively leveraged the Spring Security project that provides convenient code libraries to manage the application and reduce security risks. However, the standard authentication schemes provided by the Spring Security were not sufficient and secure for us. Furthermore, they did not satisfy all five requirements developed in the design stage ((1) secure, (2) modern, (3) scalable, (4) stateless and (5) have existing rich tooling). That is why the application's security modules were extended with a lot of custom code to support our requirements.

Firstly, to separate the concerns and provide a set of abstractions our team has opted to use HTTP request filters for intercepting the calls and validating the credentials, global Authentication Manager that selects appropriate authentication scheme based on the input and a set of Authentication Providers that implement authentication logic. Therefore, the client does not have to have any knowledge of which Authentication Provider to select, since Authentication Manager does that already.

Secondly, using the building blocks outlined above, we have implemented the user account creation. For the password hashing we have opted to use the BCrypt algorithm with the implementation provided by the Spring Security module.

Thirdly, for the user login we have implemented the email and password-based authentication, which is handled by the UsernamePasswordAuthenticationProvider, which validates the credentials and upon successful validation issues the JWT token and a refresh token.

As of the JWTs, we have decided to use a mature library that is recommended by the security experts, namely, Nimbus JOSE. To sign the tokens, we have tried to use asymmetric RSA key pairs of sizes 2048- and 4096-bits, however, we have noticed a serious performance penalty during benchmarking the 4096-bit long key, that is why we have settled on the 2048-bit key size. The JWT is signed using the private key and is validated using the public key. The advantage of this approach is that if the application needs to be ever scaled to multiple instances, then each instance can have a public key and be able to validate the token issued at another instance.

The main issue in exposing the JWT tokens to the clients, however, means that we grant the access credentials to the insecure browsers, where a 3rd party JavaScript code can easily hijack both access and refresh tokens. To mitigate that risk, we have decided to place both JWT and refresh token in 2 separate secure, HTTP only and same-site cookies. Meaning that the cookies are not accessible by client-side code, are only transmitted via an HTTPS secure channel and are only available under the same domain (first party cookies). This approach of sharing the credentials is recommended by the OWASP JWT cheat sheet and is the most common approach nowadays.

However, using cookies to store JWT and refresh tokens of course reveals another vector of attack such as Cross-Site Request Forgery. To mitigate that we have used Spring Security's module to return CSRF tokens to the client, which can be then appended to every modification HTTP request.

The JWT tokens were set to live only for 2 minutes, meaning that we had to develop client-side code to refresh the access tokens. The choice was natural, we have used HTTP interceptors provided by the Angular Framework and on every HTTP 401 (Unauthorised) error we try to refresh the user's credentials.

Moreover, the JWT token is validated by the JWT filter that sits in a general SecurityFilterChain component that proxies each request for the protected routes. That means that no protected route is left without authorization and the security measures are automatically applied.

Lastly, due to security being a very important and crucial part of the application we have written an extensive set of both unit and integration tests, testing the possible issues and attacks.

5.3 BACKEND APPLICATION

The backend application is built as a RESTful service following the 3-tier architecture. The 3-tier architecture helps the developers to make the code more manageable and testable by separating presentation, business logic and data access layers. There are several benefits in using the 3-tier architecture such as faster development, improved scalability, reliability, and security. (IBM, 2022)

In our case all the data access logic was encapsulated in repository objects that act as a proxy between the database and the application data model. Exposing clean APIs of repositories helped the team to create a mental model of the database entities and significantly reduced object-relational impedance mismatch.

Secondly, all the business logic such as validation, state transitions and lifecycle management of the domain entities was placed in a set of services that do not leak any implementation detail to the client and provide a rich API.

Thirdly, for the presentation layer we have structured it following the MVC pattern by implementing a set of controllers that are serving JSON data structures and accept commands to update the model.

Our goal was for all API endpoints to be structured with a strict adherence to the RESTful principles and to achieve the second level of Richardson Maturity Model. Each resource has a set of respective actions that can be invoked by using appropriate HTTP verbs and can be explored through the automatically generated OpenAPIv3 schema.

5.4 FRONTEND APPLICATION

As a first step of the website design, we have made the design of our application using Figma. All the wireframes can be found in Section 4.3. This has given basic understanding of how the application is supposed to look for both user, supplier, and admin. The implementation, the final look of the application can be found under each subtitle with the screen captures.

5.4.1 Login/ Sign-Up Page

Login page is the first interaction that the users will be making with the application. This allows users to authenticate themselves for getting access to the application. The main objective of this page is providing a secure gateway for the authenticated users.

New users can sign up if they have an invite code. If they are not registered users, “Sign Up with Invite Code” button can be clicked and users are supposed to enter their email address, first name, last name, password, and their invite code.

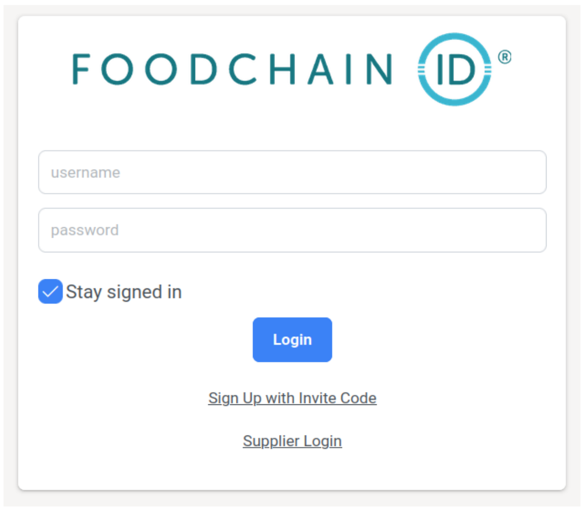


Figure 21: Login panel

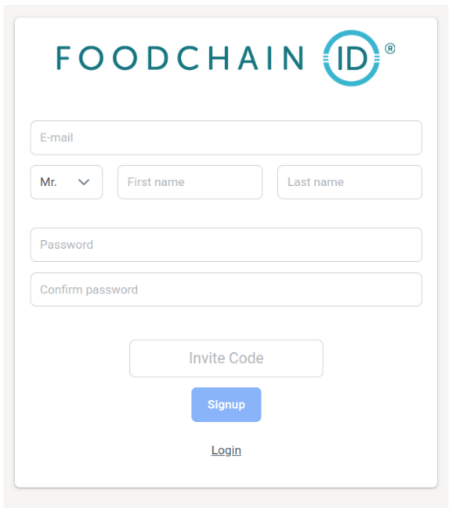
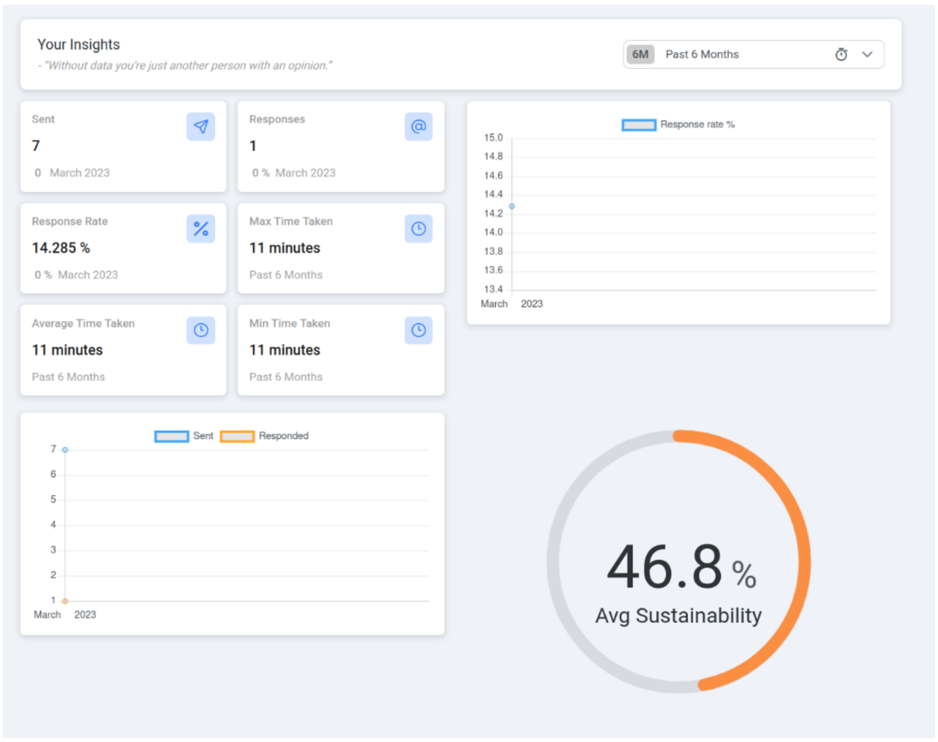


Figure 22: Signup panel

5.4.2 Dashboard Page


Your Insights
- "Without data you're just another person with an opinion."

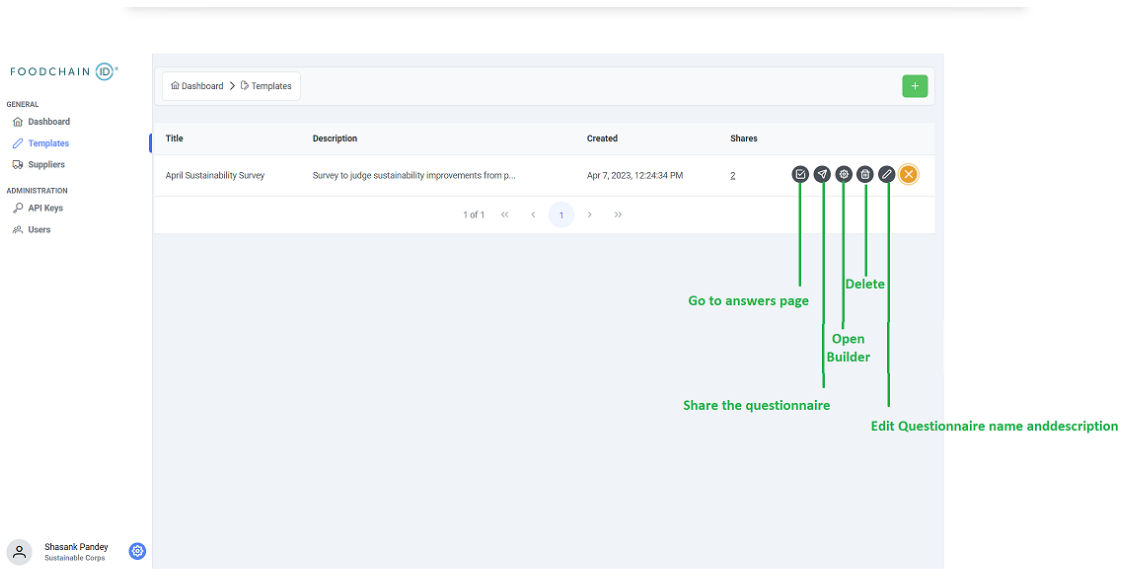
6M Past 6 Months

Sent	7	0 March 2023
Responses	1	0 % March 2023
Response Rate	14.285 %	0 % March 2023
Max Time Taken	11 minutes	Past 6 Months
Average Time Taken	11 minutes	Past 6 Months
Min Time Taken	11 minutes	Past 6 Months

Figure 23: User Dashboard/Home Page

In the home page, users can see the menu on the left side of the screen where they can redirect themselves to use the features of the application. For the admin, the dashboard page shows the analytic data of the questionnaires that have been filled by the suppliers. Administrators can see the number of templates they have sent and responses they got. Besides that, response rate by time and Sent-Responded by time graphs can be analysed.

5.4.3 Template Page



Title	Description	Created	Shares
April Sustainability Survey	Survey to judge sustainability improvements from p...	Apr 7, 2023, 12:24:34 PM	2

Figure 24: Template Page

In the template page, admin and user roles can view the existing questionnaire templates in the list view. Each template is shown with its title, description the date and time that it's been created and the number of times it has been shared.

There is a green button with the plus icon on the top right of the page. This enables users and admin to create questionnaire templates. Template title and description must be given when it's being created. Otherwise, an error message is displayed to alert the user.

