

Software Requirements Specification

MTOBridge

Team 15, Alpha Software Solutions

Badawy, Adham

Yazdinia, Pedram

Jandric, David

Vakili, Farzad

Vezina, Victor

Chiu, Darren

Contents

1	Project Drivers	1
1.1	The Purpose of the Project	1
1.2	The Stakeholders	1
1.2.1	The Client	1
1.2.2	The Customers	1
1.2.3	Other Stakeholders	2
1.2.4	User Characteristics	2
1.3	Mandated Constraints	2
1.4	Naming Conventions and Terminology	3
1.5	Relevant Facts and Assumptions	3
2	Functional Requirements	4
2.1	Behavioral Overview	4
2.2	The Scope of the Work	5
2.2.1	The Context of the Work	5
2.2.2	Work Partitioning	6
2.3	The Scope of the Product	7
2.3.1	Product Boundary	7
2.3.2	Variables	8
2.3.3	Individual Product Use Cases	8
2.4	Functional Requirements/Phase in Plan	13
3	Non-functional Requirements	20
3.1	Look and Feel Requirements	20
3.2	Usability and Humanity Requirements	20
3.3	Performance Requirements	22
3.4	Operational and Environmental Requirements	23
3.5	Maintainability and Support Requirements	23
3.6	Security Requirements	24
3.7	Cultural Requirements	24
3.8	Legal Requirements	24
3.9	Health and Safety Requirements	24
4	Project Issues	24
4.1	Open Issues	24
4.2	Off-the-Shelf Solutions	24

4.3	New Problems	24
4.4	Tasks	24
4.5	Migration to the New Product	25
4.6	Risks	25
4.7	Costs	25
4.8	User Documentation and Training	25
4.9	Waiting Room	25
4.10	Ideas for Solutions	26
5	Appendix	28
5.1	Likely Changes	28
5.2	Traceability Matrix	31
5.3	Reflection	31

Table 1: Revision History

Date	Developer(s)	Change
October 4	David	Add context/partitioning of work
October 4	Adham	First Draft of FRs
October 4	Darren	Added some Non-functional Requirements
October 4	Victor	Project Issues
October 5	Victor	Added some Non-functional Requirements
October 5	Darren	Refined Non-functional Requirements
October 5	Pedram	Added Individual Product Usecase
October 5	Adham	Finished FRs
October 5	Adham	Added Likely Changes Table
October 5	Farzad	Completed parts of project drivers
October 5	Adham	Added Prio. and Phase in Plan for FRs and Reflection Skeleton
October 5	Adham	Added Behaviour Overview, Refined Requirements, and added Reflection
October 5	Darren	Numbered and traced Non-functional Requirements
October 5	Farzad	Completed project drivers and reflection

List of Tables

1	Revision History	iii
2	Business Event List	6
3	Variable List	8
4	Risks	25
5	Traceability Matrix	31

List of Figures

1	Context Diagram of MTOBridge	5
2	Use Case Diagram of MTOBridge	7
3	Use Case 1 State Machine	9
4	Use Case 2 State Machine	10
5	Use Case 3 State Machine	11
6	Use Case 4 State Machine	12
7	Use Case 5 State Machine	12

This document describes the requirements for MTOBridge. The template for the Software Requirements Specification (SRS) is a subset of the Volere template [Robertson and Robertson \(2012\)](#). In addition, we have added a few sections not in the Volere template: Appendix (and its subsections), Phase in Plan, Behavior Overview, Variables.

1 Project Drivers

1.1 The Purpose of the Project

For years, Bridge Engineers in Ontario have based their bridge analysis on the Canadian Highway Bridge Design Code (CHBDC)(CSA S6-19) which typically features conservatism and adds excessive costs. This project aims to develop an application ‘MTOBridge’ that leverages prepared MATLAB engines (containing the logic for novel analysis methods) and better visualizes their outputs while enhancing usability. This enables the refined methods of analysis to become routine for bridge design and evaluation. Moreover, A unique feature of ‘MTOBridge’, which is otherwise unavailable in other bridge design/evaluation software, is that a direct comparison can be obtained between the results obtained from the CHBDC approximate/simplified approach and those from the refined analysis. This endeavor will facilitate confidence in using refined analysis for bridge evaluation and design.

1.2 The Stakeholders

1.2.1 The Client

Ontario Ministry of Transport (MTO) is the sponsor of this project and is making the investment to bring this program to fruition in partnership with the McMaster University Civil Engineering Department.

1.2.2 The Customers

The MTO and civil engineering firms in the bridge design industry are the customers for this software. They will decide to adopt this new tool depending on whether it will help them perform their work and how easy it is for their engineers to use the tool.

1.2.3 Other Stakeholders

- Department of Civil Engineering, McMaster
 - The proposed program will be directly developed in collaboration with the Department of Civil Engineering at McMaster University.
- Department of Computing and Software, McMaster
 - The proposed program will be directly developed by engineering students from the Department of Computing and Software at McMaster University.

1.2.4 User Characteristics

The potential users of this product are civil engineers designing and evaluating bridges. The age group is from 22 to 65 years old, with a basic understanding of how to interact with simple software user interfaces after reading a short user guide. The majority of the users are likely to work on site using workstations, however there may be users that work from home.

1.3 Mandated Constraints

C.1: The system shall be able to perform calculations using the existing and future versions of the prepared MATLAB code.

Rationale: Dr. Yang and her team, which are one of the main stakeholders, are not going to implement their code in another language.

Fit Criterion: The system shall be able to call all functions in the MATLAB engine correctly according to the analysis configuration.

C.2: The product shall operate using Windows 10.

Rationale: The client and majority of customers are using Windows 10.

Fit Criterion: The product shall run correctly on Windows 10 machines.

C.3: The product shouldn't require a MATLAB license to run the program.

Rationale: The majority of users do not have a MATLAB license and are not willing to purchase one.

Fit Criterion: The product shall be able to run the program without requiring a MATLAB license.

