Requirements Engineering and Elicitation Process

Our requirements engineering and elicitation process conforms to the standards outlined in the IEEE requirements engineering documentation [1]. After extensive research into proposed standards for software design and requirements elicitation, we collectively agreed that the standards presented in this document are the most descriptive and provide an intuitive and accessible categorization of our users' needs.

The primary incentive behind this decision was the use of modal verbs such as *May*, *Shall*, and *Should* as descriptors for the priority of our requirements [1, p.19]. These terms unequivocally outline the importance of each requirement in a manner comprehensible to both the design team and the client. This approach is supported by leading organizations in the technology industry, including NASA, which employs this standard in its own requirements matrix [2], further underscoring the robustness of this documentation quality.

The requirements were outlined to the team during a meeting with the client, who requested a multitude of features for the final product. These specifications were subsequently refined through collaborative team discussions, ensuring that each resulting component:

- Solves a problem,
- Achieves a goal, or
- Addresses a stakeholder concern.

Most importantly, each requirement was *qualified by measurable conditions* [1, p.18]. This focus on measurable outcomes will become increasingly critical during the testing phases, where compliance with agreed constraints will determine success. While initial requirements are often subject to change—42% of similar projects face issues articulating user needs [3, p.14]—we aim to mitigate this risk through regular client meetings to validate and refine the requirements as necessary.

In our requirements table, user-level elicitation is categorized into:

1. **Functional requirements**: Tasks the system must perform.
2. **Non-functional requirements**: Qualities and properties the system must exhibit.

It is acknowledged that the Agile methodology can sometimes conflict with the documentation and delivery of non-functional requirements, as it *prioritizes the incremental delivery of functional features* [4, p.1]. However, we believe the IEEE standards address this challenge effectively [1, p.24]. By identifying these requirements early in the design process, we aim to maximize functional output without sacrificing the utility and quality desired by the user.

The research conducted has enhanced our understanding of the requirements elicitation process and equipped us with tools to accurately identify user needs and describe their measurement during the testing stage. These insights provide a solid foundation for drafting and building the product for our client. We are confident that issues related to user requirements will be promptly addressed through effective communication between the team and stakeholders. Weekly sprints, guided by Agile methodology, will ensure continuous alignment with the client's vision.

ID	Description	Priority
UR_OS_COMPATIBILITY	The game shall run on Windows, Mac, and Linux operating systems.	Shall
UR_HARDWARE_COMPATABILITY	The game shall run on minimal specification hardware.	Shall
UR_DISPLAY_SCALIBILITY	The game should run correctly on different DPI displays.	Should
UR_COLOURBLIND_ACCESSIBILITY	The game shall be accessible to colorblind users.	Shall
UR_PAUSE	The user shall be able to pause the game.	Shall
UR_TIMER	The game shall run for 5 real-world minutes, representing 3 in-game years.	Shall
UR_TIME_TRACKER	The game should use weeks and semesters so the user can track the in-game time	Should
UR_AUDIO	There should be background music and sound effects.	Should
UR_COUNTER	The game shall have a counter with the amount of buildings that have been placed	Shall

UR_LOCATIONS	There should be at least one of each building location (one place to sleep, one place to learn, one place to eat, one recreational activity)	Shall
UR_LOCATIONS_MOVE	The user should be able to move buildings once they've been placed	Should
UR_LOCATIONS_SIZES	Different locations shall be different shapes and sizes so the user can easily differentiate them	Should
UR_MAP	The user shall be shown a map with sensible constraints (e.g., water, roads, or hills) that cannot be built upon, affecting building placement and overall satisfaction	Shall
UR_LOCATIONS_UPGRADE	Buildings may be able to be upgraded to allow additional features or capacity	May
UR_MENU	The game should start with a menu screen	Should
UR_LOCATION_PLACEABILITY	The system shall show to the user where they can or can't place a building.	Shall
UR_EVENTS	The game shall have at least three events occurring over the five-minute time span	Shall
UR_STUDENT_SATISFACTION	The game shall track and update a student satisfaction metric that reflects overall campus usability and enjoyment	Shall

UR_RECREATION_X2	The game shall support placing at least two different recreational buildings	Shall
UR_ACHIEVMENTS	The game shall have achievements available for the player to unlock by doing certain actions	Shall
UR_LEADERBOARD	The game shall have a leaderboard to track the highest five scores achieved	Shall
UR_LOCATIONS_DELETE	The user should be able to delete buildings after placing them down	Should

Functional requirements

ID	Description	User Requirements
FR_STOP_TIME	The user should be able to stop the time at any point during the game. The user should be able to queue buildings to be placed, which will start construction once the game is unpaused	UR_PAUSE
FR_AUDIO_MUTE	The game should have a button that allows the user to mute the sounds.	UR_AUDIO
FR_LOCATIONS_PLACEABILITY	The system shall clearly display on the grid whether a building can be placed	UR_LOCATIONS
FR_COUNT_DISPLAY	The system shall update a counter of building types to the user each time one is placed.	UR_COUNTER
FR_LOCATION_PLACEABILITY	The system shall show to the user whether they can place a building in a location on a grid or not.	UR_LOCATION_PLACEABILITY
FR_LOCATIONS_SIZE	Different locations should take up different amounts of squares on the grid.	UR_LOCATIONS_SIZE

FR_TIMER_DISPLAY	The timer shall display the in-game time in months and semesters in accordance with the user's wishes.	UR_TIMER
FR_SOUNDTRACK	The game should play a soundtrack when the title screen is loaded	UR_AUDIO
FR_EVENTS_MANAGEMENT	The system shall schedule and trigger random events at predetermined times	UR_EVENTS
FR_SATISFACTION_UPDATE	The system shall update and recalculate whenever a building is placed, moved, or an event is taking effect	UR_STUDENT_SATISFACTION
FR_RECREATION_MULTIPLICITY	The system shall allow placing at least two types of recreational buildings that are visually different	UR_RECREATION_X2
FR_ACHIEVEMENTS	The system shall have achievements that the player can unlock by doing specific actions or reaching certain goals or milestones	UR_ACHIEVEMENTS
FR_LEADERBOARD	The system shall keep track of all the scores and display the highest five scores	UR_LEADERBOARD
FR_LOCATIONS_DELETE	The systems should allow the user to delete buildings that have been placed	UR_LOCATIONS_DELETE

Non-Functional Requirements

ID	Description	User Requirements	Fit Criteria
NFR_JVM	The system shall support the JVM used by all 3 operating systems to Provide compatibility between OS.	UR_OS_COMPATIBILITY	Complete a Successful run-through of the game on all 3 systems.
NFR_DISPLAY_SIZES	The game should be playable on both laptop and	UR_DISPLAY_SCALABILITY	Test game on varying display sizes to ensure textures

	large whiteboard sized screens		scale appropriately.
NFR_HARDWARE_ COMPATABILITY	The game should run on minimal specification hardware, as expected to be used by the average user.	UR_HARDWARE_C OMPATABILITY	Test game on personal laptops which reflect average user setup.
NFR_PLAYABILITY	The game should be operable by users who have picked the game up for the first time	UR_TOOLTIPS	Have testers who have never played the game before play and collect their feedback on which game mechanics they found intuitive or unintuitive and guide on its usage
NFR_COLOURBLIN D_ACCESSIBILITY	The game shall be playable by users who are visually impaired or colourblind	UR_COLOURBLIND _ACCESSIBILITY	Interview a colourblind QA tester to check if the game is accessible for their needs.