Software Requirements Specification for Software Engineering: Document Management System

Team 15, SyncMaster
Kyle D'Souza
Mitchell Hynes
Richard Fan
Akshit Gulia
Rafeed Iqbal

November 4, 2024

Contents

Revision History

Date	Author	Notes
10/11/2024	Entire Team	Rev0 of the SRS
11/04/2024	Rafeed Iqbal	Removed OE-IAS1/IAS2
11/04/2024	Rafeed Iqbal	Added SR-S1

1 Purpose of the Project

1.1 User Business

The City of Hamilton's Water Division is responsible for the treatment and distribution of drinking water and the collection and treatment of wastewater throughout the City of Hamilton. Many pumping stations are located throughout the City to perform this work. The maintenance and upkeep of these facilities is crucial to ensure unimpeded service. The facilties team at Hamilton Water requires an application to assist in the management and security of these stations, as there is not a robust system currently in place. For detailed information on the problems the application should address, refer to the problem statement here.

1.2 Goals of the Project

The development of this application will greatly improve the efficiency and transparency of facility management at Hamilton Water and improve station security. For detailed information on the goals of this project, refer to Section 2: Goals here.

2 Stakeholders

2.1 Client

- Matt Yakymyshyn (P.Eng)
  - Role: Senior Project Manager of Technical Services
  - Interest: Team leader for the facilities team within Technical Services who are the primary group responsible for maintaining facility infrastructure at pumping stations. Primary contact at the City for the capstone team.

2.2 Customer

- Technical Services team
  - Role: Primary user within Hamilton Water

- Interest: As the primary stakeholder, the requirements of this project are aimed at their specific needs. Their main concern is that station documentation is managed as a single source of truth, and is easily distributed/retrieved with relevant parties.

2.3 Other Stakeholders

- Supervisory Control and Data Acquisition (SCADA) team
  - Role: Responsible for the control system controlling the treatment process.
  - Interest: The SCADA team also manages contractors and is interested in accessing and updating documentation maintained in this system.
  - They will have valuable input regarding existing City platforms and technologies, as well as how technology can be integrated into pumping stations.
- Corporate Security
  - Role: Pumping Station Security
  - Interest: Provide feedback on any system involving station access.

2.4 Hands-On Users of the Project

- Facilities project managers
  - The facilities project managers will be using the application daily to perform a variety of tasks. This includes verifying work was performed by contractors, accessing station documentation to share with contractors and internal staff, receiving signed forms and storing in a manner which is associated to the signee, and other tasks. They will have to be able to achieve all their needed requirements through the applications user interface, and will be a critical stakeholder to receive feedback from through user testing.
- Facilities co-op students

- Facilities co-op students will be using the application to physically authenticate on site when performing inspections, scoping work, and verifying completion of work.

- Facilities contractors

  - Many contractors access pumping stations daily. They will interact with the application each time they perform facilities related work at the station. Interactions would include uploading and receiving documentation, and authenticating their presence on site to verify completion of work. Contractors come from a wide range of backgrounds depending on their trade, but are typically skilled workers in a particular field and are of working age (18 - 65 years old).

- Facilities sub-contractors

  - Many contractors employ sub-contractors as part of their work. While these subcontractors work for the contractor who hired them, they will still require access to the application to authenticate on site and access site information. Their roles and interactions will be similar to contractors, except they will be much less familiar with the application and may only use it for a short duration as they are not regular workers.

2.5 Personas

- Greg: A facilities manager

  - Greg has a few tasks to complete this morning at work. He sees that another department has requested compliance documentation on crane inspections for the ministry, so he signs onto SyncMaster to retrieve the current version of these documents for each of these sites. While on the application, he receives a notification that a contractor he manages authenticated at site to complete a work order, but the application identified they had not completed their health and safety training. Greg prompts the system to issue this training to them, and runs a report on this contractor to see if any other employees of theirs require their training soon.

- Nancy: A newly onboarded contractor
    - Nancy is a plumber who is servicing her first work order after being hired by "Worldly Plumbing". Her company has notified her of the application she must use to authenticate for the work. When she authenticates, she is notified that the device she is servicing is located within a confined space and that she must sign and return a hazard assessment form and proof of confined space training.

2.6 Priorities Assigned to Users

- Key users
    - Facilities managers. They are the primary stakeholder of the application and it is intended to be a tool to greatly improve the efficiency of their work.
- Secondary users
    - Contractors. They will frequently use the system, but their needs are secondary to the key users.

2.7 User Participation

- Facilities managers will be frequently consulted throughout the duration of this project to demonstrate prototypes, receive feedback, and conduct user testing.
- Co-op students can also be consulted for user testing. They tend to have more availability for longer durations and will be able to provide insight into how a less experienced user interacts with the system.

2.8 Maintenance Users and Service Technicians

The facilities managers will be responsible for ensuring that day-to-day data is added to the application appropriately. The team does not have a dedicated software position, so maintainability of the technical aspects of the application would need to be brought to the IT division of the City when the project reaches that stage.

3 Mandated Constraints

3.1 Solution Constraints

C-SOL1. System must be cloud-based to fit in with current existing systems at the City of Hamilton.

3.2 Implementation Environment of the Current System

To understand the current practices at the City of Hamilton see the *Problem* section of the Problem Statement and Goals here.

3.3 Partner or Collaborative Applications

N/A

3.4 Off-the-Shelf Software

C-OTS1. The system must integrate with Sharepoint to synchronize documents in Sharepoint with documents in the system.

