
SOFTWARE REQUIREMENTS SPECIFICATION

Room8

Version 1.0.0

Prepared by:
Mohammed Abed
Maged Armanios
Jinal Kasturiarachchi
Jane Klavir
Harshil Patel

McMaster University

October 11, 2024

Revision History

Name	Date	Reason For Changes	Version
J.-M. Bruel	2021-01-22	First Draft	1.0
J.-M. Bruel	2023-01-28	Check after publication of the Handbook	1.23
J.-M. Bruel	2023-06-12	Add reqs automated numbering	1.23.1
J.-M. Bruel	2023-08-25	Add Minimum Requirements Outcome Principle	1.23.8
J.-M. Bruel	2023-12-22	Remove section numbers	1.23.12
J.-M. Bruel	2024-08-01	Add warning about non empty chapters	1.0.0

This document follows the requirements documentation structure presented in the [Handbook of requirements and business analysis](#), by Bertrand Meyer.

Contents

Goals book	3
G.1 Context and overall objective	4
G.2 Current situation	4
G.3 Expected benefits	4
G.4 Functionality overview	5
G.5 High-level usage scenarios	5
G.6 Limitations and exclusions	9
G.7 Stakeholders and requirements sources	9
Environment book	10
E.1 Glossary	10
E.2 Components	10
E.3 Constraints	10
E.4 Assumptions	10
E.5 Effects	11
E.6 Invariants	11
System book	12
S.1 Components	12
S.2 Functionality	12
S.3 Interfaces	12
S.4 Detailed usage scenarios	12
S.5 Prioritization	13
S.6 Verification and acceptance criteria	13
Project book	14
P.1 Roles and personnel	14
P.2 Imposed technical choices	14
P.3 Schedule and milestones	14
P.4 Tasks and deliverables	14
P.5 Required technology elements	14
P.6 Risks and mitigation analysis	15
P.7 Requirements process and report	15

Goals

Contents

G.1 Context and overall objective	4
G.2 Current situation	4
G.3 Expected benefits	4
G.4 Functionality overview	5
G.5 High-level usage scenarios	5
G.6 Limitations and exclusions	9
G.7 Stakeholders and requirements sources	9

Comment: *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.*

G.1 Context and overall objective

Comment: *High-level view of the project: organizational context and reason for building a system.*
Comment: *This chapter should not be empty!*

Goal 1. This is a goal example. If you need explicit (and automatic) numbering, you can use the definitions in the `.tex` template. Is refined by [3](#)

Requirement 2. *This is a requirement example. It illustrates how numbering is continuous and cross-types (if this is what you need).*

G.2 Current situation

Comment: *Current state of processes to be addressed by the project and the resulting system.*

Requirement 3. *This is a requirement example. It refines [1](#)*

G.3 Expected benefits

Comment: *New processes, or improvement to existing processes, made possible by the project's results.* **Comment:** *This chapter should not be empty!*

G.4 Functionality overview

Comment: *Overview of the functions (behavior) of the system. Principal properties only (details are in the System book).*

User & House Management: The system allows students to register, create, and manage their accounts. Students are also able to create homes and invite other students.

Cleanliness Management: Students are able to set up cameras in their shared living spaces. Machine learning algorithms are used to detect messes and assign them to the students to increase accountability and to reduce communication friction.

Scheduler: The system allows users to create and manage chore and cleaning schedules. It will also send reminders to users about their assigned chores and cleaning tasks. Additionally, users are able to "book" common areas that are available within the shared space (i.e. party room, study room, etc.) to avoid conflicts

Bill Splitter: Students are able to add shared expenses to the house and keep track of who owes what. The system will automatically split the bills between the chosen students.

Chat: The system provides an exportable chatbot to SMS group chats to aid in home management and communication. The chatbot will be able to send reminders and notifications to the group chat about upcoming events, chores, and bills.

Compliance and Data Privacy: The system follows data privacy and compliance with Personal Information Protection and Electronic Documents Act (PIPEDA), Ontario's Freedom of Information and Protection of Privacy Act (FIPPA), and Anti-Spam Legislation (CASL) to ensure that student's data is secure and not shared with third parties.

Progressive Web App: The system is hosted as a progressive web app (PWA), so users can use it on their phones and computers.

G.5 High-level usage scenarios

Comment: *Fundamental usage paths through the system.*

- UC1: Create an account
 1. User accesses the Room8 app.
 2. User selects the "Create Account" option.
 3. User fills in personal information (name, email, password).
 4. User submits the form.
 5. System validates the information and creates the account.

