UNIVERSITY OF YORK

DEPARTMENT OF COMPUTER SCIENCE

# Requirements
# Cohort 2 - Group 16 (Skloch)

## Group Members:

Charlotte MacDonald
Hollie Shackley
Luis Benito
Kaustav Das
Sam Hartley
Owen Gilmore

**Originally by:** Noah Forryan, Lewis Morgan, Naufal Tun Thamanian, Tom Owen, Dan Manby, José Fernandes of Group 19

For our meeting with the client, we split the questions into subsections. These were activities, location, scoring, time/scoring and technical. These distinct sections kept our interview focused and organised, so we didn't miss any questions. We each took a subsection to ask about, but all wrote down the responses. We also made an audio recording so that we could go back and remind ourselves of what the client asked for, in case our notes were not adequate. We focused our questions on the game's aesthetics and user experience, leaving technical details to our team's expertise. For example, we asked about what activities the client wanted, not what buttons they would press to interact with them. Our method for creating the questions was to write down any possible questions we could think of onto a shared document. Then once we did that, we organised them into the sections mentioned above and removed redundant or repeated questions. We removed redundant questions by comparing them to the product brief.

Then, after the requirements from the client were collected we could group them into user, functional system and non-functional system requirements. We felt it important to split the 3 types into different tables, as each one provides its benefit. Identifying functional requirements will make our project "become more predictable" and mean that "problems can be identified sooner"[1]. While the non-functional requirements "ensure that the system is maintainable" and "reduce rework"[2]. Finally, knowing the user requirements allows us to " identify [our] requirements for a system"[3].
These types of requirements were allocated into three corresponding tables. In all tables, there are uniquely identifiable IDs and descriptions of the associated requirements. There is a priority column from the user table. As the game is being developed as low-fidelity (per client confirmation), almost all tasks are essential to ensure the core features are built, and the graphical features are 'should' since the game will work just as well with basic graphics. The score feature is also 'should' since it is not essential to the game, but identified as an interesting feature by the client. Each functional and non-functional requirement has associated user requirements. This allows us to track the user's needs into system requirements. The non-functional requirements table has an additional column called "Fit Criteria". This is important as almost by nature NFRs can be vague and high-level. The fit criteria tell us whether or not our product has met the NFR. When it comes to testing, we will be able to verify that the system achieves the requirements set out by the client using the description column and fit criteria for NFRs. We decided on the NFRs by taking the user requirements and then making the NFRs that are associated. Ensuring we achieve what our client wants us to is vital and the below tables will allow us to do so.

User Requirements

| ID | Description | Priority |
|---|---|---|
| UR_MENU | The system will provide a menu to start the game. | Shall |
| UR_MOVEMENT | The system will allow the player to move around the map. | Shall |
| UR_MAP | The map should have some resemblance to Heslington in real life with recognisable landmarks. | Shall |
| UR_OBJECTS | Throughout the map there Shall be non-playable characters and buildings and other assets for decoration. | Shall |
| UR_INTERACT | The player should be able to interact with some of the objects/buildings/NPCs within the map. | Shall |
| UR_ENERGY | The system provides an energy score, shown as a percentage. It is affected by the user completing activities. | Shall |
| UR_GAME_END | After sleeping on the 7th day, the user will complete the game. They will be informed of their score with the game over screen with a score breakdown. | Shall |
| UR_TIME | The system should display a time of day using a clock and day of the week as a number from 1-7. | Shall |
| UR_COMPATABILTY | The game should run on a desktop. | Shall |
| UR_ANIMATION | The system will have simple animation. | Should |
| UR_GRAPHICS | The display will be 2d pixel art graphics. | Should |
| UR_PAUSE | The system will allow the user to pause the game. It will be shown with a large pause symbol displayed over the map. | Should |
| UR_SCORING | The system should display a score based on their performance in the game. | Should |
| UR_TUTORIAL | The system should provide instructions on how the player interacts with the game. | Should |
| UR_ADDITIONAL_MAP | There may be an additional map of the city center, this map must also resemble its real life counterpart by including recognisable landmarks. | Shall |
| UR_LEADERBOARD | The user should be able to view a leaderboard, listing the top 10 Scores and their respective players. | Shall |
| UR_STREAKS | The player should be able to achieve streaks that help | Shall |