C-OTS2. The system must integrate with Infor EAM, to show the status of work orders associated with a document.

C-OTS3. The system must integrate with MySDS to show relevent relevant SDS documents to users on a given site.

3.5 Anticipated Workplace Environment

C-AWE1. The workplace is an industrial environment. The application must be usable in a manner which ensures worker health and safety per the City's health and safety procedures.

C-AWE2. The workplace is a hybrid environment of in-office and on-site work. The application should be accessible from both environments, as the requirements permit.

C-AWE3. Many stations are accessed from outdoors and the user is exposed to the current weather conditions. The application must be usable in any weather condition.

3.6 Schedule Constraints

C-SCH1. A requirement is to integrate with the Infor EAM system the city intends on using however, this system will not be available until February 2025, so no testing can be done on this system until then.

C-SCH2. The project deadline is April 2, 2025.

3.7 Budget Constraints

C-BDG1. Total expenses up until April 2, 2025 must not exceed $750.

3.8 Enterprise Constraints

N/A

4 Naming Conventions and Terminology

4.1 Glossary of All Terms, Including Acronyms, Used by Stakeholders involved in the Project

- BAS: Building Automation System.
- BCOS: Beyond Compliance Operating System.
- CMMS: Computerized Maintenance Management System.
- Confined Space: A partially or fully enclosed space, not designed for continuous human occupancy, which has the potential for atmospheric hazards.
- Controlled Space: A City defined space which has the potential to become a confined space.
- CSE: Confined Space Entry

- Dry Well: A room which houses industrial equipment such as pumps and valves.
- Hazard Assessment Form: A form outlining potential hazards at a location.
- Hot Works: Work which produces ignition sources. Requires an accompanying hot works permit.
- HVAC: Heating, Ventilation, and Air Conditiong.
- Infor EAM: An Enterprise Asset Management system.
- PMATS: Plant Maintenance and Technical Services.
- PO: Purchase Order.
- PPE: Personal Protective Equipment.
- SCADA: Supervisory Control and Data Acquisition.
- SDS: Safety Data Sheets.
- Wet Well: A portion of a wastewater pumping station which receives and temporarily stores wastewater.

5 Relevant Facts And Assumptions

5.1 Relevant Facts

- There are over 100 pumping stations throughout the City.
- There are dozens of service contracts, and many contractors and staff accessing stations each day.
- Some stations are in remote locations and may have poor cell signal.
- Some properties have multiple stations at the same site.
- Certain stations have special entry procedures.

5.2 Business Rules

- External contractors do not have access to the internal network.
- Only authorized users can approve new procedures.
- There is work at stations performed through a work order and routine work which doesn't have a work order.
- Employee collective agreements have restrictions on the release of GPS logs and video recording, which would require their approval.

5.3 Assumptions

- Our project will focus on the application itself and our team will implement an authentication system adequate for our development. When the City integrates our application into their systems they would replace our authentication system with one that meets their detailed cybersecurity specifications.
- The existing entry and exit procedure for stations is not modified and our application will coexist with it.

6 The Scope of the Work

6.1 The Current Situation

The process currently employed by the stakeholder is predominantly a manual one. Currently, contractor management is a time-consuming and manual effort. The following list identifies the current situation faced by the City:

- To distribute new or revised documentation to contractors, the City sends the documents manually through email. There are no automated processes in place and this work flow is done manually for each document. There are dozens of contractors which these documents need to be distributed to, and those contractors themselves also change over time. The list of these contractors is also managed manually.
- Verification of work performed by contractors currently requires a physical visit to site. There is no automated process to collect information

such as photos of the work performed, other than manually through email.

- It is currently difficult to know whether the labour cost billed by the contractors on their invoices is reflective of the actual work performed, as that information is only recorded in a physical logbook at each site and is not electronic. A contractor could overcharge for the work and it would be difficult to prove that their bill is inaccurate.

6.2 The Context of the Work

The below figure illustrates a high-level context model of the adjacent systems:

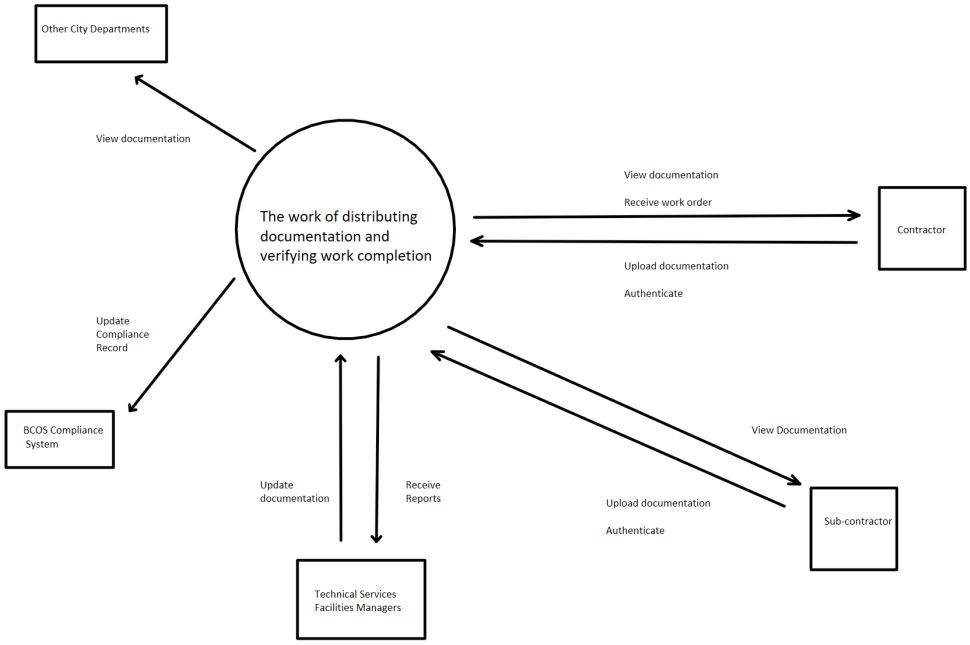

Figure 1: Context model

6.3 Work Partitioning

Event Name	Input and Output
1. Contractor Login	Contractor email and one time password (in)
2. Contractor Location Verification	GPS location (in)
3. Staff Login	Staff email and password (in)
4. Sign Document	unsigned document (in), signed document (out)
5. Upload Report	report file (in)
6. Export Reports	report specifications (in), report data (out)
7. View Station Entry Protocols	Station location (in), entry protocol (out)
8. Add Contractor to System	Contractor information (in)
9. View Contractor Summary	Contractor name (in), contractor data (out)
10. Certification Expiry	notice of document expiration to staff (out)

6.4 Specifying a Business Use Case (BUC)

The below points identify the business use cases for each of the business events by number in section 6.3.

1. Contractor authenticates themselves to City systems, proving that they are authorized to be provided City information.

2. Contractor proves they are at a site for transparency when work is being completed.

3. Staff authenticates themselves as authorized to view private data.

4. Sign a document for contractual or regulatory purposes.

5. Documents are stored for record keeping and relevant stakeholders are notified as necessary.

6. Export data of a specific type specified by the user. For example, all crane inspections in the last year.

7. View a procedure of how staff and contractors must access a specified station.

8. New contractors are frequently onboarded, and their information is recorded by the facilities managers.

9. Facilities managers commonly need to see which employees of a contractor are due for their health and safety training, WSIB expiry, etc.

10. When a piece of documentation has expired, the next version needs to be put in place by staff to replace the expired version.

7 Business Data Model and Data Dictionary

7.1 Business Data Model

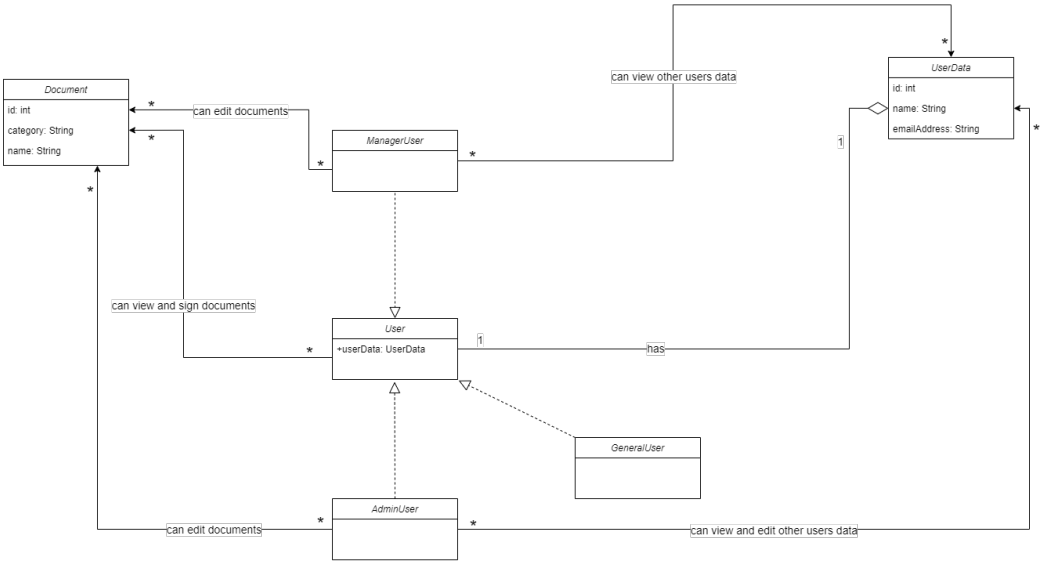

Figure 2: Data Model

7.2 Data Dictionary

Classes

- User
  - This class represents a user of the portal
  - **Attributes:**
    - userData: UserData – It is the data related to the user in question
  - **Relationships:**
    - A User can view and sign multiple Document objects (many-to-many relationship)
    - A User has a UserData object storing information about the user in question (one-to-one relationship)

  - **Sub-Classes:**
    * AdminUser
    * ManagerUser
    * GeneralUser

- AdminUser
  - This class is a sub-class of User class which reprsents an admin user of the portal
  - **Attributes:**
    * Same as User class
  - **Relationships:**
    * An AdminUser can view and edit other users data (many-to-many relationship)
    * An AdminUser can edit multiple documents (many-to-many relationship)

- ManagerUser
  - This class is a sub-class of User class which reprsents a user who holds a manager position
  - **Attributes:**
    * Same as User class
  - **Relationships:**
    * An AdminUser can view other users data (many-to-many relationship)
    * An AdminUser can edit multiple documents (many-to-many relationship)

- GeneralUser
  - This class is a sub-class of User class which reprsents a general user of the poral (such as a someone from the city staff or a contractor)
  - **Attributes:**
    * Same as User class