1.4 Naming Conventions and Terminology

Abutment: A substructure composed of stone, concrete, brick or timber supporting the ends of a bridge. It also, usually, supports the approach embankment.

Dead load: The weight of the permanent, immovable parts of a structure, such as the towers, cables, and roadway of a bridge.

Girder: A horizontal main structural member (as in a building or bridge) that supports vertical loads and consists of a single piece or of more than one piece bound together.

Live load: Refers to traffic that moves across a bridge as well as normal environmental factors such as changes in temperature, precipitation, and winds.

Span: The gap between two supports.

Truck Platoon: A close-knit convoy of self driving trucks that can travel faster and closer together than by traditional means. Expected to become common in the future.

Headway: A term for the space between the last axle of one truck in a platoon and the next axle of the subsequent truck. Used to measure spacing between trucks.

Concerned Section: A term used to define the point along a bridge for which the user is interested in viewing the loads placed on it by a platoon.

Critical Section: A term used to mean the point along a bridge that experiences the highest loads during a truck platoon's trip along a bridge.

Discretization Length: This length determines how many sections the user wants to split a bridge into for the critical section calculation. For example, a 10m long bridge with a discretization length of 1m would be split into 10 sections 1m apart. These 10 would then be compared over the course of the trip to find the critical section.

1.5 Relevant Facts and Assumptions

A.1: The existing MATLAB code is reliable and thoroughly tested.

A.2: Users have access to computers that are powerful enough to run analysis.

2 Functional Requirements

2.1 Behavioral Overview

The following few paragraphs comprise a verbal summary of the whole of section 2, for quick reference.

The program to be developed is a graphical user interface between the user and a duo of MATLAB solvers created by the Department of Civil Engineering at McMaster for load rating bridges as a truck platoon drives over them. It will allow the user to interface with the calculations in four major ways. Namely, these are:

1. Determining the specifics of a truck platoon that will be rolling over a bridge, in terms of the axle loads and spacings of the truck, the number of trucks in the platoon and the headway between the trucks.
2. Determining the specifics of a bridge the truck platoon will be rolling over, in terms of whether it is a one, two, or three span bridge and the length of each span.
3. Determining the information they require about the demands placed on the bridge by the truck platoon. Namely, the type of force, and whether they care about what happens at a particular section of the bridge, or the section with the maximum force endured by the bridge.
4. Visualizing the results of the calculations in a comprehensive and informative fashion.

There will be other features to the program as well, such as the ability to generate a text based report of the analysis outputs and allowing the user to save and load their truck and bridge configurations for ease-of-use. This is a

living document, and more functionality may be added over time.

2.2 The Scope of the Work

2.2.1 The Context of the Work

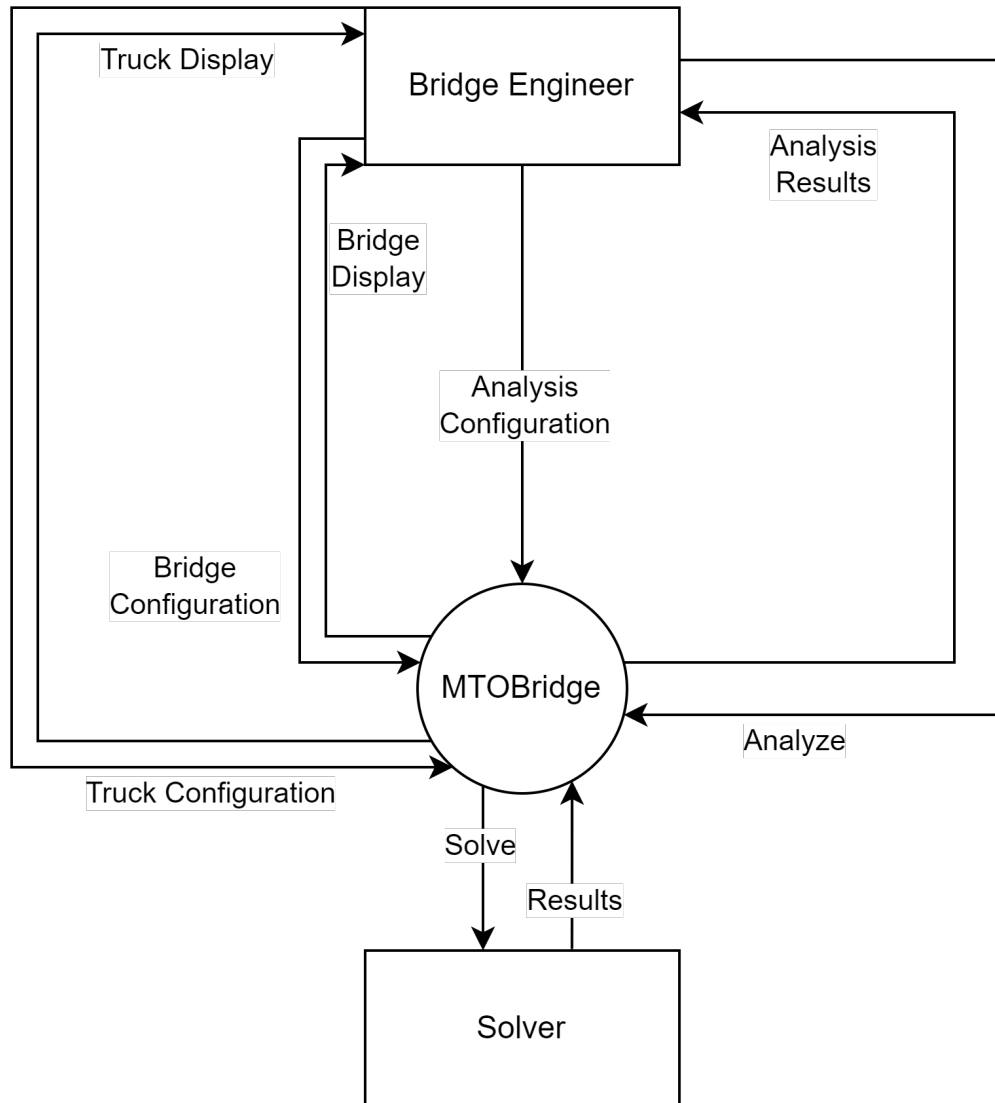


Figure 1: Context Diagram of MTOBridge

2.2.2 Work Partitioning

Table 2: Business Event List

Event Name	Input/Output	Summary
Engineer enters truck configuration	IN: Truck configuration, OUT: Truck display	Record truck configuration, show truck visualization
Engineer enters bridge configuration	IN: Bridge configuration, OUT: Bridge display	Record bridge configuration, show bridge visualization
Engineer enters analysis configuration	IN: Analysis configuration	Record analysis configuration
Engineer requests analysis	IN: Analyze request, OUT: Analysis results	Display analysis results
Time to analyze bridge	OUT: Solve request, IN: Solver results	Give configurations to solver to get results

