Requirements

Group 1
Assessment 2

Emma Hogg
Isaac Rhodes
Isioma Offokansi
Toby Meehan
Seungyoon Lee
Ali Yildirim

Requirements

Single Statement of Need: The system shall allow a user to interact and control cooks to prepare and deliver dishes on virtual customers' demands.

"You are to build a single-player game that requires managing the staff around a kitchen, who will be preparing various dishes requested by customers coming into the Piazza Restaurant."

Introduction

By following the initial brief set out for us, we've managed to set fundamental requirements that were necessary to provide for our game. Following reading the product brief for each Assessment, we felt it was necessary to meet with the customer directly to ensure we understood all requirements needed. We also asked whether any extra features, not included in the product brief, are required for the customer. This ensured that we covered everything necessary for the customer in the game so the product was as successful as possible. We felt it was important that we did this so we did not make any assumptions for implementation of the game.

Each user and software requirement has been recorded in the following tables with a unique but comprehensible ID that relates to the requirement. This was needed as we can refer to our requirements directly without confusion. The user requirements describe the different tasks that the user should be able to perform using the system. The system requirements however link to what the game should do to support the user requirements. This can be split into two sections: functional requirements link to the different tasks the system must be able to do; the non-functional requirements relate to the qualities that the system must have.

With each requirement, we made sure to write a fitting and clear description to best explain its purpose and why it must be implemented as part of our game. The selection was a group process as we would each submit a requirement and decide together if it meets our brief and is needed for the game.

For the user requirements, we made sure to avoid any confusing technical terms and focused on what the players would be able to carry out and interact with in the game.

Our functional requirements focus on what the game must do to follow the brief and meet our customer's expectations. These are the technical actions that the system will take in order to meet the user requirements.

The non-functional requirements focus on the quality of our game and the user's experience. In order to avoid potential issues in the future, we've specified a fit criteria to achieve for the game to hold our systems to a specific standard and provide an estimation on player's experience interacting with the game.

Our requirement elicitation process as described above was highly influenced by some research that we carried out prior to starting the project. An online report [3], helped us to structure the process that we took. This firstly influenced the structure of our requirements. The table format linked to the fact our requirements should be clear and concise and easy to change. Also, the inclusion of the column "Priority", has allowed us to prioritise the most important requirements over others that may not have as large of a significance. The first two steps in the "Features of Requirement Elicitation" section of the document [3] highlight the importance of the customer and information gathering. This

influenced our decision to speak to the customer regularly to ensure that the requirements we gained were accurate. Furthermore, it linked to the detailed review of the product brief to gather as much information as possible to ultimately enhance our understanding.

User Requirements

ID	Description	Priority
UR_SCENARIO_MODE	The user should be able to play the scenario game mode where it should run until all customer demands have been met, or the player has lost all reputation points.	Shall
UR_ENDLESS_MODE	The user should be able to play endless mode which keeps track of the number of customers you manage to serve before losing all your reputation points.	Shall
UR_CONTROL_SYSTEM	The game shall allow the cooks to move between stations pressing direction keys and switch between cooks by pressing a key.	Shall
UR_ITEMS	Each cook should be able to interact with and carry items retrieved from the pantry. These should be transferable to cooking stations.	Shall
UR_DEMANDS	Each customer shall arrive and give a demand, which will require a recipe to be made that the user can see.	Shall
UR_COOKS	The game shall let the player switch between up to a maximum number of cooks, one at a time. They should be locked out for a certain time when they are used up	Shall
UR_RECIPES	The user will be able to make different recipes for salads, burgers, jacket potatoes and pizza	Shall
UR_REPUTATION_POINTS	The user shall have 3 reputation points that act as lives for the player	Shall
UR_COOKING_STATIONS	The user can use a fixed number of stations to prepare and cook the ingredients for the customer at the beginning	Shall
UR_MONEY	The player can gain money from serving customer in time to use to buy back chefs and stations	Shall
UR_BUY_ITEMS	The game should allow users to buy extra cooking stations during the game.	Shall
UR_PANTRY_STATION	The game shall have a pantry for the cooks to collect ingredients from.	Shall
UR_COUNTER	The user will bring finished meals to the counter to be served to customers	Shall

