

Requirements 1

Cohort 1 Group 6

Group Members:

Hussain Alhabib

Ellen Matthews

Minnie Poon

Jason Ruan

Daniel Smith

Owen Smith

Eliciting our requirements

To facilitate the implementation of our project, we elicited requirements directly from our client via a 20-minute in-person interview. The interview, which was recorded and conducted by two group members, followed an interview schedule that had been collaboratively worked on by all team members.

This structured approach to the first client interview allowed us to generate a comprehensive document containing the client's answers to over 20 questions, which helped assist the team later on when producing the statement of requirements and respective referencing system.

Presenting our requirements

In the following group meeting, we began to interpret the client's responses and produce our requirement referencing system. After researching and discussing best practices in requirements engineering as a group, particularly in Ian Sommerville's *Software Engineering 10th Edition*, we decided to present the requirements in a tabular format, splitting them into four tables based on the previously established categories. Each table was formatted with columns for the requirement ID, description, priority, requirement link, and current status.

We considered several other formats for the referencing system, including Natural Language, Structured, and Use Case (UML) specifications. However, we found a Natural Language specification lacked the clarity and detail needed for our primary stakeholders, despite being accessible for wider less-technical audiences. The Structured specification was too complex given our project's scope, and while a UML specification seemed useful especially due to its more visual aspects it seemed redundant as our architecture section would already include graphical elements. Ultimately, we felt as a team that the tabular format with an Agile-inspired approach, that prioritised requirements based on importance, best suited our needs, enabling us to collectively determine the order and focus of requirement implementation.

Negotiating our requirements

To ensure the project aligned with the client's vision, we shared drafts of the requirements document to encourage feedback and revisions. This approach enabled us to refine the requirements with the client's input and approval, which ensured that our understanding and implementation of the game matched the client's expectations. After acting on feedback, we emailed the client our final draft of the requirements, which they approved before we proceeded with designing our requirements referencing system.

Our Requirements Referencing System

Single Statement of Need

We will design a game called UniSim that allows players to manage a university campus by placing and upgrading buildings, with success metrics for student satisfaction and university income while navigating building restrictions and random events within a 5-minute gameplay session.

Introduction to Requirement Categories

We defined our initial client questions and their respective requirements according to the following four categories: user requirements, system requirements, non-functional requirements, constraint requirements.

User requirements prioritise tasks that users should be able to perform throughout gameplay. These should avoid using technical jargon and be accessible to everyone regardless of their technical skills.

System requirements address how the game will meet users' needs. These include detailed, often technical, descriptions of various functionalities, services, and constraints that may impact the overall gameplay experience. Multiple system requirements can be written to fulfil a single user requirement.

Non-functional requirements focus on the game's attributes and qualities, and define how well the game should perform regarding usability, reliability, and performance.

Constraint requirements define various limitations or restrictions that may be imposed on the game, such as hardware limitations and any relevant legal regulations we must adhere to.

Requirements Referencing System

Each requirement has been assigned a unique ID ensuring traceability and consistency throughout our documentation. This allows us to establish connections between requirements, for example each system requirement must have a link to a user requirement. Our table includes a column to list related requirement IDs for each requirement. This will support us in visualising how the requirements relate to and may impact one another.

We initially assigned IDs in numerical order, but chose to use meaningful ID names instead. We decided this would improve readability and provide clarity when referencing the IDs in other documentation.

The finalised requirements referencing system can be found on the website.

User Requirements (URs)

ID	Description	Priority	User Requirement Link	Status
UR_BASIC_BUILDINGS	The player must be able to place at least one type of each building type; a place to learn, a place to sleep, a place to eat, and a recreational activity.	Essential	SR_PLACE_BUILDINGS	APPROVED
UR_TIME_TRACKING	The game should last for a maximum of 5 real-world minutes.	Essential	SR_TIME	APPROVED
UR_BUILDING_COUNTER	The game should track and display the number of each building type (e.g. sleep, eat, learn, recreational) placed by the player.	Essential	SR_BUILDING_COUNTER	APPROVED
...

System Requirements (SRs)

ID	Description	Priority	User Requirement Link	Status
SR_PLACE_BUILDINGS	Players should be able to place, upgrade, and demolish buildings within the game.	Essential	UR_BASIC_BUILDINGS	APPROVED
SR_TIME	Time should be tracked and shown within gameplay, lasting for a maximum of 5 real-world minutes.	Essential	UR_TIME_TRACKING	APPROVED
SR_BUILDING_COUNTER	The game should count and display the number of each building type placed by the player.	Essential	UR_BUILDING_COUNTER	APPROVED
...

To ensure our requirements fit within the page limit, some tables have been condensed. To see the remainder of the User, System & Non-Functional Requirement Tables please view the full Requirement Referencing System on the Website: [\[ADD LINK\]](#)

Non-Functional Requirements (NFRs)

ID	Description	Priority	User Requirement Link	Status
NFR_RUNS_WELL	The game should run smoothly on all laptops and desktops on major operating systems.	High	UR_DEPLOYMENT	APPROVED
NFR_NO_DELAY	The gameplay experience should be smooth, with minimal delays or lag during player interactions.	High	UR_DEPLOYMENT	APPROVED
NFR_ACCESSIBLE	The game should be accessible to as wide an audience as possible, accommodating players with different needs.	High	UR_TIPS	APPROVED
NFR_GRAPHICS	The graphics should reflect the selected environment, maintaining visual cohesion throughout the game.	High	UR_BASIC_BUILDINGS	APPROVED
...

Constraint Requirements (CRs)

ID	Description	Priority	User Requirement Link	Status
CR_LOCAL	The game should run locally on a device without needing an internet connection.	High	UR_DEPLOYMENT	APPROVED
CR_LOW_SPEC	The game should be optimised to run on low-spec devices, ensuring it is accessible to all players.	High	UR_DEPLOYMENT	APPROVED
CR_LEGAL	The game should be legally compliant, using appropriately licensed and attributed assets.	High	UR_SETTINGS, UR_BASIC_BUILDINGS	APPROVED

To ensure our requirements fit within the page limit, some tables have been condensed. To see the remainder of the User, System & Non-Functional Requirement Tables please view the full Requirement Referencing System on the Website: [\[ADD LINK\]](#)