ENG1 - Assessment 2

Updated System and User Requirements

Req2.pdf

Group 4

Mikaella Loppnow	ml2708
Tom Daly	td1026
Ethan Buss	eb2225
Dillon Pandya	dp1195
Ereife Odusi	ed781
Harriet Kirby	hk1114

The requirements were divided into direct user requirements and system requirements. Within the system requirements, both functional and non-functional requirements were gradually incorporated as the system implementation progressed. Many non-functional requirements impose direct constraints on the system to align it with stakeholders' vision and ensure the software adheres to user requirements.

In crafting the requirements specification, we also took into account the system's modes of operation. Given the system's nature, our focus was primarily on scenarios when the system is idle or encounters failures, as we had already addressed operational requirements for normal functioning. This consideration encompasses situations such as when the player is inactive or when the system encounters bugs, leading to additional system requirements.

We documented the requirements across three tables, assigning each a unique ID. The user requirements table features a colour-coded priority column, facilitating the engineering team in prioritising their implementation efforts to align closely with the project's goals. The remaining two tables are directly linked to the user requirements table, with priorities aligned to ensure efficient development, centred on meeting necessary requirements while adhering to identified constraints. Moreover, non-functional requirements are accompanied by fit criteria to enforce constraints that enable the system to fulfil user expectations.

We opted against utilising a use case design approach [1]. We felt it inappropriate and overly complex for the system's design scope. Instead, the system's workings are delineated within the architecture document, provided as a separate deliverable for clarity.

Upon thorough analysis of the system, including interviews with the customer and review of the product brief, we determined that we possess sufficient information to commence presenting our requirements. As previously outlined, these requirements will be showcased across three interlinked tables, correlating user requirements with appropriate descriptions. Additionally, our development process will entail continuous testing and refinement, enabling us to adapt and evolve the system's requirements if the need arises. The team decided on this style of requirement presentation from the given standards of how to elicit requirements given in [2]. This content was also showcased in the lectures supporting our confidence in our approach.

(No changes)

USER REQUIREMENTS

ID	Description	Priority
UR_SCORE	The player should obtain a score at the end of the game and have an overall high score.	Should
UR_PC	The game should be able to be played on most standard computers.	Should
UR_AUDIENCE	The game should be family friendly and contain appropriate content.	May
UR_TIME_FINISH	The game should last 7 days with limited activities in that time frame. Each day should contain 16 hours of playtime.	Shall
UR_INTERACTION	The game should have the player interact with the environment in order to reach the final exam. There should be interactions for studying, recreational activities and resting.	Shall
UR_MAP	The game should be set in York and there should be locations in the game set around that area that make up the map with identifiable locations.	Shall
UR_SLEEP	The player should be able to go to sleep and a day should pass onto the next.	Shall
UR_ENERGY	The player should use energy when interacting with the environment before the exam.	Shall
UR_MINIGAME	The game should have extended interaction and playability after further iteration.	May
UR_SOUND	The game should have music, sound effects and sound bites.	May
UR_ASSETS	The game should have assets that represent a realistic friendly art style.	Should
UR_EXAM_END	The game should lead to a final exam that produces a score once complete.	Shall
UR_CUSTOM	The player can be customizable before starting the game through preset choices. The controls should also be customisable.	May
UR_AVATAR	There should be an avatar that is movable around the map by the player.	Should
UR_FAILURE	The system should do something to remedy the problem if it runs into a detectable failure.	May
UR_STREAKS	The player will receive an achievement badge, known as a streak, if certain activities are carried out throughout the game.	Shall
UR_LEADERBOARD	The system shall display a leaderboard with the top 10 people who have completed the game successfully.	Shall