UR_UX	The game shall offer a pleasant user experience.	Shall
UR_TOOLTIP	A user shall be shown a prompt to indicate how to play the game (which controls do what).	May
UR_PLATFORM	The game should be run on both platforms Windows and Linux	Shall
UR_TUTORIAL	The game should have tutorial to guide players	Shall
UR_LEADERBOARD	The user should be able to see multiple leaderboards for different difficulties	Shall
UR_ACCESSIBILITY	The user should be able to easily navigate the game without any stress.	Shall
UR_RESOLUTION	The game should work on both laptop and larger resolution.	Shall
UR_TIMER	There is a timer during the game to calculate the time taken in scenario mode	Shall
UR_MINIGAME	The user will play a small minigame when chopping, baking and frying to determine whether they have failed the preparation step	Shall
UR_PREPARATION_FAIL	The users should not be able to serve food that includes items that have not been chopped, baked or cooked properly	Shall
UR_POWER_UPS	The user can obtain 5 different powerups to help them during the game.	Shall
UR_DIFFICULTY	The user can choose a level of difficulty for the game (this can be easy, normal, or hard)	Shall
UR_SAVE	Allow the user to save the game in a specific state and return to it later	Shall

Functional Requirements

ID	Description	User Requirement
FR_COOK_CONTROLLER	The system shall allow the user to control and switch the cooks using a key.	UR_CONTROL_SYSTEM
FR_SETTINGS_MENU	The system shall allow the user to change attributes of the game through a settings menu.	UR_ACCESSBILITY
FR_COOK_ACTIONS	The system shall allow the user to set actions for the cooks by means of interactions between customers or stations.	UR_COOKS

FR_CUSTOMERS_ARRIVE	Customers will arrive by themselves or in pairs or threes. Each group will split into serving individual people rather than serving the group.	UR_DEMANDS
FR_DEMANDS	A demand should appear on screen up until it has expired / has been accepted.	UR_DEMANDS
FR_SALAD	The system will allow the user to combine cut lettuce, tomatoes and onions to make a salad	UR_RECIPIES
FR_BURGER	The system will allow the user to combine fried patties and toasted buns to make a burger.	UR_RECIPIES
FR_JACKET_POTATO	The system will allow the user to combine cut baked potatoes and their filling	UR_RECIPES
FR_PIZZA	The system will allow the user to combine a cooked base, tomato sauce and cheese to make a pizza	UR_RECIPES
FR_CUTTING	The system should swap items processed at a cutting station for their chopped version. If they spend too long here the chopping step will fail.	UR_COOKING_STATION, UR_PREPARATION_FAIL
FR_FRYING	The system should be able to let users use the frying station to fry ingredients as part of the cooking process, swapping the item for the fried version. If they spend too long here the frying step will fail and the burger will be burnt and unusable.	UR_COOKING_STATIONS, UR_PREPARATION_FAIL
FR_BAKING	The system should be able to let users use the baking station to bake ingredients as part of the cooking process, swapping the item for the baked version. If they spend too long here however the item they have baked will become burnt and they will fail the step. This item will become unusable.	UR_COOKING_STATION, UR_PREPARATION_FAIL
FR_RESTART_MENU	The system shall give the user the ability to restart or exit the game when necessary	UR_ACCESSIBILITY
FR_MAIN_MENU	The user shall be greeted by a main menu when they run the program which will allow them to access other sub-menus and play the game.	UR_ACCESSBILITY
FR_TUTORIAL	There will be an option on the menu page for the user to watch the tutorial for the game	UR_TUTORIAL
FR_HIGH_SCORE	The user shall be able to see and compare the high score(s).	UR_LEADERBOARD
FR_REPUTATION_DECREMENT	The reputation should decrement if the player failed to satisfy customers demand in certain time.	UR_REPUTATION_POINTS
FR_SERVED	The game will show the number of customers the	UR_ENDLESS

