

Requirements

Cohort 3 Team 5 - alltheeb5t

Aaron Heald

Alex Gu

Arun Hill

Jade Stokes

Maksim Soshchin

Meg Tierney

Will Hall

Our team began the requirement section of this project by organising, preparing for, and completing a stakeholder meeting. We prepared a list of questions we had about the system based on the product brief and our research into suitable applications which we could use during the planning and implementation of the product. After the interview, we used the information gathered to create structured tables of requirements to better prepare and ensure smooth running for the design, creation and testing elements of the project.

The interview planning began by reading through the product brief and outlined what we believed to be the most important features for the product. We followed the idea of an open-interview described in Sommerville's *Software Engineering* [1] by asking probing questions and discussing a wide range of topics during the given time. It was also decided that we would have our whole team in the meeting so that members from all aspects of the project would hear directly from our stakeholder, as is suggested by Radigan [2] to prevent any confusion in the exchanging of information.

Following the interview, we decided it was important to follow Natani [3] and to lay out our requirements in user, functional and nonfunctional sections, so as to make them clearer to read through and communicate with others. We displayed them in tables with given priorities (low/medium/high) as explained by Bartlett which "helps your team with clarity and direction" [4]. This allowed us to focus on the most important aspects throughout design and implementation and allowed us to note what was currently out of scope for the project.

When writing the requirements we tried to follow the rules laid out by IncoSE [5] to keep our requirements accurate and unambiguous. We also used natural language throughout so that our stakeholder would be able to completely understand the document without any technical experience. Himes also mentioned the need for all requirements to be verifiable in order to "demonstrate that it fulfils the requirement" [6]. For this reason we decided to link all our functional requirements to user requirements to clearly show where each user requirement has been completed, and add ID tags to each requirement to link them to tests that we run.

It was important to our team that our stakeholder would be able to easily understand our requirements and that we completely met their needs during this project. For that reason we decided to make most of our layout decisions based on how easily someone external to our team would be able to read the document, for example setting each set of requirements out in a table and avoiding the use of technical jargon. We also ensured to ask during our interview, both if there was anything our client wanted to add that hadn't already been mentioned, and what the best methods of contact were for them.

For the purpose of the simplicity of our requirements tables and other documentation linking to them we have used abbreviations in tags:

- UR is User Requirement
- FR is Functional Requirement
- NFR is Nonfunctional Requirement

User Requirements:

Requirement	Priority	Description
UR_Family_Friendly	High	The user must not experience any violence or inappropriate content in the game as it may be demonstrated to all audiences including children.
UR_Audio	Medium	Users should have the option to enable or disable audio to improve their game experience.
UR_No_Preparation	High	The user must be able to play the game with no other information than what is given within the game itself (e.g. no outside tutorial from developers, documentation, etc.)
UR_Map	Medium	The user should be able to recognise and navigate around a 2D Pixel Art map of a university campus.
UR_Buildings	High	Users must be able to place different building types around the campus.
UR_Time	High	The user must have the ability to pause gameplay for an unspecified time. The duration of the game must be 5 real life minutes and 3 in game years.
UR_Events	High	The user must be able to interact with at least three positive, neutral or negative events throughout the game
UR_Money	Medium	The user should deal with financial constraints restricting the user's ability to build new buildings.
UR_Satisfaction	High	The user must be able to view a performance metric called Student Satisfaction throughout the game to gauge their progress.

Functional Requirements:

Requirement	Priority	Description	User Requirement
FR_Map_Display	High	The system must display a map which users can move over.	UR_Map
FR_Map_Features	High	The system must contain some pre-set features on the map.	UR_Map
FR_Time	High	The system must track the minutes which the game has been running.	UR_Time
FR_Semesters	Medium	The system should track and display the week and semester the player is currently in (2 per year + summer).	UR_Time
FR_Pause	High	The system should allow the user to pause the timer for the game by pressing a pause menu.	UR_Time
FR_Pause_Menu	High	The system must start on the pause menu and must have the option to start play.	UR_Time
FR_Settings	Low	The system may have a settings menu within	UR_Time

