Toronto Hydro
500 Commissioners Street

Tel: 416 - 542 - 8000
Tel: 416 - 542 - 8000

ITM900 - Final Report
Automated Document Completion and Email Generation System

Hunter Moore - 501112173
Abdalla Elgharbawy - 501037562
Omar Abu Awwad - 501092802
Faras Kariminjead Nobary - 501156443
Omar Almneni - 501080814

Table of Content

Project Management

Project Charter:

Project Description

Toronto Hydro currently relies on a manual process to generate Plant Damage Reports, which document incidents where underground cables are damaged during construction. The current workflow requires dispatchers to fill out Word documents manually, copy and paste recipient email lists, attach the document, and send emails manually. This process is inefficient and results in redundant steps and wasted time.

This project aims to develop a streamlined document completion and email dispatch system that improves the efficiency of the Plant Damage Report process. Instead of fully automating form completion, the focus is on creating a standardized fillable template that integrates smoothly with Outlook for quicker email dispatching.

The system will:

1. Provide a Fillable Word Template – A structured form letter that dispatchers can manually complete.
2. Maintain an Updatable Recipient List – A separate, easily modifiable field that keeps recipient information current.
3. Enable Automated Email Formatting & Dispatch – Once filled, the system will generate an email-ready document and automatically populate an email with the recipient list to reduce manual steps.
4. Ensure Seamless Outlook Integration – Allow dispatchers to send reports directly from the system without copying and pasting recipients manually.
5. Reduce Clicks & Improve Workflow – Focus on eliminating redundant steps to minimize dispatcher effort and save time.

The project will result in a proof-of-concept tool that simplifies document handling and email dispatch, significantly reducing the administrative burden on Toronto Hydro's dispatchers.

Project Justification

Purpose

Toronto Hydro’s dispatchers handle Plant Damage Reports frequently, requiring them to generate, attach, and send reports multiple times per day. The current process is manual and inefficient which is leading to:

- Increased workload for dispatchers due to redundant manual steps.
- Higher risk of human errors (copy-pasting mistakes, outdated recipient lists, incorrect document attachments).
- Inefficiencies in reporting and communication when responding to underground cable damage incidents.

This project seeks to:

- Increase processing speed by reducing the number of steps needed to complete and send a report.\
- Minimize human error by improving recipient list management and automating email formatting.
- Enhance dispatcher productivity by reducing time spent on repetitive tasks.
- Ensure accurate and up-to-date reporting to improve response efficiency.

Business Case

Problem Statement

Toronto Hydro is Toronto's main electricity distributor. They own and operate a system that delivers electricity to around 793,000 customers. Toronto Hydro’s current document management system requires significant manual effort, resulting in time inefficiencies, higher costs, and delays in reporting. Toronto Hydro dispatchers currently complete and send Plant Damage Reports manually which require them to open and manually fill a Word document, copy and paste the recipient list from previous emails, manually attach the report and compose the email, and send the email through multiple steps in Outlook. This process results in time inefficiencies, human errors, and inconsistencies in reporting.

Proposed Solution

This project will streamline the document completion and email dispatch process by providing a structured, fillable Word template to standardize data entry, automating email generation and recipient list management to eliminate copy-pasting errors, and integrating with Outlook for seamless dispatch of reports with minimal manual effort.

Key Benefits

1. Efficiency Gains – Reducing manual steps in completing and sending reports.
2. Error Reduction – Eliminating copy-paste mistakes and outdated recipient lists.
3. Productivity Improvement – Allowing dispatchers to process reports faster with fewer clicks.
4. Better Communication Flow – Ensuring reports are sent to the right recipients consistently and on time.

Project Description

High-level description of the project

Our high-level activity scope involves a comprehensive and collaborative approach to automating document completion processes. The Automated Document Completion System is designed to reduce manual effort and streamline reporting workflows for Toronto Hydro’s administrative processes.

A. In-depth Consultative Sessions with Toronto Hydro Personnel

We begin by sitting down with the team at Toronto Hydro’s to gather insights into the document completion process. This step is crucial to the success of the project, as these personnel possess intimate knowledge of day-to-day operations and the challenges associated with manual data entry. By actively engaging with key stakeholders, we will gain a good understanding of the specific bottlenecks they are facing, which will ensure that our solution is effective and aligned with operational needs.

B. Strategic Allocation of Work to Development Teams

Once we gather insights from Toronto Hydro staff, we assign specific tasks to our team based on their strengths and expertise. This assignment is strategic, ensuring that each of our team members is well-equipped to handle their specific portion of the project. Our team will conduct further research and analyze document structures with existing databases. This will help develop a scalable, future-proof system that enhances Toronto Hydro's document processing.

C. Structured Stakeholder Sessions

Weekly meetings with the Toronto Hydro Team, and our team. To discuss

D. Core Functionalities of the System

The Streamlined Plant Damage Report System will include:

1. Fillable Word Form Template – A structured, easy-to-use template where dispatchers can quickly input key information such as customer address, event number, contractor details, and a summary of the issue.
2. Automated Email Formatting & Dispatch – Once the form is completed, the system will automatically generate an email with the report attached and send it through Outlook without requiring manual recipient entry.
3. Updatable Recipient List Management – A dedicated field within the system that allows dispatchers to modify the recipient list dynamically which will ensure emails are always sent to the correct contacts without manually copying and pasting addresses.
4. Seamless Outlook Integration – The system will integrate with Outlook to streamline email dispatching, allowing dispatchers to send reports with a single click while maintaining compliance with Toronto Hydro's IT security constraints.

3. Project Objectives

A list of (SMART) objectives for the project.

Reduce Manual Effort in Document Processing

- Specific: Automate the completion of high-volume documents, including incident reports and administrative forms.
- Measurable: Achieve a 50% reduction in manual document processing time within 6 months of deployment.
- Achievable: The system will extract and autofill key fields, reducing the need for manual entry.
- Relevant: Reducing manual workload allows frontline workers to focus on operational priorities.
- Time-bound: Develop proof of concept within 3 months, ensuring measurable impact.

Business Objectives

The key objectives of the project are:

1. Streamline the Plant Damage Report Process

- Objective: Develop a system that simplifies and standardizes the document completion process, making it more efficient and user-friendly.
  Success Criteria: Reduce the time required to complete and send reports

2. Improve Data Accuracy and Communication Efficiency

- Objective: Minimize errors in report completion and ensure that recipient lists remain accurate and up to date.
  Success Criteria: Reduce reporting errors (such as incorrect recipient entries or missing information)

3. Reduce Administrative Burden on Dispatchers

- Objective: Optimize the workflow by eliminating redundant steps in the document handling and email dispatch process.
  Success Criteria: Reduce the number of manual actions (clicks, copy-pasting, email formatting) per report.

4. Ensure Seamless Outlook Integration

- Objective: Enable dispatchers to generate and send reports directly through Outlook without needing to manually format emails or manage recipient lists.
  Success Criteria:

5. Provide a Flexible and Scalable Solution

- Objective: Develop a system that can be easily modified or expanded to accommodate future workflow improvements.
  Success Criteria: Deliver a detailed blueprint outlining system architecture, user workflows, and potential future enhancements to Toronto Hydro for continued improvement.

Requirements

The functional requirements are:

1. Automated Form Completion: The system must pre-fill designated fields in incident reports using existing data.
2. Customizable Recipient List: The system must allow dispatchers to update and maintain recipient lists easily.
3. Email Integration: The system must automatically generate and send emails based on the filled form.
4. User Input Fields: Users must be able to manually enter additional details if needed.
5. Data Validation: The system must ensure required fields (e.g., event number, customer address) are filled before submission.
6. Report Generation: The system should generate and store a copy of each completed form
7. BPMN Workflow Alignment: The system should align with Toronto Hydro’s dispatching workflow to avoid unnecessary steps
8. Standardized Output Format: Completed forms must be in a structured format (e.g., Word, PDF) that integrates smoothly into existing processes.

The non-functional requirements are:

1. Ease of Use: The system must have a user-friendly interface requiring minimal training.
2. Security Compliance: Must comply with Toronto Hydro's IT security policies (e.g., no direct integration with Oracle NMS).
3. Performance & Efficiency: The system should reduce the number of clicks and manual steps, improving workflow efficiency.
4. Reliability & Availability: The system should be accessible at all times without frequent failures.
5. Scalability: The solution should be easily adaptable for potential future use in other departmental workflows in Toronto Hydro.
6. Compliance with Internal & Regulatory Standards: Must meet all company-specific document handling policies.

Constraints define the boundaries within which the project must operate.

The main limitations and restrictions in this project are:

1. **Time**
   - The automation system must be completed within the capstone project timeline.
   - Meetings with Toronto Hydro are bi-weekly, meaning feedback may be delayed.
2. **Resource**
   - The project must be developed by our capstone group without external developers.
   - Limited access to Toronto Hydro's internal systems (e.g., NMS cannot be integrated for security reasons).
   - Must use widely available software (e.g., Microsoft Word, Outlook, Visio**)** instead of custom enterprise solutions.
3. **Cost**
   - The system must be developed with free or existing tools (e.g., Google Apps and Microsoft Word Suite)
4. **Quality**
   - The form must be accurately populated with minimal errors to ensure compliance with Toronto Hydro reporting standards.
   - The recipient list must be updated dynamically, ensuring all relevant personnel receive reports without delays.
5. **Technical**
   - The system cannot integrate with Toronto Hydro's NMS system due to security restrictions.
   - Data storage options are limited—completed reports must be stored in a secure but accessible format (e.g., PDFs in a shared directory)
   - This solution must be compatible with Microsoft Outlook for streamlined email dispatching.

Acceptance Criteria

The Following criteria outlines the conditions which help determine whether the project is successful or not.

1. Automated Document & Email fill out
   a. The system must generate a well structured word document that would only require the Staff/ Dispatchers to only have to fill out the necessary information efficiently/ effectively.

   b. The automated system must ensure that the completed reports should be automatically formatted and attached to the email with little to no manual steps whatsoever.
2. Efficiency increase & Error reduction
   a. The use of this automated system must reduce the amount of clicks the the Staff/ Dispatchers have to go through to complete the form by at least 40%.
   b. The system should ensure that it stays up to date with the necessary information used throughout this process such as the use of correct recipient emails and to ensure that no spelling mistakes are made.
3. Reliability & Scalability
   a. The system should be able to accurately/ effectively operate with minimal to no down time/ delays whatsoever to ensure for proper functionality of the system.
   b. The system must be set up in a way to ensure that it will be ready to handle any future enhancements/ upgrades to be able to keep up with the necessary workflows.

Assumptions

The following assumptions are made in the development stage to help identify the project's scope, feasibility, and most importantly the expected outcome.

1. User & Training Adaptation
   a. Staff/ Dispatchers will be open to the idea of undergoing a successful integration for the transition from manual to an automated system that will reduce the amount of required work from the Staff/ Dispatchers.
   b. The interface and accessibility will require little to no additional training for what the Staff/ Dispatchers already knows.
2. Technical Viability
   a. Microsoft Word & Outlook will need to provide us with the necessary automation capability to help allow for a functional email and fill out automation system.
   b. No external third-party software will be required for the development/ implementation of the automation system.
3. Security Compliance
   a. The automated system must comply with Toronto Hydro's IT security policies, ensuring that the data and privacy aren't negatively impacted in any way shape or form.
   b. The email recipient list will only be restricted to authorized personnel that do their jobs.

Initial Risks

Potential risks that may affect the chances of making this project successful.

1. Technical Limitations
   a. Microsoft tools may have restrictions as to what can be done during the email automation based on the restrictions placed Word customization
   b. Different versions of microsoft tools can cause issues for the automated fill out options.
2. Workflow Restrictions
   a. Some dispatchers may find it difficult to undergo the changes from a manual process to an automated process requiring additional training which will cause additional charges to the company.
   b. Unforeseen problems can occur within the work environment during the implementation/ development stages.
3. IT Restrictions

a. Current IT security policies may cause restrictions on the implementation of the automated system requiring a modification to the current system.
b. Limited access to Toronto Hydro's internal systems may cause issues towards the implementation and development stages of the project.

Milestone Schedule

- Project Initiation (Week 1-2, 6 days)
  Set project scope and objectives.
  Conduct first meetings with Toronto Hydro stakeholders.
  Define requirements and limitations.

- Requirements Gathering & Analysis (Week 2-3, 6 days)
  Conduct consultative meetings with dispatchers and end users.
  Document workflow inefficiencies and process bottlenecks.

- System Design & Development (Week 3-7, 21 days)
  Develop a fillable Word template with form.
  Automate email formatting and recipient list maintenance.
  Integrate the system with Outlook for seamless dispatch.

- Testing & Refinement (Week 7-9, 11 days)
  Conduct internal testing to identify potential system errors.
  Get feedback from Toronto Hydro dispatchers for the purpose of improving.

- Deployment & Training (Week 9-11, 11 days)
  Finalize the entire system installation.
  Provide end-user training.
  Collect final feedback and perfect the workflow.

- Project Closure (Week 12, 6 days)
  Obtain final acceptance from Toronto Hydro.
  Submit final documents.
  Review success of project versus original goals.

Budget Summary

As this is a capstone project with no budget for third-party tools, cost assumptions revolve around utilizing available resources:

Software & Tools:
Microsoft Word & Outlook (Available through Toronto Hydro's existing IT infrastructure)
No third-party API or software licensing fees.

Human Resources:
Project work performed by student team members with no external developers.

No additional payroll expenses.

Training & Implementation:
Performed in-house with no additional costs.
Stakeholder training through online sessions.

Total Estimated Budget: $0 (Utilizing free or available tools)

Project Manager
Project Manager: Omar Almneni
Responsible for maintaining the project schedule and deliverables.
Ensures the project is consistent with Toronto Hydro's objectives.
Is responsible for conflict resolution, risk management, and stakeholder communications.
Coordinates the efforts of teams to complete milestones effectively.

Conflict Resolution

Effective conflict resolution is essential to ensuring the success of this project. Given the collaborative nature of this initiative, conflicts may arise due to differences in expectations, resource constraints, or technical disagreements. The following conflict resolution strategy will be implemented:

1. **Early Identification & Open Communication**
   - Team members will be encouraged to identify and report conflicts as soon as they arise.
   - Open and respectful communication will be prioritized to address concerns constructively.

2. **Escalation Process**
   - Step 1: Conflicts will first be resolved informally through direct discussions among the involved parties.
   - Step 2: If unresolved, the matter will be brought to the Project Manager, who will mediate a resolution.
   - Step 3: If further escalation is needed, the conflict will be addressed in a team meeting for collective input and resolution.
   - Step 4: If the issue remains unresolved, it will be escalated to Toronto Hydro project stakeholders or sponsors for final arbitration.

3. **Consensus Type Decision Making**
   - Whenever possible, decisions will be made through consensus to ensure all team members are aligned with project objectives.

4. **Documenting Resolutions**
   - All major conflict resolutions will be documented to prevent recurring issues and to establish best practices for future projects.