6. System authenticates the user.
 7. System redirects the user to their dashboard.
- UC2: Log in to the system
 1. User accesses the Room8 app.
 2. User selects the "Log In" option.
 3. User enters email and password.
 4. User submits the form.
 5. System authenticates the credentials and logs in the user.
 6. System redirects the user to their dashboard.
 - UC3: Log out of the system
 1. User accesses the main menu of the Room8 app.
 2. User selects the "Log Out" option.
 3. System prompts for confirmation.
 4. User confirms the action.
 5. System logs the user out and redirects to the login screen.
 - UC4: Create a home
 1. User navigates to their dashboard.
 2. User selects the "Create Home" option.
 3. User enters home details (home name, address, etc.).
 4. User sends invitations to housemates by entering their email addresses.
 5. System sends email notifications to invited users.
 6. Invited users accept or decline the invitation.
 - UC5: Join a home
 1. User receives an email invitation to join a home.
 2. User clicks on the provided link in the invitation.
 3. System redirects user to the Room8 app.
 4. User logs in or creates an account if not logged in.
 5. User confirms their decision to join the home.
 6. System adds the user to the home group.
 - UC6: Leave a home
 1. User navigates to their Home settings.
 2. User chooses the "Leave Home" option.

3. System prompts for confirmation.
 4. User confirms the action.
 5. System removes the user from the home group.
- UC7: Update user profile
 1. User navigates to the "Profile" section.
 2. User selects the "Edit Profile" option.
 3. User modifies personal information (name, profile picture, etc.).
 4. User saves changes.
 5. System updates the user's profile information.
 - UC8: Schedule a chore
 1. User navigates to the "Schedule" section.
 2. User selects "Add New Chore".
 3. User inputs chore details (name, description, time, frequency, assigned users, etc.).
 4. System saves the chore and sends reminders to the assigned users.
 - UC9: Edit a chore
 1. User navigates to the "Schedule" section.
 2. User selects a scheduled chore.
 3. User chooses the "Edit" option.
 4. User modifies the chore details (name, description, time, frequency, assigned users, etc.).
 5. System updates the chore and notifies relevant housemates.
 - UC10: Complete a chore
 1. User navigates to the "Schedule" section.
 2. User selects a chore assigned to them.
 3. User chooses the "Complete" option.
 4. System marks the chore as complete and updates the chore history.
 - UC11: Add event to schedule
 1. User navigates to the "Schedule" section.
 2. User selects the "Add Event" option.
 3. User inputs event details (date, time, description).
 4. User sets reminders or notifications if needed.

5. System adds the event to the users' schedule.
- UC12: View chore history
 1. User navigates to the "Schedule" section.
 2. User selects the "Chore History" option.
 3. System displays a list of completed and pending chores.
 4. User can filter or search through the history.
 - UC13: Add an expense
 1. User navigates to the "Bill Splitter" section.
 2. User selects the "Add Expense" option.
 3. User enters the total expense amount and description.
 4. User selects which housemates are responsible for the expense.
 5. System splits the expense and notifies the involved users.
 - UC14: Edit an expense
 1. User navigates to the "Bill Splitter" section.
 2. User selects an expense to edit.
 3. User modifies the expense details (amount, description, responsible users).
 4. System updates the expense and notifies the involved users.
 - UC15: Pay an expense
 1. User navigates to the "Bill Splitter" section.
 2. User selects their profile's bills.
 3. User selects an unpaid bill.
 4. User marks the bill as paid.
 5. System updates the payment status.
 6. System notifies other housemates about the payment status.
 - UC16: View expense history
 1. User navigates to the "Bill Splitter" section.
 2. User selects the "Expense History" option.
 3. System displays a list of past payments and outstanding bills.
 4. User can filter the history by type or amount.
 - UC17: View current cleanliness status
 1. User navigates to the "Cleanliness Management" section.
 2. System displays the current cleanliness score and detected messes.

3. User selects another user to view their cleanliness score.
 4. System displays the selected user's cleanliness score and pending tasks.
- UC18: View cleanliness history
 1. User navigates to the "Cleanliness Management" section.
 2. User selects the "Cleanliness History" option.
 3. System displays a list of detected messes and cleanliness scores over time.
 4. User can filter the history by location or date.
 - UC19: Add chatbot to group chat
 1. User navigates to the "Chat Management" section.
 2. User selects the option to add a chatbot to their house group chat.
 3. User inputs notification and reminder settings.
 4. System generates an SMS numbers and instructions to add the chatbot to the group chat.
 5. The chatbot sends a welcome message to the group.
 - UC20: Edit chatbot settings
 1. User navigates to the "Chat Management" section.
 2. User selects the "Chatbot Settings" option.
 3. User configures the chatbot's behavior (e.g., reminders, notifications).
 4. User saves the settings.
 5. System updates the chatbot based on the new settings.