| | | |
|---|---|---|
| | to add points to a player's score. | |
| UR_AVATARS | The player must be able to choose from at least two distinct avatars for their playable in-game character. | Shall |
| UR_CREDITS | The user should be able to view a credits screen, crediting sources of assets. | Shall |
| UR_MAP_OVERVIEW | The user may be able to zoom out, to see an entire view of the map (with labels). | May |
| UR_BUILDING_INTERIOR | The user may be able to enter at least one building, with its own sub map containing multiple interactable activities. Furthermore each building's interior must be representative of its real life counterpart. | May |
| UR_SOUND | There may be appropriate background music (happy/positive vibes) and suitable sound effects for the actions of the player/NPCs. | May |
| UR_ACCESSIBLE | The game shall also cater to colour blind users | Shall |

Functional System Requirements

| ID | Description | User Requirements |
|---|---|---|
| FR_MENU_NAVIGATION_CONTROLS | The mouse will be used for navigation: | UR_MENU |
| FR_MENU_SELECT_CONTROLS | Left click will be used to select items in the menu. | UR_MENU |
| FR_START_GAME | When the start button is selected, the user is then shown a screen where the tutorial begins (tells the player their objectives and controls for their character). After clicking continue, the user must be able to select an avatar for their playable in-game character. Then the game begins and the screen changes to display the map and sprite. | UR_MENU |
| FR_ADDITIONAL_MAP | The additional map may be representative of Campus West/York City Centre/Heslington village and include at least one (club, shop cafe, Library) The additional map must be accessed by interacting with a bus stop to switch maps. | UR_ADDITIONAL_MAP |
| FR_MAP_BUS_STOP | The player can interact (press E) with a bus stop in order to switch between the two maps: Campus and the City centre. | UR_MAP, UR_ADDITIONAL_MAP, UR_INTERACT |

| | | |
|---|---|---|
| FR_INTERACT_CONTROLS | The user can interact with a given object by clicking E () | UR_INTERACT |
| FR_INTERACT_PROMPT | When the user is close enough to an interactable object, there will be a prompt for them to interact with the object | UR_INTERACT |
| FR_INTERACT_DISTANCE | The system will only allow interaction from within a short distance of the object/building they are trying to interact with. | UR_INTERACT |
| FR_PLAY_SPORT | The user can play sports in the sports village, by interacting with the gym. | UR_INTERACT |
| FR_FEED_DUCKS | The user can feed ducks by interacting with the lake. | UR_INTERACT |
| FR_STUDY | The user can study by interacting with the Computer Science Building or the Ron Cooke Hub. | UR_INTERACT |
| FR_DOUBLE_STUDY | The system will allow the user to study twice in a day only once in the whole game. | UR_INTERACT |
| FR_EAT | The user can eat a meal by interacting with the Piazza. | UR_INTERACT |
| FR_SLEEP | The user can sleep in their room, ending the day, by interacting with accommodation. Doing so increments the day by one, resets the time to the morning and sets the energy to 100% | UR_INTERACT, UR_ENERGY, UR_TIME |
| FR_ACTIVITY_ENERGY | When the user takes part in an activity the following energy percentages should be subtracted from the player's energy level for each respective activity: studying 20%, clubbing 50%, eat 10%,<br>feeding ducks/relaxing 10%<br>gym 20%. | UR_BOOK<br>UR_SPORT<br>UR_DUCK<br>UR_STUDY<br>UR_ENERGY |
| FR_ACTIVITY_ENERGY_REQ | The system will not allow the user to perform an activity when the resultant energy is < 0%. | UR_BOOK<br>UR_SPORT<br>UR_DUCK<br>UR_STUDY<br>UR_EAT<br>UR_ENERGY |
| FR_SCORING | Each time the user completes a beneficial activity such as sleeping, eating, studying, working out or interacting with NPCs, their score will be increased by a set amount, for each day that the user doesn't study or eat enough their score will be decreased by a set amount. | UR_SCORING |