Lastly there is a button on each questionnaire, which shows all available actions for that template, such as, going to the answers page (shown in section 5.4.9)and sharing the Questionnaire (shown in section 5.4.5), opening the template builder (shown in section 5.4.4) and lastly, editing or deleting the questionnaire

5.4.4 Template Builder

The Template builder is where users can create their questionnaire structures. It provides users options to interact with every aspect of a questionnaire, from its title and description. To adding new stages. A user can select a stage, to see its questions, add new questions and correspondingly add options and their custom scores on each question.

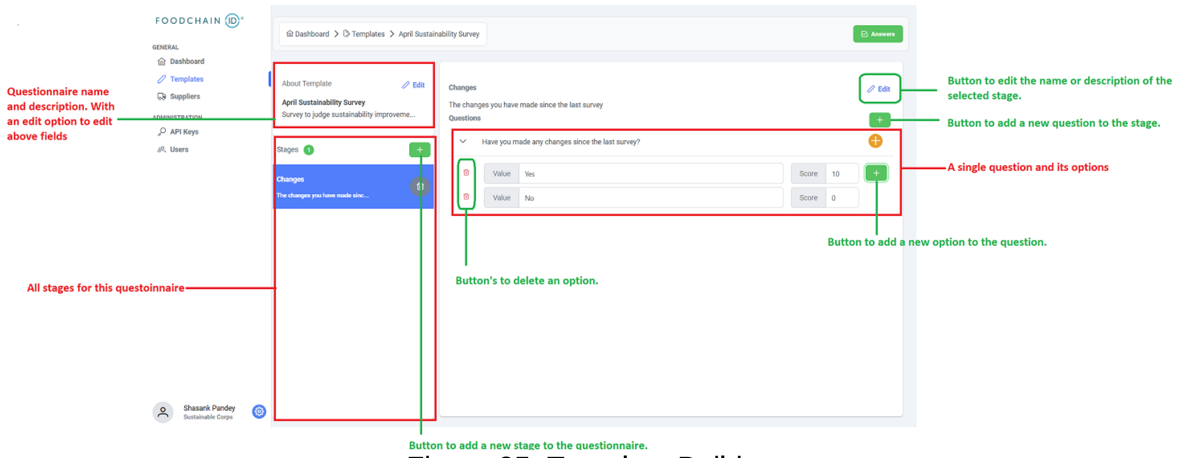
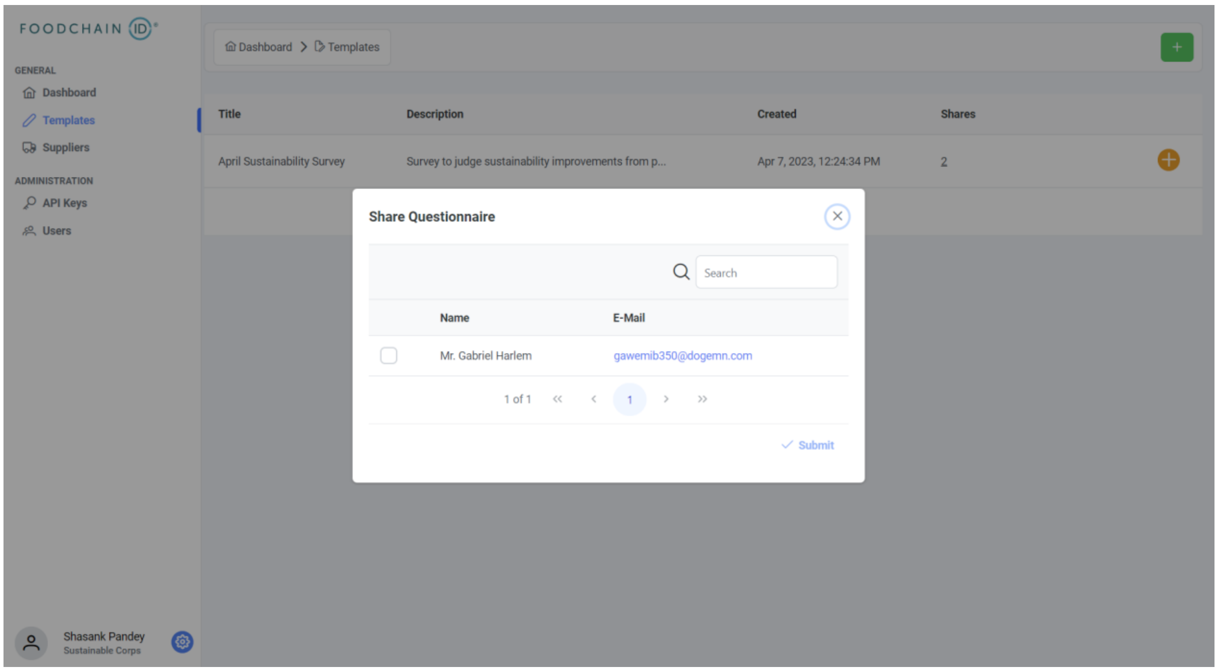

Questionnaire name and description. With an edit option to edit above fields
All stages for this questoinnaire
Button to edit the name or description of the selected stage.
Button to add a new question to the stage.
A single question and its options
Button to add a new option to the question.
Button's to delete an option.
Button to add a new stage to the questionnaire.
FOODCHAIN ID
Dashboard
Templates
Suppliers
API Keys
Users
April Sustainability Survey
Changes
Have you made any changes since the last survey?
Shasank Pandey

Figure 25: Template Builder

5.4.5 Template Sharing Dialog



Title	Description	Created	Shares
April Sustainability Survey	Survey to judge sustainability improvements from p...	Apr 7, 2023, 12:24:34 PM	2

Share Questionnaire

Search

	Name	E-Mail
☐	Mr. Gabriel Harlem	gawemib350@dogemn.com

1 of 1 « ‹ 1 › »

Submit

Figure 26: Question sharing dialog.

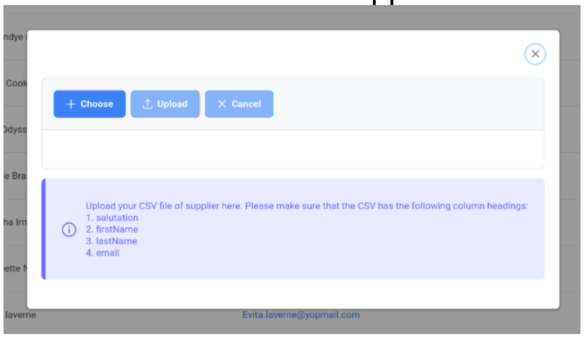
When the user presses the share button shown in Figure 24, a dialog is opened showing all the suppliers the user has, and therefore can share the questionnaire to. The Dialog shows the suppliers in a tabulated form, and allows the user to search for a supplier, by their name and email.

Once they select one or more suppliers to share the questionnaire with, they can use the submit button to close the dialog.

5.4.6 Suppliers Page

This is the page for viewing the suppliers. Their names and emails are listed. User and Admin roles can both see, edit, and add suppliers. Suppliers can be added in two ways i.e., users can add a supplier manually with the green plus button, or they can upload a CSV file with the yellow upload button. Application will save all the supplier information to the Suppliers page automatically.

When suppliers are added, they can be edited or removed from the list. The yellow plus button next to each supplier is for executing these actions.



Upload your CSV file of supplier here. Please make sure that the CSV has the following column headings:
1. salutation
2. firstName
3. lastName
4. email

Figure 27: Add suppliers from a CSV file

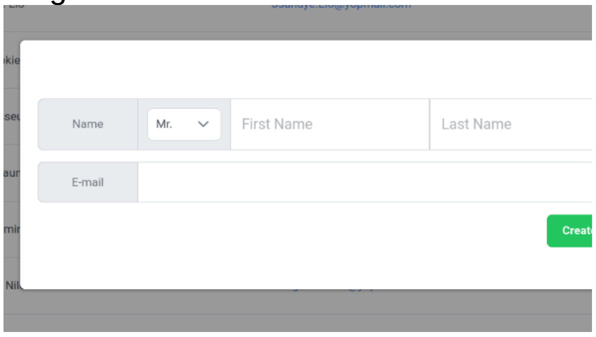
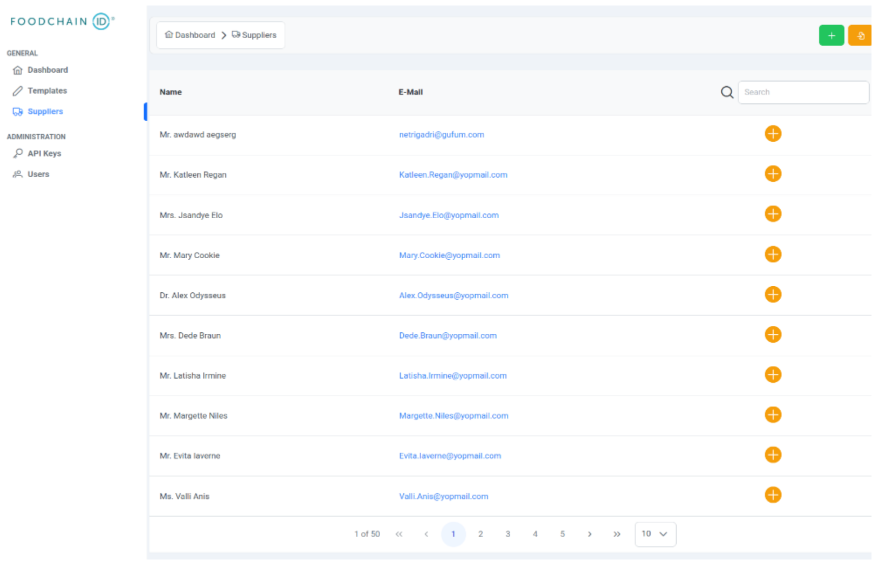


Figure 28: Add suppliers manually



Name	E-Mail
Mr. awdawd aegserg	netrigadri@gufum.com
Mr. Katleen Regan	Katleen.Regan@yopmail.com
Mrs. Jsandye Elo	Jsandye.Elo@yopmail.com
Mr. Mary Cookie	Mary.Cookie@yopmail.com
Dr. Alex Odysseus	Alex.Odysseus@yopmail.com
Mrs. Dede Braun	Dede.Braun@yopmail.com
Mr. Latisha Irmine	Latisha.Irmine@yopmail.com
Mr. Margette Niles	Margette.Niles@yopmail.com
Mr. Evita laverne	Evita.laverne@yopmail.com
Ms. Valli Anis	Valli.Anis@yopmail.com

Figure 29: Suppliers page

5.4.7 Users Page

This page can be seen by the administrator. It shows the list of registered users with their user id, email address, first name and last name and their roles. Admins can also invite users to create an account in the system. This can be done by the “Invite” button. In the page where you can invite users, there is “Add” button on the bottom left. When this button is triggered, the admin can enter the user email address and assign a role for the user. If there should be many users supposed to be added, it is also possible to add users by CSV files. When all the users are added, “Submit” button will save the users into the “Users” page.