  - **Relationships:**
    - Same as User class

- Document
  - This class represents a document stored on the portal
  - **Attributes:**
    - id: int – The id of the document
    - category: String – The category of the document (such as "health and safety")
    - name: String – The name of the document (such as "first aid instructions")
  - **Relationships:**
    - A Document can be viewed and signed by many User objects (many-to-many relationship)
    - A Document can be edited by many AdminUser and ManagerUser objects (many-to-many relationship)

- UserData
  - This class represents the information of a given user
  - **Attributes:**
    - id: int – The id of the document
    - name: String – The name of the user
    - emailAddress: String – The email address of the user
  - **Relationships:**
    - A UserData object is linked to one User object (one-to-one relationship)
    - UserData objects of given users can be accessed by many AdminUser objects (many-to-many relationship)
    - UserData objects of given users can be accessed by many ManagerUser objects (many-to-many relationship)

8 The Scope of the Product

8.1 Product Boundary

The table below displays aspects of the application which are in scope and aspects which are out of scope.

In Scope	**Out of Scope**
Designing the application in a way where it is able to interact with the external work order system is a stretch goal.	Designing any aspects of a full work order system is out of scope for this project.
The application will implement an authentication system sufficient to demonstrate contractors authenticating themselves onto sites.	A thorough cybersecurity analysis for the authentication is not in scope as it is not the primary focus for the applications functionality.

8.2 Product Use Case Table

User	Use Case
Facilities Manager	- View contractor authentication data to determine that a contractor was at a particular location and completed the work expected and invoiced accordingly. - Generate a report of compliance documentation in the system for easy sharing to relevant third parties. - Manage the health and safety documentation and training status of their contracts.
Contractors	- Authenticate their arrival to site to confirm work is being performed. - Access station entry and exit procedures, hazard forms, and other safety information which should be known when accessing the station. - Sign and acknowledge forms and return them to the City. - Upload photographs, documents, and comments about work done at the station.

8.3 Individual Product Use Cases (PUC's)

N/A.

9 Functional Requirements

9.1 Formal Definitions

- Let $U = \{u_1, u_2, \cdots, u_i\}$ be the set of users in the system.

- Let $R = \{r_1, r_2, \cdots, r_r\}$ be the set of roles (e.g. manager, contractor, subcontractor).
- Let $A = \{a_1, a_2, \cdots, a_s\}$ be the set of actions (e.g. upload, authenticate, sign, view).
- Let $C = \{c_1, c_2, \cdots, c_t\}$ be the set of compliance requirements (e.g. safety training, hazard assesments).
- Let $I = \{l_1, l_2, \cdots, l_q\}$ be the set of allowed locations.
- Let $loc(u_i)$ represent the current location of user u_i.
- Let $D = \{d_1, d_2, \cdots, d_n\}$ be the set of documents.
- Let $hasRole(u_i, r_k)$ represent that user $u_i \in U$ has a role $r_k \in R$.
- Let $notify(u_m, u_i)$ represent a notification from user $u_m \in U$ to user $u_i \in U$.
- Let $notify(u_i)$ represent a notification from the system to user $u_i \in U$
- Let $manages(u_m, u_i)$ represent that user $u_m \in U$ manages user $u_i \in U$.
- Let $permitted(r_k, a_s, d_j)$ represent that a user with role $r_k \in R$ is permitted to perform action $a_s \in A$ on document $d_j \in D$
- Let $requires(a_s, c_t)$ represent that action $a_s \in A$ requires compliance requirement $c_t \in C$
- Let $T(u_i) \subset C$ represent the set of completed training for a user $u_i \in U$.
- Let $assoc(d_j, u_i)$ denote that document d_j is associated with user u_i.
- Let $upload(u_i, d_j)$ represent the action of user $u_i \in U$ uploading document $d_j \in D$.
- Let $performAction(u_i, a_s, d_j)$ represent that user $u_i \in U$ performs action $a_s \in A$ on document $d_j \in D$.
- Let $expired(c_i)$ represent that compliance document $c_i \in C$ has expired.

9.2 Formal Expressions

1. $\forall u_i \in U, \forall d_j \in D, \forall r_k \in R, \forall a_s \in A$:

   $$hasRole(u_i, r_k) \land permitted(r_k, a_s, d_j) \implies performAction(u_i, a_s, d_j)$$

2. $\forall u_i \in U, \forall c_t \in C$:

   $$(requires(a_s, c_t) \land a_s = sign \implies (c_t \in T(u_i)) \implies performAction(u_i, a_s, d_j))$$

3. $\forall u_i \in U, \forall d_j \in D$:

   $$(loc(u_i) \in L) \implies upload(u_i, d_j)$$

4. $\forall u_i \in U, \forall d_j \in D$:

   $$(loc(u_i) \notin L) \implies \neg upload(u_i, d_j)$$

5. $\forall u_i \in U, \forall d_j \in D$:

   $$sign(u_i, d_j) \implies assoc(d_j, u_i)$$

6. $\forall u_i \in U, \forall c_t \in C, \forall u_m \in U$:

   $$(manages(u_m, u_i) \land c_t \notin T(u_i)) \implies notify(u_m, u_i)$$

7. $\forall u_i \in U, \forall c_t \in C$

   $$(expired(c_t) \implies notify(u_i))$$

9.3 Functional Requirements

FR1. System should support at a minimum the docx, xlsx, and pdf file formats for upload and viewing.

FR2. Documents in this system must be synchronized with Sharepoint, this means any changes in this system must be reflected in the corresponding Sharepoint document, and any changes to the document in Sharepoint must be reflected in this system.