2.3 The Scope of the Product

2.3.1 Product Boundary

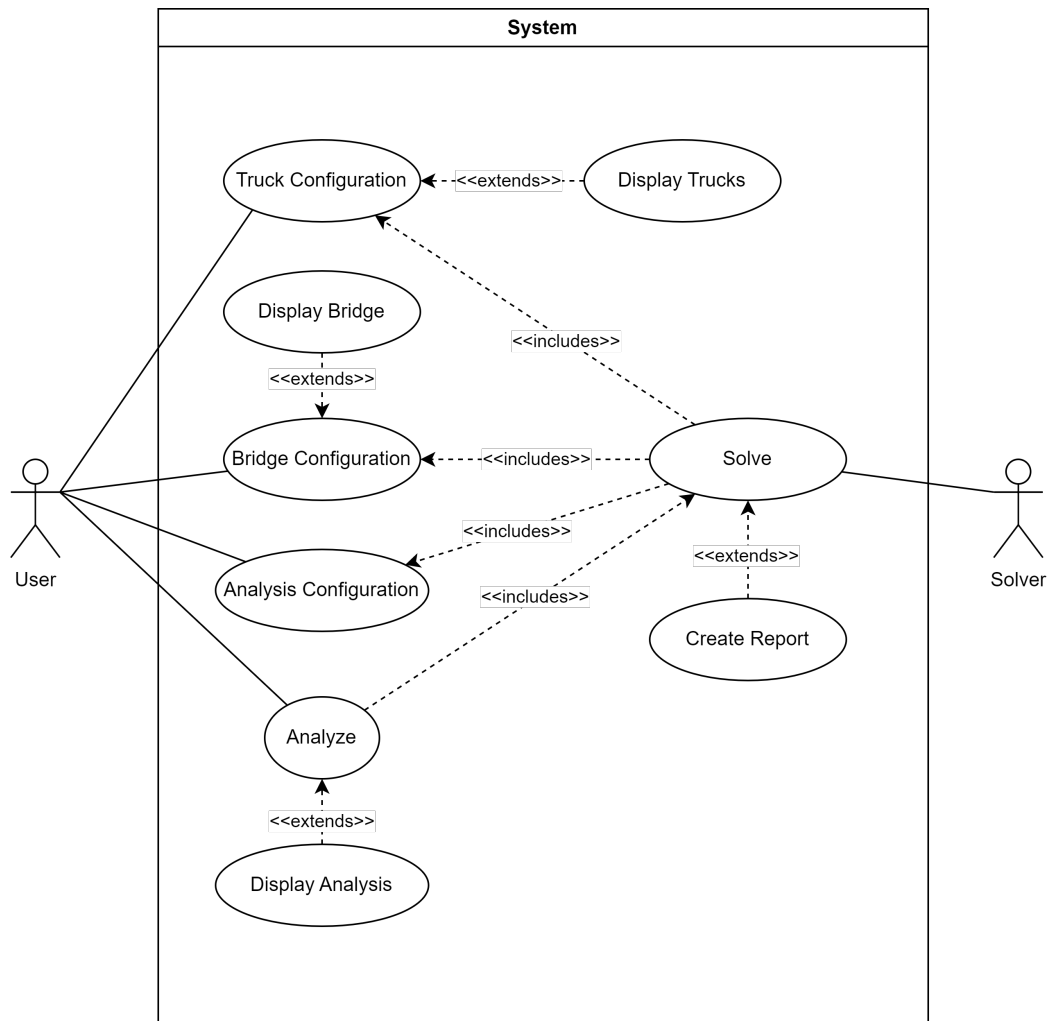


Figure 2: Use Case Diagram of MTOBridge

2.3.2 Variables

Table 3: Variable List

Name	Type	Description
analyze	bool	User has requested analysis?
displayTrucks	bool	Display truck visualization?
displayBridge	bool	Display bridge visualization?
displayResults	bool	Display analysis results?
okPressed	bool	Represents whether or not invalid message was accepted
validateTruck	bool	Is validation done on truck configuration?
validateBridge	bool	Is validation done on bridge configuration?
validateAnalysisConfig	bool	Is validation done on analysis configuration?
writeReport	bool	Write results to a file?
input	string	Text representing user input
config	Config	Abstract representation of configuration, with definable fields and options
analysisConfig	Config	Configuration of analysis
bridgeConfig	Config	Configuration of a bridge
truckConfig	Config	Configuration of truck platoon, contains many fields
results	SolverResults	Abstract representation of output from Solver

2.3.3 Individual Product Use Cases

User navigates to the Truck Configuration Tab

Primary Actors: Engineers

Pre Condition: None

Post Condition: Input values from the user conforms to the valid range required for the calculations.

Basic Flow:

- System asks user to enter a truck parameter.
- User enters truck parameter.
- System checks parameter.

Alternative If the parameter is incorrect, the user is alerted.

Extension:

- User requests an animation of the truck platoon.
- System provides an animation of the truck platoon.
- User interacts with the animation of the trucks.