Project Sponsor Approval

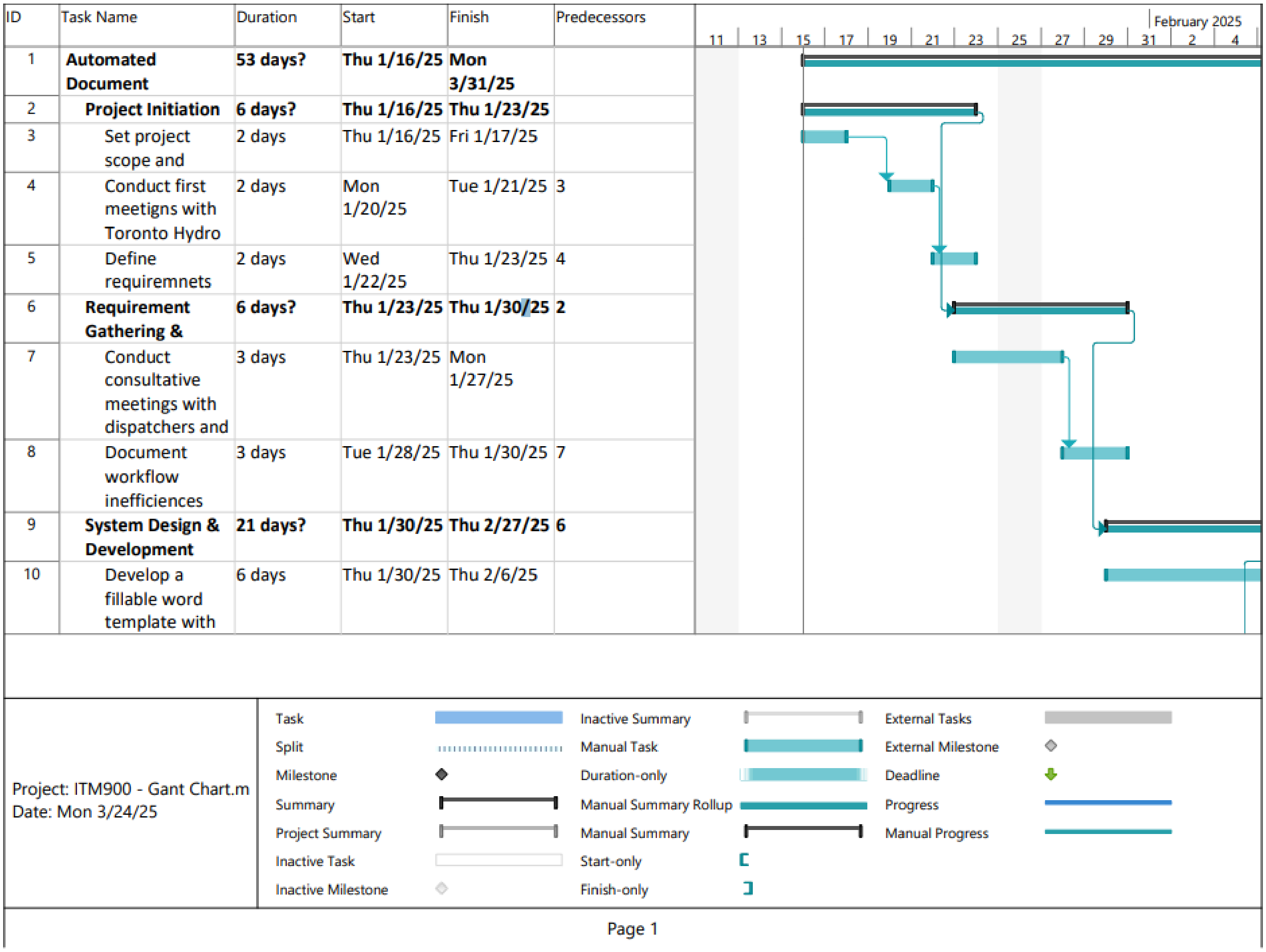
Project Sponsor: *Michael Little (Toronto Hydro - Client Representative)*

Date: ____________________

Approval Signature: Michael Little Digitally signed by Michael Little DN: cn=Michael Little, email=mlittle@torontohydro.com Date: 2025.03.31 11:58:49 -04'00' ____________________________

I hereby acknowledge that I have reviewed and approved this project charter, including the project objectives, scope, and responsibilities outlined herein. By signing this document, I confirm my commitment to supporting the successful execution of the **Automated Document Completion System and Email Generation System** for Toronto Hydro.

Project Schedule (Gantt Chart):



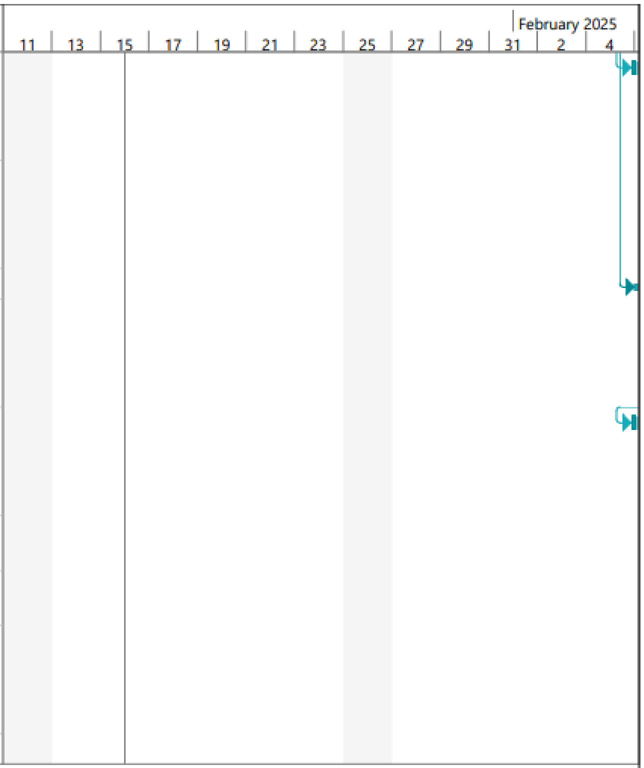
ID	Task Name	Duration	Start	Finish	Predecessors
1	**Automated Document**	**53 days?**	**Thu 1/16/25**	**Mon 3/31/25**	
2	**Project Initiation**	**6 days?**	**Thu 1/16/25**	**Thu 1/23/25**	
3	Set project scope and	2 days	Thu 1/16/25	Fri 1/17/25	
4	Conduct first meetigns with Toronto Hydro	2 days	Mon 1/20/25	Tue 1/21/25	3
5	Define requiremnets	2 days	Wed 1/22/25	Thu 1/23/25	4
6	**Requirement Gathering &**	**6 days?**	**Thu 1/23/25**	**Thu 1/30/25**	**2**
7	Conduct consultative meetings with dispatchers and	3 days	Thu 1/23/25	Mon 1/27/25	
8	Document workflow inefficiences	3 days	Tue 1/28/25	Thu 1/30/25	7
9	**System Design & Development**	**21 days?**	**Thu 1/30/25**	**Thu 2/27/25**	**6**
10	Develop a fillable word template with	6 days	Thu 1/30/25	Thu 2/6/25	

Project: ITM900 - Gant Chart.m
Date: Mon 3/24/25

Task		Inactive Summary		External Tasks	
Split		Manual Task		External Milestone	◇
Milestone	◆	Duration-only		Deadline	⇩
Summary		Manual Summary Rollup		Progress	
Project Summary		Manual Summary		Manual Progress	
Inactive Task		Start-only	[		
Inactive Milestone	◇	Finish-only	]		

Page 1

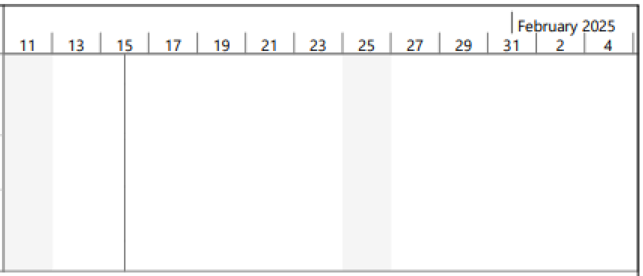
ID	Task Name	Duration	Start	Finish	Predecessors
11	Automate email formatting and recipient list maintenance	6 days	Thu 2/6/25	Thu 2/13/25	10
12	Integrate the system with outlook for seamless	11 days	Thu 2/13/25	Thu 2/27/25	11
13	**Testing & Refinem**	**11 days?**	**Thu 2/27/25**	**Thu 3/13/25**	**9**
14	Conduct internal testing to identify potential	6 days	Thu 2/27/25	Thu 3/6/25	
15	Get feedback from Toronto Hydro dispatchers for	6 days	Thu 2/6/25	Thu 2/13/25	14
16	**Deployment & Training**	**11 days?**	**Thu 3/13/25**	**Thu 3/27/25**	**13**
17	Finalize the system	6 days	Thu 3/13/25	Thu 3/20/25	
18	Collect final feedback and perfect the workflow	6 days	Thu 3/20/25	Thu 3/27/25	
19	**Project Closure**	**3 days?**	**Thu 3/27/25**	**Mon 3/31/25**	**16**



Project: ITM900 - Gant Chart.m
Date: Mon 3/24/25

Task	▬	Inactive Summary	▭	External Tasks	▬
Split	┈	Manual Task	▬	External Milestone	◇
Milestone	◆	Duration-only	▬	Deadline	⬇
Summary	┣━┫	Manual Summary Rollup	▬	Progress	━
Project Summary	┣━┫	Manual Summary	┣━┫	Manual Progress	━
Inactive Task	▭	Start-only	[		
Inactive Milestone	◇	Finish-only	]		

ID	Task Name	Duration	Start	Finish	Predecessors
20	Obtain final acceptance from Toronto	2 days	Thu 3/27/25	Fri 3/28/25	
21	Submit final documents	2 days	Fri 3/28/25	Mon 3/31/25	20
22	Review success of project versus original	1 day?	Fri 3/28/25	Mon 3/31/25	21



Project: ITM900 - Gant Chart.m
Date: Mon 3/24/25

Task	▬	Inactive Summary	▭	External Tasks	▬
Split	┈	Manual Task	▬	External Milestone	◇
Milestone	◆	Duration-only	▬	Deadline	⬇
Summary	━	Manual Summary Rollup	━	Progress	━
Project Summary	━	Manual Summary	━	Manual Progress	━
Inactive Task	▭	Start-only	[		
Inactive Milestone	◇	Finish-only	]		

Identification of the problem (s)/opportunities

Overview of the problem(s)/opportunities:

Toronto Hydro's current workflow for their incident forms presents several problems. These problems include many time-consuming processes, redundant manual tasks, high risk of human errors through repetitive copy-pasting, and an overall inefficient method for dispatching emails. To address these challenges, there are opportunities to automate the report generation using a fillable Word document. This form will significantly enhance efficiency. Additionally implementing an automated email dispatch system would streamline communications.

Problems:

- Time consuming through silly redundant steps.
- Open for human error while copy pasting.
- There can be inconsistencies with the formatting.
- Inefficient way of dispatching the emails.

Opportunities:

- To automate the report generation process with a fillable word document to ensure effectiveness and efficiency.
- To integrate an automated email dispatch system.
- Ensures that there is a standardized system that eliminates most of the issues that humans are prone to make.
- Most importantly it reduces the workload for employees over time and enhances productivity.

Diagnostic (causes-problem-consequences):

The primary causes of issues in the current Plant Damage Report include manual procedures where dispatchers are required to fill out forms and send emails manually, increasing the likelihood of errors. These root causes contribute to significant problems. Problems that include time consuming processes like tedious and redundant steps. These manual steps result in increased workload which accumulates into a substantial time commitment. **Causes:**

- Manual Process: Dispatchers are required to manually fill the form out and send the emails.
- Copy Paste Error: There isn’t an automated recipient list management system.
- Lack of Standardization: This is the cause for an inconsistent report / email generation formatting and for the workflow.

Problems:

- The current Plant Damage Report Process is time consuming due to tedious steps that can be easily overcome, Prone to a lot of errors, and most importantly inefficient to the work process.

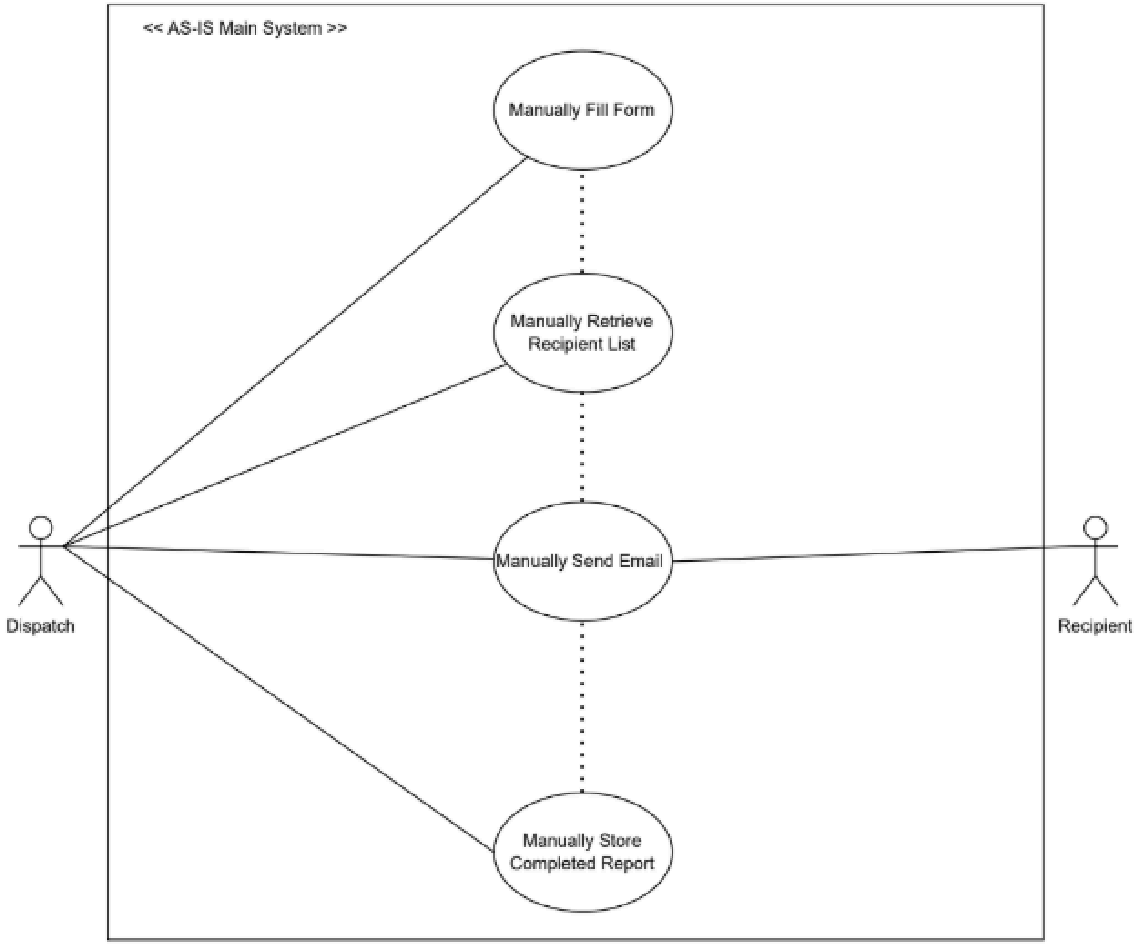
Consequences:

- Increased Workload: Causes for extra steps that take little time but due to the high level of reports it becomes repetitive and highly time consuming over time.
- Higher Error Rates: Causes for the wrong recipient to be included in the report or incomplete tasks, to which this will require re-work to ensure the report is completed.
- Delayed Reports: This will cause slower response/ reporting times to damage incidents which will then cause further delays within the work flows/ process.