| | | |
|---|---|---|
| FR_SCORING_GAME_OVER | The user can only find out their score once the game is over. | UR_SCORING |
| FR_PAUSE | When the game is paused a pause menu is loaded. | UR_PAUSE |
| FR_PAUSE_CONTROLS | Esc is used to initiate and end the pause. | UR_PAUSE |
| FR_PAUSE_FREEZE | When the pause is initiated, no other buttons can be pressed and the time pauses. | UR_PAUSE<br>UR_MOVEMENT<br>UR_TIME |
| FR_TIME_LIMIT | When the time of day is 'night', the player cannot complete activities other than sleep. | UR_SLEEP<br>UR_BOOK<br>UR_SPORT<br>UR_DUCK<br>UR_STUDY<br>UR_EAT<br>UR_TIME |
| FR_ENERGY | The player's energy score should be available to the user whilst playing the game. This will be shown by an energy bar in the bottom left of the screen. | UR_ENERGY |
| FR_TIME | The in-game time of day as well as the day (1-7) of the week should be displayed to the user (as part of an in-game overlay). | UR_TIME |
| FR_SCORE_CALC | The system should provide a display of the score from the game based on the number of studies, meals and recreational activities. | UR_SCORE |
| FR_LEADERBOARD | The leaderboard should be displayed in a table-like format with the left column displaying the rank of the instance, middle column for the names of the players and the right column containing the players' scores, with each row containing each instance. The rows should be sorted by score descending. | UR_LEADERBOARD |
| FR_STREAKS | The player should be able to complete 3 streaks. The player can only be informed of these streaks upon completing the game and fulfilling the streaks' requirements, making them somewhat hidden in a sense. | UR_STREAKS |
| FR_STREAK_GYM_BRO | Requirement: Go to the gym at least 3 times a week | UR_STREAKS |
| FR_STREAK_LOTD | Requirement: Feed the ducks 6 times in the week | UR_STREAKS |

| | | |
|---|---|---|
| FR_STREAK_JOGGER | Requirement: Walk at least 200 steps each day | UR_STREAKS |
| FR_ACCESSIBLE | The game shall make use of textures and high contrasting colours, to help distinguish between graphics for the colour blind. | UR_ACCESSIBLE |
| FR_MAP_OVERVIEW_CONTROLS | The M key will be used to toggle between the regular perspective and the map view | UR_MAP_OVERVIEW |
| FR_MAP_OVERVIEW_MOVEMENT | Whilst the player is in the map overview view, they will be unable to control their character. | UR_MAP_OVERVIEW, UR_MOVEMENT |
| FR_CREDITS | A credits screen containing information about the creators, game engine, assets etc. shall be accessible through the main menu | UR_MENU, UR_CREDITS |
| FR_TUTORIAL | The tutorial should inform the player of the in-game controls as well as their objective: To have the best experience at York - which involves a balanced life of studying, sleeping, eating and recreational activities. UR_TUTORIAL | The tutorial should inform the player of the in-game controls as well as their objective: To have the best experience at York - which involves a balanced life of studying, sleeping, eating and recreational activities. UR_TUTORIAL |
| FR_AVATARS | The player must be able to choose from at least two distinct avatars for their playable in-game character. These should be masculine and feminine. | UR_AVATARS |

Non-functional System Requirements

| ID | Description | User Requirements | Fit Criteria |
|---|---|---|---|
| NFR_RESPONSE | The resultant action is performed within 0.16 seconds after the user input. | UR_MOVEMENT | The rate at which the system checks for inputs is greater than 0.16 seconds. This will check for inputs every frame, with a frame rate set to 60. Use stress tests to measure this with large volumes |

| | | | |
|---|---|---|---|
| | | | of inputs. |
| NFR_COMPATABILTY | The game should be compatible with desktop computers (Windows 10) supporting good response times and few errors. | UR_COMPATABILTY | The response times are less than 0.1 seconds while running on a desktop computer. |
| NFR_USABILITY | The user should understand how to interact with our game. | UR_TUTORIAL | When the user presses P in the game, a tutorial will be displayed, explaining how to use the game. Furthermore, when they start the game, a tutorial will be displayed. |

**References**

[1] Nuclino. *A Guide to Functional Requirements (with Examples)*. www.nuclino.com [Online]. Available at: https://www.nuclino.com/articles/functional-requirements [Accessed: 17th of March, 2024].

[2] Geeks For Geeks. (2024, Jan, 17th). *Non-functional Requirements in Software Engineering*. geeksforgeeks.org [Online]. Available at: https://www.geeksforgeeks.org/non-functional-requirements-in-software-engineering [Accessed: 17th of March, 2024].

[3] Indeed Editorial Team (2022, Dec, 19th). *What is a user requirements specification? (Plus elements)*. uk.indeed.com. [Online]. Available at: https://uk.indeed.com/career-advice/career-development/user-requirements-specification [Accessed: 17th of March, 2024].