REQUIREMENTS

Group 3

Liam Martin

Aaliya Williams

Lucy Crabtree

Kai Nichol

Sammy Hori

Tim Gorst

Zac Ribbins

Our team approached the elicitation and negotiation of requirements systematically to ensure alignment with both the project brief and customer expectations. Upon receiving the project brief, we analysed and discussed potential game ideas, styles and functionality. We then booked a meeting with our customer, in which we could ask questions and discover any functionality or requirements that were unmentioned within the brief. To prepare for the customer meeting we created a set of questions to ask in order to elicit the requirements. These questions were separated into different sections creating a more organised meeting, allowing for questions to flow naturally. The requirements were negotiated with our customer to prioritise functionality within a realistic timeframe and provide ourselves with creative freedom, for design elements and fine tuning for functionality. Whilst also ensuring the customer's expectations are met and they are satisfied.

After eliciting the requirements from the customer, we began researching to find the most suitable method for presenting our requirements in. Our requirements are presented in this way as by having them in a clear and concise manner, will help us ensure that the product fulfils all elicited requirements and the product functionality performs as expected. The requirement should correctly define everything that is necessary for our design and the functionality the product will perform, from basic system requirements to the more complex user requirements.

Following the elicitation phase, our team conducted research to determine the most suitable method for presenting the requirements. Our chosen approach focuses on clarity and conciseness to facilitate comprehensive understanding and implementation. We opted for a structured format, inspired by the Software Requirements Specification (SRS) document [1], which is widely recognised for its effectiveness in conveying critical project details. This format includes an introduction, a general description of the software, and specific requirement specifications, ensuring comprehensive coverage of system and user needs.

We opted for the MoSCoW method [2] to prioritise the importance of each requirement. To enhance clarity and accessibility, we structured the requirements in a tabular format. This approach improves readability and adaptability, facilitating alignment with stakeholders' goals. Each requirement, identified by a unique ID, comprehensively covers various aspects of the game.

Throughout the development process, these requirements have been consistently reiterated and refined. They now encompass every essential element of the game, ensuring thorough coverage and alignment with project objectives.

User Requirements

ID	Description	Priority
UR_GAME_INTERFACE	The game interface shall be simplistic, realistic, and easy to understand, so anybody can pick up the game and play quickly	Must
UR_STYLE	The style and theme presented to the user shall be neutral or humorous.	Must
UR_UX	The game can be played on various screen resolutions.	Must
UR_AUDIENCE	The game is target towards people thinking of attending university or people at university	Must
UR_REAL_TIME	The game will take around 5-10 minutes for someone who is familiar with the game to complete	Must
UR_OBJECTIVE	The games objective is to complete activities by interacting with the map through the game to improve the score they will receive at the end of the game	Must
UR_PLATFORM	The game shall be playable on a range of platforms such as desktops and laptops	Must
UR_CONTROLS	The game should have recognisable and standard controls (WASD)	Must
UR_MOVEMENT	Users shall be able to navigate round the map	Must
UR_CHOICES	Users shall be able to choose the activities they want to spend the day doing	Must
UR_SCORE	Users shall be able to see the score they got from playing the game	Must
UR_GAME_STATS	Users shall be able to tell some game stats such as energy, time, and day	Must
UR_CUSTOMISATION	Users should be able to select their character model	Should
UR_PREFERENCES	Users should be able to change settings to their liking for example music volume	Should
UR_GAME_OPTIONS	Users can interact with the game state such as starting and exiting	Must

UR_LEADERSHIP	Users shall see a leaderboard at the end of the game with the names and scores of the top ten users	Must
UR_ACHIEVEMENTS	Users shall see the achievements they earned throughout gameplay at the end of the seven days	Must