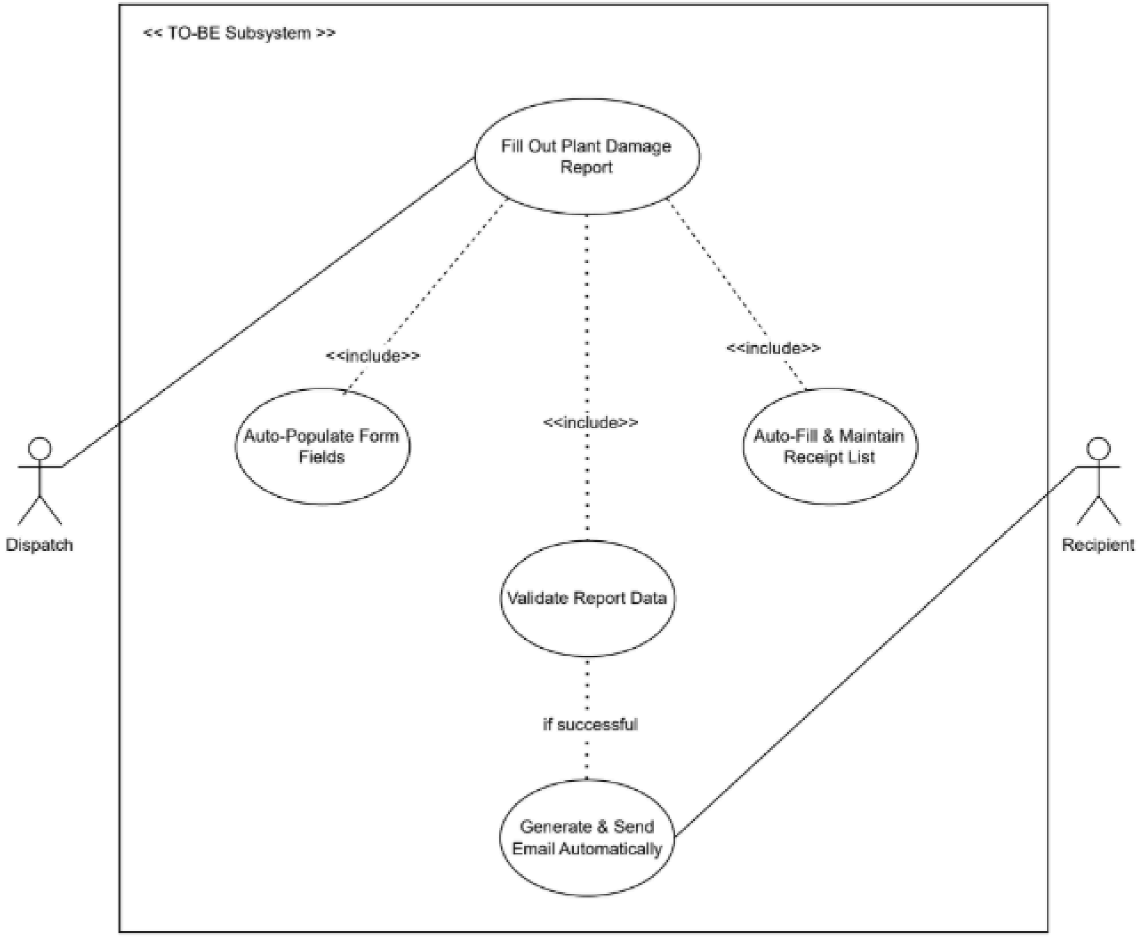
Requirement Specification & Modeling

Use Case Model:

Toronto Hydro AS-IS Main System Current Diagram shows the manual process where dispatchers fill out forms, retrieve recipient lists, send emails, and store reports manually. This process is repetitive, error-prone, and time-consuming, leading to inefficiencies in the workflow.



Toronto Hydro TO-BE Main System Future Diagram represents the streamlined and automated process where the dispatcher's form is autofilled, and the email is sent with the report automatically attached, reducing errors. This will reduce human errors, time delays, and improve overall efficiency in the report generation and dispatch process.



<< TO-BE Subsystem >>
Fill Out Plant Damage Report
<<include>>
<<include>>
<<include>>
Auto-Populate Form Fields
Auto-Fill & Maintain Receipt List
Dispatch
Recipient
Validate Report Data
if successful
Generate & Send Email Automatically

Use Case Descriptions:

The use case descriptions outline the key interactions between the system and its users, focusing on automating the Plant Damage Report Process. The use cases include managing the recipient list, automatically creating and sending emails, filling out the report, auto-filling form fields, and validating data. These use cases aim to improve the overall effectiveness of the plant damage reporting process by lowering manual labour, minimising mistakes, and streamlining communication.

Use Case	**Fill Out Plant Damage Report**
Actor (s)	Dispatcher
Preconditions	● The dispatcher is logged into the system. ● The system has access to the recipient list.
Trigger	Dispatcher initiates a new plant damage report.
Main Flow	1. Dispatcher selects "New Report" from the interface.

	2. The system auto-populates fields: ● Dispatcher Name (from login). ● Date & Time (system timestamp). 3. Dispatcher manually enters the remaining details: ● Incident Location, Contact Name, Locator, Trouble Ticket Number. 4. 4. Dispatcher reviews and submits the report.
Alternate Flow	**3a.** If an incomplete form is detected, the system prevents submission and prompts for missing details. **4a.** If incorrect formatting (e.g., invalid email) is detected, the system prompts the dispatcher to correct it.
Postconditions	● The report is validated and stored in the database. ● The system automatically sends the report to the recipient list (no manual email copy-pasting needed).

Use Case	**Auto-Populate Form Fields**
Actor (s)	System
Preconditions	The dispatcher has initiated a new report.
Trigger	Dispatcher opens the form
Main Flow	1. The system retrieves stored data relevant to the report. 2. The **system pre-fills fields**: - Dispatcher Name (from login session). - Date & Time (system-generated timestamp). - Incident Location (if past data exists). 3. The dispatcher reviews and modifies pre-filled values if needed.
Alternate Flow	2a. If no relevant data is found, fields remain blank for manual entry.
Postconditions	● Some fields are pre-filled, reducing manual effort. ● Dispatchers can proceed with verification & submission.

Use Case	**Validate Form Data**
Actor (s)	System
Preconditions	The dispatcher has filled out the incident report.
Trigger	Dispatcher attempts to submit the form.

Main Flow	1. **System checks all required fields** (e.g., Incident Type, Contact Info, Trouble Ticket Number). 2. **System validates formats**: ● Phone numbers must be numeric. ● Email addresses must contain "@". 3. 3. If all data is correct, the system **allows submission**.
Alternate Flow	3a. If data is missing, the system displays an error message and prevents submission. 4a. If invalid values are entered, the system suggests corrections.
Postconditions	● Form meets required validation criteria. ● System allows the form to be submitted.

Use Case	**Auto-Fill & Maintain Recipient List**
Actor (s)	System, Dispatcher
Preconditions	A plant damage report is being created.
Trigger	Dispatcher selects **"Submit Report".**
Main Flow	**1.** System dynamically retrieves the recipient list based on location and incident type. **2.** The system auto-fills the recipient emails into the report submission form. **3.** Dispatcher reviews and submits.
Alternate Flow	**2a.** If recipients need to be updated, the dispatcher manually modifies the list. **3a.** The updated recipient list is saved for future use.
Postconditions	● **The recipient list is automatically applied (no manual email entry).** ● **If updated, the system remembers the changes for future reports.**

Use Case	**Generate & Send Email Automatically**
Actor (s)	Dispatcher, System
Preconditions	A validated report is ready for submission.
Trigger	Dispatcher submits the report.
Main Flow	1. System generates an email containing the report. 2. System automatically attaches the filled-out form. 3. System sends the email to all recipients.

Alternate Flow	2a. If email sending fails, the system notifies the dispatcher with an error message
Postconditions	● Report is sent without manual email composition. ● Recipients receive the necessary information automatically.

Business Process Modeling

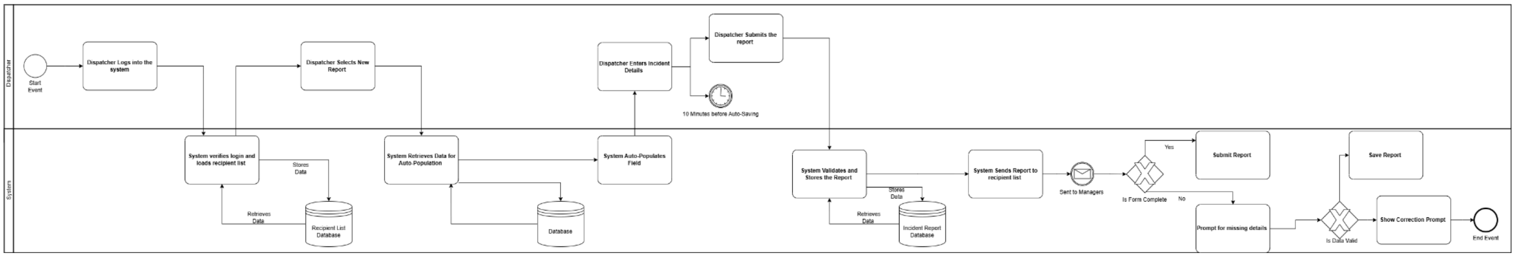
Business Process Management Notation Diagrams:

The following BPMN diagrams are theoretical processes for Toronto Hydro to use to fully automate their document completion system. They are a comprehensive set of automated processes designed to streamline the workflow associated with dispatcher reports and email communications. These diagrams depict automated solutions aimed at enhancing accuracy, efficiency, and consistency across multiple operational stages. For a closer look at the diagrams, click this link: [BPMN Diagrams - Group 1](#).

1. Fill Out Plant Damage Report

Overview:

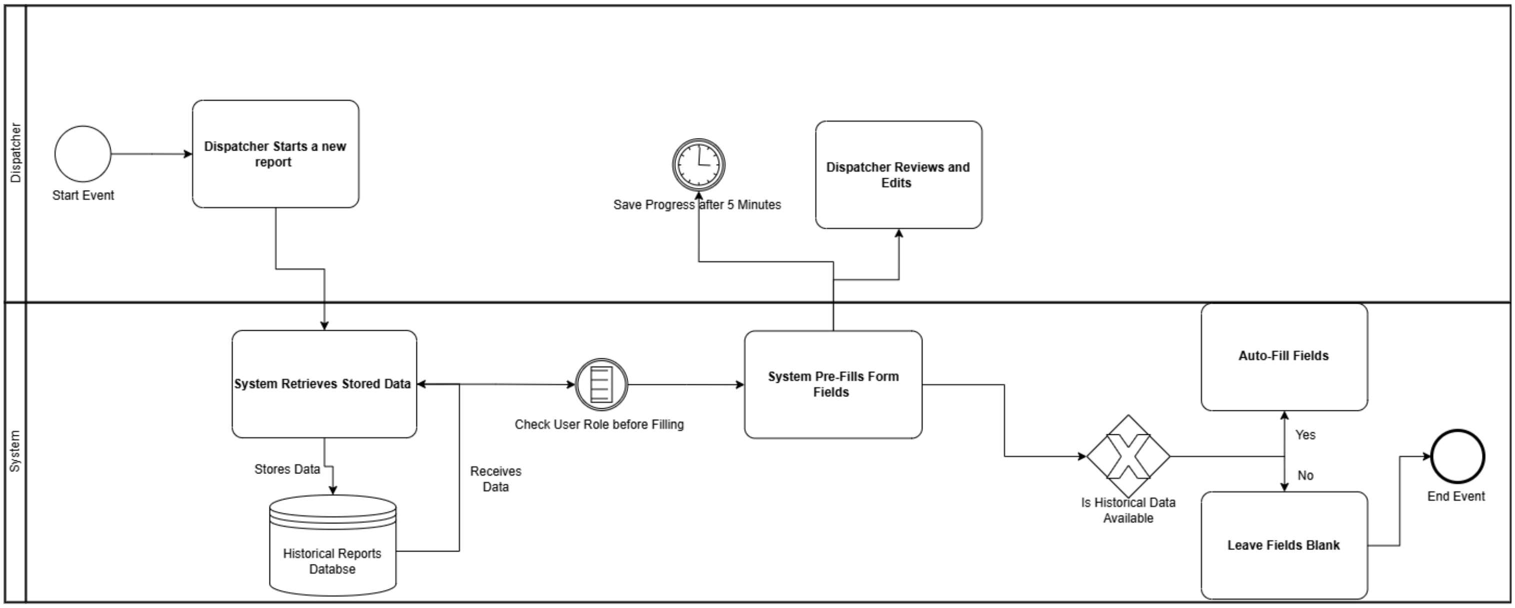
- This BPMN diagram presents the full workflow for creating and submitting a plant damage incident report, highlighting automation and validation stages.


Dispatcher
System
Start Event
Dispatcher Logs into the system
Dispatcher Selects New Report
Dispatcher Enters Incident Details
Dispatcher Submits the report
10 Minutes before Auto-Saving
System verifies login and loads recipient list
Stores Data
Retrieves Data
Recipient List Database
System Retrieves Data for Auto-Population
Database
System Auto-Populates Field
System Validates and Stores the Report
Stores Data
Retrieves Data
Incident Report Database
System Sends Report to recipient list
Sent to Managers
Is Form Complete
Yes
No
Submit Report
Prompt for missing details
Is Data Valid
Save Report
Show Correction Prompt
End Event

2. Auto-Populate Form Fields

Overview:

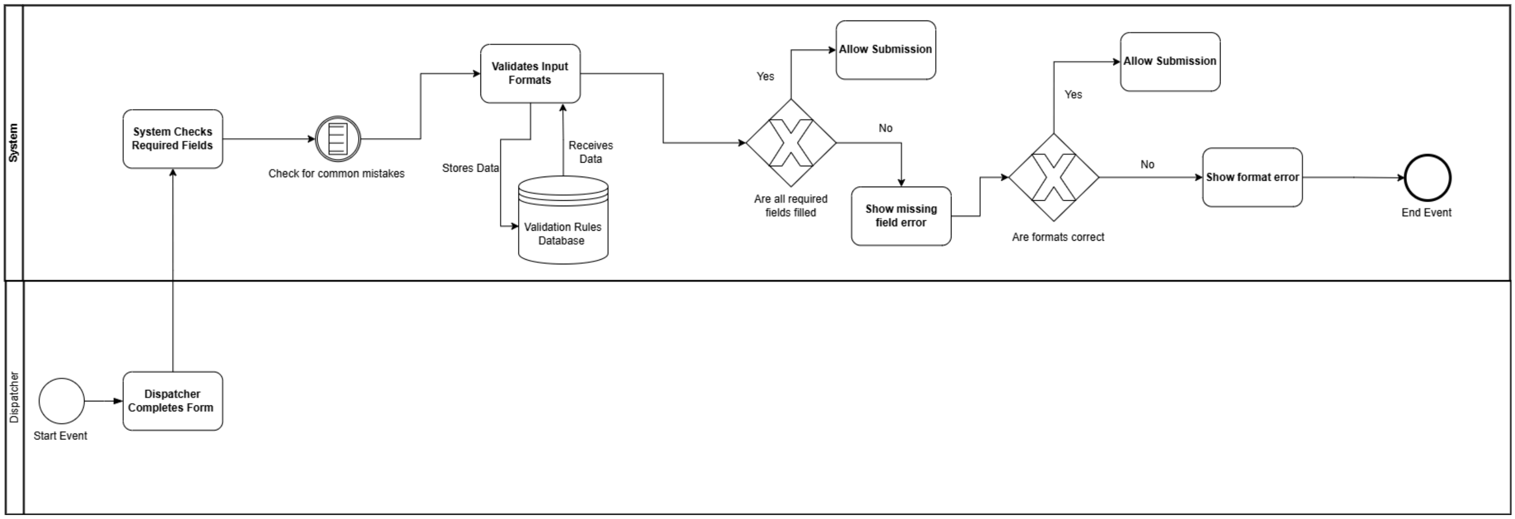
- This diagram demonstrates a system designed to automate data entry by pre-filling form fields using historical data, thus streamlining dispatcher workflows.


