

Requirements

Cohort 3 Team 1: Pixels of Promise

Movitz Aaron, Jake Adams, Oliver Belam, Anna Burt, Emily Foran, Sky
Haines-bass, Job Young

University of York

ENG1: Software & Systems Engineering

November 11, 2024

2 Requirements

2.1 Introduction

a) Write a succinct introduction explaining how requirements were elicited and negotiated, and why they are presented as they are. Your submission should evidence research into requirements specification and presentation.

The first step we took towards eliciting the various user and system requirements was conducting a structured interview with a mixture of open and close-ended questions with our stakeholder, Tommy. This was with the intention of expanding our knowledge of the product brief and gaining clarification on points we deemed unclear. Beforehand, we held a team meeting in which we went through the assessment document and product brief, discussing the features as an overview, then focusing on elements we felt could be expanded upon. Based on this, we drafted a set of interview questions ranging from inquiring about the game's target audience, to clarification on the student satisfaction metric. We subsequently streamlined this to condense it down to only those relevant to the requirements section, allowing us to efficiently gather relevant information in the limited time that we had during our initial interview.

After interviewing Tommy, we began eliciting the requirements upon analysis of the responses given. Throughout which, our team maintained discussions regarding the project constraints, particularly those identified within the stakeholder meeting. For example, we discussed aspirational design plans for the project, which the stakeholder deemed outside of the project scope. Upon this, we refined the project plan and associated requirements to be more realistically achievable within the initial project deadline of November 11th.

We decided to present these in three distinct tables: user requirements, functional requirements and non-functional requirements. We decided to separate them into these categories and utilise a unique ID referencing system as it improves readability, with the aim of streamlining the process of checking our implementation fulfils them later on in the project. This structure will also facilitate easy modifications or additions to the requirements if anything is identified during a future stakeholder meeting.

2.2 User & System Requirements

b.) Give a systematic and appropriately-formatted statement of user and system requirements

2.2.1 Single Statement Of Need (SSON)

The game will allow players to design their own university campus, with the goal of maximising student satisfaction.

2.2.2 User requirements:

ID	Description	Priority
UR_BUILD	The game shall allow the user to place and manage several different building types.	Shall
UR_LIMIT	The system shall limit the number of buildings the user is able to place	Should
UR_USABILITY_PROMPT	The game shall contain an instruction/prompt to aid with usability.	Shall
UR_SATISFACTION_METRIC	The game shall track student satisfaction so the user understands how well they are doing in the game based on their response to planned or unplanned events.	Shall
UR_UX	The game shall provide an intuitive interface for the user to interact with the game	May

2.2.3 Functional requirements table:

ID	Description	User requirements
FR_BUILD_ACADEMICS	The game shall allow the user to place at least one building for academics.	UR_BUILD
FR_BUILD_ACCOMM	The game shall allow the user to place at least one building for students to live and sleep.	UR_BUILD
FR_BUILD_EATERY	The game shall allow the user to place at least one building where students can eat	UR_BUILD
FR_BUILD_RECREATIONAL	The game shall allow the user to place at least one building where students can participate in recreational activities.	UR_BUILD
FR_BUILDING_PLACEMENT	The system will allow the player to place buildings on the map and change their position throughout the game. (providing map constraints)	UR_BUILD

FR_BUILDING_DELETION	The system will allow the player to delete and remove buildings that were placed on the map.	UR_BUILD
FR_TIME_SIMULATION	The game shall simulate progression throughout the whole 5 minutes.	UR_USABILITY_PROMPT
FR_GAME_PAUSE	The game shall allow players to pause and resume gameplay whenever they wish.	UR_USABILITY_PROMPT
FR_HINTS	The system will give tips to players throughout the game when requested, to <u>support</u> efficient building placement.	UR_USABILITY_PROMPT
FR_COST	The system shall <u>limit</u> the number of buildings the user is able to place through a budget / currency system	UR_LIMIT
FR_CURRENCY	The system shall calculate and accordingly manage user currency, decreasing it appropriately upon building new places and increasing it appropriately according to student satisfaction and building demolition refund amounts.	UR_LIMIT
FR_MAX_BUILDINGS	The system shall <u>limit</u> the number of buildings <u>of a specific type</u> the user is able to place by not allowing the user to place any more than the building type's specified maximum number , which will differ based on demand for the resource each building provides.	UR_LIMIT
FR_SATISFACTION_CALC	The system shall calculate student satisfaction according to building placement.	UR_SATISFACTION_METRIC
FR_EVENT_AMOUNT	The system shall include a minimum of three events which happen over the course of the game.	UR_SATISFACTION_METRIC
FR_EVENT_RESPONSE	The game shall allow the user to respond to planned and unplanned events, providing players with response options, overall impacting satisfaction.	UR_SATISFACTION_METRIC
FR_EVENT_TRIGGER	The system should include a triggering mechanism for the	UR_SATISFACTION_METRIC

	various types of events (negative, positive, or neutral), whether it be the action of the user or designated increments in the 5-minute game timer	
FR_EVENT_TYPE	The system can either present positive events that increase the student satisfaction level, negative events which need to be mitigated against, or neutral, which could trick the user as mattering	UR_SATISFACTION_METRIC

2.2.4 Non-functional requirements table:

ID	Description	User requirements	Fit criteria
NFR_ADAPT	The system should handle the increasing complexity of the campus design in an efficient manner, while providing feedback to the user	UR_SATISFACTION_METRIC	There should be no increase in visible system lag upon approaching and reaching the building placement limit.
NFR_PLACEMENT	The system should render and register buildings when placed and managed with minimal lag, maintaining responsiveness.	UR_BUILD	The user should not have to wait more than 1.5 seconds for a placed building to render.
NFR_PERFORMANCE	The system should load the initial game state within seconds, keeping consistent frame rate throughout the playability time.	UR_UX	The FPS should be maintained at about 45fps throughout the duration of the game.
NFR_ERROR_HANDLING	The system should handle unexpected user errors or failures. This includes use-friendly error messages, logging errors, and overall causing minimal impact to gameplay.	UR_USABILITY_PROMPT	The system should crash <5% of the time, and should display an appropriate prompt concisely explaining why what the user did is not allowed.
NFR_INTUITIVE	The system should be intuitive and easy to understand and use without training.	UR_UX	100% of prompts and instructions shown should not contain technical language.