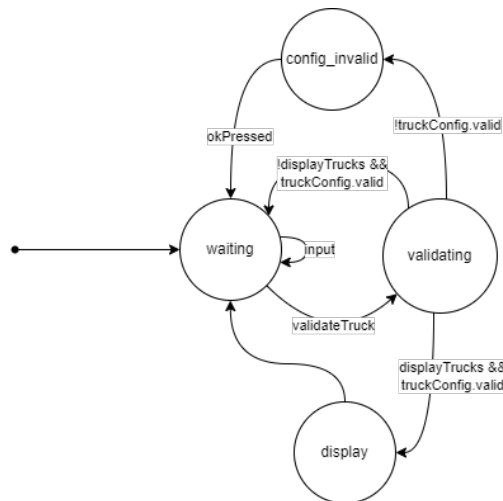


Figure 3: Use Case 1 State Machine

User navigates to the Bridge Configuration Tab

Primary Actors: Engineers

Pre Condition: None

Post Condition: Input values from the user conforms to the valid range

required for the calculations.

Basic Flow:

- System asks the user to enter a bridge parameter.
- User enters the bridge parameters.
- System checks parameter.

Alternative If the parameter is incorrect, the user is alerted.

Extension:

- User requests an animation of the bridge.
- System provides an animation of the bridge.
- User interacts with the animation of the bridge.

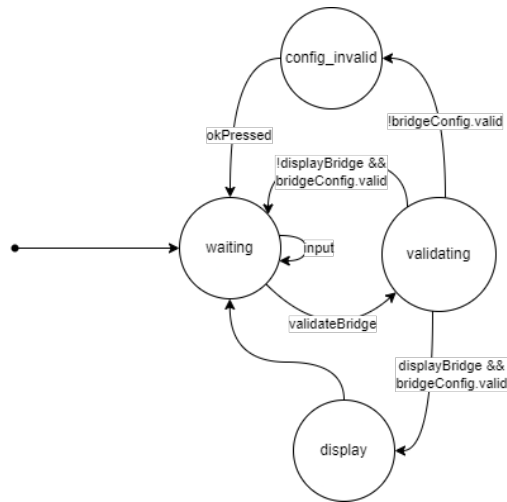


Figure 4: Use Case 2 State Machine

User navigates to the Analysis Configuration Tab

Primary Actors: Engineers

Pre Condition: None

Post Condition: Input values from the user conforms to the valid range required for the calculations.

Basic Flow:

- System asks the user to enter an analysis parameter.
- User enters the analysis parameters.
- System checks parameter.

Alternative If the parameter is incorrect, the user is alerted.

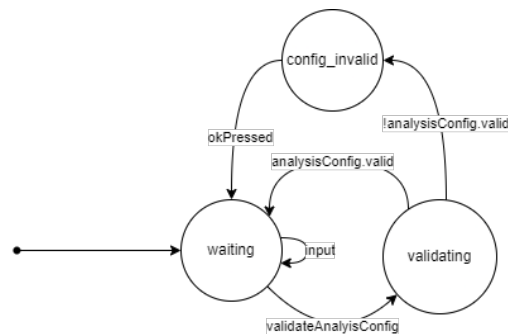


Figure 5: Use Case 3 State Machine

User starts the Analyze process.

Primary Actors: Engineers

Pre Condition: None

Post Condition: None

Basic Flow:

- User selects the Analyze tab.
- The system validates the configurations.

Alternative If the configurations are not correct, the user is alerted.

- Configurations are sent to Solver.
- Text-based result from Solver is returned.

Extension:

- Visualization of results is displayed.

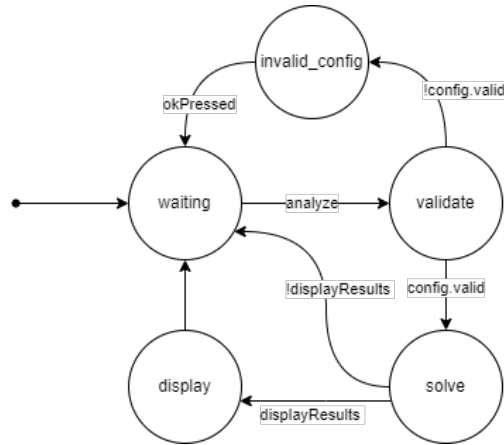


Figure 6: Use Case 4 State Machine

The solver solves the mathematical model using the given inputs from the system.

Primary Actors: Solver

Pre Condition: Truck, Bridge and Analysis Configuration are all valid.

Post Condition: Received results valid based on the testing spec.

Basic Flow:

- The system receives the configuration data.
- The system sends the data to the Solver where the equations are solved using the given parameters.
- Solver returns text-based results.

Alternative The solver cannot solve the equations using the given input. The user is alerted to check their parameters.

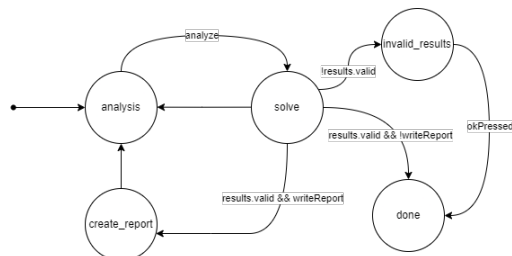


Figure 7: Use Case 5 State Machine

2.4 Functional Requirements/Phase in Plan

FR.1 The program shall have ability to run the existing and future versions of the prepared MATLAB solver.

Rationale: To perform any analysis, the UI needs to gather outputs from via the backend MATLAB code.

Fit Criterion: The program can call the MATLAB functions, and the outputs shall match their the expected output.

Priority/Phase in Date: High/Nov 7th

Phase in Date Rationale: This will be the first functionality implemented as it is necessary for proof of concept.

Traceability: C.1, A.1.

FR.2 The program shall allow the user to define the characteristics of the truck platoon, including axle loads, spacings, the number of trucks, and headway.

Rationale: The goal of the program is to determine the forces exerted by a given platoon on a bridge, so flexibility in the characteristics of the platoon are necessary for the relevance of the simulation.

Fit Criterion: There exists a method of input (text based, dropdown list, etc.) that allows users to correctly specify those 4 characteristics of the platoon.