Dispatcher
System
Start Event
Dispatcher Starts a new report
Save Progress after 5 Minutes
Dispatcher Reviews and Edits
System Retrieves Stored Data
Check User Role before Filling
System Pre-Fills Form Fields
Auto-Fill Fields
Stores Data
Receives Data
Historical Reports Databse
Is Historical Data Available
Yes
No
Leave Fields Blank
End Event

3. Validate Form Data

Overview:

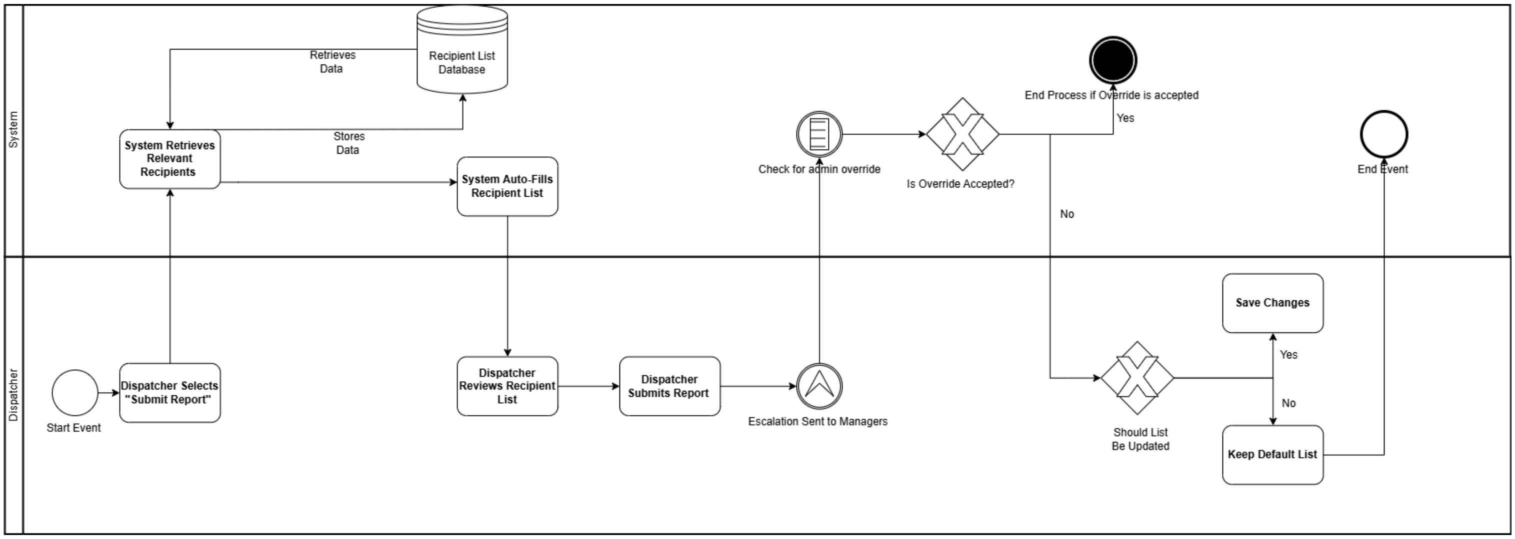
- This process diagram details a system validation mechanism ensuring completeness and correctness of form data submitted by dispatchers.



4. Auto-Fill & Maintain Recipient List

Overview:

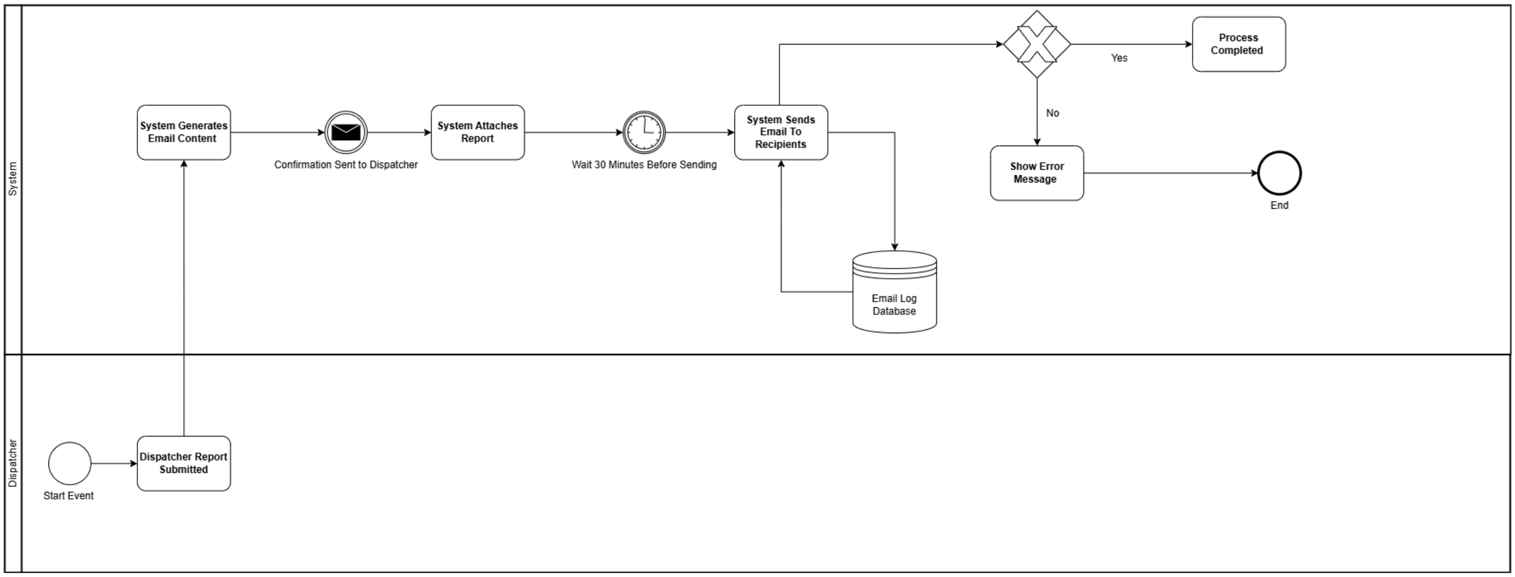
- This BPMN outlines a workflow designed to automate the recipient list management, enhancing accuracy and reducing manual input.


System
Dispatcher
Recipient List Database
Retrieves Data
Stores Data
System Retrieves Relevant Recipients
System Auto-Fills Recipient List
Check for admin override
Is Override Accepted?
End Process if Override is accepted
Yes
No
End Event
Start Event
Dispatcher Selects "Submit Report"
Dispatcher Reviews Recipient List
Dispatcher Submits Report
Escalation Sent to Managers
Should List Be Updated
Save Changes
Yes
No
Keep Default List

1. Generate & Send Email Automatically

Overview:

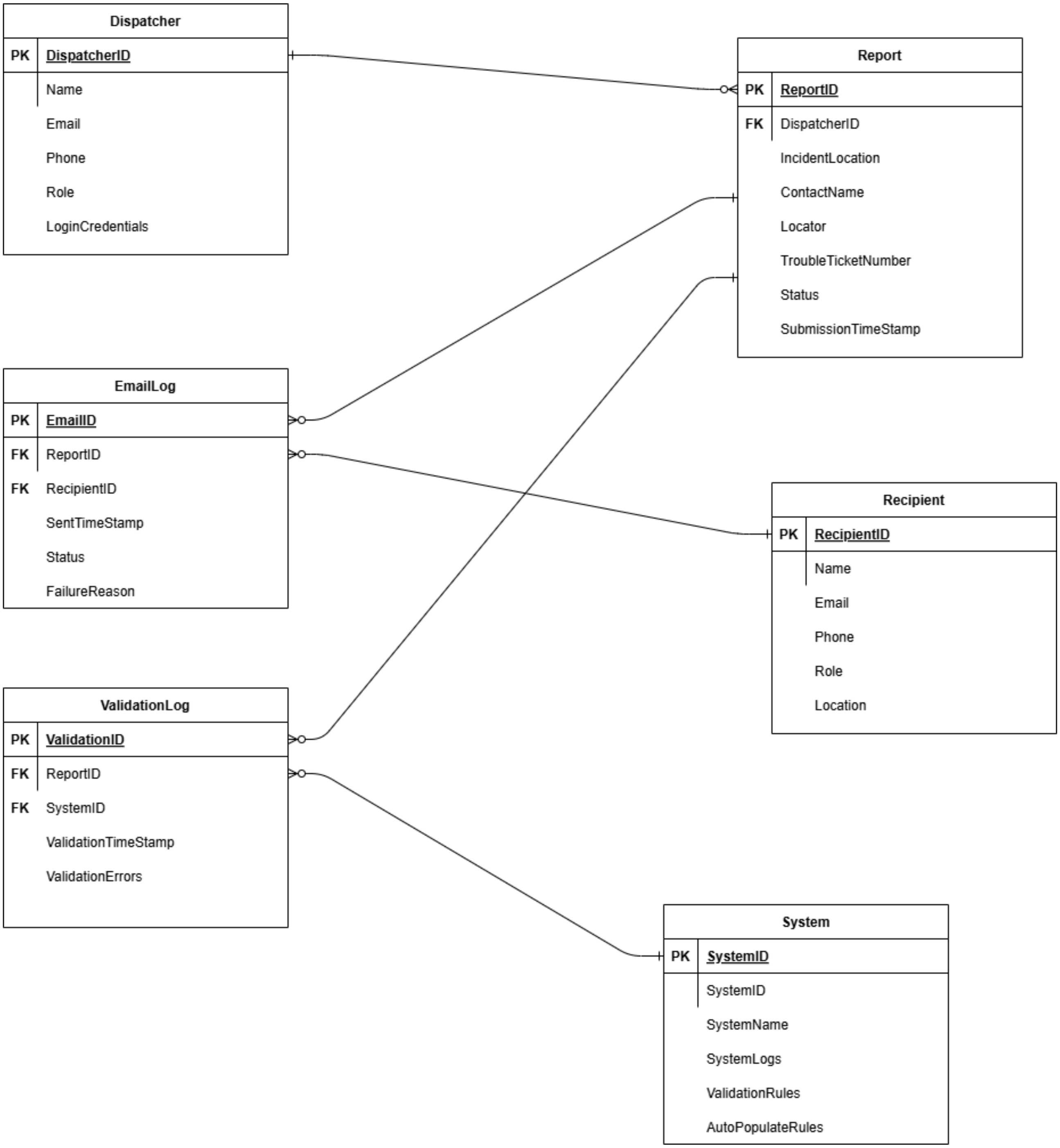
- This diagram illustrates an automated system process to generate and dispatch emails once a dispatcher submits a report.


System
Dispatcher
System Generates Email Content
Confirmation Sent to Dispatcher
System Attaches Report
Wait 30 Minutes Before Sending
System Sends Email To Recipients
Email Log Database
Yes
Process Completed
No
Show Error Message
End
Start Event
Dispatcher Report Submitted

Data Modeling & Design of the new IS solution

Entity Relationship Diagram:

This Entity-Relationship Diagram (ERD) illustrates a blueprint for a structured database design that supports dispatcher reporting activities. The ERD clearly defines entities and relationships for an automated dispatcher reporting system which will help data management and operational efficiency.



Database schema:

The database schema found below outlines the structure for managing dispatcher activities and incident reporting within an automated workflow system. This schema allows for better data management and ensures efficient, accurate reporting and communication processes.

Dispatcher Table stores dispatcher details and login credentials

- **Primary Key: DispatcherID**
- **Relationships: One-to-Many** since a dispatcher can submit multiple reports

Column Name	Data Type	Description
DispatcherID	INT (PK, AUTO_INCREMENT)	Unique ID for each dispatcher
Name	VARCHAR(100)	Dispatcher’s full name
Email	VARCHAR(255)	Dispatcher’s email (used for login & tracking)
Phone	VARCHAR(20)	Dispatcher’s contact number
Role	VARCHAR(50)	Dispatcher’s role (e.g., Field Dispatcher)
LoginCredentials	VARCHAR(255)	Encrypted password for authentication

Report Table stores incident reports

- **Primary Key: ReportID**
- **Foreign Key: DispatcherID** as it links the report to the dispatcher
- **Relationships: One-to-Many** as a dispatcher can create multiple reports

Column Name	Data Type	Description
ReportID	INT (PK, AUTO_INCREMENT)	Unique ID for each report
DispatcherID	INT (FK)	Links report to dispatcher
IncidentLocation	VARCHAR(255)	Location of the incident
ContactName	VARCHAR(100)	Name of the person reporting
Locator	VARCHAR(100)	Technician or supervisor handling it
TroubleTicketNumber	VARCHAR(50)	Reference number for tracking
Status	ENUM('Pending', 'Submitted', 'Reviewed')	Report progress status
SubmissionTimeStamp	TIMESTAMP	Timestamp of report submission

Recipient Table stores recipient details for automated emails

- **Primary Key: RecipientID**
- **Relationships: Many-to-Many** since a report can be sent to multiple recipients

Column Name	**Data Type**	**Description**
RecipientID	INT (PK, AUTO_INCREMENT)	Unique ID for each recipient
Name	VARCHAR(100)	Name of the recipient
Email	VARCHAR(255)	Recipient’s email address
Phone	VARCHAR(20)	Contact number
Role	VARCHAR(50)	Role of the recipient (e.g., Supervisor)
Location	VARCHAR(255)	Assigned location (if applicable)

EmailLog stores logs of sent emails

- **Primary Key: EmailID**
- **Foreign Keys: ReportID and RecipientID**
- **Relationships: One-to-Many** since a report can have multiple email logs

Column Name	**Data Type**	**Description**
EmailID	INT (PK, AUTO_INCREMENT)	Unique ID for each email record
ReportID	INT (FK)	Links email to the report sent
RecipientID	INT (FK)	Links email to the recipient
SentTimeStamp	TIMESTAMP	Time the email was sent
Status	ENUM('Sent', 'Failed', 'Pending')	Delivery status
FailureReason	TEXT	If failed, reason for failure

ValidationLog tracks report validation attempts

- **Primary Key: ValidationID**
- **Foreign Keys: ReportID and SystemID**
- **Relationships: One-to-Many** since a report can undergo multiple validations

Column Name	**Data Type**	**Description**
ValidationID	INT (PK, AUTO_INCREMENT)	Unique ID for each validation
ReportID	INT (FK)	Links validation to a report
SystemID	INT (FK)	Links to the system performing validation
ValidationTimeStamp	TIMESTAMP	Time the validation occurred
ValidationErrors	TEXT	Stores any validation errors detected