	user has served in the current game	
FR_ITEM_DESCRIPTION	Each item will be described in text to avoid confusion of the design	UR_ITEMS
FR_POWERUP	The system will activate a power up when the cook runs into it at different points in the game. Multiple power ups can be activated at the same time.	UR_POWER_UPS
FR_POWERUP_APPEAR	The power ups will appear randomly in the game for chefs to pick up	UR_POWER_UPS
FR_POWERUP_VISUAL	When a chef has a powerup activated there should be a visual aid in order to see that this is activated	UR_POWER_UPS
FR_MINIGAME	The system will show a visual and allow the user to play the minigame and determine whether they have failed the step or not	UR_MIINIGAME
FR_DIFFICULTY_SET	There will be a page before the game starts that will allow the user to choose a difficulty before the game starts	UR_DIFFICULTY
FR_DIFFICULTY _SETTING	Easy mode will consist of customers waiting longer with power ups appearing more often. Hard mode is the opposite to this with recipes being more prone to failure	UR_DIFFICULTY
FR_BUY_ITEMS	The user can interact with the game to buy back cooking stations during the game in order to help win harder game modes. (The customer mentioned the lack of need to buy back chefs if it was not needed for the number of stations)	UR_BUY_ITEMS
FR_MONEY	The screen will show the amount of money that the user has earned in the current game in order for them to buy back stations.	UR_MONEY
FR_SAVE	During the game there will be an option in the pause menu where you can quit the game but then save the state so that the game can be continued later on	UR_SAVE

Non-Functional Requirements

ID	Description	User Requirements	Fit Criteria
NFR_OPERABILITY	The system shall be operable by any person who uses the game on Windows and Linux	UR_UX	90% of new players should be able to play the game without any external help
NFR_DOCUMENTATIO	The system shall be	UR_UX	99% of players should

N	accompanied by detailed instructions on how to set up and play the game.		be able to access these instructions and understand how to play
NFR_TOOLTIP	The user shall be able to understand how to play the game based on the tooltips, which will contain no technical jargon	UR_TOOLTIP	95% of users will understand the gameplay based off of tooltips
NFR_RESPONSIVENESS	The game shall respond quickly to user input	UR_UX	The game will respond within <1/60 seconds to user input according to recent research about the latency of current video games, see [1]
NFR_AVAILABILITY	The game should be playable for the majority of the time it is open. The game will run as expected here	UR_UX	The user should be able to play the game >95% with no errors or crashes
NFR_PLATFORM_AVAILABILITY	The game should be playable on the systems of most players	UR_PLATFORM	The game should be able to run on 97.64% of computers (96.37% Windows, 1.27% Linux), according to a recent study [2].
NFR_FRAMERATE	The game should have a high framerate the ensure a smooth experience	UR_UX	The game should perform at 30fps or higher which is the average at which a video game runs according to recent research [1] however the higher the better.

References

[1] Richard Leadbetter, Console Gaming: The Lag Factor Sept. 5, 2009 [Online]. Available: https://www.eurogamer.net/digitalfoundry-lag-factor-article?page=3 [Accessed: Mar. 22, 2023]

[2] Steam, Steam Hardware and Software: March 2023. [Online]. Available: "https://store.steampowered.com/hwsurvey/Steam-Hardware-Software-Survey-Welcome-to-Steam" [Accessed: Mar. 22, 2023]

[3] Lunalovegood, Software Engineering | Requirements Elicitation. [Online] Available: https://www.geeksforgeeks.org/software-engineering-requirements-elicitation/ [Accessed: Mar. 22, 2023]