FR3. The system should have different access levels depending on the type of user. Only Admin users should be able to change access permissions of other users and have read and write accesses to all documents in the system. Contractors/general users should only have read access to documents and be able to sign documents. See (1) in **Section 9.2 Formal Expressions**.

FR4. The current site, work order number, job details, and entry/exit time of contractors should be visible to admin users in the system.

FR5. The system must implement a form of geolocation verification before contractors are able to authenticate and upload documents. See (3) and (4) in **Section 9.2 Formal Expressions**.

FR6. The system must notify users when a complicance document has expired. See (7) in **Section 9.2 Formal Expressions**.

FR7. The system must store the date, time, name of person, and name of document when a user acknowledges a document. See (5) in **Section 9.2 Formal Expressions**.

FR8. The system must notify the manager of a contractor when a contractor attempts to authenticate and upload documents at a particular site prior to completion of required training. See (6) in **Section 9.2 Formal Expressions**.

FR9. The system must prevent a user from authenticating or uploading documents if the required training for the specific site is not complete. See (2) in **Section 9.2 Formal Expressions**.

10 Look and Feel Requirements

10.1 Appearance Requirements

LF-AP1. The colour palette of the application should align with other applications used by the city of Hamilton.
Rationale: Aligning with the look of current applications at the city of Hamilton creates a consistent feel for current employees using the application.

10.2 Style Requirements

LF-ST1. The application should have a responsive design which considers different screen sizes and orientations.
Rationale: The application may be used on a variety of different devices and therefore must be able to account for varying screen sizes and orientations.

11 Usability and Humanity Requirements

11.1 Ease of Use Requirements

UH-EU1. Users between the ages of MIN_AGE and MAX_AGE, regardless of technical expertise, must be able to discover at least 70% of the web application functionality within 10 minutes of being introduced to the system without any explaination or training.
Rationale: Usability is of high importance to the users of the system. They will be mainly non-technical and must be able to learn and use the system quickly if it is to be of any value to them.

UH-EU2. The system should provide undo options or warnings for irreversible actions and 95% of users should successfully use these features when prompted.
Rationale: It is important to provide users of the systems with confirmations or warnings to prevent errors and if they do happen, make it easy to recover from them.

11.2 Personalization and Internationalization Requirements

N/A

11.3 Learning Requirements

UH-LR1. Users between the ages of MIN_AGE and MAX_AGE should not take more than 10 minutes to a learn feature of the web application after it is discovered.
Rationale: A well designed system should give the user the most amount of functionality while being easy to understand. Users will be able to productive with the system in a short amount of time, reducing frustration.

UH-LR2. Users between the ages of MIN_AGE and MAX_AGE should be able to complete the system onboarding tutorial or walkthrough within 5 minutes, and 80% of users should report feeling confident in using the system afterward.
Rationale: It is important that users are able to understand system documentation quickly to reduce mental overhead and increase productivity.

11.4 Understandability and Politeness Requirements

UH-UP1. The application must not contain symbols or allusions that may be offensive or politically charged.
Rationale: As the system is to be used by people of diverse backgrounds and in a professional setting, it is important to keep a professional and neutural tone.

UH-UP2. System error messages should clearly explain the issue and suggest a resolution within 2 sentences. 90% of users between the ages of MIN_AGE and MAX_AGE should report understanding the error and being able to resolve the issue without external support.
Rationale: Useful system feedback is crucial since users need to know what the system is doing and how it interprets their input in order to determine next steps.

11.5 Accessibility Requirements

UH-AS1. The application must adhere to accessibility policies set by the City Of Hamilton Linked Here.
Rationale: It is important to meet accessibility standards so that the system is easy to use and learn for people of different backgrounds and abilities.

12 Performance Requirements

12.1 Speed and Latency Requirements

PR-SL1. Average time for retrieving the documents should not exceed 2 seconds

PR-SL2. Synching documents with SharePoint should be completed in less than 7 seconds for file sizes up to 15 MB

PR-SL3. The portal's user interface should be responsive and should react accordingly with the user interactions.

12.2 Safety-Critical Requirements

PR-SC1. All documents should be verified for integrity during upload and retrieval

PR-SC2. Document acesss restriction should be based on user roles

12.3 Precision or Accuracy Requirements

PR-PA1. Search queries related to documents or user data should return at least 95% relevant results

PR-PA2. A data integrity of 99.99% should be maintained to ensure no corruption or data loss during upload or retrieval

12.4 Robustness or Fault-Tolerance Requirements

PR-RFT1. The system should handle unexpected inputs and events without crashing

PR-RFT2. The system should be able to detect and alert on any anomalies or potential failures

12.5 Capacity Requirements

PR-CR1. The system should be able to support peak data transfer rate of 1 Gbps for document uploads and downloads

PR-CR2. The system should be able to support individual file sizes up to 1GB

12.6 Scalability or Extensibility Requirements

PR-SE1. The system should allow nesting of n number of documents and categories. For example, document category A can have 5 documents and sub document category B under it. (which in turn has sub-categories of documents under it).

PR-SE2. The system should be able to distribute traffic evenly and avoid single points of failures

12.7 Longevity Requirements

PR-LR1. The system should have a modular design to easily allow the introduction of new features

13 Operational and Environmental Requirements

13.1 Expected Physical Environment

OE-PE1. Application should be functional in City of Hamilton, Water Division sites and offices.

13.2 Wider Environment Requirements

OE-WE1. Application should be functional on Mobile and Desktop web browser layouts.

OE-WE2. Application should be able to run on Chrome, Microsoft Edge, and Mobile Browsers.

