

Software Requirements Specification for SFWRENG 4G06 - Capstone Design Process: subtitle describing software

Team 17, DomainX

Awurama Nyarko
Haniye Hamidizadeh
Fei Xie
Ghena Hatoum

October 8, 2025

Table 1: Revision History

Date	Developer(s)	Change
October 6, 2025	Fei Xie	First Draft of Sections: Cost to Ideas for Solution
Date2	Name(s)	Description of changes
...

Contents

1	Purpose of the Project	vi
1.1	User Business	vi
1.2	Goals of the Project	vi
2	Stakeholders	vi
2.1	Client	vi
2.2	Customer	vi
2.3	Other Stakeholders	vi
2.4	Hands-On Users of the Project	vi
2.5	Personas	vi
2.6	Priorities Assigned to Users	vi
2.7	User Participation	vii
2.8	Maintenance Users and Service Technicians	vii
3	Mandated Constraints	vii
3.1	Solution Constraints	vii
3.2	Implementation Environment of the Current System	vii
3.3	Partner or Collaborative Applications	vii
3.4	Off-the-Shelf Software	vii
3.5	Anticipated Workplace Environment	vii
3.6	Schedule Constraints	vii
3.7	Budget Constraints	vii
3.8	Enterprise Constraints	viii
4	Naming Conventions and Terminology	viii
4.1	Glossary of All Terms, Including Acronyms, Used by Stakeholders involved in the Project	viii
5	Relevant Facts And Assumptions	viii
5.1	Relevant Facts	viii
5.2	Business Rules	viii
5.3	Assumptions	viii
6	The Scope of the Work	viii
6.1	The Current Situation	viii
6.2	The Context of the Work	viii
6.3	Work Partitioning	ix

6.4	Specifying a Business Use Case (BUC)	ix
7	Business Data Model and Data Dictionary	ix
7.1	Business Data Model	ix
7.2	Data Dictionary	ix
8	The Scope of the Product	ix
8.1	Product Boundary	ix
8.2	Product Use Case Table	ix
8.3	Individual Product Use Cases (PUC's)	ix
9	Functional Requirements	ix
9.1	Functional Requirements	ix
10	Look and Feel Requirements	x
10.1	Appearance Requirements	x
10.2	Style Requirements	x
11	Usability and Humanity Requirements	x
11.1	Ease of Use Requirements	x
11.2	Personalization and Internationalization Requirements	x
11.3	Learning Requirements	x
11.4	Understandability and Politeness Requirements	x
11.5	Accessibility Requirements	x
12	Performance Requirements	x
12.1	Speed and Latency Requirements	x
12.2	Safety-Critical Requirements	xi
12.3	Precision or Accuracy Requirements	xi
12.4	Robustness or Fault-Tolerance Requirements	xi
12.5	Capacity Requirements	xi
12.6	Scalability or Extensibility Requirements	xi
12.7	Longevity Requirements	xi
13	Operational and Environmental Requirements	xi
13.1	Expected Physical Environment	xi
13.2	Wider Environment Requirements	xi
13.3	Requirements for Interfacing with Adjacent Systems	xii
13.4	Productization Requirements	xii

13.5 Release Requirements	xii
14 Maintainability and Support Requirements	xii
14.1 Maintenance Requirements	xii
14.2 Supportability Requirements	xii
14.3 Adaptability Requirements	xii
15 Security Requirements	xii
15.1 Access Requirements	xii
15.2 Integrity Requirements	xii
15.3 Privacy Requirements	xiii
15.4 Audit Requirements	xiii
15.5 Immunity Requirements	xiii
16 Cultural Requirements	xiii
16.1 Cultural Requirements	xiii
17 Compliance Requirements	xiii
17.1 Legal Requirements	xiii
17.2 Standards Compliance Requirements	xiii
18 Open Issues	xiii
19 Off-the-Shelf Solutions	xv
19.1 Ready-Made Products	xv
19.2 Reusable Components	xv
19.3 Products That Can Be Copied	xv
20 New Problems	xviii
20.1 Effects on the Current Environment	xviii
20.2 Effects on the Installed Systems	xix
20.3 Potential User Problems	xix
20.4 Limitations in the Anticipated Implementation Environment That May Inhibit the New Product	xx
20.5 Follow-Up Problems	xx
21 Tasks	xx
21.1 Project Planning	xx
21.2 Planning of the Development Phases	xxi

22 Costs	xxi
23 User Documentation and Training	xxii
23.1 User Documentation Requirements	xxii
23.1.1 User Manual	xxii
23.1.2 Release Manual	xxii
23.2 Training Requirements	xxii
24 Waiting Room	xxiii
25 Ideas for Solution	xxiii

1 Purpose of the Project

1.1 User Business

Insert your content here.

