

# Requirements Elicitation

Cohort 3 Team 5 - alltheeb5t

Aaron Heald

Alex Gu

Arun Hill

Jade Stokes

Maksim Soshchin

Meg Tierney

Will Hall

## Introduction

We started by looking at the assessment brief and identifying all the key tasks and objectives that our game must fulfil. Our goal was to make sure that the game met every criterion that was outlined in the brief. From this, we wrote down a list of questions to ask our client which both clarified the brief and allowed us to gain feedback on our own ideas. For example, we were unsure on how the student satisfaction was going to be measured and what devices our game would be running on.

Next, we conducted research into what the best practices are for eliciting and documenting requirements. This led to an exploration of writing a requirements specification - the industry standard for presenting requirements. We decided to split it into three categories - user, functional, and non-functional requirements. Which allows us to easily cover each aspect of the system ensuring a balanced product.

We put ourselves in the shoes of the customer by considering their perspective on what the game should achieve and how it would fulfil the requirements. We went through each point of the brief and brainstormed specific features the game would need to include. For example, we considered things that could potentially be limitations of placing buildings into the map, like placing them too close together or too close to a dangerous place like a lake. This detailed brainstorming allowed us to filter each question into one of the three categories we created, whilst still covering the whole brief.

Now that we had a list of questions, we developed a scenario walking through the game from the user's perspective, visualising how the player would interact with the interface. This helped us outline the game's flow, features and key interactions. During the interview with our prospective client, we presented this walkthrough, to ensure that our requirements aligned. As we described the game's features, he provided us feedback, pointing out areas that needed adjustment. We then refined the requirements based on his input, ensuring that the game aligned with their expectations.

During the interview we also received a few requirements from our customer such as; making sure that changing the resolution of the game doesn't cause errors, including a basic tutorial and including more accessibility settings such as colour blind support.

After taking over this game during the second half of the project, new requirements were given within the brief for the game, meaning new system requirements needed to be outlined for the development and architecture teams. We decided not to conduct a stakeholder interview for this part of the project due to time constraints and already having asked about some of the full brief in our own group's initial interview (mainly concerning events).

We also went through each current requirement and decided if it needed to be updated or reworked, as well as deciding if any new requirements had to be added to fit the current state of the game. These changes included moving certain functional requirements to be non-functional or vice versa if they fit those categories better (e.g. **FR\_ACCESSIBILITY** becoming **NFR\_BUTTONS**), merging similar requirements together to reduce confusion, and adding extra requirements like **UR\_FINANCE** to reflect both the current code and what our group wanted from this feature of the game.

Definitions:

- \* **New Requirements**
- \*\* **Updated Requirements**
- **UR: User Requirement**
- **FR: Functional Requirement**
- **NFR: Non-Functional Requirement**

## User Requirements

Requirement ID	Description	Priority
UR_EXPERIENCE	The user shall have a pleasant experience when playing the game.	Shall
UR_BUILDINGS	The user shall be able to place a variety of different buildings.	Shall
UR_ACCESSIBILITY	The user shall be able to use the game without needing any training, and the game shall be playable for users with colour blindness.	Shall
UR_SOUND	The user shall be able to listen to or mute enjoyable in-game music.	Shall
<b>UR_LEADERBOARD**</b>	The user shall be able to save their final satisfaction score and see the top 5 saved scores on a leaderboard.	Shall
UR_SYSTEM	The user shall be able to play the game in university software labs with no additional hardware.	Shall
UR_TIME	The game shall last only 5 minutes.	Shall
UR_MAP	The user shall be able to navigate a fixed size map where they can view and place all buildings.	Shall
<b>UR_MENU*</b>	The user shall be able to interact with several menu screens throughout the game.	Shall
UR_COUNTER	The user shall be able to see a count of the number of buildings they have placed.	Shall
<b>UR_EVENTS*</b>	The user shall be able to experience random events in the game that will affect their playthrough differently.	Shall
<b>UR_SATISFACTION*</b>	The user will be able to increase or decrease a metric for student satisfaction which will influence their final score.	Shall
<b>UR_FINANCE*</b>	The user will be able to gain money as the game progresses that will allow them to build more things on campus.	Shall

<b>UR_ACHIEVEMENTS</b> *	The user shall be able to unlock several achievements within the game based on their actions when playing.	Shall
-----------------------------	--	-------