13.3 Requirements for Interfacing with Adjacent Systems

OE-IAS1. ~~Application should integrate with existing SharePoint repositories.~~

OE-IAS2. ~~Application should be able to provide up-to-date Safety Data Sheets from MySDS.~~
Direct integration with SharePoint and MySDS is no longer required by the stakeholders due to city policy and security concerns. The requirement for related functionality is covered by SR-IR2.

OE-IAS3. Application should be open for integration with upcoming Work Order tracking system in the city's Enterprise Asset Management software.

13.4 Productization Requirements

N/A

13.5 Release Requirements

OE-REL1. A changelog should be generated with every release documenting changes in features, requirements and fixes made.

OE-REL2. A release is defined as a Revision. Every revision should be a major deployment of new features and/or fixes into production.

OE-REL3. Expected release of Revision 0: February 1st, 2024

OE-REL4. Expected release of Revision 1: March 30th, 2025

14 Maintainability and Support Requirements

14.1 Maintenance Requirements

MS-MTN1. A deployment of the system should take no more than 30 minutes (not including testing, and building time).

MS-MTN2. The build time of the system should be no longer than 10 minutes (not including testing time).

MS-MTN3. All automated tests should be able to run in under 10 minutes

MS-MTN4. The system should have rigourous unit testing, line coverage should be $\geq 95\%$, branch coverage should be $\geq 90\%$.

MS-MTN5. All core functionalities of the system (i.e. Functional Requirements), should have both automated end-to-end and unit testing corresponding to them

MS-MTN6. The project must be able to be maintained by its users, as original developers will not be maintaining it after April 2, 2025.

14.2 Supportability Requirements

MS-SUP1. The application should have user-facing documentation on how to use the core functionalities of the system (i.e. functionalities described in functional requirements).

MS-SUP2. The application should have documentation for all API's for future maintainers.

MS-SUP3. The application should have documentation of internal functions and abstractions for future maintainers.

MS-SUP4. The application should have documentation on deployment, so users can deploy this application for themselves.

14.3 Adaptability Requirements

MS-ADP1. The application must be able to run on at least Google Chrome and Microsoft Edge browsers.

MS-ADP2. The application must be able to run on tablets, smartphones, and laptops.

MS-ADP3. The application must be able to run on Android, IOS, and Windows 10

15 Security Requirements

15.1 Access Requirements

SR-AR1. Only authorized contractors and employees should be able to access the portal for completing work on site.

SR-AR2. Only users with managerial level access should be able to get data on contractors and employees.

SR-AR3. Only users with administrative access should be able to make changes to site settings.

SR-AR4. The upload and changing of certain file classes will be tied to different access levels.

15.2 Integrity Requirements

SR-IR1. The application should not allow the addition, removal or modification of files by users without proper access and authentication.

SR-IR2. The documents displayed should match the appropriate version available on SharePoint and/or MySDS.

SR-IR3. The application should minimize the amount of bad or incomplete data inputted through the portal.

15.3 Privacy Requirements

SR-PR1. Data transmission will be encrypted.

SR-PR2. The application should protect private information in accordance with any and all government privacy laws and policies.

15.4 Audit Requirements

SR-AU1. The application will adhere to any and all audit rules followed by City departments.

15.5 Immunity Requirements

SR-IMR1. Security vulnerabilities and weaknesses should be patched within a week of their discovery.

15.6 Safety Requirements

SR-S1. The system shall notify the facilities manager if a contractor declines to sign a document.

16 Cultural Requirements

16.1 Cultural Requirements

CR-CR1. The application must abide by the City of Hamilton's strategic plan, which is outlined here.

17 Compliance Requirements

17.1 Legal Requirements

CR-L1. The application is to strictly abide by all regulations governing the treatment of water and wastewater. This includes, but is not limited to the *Safe Drinking Water Act* (SDWA), 2002, the *Ontario Water Resources Act* (OWRA), the *Environmental Protection Act* and the

Clean Water Act.

CR-L2. The application is to process any sensitive information per the *Personal Information and Protection and Electronic Documents Act* and the *Freedom of Information and Protection of Privacy Act*.

17.2 Standards Compliance Requirements

CR-S1. Documentation which must be stored on the City's BCOS system are to be stored on the application in a manner which can transfer the data. Our application will align with the City's DWQMS (Drinking Water Quality Management System).

18 Open Issues

1. The authentication methods for new users must meet security standards required for deployment. A feasible solution which falls within the scope of the project has yet to be determined.
2. The security of the connection between the application and the city's resources must be at a certain standard to be deployed. The feasibility of reaching the required standards has yet to be determined.
3. The feasibility of integrating the application with the City of Hamilton's current systems (e.g., MySDS, SharePoint) has yet to be determined. We require documentation and testing to assess compatibility.
4. The feasibility of integration with upcoming systems, such as the work order tracking system in Infor EAM and security solutions, has yet to be determined. We require further information and documentation from the City.

19 Off-the-Shelf Solutions

19.1 Ready-Made Products

Currently there exist many document management systems (i.e. Google Docs, Sharepoint). However, They miss some of the clients major require-

ments. The city wants to be able to integrate with their work order management system to show the status of a work order that is associated with any given document, but existing solutions do not provide this capability. They also want to be able to verify that people were at a given site, when completing work, which again there isn't a ready made product to do.

19.2 Reusable Components

We can use Sharepoint as file storage, since the city wants Sharepoint and this system to be in sync, and storing the files in two seperate locations and then syncing them will introduce a lot of overhead. Instead, all files can just be stored on Sharepoint.