Priority/Phase in Date: Medium/Dec 20th

Phase in Date Rationale: This will be among the later functionalities as it is not important for the proof of concept.

Traceability: N/A: No constraints or assumptions trace to this requirement.

FR.3 The program shall allow the user to visualize a truck platoon based on the defined characteristics.

Rationale: This is to assist the user in verifying the correctness of their inputs into the program. As we are making a GUI, visual feedback is paramount to the functionality of the program.

Fit Criterion: There exists some visual representation of the truck platoon that changes to correctly reflect the impact of changes in user input.

Priority/Phase in Date: Medium/Jan 5th

Phase in Date Rationale: This will be among the last functionalities implemented as it is dependent on FR.2.

Traceability: N/A: No constraints or assumptions trace to this requirement.

FR.4 The program shall allow the user to save their current truck platoon configuration for later use.

Rationale: Users may be interested in analyzing the demands a particular truck platoon places on bridges in multiple situations, and over multiple sessions. This reduces the repetitive remaking of specific configurations to facilitate efficient use of the program.

Fit Criterion: The program is capable of storing a truck platoon configuration in some form to be retrieved later.

Priority/Phase in Date: Low/Jan 25th

Phase in Date Rationale: This will be among the last functionalities implemented as it is among the lowest priority requirements.

Traceability: N/A: No constraints or assumptions trace to this requirement.

FR.5 The program shall allow the user to load their previously saved truck platoon configurations.

Rationale: Accompanies FR.4, the saved configurations should have the ability to be loaded to facilitate efficient use of the program.

Fit Criterion: The program shall correctly recreate a truck platoon configuration from a previously saved configuration.

Priority/Phase in Date: Low/Jan 30th

Phase in Date Rationale: This will be among the last functionalities implemented as it is among the lowest priority requirements.

Traceability: N/A: No constraints or assumptions trace to this requirement.

FR.6 The program shall allow the user to define the characteristics of the bridge, including the number of spans and the length of each span.

Rationale: Different bridges will react to the same truck platoon differently, therefore specifying the relevant characteristics of the bridge is necessary for

the relevance of the simulation.

Fit Criterion There exists a method of input (text based, dropdown list, etc.) that allows users to correctly specify those 2 characteristics of the bridge.

Priority/Phase in Date: Medium/Dec 20th

Phase in Date Rationale: This will be among the later functionalities as it is not important for the proof of concept.

Traceability: N/A: No constraints or assumptions trace to this requirement.

FR.7 The program shall allow the user to visualize a bridge based on the defined characteristics.

Rationale: This is to assist the user in verifying the correctness of their inputs into the program. As we are making a GUI, visual feedback is paramount to the functionality of the program.

Fit Criterion: There exists some visual representation of the bridge that changes to correctly reflect the impact of changes in user input.

Priority/Phase in Date: Medium/Jan 5th

Phase in Date Rationale: This will be among the last functionalities implemented as it is dependent on FR.6.

Traceability: N/A: No constraints or assumptions trace to this requirement.

FR.8 The program shall allow the user to save their current bridge configuration for later use.

Rationale: Users may be interested in analyzing the demands on a particular bridge in multiple situations, and over multiple sessions. This reduces the repetitive remaking of specific configurations to facilitate efficient use of the program.

Fit Criterion: The program is capable of storing a bridge configuration in some form to be retrieved later.

Priority/Phase in Date: Low/Jan 25th

Phase in Date Rationale: This will be among the last functionalities implemented as it is among the lowest priority requirements.

Traceability: N/A: No constraints or assumptions trace to this requirement.

FR.9 The program shall allow the user to load in their previously saved bridge configurations.

Rationale: Accompanies FR.8, the saved configurations should have the ability to be loaded to facilitate efficient use of the program.

Fit Criterion: The program shall correctly recreate a bridge configuration from a previously saved configuration.

Priority/Phase in Date: Low/Jan 30th

Phase in Date Rationale: This will be among the last functionalities implemented as it is among the lowest priority requirements.

Traceability: N/A: No constraints or assumptions trace to this requirement.

FR.10 The program shall allow the user to define which of the two solvers they are interested in using.

Rationale: The MATLAB backend can solve for the demand on a concerned section, as well as the critical section, which are two important analyses that a user may need to perform.

Fit Criterion: At least one method of input (text based, dropdown list, etc) exists that allows the user to accurately choose which solver they are would like to use.

Priority/Phase in Date: High/Nov 20th

Phase in Date Rationale: This functionality will be implemented relatively early as it is an important part of the core program.

Traceability: N/A: No constraints or assumptions trace to this requirement.

FR.11 The program shall allow the user to define a concerned section.

Rationale: The first solver outputs the results at a specific section, which requires the concerned section to be defined.

Fit Criterion: There exists a method of input (text based, dropdown list, etc) that allows the user to accurately specify a concerned section.

Priority/Phase in Date: High/Nov 25th

Phase in Date Rationale: This functionality will be implemented relatively early as it is an important part of the core program.

Traceability: N/A: No constraints or assumptions trace to this requirement.

FR.12 The program shall allow the user to define a discretization length.

Rationale: The second solver finds the critical section, which requires the discretization length for analysis.

Fit Criterion: There exists a method of input (text based, dropdown list, etc) that allows the user to accurately specify a discretization length.

Priority/Phase in Date: High/Nov 30th

Phase in Date Rationale: This functionality will be implemented relatively early as it is an important part of the core program.

Traceability: N/A: No constraints or assumptions trace to this requirement.

FR.13 The program shall allow the user to define the type of force on the bridge, i.e. positive moment, negative moment, or shear.

Rationale: There are a variety of different forces placed on a bridge as a platoon drives over, all of which have an important place in the analysis. Allowing the user to define which of the 3 they are interested in seeing is necessary for the functionality of the simulation.

Fit Criterion: There exists a method of input (text based, dropdown list, etc) that allows the user to accurately specify the desired force for analysis.