## Functional Requirements

Requirement ID	Description	User Requirement
FR_BUILDING_TYPE	Every building should have a single purpose such as teaching, food, etc.	UR_BUILDINGS
FR_CONTROLS	The system shall be operated by using a standard mouse and keyboard.	UR_SYSTEM
FR_SOUND	The system should have a sound system integrated with the ability to change the sound files.	UR_SOUND
FR_MUTEABLE	There should be a mute button so that the game can be played without any sound playing.	UR_SOUND
FR_SCALING	The system shall be able to scale to small and large screens with low hardware requirements.	UR_SYSTEM
FR_TIMER	The system shall be able to keep track of the time of the game and end when the timer has finished.	UR_TIME
FR_MAP	The system must support a map that can fit all required buildings for the player to fulfil objectives.	UR_MAP
<b>FR_SATISFACTION*</b>	The system shall keep track of student satisfaction as a percentage and update based on the environment, displaying this to the player.	UR_SATISFACTION
<b>FR_PAUSE*</b>	The system shall allow the user to pause the game timer and view a menu of options at any point.	UR_MENU, UR_TIME
<b>FR_TUTORIAL*</b>	The system shall display a brief introduction of the game with controls to the user upon starting a new playthrough	UR_MENU
<b>FR_NEW_MONEY*</b>	At the beginning of each in-game semester the system will give the user a lump sum of money to spend on new buildings.	UR_TIME, UR_FINANCE
<b>FR_BUILD_TIME*</b>	The system shall take a set amount of time to 'build' a new building.	UR_BUILDINGS
FR_ACCOMMODATION_BUILDING	There shall be at least 1 placeable accommodation building.	UR_BUILDINGS
FR_LEARNING_BUILDING	There shall be at least 1 placeable building for students to learn in.	UR_BUILDINGS
FR_EATING_BUILDING	There shall be at least 1 placeable building for students to eat in.	UR_BUILDINGS
FR_RECREATIONAL_BUILDING	There shall be at least 2 placeable buildings for recreational activities.	UR_BUILDINGS

<b>FR_NO_OVERLAP</b>	The system should not allow buildings to be placed on top of one another, or on squares that are off limits (e.g. in a lake)	UR_BUILDINGS
<b>FR_INCOME*</b>	The system shall give users a small amount of money over time for each Eating Building built.	FR_BUILDINGS, FR_COUNTER, FR_FINANCE
<b>FR_COUNT*</b>	The system shall keep track of the total number of buildings currently on the map for the user to see.	UR_BUILDINGS, UR_COUNTER
<b>FR_LEADERBOARD**</b>	The system shall have an option to display a leaderboard to see the 5 previous top scores.	UR_LEADERBOARD
<b>FR_SAVES**</b> (NFR_SAVES)	The system shall save scores of players playing the game, given that they are higher than any of the scores on the leaderboard. A player's previously attained achievements should also be included.	UR_LEADERBOARD, UR_ACHIEVEMENTS
<b>FR_EVENT_GENERATOR*</b>	The system will randomly choose an event to occur 1 time a year	UR_EVENTS
<b>FR_EVENT_RESULT*</b>	The system will assign a positive, negative or neutral change to student satisfaction and balance after an event has occurred.	UR_EVENTS, UR_SATISFACTION, UR_FINANCE
<b>FR_EVENT_DISPLAY*</b>	The system will display a brief description of the event that is currently occurring and the effect of it on satisfaction and finances.	UR_EVENTS
<b>FR_EVENT_CHOICE*</b>	For certain events the system will allow the user to choose between two options to affect their game.	UR_EVENTS
<b>FR_ACHIEVEMENT_MENU*</b>	The system shall have a menu that the player can view containing all possible achievements in game.	UR_ACHIEVEMENTS
<b>FR_STRIKE_EVENT*</b>	The system shall display a negative event where lecturers go on strike. Any buildings being built will pause until the strike ends and satisfaction will decrease.	UR_EVENTS UR_BUILDINGS UR_SATISFACTION
<b>FR_STRIKE_CHOICE*</b>	The system shall give the user a choice to end the strike event early by paying the lecturers more money (and buildings will cost more to maintain) or letting it run its course which decreases satisfaction until it ends.	UR_EVENTS UR_BUILDINGS UR_FINANCE UR_SATISFACTION
<b>FR_BUS_CHANGE_EVENT*</b>	The system shall display a negative event where bus routes around the university have changed, decreasing satisfaction.	UR_EVENTS UR_SATISFACTION
<b>FR_FEE_INCREASE_EVENT*</b>	The system shall display a neutral event where tuition fees increase for students. This will increase money gained but decrease satisfaction.	UR_EVENTS UR_FINANCE UR_SATISFACTION
<b>FR_SPONSOR_EVENT*</b>	The system shall display a positive event where the university gains a sponsor, allowing the user to build one building for free.	UR_EVENTS UR_BUILDINGS
<b>FR_ALUMNI_EV</b>	The system shall display a positive event where the	UR_EVENTS