		the pause menu.	
FR_Mute_Buttons	Low	The system may have a button to mute the music and sound effects.	UR_Audio
FR_Audio_Sliders	Low	The system may have a slider to adjust how loud the audio is.	UR_Audio
FR_Tutorial	High	The system must display a brief explanation of the game before the game starts.	UR_No_Preparation
FR_Place_Building	High	The system must allow a user to place buildings around the campus.	UR_Buildings
FR_Budget	High	The system must not allow the user to place a building if they do not meet the required budget.	UR_Buildings
FR_Move_Building	Medium	The system should allow the user to move previously placed buildings.	UR_Buildings
FR_Building_Timer	Medium	The system should take a set time to build each building once it has been 'placed'.	UR_Buildings
FR_Building_Counter	High	The system must have a counter which tracks and displays the number of placed buildings for each type of building.	UR_Buildings
FR_Building_Type	High	Each building must have a certain type that describes what function it has (eg. accommodation, recreational, etc.)	UR_Buildings
FR_Accommodation_Building	High	The system must allow the user to place an accommodation building.	UR_Buildings
FR_Entertainment_Building	High	The system must allow the user to place entertainment buildings.	UR_Buildings
FR_Study_Building	High	The system must allow the user to place study buildings.	UR_Buildings
FR_Catering_Building	High	The system should allow the user to place catering buildings.	UR_Buildings
FR_Remove_Building	Medium	The system should allow the user to remove the buildings if they can meet the required budget.	UR_Buildings
FR_Building_Shape	Medium	Different building types should have different shapes so they are differentiable by shape not colour.	UR_Buildings
FR_Building_Interaction	Medium	The system should increase the satisfaction score if appropriate buildings are placed close together.	UR_Buildings, UR_Satisfaction
FR_Satisfaction_Score	High	The system must keep track of and display the satisfaction score. It should be highlighted red if it is below the threshold.	UR_Satisfaction

FR_End_Screen	Medium	The system should display a screen showing the user's final score and an option to play again after 5 minutes.	UR_Satisfaction
FR_Money_Tracker	High	The system must track the amount of money the user has and display it on screen.	UR_Money
FR_Income_Expenses	Medium	The system should have multiple ways to increase and decrease the budget.	UR_Money
FR_Events_Popup	High	The system must create a pop up informing the player of an ongoing event.	UR_Events
FR_Positive_Events	High	The system must create a positive event that increases satisfaction and/or budget	UR_Events
FR_Neutral_Events	High	The system must create a neutral event that doesn't affect satisfaction or budget	UR_Events
FR_Negative_Events	High	The system must create a negative event that decreases satisfaction and/or budget	UR_Events

Non Functional Requirements:

Requirement	Priority	Description
NFR_OS	High	The game must run smoothly on Windows, Linux and Mac OS.
NFR_Scale	High	The system must scale to different window sizes and still correctly display the game.
NFR_Licencing	High	The system must not violate the licensing requirements of any of the 3rd party libraries, tools and assets used.
NFR_Target_Audience	Medium	The system should be easy for users of all ages to use, understand and interact with.
NFR_New_Users	High	The game must be easy for users with no prior experience to use, understand and interact with.
NFR_Access	Medium	The system should be operable by users who are colour blind or have impaired vision.
NFR_Metrics	High	Tracked metrics such as satisfaction and money must be updated within 1 second of action by user/event.
NFR_Pause_Menu	High	The game's pause menu must appear within 1 second of the pause button being pressed.
NFR_End_Screen	High	The system's end screen must be displayed within 2 seconds of the 5 minute timer ending.
NFR_Buildings	High	Buildings must start to be placed, destroyed or moved within 1 second of the user's interactions with the game.
NFR_Timer	High	The system should ensure that the game lasts five minutes.

Requirements References

- [1] I. Sommerville, *Software Engineering*, 10th ed. Harlow(England): Pearson Education Limited, 2015, pp. 115, 116. Accessed: Oct. 30, 2024. [Online]. Available: <https://ebookcentral.proquest.com/lib/york-ebooks/reader.action?docID=5185655&ppg=9>
- [2] D. Radigan, "Agile Requirement Documents: Your Product Blueprint," *Atlassian*. <https://www.atlassian.com/agile/product-management/requirements> (accessed Oct. 30, 2024).
- [3] D. Natani, "The Art of Writing Good Requirements," *Atlassian Community*, Sep. 15, 2020. <https://community.atlassian.com/t5/Jira-articles/The-art-of-writing-good-requirements/ba-p/1482103> (accessed Oct. 30, 2024).
- [4] J. Bartlett, "How to Write Good Requirements (With Example)," *TestLodge Blog*, Aug. 28, 2023. <https://blog.testlodge.com/writing-good-requirements/> (accessed Oct. 30, 2024).
- [5] INCOSE, M. Ryan, and L. S. Wheatcraft, "INCOSE Guide to Writing Requirements v3.1 – Summary Sheet," INCOSE-TP-2010-006-03.1, Apr. 2022, pp. 3. Accessed: Oct. 30, 2024. [Online]. Available: https://www.incose.org/docs/default-source/working-groups/requirements-wg/rwg_products/incose_rwg_gtwr_summary_sheet_2022.pdf?sfvrsn=a95a6fc7_2
- [6] E. Himes, "8 Tips for Writing Better Requirements," *PTC*, Sep. 25, 2024. <https://www.ptc.com/en/blogs/alm/8-tips-for-writing-requirements> (accessed Oct. 30, 2024).