

Our Awesome Project

Requirements Standard Plan

Saad Salman, Author 2, Author 3

Version 1, 2025-09-22

Contents

Academic Integrity Disclaimer	3
1 Control Information	4
2 (G) Goals	5
2.1 (G.1) Context and Overall Objectives	5
2.2 (G.2) Current Situation	6
2.3 (G.3) Expected Benefits	6
2.4 (G.4) Functionality Overview	6
2.5 (G.5) High-level Usage Scenarios	7
2.6 (G.6) Limitations and Exclusions	7
2.7 (G.7) Stakeholders and Requirements Sources	8
3 (E) Environment	9
3.1 (E.1) Glossary	9
3.2 (E.2) Components	9
3.3 (E.3) Constraints	9
3.4 (E.4) Assumptions	9
3.5 (E.5) Effects	10
3.6 (E.6) Invariants	10
4 (S) System	11
4.1 (S.1) Components	11
4.2 (S.2) Functionality	11
4.3 (S.3) Interfaces	11
4.4 (S.4) Detailed Usage Scenarios	11
4.5 (S.5) Prioritization	11
4.6 (S.6) Verification and Acceptance Criteria	11
5 (P) Project	12
5.1 (P.1) Roles and Personnel	12
5.2 (P.2) Imposed Technical Choices	12
5.3 (P.3) Schedule and Milestones	12
5.4 (P.4) Tasks and Deliverables	12
5.5 (P.5) Required Technology Elements	12
5.6 (P.6) Risk and Mitigation Analysis	12
5.7 (P.7) Requirements Process and Report	12

Academic Integrity Disclaimer

We would like to acknowledge that as dedicated students of McMaster University, we have thoroughly read and comprehended the Academic Integrity Policy published by the university. We are committed to upholding the principles of academic honesty and integrity in all aspects of our educational journey. We understand the importance of acknowledging the work and ideas of others, and we pledge to ensure that all our academic endeavors are conducted with the utmost originality and compliance with the university's policy.

We affirm that the content presented in this document is entirely our own, and any external sources used have been appropriately cited and referenced.

Saad Salman

As I submit my work, I, Saad Salman, take full responsibility for the integrity of my work and promise to avoid any form of plagiarism, cheating, or dishonest behavior. This acknowledgment serves as a testament to my dedication to academic excellence and the fostering of a trustworthy academic community at McMaster University.

Author 2

As I submit my work, I, Author 2, take full responsibility for the integrity of my work and promise to avoid any form of plagiarism, cheating, or dishonest behavior. This acknowledgment serves as a testament to my dedication to academic excellence and the fostering of a trustworthy academic community at McMaster University.

Author 3

As I submit my work, I, Author 3, take full responsibility for the integrity of my work and promise to avoid any form of plagiarism, cheating, or dishonest behavior. This acknowledgment serves as a testament to my dedication to academic excellence and the fostering of a trustworthy academic community at McMaster University.

Policy: <https://secretariat.mcmaster.ca/app/uploads/Academic-Integrity-Policy-1-1.pdf>

1 Control Information

Table 1: *
Versioning and Delivery

Version	Delivery Deadline	Delivered	Feedback Received	Integrated	Notes
V1					
V2					
V3					

Saad Salman

Here is a quick biography of Saad Salman. You can contact them at salmam12@mcmaster.ca.

Author 2

Here is a quick biography of Author 2. You can contact them at a2@mcmaster.ca.

Author 3

Here is a quick biography of Author 3. You can contact them at a3@mcmaster.ca.

2 (G) Goals

Reading Guide

Goals are “needs of the target organization, which the system will address”. While the development team is the principal user of the other books, the Goals book addresses a wider audience: essentially, all stakeholders [1]. It must contain enough information to provide—if read just by itself—a general sketch of the entire project. To this effect, chapter G.3 presents a short overview of the system and G.1 will typically include some key properties of the environment. As it addresses a wide readership, it should be clear and minimize the use of specialized technical terms. Together, G.1, G.2 and G.3 describe the rationale for the project. It is important to state these justifications explicitly. Typically, they are well understood at the start of the project, but management and priorities can change [1].