System stores system-wide setting and automation rules

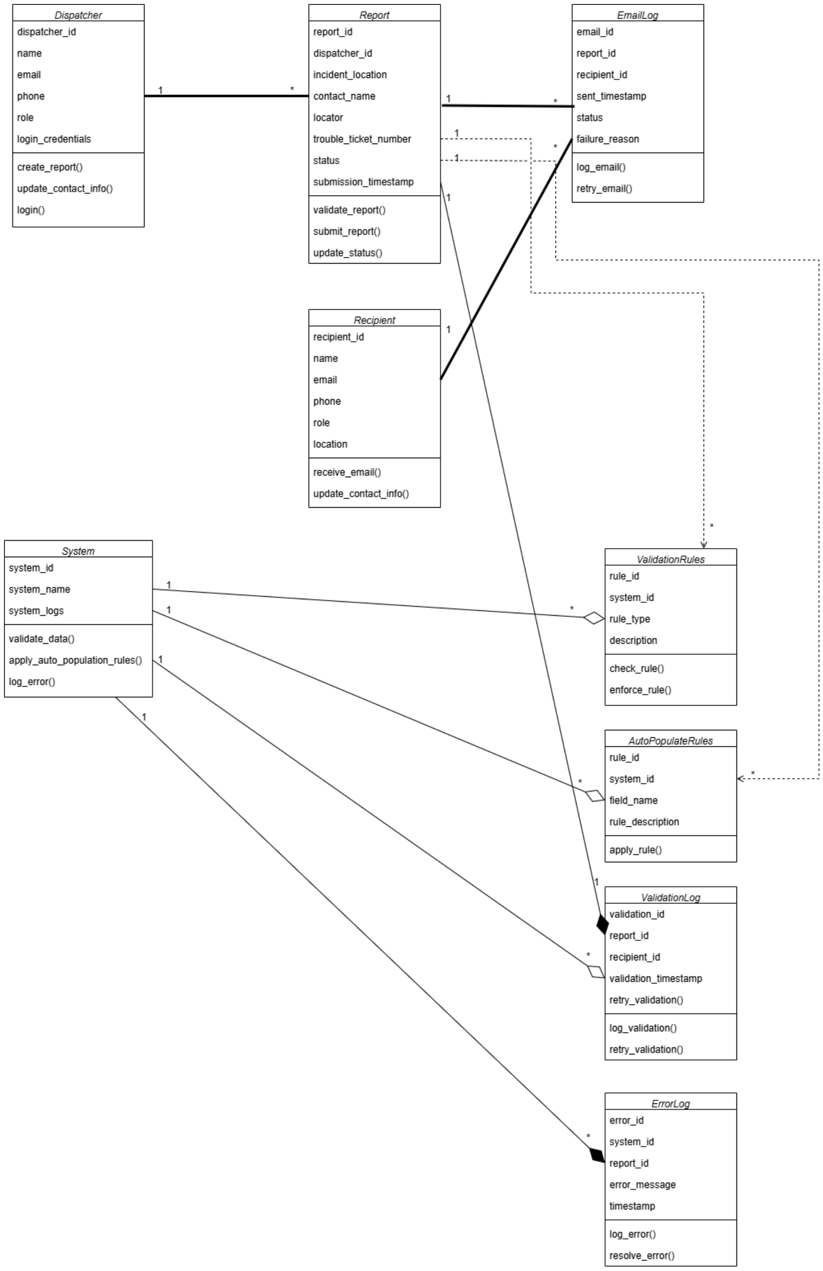
- **Primary Key: SystemID**
- **Relationships: One-to-many** as system’s settings apply to multiple validation logs

Column Name	**Data Type**	**Description**
SystemID	INT (PK, AUTO_INCREMENT)	Unique ID for system configuration
SystemName	VARCHAR(100)	Name of the system module
SystemLogs	TEXT	Logs of system activities
ValidationRules	TEXT	Predefined validation rules for reports
AutoPopulateRules	TEXT	Rules for auto-populating form fields

Structural & Behavioral Design of the new IS solution

<u>Class Diagram:</u>

This class diagram represents the structure and relationships of a system designed for managing dispatcher activities, reporting incidents, and automating associated processes. It shows how components of the system interact. This diagram is primarily used for understanding object-oriented design, which helps with clear communication among developers, and supporting efficient system implementation and maintenance. For a better look at the diagram, click this link: [Class Diagram - Group 1]



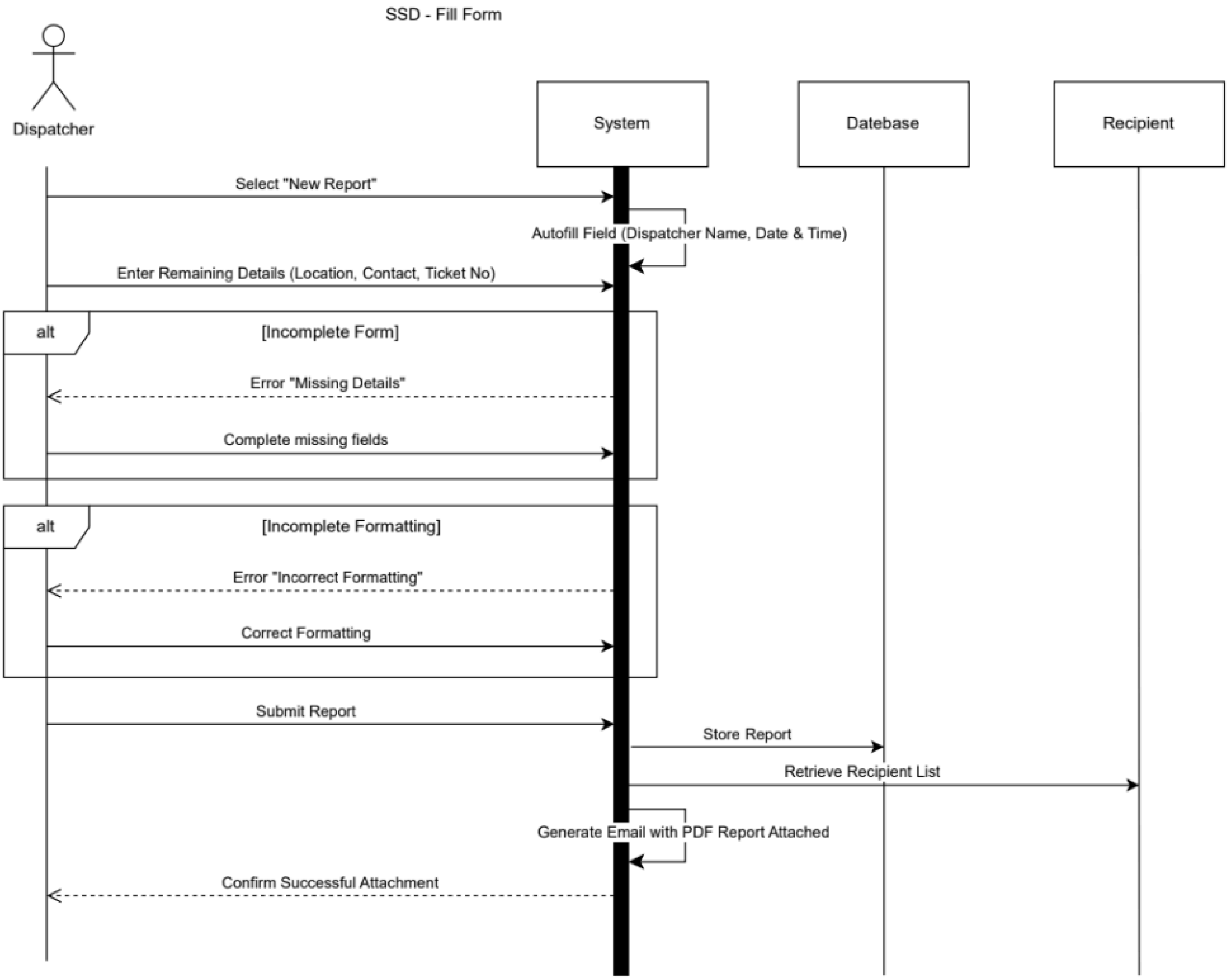
Sequence diagrams:

Paragraph describing the overall sequence diagrams

The sequence diagrams show how the system and actors interact in five key use cases for automating the Plant Damage Report process. These diagrams help demonstrate the flow of information and the steps involved in automating manual tasks like form filling, recipient management, and email dispatching. The sequence diagrams highlight system-user interactions, validations, and error handling while illuminating the system's operations step-by-step. For a closer look at the diagrams, click these links to view over all diagrams: Sequence DIagram 1, Sequence Diagram 2, Sequence Diagram 3, Sequence Diagram 4, and Sequence Diagram 5

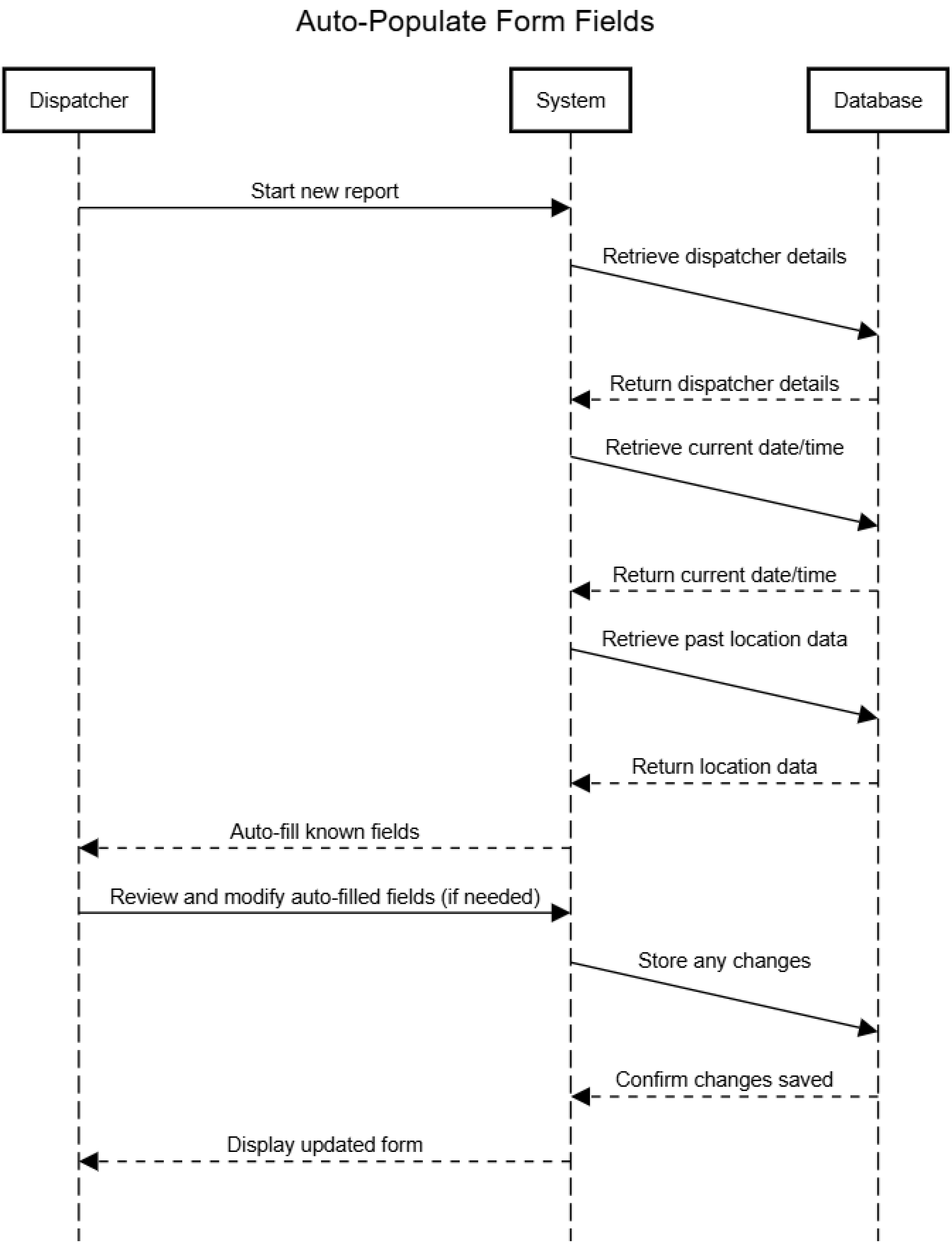
Fill Out Plant Damage Report

The dispatcher initiates a new report, auto-fills data from previous records, and validates the form before submission.



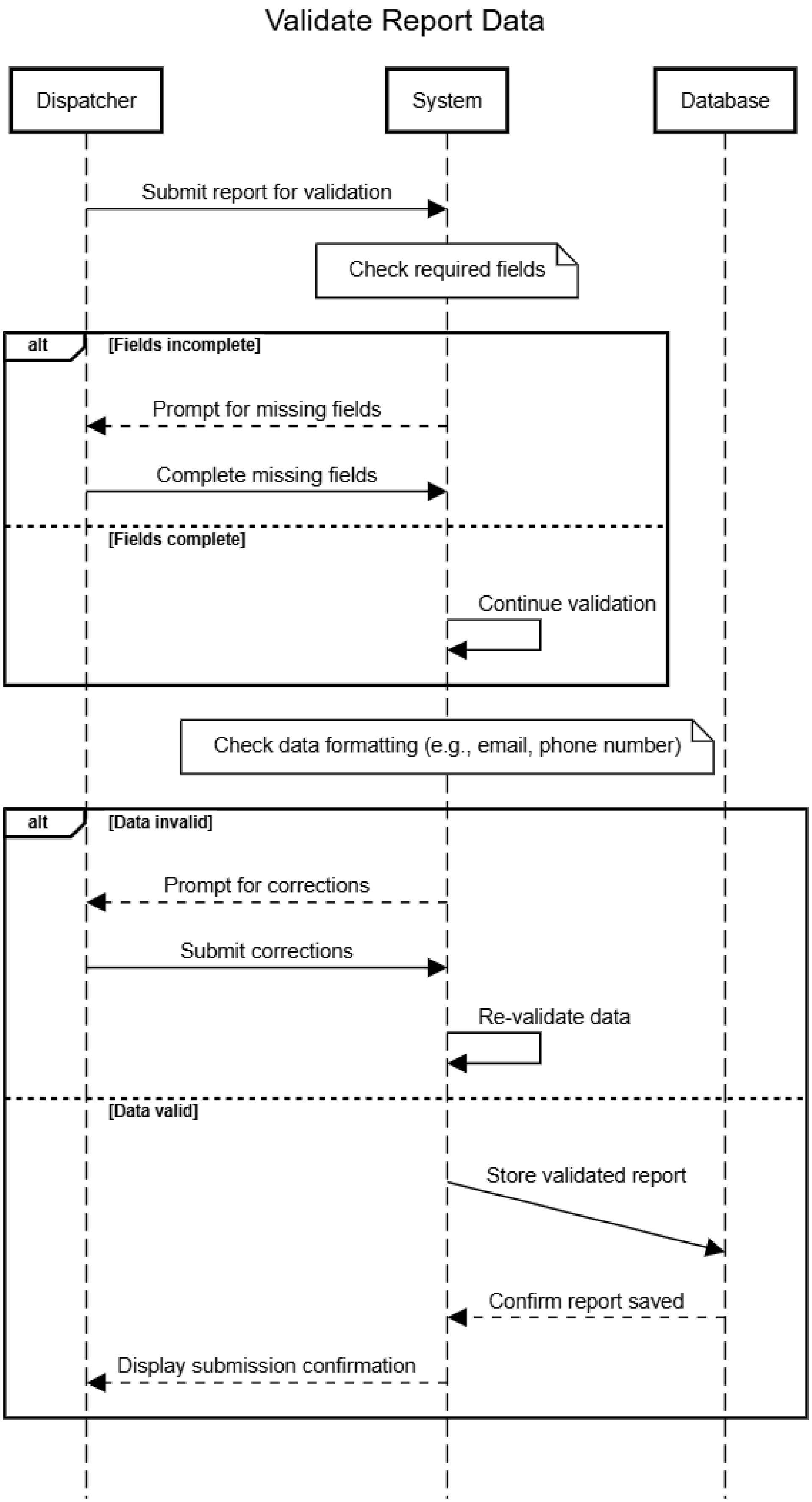
Auto-Populate Form Fields

System automatically retrieves and fills in relevant data fields based on previous reports on existing information.