1.2 Goals of the Project

Insert your content here.

2 Stakeholders

2.1 Client

Insert your content here.

2.2 Customer

Insert your content here.

2.3 Other Stakeholders

Insert your content here.

2.4 Hands-On Users of the Project

Insert your content here.

2.5 Personas

Insert your content here.

2.6 Priorities Assigned to Users

Insert your content here.

2.7 User Participation

Insert your content here.

2.8 Maintenance Users and Service Technicians

Insert your content here.

3 Mandated Constraints

3.1 Solution Constraints

Insert your content here.

3.2 Implementation Environment of the Current System

Insert your content here.

3.3 Partner or Collaborative Applications

Insert your content here.

3.4 Off-the-Shelf Software

Insert your content here.

3.5 Anticipated Workplace Environment

Insert your content here.

3.6 Schedule Constraints

Insert your content here.

3.7 Budget Constraints

Insert your content here.

3.8 Enterprise Constraints

Insert your content here.

4 Naming Conventions and Terminology

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

Insert your content here.

5 Relevant Facts And Assumptions

5.1 Relevant Facts

Insert your content here.

5.2 Business Rules

Insert your content here.

5.3 Assumptions

Insert your content here.

6 The Scope of the Work

6.1 The Current Situation

Insert your content here.

6.2 The Context of the Work

Insert your content here.

6.3 Work Partitioning

Insert your content here.

6.4 Specifying a Business Use Case (BUC)

Insert your content here.

7 Business Data Model and Data Dictionary

7.1 Business Data Model

Insert your content here.

7.2 Data Dictionary

Insert your content here.

8 The Scope of the Product

8.1 Product Boundary

Insert your content here.

8.2 Product Use Case Table

Insert your content here.

8.3 Individual Product Use Cases (PUC's)

Insert your content here.

9 Functional Requirements

9.1 Functional Requirements

Insert your content here.

10 Look and Feel Requirements

10.1 Appearance Requirements

Insert your content here.

10.2 Style Requirements

Insert your content here.

11 Usability and Humanity Requirements

11.1 Ease of Use Requirements

Insert your content here.

11.2 Personalization and Internationalization Requirements

Insert your content here.

11.3 Learning Requirements

Insert your content here.

11.4 Understandability and Politeness Requirements

Insert your content here.

11.5 Accessibility Requirements

Insert your content here.

12 Performance Requirements

12.1 Speed and Latency Requirements

Insert your content here.

12.2 Safety-Critical Requirements

Insert your content here.

12.3 Precision or Accuracy Requirements

Insert your content here.

12.4 Robustness or Fault-Tolerance Requirements

Insert your content here.

12.5 Capacity Requirements

Insert your content here.

12.6 Scalability or Extensibility Requirements

Insert your content here.

12.7 Longevity Requirements

Insert your content here.

13 Operational and Environmental Requirements

13.1 Expected Physical Environment

Insert your content here.

13.2 Wider Environment Requirements

Insert your content here.

13.3 Requirements for Interfacing with Adjacent Systems

Insert your content here.

13.4 Productization Requirements

Insert your content here.

13.5 Release Requirements

Insert your content here.

14 Maintainability and Support Requirements

14.1 Maintenance Requirements

Insert your content here.

14.2 Supportability Requirements

Insert your content here.

14.3 Adaptability Requirements

Insert your content here.

15 Security Requirements

15.1 Access Requirements

Insert your content here.

15.2 Integrity Requirements

Insert your content here.

15.3 Privacy Requirements

Insert your content here.

15.4 Audit Requirements

Insert your content here.

15.5 Immunity Requirements

Insert your content here.

16 Cultural Requirements

16.1 Cultural Requirements

Insert your content here.

17 Compliance Requirements

17.1 Legal Requirements

Insert your content here.

17.2 Standards Compliance Requirements

Insert your content here.