<b>ENT*</b>	university is given a donation by an alumni, gaining the user a lump sum of money.	UR_FINANCE
<b>FR_FLOOD_EVENT*</b>	The system shall display a negative event where a flood occurs, destroying one building on the campus.	UR_EVENTS UR_BUILDINGS
<b>FR_FIRE_EVENT*</b>	The system shall display a negative event where one building catches on fire. If the user does not click on the building within 2 seconds the building will burn down, decreasing satisfaction, however the user will receive insurance money.	UR_EVENTS UR_BUILDINGS UR_TIME UR_SATISFACTION UR_FINANCE
<b>FR_AWARD_EVENT*</b>	The system shall display a neutral event where a student wins an award. Nothing changes.	UR_EVENTS
<b>FR_ROSES_EVENT*</b>	The system shall display a positive event where the university competes in Roses. Buildings will generate double income during this event.	UR_EVENTS UR_FINANCE
<b>FR_ROSES_RESULT_CALCULATION*</b>	If the user has built more than 5 recreational buildings by the end of roses, the system shall give the user a 50% chance of winning. Otherwise, the chance is 1 in 3.	UR_EVENTS UR_BUILDINGS UR_COUNTER
<b>FR_ROSES_FINAL*</b>	The system shall display whether the user has won or lost Roses at the end of the in-game year. If the user wins, satisfaction increases, if they lose it decreases.	UR_EVENTS UR_SATISFACTION
<b>FR_MFA_UNLOCK*</b>	The system shall unlock the achievement "Mike Freeman Award" when student satisfaction is above 70% for 3 minutes straight. The system shall begin a separate timer when this occurs.	UR_ACHIEVEMENTS, UR_TIME, UR_SATISFACTION
<b>FR_BM_UNLOCK*</b>	The system shall unlock the achievement "Bare Minimum" if the user only places 1 of each building type throughout the game.	UR_ACHIEVEMENTS, UR_COUNT
<b>FR_PRIORITIES_UNLOCK*</b>	The system shall unlock the achievement "Priorities" if their study building count is double that of their recreational buildings after 20 buildings have been placed.	UR_ACHIEVEMENTS, UR_COUNT
<b>FR_ITAU_UNLOCK*</b>	The system shall unlock the achievement "Is This A University?" if their recreational building count is double that of their recreational buildings after 20 buildings have been placed.	UR_ACHIEVEMENTS, UR_COUNT
<b>FR_UNLUCKY_UNLOCK*</b>	The system shall unlock the achievement "Unlucky" if the player encounters 3 negative events in one game.	UR_ACHIEVEMENTS, UR_EVENTS
<b>FR_LUCKY_UNLOCK*</b>	The system shall unlock the achievement "Lucky" if the player encounters 3 positive events in one game.	UR_ACHIEVEMENTS, UR_EVENTS
<b>FR_INDECISIVE_UNLOCK*</b>	The system shall unlock the achievement "Indecisive" if more than 20 buildings are deleted throughout the game.	UR_ACHIEVEMENTS, UR_BUILDINGS
<b>FR_CS_UNLOCK*</b>	The system shall unlock the achievement "Clean Slate" if all buildings are removed on campus after the player has built a total of 10 buildings.	UR_ACHIEVEMENTS, UR_BUILDINGS, UR_COUNT

<b>FR_SAVIOUR_UNLOCK*</b>	The system shall unlock the achievement "Saviour" if the player saves a building during the fire event.	UR_ACHIEVEMENTS, UR_EVENTS
<b>FR_BUSY_UNLOCK*</b>	The system shall unlock the achievement "Busy Campus" if 40 buildings have been built	UR_ACHIEVEMENTS, UR_COUNT
<b>FR_CHANGE_UNLOCK*</b>	The system shall unlock the achievement "Master of Change" if student satisfaction is under 30% for the first 3 minutes of the game but the player manages to win.	UR_ACHIEVEMENTS, UR_SATISFACTION
<b>FR_REALISTIC_UNLOCK*</b>	The system shall unlock the achievement "Realistic" if finances go below -24000.	UR_ACHIEVEMENTS, UR_FINANCE
<b>FR_GAME_END*</b>	The system shall end the game after 5 minutes.	UR_TIME
<b>FR_END_SCREEN*</b>	The system shall display the final satisfaction score to the player at the end of the game.	UR_SATISFACTION, UR_MENU
<b>FR_WIN*</b>	The system shall show the player whether they have 'won' the game if their satisfaction is above 50% at the end of the game.	UR_SATISFACTION

## Non-Functional Requirements

Requirement ID	Description	Appropriate fulfilment of criteria	User Requirement
NFR_EASE_OF_USE	The system shall be operable by A-level students with no training.	First time users can >70% of the time play the game without asking questions.	UR_ACCESSIBILITY
<b>NFR_SYSTEM_REQUIREMENTS**</b> (FR_SYSTEM_REQUIREMENTS)	The system shall be usable on a big screen with a software lab pc (integrated intel graphics, w11 and 16gb of ram)	The game runs without any noticeable lag on >95% of playthroughs.	UR_SYSTEM
<b>NFR_BUTTONS**</b> (FR_ACCESSIBILITY)	Buttons must have clear descriptions and easy to understand Bugs Identified functions.	>95% of users will be able to understand the interactive elements of the game without blatant explanation.	UR_ACCESSIBILITY
NFR_COLOUR	The system shall use a colour scheme that wouldn't be confusing to colour blind people.	Colour blind users can successfully play the game without needing help.	UR_ACCESSIBILITY
<b>NFR_METRICS*</b>	The system will update metrics such as student satisfaction and finances consistently.	Metrics are updated on the screen within a second of changing.	UR_FINANCE, UR_SATISFACTION
NFR_FAST_PLACEMENT	Placed buildings will appear instantly on the map.	Buildings appear <2 seconds after placing.	UR_BUILDINGS