Auto-Populate Form Fields
Dispatcher
System
Database
Start new report
Retrieve dispatcher details
Return dispatcher details
Retrieve current date/time
Return current date/time
Retrieve past location data
Return location data
Auto-fill known fields
Review and modify auto-filled fields (if needed)
Store any changes
Confirm changes saved
Display updated form

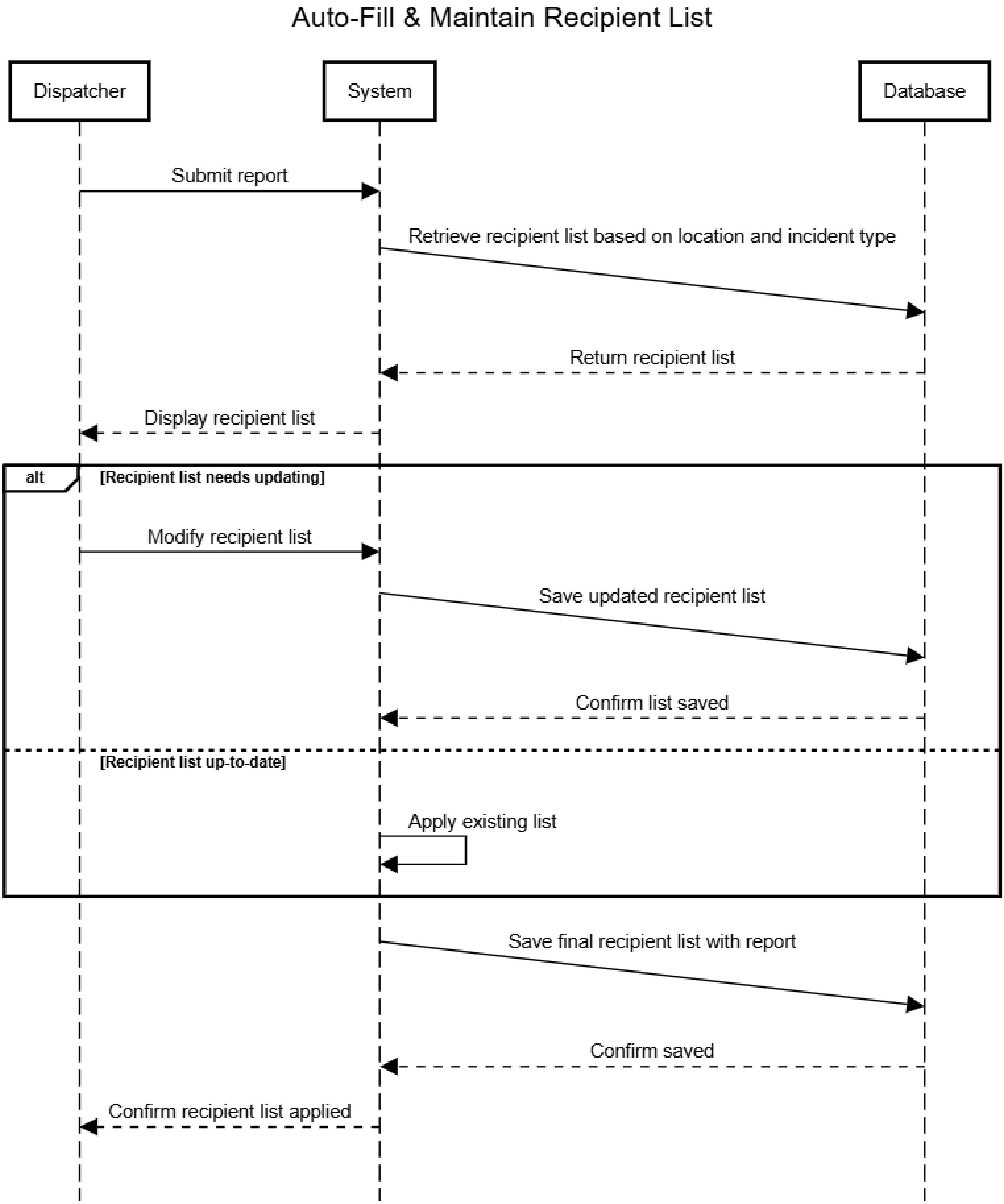
Validate Form Data

System checks the form for required fields and correct formats, ensuring completeness before submission.



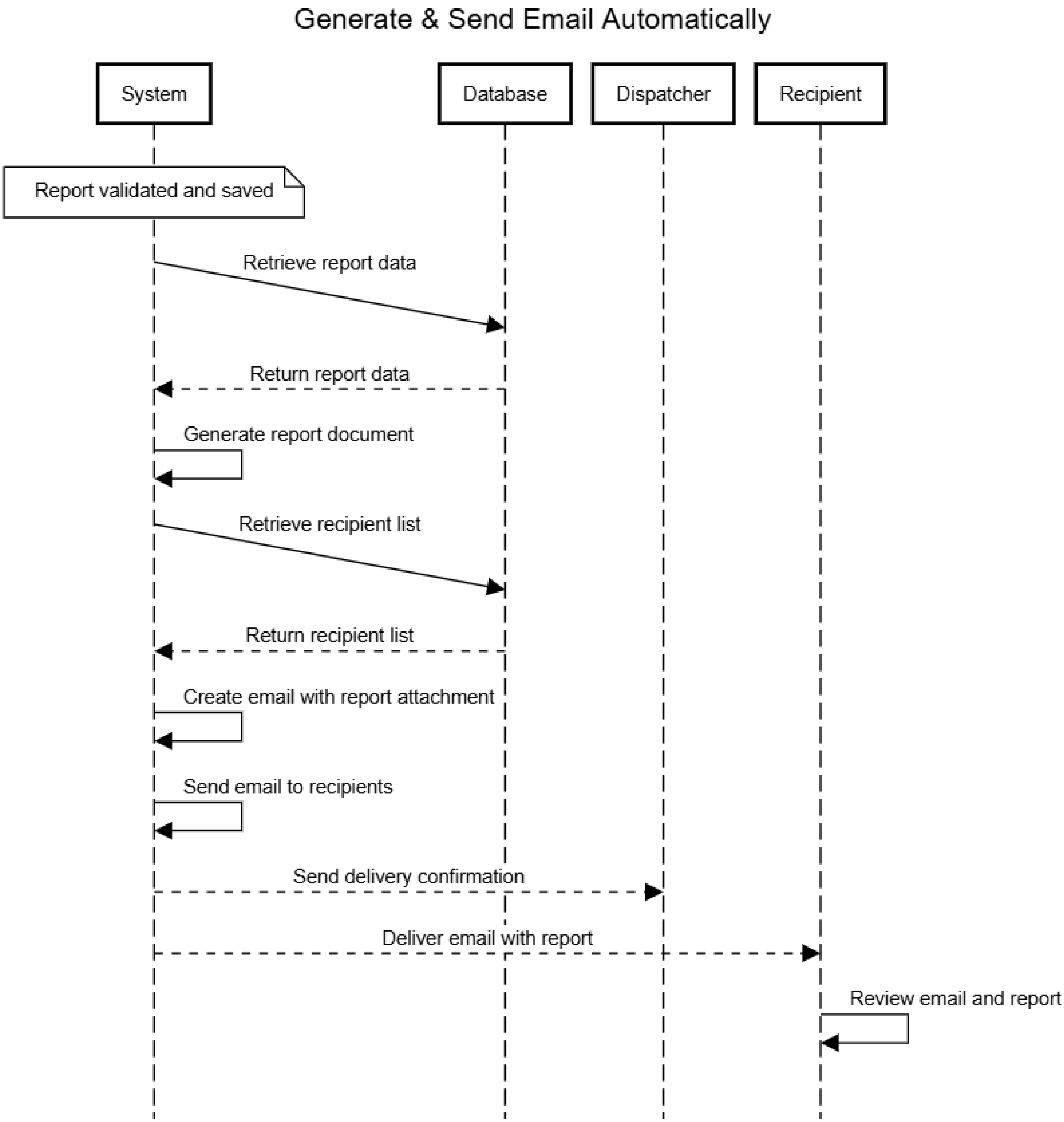
Auto-Fill & Maintain Recipient List

System dynamically retrieves the recipient list based on the incident details and allows dispatchers to update it if necessary


Auto-Fill & Maintain Recipient List
Dispatcher
System
Database
Submit report
Retrieve recipient list based on location and incident type
Return recipient list
Display recipient list
alt
[Recipient list needs updating]
Modify recipient list
Save updated recipient list
Confirm list saved
[Recipient list up-to-date]
Apply existing list
Save final recipient list with report
Confirm saved
Confirm recipient list applied

Generate & Send Email Automatically

Once the report is validated, the system generates an email, attaches the report, and sends it automatically to the relevant recipients.

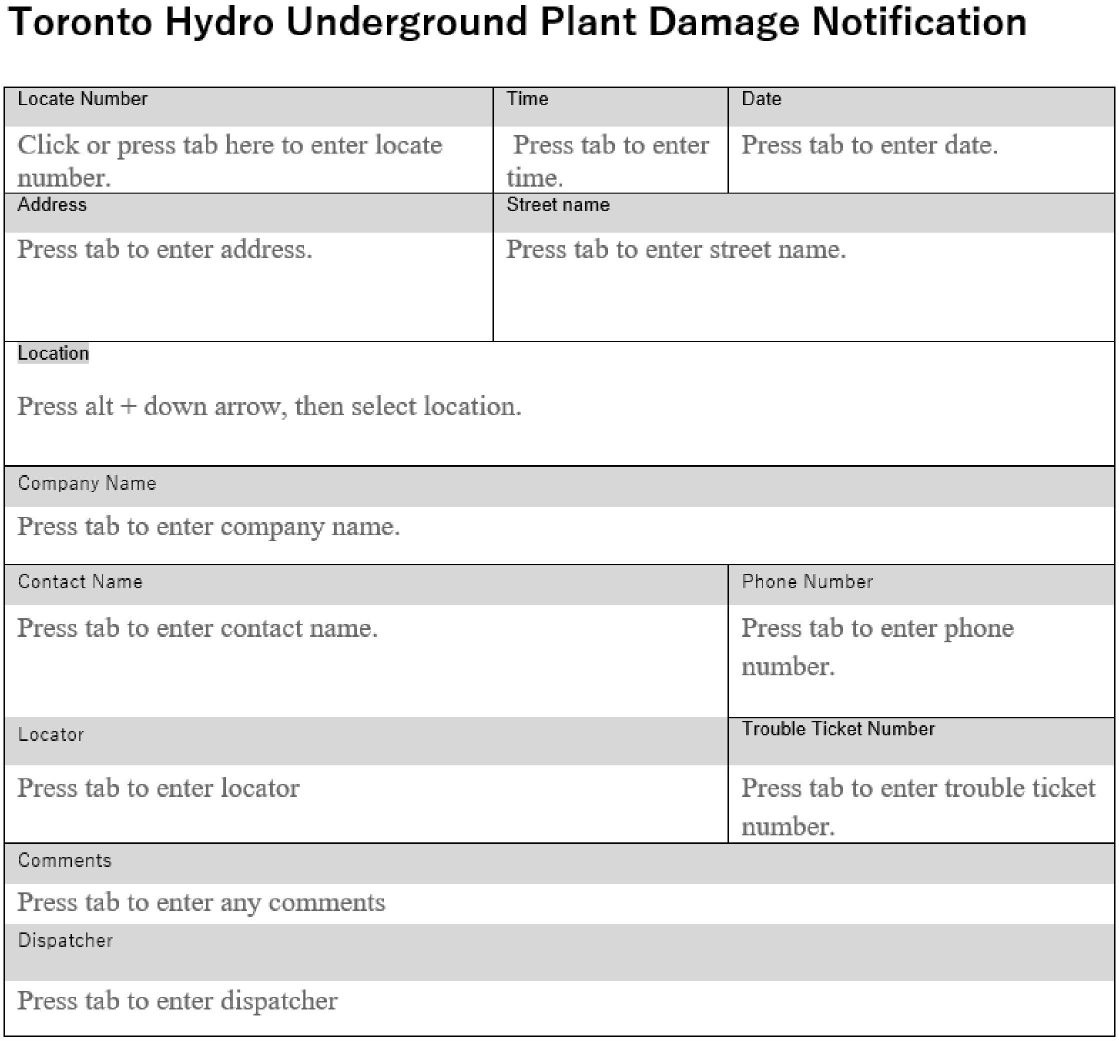


Proof-of-Concept Implementation prototype

Prototype of system:

The Toronto Hydro Underground Plant Damage Notification Form is a document used to report incidents of damage to underground infrastructure. The objective of the automated form was to reduce the clicks of a dispatcher, and in turn save time. This was done through a macro enabled Word document that used VBA code to automatically iterate through fields once they are completed using the tab key. The form includes fillable fields for all the needed information. Once completed, the form can generate an automated email through outlook. The email includes all the relevant information along with a PDF version of the form itself with the information filled out. Along with the information, the form includes a dynamic recipient list which can be changed or edited to reflect any changes or updates to the recipients. This recipient list is automatically added as the recipients to the generated email. Some images of the form cannot be included due to sensitive information being included.