Priority/Phase in Date: Medium/Dec 15th

Phase in Date Rationale: This functionality will be implemented somewhere in the middle as it is an important part of the program.

Traceability: N/A: No constraints or assumptions trace to this requirement.

FR.14 The program shall be capable of visualizing the results of the concerned section calculation.

Rationale: This is the main purpose of the GUI, i.e. displaying the results of the MATLAB backend calculations visually to the user. This is one of the two main calculations to be represented, so this functionality is necessary.

Fit Criterion: There exists an accurate visualization of the mathematical

results of the concerned section calculation.

Priority/Phase in Date: Extremely High/Dec 10th

Phase in Date Rationale: While this functionality is the core feature of the program, its proper implementation depends on many other FRs, so it will be fully implemented on the later side.

Traceability: N/A: No constraints or assumptions trace to this requirement.

FR.15 The program shall be capable of visualizing the results of the critical section calculation.

Rationale: This is essentially the main purpose of the GUI, i.e. displaying the results of the MATLAB backend calculations visually to the user. This is one of the two main calculations to be represented, so this functionality is very necessary.

Fit Criterion: There exists an accurate visualization of the mathematical results of the critical section calculation.

Priority/Phase in Date: Extremely/Dec 10th

Phase in Date Rationale: While this functionality is the core feature of the program, its proper implementation depends on many other FRs, so it will be fully implemented on the later side.

Traceability: N/A: No constraints or assumptions trace to this requirement.

FR.16 The program shall be capable of outputting a report summarizing the resulting analysis.

Rationale: Having a representation of the data presented in the program after the current session of use is over is helpful for engineers comparing different analyses.

Fit Criterion: The program outputs an accurate report that contains the configurations and outputs from an analysis.

Priority/Phase in Date: Medium/Jan 15th

Phase in Date Rationale: This will be among the last functionalities implemented as it is dependent on many other FRs.

Traceability: N/A: No constraints or assumptions trace to this requirement.

FR.17: The system must be able to synchronize the display of the truck platoon with the bridge load. A correction will be made if the two parts are out of sync, or an error will be displayed if necessary.

Rationale:: While splitting the calculation display modules into two modules, truck platoon and bridge load display, can simplify each part, it can lead to mismatch in each respective display.

Fit Criterion:: The program keeps the position of the truck platoon and the bridge load in sync for all frames of the display.

Priority/Phase in Date: Medium/Jan 15th

Phase in Date Rationale: This will be among the last functionalities implemented as it is dependent on many other FRs.

Traceability:: SR.1: This traces to the [Hazard Analysis](#) which was written after this document.

FR.18: The system must have thread safety between the UI and other connected components.

Rationale: In order to avoid race conditions and deadlocks that could result in undesirable behavior, thread safety must be an integral part of the interaction between components.

Fit Criterion:: The program is analyzed with static analysis and tested for concurrency issues, with none found.

Priority/Phase in Date: Medium/Jan 15th

Phase in Date Rationale: This will be among the last functionalities implemented as it is dependent on many other FRs.

Trace: SR.2: This traces to the [Hazard Analysis](#) which was written after this document.

FR.19: The system must validate numeric values are within an acceptable range before being passed on to other components.

Rationale: Accidental changes of input can cause massive shifts in the analysis and cause inaccurate results (e.g. adding extra 0's).

Fit Criterion:: The program does not allow numbers outside of an appropriate range to be inputted where numeric inputs are expected.

Priority/Phase in Date: Medium/Jan 15th

Phase in Date Rationale: This will be among the last functionalities im-

plemented as it is dependent on many other FRs.

Trace: SR.5: This traces to the [Hazard Analysis](#) which was written after this document.

FR.20: The system must validate input types before being passed on to other components.

Rationale: This is to prevent accidental mix and match of inputs, e.g. a string where a number is expected.

Fit Criterion:: The program does not accept inappropriate inputs in the program.

Priority/Phase in Date: Medium/Jan 15th

Phase in Date Rationale: This will be among the last functionalities implemented as it is dependent on many other FRs.

Trace: SR.6: This traces to the [Hazard Analysis](#) which was written after this document.

3 Non-functional Requirements

3.1 Look and Feel Requirements

NFR.1 The graphics will be informative.

Rationale: The user shall gain value out of the presence of graphics.

Fit Criterion: Civil engineers who use the program will identify and understand the different graphic elements used to represent bridge parts in at least 90% of cases.

Traceability: FR.3, FR.7, FR.14, FR.15.

3.2 Usability and Humanity Requirements

NFR.2 The program will be intuitive to use.

Rationale: The program shall be easy to use for its intended audience.

Fit Criterion: 90% of civil engineers can use the program to generate a

bridge system analysis within 5 minutes of introduction.

Traceability: N/A; this is an important quality of the program as a whole and so does not map to any individual FR, constraint, or assumption.

NFR.3 The product will appear correctly on different display resolutions.

Rationale: Users of the product may wish to use it on displays of different resolutions.

Fit Criterion: The program will be viewed and its appearance validated on displays of different resolutions. The display resolutions tested will include 1920x1080, 2560x1440, 3840x2160, 1280x720, and 1366x768.

Traceability: C.2.

NFR.4 The product will allow for the resizing of text.

Rationale: Users of the product may wish to increase text size to allow for easier reading of the text.

Fit Criterion: We will ensure that the text size within the program is resizable within a range of at least 8pt to 32pt font, and that the program still functions correctly when the text size is changed.

Traceability: FR.3, FR.7, FR.14, FR.15.

NFR.5 The program will be easy to install and run.

Rationale: The program cannot be difficult to install for the users of our product.

Fit Criterion: The time it takes to download, install, and run our program will not exceed 30 minutes.

Traceability: C.2, C.3.

NFR.6 UI elements which provide similar functionality will have a similar look.

Rationale: Having UI elements with comparable functionality be visually consistent will help with usability.

Fit Criterion: We will classify UI elements into categories and ensure that all elements within each categories are visually similar. Civil engineers will be able to associate UI elements with their corresponding categories in at

least 90% of cases.