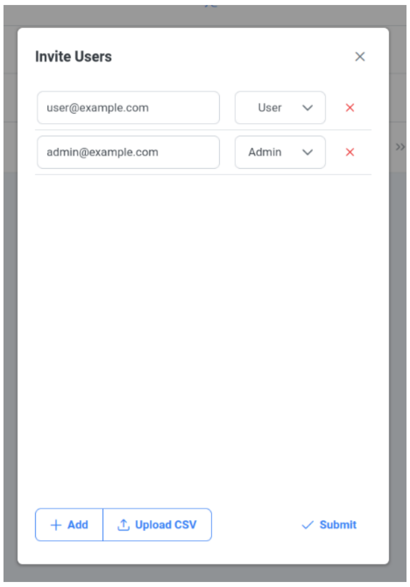
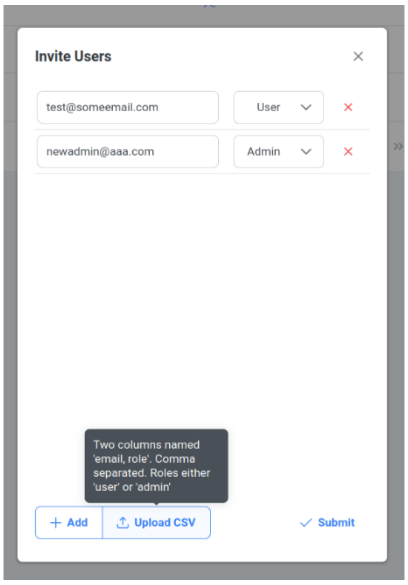
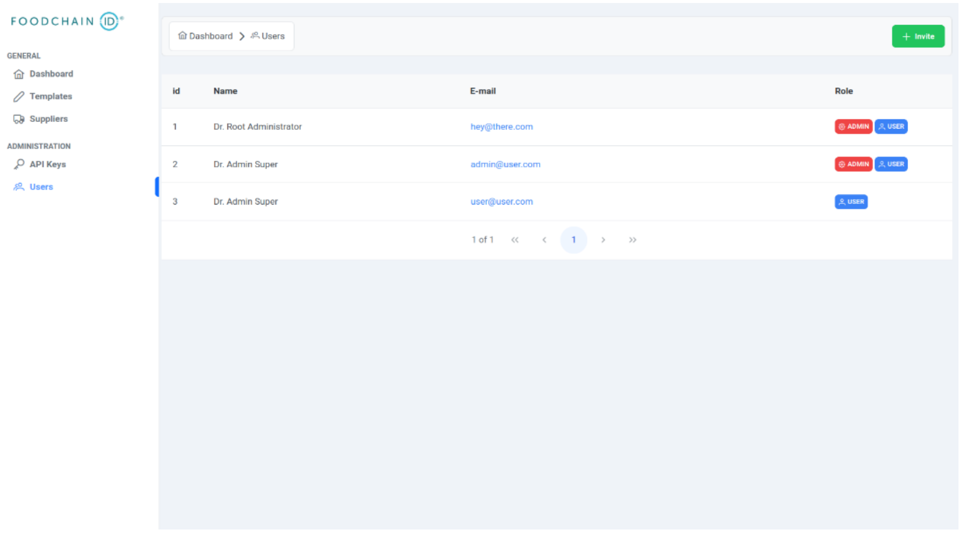


Figure 30: Add invite users manually



Two columns named 'email, role'. Comma separated. Roles either 'user' or 'admin'

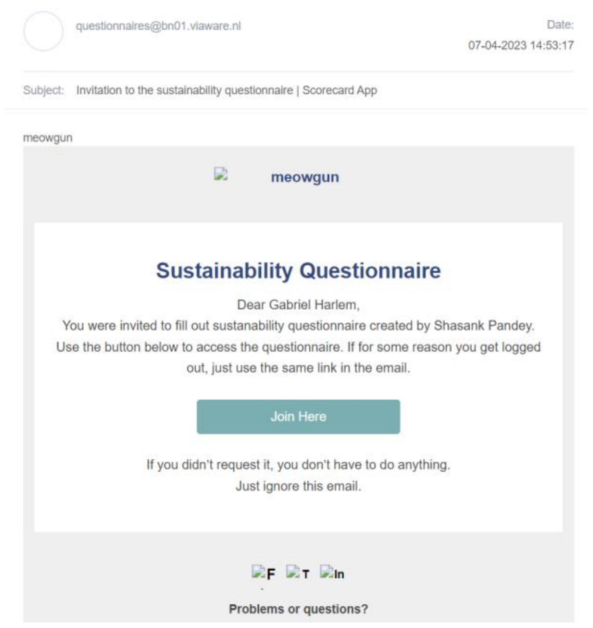
Figure 31: Invite users from a CSV file



id	Name	E-mail	Role
1	Dr. Root Administrator	hey@there.com	ADMIN USER
2	Dr. Admin Super	admin@user.com	ADMIN USER
3	Dr. Admin Super	user@user.com	USER

Figure 32: Users page

5.4.8 Questionnaire Answering Pages


questionnaires@bn01.viaware.nl

Date: 07-04-2023 14:53:17

Subject: Invitation to the sustainability questionnaire | Scorecard App

meowgun

meowgun

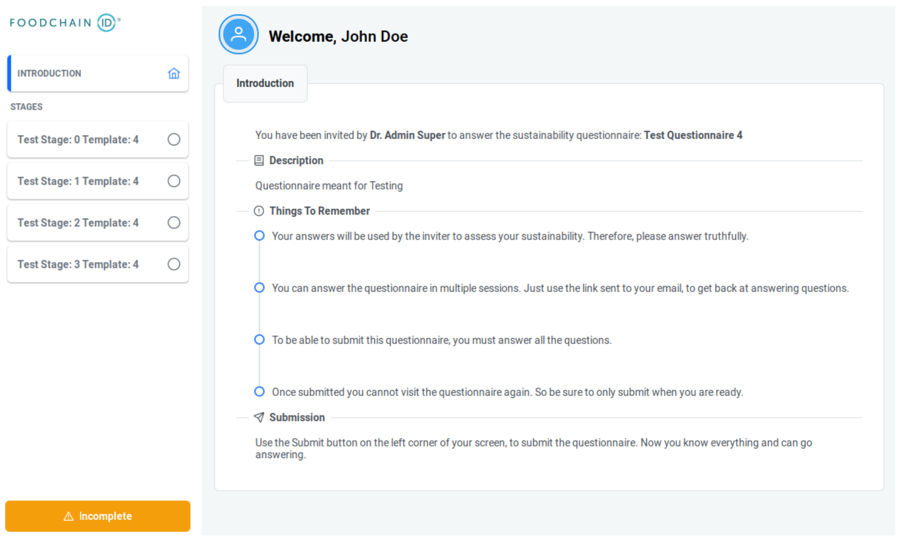
Sustainability Questionnaire

Dear Gabriel Harlem,
You were invited to fill out sustanability questionnaire created by Shasank Pandey. Use the button below to access the questionnaire. If for some reason you get logged out, just use the same link in the email.

If you didn't request it, you don't have to do anything. Just ignore this email.

Problems or questions?

Figure 33: Email received by supplier to answer a questionnaire.


FOODCHAIN ID
INTRODUCTION
STAGES
Test Stage: 0 Template: 4
Test Stage: 1 Template: 4
Test Stage: 2 Template: 4
Test Stage: 3 Template: 4
Incomplete

Welcome, John Doe

Introduction

You have been invited by **Dr. Admin Super** to answer the sustainability questionnaire: **Test Questionnaire 4**

Description

Questionnaire meant for Testing

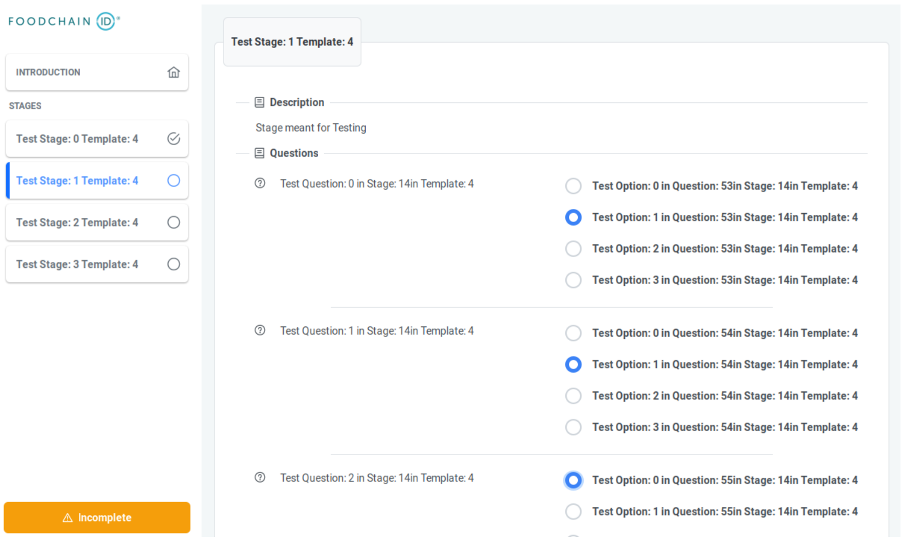
Things To Remember

- Your answers will be used by the inviter to assess your sustainability. Therefore, please answer truthfully.
- You can answer the questionnaire in multiple sessions. Just use the link sent to your email, to get back at answering questions.
- To be able to submit this questionnaire, you must answer all the questions.
- Once submitted you cannot visit the questionnaire again. So be sure to only submit when you are ready.

Submission

Use the Submit button on the left corner of your screen, to submit the questionnaire. Now you know everything and can go answering.

Figure 34: Questionnaire introduction page


FOODCHAIN ID
INTRODUCTION
STAGES
Test Stage: 0 Template: 4
Test Stage: 1 Template: 4
Test Stage: 2 Template: 4
Test Stage: 3 Template: 4
Incomplete

Test Stage: 1 Template: 4

Description

Stage meant for Testing

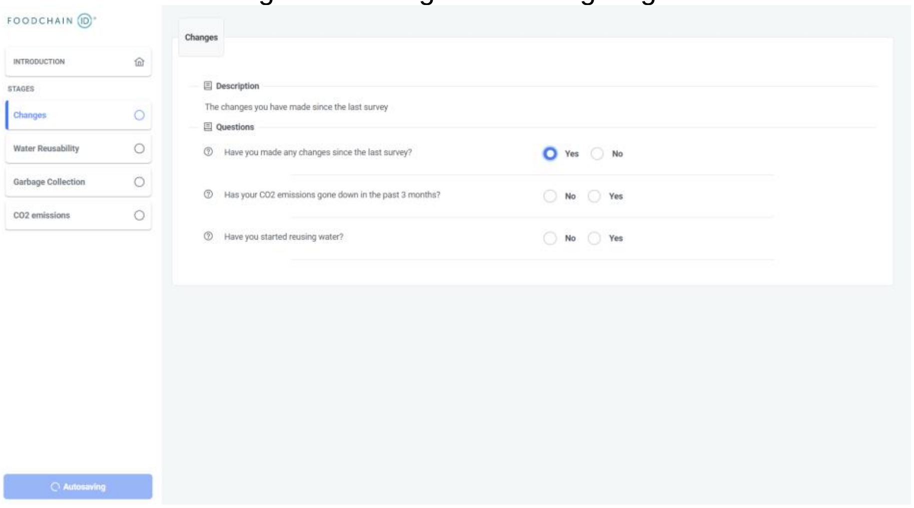
Questions

Test Question: 0 in Stage: 14in Template: 4
- Test Option: 0 in Question: 53in Stage: 14in Template: 4
- [✓ Test Option: 1 in Question: 53in Stage: 14in Template: 4]
- Test Option: 2 in Question: 53in Stage: 14in Template: 4
- Test Option: 3 in Question: 53in Stage: 14in Template: 4

Test Question: 1 in Stage: 14in Template: 4
- Test Option: 0 in Question: 54in Stage: 14in Template: 4
- [✓ Test Option: 1 in Question: 54in Stage: 14in Template: 4]
- Test Option: 2 in Question: 54in Stage: 14in Template: 4
- Test Option: 3 in Question: 54in Stage: 14in Template: 4

Test Question: 2 in Stage: 14in Template: 4
- [✓ Test Option: 0 in Question: 55in Stage: 14in Template: 4]
- Test Option: 1 in Question: 55in Stage: 14in Template: 4

Figure 35: Stage Answering Page



Changes

Description

The changes you have made since the last survey

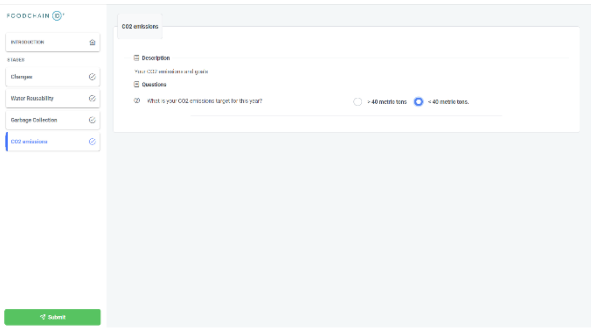
Questions

Have you made any changes since the last survey? [✓ Yes] No

Has your CO2 emissions gone down in the past 3 months? No Yes

Have you started reusing water? No Yes

Figure 36: AutoSaving of questionnaires.



