FortisBC Emergency Form

Group 13: Isa Nafieiev, Yurii Huba, Kirill Markin, Ivan Postolov, Damir Zharikessov

Client: FortisBC, Amir Kbah

Date: 07/18/2023

Abstract

The Gas Leak Reporting System serves FortisBC clients to efficiently report gas leaks or emergencies. The login page now offers two distinct options: one for registered users, whose datais retrieved from FortisBC's database, and another for non-registered users, who will need to input their information for the FortisBC emergency group.

During form submission, users encounter two sets of questions related to meter hazards and appliance hazards. Upon completion, users are directed to a final page featuring an 'additionalinformation' text box and a submit button. The system's streamlined interface and dynamic questionnaires further enhance the user experience, ensuring prompt and accurate reporting toFortisBC.

Customer

The intended primary user of this software is FortisBC. The system's primary objective is to offerFortisBC clients an efficient and user-friendly platform to promptly report gas leaks or emergencies.

Iteration 1:

In this iteration we followed an Agile methodology, also incorporating pair programming.

We conducted regular group meetings with our client on Mondays to discuss the week's tasks and ensure alignment. Additionally, we prioritized creating a landing page for non-users to entertheir information and proceed to the form filling process.

On the form page users can see personal information they provided such as name, last name, phone number, and address. The form also includes a series of questions related to the type of emergency and the nature of the problem.

The page has a progress bar to indicate the completion status of the form. It features a user-friendly interface with a logo displayed at the top.

The questions on the form are organized into different sections based on the type of emergency, such as fire department/first response or general public, and the type of building involved, such as public use or residential/private use. We also implemented a decision tree that dynamically

adjusts the questions based on the user's answers. This means that as users select their responses,additional questions specific to their situation will appear or disappear. For example, if a user indicates a meter hazard as the nature of the emergency, a set of questions related to the meter will be displayed. Similarly, if an appliance-related emergency is selected, specific questions about the appliance will be shown.

The page aims to gather essential information from users to help emergency responders betterunderstand and address the situation. It uses a simple and intuitive design, making it easy for users to provide the necessary details in case of an emergency.

At the end of Iteration 1, we will conduct a retrospective session where we will discuss the progress made, lessons learned, and gather feedback from all team members. This will ensure that we continuously improve our workflow and address any concerns or suggestions raised bythe team.

Please note that the attached "mockup.png" file in this folder provides a visual representation ofthe user interface design for your reference.

Iteration 2:

In the second iteration of our web application project, we made significant progress in enhancingthe user experience, streamlining data collection, and implementing important functionalities.
Building upon the foundation laid in the first iteration, we focused on finalizing the form logic, enabling automated email communications, improving the login system, and creating user- specific features. We completed the implementation of the form logic and modals. The dynamicdecision tree, which was introduced in the first iteration, is now fully functional and seamlessly adjusts the questions based on the user's responses. Users will experience a smoother and more tailored form-filling process, as the questions displayed precisely match the nature of their emergency situation. This ensures that we gather the most relevant and essential information from the users to assist emergency responders effectively. We added a new feature that allows users to generate a PDF document containing all the questions they answered in the form. This PDF serves as a summary of the information provided and can be saved or printed for users' reference or sharing with emergency responders. The PDF creation process is straightforward and easily accessible on the form completion page. To enhance communication and ensure timely updates, we implemented an automated email system. Users will now receive confirmation emails upon submitting the form, providing them with reassurance that their information has been successfully received. Additionally, the system will automatically send relevant information to the appropriate emergency responders, enabling them to prepare and respond efficiently. To guide users through the form-filling process and make it even more user- friendly, we created a comprehensive PDF document containing step-by-step instructions. When users access the application, this PDF opens in a new page, ensuring that users have clear guidance throughout the entire process. The user instructions PDF is designed to be concise and visually engaging, making it easier for users to