Traceability: FR.3, FR.7, FR.14, FR.15.

3.3 Performance Requirements

NFR.7 The program will safely handle unusual user inputs.

Rationale: Program shall be robust and not prone to failure due to common invalid inputs.

Fit Criterion: The program will not freeze or crash as a direct result of a user providing inputs to the system.

Traceability: FR.2, FR.6, FR.11, FR.12, FR.13.

NFR.8 The program will be able to handle missing dependencies.

Rationale: The program shall be able to handle and warn of absent files.

Fit Criterion: The program will produce an error message when MATLAB scripts are absent or unable to run.

Traceability: FR.1.

NFR.9 UI elements will react promptly to user input.

Rationale: The users of the program will want the UI to react quickly to their input.

Fit Criterion: The UI will graphically update to indicate it has acknowledged user inputs within 100ms of the input.

Traceability: FR.3, FR.7, FR.14, FR.15.

NFR.10 UI will not be unreasonably slow.

Rationale: UI shall not introduce substantial delay beyond what is needed to calculate results.

Fit Criterion: The program delay when calculating and displaying results will not exceed the underlying MATLAB script's execution time by 10%.

Traceability: FR.3, FR.7, FR.14, FR.15.

NFR.11 The program will be precise.

Rationale: The program must be reasonably precise to provide value for simulating bridges under load.

Fit Criterion: Calculations are accurate to within 1% relative error of similar bridge simulation engines.

Traceability: C.1, FR.1.

3.4 Operational and Environmental Requirements

NFR.12 The program will run without slowdown on expected users' (MTO engineers) computers.

Rationale: The program must be able to run within requirements on the computers that it is intended to be used on.

Fit Criterion: Performance testing will be done on a computer with the same (or reasonably similar) hardware.

Traceability: C.1.

3.5 Maintainability and Support Requirements

NFR.13 The product shall be easily maintainable.

Rationale: The code must be easily maintainable to allow for future bug fixes and/or feature additions.

Fit Criterion: We will use file length, method length, and nesting depth as our primary indicators of code maintainability. 75% of our files will have a length of less than 750 lines, 75% of our methods will have a length of less than 75 lines, and 75% of our methods will have a maximum nesting depth of less than 5.

Traceability: N/A; this is an important quality of the program as a whole and so does not map to any individual FR, constraint, or assumption.

3.6 Security Requirements

N/A; this project does not involve significant communication of user data.

3.7 Cultural Requirements

N/A; this project is not complex enough to require translation.

3.8 Legal Requirements

N/A; this project is not

3.9 Health and Safety Requirements

N/A; this project is only concerned with the graphical representation of the underlying bridge calculations.

4 Project Issues

4.1 Open Issues

N/A; there are currently no open issues for our project.

4.2 Off-the-Shelf Solutions

N/A; since we are using novel calculations designed specifically for this project, there are no comparable off-the-shelf solutions to our project.

4.3 New Problems

N/A; there are currently no new problems for our project

4.4 Tasks

N/A; there are currently no required tasks for our project.

4.5 Migration to the New Product

N/A; this product is novel and would not be replacing an existing product.

4.6 Risks

The major risks associated with this project are:

Table 4: Risks

Risk	Probability	Impact
Not being able to integrate our program with the MATLAB script provided to us.	Very low	Very high
Not being able to have a responsive UI while calling the MATLAB script.	Low	High
Not being able to create a UI that allows for structure sketching.	Moderate	Moderate
The underlying MATLAB script changes.	Moderate	Moderate

4.7 Costs

No monetary costs are involved in creating this product. The only costs in developing the product are the project team's time, and Dr. Yang's and her graduate students' time.

4.8 User Documentation and Training

An extensive user manual with case study examples will be produced alongside the product. This documentation will allow for bridge engineers to become familiar and comfortable using the software. We will also train Dr. Cancan Yang and their graduate students to use the product so that they can effectively present the product to the MTO.

4.9 Waiting Room

N/A; we don't currently have any requirements in the project waiting room.

4.10 Ideas for Solutions

N/A; we have not currently made any decisions about solution ideas.

References

James Robertson and Suzanne Robertson. *Volere Requirements Specification Template*. Atlantic Systems Guild Limited, 16 edition, 2012.

5 Appendix

This section contains additional information related to the requirements laid out previously in the document, as well as a reflection.

5.1 Likely Changes

FR.1

Likelihood Of Change: Very Unlikely

Rationale: This is a core feature.

FR.2

Likelihood Of Change: Unlikely

Rationale: Testing different truck platoons is very important to the simulation's usefulness

FR.3

Likelihood Of Change: Unlikely

Rationale: Visual Feedback is a core part of the usefulness of the UI

FR.4

Likelihood Of Change: Possible

Rationale: This is a 'nice to have' feature.

FR.5

Likelihood Of Change: Possible

Rationale: Same Likelihood as FR.4, as if that changes, this changes

FR.6

Likelihood Of Change: Unlikely

Rationale: Testing different types of bridges is very important to the simulation's usefulness

FR.7

Likelihood Of Change: Unlikely

Rationale: Visual Feedback is a core part of the usefulness of the UI

FR.8

Likelihood Of Change: Possible

Rationale: This is a 'nice to have' feature.

FR.9

Likelihood Of Change: Possible

Rationale: Same Likelihood as FR.8, as if that changes, this changes

FR.10

Likelihood Of Change: Very Unlikely

Rationale: This is a core feature.

FR.11

Likelihood Of Change: Very Unlikely

Rationale: This is a core feature.

FR.12

Likelihood Of Change: Very Unlikely

Rationale: This is a core feature.

FR.13

Likelihood Of Change: Unlikely

Rationale: Bridge Engineers are very interested in seeing multiple different types of demands on their bridge, not just one

FR.14

Likelihood Of Change: Extremely Unlikely

Rationale: This is the main purpose of the program.

FR.15

Likelihood Of Change: Extremely Unlikely

Rationale: This is the main purpose of the program.