Functional Requirements

ID	Description	User Requirements	Priority
FR_GAME_START	The system shall allow the user to start the game from the menu screen	UR_GAME_OPTIONS	Must
FR_GAME_QUIT	The system shall allow the user to quit the game from the menu screen	UR_GAME_OPTIONS	Must
FR_GAME_END	The system shall end the game when the user has the character sleep on the 7th day	UR_GAME_STATS	Must
FR_GAME_END_STATS	The system shall show the users the amount of activities they have completed on the end screen	UR_SCORE, UR_GAME_STATS	Must
FR_CHARACTER_SELECTION	The system should allow the user to select from a range of characters at the start of the game	UR_CUSTOMISATION	Should
FR_CHARACTER_MOVEMENT	The system shall allow the user to move their character around the map	UR_MOVEMENT	Must
FR_CHARACTER_COLLISION	The system shall never allow the user to move the character through collidable objects	UR_MOVEMENT	Must
FR_CHARACTER_INTERACTION	The system shall allow the user to interact with the map	UR_CONTROLS, UR_OBJECTIVE	Must
FR_SCREENS	The system shall show the user different screens relating to their current stage in the game (start, settings, maps, end)	UR_GAME_INTERFACE	Must
FR_MAP	The system shall show user the map the character is currently on	UR_MOVEMENT	Must
FR_MAP_BUILDINGS	The system shall include common Heslington East campus buildings on the map for the user to enter and a map for each building	UR_GAME_INTERFACE	Must
FR_MAP_TRANSITION	The system shall allow the user to transition between buildings by walking to the doors	UR_GAME_INTERFACE, UR_MOVEMENT	Must
FR_ACTIONS	The system shall allow the user to complete actions	UR_CHOICES	Must

	throughout the day		
FR_ACTIONS_INDICATION	The system should indicate when an action has or hasn’t been completed	UR_GAME_INTERFACE	Should
FR_ACTIONS_TYPES	The system shall contain different types of actions for the user to complete (study, eat, relax, sleep) (one of each)	UR_CHOICES	Must
FR_ACTIONS_INSUFFICIENT	The system shall never allow the user to complete an action if they don’t have a sufficient amount of any resource (time, energy).	UR_CHOICES	Must
FR_STATS	The system shall keep track of the games stats	UR_SCORE, UR_GAME_STATS	Must
FR_STATS_UPDATE	The system shall update the stats when an action has occurred	UR_GAME_STATS	Must
FR_STATS_RESET	The system shall reset some stats at the end of each day (energy, time)	UR_GAME_STATS	Must
FR_STATS_SHOW	The system shall show the stats for the day to the user so they can keep track of them	UR_GAME_STATS, UR_GAME_INTERFACE	Must
FR_LEADERBOARD	The system shall keep track of all users who have successfully completed the game and show stats of the top ten users	UR_SCORE,UR_LEADERSHIP	Must
FR_ACTIVITY_COUNTER	The system shall keep track of the number of individual activities the user has completed	UR_ACHIEVEMENTS	Must
FR_ACTIVITY_STREAK	The system shall award the user a streak for each activity they complete multiple times throughout the duration of the game	UR_ACHIEVEMENTS	Must
FR_MUSIC	The system should adjust the volume of the background music based on user input	UR_PREFERENCES	Should

Non-Functional Requirements

ID	Description	Fit Criteria	User Requirements	Priority
NFR_SIMPLE_INTERFACE	The system shall have a simple interface which is easy to understand	Understandable in <5 seconds	UR_GAME_INTERFACE	Must
NFR_GAME_DESIGN	The games visuals shall be mostly realistic with some artistic interpretation so that it remains clear	User can complete a task without needing help	UR_GAME_INTERFACE	Should

NFR_GAME_PROGRESSION	Progression through the game will be straight forward and not complicated allowing for the game to be completed in 5-10 minutes	Game length is between 5 and 10 minutes	UR_REAL_TIME	Must
NFR_COMPATABILITY	The game can run on laptops and desktops	The game should run on Windows, Linux and MAC OS	UR_PLATFORM	Must
NFR_SCREEN_SCALE	The game UI adapts to the screen size someone is playing on	The game should be rendered correctly on common screen sizes	UR_UX	Must
NFR_STANDARD_CONTROLS	The game has industry standard controls for instance movement being WASD or arrow keys	Users should be able to figure out the controls in < 5 mins	UR_CONTROLS	Must

Constraint Requirements

ID	Description	Type
CR_DESIGN_TECH	The game design must utilise appropriate technology to meet performance and functionality requirements specified by the stakeholder.	Design
~~TECH_CONSTRAINT_CODE_LANG~~	~~The game shall be coded in Java~~	~~Shall~~
~~TECH_CONSTRAINT_JAVA_VER~~	~~Required to use Java version 11~~	~~Shall~~

References:

[1] GeeksForGeeks, “Software Requirement Specification (SRS) Format,” *GeeksforGeeks*, Jun. 18, 2020. https://www.geeksforgeeks.org/software-requirement-specification-srs-format/ (accessed May 13, 2024).

[2] K. Brush, “What is the MoSCoW Method?,” *TechTarget*, Apr. 2020. https://www.techtarget.com/searchsoftwarequality/definition/MoSCoW-method