understand and follow. To facilitate the automated email functionality, we set up an email API server. This server handles the generation and delivery of emails to the appropriate recipients, ensuring reliable and efficient communication. The email API server integrates seamlessly with the web application, allowing for seamless email management without impacting the overall system performance. To provide users with more control over their information, we implemented user controllers. This feature enables users to access and edit their personal details and preferences, such as name, contact information, and emergency contact details. Users can now make updates to their profiles, ensuring that their information is always accurate and up to date. Moreover, we introduced dynamic topic updates for the automated emails. This means that the content of the email sent to emergency responders adapts based on the type of emergency and the information provided by the user. This ensures that the responders receive clear and relevant information, improving their preparedness and response. We made significant enhancements to the login system based on the feedback received during the retrospective session at the end of the first iteration. The error messages have been revised to be more informative and user-friendly, making the login process smoother and more intuitive. Users can now access their accounts securely and conveniently, ensuring a seamless experience throughout their interaction with the application. By the end of the second iteration, we have made substantial progress in expanding the functionality and user experience of our web application. The implemented features, such as automated emails, dynamic questionnaires, and user-specific controls, will contribute significantly to a more efficient and effective emergency response process. We are committed to continuously improvingour application and look forward to conducting another retrospective session to gather feedback and insights for further enhancement in the upcoming iteration.

Iteration 3:

In the 3rd iteration of our web application project, we focused on enhancing the user experience and implementing critical functionalities. We successfully enabled Google and Facebook login options, allowing users to access the system conveniently with their existing credentials. The system now verifies Google or Facebook accounts using the respective APIs and pre-fills the form with relevant data from the user's Fortis account if available.

We also introduced a contact info verification feature, allowing both Regular Users and Non-Users to review and edit their contact information before submitting the gas leak reporting form. For Non-Users, any changes made during verification are updated in the database and form summary. For Regular Users, changes are only applied to the form summary.

To improve tracking and organization, we implemented a unique form identifier. Each submitted form now receives a distinct form ID, which is recorded along with submission time and user ID in a separate database table. Additionally, all summaries sent to Fortis managers include both user and form IDs for easy reference.

During this iteration, we addressed a bug related to Non-Users' phone numbers. The bug was fixed, ensuring that phone number changes are correctly updated for Non-Users in the form

submitted.

The application has made significant progress in expanding functionality, user experience, and efficiency in the emergency response process.

Competitive Analysis

The Gas Leak Reporting System continues to stand out among existing solutions with its ongoing commitment to providing a streamlined and intuitive user interface. The system now offers personalized login options, empowering users to access their accounts securely and conveniently. Furthermore, the introduction of tailored questionnaires based on the type of emergency further enhances the system's ability to gather relevant and precise information. By incorporating these advancements, our system continues to improve the efficiency and accuracyof incident reporting to FortisBC, enabling prompt and effective response measures.

User stories

Login and Signup

Description: As a User or Non-User, I want to access a secure and personalized experience by either logging in with my existing credentials or signing up for a new account before entering thegas leak reporting form.

Actors: User, NonUser

Actions:

1. The User or Non-User accesses the Gas Leak Reporting System.
2. If the User is already registered, they enter their existing credentials (username andpassword) to log in.
3. If the Non-User does not have an account, they select the signup option and provide thenecessary information to create a new account.
4. After successful login or signup, the User or Non-User gains access to their personalizeddashboard.

Preconditions:

- The User or Non-User accesses the Gas Leak Reporting System.
- The login and signup functionalities have been implemented.

Postconditions:

- The User or Non-User gains access to the gas leak reporting form aftersuccessful login or

signup.

System Tests:

- User provides correct credentials and is redirected to their dashboard.
- Non-User successfully completes the signup process and is directed to their dashboard.

Enhanced Form Logic

Description: As a Regular User or Non-User, I want the gas leak reporting form to dynamically adjust the questions based on my responses, so I only see relevant questions tailored to my specific emergency situation.

Actors: Regular User, NonUser

Actions:

1. The user enters the gas leak reporting form.
2. The system presents a series of questions related to the type of emergency and the natureof the problem.
3. As the user selects their responses, the form dynamically adjusts, showing or hidingadditional questions specific to their situation.

Preconditions:

- The user or non-user accesses the gas leak reporting form.
- The form logic and decision tree have been finalized and implemented.

Postconditions:

- The user or non-user completes the form with relevant and essential informationbased on their specific emergency situation.

System Tests:

- User selects "Meter Hazard" as the nature of the emergency, and additional questions relatedto meters appear.
- User chooses "Appliance-Related Emergency," and specific questions about the appliance aredisplayed.