FR.16

Likelihood Of Change: Unlikely

Rationale: A report is important to preserve the simulation's findings

FR.17

Likelihood Of Change: Extremely Unlikely

Rationale: Safety requirement, would only change if the entire project was different

FR.18

Likelihood Of Change: Extremely Unlikely

Rationale: Safety requirement, would only change if the entire project was different

FR.19

Likelihood Of Change: Extremely Unlikely

Rationale: Safety requirement, would only change if the entire project was different

FR.20

Likelihood Of Change: Extremely Unlikely

Rationale: Safety requirement, would only change if the entire project was different

5.2 Traceability Matrix

Table 5: Traceability Matrix

NFR	Traceable to
NFR.1	FR.3, FR.7, FR.14, and FR.15
NFR.2	N/A; this is an important quality of the program as a whole and so does not map to any individual FR, constraint, or assumption.
NFR.3	C.2
NFR.4	FR.3, FR.7, FR.14, and FR.15
NFR.5	C.2 and C.3
NFR.6	FR.3, FR.7, FR.14, and FR.15
NFR.7	FR.2, FR.6, FR.11, FR.12, and FR.13
NFR.8	FR.1
NFR.9	FR.3, FR.7, FR.14, and FR.15
NFR.10	FR.3, FR.7, FR.14, and FR.15
NFR.11	C.1 and FR.1
NFR.12	C.1
NFR.13	N/A; this is an important quality of the program as a whole and so does not map to any individual FR, constraint, or assumption.
NFR.14	N/A; this is an important quality of the program as a whole and so does not map to any individual FR, constraint, or assumption.
NFR.15	N/A; this is an important quality of the program as a whole and so does not map to any individual FR, constraint, or assumption.

5.3 Reflection

As is the nature of a Capstone course, this project will be a learning experience for all 6 of us. We will all have to learn new skills and improve on old ones to see this through successfully. Below is a detailed breakdown from each Individual on what they feel will be their most important learnings from

this project, and how they plan on going about them.

Adham

Most Important Learning(s): Personally, I believe my biggest knowledge gaps are C++ knowledge, in terms of both general language proficiency as well as unfamiliarity with tools and standards such as Valgrind and clang.

Plan to learn what is needed: I plan to do some self study to acquaint myself with C++, I am familiar with other OOP languages such as Java and Python, and have a good bit of experience with C as well, so I should be able to learn the ropes of C++ on my own. For more specific tools or libraries such as QT, or any other more in depth questions, I will leverage David and Victor, as they are both much more familiar with C++ than I am, to get me over any learning roadblocks. I could also just learn as I go through the implementation, and while that would be more exciting, I think I'll go with the former option for the sake of our group and the project.

David

Most Important Learning(s): I feel I will need to gather some domain specific knowledge, specifically related to bridge engineering. I feel I will also need to work on my technical writing skills (formal and succinct).

Plan to learn what is needed: We have an amazing resource in Dr. Yang and her students, as they are domain experts in bridge engineering. I think throughout the project, speaking to them about bridge engineering will help me learn a lot. I also think a lot can be learned from Google, since a lot of the simpler concepts can be learned about through various sources. I will mainly rely on speaking to Dr. Yang and her students, since I know they are a reliable source for information, especially in terms of the knowledge required for this project. Technical writing can be improved through gaining valuable feedback from our TA on my writing. This can also be improved by practicing an iterative process on my writing, where I can go back and try to simplify my writing while maintaining its meaning. I will mainly rely on the latter, since I feel I can learn a lot more with the iterative process.

Victor

Most Important Learning(s): The knowledge I feel I need to learn the most is domain specific knowledge about civil engineering, specifically about bridge design. I think that this knowledge will be necessary to be able to

successfully complete this project because the program is being designed for bridge engineering.

Plan to learn what is needed: To acquire this knowledge I plan to both do personal research into the subject, and then fill any gaps in my knowledge during our meetings with Dr. Yang through questions. I believe these two strategies will complement each other well as I will be able to acquire most of the knowledge needed without wasting meeting time, then I will be able to ensure that I have a complete understanding using the knowledge of a domain expert.

Darren

Most Important Learning(s): I have little knowledge of bridge engineering. Though I have written them before, I also struggle with formalized project documentation such as design documents. Additionally, while I've worked on software engineer project teams in the past, this is the first software project where I expect to be as involved as all of my peers rather than a team with far more or far less experience on the project and interacting with all of them regularly.

Plan to learn what is needed: I will make use of the meetings with Dr. Yang and her students to gain a deeper understanding of why we are building this product, what came before it, and what it will be used for. I will aim to take a more proactive approach to team meetings and not shying away from feedback while also remaining mindful of potential conflict. I will maintain contact 'off-hours' and open communications with team members.

Farzad

Most Important Learning(s): I believe success of any project relies on team management the most. Adopting any level of agile methodologies can not only deepen our understanding of what are the best approaches through trial and error and prototyping but also it allows our stakeholders to have a lot more influence on the outcome of the project by providing rapid feedback. Even though the course is enforcing waterfall by having step by step milestones and documentation, I believe we can still add a little bit of agile to the mix to reach a perfect balance as this is usually the case in real life.

Plan to learn what is needed: In order to become more agile everyone in the team including myself needs to get trained on the matter. On top of

that we need to have a retrospective meeting after each sprint to see which agile practices were performed well or inadequately. Both approaches are required, however educating myself on the matter has the highest priority since no meaningful retrospective session can be held if at least some people don't have a deep enough understanding of agile.

Pedram

Most Important Learning(s): Personally, the most important learning includes deepening my knowledge of C++ frameworks, specifically Qt which we will be likely using to create the MTOBridge features. I will take this opportunity to learn deeper about interactions between programming languages and how different programs can be synced. I think this is also a great opportunity to understand design thinking and how it can go beyond the everyday user and be applied to an actual engineer.

Plan to learn what is needed: My plan involves rapidly understanding of the tools through frequent building and testing. This process can be accelerated with the use of proper documentation or even help of the friends. It's also crucial to understand about what's happening in the engine, requiring us to increase our domain knowledge.