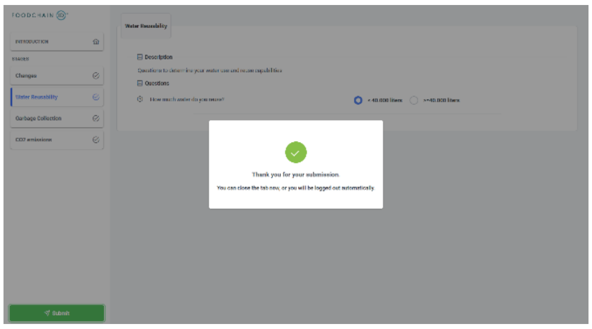
CO2 emissions

Description

Your CO2 emissions and goals

Questions

What is your CO2 emissions target for this year? > 40 metric tons [✓ < 40 metric tons.]



Water Reusability

Description

Questions to determine your water use and reuse capabilities

Questions

How much water do you reuse? [✓ < 40.000 liters] >=40.000 liters

Thank you for your submission.

You can close the tab now, or you will be logged out automatically.

Figure 37a: Questionnaire Submission Button Active

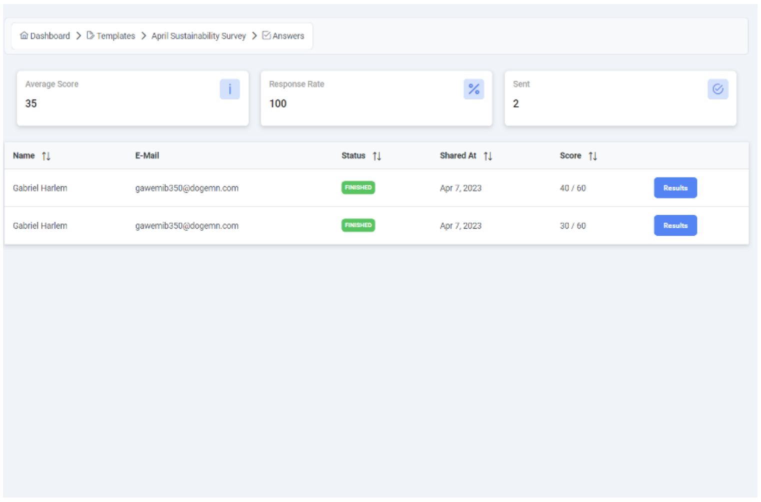
Figure 37b: Questionnaire Submission

When a questionnaire is shared with a supplier, they receive an email, as shown in Figure 33, They can use the “Join Here” button in the email, to directly be logged in and answer the questionnaire. The first page they are shown is an introduction to the questionnaire, and who invited them to answer it, this can be seen in Figure 34. They can then go through the stages and answer each stage like in Figure 35.

The “Incomplete Button” shows the current status of the questionnaire, whenever the questionnaire is auto-saved, the button is changed to reflect and notify the user of the saving as can be seen in Figure 37a.

Lastly, upon submission, the user is shown the screen in Figure 37b and is logged out of the platform.

5.4.9 Questionnaire Results Pages



Dashboard > Templates > April Sustainability Survey > Answers

Average Score 35 | Response Rate 100 | Sent 2

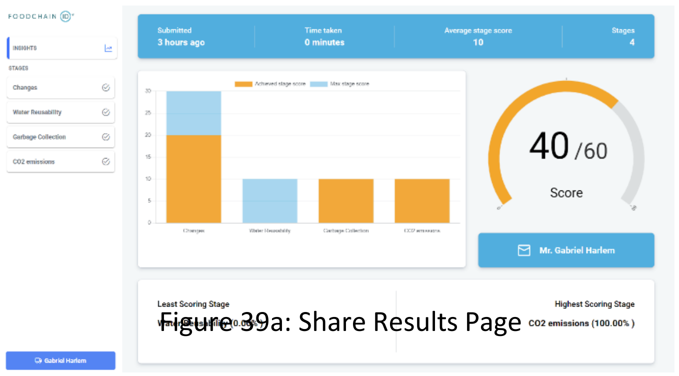
Name	E-Mail	Status	Shared At	Score	
Gabriel Harlem	gawemib350@dogemn.com	FINISHED	Apr 7, 2023	40 / 60	Results
Gabriel Harlem	gawemib350@dogemn.com	FINISHED	Apr 7, 2023	30 / 60	Results

Figure 38: Questionnaire Results Page

The Questionnaire Results page shown in Figure 38, shows all the suppliers, the questionnaire was shared with, along with some statistics like the average score, the response rate, and the total number of times it was shared.

There is also a table that shows the supplier the questionnaire was shared to, the status of the share which can be SENT, ANSWERING or FINISHED along with the time of sharing and the score they received.

Lastly, the user can use the Results button to check how a supplier answered the questionnaire. Using that button opens the screen shown in Figure 39a.

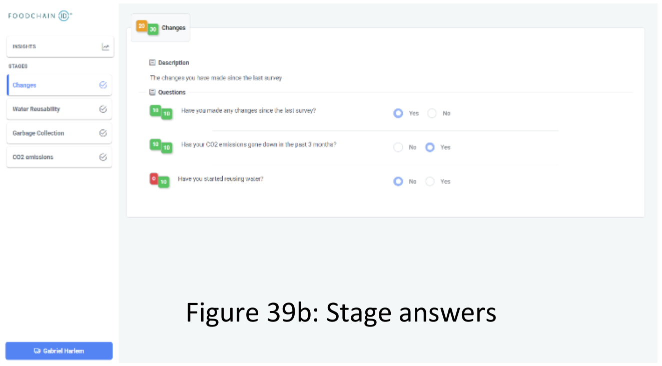


Submitted 3 hours ago | Time taken 0 minutes | Average stage score 10 | Stages 4

Least Scoring Stage: Water Reusability (0.00%)

Highest Scoring Stage: CO2 emissions (100.00%)

Figure 39a: Share Results Page



Changes

Description

The changes you have made since the last survey

Questions

Have you made any changes since the last survey? [✓ Yes] No

Has your CO2 emissions gone down in the past 3 months? No [✓ Yes]

Have you started reusing water? [✓ No] Yes

Figure 39b: Stage answers

When the user is looking at the results of a particular share, he is shown statistics of how the supplier answered the questionnaire, like a chart of how much the supplier scored out of the total for each stage, their total scoring, the stage with the least score and the stage with the highest score.

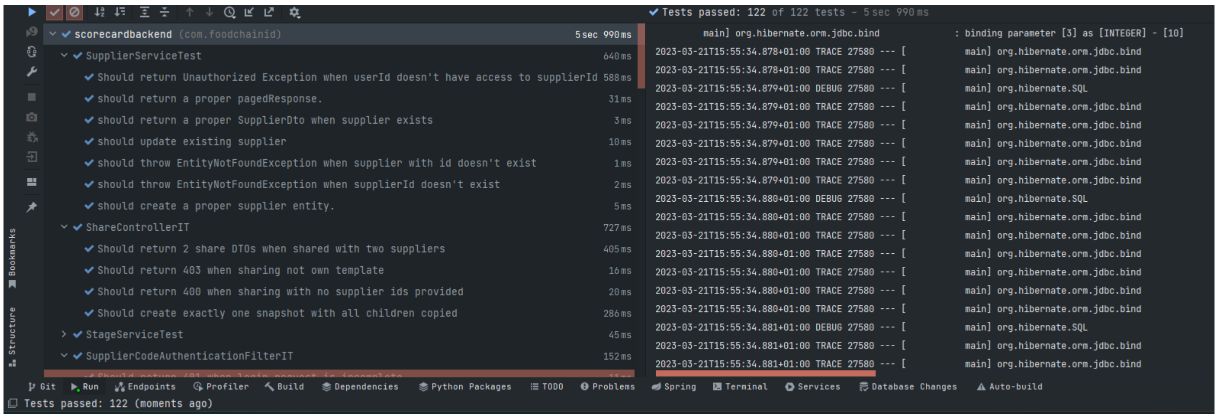
Moreover, the user can also see, information about when the supplier submitted the quesitonnaire, how much time he took and what was their average score per stage.

The user can also click on individual stages, in the sidebar, which opens the screen shown in Figure 39b, where they can see what the supplier answered for each question and what was their score for that answer.

6. TESTING

Automated tests are a vital part of any computer system that ensures correctness of the code and increases degree of developer confidence. Therefore, the team set as its initial target to test every part of the business logic.

Our testing strategy included 3 types of tests, namely Unit, Integration and User testing, each designed to test a different layer of the application. Figure 40 shows a test run of our 122, Unit and integration tests, all having passed successfully. These tests and their goals are described in greater details in the following sections.



```
scorecardbackend (com.foodchainid)                                                        5 sec 990 ms
  SupplierServiceTest                                                                           640 ms
    ✓ Should return Unauthorized Exception when userId doesn't have access to supplierId        588 ms
    ✓ should return a proper pagedResponse.                                                      31 ms
    ✓ should return a proper SupplierDto when supplier exists                                     3 ms
    ✓ should update existing supplier                                                            10 ms
    ✓ should throw EntityNotFoundException when supplier with id doesn't exist                    1 ms
    ✓ should throw EntityNotFoundException when supplierId doesn't exist                          2 ms
    ✓ should create a proper supplier entity.                                                     5 ms
  ShareControllerIT                                                                             727 ms
    ✓ Should return 2 share DTOs when shared with two suppliers                                 405 ms
    ✓ Should return 403 when sharing not own template                                            16 ms
    ✓ Should return 400 when sharing with no supplier ids provided                               20 ms
    ✓ Should create exactly one snapshot with all children copied                               286 ms
  StageServiceTest                                                                               45 ms
  SupplierCodeAuthenticationFilterIT                                                            152 ms
```

```
✓ Tests passed: 122 of 122 tests – 5 sec 990 ms
        main] org.hibernate.orm.jdbc.bind           : binding parameter [3] as [INTEGER] - [10]
2023-03-21T15:55:34.878+01:00 TRACE 27580 --- [           main] org.hibernate.orm.jdbc.bind
2023-03-21T15:55:34.878+01:00 TRACE 27580 --- [           main] org.hibernate.orm.jdbc.bind
2023-03-21T15:55:34.879+01:00 DEBUG 27580 --- [           main] org.hibernate.SQL
2023-03-21T15:55:34.879+01:00 TRACE 27580 --- [           main] org.hibernate.orm.jdbc.bind
2023-03-21T15:55:34.879+01:00 TRACE 27580 --- [           main] org.hibernate.orm.jdbc.bind
2023-03-21T15:55:34.879+01:00 TRACE 27580 --- [           main] org.hibernate.orm.jdbc.bind
2023-03-21T15:55:34.879+01:00 TRACE 27580 --- [           main] org.hibernate.orm.jdbc.bind
2023-03-21T15:55:34.879+01:00 TRACE 27580 --- [           main] org.hibernate.orm.jdbc.bind
2023-03-21T15:55:34.880+01:00 DEBUG 27580 --- [           main] org.hibernate.SQL
2023-03-21T15:55:34.880+01:00 TRACE 27580 --- [           main] org.hibernate.orm.jdbc.bind
2023-03-21T15:55:34.880+01:00 TRACE 27580 --- [           main] org.hibernate.orm.jdbc.bind
2023-03-21T15:55:34.880+01:00 TRACE 27580 --- [           main] org.hibernate.orm.jdbc.bind
2023-03-21T15:55:34.880+01:00 TRACE 27580 --- [           main] org.hibernate.orm.jdbc.bind
2023-03-21T15:55:34.880+01:00 TRACE 27580 --- [           main] org.hibernate.orm.jdbc.bind
2023-03-21T15:55:34.880+01:00 TRACE 27580 --- [           main] org.hibernate.orm.jdbc.bind
2023-03-21T15:55:34.881+01:00 DEBUG 27580 --- [           main] org.hibernate.SQL
2023-03-21T15:55:34.881+01:00 TRACE 27580 --- [           main] org.hibernate.orm.jdbc.bind
2023-03-21T15:55:34.881+01:00 TRACE 27580 --- [           main] org.hibernate.orm.jdbc.bind
```

```
Tests passed: 122 (moments ago)
```

Figure 40: Backend Tests.

