Requirements

Cohort 2- Group 1

Members

Ollie Barlow (vrq506)
Jacob Barnes (bdp514)
Zayaan Bawany (fst508)
Lily Boldbaatar (jkk522)
Georgia Briggs (rxm503)
Evelyn Gravett (qnf502)
Hassan Yunus (wfv505)

These requirements have been determined following a series of group meetings, as well as our main client meeting. For clarity, these have been separated into 4 individual subheadings: User Requirements, System Requirements, Functional Requirements, and Non-Functional Requirements. Across the four sections, key requirements are explained in the description below:

User Requirements

With the research conducted, we were able to label out 12 requirements. This was largely based on the 3 primary objectives listed in the Product Brief [3]. ‘Ensure the campus is not just functional but also intuitive and fun for students to be at’. This objective meant that UR_UX and UR_UI, focused around user experience, was a standard requirement and therefore we chose not to provide much discussion time to this, as we all had the same goal in creating a pleasant experience. ‘Ensure you make a campus with what is needed for it to operate correctly’, meant that the same discussion consideration could be applied to UR_BUILDINGS. While aware of the game time limit of 5 minutes [3], we thought providing the user with a pause option would allow them to understand exactly what was happening in the game. During our research, we came across a university simulation game, ‘Model Uni: My First Uni’ [2]. This simulation provided a basic uni-sim experience, allowing us to identify issues we could come across in the game. We found that it was common for the user to get overwhelmed with how fast the game would operate, leading to student satisfaction issues as users would not react as fast as the game would require when at full speed. Therefore, we felt it was appropriate to implement UR_PAUSE, but also UR_DISPLAY_TIMER and UR_BUILDING_COUNTER to give the user a clearer understanding of what was happening in the game. Furthermore, we decided a percentage metre to determine student satisfaction was most appropriate, therefore implementing UR_SATISFACTION_METRE. In order to produce a game suitable for everybody, we also suggested providing the user with a set of accessibility options, allowing them to modify some aspects of the game to make it an enjoyable experience, in line with both the first and second primary objectives.

Functional Requirements

As identified in the product brief, the goal of the game is to achieve the best student satisfaction possible. This meant that the placement of buildings and overall design of the university had to play as a factor in the game. These functionalities were also labelled in accordance to the specific ‘user requirements’ they amend to. As the objective of the game is to achieve the highest satisfaction level possible, and as the satisfaction level was determined by a percentage, a greater focus had to be placed on limiting factors to make it harder for the user. An example of this was FR_CONSUTRCTION_SATISFACTION_LOSS which creates a dilemma for the user if they choose to upgrade buildings during term time. While an upgrade could mean greater satisfaction in the long run, this could severely impact student satisfaction if they are unable to use those buildings. This raises the question on what factors determine this satisfaction percentage and how impactful each factor is. Therefore, we chose to create a menu display under FR_SATISFACTION_BREAK_DOWN which allows the user to understand exactly what is affecting the satisfaction.

Non-Functional Requirements

Alongside factors that were dependent on user input and therefore impacted the satisfaction rate, we determined a set of factors that would impact user experience. While this may not directly affect the game, it could affect user satisfaction as they may not find the game enjoyable if the user experience is poor. Therefore, a good

User Requirements

ID	Description	Priority
UR_UX	The user experience is pleasant	Shall
UR_BUILDINGS	The user can place buildings	Shall
UR_UI	The user interface is clear and pleasant to look at	Shall
UR_DISPLAY_TIMER	The game shall display a visible timer to the user to keep track of time	Shall
UR_BUILDING_COUNTER	The game shall display a counter showing the number of each type of building placed	Shall
UR_PAUSE	The user can pause the game at anytime	Shall
UR_UPGRADE	The user can upgrade buildings	Maybe
UR_SATISFACTION_METRE	The system shall display the student satisfaction, which then has its change broken down.	Shall
UR_ACCESSABLITY_SETTINGS	The user choose from some accessibility settings	Should
UR_EVENTS	The user will get a pop up explaining the current event	Shall
UR_KEYBINDS	The user can use keybinds to quickly perform an action	Maybe