User Instructions PDF

Description: As a Regular User or Non-User, I want to receive instructions regarding my future actions. Also, I want to know future steps FortisBC will take in order to resolve my emergency

situation.

Actors: Regular User, Non-User

Actions:

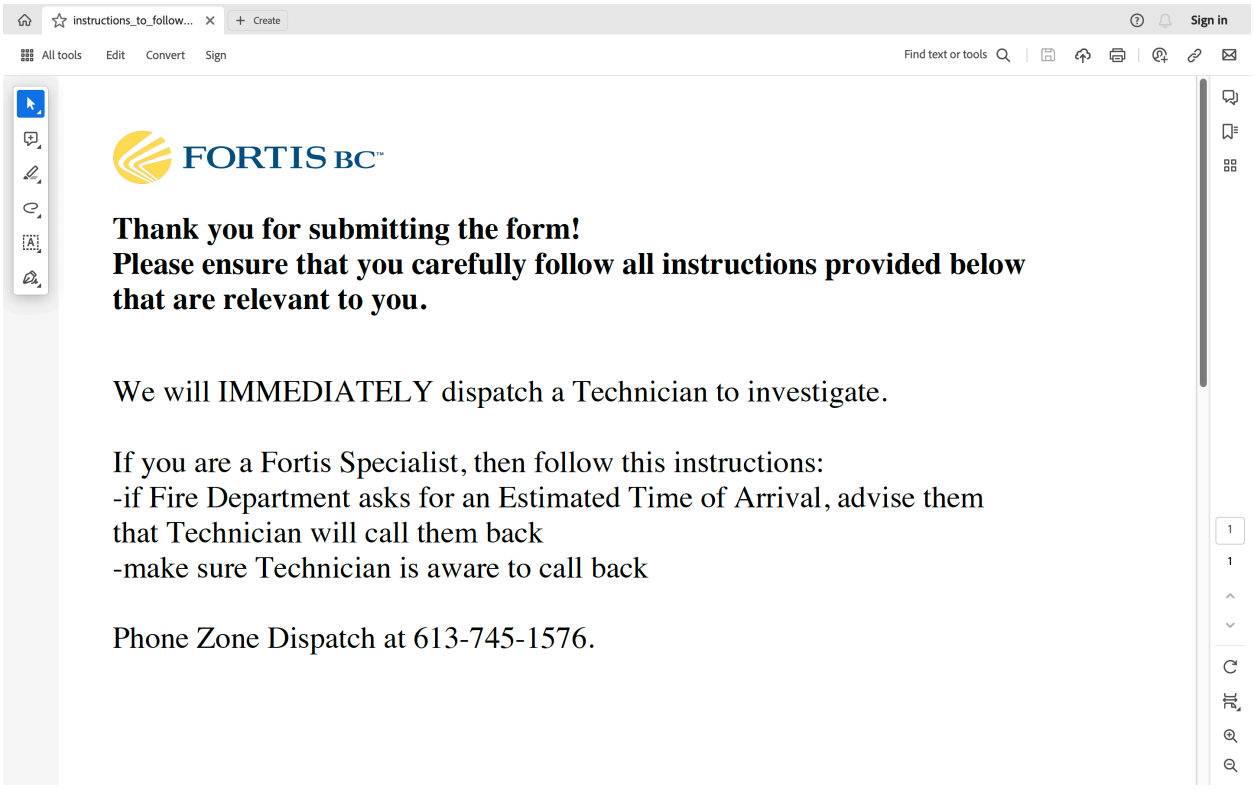
1. The user or non-user completes and submits the gas leak reporting form.
1. The system presents a user instructions PDF in a new page with step-by-step guidance onfilling the gas leak reporting form.
2. The system saves PDF file to the user device .

Preconditions: The user or non-user submits the gas leak reporting form.

Postconditions: The user or non-user gains clear and concise guidance on further clearance

System Tests:

- User opens the user instructions PDF and verifies that it containscomprehensive guidance on form completion.



Dynamic Topic Updates and Email sending

Description: As a Regular User or Non-User, I want the automated email content sent to emergency responders to adapt based on the type of emergency and the information provided, ensuring responders receive relevant and clear details.