G.6 Limitations and exclusions

Comment: *Aspects that the system need not address.*

G.7 Stakeholders and requirements sources

Comment: *Groups of people who can affect the project or be affected by it, and other places to consider for information about the project and system.* **Comment:** *This chapter should not be empty!*

Environment

Contents

G.1 Context and overall objective	4
G.2 Current situation	4
G.3 Expected benefits	4
G.4 Functionality overview	5
G.5 High-level usage scenarios	5
G.6 Limitations and exclusions	9
G.7 Stakeholders and requirements sources	9

Comment: *The Environment book describes the application domain and external context, physical or virtual (or a mix), in which the system will operate.*

E.1 Glossary

Comment: *Clear and precise definitions of all the vocabulary specific to the application domain, including technical terms, words from ordinary language used in a special meaning, and acronyms. This chapter should not be empty!*

E.2 Components

Comment: *List of elements of the environment that may affect or be affected by the system and project. Includes other systems to which the system must be interfaced.*

E.3 Constraints

Comment: *Obligations and limits imposed on the project and system by the environment.* **Comment:** *This chapter should not be empty!*

E.4 Assumptions

Comment: *Properties of the environment that may be assumed, with the goal of facilitating the project and simplifying the system.*

E.5 Effects

Comment: *Elements and properties of the environment that the system will affect.*

E.6 Invariants

Comment: *Properties of the environment that the system's operation must preserve.*

System

Contents

G.1 Context and overall objective	4
G.2 Current situation	4
G.3 Expected benefits	4
G.4 Functionality overview	5
G.5 High-level usage scenarios	5
G.6 Limitations and exclusions	9
G.7 Stakeholders and requirements sources	9

Comment: *The System book refines the Goal one by focusing on more detailed requirements about the system under development, mainly its constituents, behaviors and properties.*

S.1 Components

Comment: *Overall structure expressed by the list of major software and, if applicable, hardware parts.*

Comment: *This chapter should not be empty!*

S.2 Functionality

Comment: *One section, S.2.n, for each of the components identified in S.2, describing the corresponding behaviors (functional and non-functional properties).* **Comment:** *This chapter should not be empty!*

S.3 Interfaces

Comment: *How the system makes the functionality of S.2 available to the rest of the world, particularly user interfaces and program interfaces (APIs).*

S.4 Detailed usage scenarios

Comment: *Examples of interaction between the environment (or human users) and the system: use cases, user stories.*

S.5 Prioritization

Comment: *Classification of the behaviors, interfaces and scenarios (S.2, S.3 and S.4) by their degree of criticality.*

S.6 Verification and acceptance criteria

Comment: *Specification of the conditions under which an implementation will be deemed satisfactory.*

Project

Contents

G.1 Context and overall objective	4
G.2 Current situation	4
G.3 Expected benefits	4
G.4 Functionality overview	5
G.5 High-level usage scenarios	5
G.6 Limitations and exclusions	9
G.7 Stakeholders and requirements sources	9

Comment: *The Project book describes all the constraints and expectations not about the system itself, but about how to develop and produce it.*

P.1 Roles and personnel

Comment: *Main responsibilities in the project; required project staff and their needed qualifications.*

P.2 Imposed technical choices

Comment: *Any a priori choices binding the project to specific tools, hardware, languages or other technical parameters.*

P.3 Schedule and milestones

Comment: *List of tasks to be carried out and their scheduling.* **Comment:** *This chapter should not be empty!*

P.4 Tasks and deliverables

Comment: *Details of individual tasks listed under P.3 and their expected outcomes.* **Comment:** *This chapter should not be empty!*

P.5 Required technology elements

Comment: *External systems, hardware and software, expected to be necessary for building the system.*

P.6 Risks and mitigation analysis

Comment: *Potential obstacles to meeting the schedule of P.4, and measures for adapting the plan if they do arise.*

P.7 Requirements process and report

Comment: *Initially, description of what the requirements process will be; later, report on its steps.*

Appendix — Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Lifelong Learning.

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?