6.1 UNIT TESTING

Unit Tests were our primary focus, and therefore we created a lot of them. The reasoning behind it was that Unit Tests would test our components in isolation from each other, ensuring that the output is deterministic.

The reason why we tried to maximize the number of unit tests as opposed to other means of testing lies in the notion of Testing Pyramid. According to the Testing Pyramid developed by Mike Cohen (2015) in his book Succeeding with Agile, he outlines that unit tests are extremely fast and computationally inexpensive means of ensuring that a software operates correctly. Mike Cohen encourages developers to cover most of the business functionality by using unit tests because that will decrease the build and testing times at the expense of confidence.

For our application we wrote around 75 Unit Tests, with the focus on testing the service layer of our application.

6.2 INTEGRATION TESTING

To verify that the software integrates well with external components such as databases and 3rd party APIs, the team ensured that all these interactions are covered by a set of integration tests that do not use any mocks and simulate real production settings.

As a rule of thumb, the integration test was written whenever the code performed database operations, 3rd party API calls and of course authentication/authorization.

To simplify the integration testing and to make it as close as possible to the real-world settings, we decided to use the Test Containers library that creates an identical infrastructure as in production by spinning up the Docker containers.

6.3 USER TESTING

The key to acceptance of the software is not only its ability to perform correctly but also being useful to the end user. Therefore, we also performed User Acceptance Testing into the workflow. Before each production release the whole application was deployed to the staging environment where it was validated by the client and if everything was in order, the same application was deployed to production.

We also tested the application ourselves, to ensure all components worked together and the application did perform the tasks it was designed to. These tests were done at the end of the sprints, individually, and any bugs, if found, were converted into issues that the team picked up on in the next sprint.

7. CONCLUSION

After the completion of our development cycle, we showcased the final version of the application to our client i.e., FoodChainID, who were very happy with the capabilities of the application.

We were able to achieve all the “Must” functional requirements along with some of the other requirements. Lastly, the testing strategy employed by us, proved to be very helpful in not only creating a software with tested business logic, but also in easing our developmental stress, when introducing breaking changes.

Furthermore, we created a User Manual that can be found in Appendix A. The aim of the manual was to allow the users of this application to understand how each component works, so that they can extract the most out of the features created by us.

To conclude the report, we have presented a detailed risk analysis of the project along with future improvements that can be made to the project. Lastly, there is also a table to signal each member’s contribution throughout the project development.

7.1 RISK ANALYSIS

As the web application was designed to be used in production, assessing the risks of the platform is important. Which is why, we tried to identify any major risks with the application.

In our analysis, we focused on security and the core business logic, to ensure there were no risks with these areas. As both these aspects were well tested, we have complete confidence in their capabilities.

However, a risk that we did identify, was with how some users might use it. The application’s major goal is to simplify the collection of data from suppliers, however as the questions on the platform are multiple-choice, there is a risk of suppliers just choosing random options in order to complete the questionnaire.

We do provide users with the time taken to complete answering a questionnaire, to help them know if a supplier chose random options or actually paid attention to the questions, however, more can be done to assess this risk in future iterations.

7.2 FUTURE IMPROVEMENTS

As the application was developed in a very short period, we focused on developing the most important requirements and therefore there are certain improvements that we would have liked to incorporate if we had more time.

As mentioned in section 7.1, there is a risk of suppliers not answering the questionnaire properly, this can be tackled with asking suppliers to upload official sustainability documents with their corresponding questions, moreover, more metrics can be incorporated in the answering phase, to have a better understanding of how the questions were answered.

Lastly, the application could have integrations with various ERP systems allowing users to connect them and import suppliers directly from their ERP systems along with the current functionality of importing suppliers from a CSV file.

7.3 CONTRIBUTIONS

All the members worked to create the web application and this report. The table below mentions all the members and their contributions.

TEAM MEMBER	CONTRIBUTIONS
AKMAL ALIKHUJAEV	Security, Testing, Backend, Frontend, Backend Design, Report
BUGRA VEYSEL YILDIZ	High Level Design, Report, Diagrams, Backend, User Manual
SHASANK SEKHAR PANDEY	Testing, Backend, Frontend, Backend Design, Report, Security
VLADIMIR KOBZEV	Frontend Design, Frontend, Report

8. REFERENCES

1. Cohn, M., & Lister, T. R. (2015). *Succeeding with agile: Software development using scrum*. Pearson.
2. Fowler, M. (2006, May 1). *Continuous integration*. martinfowler.com. Retrieved March 15, 2023, from https://martinfowler.com/articles/continuousIntegration.html
3. Hammant, P. (2017). *Trunk based development*. Trunk Based Development. Retrieved March 15, 2023, from https://trunkbaseddevelopment.com/
4. IBM. (2022). *What is three-tier architecture*. IBM. Retrieved March 21, 2023, from https://www.ibm.com/topics/three-tier-architecture
5. Newman, S. (2021). Build - A Brief Introduction to Continuous Integration. In *Building microservices: Designing fine-grained systems* (pp. 197–201). essay, O'Reilly.

9. APPENDIX A

User Manual

TABLE OF CONTENTS

PURPOSE / DESCRIPTION

Sustainability Scorecard web-application is a tool that is designed to solve the problem of judging the sustainability of one's practices. In the platform, it can be done by users who create questionnaires with a predefined grading scheme, that can be shared with the suppliers. The progress of these shared questionnaires can be tracked and once completed, can be used by the business to see the answers and the sustainability score of the partner. Helping them understand the sustainability of their own business in the process.

The goal of the project is to create a web application, where the administrator i.e. FoodChainID, can invite their customers to create accounts. These customers can then use the application to create sustainability questionnaire templates, or use global templates developed by experts at FoodChainID, share these templates with their suppliers and track their progress. From the supplier side, the application should provide an interface to answer and submit the questionnaire.

This document is a guide for performing the features of the Sustainability Scorecard Application for all the user roles.

USER GUIDE

SIGN UP/LOGIN

1. Login page is the first interaction that the users will be making with the application. If you already have, enter the username and password credentials and click **Login**. If you are not a registered user and have an invite code, you can click on **Sign Up with Invite Code** as it can be seen in Figure A.1.

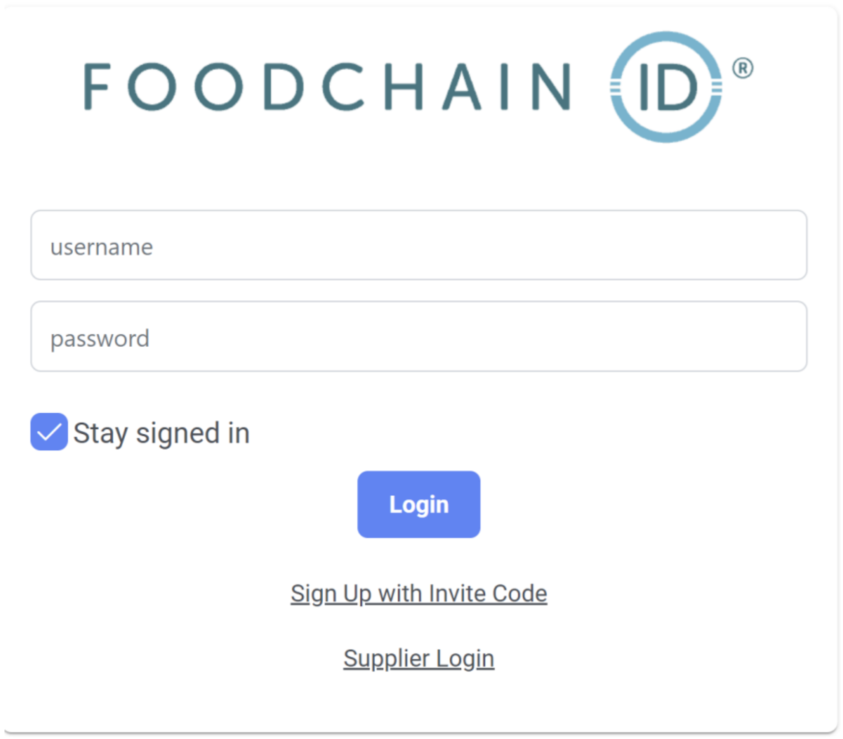
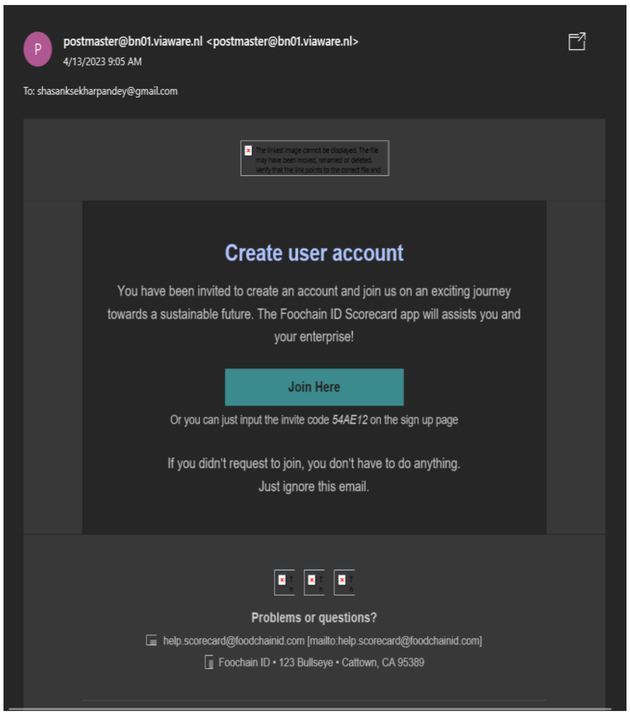


Figure A.1: Login panel

2. In the Sign Up page, new users can sign up with their email address, first and last name, password and their invite code. After all the parts have been filled, the **Signup** button can be clicked to proceed.



postmaster@bn01.viaware.nl <postmaster@bn01.viaware.nl>
4/13/2023 9:05 AM

To: shasanksekharpandey@gmail.com

The linked image cannot be displayed. The file may have been moved, renamed or deleted. Verify that the link points to the correct file and

Create user account

You have been invited to create an account and join us on an exciting journey towards a sustainable future. The Foochain ID Scorecard app will assists you and your enterprise!

Join Here

Or you can just input the invite code *54AE12* on the sign up page

If you didn't request to join, you don't have to do anything.
Just ignore this email.