18 Open Issues

The following unresolved items could materially affect the design, deployment, or operation of the NNL Assessment Tool. The issues in Table [2](#) will be tracked until closure to reduce delivery risk and surprises in later phases.

Table 2: Open issues to track and resolve

Issue #	Summary	Cross-Reference	Stakeholder	Action Required	Status
OI-01	Finalize hosting environment for the tool (e.g., internal server or university-managed cloud).	Operational and Environmental Requirements	CAS Supervisor, Infrastructure Team	Confirm approved infrastructure with IT	Pending
OI-02	Confirm the final list of neural network libraries to include in the tool's database (based on domain-expert review).	Scope of the Work	Research Subteam, Domain Expert	Schedule and complete review with domain expert	Pending
OI-03	Select user interface (UI) framework and visualization library (e.g., React with Chart.js vs. D3).	Functional Requirements	Development Team	Evaluate options and record decision	In progress
OI-04	Decide Excel integration	Functional Requirements	Research Subteam, Development	Define use cases and confirm	Pending

19 Off-the-Shelf Solutions

This section identifies existing tools, software, and components that could be leveraged to reduce development time and cost for the Neural Network Libraries (NNL) Assessment Tool. It outlines reusable libraries and products that can be legally copied or adapted to accelerate development and ensure maintainability.

The goal is to reuse proven, reliable components and avoid reinventing existing functionality, thereby optimizing development effort and leveraging established best practices.

19.1 Ready-Made Products

Several existing platforms provide partial functionality aligned with the tool's goals, particularly in data visualization, analytics, and automation. However, none fully satisfy the requirement for automated data gathering, Analytic Hierarchy Process (AHP) analysis, and integrated visualization.

These products may serve as inspiration or integration points (e.g., Excel import/export) but cannot replace the custom automation and analysis required by the NNL Assessment Tool.

19.2 Reusable Components

The following open-source libraries and frameworks will be incorporated to support data collection, analysis, and visualization.

Reusing these components ensures consistency, leverages reliability, and minimizes custom development effort.

19.3 Products That Can Be Copied

Some open-source or academic research dashboards share functional similarities with the NNL Assessment Tool and may inform design or architecture.

Adaptation saves design time and provides validated frameworks for implementation while maintaining legal compliance.

Considerations:

- Licensing for third-party libraries must be reviewed before adoption.
- Integration testing is required to confirm compatibility.

Table 3: Ready-Made Products Relevant to the NNL Assessment Tool

Product	Description	Relevance to Project	Limitations
Microsoft Excel	Spreadsheet with storage, formulas, and charts.	Used as baseline for current manual process.	Lacks automation, scalability, and centralized access.
Google Sheets	Cloud-based collaborative spreadsheet.	Enables multi-user editing and sharing.	Limited automation, manual data import.
Power BI / Tableau	Advanced analytics and visualization tools.	Suitable for dashboards and comparative graphs.	Licensing cost, limited AHP customization.
SurveyMonkey / Google Forms	Online data collection tools.	Useful for gathering expert feedback.	No direct database or analytics integration.

Table 4: Reusable Components

Component	Purpose	Source	Justification
Python Pandas	Data manipulation and analysis.	Open-source	Handles tabular data and transformations efficiently.
Requests (Python)	Retrieve data from GitHub / PyPI APIs.	Open-source	Automates collection of repository metrics.
Matplotlib / Plotly / Chart.js	Visualization libraries.	Open-source	Create interactive and exportable graphs for dashboards.
AHPy / ahpy	Analytic Hierarchy Process implementation.	Open-source	Automates pairwise comparison scoring.
Flask	Backend web framework.	Open-source	Manages API integration and data handling.
React	Frontend user interface framework.	Open-source	Enables responsive dashboards and visualization.
MySQL / SQLite	Database systems.	Open-source	Store collected data and evaluation results.

- Long-term maintenance and documentation quality will influence selection.

Each reused or adapted product will be documented with:

- Name and source
- Functionality
- Integration plan
- Licensing details
- Selection status

20 New Problems

This section identifies potential conflicts, risks, or issues that may arise as a result of implementing the NNL Assessment Tool within McMaster University’s research environment. The purpose is to anticipate and document any negative effects or dependencies introduced by the new system.

