2. Requirements

Part a)

After arranging a meeting with the stakeholders, the next process was to decide on the requirements needed for the project. Using the provided product brief, which specified all the compulsory requirements needed for the first assessment, we were able to elicit the majority of requirements needed for the product. Next, we prepared a list of questions for the stakeholders, where members of the group inquired about additional specifications/details that should be incorporated into the product. Then, the questions were posed to the stakeholders, which consisted of an ENG1 lecturer and The University of York Communications Office. The questions were mainly feature-related, sound effect related, user-related, and some were non-feature related. The feature related questions included questions such as how long a burger should cook and etc. The sound effect related questions were about the overall audio of the game. The user related questions included questions such as who will be the users of the game and if they are familiar with the technology of the game. The non-functional questions were asking about the system's usability, availability, and security. There were also questions posed inquiring about The University of York Communications Office's requirements, which were not that many since they had no requirements.

After finishing the meeting with the stakeholders, we had a clear understanding of the product's requirements and next we had to start documenting the requirements. First, we had to conduct research[1] into requirements specification and presentation. We examined published documents[2] and also extracted information from the book to help our process. For requirements specification, the general consensus was that the best notation to use for the requirements is natural language, seeing that after going through the "Software engineering" book[3] and searching through various articles, it was clear that was the optimal choice. Natural language specification is easy to understand and expressive, meaning that it is comprehensible to everyone and can be used for all the requirements we've elicited. We also decided that each requirement should have a consistent meaningful name instead of a number id, which makes it easier to identify them later on. Lastly, for the requirements presentation, we agreed to use three tables–one for user requirements and two for system requirements. Tables are organised, simple, and easy to follow. Furthermore, by using tables we were able to easily organise the requirements in hierarchical order and order them by an appropriate priority. Tables are also easy to extend, which makes it easier to add and refine requirements throughout the development lifecycle if necessary.

Going into assessment 2, we felt that our first main priority was to update this document with the new requirements as introduced in the assessment brief and further announcements. By doing this, we felt that it would make planning and keeping on track with development easier. We paid close attention to the structure that the previous team had set out in order to maintain the traceability of the requirements and keep the document cohesive.

Part b)

SSON: "The game shall enable players to control multiple chefs to prepare and cook food at each customer's request."

User Requirements

ID	Description	Priority
UR_CONTROL_COOKS *	The game shall allow the player to control *three* chefs individually (with the possibility that this number is increased through upgrades)	Shall
UR_INGREDIENTS	The player shall be able to collect ingredients	Shall
UR_COOK_FOOD *	The player shall be able to make salads, *pizzas*, *jacket potatoes* and burgers	Shall
UR_SERVE_FOOD *	The player shall be able to serve salads, *pizzas*, *jacket potatoes* and burgers to customers	Shall
UR_CUSTOMERS	The game shall have a fixed number of customers to serve that require one dish each	Shall
UR_FAILING_STEPS *	The player *shall* be able to overcook or fail making food or serving customers	Shall
UR_WRONG_INGREDIENT	The player shall be able to get rid of ingredients that they have accidentally created	Shall
UR_UX	The game shall offer a pleasant user experience	Shall
UR_INSTRUCTIONS	The instructions to cook food shall be displayed to the user along with controls	Shall
UR_GRAPHICS	The graphics of the game shall be clear and easy to understand. The graphics shall also be child friendly.	Shall

UR_SCALABILITY	The game shall be able to be displayed on both big and small screens	Shall
UR_COMPATIBILITY	The game shall be able to be played on multiple operating systems	Shall
UR_ACCESSIBILITY	The game should provide accessibility options	Should
UR_BRANDING	The level should be consistent with the Piazza building and contain UoY branding	Should

UR_TIME_TO_COMPLETE	The game should take 5-6 minutes on average to complete, but may go on for longer, particularly in the endless mode	Should
UR_SETTINGS	The game should provide the option to customise settings to the player's preference	Should
UR_SOUND	The game may have sound effects	May
UR_SAVE_GAME	The user shall be allowed to save their progress at any point during or after the game	Shall
UR_CURRENCY	The player shall earn an in game currency and use it to upgrade features of their kitchen	Shall
UR_DIFFICULTY	The user shall be able to choose between three levels of difficulty	Shall
UR_POWERUPS	The game shall provide 5 powerups which the user can unlock and use	Shall
UR_MODES	The user shall be able to choose between which mode they want to play, endless or scenario mode	Shall