19.3 Products That Can Be Copied

N/A

20 New Problems

20.1 Effects on the Current Environment

1. The application should recognize and interact with existing systems in a way that complements rather than competes with them. It should leverage existing data and processes instead of recreating or duplicating them. It should only introduce new workflows or tasks when no suitable existing solution is in place.

2. If an existing business process can handle a particular task more effectively, the application should delegate that task rather than attempt to perform it redundantly.

20.2 Effects on the Installed Systems

1. The application should not change or interfere with the host system's configuration, performance, or files except for the necessary input and output operations.

2. When interacting with other systems, the application should only retrieve necessary data and send data if required, but only as specified, without altering or influencing the external systems' operations or configurations.

20.3 Potential User Problems

1. The user may not have access to the internet.
2. The user may not have a device which can run the application.
3. The user may not be comfortable with giving the application permission to view their location.
4. The user may not follow the intended flow for the system.
5. The user may forget to logout or end their session when leaving site.

20.4 Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

N/A

20.5 Follow-Up Problems

1. Business processes might change, changing the requirements of the application.
2. New software solutions may be introduced which make some features redundant.
3. Regulations may change adding or removing requirements.

21 Tasks

21.1 Project Planning

Project deliverables should be completed by the deadlines given in the course outline. GitHub will be used to track project milestones and tasks. Tasks

will be assigned to individual team members or to groups. All work will be reviewed by other members of the team before being committed to the project. Feedback received from stakeholders, TAs, or the professor will be implemented in the project, and requirements will be changed accordingly.

Task 1. Set-up codebase and begin development of project.

Task 2. Work on documentation and deliverables.

Task 3. Get feedback from stakeholders, TAs, and the professor and implement suggested changes.

21.2 Planning of the Development Phases

1. *Proof of Concept*: Will start development after October 9th, 2024. Aim to complete by November 4th.
2. *Rev. 0*: Aim to complete by February 1st, 2024.
3. *Rev. 1*: Aim to complete by March 30th, 2024.
4. *Future revisions*: TBD

22 Migration to the New Product

22.1 Requirements for Migration to the New Product

MI-NP1. The system must be compatible with existing user roles in Active Directory.
Rationale: Compatibility with existing business rules is needed to ensure migration can be completed in a short period of time without having to defined new roles.

22.2 Data That Has to be Modified or Translated for the New System

MI-TR1. Information stored on paper must be digitized for consumption for the system.
Rationale: Some content that the system is to consume has not

yet been digitized. It will have to be digitized before the system is able to use it.

23 Costs

The cost for the application should not exceed $750 unless approved by the professor and the stakeholders for the project.

It is expected that the team will spend 40 man-hours per week on the project until its completion.

Item	Cost	Description
Cloud Services	$ TBD	Amazon Web Services (AWS)
Domain Name	$ TBD	TBD

24 User Documentation and Training

24.1 User Documentation Requirements

UDT-DR1. A user manual is required which outlines a procedure for the stakeholder use cases. This is needed for the facilities managers user group.

UDT-DR2. A set-up manual specifying how both city employees and contractors can create an account in the system. This manual should include troubleshooting techniques for common problems.

24.2 Training Requirements

UDT-TR1. A training video will need to be recorded to demonstrate the functionality of the final product. In particular, it must show how each of the facilities managers use cases are performed. The training video must available to anyone the City feels would benefit from this training.

25 Waiting Room

There are no more requirements that will be added to the system as of this moment. This may change as the project evolves in the future.

26 Ideas for Solution

1. Have QR codes available to users during entry to a site. These codes will direct new users to the portal. Once on the portal, email verification can be used to authenticate their identity. The name and email should match the City's contractor or employee records.

2. Implement GPS functionality within the app to verify if users are physically present on-site when using the portal.

3. Generate and provide new users with one-time passcodes to grant access to the portal and authenticate user identity.

4. Use modular design that supports future integration with upcoming systems. Use dummy APIs to test the functionality.

Appendix — Symbolic Parameters

$MIN\_AGE = 18$
$MAX\_AGE = 70$

Appendix — Reflection

1. What went well while writing this deliverable?

   Kyle - For this deliverable we had worked on the document much earlier in advance of the deadline than we did for the problem statement and goals, which led to a higher quality of work. Also since we started earlier we had much less issues with merge conflicts than last time when we all started merging at about the same time.

   Mitchell - I think that our team improved upon our workflow from the first deliverable. The delegation of tasks and management of outstanding work in Github issues was executed by the team much more effectively.

   Rafeed - We decided to use LaTeX for this deliverable, which had a much better workflow, especially when it came time to do PRs and merges. The division of labour also worked out really well, with everyone doing their parts on time and with high quality. Kyle and Richard did a great job reviewing my PRs objectively, increasing the quality of work I submitted.

   Richard - I think our team did a good job in translating high level use cases and requirements from our stakeholders into system requirements that we can implement. Communication between our team and the stakeholders went smoothly which gave us ample time to elicit potential requirements.

   Akshit - I think our team did a fantastic job overall for this deliverable. The particular thing I liked was how we were able to brainstorm the together the requirements, constraints and potential problems for the project. All team members helped each other whenever required

and as a result we were able to finish smoohtly with this deliverable.