Actors: Regular User, Non-User

Actions:

1. The user or non-user submits the gas leak reporting form.
2. The system generates an automated PDF file.
3. The content of the PDF adapts based on the type of emergency and user-provided
4. The system generates an automated email for emergency responders.
5. Email topic contains code with a encoded type of emergency for the emergency operator

Preconditions:

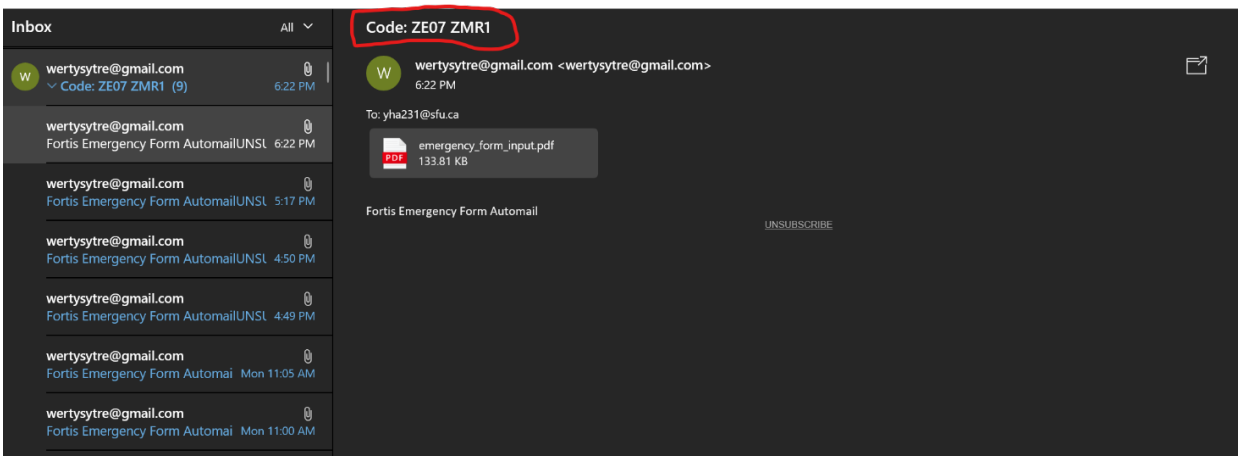
- The user or non-user completes the gas leak reporting form.
- The email API server is set up to handle dynamic content.

Postconditions:

- Emergency responders receive an email with content specific to the reported emergency

System tests:

- User submits the form, and responders receive an email tailored to the type ofemergency and provided information.



Personal Details Editing

Description: As a NonUser or User, I want the ability to edit my personal details and preferences, in case I provided wrong information while signing up or I'm currently witnessing an emergency in someone else's house.

Actors: Regular User, NonUser

Actions:

1. The Regular User logs in to their account or NonUser signs up and proceed to the form.
2. The system allows the user to modify their personal information, such last name, contact details, and address while filling out the form.

Preconditions:

- The emergency's witness logs into the form.
- User controllers for accessing and editing personal information have been implemented

Postconditions:

- The correct contact details are provided to emergency responders

System Tests:

- User logs in and verifies that they can modify their personal information asdesired.

Facebook and Google Login

Description: as a Regular User, I want to access the system using my Google or Facebook account, so I can conveniently log in with my existing credentials.

Actors: Regular User

Actions:

1. The user chooses the login option using Google or Facebook.
2. The user enters their email and password for the selected account.
3. The system verifies the Google or Facebook account using the corresponding API.
4. The system checks if the entered email exists in the Fortis database.
5. If the email is found in the database, the form is prefilled with the corresponding data.
6. If the email is not found, the user proceeds to fill the form manually.

Preconditions:

- The user accesses the gas leak reporting form.
- The user has an account registered with Google or Facebook.
- The user has an account with Fortis registered on the same email as the Google or Facebook account.

Postconditions:

- If the user's email is found in the Fortis database, the form is prefilled with the corresponding data.
- If the user's email is not found, the user proceeds with filling the form manually.

Contact Info Verification

Description: As a Regular User or Non-User, I want to review and edit my contact information before submitting the emergency reporting form.

