

# Requirements

## Cohort 1 Group 1

Charlie Piper, Chris Oulton, Ella Daramola, Dillon Anthony, Kevin Thomas,  
Shirin Sitara Alok Kumar, Tom Haslam

Upon receiving the project brief we read and discussed the contents as a group. Having analysed the brief we began coming up with 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.

The research we undertook to develop our requirements in this way, ranged from previous experience with writing requirements to different methodologies to specify the requirements. We discovered that a software requirements specification (SRS) document is a common way of presenting the requirements. A SRS should answer these critical questions: what should the software do; how should the software behave; what are the performance requirements; and what are the constraints to the system. Our requirements document should contain an introduction, which describes what the software should do and what it should not do; a general description of the software, focussing on the functionality and system and user interfaces; and the specific requirement specification, which gives specific details about each function and outcomes for the product.

User Requirements Table

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	Shall
UR_STYLE	The style and theme presented to the user shall be neutral or humours	Shall
UR_UX	The game can be played on many different screen resolutions	Shall
UR_AUDIENCE	The game is target towards people thinking of attending university or people at university	Shall
UR_REAL_TIME	The game will take around 5-10 minutes for someone who is familiar with the game to complete	Shall
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	Shall
UR_PLATFORM	The game shall be playable on a range of platforms such as desktops and laptops	Shall
UR_CONTROLS	The game should have recognisable and standard controls	Shall
UR_MOVEMENT	Users shall be able to navigate round the map	Shall
UR_CHOICES	Users shall be able to choose the activities they want to spend the day doing	Shall
UR_SCORE	Users shall be able to see the score they got from playing the game	Shall
UR_GAME_STATS	Users shall be able to tell some game stats such as energy, time, and day	Shall
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	Shall

## 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	Shall
FR_GAME_QUIT	The system shall allow the user to quit the game from the menu screen	UR_GAME_OPTIONS	Shall
FR_GAME_END	The system shall end the game when the user has the character sleep on the 7th day	UR_GAME_STATS	Shall
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	Shall
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	Shall
FR_CHARACTER_COLLISION	The system shall never allow the user to move the character through collidable objects	UR_MOVEMENT	Shall
FR_CHARACTER_INTERACTION	The system shall allow the user to interact with the map	UR_CONTROLS, UR_OBJECTIVE	Shall
FR_SCREEN	The system shall show the user different screens relating to their current stage in the game (start, settings, maps, end)	UR_GAME_INTERFACE	Shall
FR_MAP	The system shall show user the map the character is currently on	UR_MOVEMENT	Shall
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	Shall
FR_MAP_TRANSITION	The system shall allow the user to transition between maps by walking to the edge of the map	UR_GAME_INTERFACE, UR_MOVEMENT	Shall
FR_ACTIONS	The system shall allow the user to complete actions throughout the day	UR_CHOICES	Shall
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	Shall
FR_ACTIONS_INSUFFICIENT	The system shall never allow the user to complete an action if they don't have a sufficient amount of any	UR_CHOICES	Shall

	resource (time, energy).		
FR_STATS	The system shall keep track of the games stats	UR_SCORE, UR_GAME_STATS	Shall
FR_STATS_UP DATE	The system shall update the stats when an action has occurred	UR_GAME_STATS	Shall
FR_STATS_RE SET	The system shall reset some stats at the end of each day (energy, time)	UR_GAME_STATS	Shall
FR_STATS_SH OW	The system shall show the stats to the user so they can keep track of them	UR_GAME_STATS, UR_GAME_INTERF ACE	Shall

#### Non-Functional Requirements

ID	Description	Fit Criteria	User Requirements	Priority
NFR_SIMPLE_I NTERFACE	The system shall have a simple interface which is easy to understand	Understandable in <5 seconds	UR_GAME_INTERF ACE	Shall
NFR_GAME_DE SIGN	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_INTERF ACE	Shall
NFR_GAME_PR OGRESSION	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	Shall
NFR_COMPATA BILITY	The game can run on laptops and desktops	The game should run on most common operating systems	UR_PLATFORM	Shall
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	Shall
NFR_STANDAR D_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	Shall

#### Constraint Requirements

TECH_CONSTRAINT_CODE_LANG	The game shall be coded in Java	Shall
TECH_CONSTRAINT_JAVA_VER	Required to use Java version 11	Shall