Images of form:

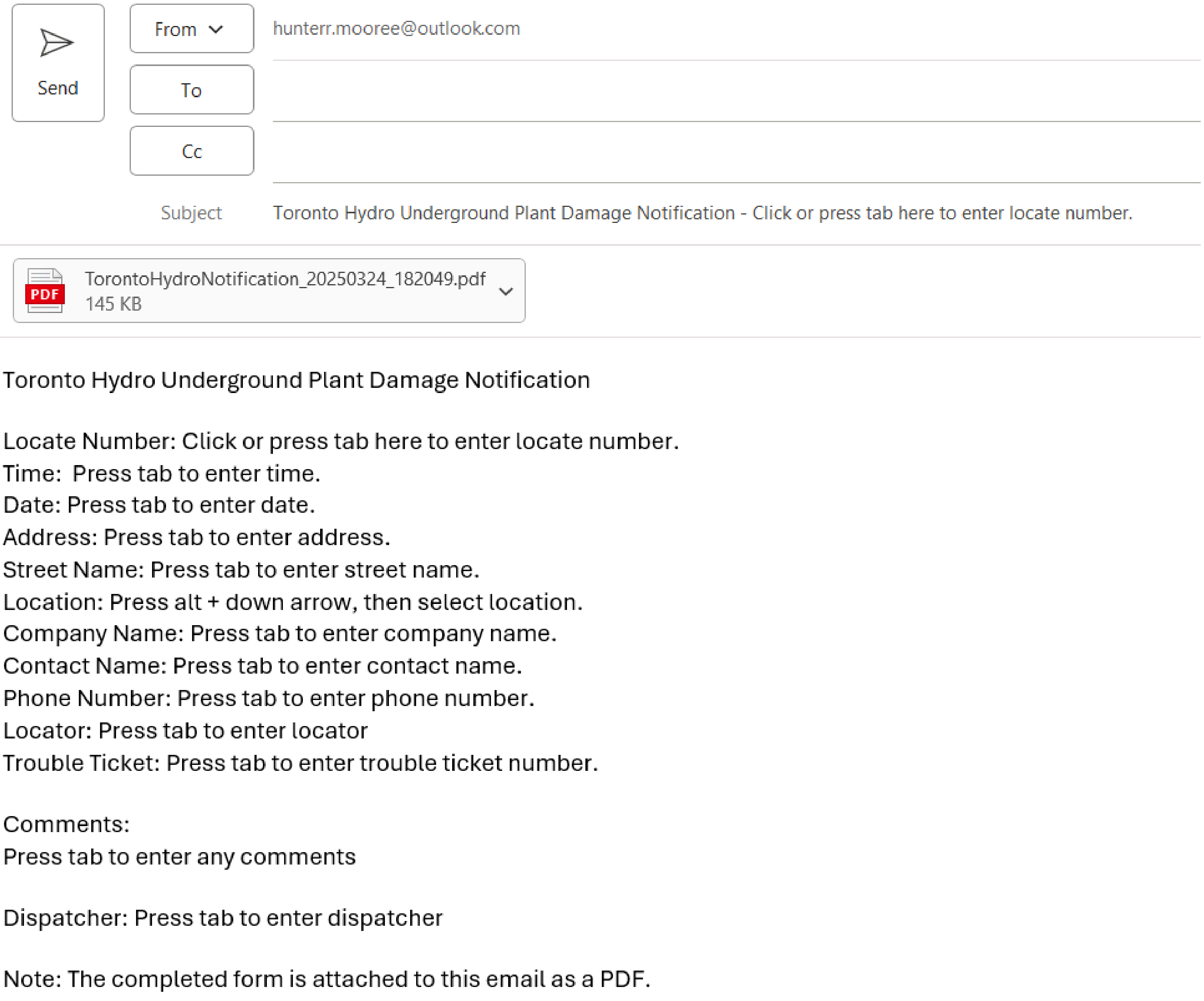


Toronto Hydro Underground Plant Damage Notification

Locate Number	Time	Date
Click or press tab here to enter locate number.	Press tab to enter time.	Press tab to enter date.
Address	Street name	
Press tab to enter address.	Press tab to enter street name.	
Location		
Press alt + down arrow, then select location.		
Company Name		
Press tab to enter company name.		
Contact Name		Phone Number
Press tab to enter contact name.		Press tab to enter phone number.
Locator		Trouble Ticket Number
Press tab to enter locator		Press tab to enter trouble ticket number.
Comments		
Press tab to enter any comments		
Dispatcher		
Press tab to enter dispatcher		

Generate Hydro Underground Plant Damage Notification Email:

Generate Email



From: hunterr.mooree@outlook.com

Subject: Toronto Hydro Underground Plant Damage Notification - Click or press tab here to enter locate number.

TorontoHydroNotification_20250324_182049.pdf
145 KB

Toronto Hydro Underground Plant Damage Notification

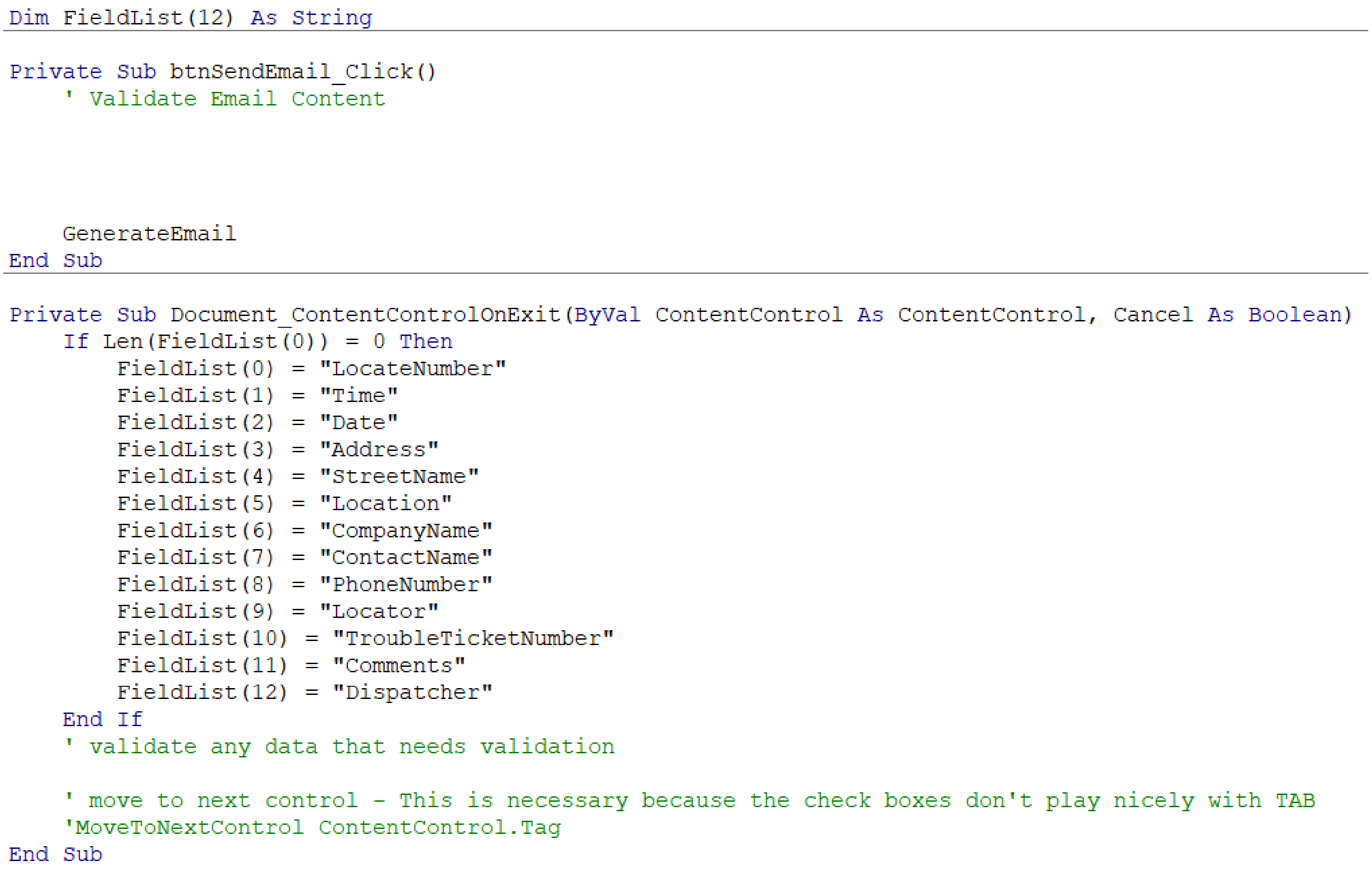
Locate Number: Click or press tab here to enter locate number.
Time: Press tab to enter time.
Date: Press tab to enter date.
Address: Press tab to enter address.
Street Name: Press tab to enter street name.
Location: Press alt + down arrow, then select location.
Company Name: Press tab to enter company name.
Contact Name: Press tab to enter contact name.
Phone Number: Press tab to enter phone number.
Locator: Press tab to enter locator
Trouble Ticket: Press tab to enter trouble ticket number.

Comments:
Press tab to enter any comments

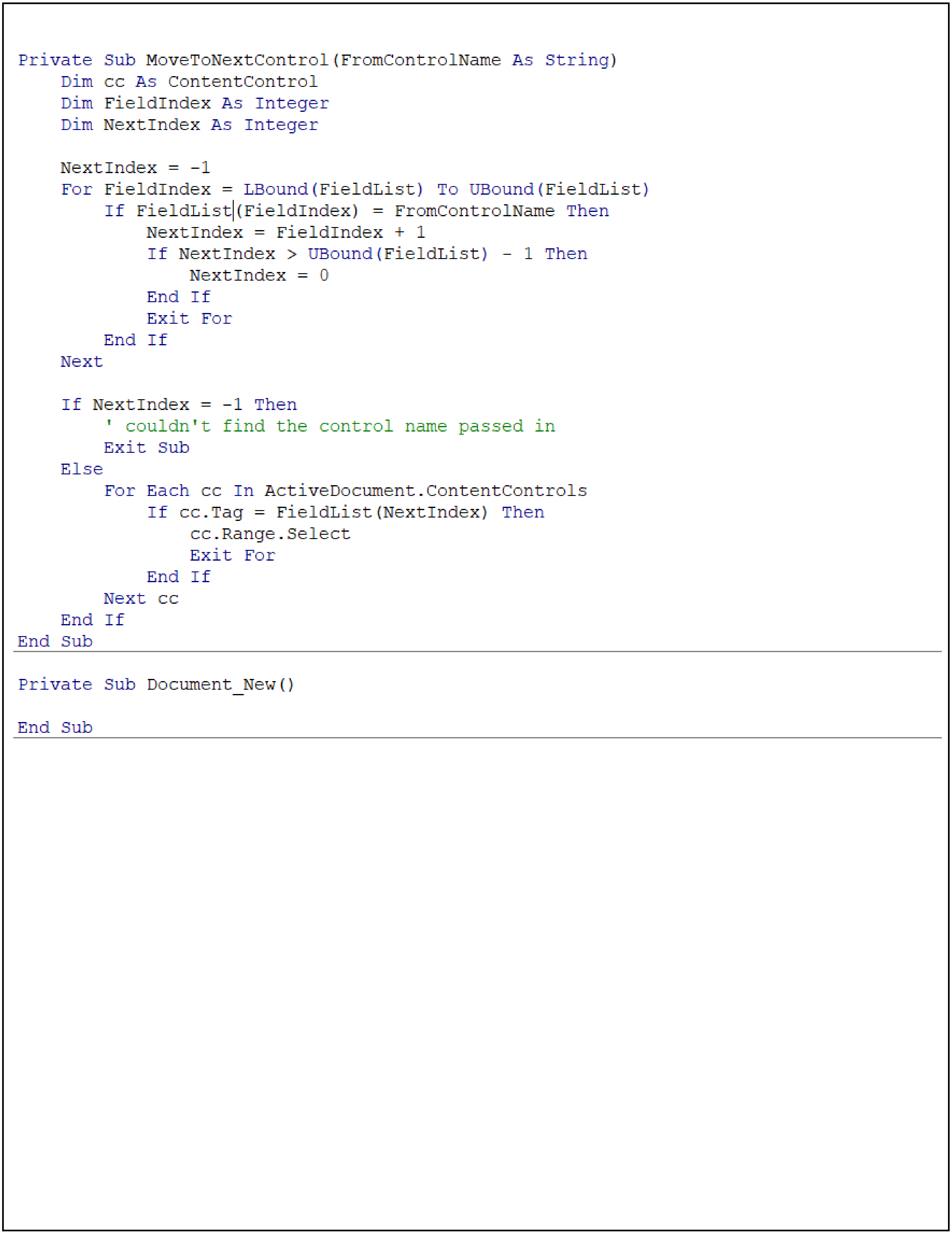
Dispatcher: Press tab to enter dispatcher

Note: The completed form is attached to this email as a PDF.

Images of the VBA code:

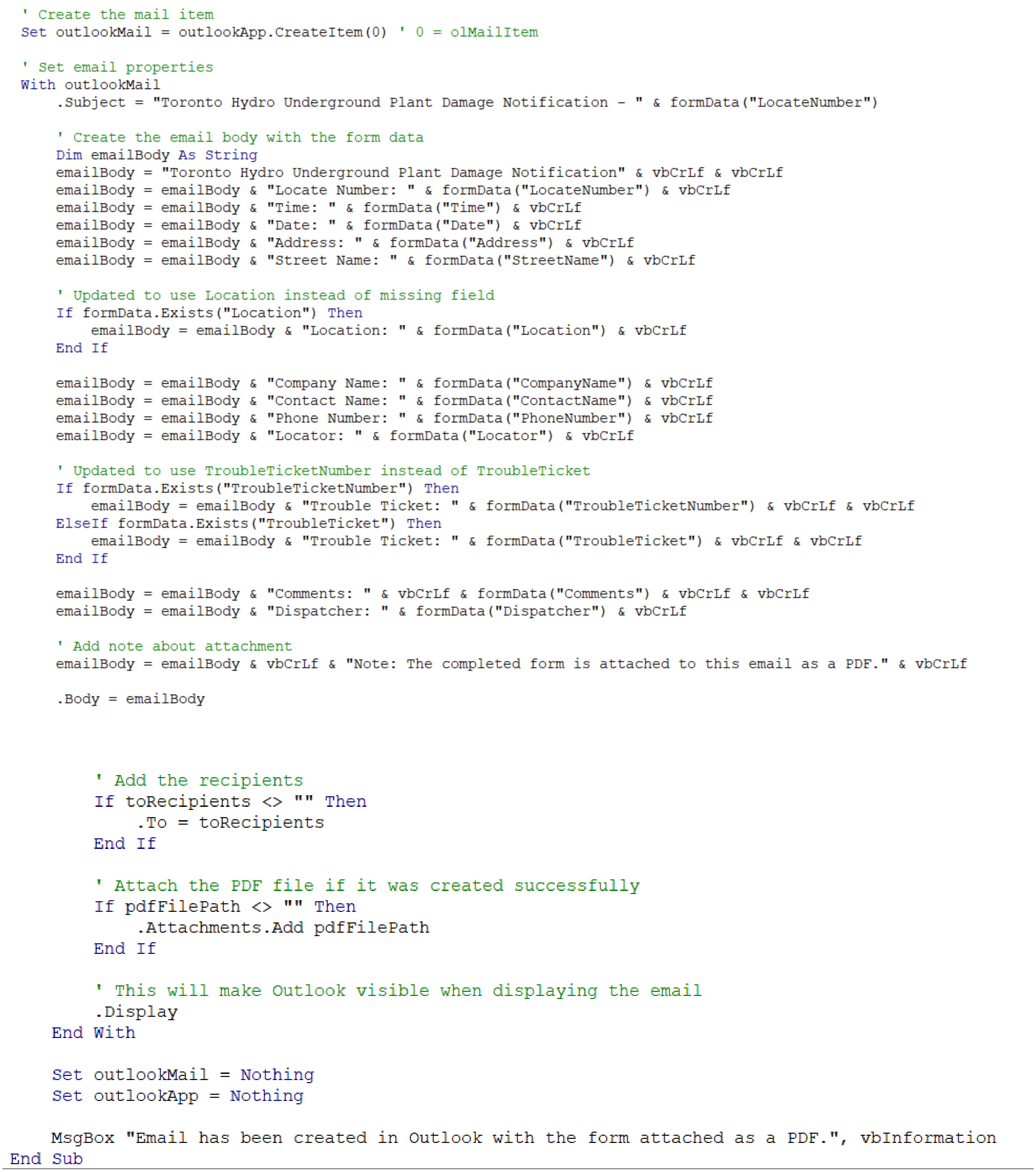


```
Dim FieldList(12) As String

Private Sub btnSendEmail_Click()
    ' Validate Email Content




    GenerateEmail
End Sub

Private Sub Document_ContentControlOnExit(ByVal ContentControl As ContentControl, Cancel As Boolean)
    If Len(FieldList(0)) = 0 Then
        FieldList(0) = "LocateNumber"
        FieldList(1) = "Time"
        FieldList(2) = "Date"
        FieldList(3) = "Address"
        FieldList(4) = "StreetName"
        FieldList(5) = "Location"
        FieldList(6) = "CompanyName"
        FieldList(7) = "ContactName"
        FieldList(8) = "PhoneNumber"
        FieldList(9) = "Locator"
        FieldList(10) = "TroubleTicketNumber"
        FieldList(11) = "Comments"
        FieldList(12) = "Dispatcher"
    End If
    ' validate any data that needs validation

    ' move to next control - This is necessary because the check boxes don't play nicely with TAB
    'MoveToNextControl ContentControl.Tag
End Sub
```