Control Information

Table 2: *					
Table 1. Our Awesome Project — Versioning Information — Goal Book					
Section	Version	Lead	Delivered on	Reviewer	Approved on
G.1	M1	SS	Sept-21-2025	SP	Sept-22-2025
G.2	M1	SS	Sept-21-2025	SP	Sept-22-2025
G.3					
G.4					
G.5					
G.6					
G.7					

2.1 (G.1) Context and Overall Objectives

High-level view of the project: organizational context and reason for building a system. It explains why the project is needed, recalls the business context, and presents the general business objectives [1].

New university students trying to make connections with each other can often find it difficult due to the limited interactions they have outside of their program. *HammerCorp Inc.* will address this by developing a web and mobile app called **ACME Connect**, which connects students based on similar interests or experiences, as well as student clubs or institutional activities. It aims to improve the overall university experience by creating a tightly-knit student community where integration of official activities happens at the student level. With safety as a primary goal, this project plans to be available to universities across North America, with a pilot program starting at McMaster University.

2.2 (G.2) Current Situation

Current state of processes to be addressed by the project and the resulting system. It describes the current situation upon which the system is expected to improve [1].

At the moment, students are limited to in-person connections within their program or at Welcome Week events, which are often not sufficient for engagement and networking. Residence and international students may feel disconnected in a new place. Student clubs and associations also find it difficult to engage potential members. Although there are in-person events for new students and club booths around campus, those are temporary pop-ups and may not reach everyone. There is no central place for everyone to connect outside those events. Discord servers exist; however, they are often not institutionally based, which can compromise safety and allow complete strangers not affiliated with the school. Looking at these challenges, **ACME Connect** plans to address them and create a safe environment for students and institutions to foster lasting relationships throughout the university journey.

2.3 (G.3) Expected Benefits

New processes, or improvements to existing processes, made possible by the project's results. It presents the business benefits expected from successful execution of the project. This chapter is the core of the Goals book, describing what the organization expects from the system [1].

Nothing available at this point.

2.4 (G.4) Functionality Overview

Overview. Hitchly is a campus-focused ride sharing service for the McMaster community. It verifies users by McMaster email, collects commute and vehicle data, matches riders and drivers by schedule and location, estimates cost sharing, and supports post-trip ratings.

Functional requirements

1. **User verification.** At sign-up, users must verify a McMaster-affiliated email address before any ridesharing features become available. This ensures only students, staff, and faculty access the platform and underpins trust and safety.
2. **Profile & schedule input.** Riders and drivers provide basic profile details plus commute timetable, typical routes/areas, and preferences. Drivers additionally record vehicle and license information to enable appropriate matching and accountability.
3. **Ride matching.** The system proposes compatible rider–driver pairings based on timetable overlap and approximate location. A match only becomes a trip after both sides confirm.
4. **Cost estimate.** For each proposed trip, Hitchly computes a per-rider cost share that helps drivers offset expenses and helps riders save compared to solo travel. Estimates use driver-provided cost inputs at a simple, transparent level.
5. **Ratings & reviews (extension).** After trips, users can leave ratings and short reviews to inform future choices and encourage good behavior. These signals improve decision-making over time while remaining out of scope for the initial PoC.

Most important two

User verification and **Ride matching** are most important. Verification directly addresses safety/trust, a primary adoption risk; matching is the core value creation that reduces search time and enables cost sharing. Together they validate the product's viability and de-risk the project early.

Non-functional requirements

1. **Reliability (matching correctness).** Under typical peak usage, the matching function should consistently produce correct, reproducible pairings for the same inputs. Incorrect or inconsistent matches are treated as defects and prioritized in testing.
2. **Usability (low-friction flows).** Common tasks (sign-up/verification, offering a ride, requesting a ride, confirming a match) should be understandable without training and completable in a small number of obvious steps. Clear labels and uncluttered presentation minimize onboarding time for first-time users.

RACI for requirements

Requirement	Dev Team	Drivers	Riders	McMaster Univ. / Hamilton Community
User verification	R/A	C	C	I
Profile & schedule input	R/A	C	C	I
Ride matching	R/A	C	C	I
Cost estimate	R/A	C	C	I
Ratings & reviews (extension)	R/A	C	C	I
Reliability (NFR)	R/A	C	C	I
Usability (NFR)	R/A	C	C	I

Legend: R = Responsible (executes the work); A = Accountable (final sign-off); C = Consulted (provides input); I = Informed (kept up to date).