Problems or questions?

help.scorecard@foodchainid.com [mailto:help.scorecard@foodchainid.com]

Foochain ID • 123 Bullseye • Cattown, CA 95389

Figure A.2: Invite email

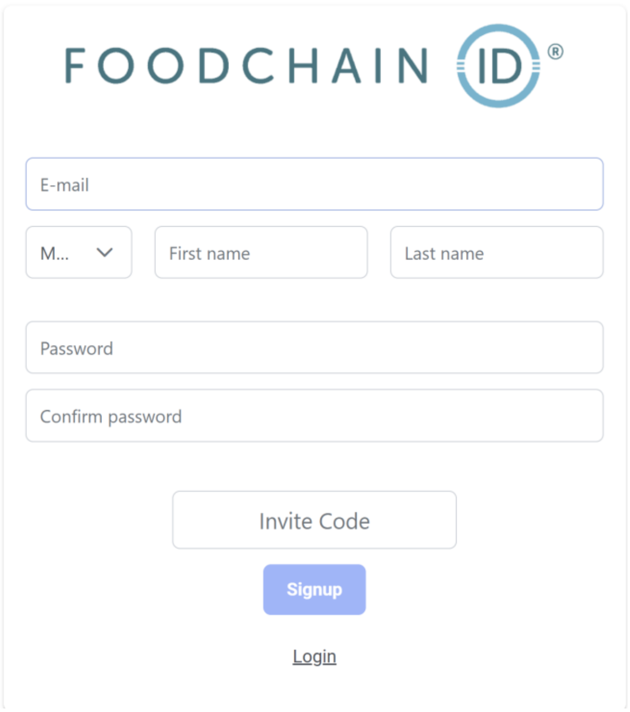
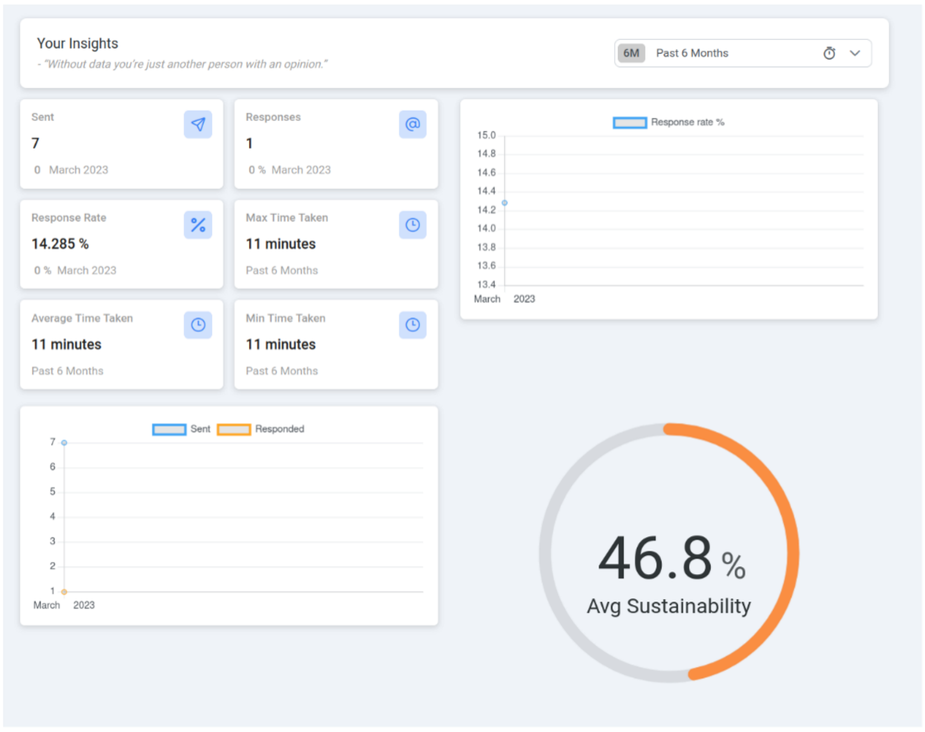


Figure A.3: Invite email

DASHBOARD PAGE

After you sign up, you will be redirected to your account. Dashboard page is the first page that you will be redirected to. In this page you can see the analytic data of the questionnaires that have been filled by the suppliers. An example of the dashboard page can be found in Figure A.4.



Your Insights
- "Without data you're just another person with an opinion."

6M Past 6 Months

Sent
7
0 March 2023

Responses
1
0 % March 2023

Response Rate
14.285 %
0 % March 2023

Max Time Taken
11 minutes
Past 6 Months

Average Time Taken
11 minutes
Past 6 Months

Min Time Taken
11 minutes
Past 6 Months

46.8 %
Avg Sustainability

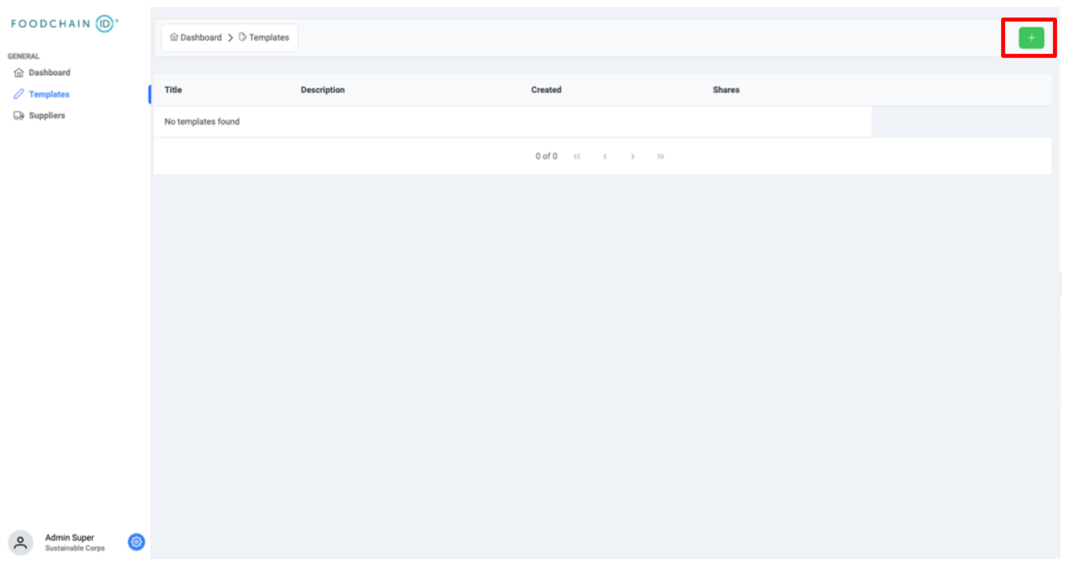
Figure A.4: Dashboard Page

TEMPLATES PAGE

In the template page, you can view the existing questionnaire templates listed with their titles, descriptions, date and time that it's been created and the number of shares.

1. Create a Questionnaire Template

To be able to create a questionnaire template, you can click on the green plus icon which is located on the top right corner of the page. It can be seen in the red box in Figure A.5. The pop up page will appear for you to enter the title and description of the template. Once you are done, you can click on **Create** to continue.



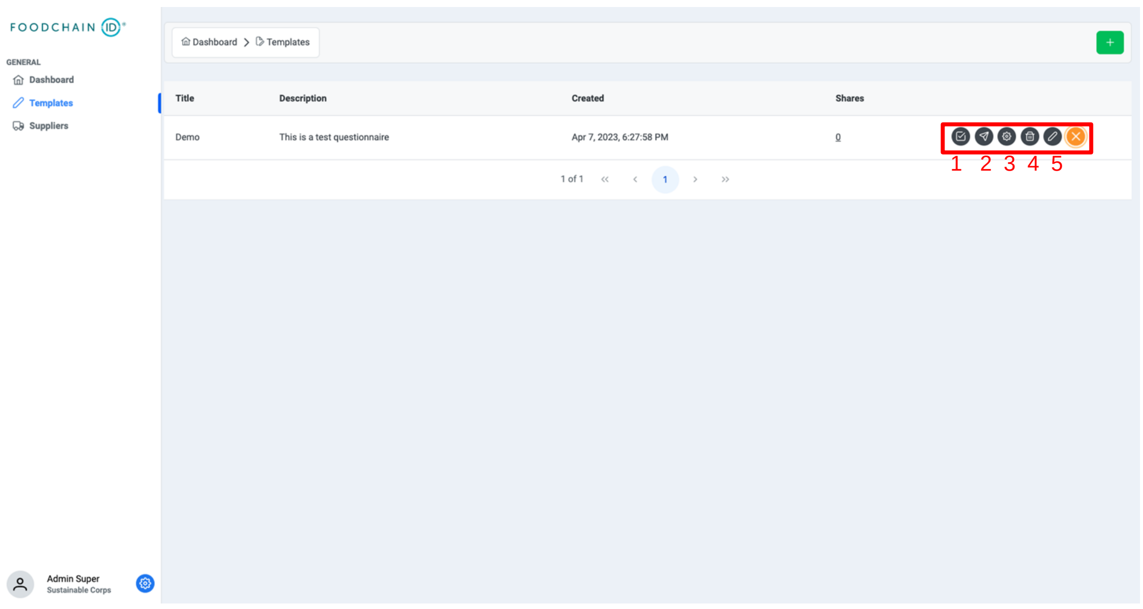
Title	Description	Created	Shares
No templates found			

0 of 0

Figure A.5: Templates page

2. Manage Questionnaire Template

Once you created the questionnaire, you can edit it by clicking the orange plus icon which is located next to each of the questionnaire that you have created. It has been shown with a red box in Figure A.6. Once you click on the plus icon, four buttons will appear.



Title	Description	Created	Shares
Demo	This is a test questionnaire	Apr 7, 2023, 6:27:58 PM	0

Figure A.6: Edit questionnaire

3. Edit or Delete the Questionnaire Template

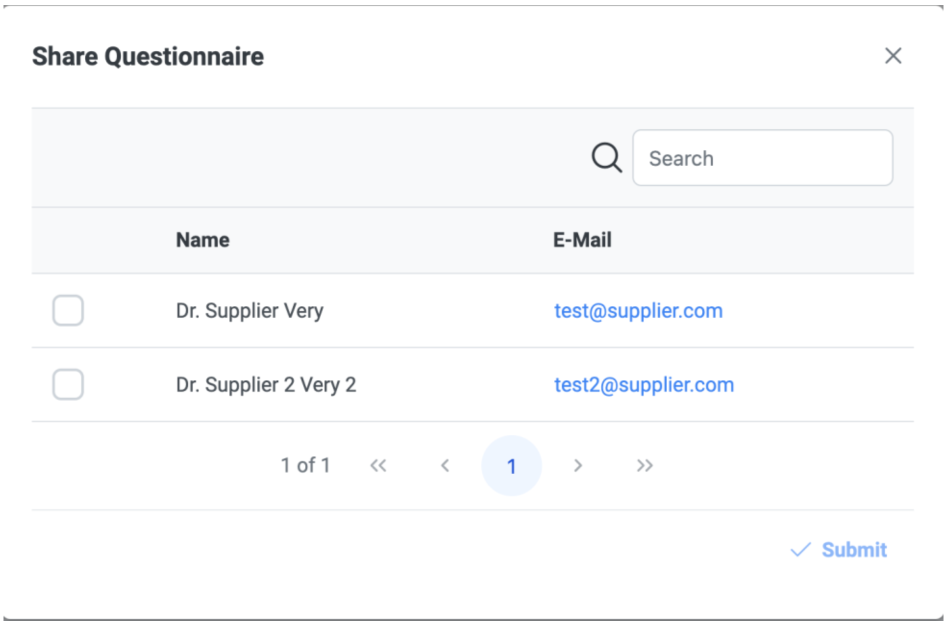
There is a **Delete** button which is shown by button number 4 in Figure A.6. It can be used for removing the questionnaire template. **Edit** button which is represented by button number 5 in Figure A.6 for updating the template name and description.

4. View the Analytics of Each Template

The icon with the **checkbox** which is represented by button number 1 in Figure A.6 shows the analytics of the questionnaire template. It shows the average score, response rate, the number of suppliers you sent and details of each person who filled out the questionnaire.

5. Share the Questionnaire Template

The button number 2 in Figure A.6 is for **sharing** the questionnaire with the suppliers. Once you click on that button, the page where you can select the suppliers will appear. It can be seen in Figure A.6.1. You can select or search the supplier that you want to share the questionnaire with.