System Requirements
Functional Requirements

ID	Description	User Requirements
FR_SYS_REPONSE	The system responds to user inputs clearly	UR_UI
FR_BUILDING_PLACE	The system allow the user to place buildings in	UR_BUILDINGS

MENT	buildable areas only	
FR_BUILDING_SATISFACTION_GAIN	After a building is built it will increase the satisfaction for a period of time	UR_BUILDINGS
FR_CONSTRUCTION_SATISFACTION_LOSS	Building/upgrading a building during term time will result in satisfaction loss for a period of time	UR_BUILDINGS UR_UPGRADE
FR_TIMER_DISPLAY	The game shall implement a timer counting down from 5 minutes in real time, allowing the user to monitor the time elapsed	UR_DISPLAY_TIMER
FR_BUILDING_COUNTER	The game shall include a counter that updates whenever a building is placed, showing the number of each type	UR_BUILDING_COUNTER
FR_SATISFACTION_BREAK_DOWN	The system shall display what is affecting the student satisfaction, when the satisfaction metre is clicked on	UR_SATISFACTION_METRE
FR_BUILDING_OVERLAP	The system shall not allow a building to be placed if it will overlap with another building	UR_BUILDINGS
FR_BUILDING_CONSTRUCTION	Buildings take time to be constructed before they can have a positive satisfaction gain	UR_BUILDINGS
FR_SATISFACTION_TRACK	The system shall display where the satisfaction will end up over time.	UR_SATISFACTION_METRE
FR_COLOUR_BLIND_MODE	The system shall allow the user to select if they want a colour blind mode on.	UR_ACCESSABILITY_SETTINGS
FR_MOUSE_SENSITIVY_SLIDER	The system shall allow the user to choose a mouse sensitivity	UR_ACCESSABILITY_SETTINGS
FR_RANDOM_EVENT	When there is not an event cooldown, a random event can happen from a set list.	UR_EVENT
FR_SET_EVENT	If the events are not on cooldown, then at a certain time a set event will happen	UR_EVENT
FR_SATISFACTION_PAUSE	When the game is paused, the satisfaction will not change.	UR_PAUSE
FR_TIMER_PAUSE	When the game is paused, the timer will not change	UR_PAUSE
FR_MULTIPLE_BUILDINGS	When there are multiple buildings being built/upgraded at once, there is an increasing multiplier on the satisfaction modifier from the building's construction during term time.	UR_BUILDINGS
FR_UPGRADE_SATISFACTION_GAIN	When a building’s upgrade has finished construction, satisfaction will increase for a	UR_UPGRADE

	period of time.	

Non-Functional Requirements

ID	Description	User Requirements	Constraints
NFR_SYS_REPONSE_SPEED	The system responds to user inputs quickly	UR_UX	In <0.1s from time of input 90% of the time
NFR_COUNTER_VISIBILITY	The building counter shall update in real time whenever a building is placed	UR_BUILDING_COUNTER	Updates within 1 second of placement
NFR_TIMER_VISIBILITY	The timer shall be visible and readable at all time during gameplay	UR_DISPLAY_TIMER	Timer visibility is ensured throughout the play
NFR_SATISFACTION_MULT	Buildings satisfaction gain/loss shall be multiplied depending on vicinity to accommodation	UR_BUILDINGS	*Put multiplier here when decided*
NFR_SATISFACTION_METRE_VISIBILTY	The satisfaction metre shall be visible and readable at all times.	UR_SATISFACTION_METRE	Satisfaction metre is visible 99% of the time
NFR_READABLITILY	All user-facing messages shall be in plain-english and without technical jargon	UR_UX	This should be the case 95% of the time
NFR_EVENT_COOLDOWN	There shall be period where an event can not happen after an event has happened	UR_EVENTS	The cooldown shall be 1 minute.
NFR_MULTIPLE_CONSTRUCTION	There shall be a limit on the amount of buildings that can be built/upgraded at once.	UR_BUILDINGS	The limit of 3 buildings.