Non-functional requirements

ID	Description	User Requirements	Fit Criteria
NFR_AVAILABILITY	The system shall be highly available	UR_CONTROL_CHEFS	Uptime: 100% during the open days
NFR_DOCUMEN TA TION	The system shall have a guide that details all its functions	UR_INSTRUCTI ONS	Clear instructions on how to play the game
NFR_OPERATAB ILI TY	The system shall be operable by customers that have no previous experience with the game	UR_INSTRUCTI ONS & UR_GRAPHICS	Easy to understand interface with clear instructions
NFR_ACCESSIB ILI TY	The system shall be operable by those with accessibility issues	UR_ACCESSIBIL IT Y	Cater for those with accessibility needs
NFR_USABILITY	The system shall contain no technical jargon	UR_INSTRUCTI ON S	Not use any complicated terminology

Functional Requirements

ID	Description	User Requirements
FR_CHANGE_PLAYABL E_ CHARACTER	The system shall let the user switch control between playable characters	UR_CONTROL_CHEFS
FR_MOVE_PLAYABLE_ CH ARACTER	The system shall have controls that move the playable character	UR_CONTROL_CHEFS
FR_GRAB_ITEMS	The system shall allow the player to grab various in-game items	UR_INGREDIENTS

FR_FLIP_AND_CHOP	The system shall allow the player to flip and chop certain items.	UR_COOK_FOOD
FR_PLACE_ITEMS	The system shall let the player place items after grabbing them.	UR_SERVE_FOOD
FR_REMOVE_ITEMS	The system shall let the player completely remove items from the game	UR_WRONG_INGREDIENT
FR_OVERCOOKING	The system shall allow items to be overcooked or overbaked	UR_FAILING_STEPS
FR_SERVE_CUSTOMER	The system shall let the player serve the customer their order	UR_CUSTOMERS
FR_CUSTOMERS_LEAVE	The system shall have customers leave after a certain amount of time if they have not been served	UR_CUSTOMERS
FR_GUIDE_USER	The system shall subtly guide the user and make sure they finish each task successfully every time	UR_INSTRUCTIONS
FR_FULL_SCREEN	The system should let the user play on full screen mode	UR_SCALABILTY
FR_COLOR_BLINDNESS	The system should let a user with colour blindness choose a suitable colour palette to enhance their gaming experience	UR_ACCESSIBILTY
FR_LOADING_SCREEN	The system should display the logo when the system is loading	UR_BRANDING

FR_TIMER	The system should have a timer that shows how much time has elapsed	UR_TIME_TO_COMPLETE
FR_SAVE_CHANGES	The system should remember the user's settings	UR_SETTINGS
FR_VERIFY_SETTINGS'_CHANGES	The system should verify if the user would like to save the changes	UR_SETTINGS
FR_MUTE_SFX	The system shall let players mute sound effects/music	UR_SOUND
FR_SAVE_GAME_STATE	The system shall let the user save their progress during the game to return to later	UR_SAVE_GAME
FR_EARN_MONEY	The system shall give the user an amount of in game currency, depending on how well they performed	UR_CURRENCY
FR_INVEST_EARNINGS	The system shall let the user invest their earnings after a game in order to improve their kitchen	UR_CURRENCY
FR_DIFFICULTY_SELECTION	The system shall allow the user to select between three different difficulty levels at the home screen	UR_DIFFICULTY
FR_DIFFICULTY_AFFECTS	The behaviour of the game (customer waiting time, upgrade cost etc) shall change depending on which difficulty was selected	UR_DIFFICULTY
FR_POWERUP_ACTIVATION	The system shall provide functionality to activate one of the powerups	UR_POWERUPS

FR_POWERUP_COOLDOWN	The system shall allow for a cooldown period after a powerup has been used, where it can't be activated again until this period is over	UR_POWERUPS
FR_POWERUP_ABILITIES	The system shall provide 5 powerups with distinct abilities	UR_POWERUPS
FR_SCENARIO_MODE	There shall be a scenario mode where a fixed number of customers appear	UR_MODES
FR_ENDLESS_MODE	There shall be an endless mode, where customers keep appearing until the player loses	UR_MODES