Share Questionnaire

Search

	Name	E-Mail
☐	Dr. Supplier Very	test@supplier.com
☐	Dr. Supplier 2 Very 2	test2@supplier.com

Figure A.6.1: Share the questionnaire

6. Add Stages to the Template

The setting icon, button number 3 in Figure A.6 is for **adding stages** to the questionnaire. When you click on the button, the page will appear where you can see all the stages. To be able to create a stage, you can click on the green plus icon, button number 1 in Figure A.6.2. Enter the title and description about the stage and then click on **Create.**

Once you add the stage, it will appear on the right side of the screen. It is possible to edit and update the stage name and description by clicking on the **Edit** button. It can be seen in Figure A.6.2, button 2. To be able to add the questions to the stage, there is a plus button located next to Questions which can be seen with button number 3 in Figure A.6.2. When the plus button is clicked, you can enter the title for your question. The question will be added when you click on the **Create** button. Its name can be edited, or the question can be deleted from the menu that appears when the orange plus icon is triggered, which is button number 4 in Figure A.6.2. From the drop-down menu, with button number 5, users can enter the value and the score of the question. Then the orange **Check** button can be clicked to save**.**

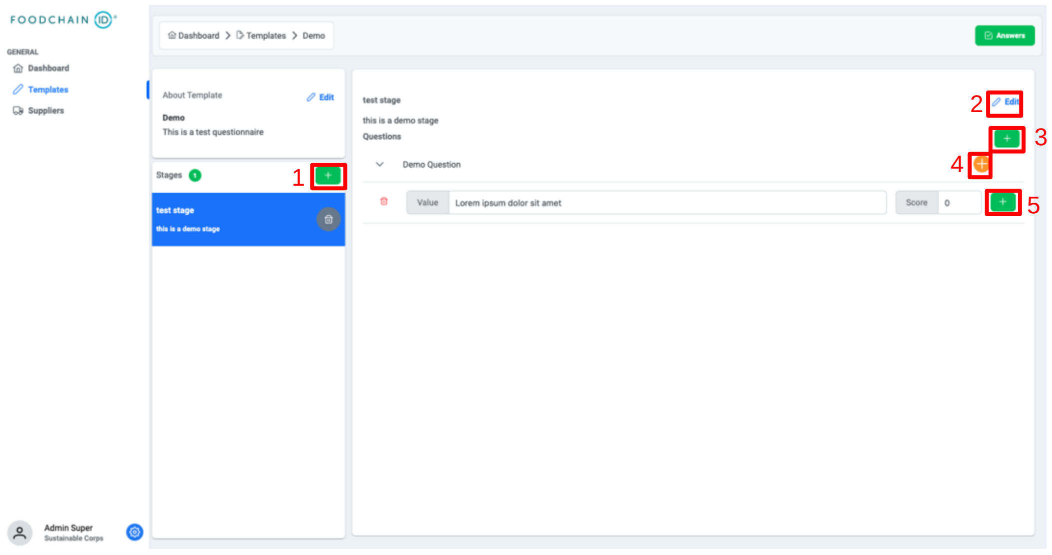


Figure A.6.2: Adding questions to the stage

SUPPLIER PAGE

This is the page for viewing the suppliers. Their names and emails are listed. You can both see, edit and add suppliers.

1. Add Supplier

It is possible to add suppliers both manually and from .csv file format. To add a supplier manually, there is a green plus button located at the top right of the page. It can be seen from Figure A.7 below. First name, last name and email of the supplier needs to be entered. Then, clicking the Create button will add the supplier into the Suppliers list.

The orange upload file button, which is located at the top right of the page can be used to add suppliers from a CSV file. The panel for uploading CSV files is shown below in Figure A.7.1. Make sure the CSV file is in the specified structure. **Choose** button can be clicked to select the file and then click **Upload** to save the suppliers.

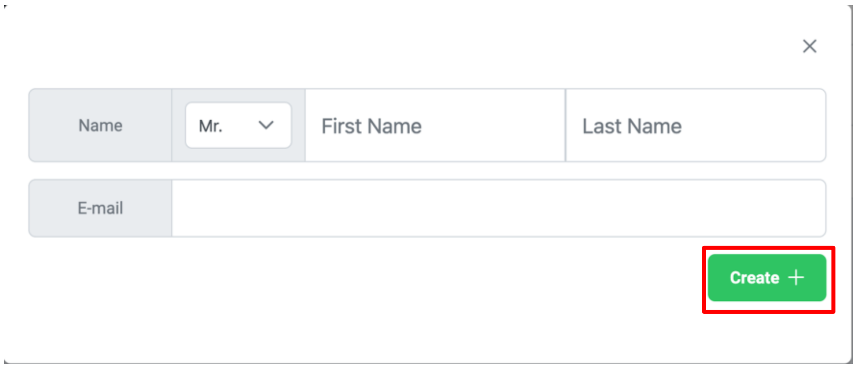
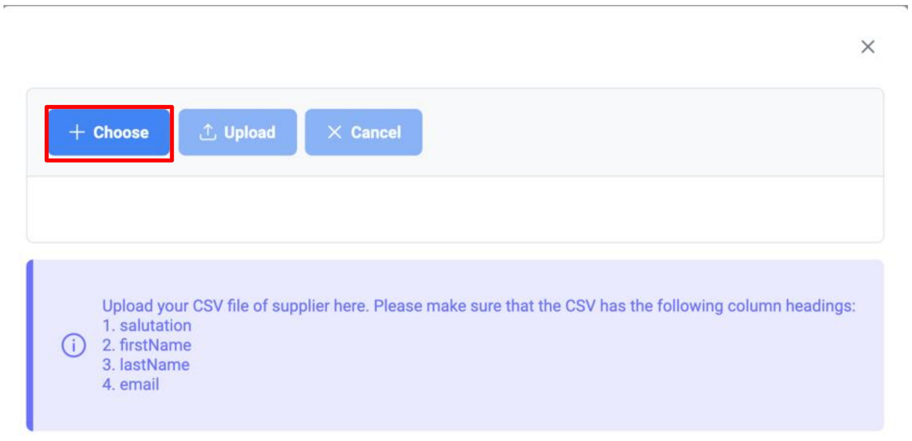


Figure A.7: Add Supplier Manually

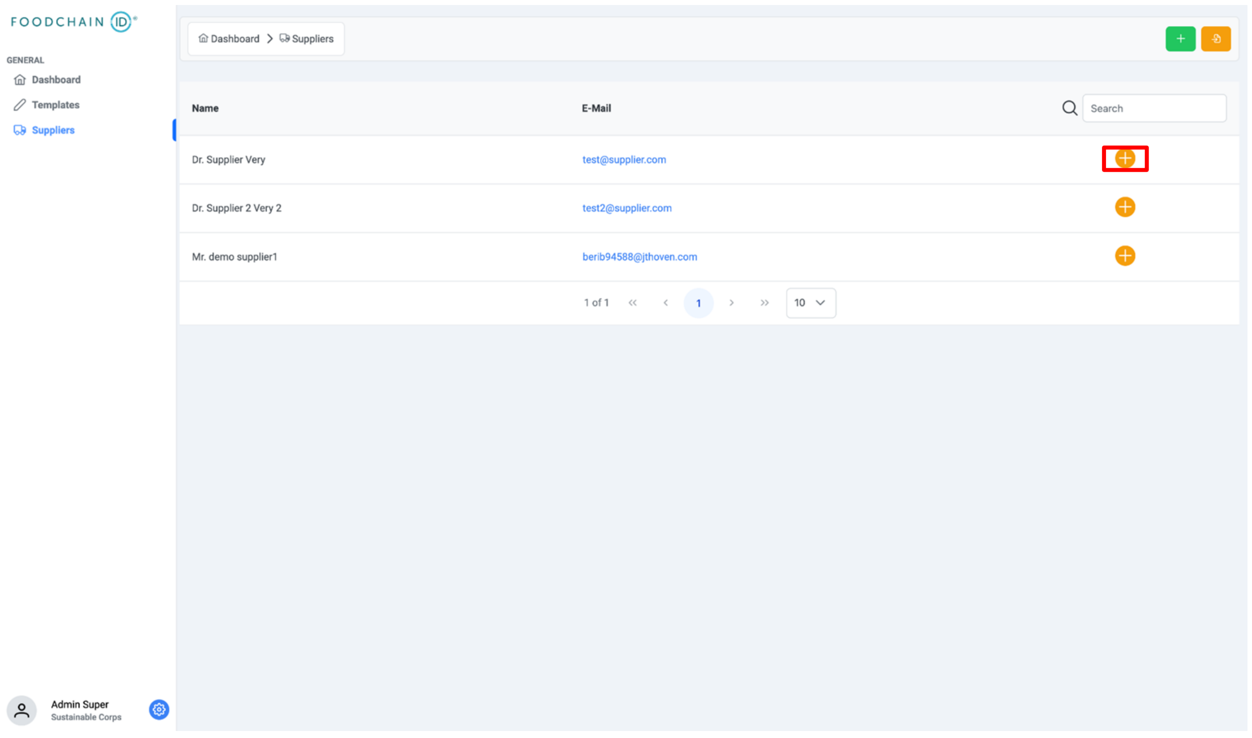


Upload your CSV file of supplier here. Please make sure that the CSV has the following column headings:
1. salutation
2. firstName
3. lastName
4. email

Figure A.7.1: Add Supplier from CSV file

2. Edit or Delete Supplier

Supplier data can be edited, or supplier can be deleted from the list. This can be done by clicking the orange plus icon located next to the specific supplier, as it is shown by the red box in Figure A.8.



Name	E-Mail
Dr. Supplier Very	test@supplier.com
Dr. Supplier 2 Very 2	test2@supplier.com
Mr. demo supplier1	berib94588@jthoven.com

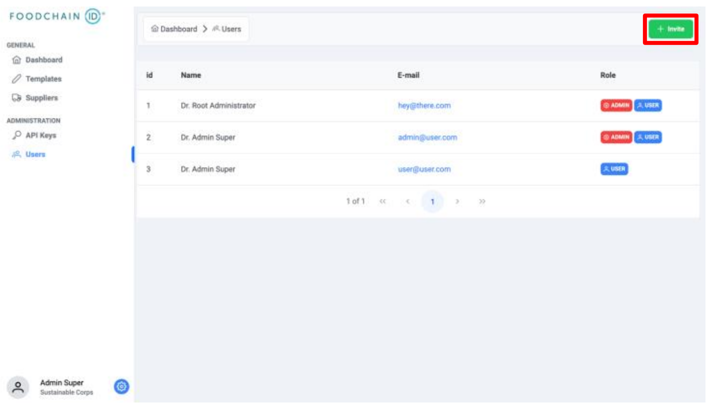
Figure A.8: Edit or Delete the saved supplier

ADMIN GUIDE

Admin role has ability to perform all the features that User role can. In addition to those features, you are able to invite users to the web application. In the menu which is located on the left side of the page, there is the Users page.

INVITE USERS

Admin role can view the list of users in the Users page. Their user ids, names, email addresses and roles are represented in the list. As it can be seen in the box in Figure A.9, there is an **Invite** button located at the top right of the page. This will bring you the Invite Users panel. It can be seen from Figure A.9.1. You need to enter the user email address and specify the role of the user from the dropdown menu.



id	Name	E-mail	Role
1	Dr. Root Administrator	hey@there.com	ADMIN USER
2	Dr. Admin Super	admin@user.com	ADMIN USER
3	Dr. Admin Super	user@user.com	USER

Figure A.9: Managing Users page

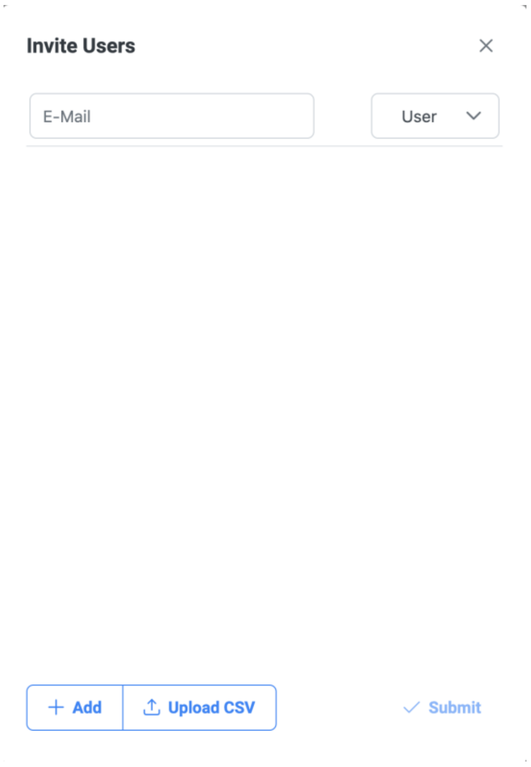
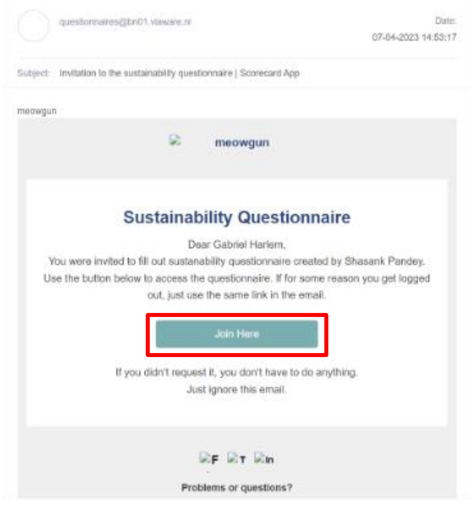


Figure A.9.1: Invite Users panel

SUPPLIER GUIDE

ANSWER QUESTIONNAIRE

Once the User role shares the template with the Supplier, an invite email is sent to the supplier's email address. An example of the email can be seen in Figure A.10. You can use the **Join Here** button in the email, to directly be logged in and answer the questionnaire. The first page that is shown is an introduction to the questionnaire, and who invited them to answer it, this can be seen in Figure A.10.1. You can then go through the stages and answer each stage like in Figure A.10.2.



questionnaires@bn01.viaware.nl
Date:
07-04-2023 14:53:17

Subject: Invitation to the sustainability questionnaire | Scorecard App

meowgun

meowgun

Sustainability Questionnaire

Dear Gabriel Harlem,
You were invited to fill out sustanability questionnaire created by Shasank Pandey.
Use the button below to access the questionnaire. If for some reason you get logged out, just use the same link in the email.