Actors: Regular User, Non-User

Actions:

1. The user triggers one of the final questions of the form.
2. A pop-up appears, allowing the user to edit their contact info if needed.
3. For Non-Users, the system updates their data in both the database and form summary upon form submission.
4. For Regular Users, the system updates only the form summary with any changes made during the last-minute verification.

Preconditions:

- The user or Non-User accesses the gas leak reporting form.
- The user triggers the final question.
- The user submits the form successfully.

Postconditions:

- Any changes made during the verification are appropriately reflected in the database and form summary.

Form Unique Identifier

Description: As a Regular User, I want my submitted form to have a unique identifier in the database for dispatcher’s easy identification and tracking.

Actors: Regular User

Actions:

1. The user submits the form.
2. The system assigns a unique identifier (form ID) to the submitted form.
3. The system records the form ID, submission time, and user ID in a separate table in the database.
4. All summaries sent to Fortis managers include both the user ID and form ID for easy referencing.

Preconditions:

- The user accesses the gas leak reporting form.
- The user submits the form successfully.

Postconditions:

- All forms submitted by users of Fortis are recorded in a separate table in the database with a unique form ID.
- The form ID is included in the summary of the submitted form sent to Fortis managers.

Acceptance Test for the Whole Project:

1. Create test cases for each user story and ensure all acceptance criteria are met.
2. Perform integration tests to ensure smooth interaction between different features and components.
3. Conduct end-to-end testing of the application to verify the entire user journey, including login, form filling, verification, and form submission.
4. Test various scenarios, including successful form submissions, form submissions with contact info changes, and form submissions with new user registrations.
5. Verify that automated emails are sent correctly to users upon form submission and to emergency responders with dynamic content based on the user's responses.
6. Validate the PDF generation functionality and ensure that users can save or print their form summaries successfully.
7. Perform load testing to assess the application's performance under various user loads.
8. Conduct security testing to identify and address potential vulnerabilities.
9. Gather feedback from users to ensure that the application meets their needs and expectations.
10. Conduct a retrospective session to collect insights and identify areas for further improvement in future iterations.

Velocity Discussion

First iteration

We collected initial stories from the client (intense interaction with customer) and implemented stories, that lied in the foundation of the project. Overall, the velocity of the first iteration was rather slow, which can be explained by high level of interactions with client and issues related to equal contribution of the team members. Also first iteration didn’t include many tests, which was caused by uncertainty in the requirements of the client.

Second iteration

We finalized the requirements from the client and started working on the created foundation. Learning from our mistakes allowed to speed up project velocity, so that we were able to finish most of the features initially required by the client. Foundation that was lied in the first iteration allowed us to make great visible progress in the second.

Third Iteration

Third iteration kicked off with client enhancing our project scope with new stories, which we were able to successfully implement in addition to finishing all stories that we didn’t cover in 2 iteration. Large portion of 3-rd iteration was dedicated to acceptance testing (contract

requirements satisfaction).
3rd iteration had peak velocity of our project, because we had to finish the implementation of newly discussed features.

Distribution of work in a team

Yurii -- An easy-to-use form that explains every feature using radio-buttons. Client interaction and creation of user mock-ups. Finalized the full logic of the form questions and final modals. Also, worked on form questions PDF creation (in collaboration with Ivan), automated email logic (in collaboration with Ivan), created PDF for user instructions, set up an email API server for automated emails, edited documentation, updated UI mock-ups, and was responsible for client interaction, pair programmed Google with Ivan and finized Facebook logins and finalized workflow diagram and other final documentation, finalized progress bar, finalized ending design of the form with Ivan.

Ivan – Integrating a database to securely store user authentication information and form submissions. Collaborated with Yurii on automated email logic, updated classes and controllers for users, implemented dynamic editing of user fields, and set up email dynamic topic updates in the email and added full logic of user login, also added form reset. Additionally, worked on the final documentation and assisted in client interaction, pair programmed Google login, finished form id logic and storing finalized ending design of the form with Yurii.

Kirill – Designing a visually appealing user interface using CSS frameworks or libraries. Started progress bar and logic of the form final modals with instructions. Finished last minute check pop up of the form.

Isa – Created the dictionary for dialog boxes of instructions for users.

Damir – Worked on documentation and helped with Facebook login.