20.1 Effects on the Current Environment

The new tool will operate within McMaster University’s research infrastructure and will rely on institutional hosting (e.g., internal servers). It may introduce additional server load and require IT resources for maintenance.

Motivation: To ensure the new tool integrates smoothly without disrupting existing research tools, storage systems, or academic workflows.

Examples:

- Increased data storage requirements may conflict with current quotas.
- Additional IT workload for managing hosting or user accounts.

Considerations:

- Coordination with the IT infrastructure team is required to confirm compatibility with university servers.

- Ensure the tool does not negatively affect access to existing research applications or networks.

Form: Documented assessment of integration impact with existing systems, supported by infrastructure review.

20.2 Effects on the Installed Systems

The new tool will interface with existing systems such as Excel, university authentication systems (SSO), and potentially university-hosted databases.

Motivation: Identify dependencies between the tool and existing platforms, ensuring stable coexistence and avoiding version conflicts.

Considerations:

- Compatibility with current Microsoft Excel versions.
- Security compliance when connecting to McMaster’s SSO.
- Avoid introducing vulnerabilities or version mismatches.

Form: Integration map specifying systems affected, their current versions, and compatibility requirements.

20.3 Potential User Problems

Potential user challenges include onboarding, usability learning curves, and confusion regarding data visualization or interpretation.

Motivation: Ensure researchers and users can adopt the system efficiently without frustration or misinterpretation of outputs.

Considerations:

- Provide user documentation and tutorials.
- Offer training sessions or quick-start guides.
- Establish a support contact for reporting issues.

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

Possible constraints include limited hosting capacity, restricted access to certain cloud features, and dependence on McMaster’s infrastructure approval.

Motivation: Identify environmental limitations that could delay deployment or reduce tool performance.

Examples:

- Hosting quotas may limit database scaling.
- University IT policies may restrict certain libraries or APIs.
- Limited access to high-performance computing resources.

Considerations: Execute a full review to confirm infrastructure readiness.

20.5 Follow-Up Problems

Potential long-term issues include sustaining the tool after project completion and keeping data updated as new Neural Network Libraries emerge.

Motivation: Ensure the system remains relevant and operational beyond the capstone timeline.

Considerations:

- Define ownership and maintenance responsibilities after project handover.
- Plan for version updates, new library integrations, and user feedback loops.
- Ensure continuity when original developers graduate.

21 Tasks

21.1 Project Planning

The Neural Network Libraries (NNL) Assessment Tool will be delivered using a hybrid development approach that combines Agile iterations with struc-

tured milestone-based deliverables aligned with the McMaster Software Capstone schedule.

The lifecycle is divided into major phases: Requirements, Design, Implementation, Testing, and Deployment, each building toward a functional and hosted tool that supports the research team in evaluating neural network libraries.

This approach ensures iterative feedback from supervisors and domain experts after each milestone, enabling continuous refinement. Development will be managed through GitHub for version control, VS Code for coding, and LaTeX for documentation.

The tool will be hosted on McMaster’s internal infrastructure or an approved equivalent, with key non-functional activities such as user onboarding, data migration, and training planned in the later stages.

21.2 Planning of the Development Phases

Each phase contributes to the development of a usable and reliable product. Feedback loops will be incorporated at each stage to ensure alignment with stakeholder expectations, compliance with requirements, and delivery of a secure, maintainable, and user-friendly system.

22 Costs

There is no development cost for this project, due to the nature of this being a capstone project.

The costs of hosting the required services, such as the database and the web application will depend on McMaster University’s existing infrastructure. However, estimating using a common cloud provider such as Amazon Web Services (AWS).

Using the Relational Database Service (RDS) for database and the Elastic Cloud Computing (EC2) service for hosting. Assuming around a maximum usage of 20 Hours/Month, the total cost of hosting is 11.63 CAD per month, as shown in Table 8.

23 User Documentation and Training

23.1 User Documentation Requirements

23.1.1 User Manual

The user manual will highlight the key features of the product, and provide additional details for installing, setup and usage that wasn't previously covered in the project's [README](#).

The maintenance of this document will be the responsibility of the development team. Changes to the product's features, such as adding new features or altering existing features must be reflected in the user manual upon release of the update.

23.1.2 Release Manual

The release manual will cover the release process required for future releases of the product, found in the [Development Plan](#). It should include the whole CI/CD lifecycle, including team standards, release labelling, and more.