2.5 (G.5) High-level Usage Scenarios

Fundamental usage paths through the system, stated in user terms and independent of the system's structure. Detailed usage scenarios appear in the System book (S.4) [1].

(High level use case diagram placeholder)

Nothing available at this point.

2.6 (G.6) Limitations and Exclusions

Aspects that the system need not address; scoping the requirements. (Risks and obstacles belong to P.6.) [1].

Nothing available at this point.

2.7 (G.7) Stakeholders and Requirements Sources

Stakeholder categories that can affect or be affected by the project, and other sources of requirements information [1].

Nothing available at this point.

3 (E) Environment

The Environment book describes the application domain and external context (physical or virtual) in which the system will operate [1].

Control Information

Table 3: *

Table 2. Our Awesome Project — Versioning Information — Environment Book

Section	Version	Lead	Delivered	Reviewer / Approved
E.1				
E.2				
E.3				
E.4				
E.5				
E.6				

3.1 (E.1) Glossary

Clear and precise definitions of all vocabulary specific to the application domain [1].

Nothing available at this point.

3.2 (E.2) Components

List of environment elements that may affect or be affected by the system and project, including other systems for interfacing [1].

Nothing available at this point.

3.3 (E.3) Constraints

Obligations and limits imposed by the environment (business rules, physical laws, engineering decisions) [1].

Nothing available at this point.

3.4 (E.4) Assumptions

Environment properties assumed to hold to facilitate the system's construction [1].

Nothing available at this point.

3.5 (E.5) Effects

Elements and properties of the environment that the system will affect [1]. Effects of the application include:

- **Improved student belonging and wellbeing:** The application fosters safe and supportive environments that help students feel less isolated and more connected during the transition to university.
- **Increased participation in campus life:** By lowering barriers to joining clubs, events, and residence activities, the app encourages higher engagement with student associations and university-led initiatives.
- **Enhanced student retention and success:** With better access to support networks and resources, students are less likely to drop out due to isolation or lack of integration.

3.6 (E.6) Invariants

Environment properties that the system's operation must preserve [1].

Nothing available at this point.

4 (S) System

The System book refines the Goals book by focusing on more detailed requirements.

4.1 (S.1) Components

Overall structure expressed by the list of major software and, if applicable, hardware parts [1].

Nothing available at this point.

4.2 (S.2) Functionality

The bulk of the System book: functional and non-functional behaviors [1].

Nothing available at this point.

4.3 (S.3) Interfaces

How the system makes S.2 functionality available to the rest of the world (UIs and APIs) [1].

Nothing available at this point.

4.4 (S.4) Detailed Usage Scenarios

User stories and examples of interaction between users/environment and the system [1].

Nothing available at this point.

4.5 (S.5) Prioritization

Classification of behaviors, interfaces, and scenarios by criticality [1].

Nothing available at this point.

4.6 (S.6) Verification and Acceptance Criteria

Conditions under which an implementation will be deemed satisfactory; V&V levels [1].

Nothing available at this point.

5 (P) Project

5.1 (P.1) Roles and Personnel

Main responsibilities, required staff, and qualifications [1].

Nothing available at this point.

5.2 (P.2) Imposed Technical Choices

A priori choices binding the project to specific tools, hardware, languages or other technical parameters [1].

Nothing available at this point.

5.3 (P.3) Schedule and Milestones

List of tasks to be carried out and their scheduling [1].

Nothing available at this point.

5.4 (P.4) Tasks and Deliverables

Details of individual tasks and expected outcomes, associated with milestone dates [1].

Nothing available at this point.

5.5 (P.5) Required Technology Elements

External systems, hardware and software expected to be necessary for building the system [1].

Nothing available at this point.

5.6 (P.6) Risk and Mitigation Analysis

Potential obstacles to meeting the schedule and measures for adapting the plan [1].

Nothing available at this point.

5.7 (P.7) Requirements Process and Report

Initially: description of the requirements process; later: report on its steps [1].

Nothing available at this point.

References

- [1] Bertrand Meyer. *Handbook of Requirements and Business Analysis*. Springer, 2022.
- [2] Ian Sommerville and Peter Sawyer. *Requirements Engineering: A Good Practice Guide*. Wiley, 1997.