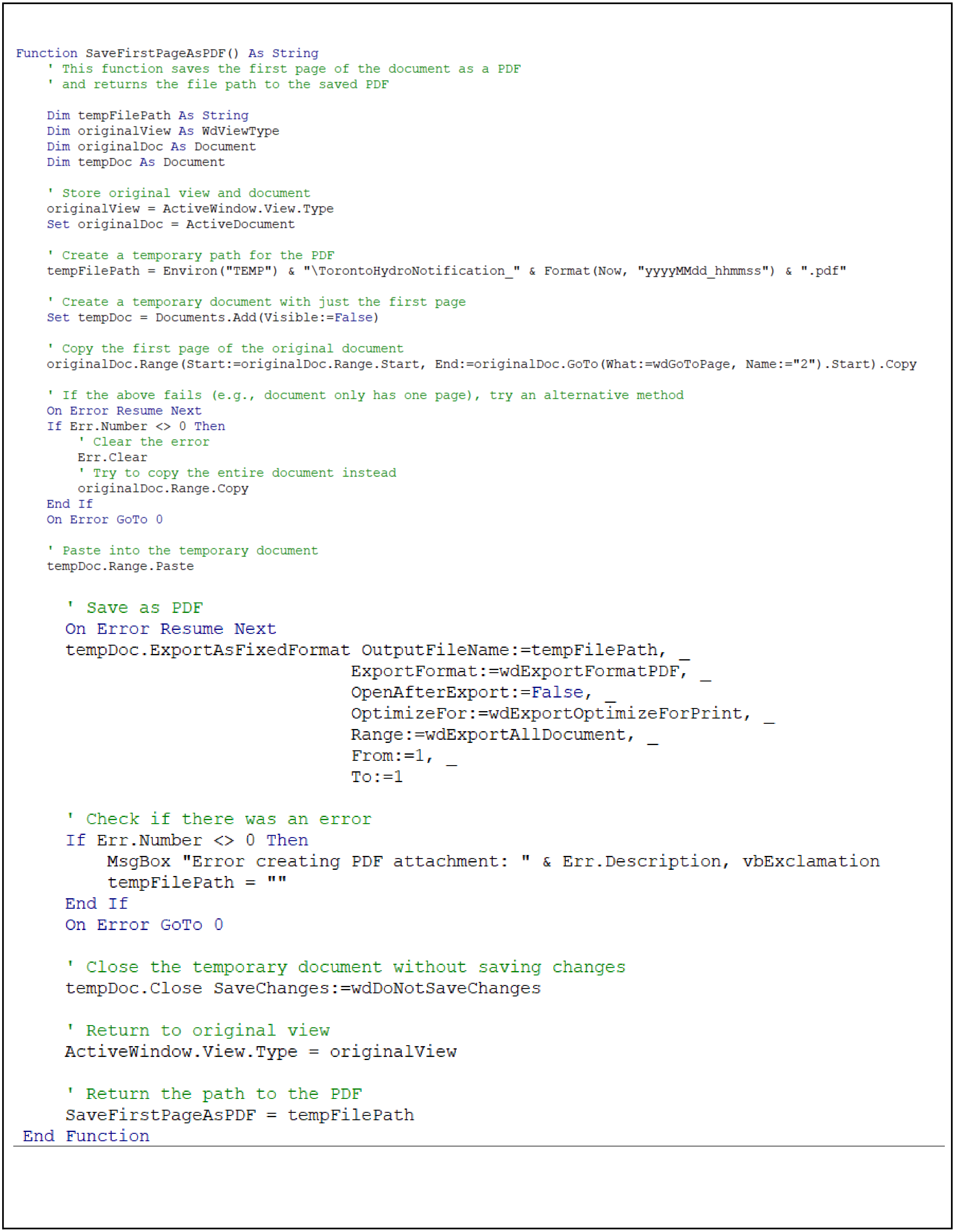


```
Private Sub MoveToNextControl(FromControlName As String)
    Dim cc As ContentControl
    Dim FieldIndex As Integer
    Dim NextIndex As Integer

    NextIndex = -1
    For FieldIndex = LBound(FieldList) To UBound(FieldList)
        If FieldList(FieldIndex) = FromControlName Then
            NextIndex = FieldIndex + 1
            If NextIndex > UBound(FieldList) - 1 Then
                NextIndex = 0
            End If
            Exit For
        End If
    Next

    If NextIndex = -1 Then
        ' couldn't find the control name passed in
        Exit Sub
    Else
        For Each cc In ActiveDocument.ContentControls
            If cc.Tag = FieldList(NextIndex) Then
                cc.Range.Select
                Exit For
            End If
        Next cc
    End If
End Sub

Private Sub Document_New()

End Sub
```

```vba
Sub GenerateEmail()

    ' Create a dictionary to store all content control values
    Dim formData As Object
    Set formData = CreateObject("Scripting.Dictionary")

    ' Read all the values from content controls
    Dim cc As ContentControl
    For Each cc In ActiveDocument.ContentControls
        If cc.Tag <> "" Then
            formData.Add cc.Tag, cc.Range.Text
        End If
    Next cc

    ' Get email recipients from content control
    Dim toRecipients As String
    toRecipients = GetRecipientsFromControl("EmailRecipients")

    ' Save the first page as a PDF
    Dim pdfFilePath As String
    pdfFilePath = SaveFirstPageAsPDF()

    ' Create a new Outlook email
    Dim outlookApp As Object
    Dim outlookMail As Object

    ' First try to get existing Outlook application
    On Error Resume Next
    Set outlookApp = GetObject(, "Outlook.Application")

    ' If that failed, try to create a new instance
    If outlookApp Is Nothing Then
        ' Clear the error state before trying again
        Err.Clear
        Set outlookApp = CreateObject("Outlook.Application")
    End If

    ' Check if we still have an error or no Outlook instance
    If Err.Number <> 0 Or outlookApp Is Nothing Then
        MsgBox "Could not open Outlook. Please make sure Outlook is installed and try again.", vbExclamation
        Exit Sub
    End If
    On Error GoTo 0
```



```vba
    ' Create the mail item
    Set outlookMail = outlookApp.CreateItem(0) ' 0 = olMailItem

    ' Set email properties
    With outlookMail
        .Subject = "Toronto Hydro Underground Plant Damage Notification - " & formData("LocateNumber")

        ' Create the email body with the form data
        Dim emailBody As String
        emailBody = "Toronto Hydro Underground Plant Damage Notification" & vbCrLf & vbCrLf
        emailBody = emailBody & "Locate Number: " & formData("LocateNumber") & vbCrLf
        emailBody = emailBody & "Time: " & formData("Time") & vbCrLf
        emailBody = emailBody & "Date: " & formData("Date") & vbCrLf
        emailBody = emailBody & "Address: " & formData("Address") & vbCrLf
        emailBody = emailBody & "Street Name: " & formData("StreetName") & vbCrLf

        ' Updated to use Location instead of missing field
        If formData.Exists("Location") Then
            emailBody = emailBody & "Location: " & formData("Location") & vbCrLf
        End If

        emailBody = emailBody & "Company Name: " & formData("CompanyName") & vbCrLf
        emailBody = emailBody & "Contact Name: " & formData("ContactName") & vbCrLf
        emailBody = emailBody & "Phone Number: " & formData("PhoneNumber") & vbCrLf
        emailBody = emailBody & "Locator: " & formData("Locator") & vbCrLf

        ' Updated to use TroubleTicketNumber instead of TroubleTicket
        If formData.Exists("TroubleTicketNumber") Then
            emailBody = emailBody & "Trouble Ticket: " & formData("TroubleTicketNumber") & vbCrLf & vbCrLf
        ElseIf formData.Exists("TroubleTicket") Then
            emailBody = emailBody & "Trouble Ticket: " & formData("TroubleTicket") & vbCrLf & vbCrLf
        End If

        emailBody = emailBody & "Comments: " & vbCrLf & formData("Comments") & vbCrLf & vbCrLf
        emailBody = emailBody & "Dispatcher: " & formData("Dispatcher") & vbCrLf

        ' Add note about attachment
        emailBody = emailBody & vbCrLf & "Note: The completed form is attached to this email as a PDF." & vbCrLf

        .Body = emailBody



        ' Add the recipients
        If toRecipients <> "" Then
            .To = toRecipients
        End If

        ' Attach the PDF file if it was created successfully
        If pdfFilePath <> "" Then
            .Attachments.Add pdfFilePath
        End If

        ' This will make Outlook visible when displaying the email
        .Display
    End With

    Set outlookMail = Nothing
    Set outlookApp = Nothing

    MsgBox "Email has been created in Outlook with the form attached as a PDF.", vbInformation
End Sub
```



```
Function SaveFirstPageAsPDF() As String
    ' This function saves the first page of the document as a PDF
    ' and returns the file path to the saved PDF

    Dim tempFilePath As String
    Dim originalView As WdViewType
    Dim originalDoc As Document
    Dim tempDoc As Document

    ' Store original view and document
    originalView = ActiveWindow.View.Type
    Set originalDoc = ActiveDocument

    ' Create a temporary path for the PDF
    tempFilePath = Environ("TEMP") & "\TorontoHydroNotification_" & Format(Now, "yyyyMMdd_hhmmss") & ".pdf"

    ' Create a temporary document with just the first page
    Set tempDoc = Documents.Add(Visible:=False)

    ' Copy the first page of the original document
    originalDoc.Range(Start:=originalDoc.Range.Start, End:=originalDoc.GoTo(What:=wdGoToPage, Name:="2").Start).Copy

    ' If the above fails (e.g., document only has one page), try an alternative method
    On Error Resume Next
    If Err.Number <> 0 Then
        ' Clear the error
        Err.Clear
        ' Try to copy the entire document instead
        originalDoc.Range.Copy
    End If
    On Error GoTo 0

    ' Paste into the temporary document
    tempDoc.Range.Paste

    ' Save as PDF
    On Error Resume Next
    tempDoc.ExportAsFixedFormat OutputFileName:=tempFilePath, _
                               ExportFormat:=wdExportFormatPDF, _
                               OpenAfterExport:=False, _
                               OptimizeFor:=wdExportOptimizeForPrint, _
                               Range:=wdExportAllDocument, _
                               From:=1, _
                               To:=1

    ' Check if there was an error
    If Err.Number <> 0 Then
        MsgBox "Error creating PDF attachment: " & Err.Description, vbExclamation
        tempFilePath = ""
    End If
    On Error GoTo 0

    ' Close the temporary document without saving changes
    tempDoc.Close SaveChanges:=wdDoNotSaveChanges

    ' Return to original view
    ActiveWindow.View.Type = originalView

    ' Return the path to the PDF
    SaveFirstPageAsPDF = tempFilePath
 End Function
```

```
Function GetRecipientsFromControl(controlTag As String) As String
    ' Retrieves email recipients from a specified content control
    Dim cc As ContentControl
    Dim recipients As String
    recipients = ""

    ' Look for content control with specified tag
    For Each cc In ActiveDocument.ContentControls
        If cc.Tag = controlTag Then
            ' Found the control, get its text
            recipients = Trim(cc.Range.Text)
            Exit For
        End If
    Next cc

    ' Clean up the recipient list - replace any linebreaks with semicolons
    recipients = Replace(recipients, vbCr, ";")
    recipients = Replace(recipients, vbLf, ";")
    recipients = Replace(recipients, vbCrLf, ";")

    ' Replace multiple semicolons with single ones
    Do While InStr(recipients, ";;") > 0
        recipients = Replace(recipients, ";;", ";")
    Loop

    ' Remove leading/trailing semicolons
    If Left(recipients, 1) = ";" Then recipients = Mid(recipients, 2)
    If Right(recipients, 1) = ";" Then recipients = Left(recipients, Len(recipients) - 1)

    GetRecipientsFromControl = recipients
End Function
```

Implementation and deployment & risk management plan for the new IS solution

Implementation and deployment activities:

In order efficiently Implement and Deploy this form:

- System Preparation
  - Create a Dedicated Form Folder
    - Create a dedicated folder on Toronto Hydro's network or shared drive specifically for this document
    - Example path: \\shared\TorontoHydro\DamageNotifications\
    - Make sure all necessary staff have access permissions to this folder
  - Configure Trust Settings in Microsoft Word
    - Open Microsoft Word
    - Click on "File" > "Options"
    - Select "Trust Center" from the left menu
    - Click "Trust Center Settings"
    - Select "Trusted Locations" from the left menu
    - Click "Add new location"
    - Browse to the dedicated folder you created
    - Check "Subfolders of this location are also trusted" if needed
    - Click "OK" to confirm
  - Place the Document in the Trusted Location
    - Copy the Toronto Hydro Underground Plant Damage Notification document to the trusted folder
    - Consider creating a shortcut to this location on users' desktops for easy access

- User Configuration
  - Email Setup
    - Make sure Microsoft Outlook is installed on all computers that will use this form
    - Verify users have appropriate email permissions to send notifications
    - Test email functionality with your IT department to ensure everything works
  - Content Control Configuration
    - "Email Recipients" content control box if found under the "Generate Email" button
    - To add, delete, or change any of the recipients, be sure to stay with the given format. That format being:
      - Exampleemail@gmail.com
Exampleemail@gmail.com
Exampleemail@gmail.com
    - Consider having a designated administrator maintain this list

- User Training
  - Provide User Training

        - Train all needed staff on how to access and use the document
        - Make sure all staff know the importance of opening the document from the trusted location
        - Demonstrate the field navigation using the Tab key
        - Show how to generate and send the notification email

- Testing and Verification
    - Conduct Testing
        - Test the form with sample data and emails
        - Verify email generation works correctly
        - Confirm PDF attachments are generated and attached properly
        - Test with different user accounts to ensure consistent functionality
    - Verify Security Settings
        - Confirm that macros run without security warnings when opened from the trusted location
        - Test opening from non-trusted locations to verify security controls are working

<u>Risk assessment and mitigation activities:</u>

Implementing the automated document and email generation system will result in significant operational improvements, however, there are some potential challenges that must be anticipated and addressed. The following points outline implementation issues that may arise along with recommended solutions to mitigate their impact effectively

- User Resistance - Dispatchers might struggle with the ability to adapt to the changes and would prefer sticking to the original way of doing things, this can be overcome through additional training and hands on support.
- Integration Challenges - Compatibility problems can occur with the current IT Policies, to which this can be overcome by doing pre deployment checks.
- Data Errors - Incorrect auto filled fields can be prone to happen even when automated this can be overcome by doing countless validation checks for both the auto fill system and automated email generation.
- Deployment Failures - Unexpected bugs can occur in such unexpected ways, to which this can be overcome by doing phased-deployment/ rollout while monitoring for any defects closely.