SYSTEM REQUIREMENTS: *Non-Functional Requirements* **(NFR)**

ID	Description	User Requirements	Fit Criteria
NFR_PC	The game should be playable on PC and standard systems	UR_PC	Don't make the system operational for Android or IOS.
NFR_SCREEN	The game should fit any standard computer screen	UR_PC	The game should be scalable to fit different aspect ratios and screen sizes.
NFR_PG	All content within the system should be family friendly	UR_AUDIENCE	No references to alcohol, drugs or other dark themes.
NFR_TIME_FINISH	Game should be relatively quick to finish	UR_TIME_FINISH	The game should take 5-10 minutes to complete.
NFR_TIME_PASS	Some time shall pass with every interaction in a specific increment.	UR_TIME_FINISH	Time passing should be reasonable within a 16 hour day time frame. <= 1 hour per interaction.
NFR_LOCATION	Map must be recognisable to york, with intractable locations unique to york.	UR_MAP	Must contain some of Heslington East campus with at least; 1 place to study (max 2), 1 place to sleep, 3 places for recreational activities (max 6) and 1 place to eat (max 3)
NFR_SOUND	The sounds should be obtained from copyright free sources fitting the style of the game.	UR_SOUND	Should be legally obtained and licensed.
NFR_ASSETS	The assets should be obtained from copyright free sources fitting the style of the game	UR_ASSETS	Should be legally obtained and licensed.
NFR_SLEEP	The player should be able to sleep whenever they wish and the day should pass onto the next	UR_SLEEP	There should be no sleep prevention in the game so progression to the end exam is not stopped.
NFR_EXAM_TIME	The exam can be taken after 7 days of time has passed.	UR_EXAM_END	Do not allow the player to take the exam before 7 days of in game time have passed. The exam and completion of the game has to take place on the 7th day.
NFR_STUDYING	Studying can be done everyday at least once. The game should allow studying to catch up if the player missed one day	UR_EXAM_END UR_INTERACTION	Studying to catch up if the player missed the day before can only occur once per game session.
NFR_IDLE	If the system is idle it should prevent a situation where the score of the player is affected.	UR_SCORE	No time or energy should pass when the system is idle. The score should not be affected.

SYSTEM REQUIREMENTS: *Functional Requirements* **(FR)**

ID	Description	User Requirements
FR_SCORE	A score must be calculated based on how well the player did in the game and in the end exam.	UR_SCORE
FR_HIGHSCORE	When a player achieves a new high score the current high score should be updated.	UR_SCORE
FR_TIME_PASS	There should be a passing of time while the player is interacting with the game or using energy.	UR_TIME_FINISH
FR_FEEDBACK	Each interaction should show the user how this has affected their time and energy, with a brief pop up or an animation or sound que.	UR_INTERACTION
FR_MOVEMENT	The map should be able to be moved around inside as an avatar representing the player.	UR_MAP
FR_ENERGY_BAR	The game should have an energy bar displayed that contains a certain amount of energy in the day.	UR_ENERGY
FR_ENERGY_USE	Energy can be spent on interactions which will make the energy bar total fall in increments for that interaction.	UR_ENERGY
FR_ITERATIONS	The system should be designed such that further iterations to areas of the game requiring polish are possible.	UR_MINIGAMES UR_CUSTOM
FR_END_GAME	The system should allow the player to take the exam after 7 days and therefore obtain a score. The system should end the game in the process after the final exam showcasing a score.	UR_EXAM_END UR_SCORE
FR_CUSTOM_AV	The system should allow the player to select different preset customizable features for their player's avatar.	UR_AVATAR
FR_AVATAR_USE	The avatar should be able to interact with all other system functionality in order to reach the end game.	UR_AVATAR
FR_CONTROLS	The system should allow the modifications of controls to play the game.	UR_CUSTOM
FR_SLEEP	The game should allow the player to sleep at a designated location in campus east.	UR_SLEEP
FR_RECREATION	The game should allow the player to do recreational activities through interactions.	UR_INTERACTION
FR_STUDYING	The game should allow the player to study through interactions. There should be an expected amount of studying to do well in the final exam and this should be tracked in some way.	UR_INTERACTION UR_EXAM_END
FR_FAILURE	The system should provide feedback if an error was detected or attempt to correct itself to keep running if something abnormal occurs.	UR_FAILURE
FR_SCORE_CHANGE	The system should change the score based on specific conditions associated with what interactions were performed.	UR_SCORE UR_INTERACTIONS
FR_TIME	The system should have a time displayed that passes after interactions.	UR_TIME_FINISH
FR_LEADERBOARD	The leaderboard shall be findable in the menu throughout the game and also after completion of the game in the game over screen.	UR_LEADERBOARD
FR_STREAKS	The player's achieved streaks will be displayed after successful	UR_STREAKS

	completion of the game in the game over screen.	

References

[1] I. Jacobson. Object Oriented Software Engineering: A Use Case Driven Approach 1st Edition. Addison Wesley: 1992.

[2] IEEE 29148-2018 ISO/IEC/IEEE International Standards Association. (2018-11-30). *Systems and software engineering - Life cycle processes - Requirements engineering* [Online]. Available: https://standards.ieee.org/ieee/29148/6937/