The maintenance of this document will be the responsibility of the development team. Changes to the CI/CD process, either from the development team or the broader McMaster University infrastructure team should be reflected before the next release.

23.2 Training Requirements

- 1. Users should be able to use the tool and utilize key features immediately after following the tutorial**
- 2. Users should be able to find key features without consulting additional documentation 95% of the time.**

The responsibility of the training will first fall towards the development team of the tool. They must ensure the provided in-tool tutorial is up-to-date and sufficient to help a user understand all key features.

Additional training will be provided to the supervisor, hosted by the development team. Subsequent training will be the responsibility of the supervisor,

if needed to train future users of the tool, in the format that the supervisor chooses.

24 Waiting Room

1. Comparing across Domains

The user should be able to compare two (or more) completed domain analysis against each other.

2. Versioning of Domains

Users can revisit and update completed domains, adding new data or editing existing ones. Allowing users to view the evolution of the state of best practice for the domain.

25 Ideas for Solution

The following is the development team's idea for the user interface of the tool. Figure 1 shows the initial concept, drawing inspiration from [Octave Online](#), the cloud IDE for Matlab.

The main section of the tool will be where the data is displayed and gathered for each domain. With the sections that can be automatically gathered differentiated using a different colour, such as the gray shown in Figure 1.

The left sidebar will contain all the domains, with indications on whether it's completed or not. As well as providing a filter to quickly search for a specific domain.

Product		Description	Adaptation Potential
Software Assessment Dashboards		Tools that analyze open-source metrics.	Their structure can guide data collection and dashboard design.
University Research Repositories		Academic dashboards for research analytics.	Useful for UI and data categorization strategies.

Figure 1: DomainX UI, inspired by Octave Online

Appendix — Reflection

The purpose of reflection questions is to give you a chance to assess your own learning and that of your group as a whole, and to find ways to improve in the future. Reflection is an important part of the learning process. Reflection is also an essential component of a successful software development process.

Reflections are most interesting and useful when they're honest, even if the stories they tell are imperfect. You will be marked based on your depth of thought and analysis, and not based on the content of the reflections themselves. Thus, for full marks we encourage you to answer openly and honestly and to avoid simply writing "what you think the evaluator wants to hear."

Please answer the following questions. Some questions can be answered on the team level, but where appropriate, each team member should write their own response:

1. What went well while writing this deliverable?
2. What pain points did you experience during this deliverable, and how did you resolve them?
3. How many of your requirements were inspired by speaking to your client(s) or their proxies (e.g. your peers, stakeholders, potential users)?
4. Which of the courses you have taken, or are currently taking, will help your team to be successful with your capstone project.
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.
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?

Table 6: Project Planning Phases

Phase	Description	Key Activities
Initiation	Define problem, scope, and objectives	Document review, team formation, feasibility assessment
Requirements	Gather and formalize system requirements	Stakeholder interviews, drafting of SRS and Hazard Analysis
Design	Establish system architecture and interfaces	UML mockups, data flow diagrams, schema design
Implementation	Build core tool functionality	Develop modules, integrate APIs, implement Excel import/export
Validation & Verification	Test Unit and evaluate	Unit and integration testing, review sessions, issue resolution

Table 7: Development Phase Plan

Phase Name	Benefit to User	Operational Date	Open Components	Functional Requirements	Notes
Requirements & Analysis	Clarify system objectives and constraints	Week 1–6	Git, LaTeX	Requirements documentation	Needs Functional Requirements
Design	Define architecture and interfaces	Week 10–16	UML, wire-framing software	UI and backend design	Must be
Implementation	Deliver core product functionality	Week 16–19	IDE, databases, APIs	Tool modules, automation scripts	Requires usability it,
Testing & Validation	Ensure product meets all criteria	Week 17–22	Test suites, CI/CD	Verification of features	Part of
Deployment	Provide accessible	Week 22–26	Hosted platform, docu-	Hosted application	Should be accessible, simple

Name	Configuration	Estimated Usage	Total Cost
RDS for MySQL	Two vCPU and 8GB of Memory	20 Hours/Month	9.76 CAD/Month
EC2	t4g.large Instance	20 Hours/Month	1.87 USD/Month

Table 8: Price estimates and total costs for two AWS services.