2. What pain points did you experience during this deliverable, and how did you resolve them?

   Kyle - Some of our team members had a better idea of the stakeholders and their current workplace environment whereas others had a better understanding of their vision for the project and their requirements. We resolved this by attempting to divide up work based on peoples strengths, while still adding everyone to PR's as reviewers so they could try to understand the sections that maybe weren't their strength or at least attempt to get a better understanding.

   Mitchell - Viewing the most up to date version of the pdf is difficult as commiting it to version control creates merge conflicts. Our team decided to work around this problem by not commiting the updated version of the pdf's in our branches, and commiting a final good version at the end of the deliverable.

   Rafeed - The workflow does not allow for live collaboration so it is difficult to get the input of other team members on the parts we are doing without them having to review your PRs, but everyone was very diligent with their reviews so there wasn't too long of a delay to get everyone's ideas together. We also had multiple meetings in-between to get everyone on the same page.

   Richard - The division of work was somewhat uneven for this deliverable. Some team members were more familiar with the stakeholder's business and use cases which led to them contributing more than others. This was mitigated through participation in document reviews and conversations with stakeholders so that the rest of the team could also learn more about the stakeholder and contribute more in the future.
   Akshit - I think one pain point that we experienced during this deliverable was setting up Latex locally on our machines. Sometimes, some team members received unexpected errors during compilation but this was easily overcome by asking help from the fellow members.

3. How many of your requirements were inspired by speaking to your client(s) or their proxies (e.g. your peers, stakeholders, potential users)?

   The vast majority of our requirements were informed by speaking to our clients or their proxies. Only really the look and feel, and performance requirements were not informed by direct discussion with the City.

4. Which of the courses you have taken, or are currently taking, will help your team to be successful with your capstone project.

   Kyle - The requirements course (3RA3) will be very helpful because it taught us about requirements documents and how to elicit requirements. 3A04 will also be useful because it was the first course where we applied requirements documents to a project, and taught us about architecture styles for larger systems like the one we will make in this project. The software testing course (3S03) will be useful when it comes to creating test cases for this project and verifying the requirements. The human computer interfaces course (4HC3) that we are currently taking will help with informing decisions about the UI of our application and making sure to keep our clients in mind for all design decisions.

   Mitchell - The introduction to software development course (2AA4) will be useful as it serves as the basis for software design which we will use extensively throughout this project.

   Rafeed - 3A04 was good experience for working on large-scale development as a team. 3RA3 was useful in creating this document and eliciting requirements from the stakeholders. 3S03 will be useful when creating a test plan. 4HC3 will be useful when designing an intuitive user-friendly interface.

   Richard - The former 3XA3 would help as it taught a lot about the software development process and has many deliverables that overlaped with the capstone course. For more specialized skills such as requirement writing and test plan writing, the skills learned from 3RA3 and 3S03 will be helpful.

Akshit - The SE 3RA3 and 3A04 courses have been instrumental in my success so far in the course so far as they provided me hands on experience writing SRS documents. I also believe the experience from SE 2AA4 and 3BB4 courses will help me during the implementation phase.

5. What knowledge and skills will the team collectively need to acquire to successfully complete this capstone project? Examples of possible knowledge to acquire include domain specific knowledge from the domain of your application, or software engineering knowledge, mechatronics knowledge or computer science knowledge. Skills may be related to technology, or writing, or presentation, or team management, etc. You should look to identify at least one item for each team member.

   Kyle - Personally I have a lot of experience with using AWS and python, so the backend part of our tech stack is not much of a concern for me. However, I have limited experience with javascript or the NextJS framework that we plan on using, so I need to learn more about those.

   Mitchell - I have limited experience working in either frontend or backend design. I will have to work closely with my team members and the instructional team to develop these needed skills throughout the project. My strength is my project management experience and being a subject matter expert in the client's area of work (water and wastewater plant maintenance).

   Rafeed - I have experience with solution delivery and project management from my co-op. It may be useful in for this capstone project when generating new features, deciding to cut features, or if the project faces any roadblock.

   Richard - My skills are in systems programming and robotics, so most web technologies required for this project will be new to me. While I do have previous working with web technologies such as Node.Js and React, I will have to review these skills to refresh my understanding.

   Akshit - I have experience in both frontend and backend but I have

not worked with cloud technologies in the past on a regular basis. Since we are planning to use AWS, I will have to learn it and I am looking forward to it.

6. For each of the knowledge areas and skills identified in the previous question, what are at least two approaches to acquiring the knowledge or mastering the skill? Of the identified approaches, which will each team member pursue, and why did they make this choice?

   Kyle - In order to learn the javascript and NextJS I will watch videos online explaining them and how they work. I will also read the documentation for both and try using them in my spare time to get a better understanding of them.

   Mitchell - When we have decided on the specific technologies to use, I will follow the introductory tutorials provided by the creators of those software products. I will document my progress of this learning on Github. I will also discuss with my team members what tasks would be suitable for me to work on at my skill level, to help develop my skills while still contributing to the teams success.

   Rafeed - Learning the tech stack we will be using is crucial for providing value to this project. Two approaches to learning would be to read documentation, and watch guides. While I will be doing both, I will be mostly watching guides and following along, as I feel I learn the best this way.

   Richard - The best approach for me is to build. Hands on experience with the technology helps me learn pratical uses quickly. I also find that doing improves my research skills relating to that technology through reading the documentation.

   Akshit - I planning on watching tutorials on youtube related to AWS and also completing the basic certification course offered by Amazon. I also plan on building a small personal learning project on the side which uses AWS.

References

[1] J. Robertson and S. Robertson, *Volere Requirements Specification Template*, 16 ed., 2012.