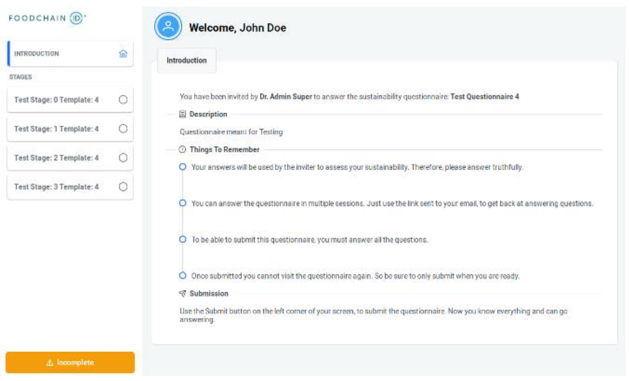
Join Here

If you didn't request it, you don't have to do anything.
Just ignore this email.

F T in

Problems or questions?

Figure A.10: Example of Invitation Email



FOODCHAIN ID

INTRODUCTION

STAGES

Test Stage: 0 Template: 4
Test Stage: 1 Template: 4
Test Stage: 2 Template: 4
Test Stage: 3 Template: 4

Incomplete

Welcome, John Doe

Introduction

You have been invited by **Dr. Admin Super** to answer the sustainability questionnaire: **Test Questionnaire 4**

Description

Questionnaire meant for Testing

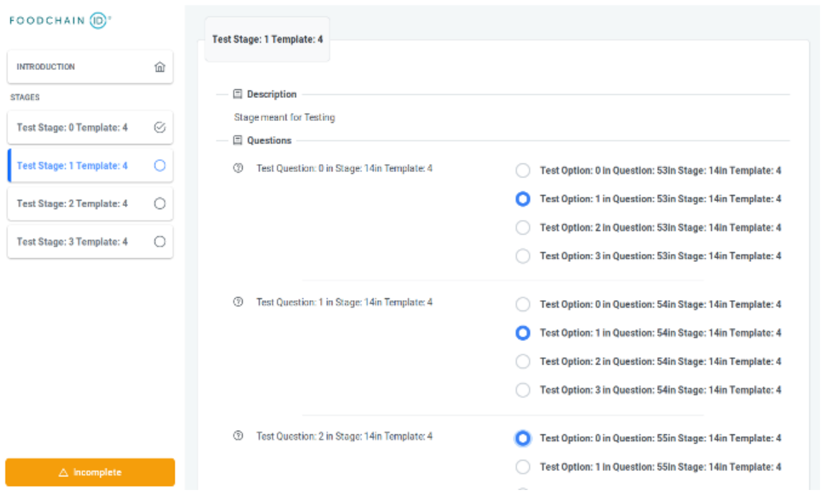
Things To Remember

- Your answers will be used by the inviter to assess your sustainability. Therefore, please answer truthfully.
- You can answer the questionnaire in multiple sessions. Just use the link sent to your email, to get back at answering questions.
- To be able to submit this questionnaire, you must answer all the questions.
- Once submitted you cannot visit the questionnaire again. So be sure to only submit when you are ready.

Submission

Use the Submit button on the left corner of your screen, to submit the questionnaire. Now you know everything and can go answering.

Figure A.10.1: Questionnaire Introduction Page



FOODCHAIN ID

INTRODUCTION

STAGES

Test Stage: 0 Template: 4 ✓
Test Stage: 1 Template: 4
Test Stage: 2 Template: 4
Test Stage: 3 Template: 4

Incomplete

Test Stage: 1 Template: 4

Description

Stage meant for Testing

Questions

Test Question: 0 in Stage: 14in Template: 4
- Test Option: 0 in Question: 53in Stage: 14in Template: 4
- [✓ Test Option: 1 in Question: 53in Stage: 14in Template: 4]
- Test Option: 2 in Question: 53in Stage: 14in Template: 4
- Test Option: 3 in Question: 53in Stage: 14in Template: 4

Test Question: 1 in Stage: 14in Template: 4
- Test Option: 0 in Question: 54in Stage: 14in Template: 4
- [✓ Test Option: 1 in Question: 54in Stage: 14in Template: 4]
- Test Option: 2 in Question: 54in Stage: 14in Template: 4
- Test Option: 3 in Question: 54in Stage: 14in Template: 4

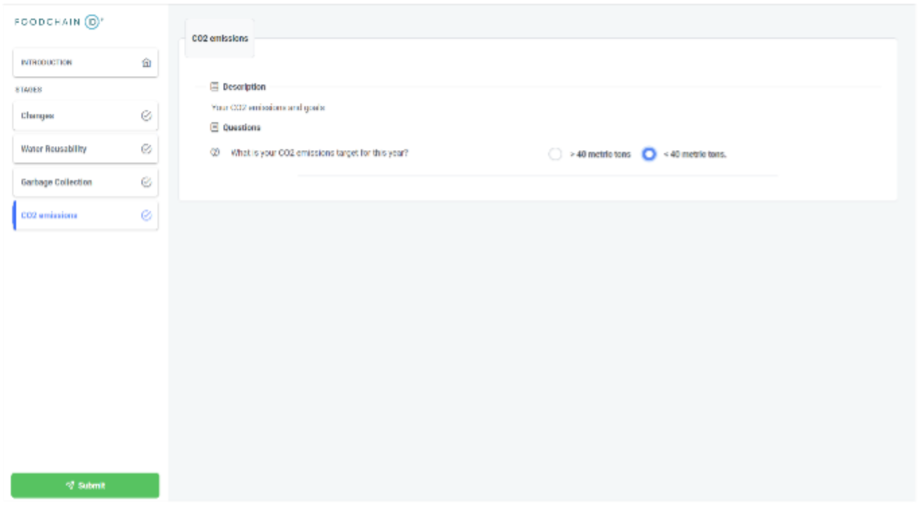
Test Question: 2 in Stage: 14in Template: 4
- [✓ Test Option: 0 in Question: 55in Stage: 14in Template: 4]
- Test Option: 1 in Question: 55in Stage: 14in Template: 4

Figure A.10.2: Stage Answering

SUBMIT QUESTIONNAIRE

The “Incomplete Button” shows the current status of the questionnaire, whenever the questionnaire is auto saved, the button is changed to reflect and notify the user as it can be seen in Figure A.11.

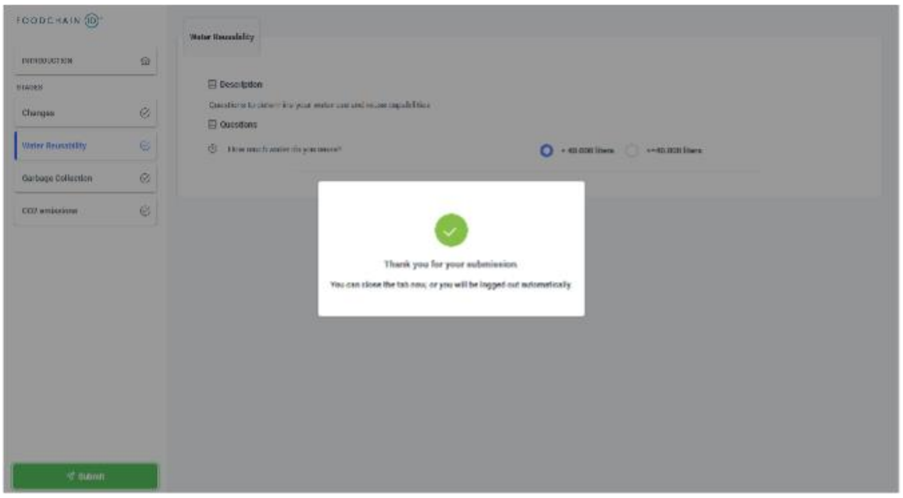
Lastly, upon submission with the Submit button located at the bottom left of the page, the user is shown the screen in Figure A.11.1 and is logged out of the platform.



Your CO2 emissions and goals

What is your CO2 emissions target for this year? > 40 metric tons [✓ < 40 metric tons.]

Figure A.11: Questionnaire Submission Button Active



Questions to determine your water use and reuse capabilities

How much water do you reuse? [✓ < 40.000 liters] >=40.000 liters

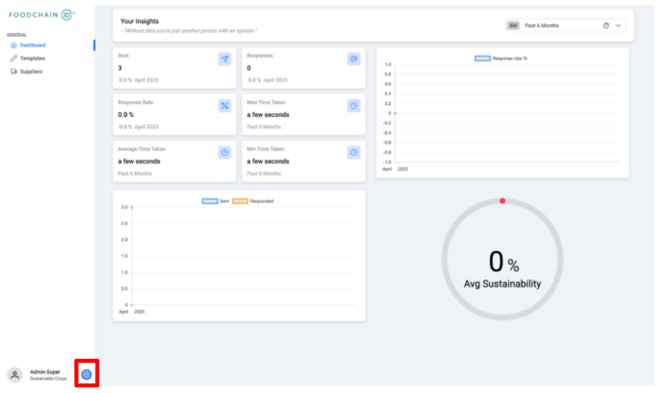
Thank you for your submission

You can close the tab now, or you will be logged out automatically

Figure A.11.1: Questionnaire Submission

LOGOUT FROM THE SYSTEM

To be able to logout from the system, there is a setting icon which can be seen in the bottom left of the page which is specified with a red box in Figure A.12. In the next figure, Figure A.12.1, the button for logging out has been shown.



Your Insights
- "Without data you're just another person with an opinion."
6M Past 6 Months

Sent
3
0.0 % April 2023

Responses
0
0.0 % April 2023

Response Rate
0.0 %
0.0 % April 2023

Max Time Taken
a few seconds
Past 6 Months

Average Time Taken
a few seconds
Past 6 Months

Min Time Taken
a few seconds
Past 6 Months

0 %
Avg Sustainability

Figure A.12: Menu for logging out

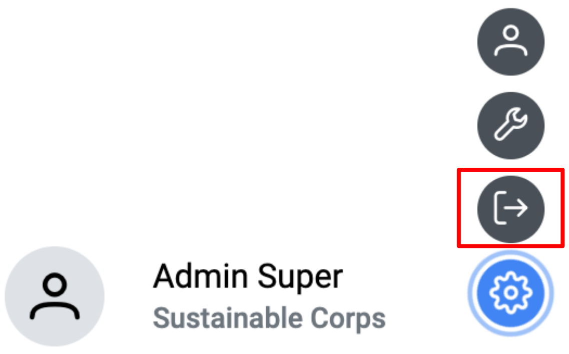


Figure A.12.1: Logout button