Requirements

Team 7 - Broken Designers

Adam Brown
Morgan Francis
Shabari Jagadeeswaran
Oliver Johnstone
Laura Mata Le Bot
Rebecca Stone

Introduction

To understand exactly what our customer desired from this project, we arranged an initial meeting to discuss the expectations the customer had from the product. We wanted to clarify some of the ambiguities in requirements and missing requirements in the brief.

Before the interview, we prepared questions for the customer in order to target our points of concern, mainly to do with the missing requirements from the brief. The questions we asked targeted three different areas to do with the requirements, being functional, non-functional and constraint requirements. Functional requirements focused on the workings of playing the game, with questions targeting the controls and inputs the customer wanted to be able to play the game. Questions based on non-functional requirements focused on aspects the game needed, but weren't things the player would control or interact with, such as questions about the graphical style of the game and the perspective of the player camera. Finally, questions referring to the constraint requirements focused on the aspects of the technology that will be running the game, such as the systems the game should run on.

From the first meeting with the customer, we were able to come to an understanding on many points of contention. These included things like the freedom to use any input system,camera perspective (although top down was suggested) and graphical style we saw fit. Additionally, we learnt that the game must run on desktops with Windows, Linux or both and should work on multiple display sizes. Final points from the first meeting were that no preference was shown for the use of a grid map and that meals should not have variation.

At the start of the second half of the project, we decided to have another interview with the client in order to discuss the new requirements that were given to us. Just like before, we planned questions to ask before coming to the meeting by polling ideas in a group session. This time, questions were split in accordance with the deliverables they were concerned with. In addition to this, we still asked questions to do with the user, functional and non-functional requirements. Unlike the first interview, most of the questions focused on clarification of requirements rather than eliciting them. This was because we already had a good idea of the product we were going to make, so more detailed clarifications were more important.

This meeting helped to shed light on key points that were needed for the second half of the project. In particular, our customer asked for three difficulty options (being easy,normal and hard) as well as specifically asking for a tea power up, which when used would help increase the time the player had to serve customers. We also cleared up some confusion to do with the save files, such as the state of the saves after the game is turned off. Answers to do with questions that were not about the game included being told that we were allowed to modify the previous teams deliverables, being allowed to use other methods for planning the project instead of Gantt charts (mainly Trello) and clarification on refactoring the website. Finally, we negotiated with the customer about how often we should update them, as well as asking if they would like to see the project before the deadline. The outcome of the negotiation led to the agreement that we would update the customer every two weeks by email with progress, but they said they did not need to see the project before the deadline unless we wanted them to.

We decided presenting our requirements in requirement tables would help us understand and break down the requirements into manageable pieces. We formatted this table using the IEEE Standard Requirements Engineering document [1]. This document suggests creating a small amount of user requirements with descriptions of what they entail, as well as a priority level for them. These will be split into several functional requirements or non-functional requirements that go into more detail about the individual requirements. The ID's of the user requirements are used to link the non-functional requirements and functional requirements to the user requirements. Non-functional requirements will also include an additional column, describing how we will test to see if the requirement has been met. This is done largely due to the fact that non-functional requirements cover a wide range of possible users,devices and other external considerations. This makes it impossible to test every possibility, so criteria that ensures the likely factors work is used instead. This table also makes use of "Users", which are our expected players. We expect "Users" to have at least some knowledge of other PC games, or have played games in the past. The Term was used to avoid repetition within requirements.

Bibliography

[1] ISO/IEC/IEEE 29148-2018 International Standard - Systems and software engineering – Life cycle processes – Requirements Engineering, IEEE Standards Association, 2018.

[2] Steam, *Steam Hardware & Software Survey: March 2023* ,https://store.steampowered.com/ [Online]. Available: https://store.steampowered.com/hwsurvey/Steam-Hardware-Software-Survey-Welcome-to-Steam [Accessed: 9th April 2023]

User Requirements Table

ID	Description	Priority
UR_TUTORIAL	The System shall provide a guide on how to use and play the game. This will be done through a menu in the game, showing the controls.	Shall
UR_CHEF	The game shall allow the user to move a chef and switch between at least three chefs.	Shall
UR_GAMEPLAY	The user shall control the chefs to pick up items and interact with those items to make food, serve customers and eventually win or continue the game. The user should also receive money in order to buy more stations or buy more chefs.	Shall
UR_MODE	The user should be able to pick between two game modes, being endless and scenario. The endless mode should keep going until the player runs out of reputation and scenario should end after all customers are served	Shall
UR_TIMER	There shall be a timer that keeps track of time taken in the scenario.	Shall
UR_ITEM	There shall be items the user can make the chef pick up and interact with. Items should be able to be carried in a stack for both the chef. Items should change as a result of an interaction	Shall
UR_STATION	The game shall have stations that the user can interact with, either by cooking, cutting, baking or crafting. Failing an interaction should result in waste being produced	Shall
UR_SAVE	The user shall be allowed to save their progress at any	Shall

	point, and return even if they close the game down afterwards through loading the game.	
UR_GRAPHICS	The graphics should be engaging and convey key information.	Should
UR_DEVICE	The game should run on multiple desktop environments.	Should
UR_MENU	The game shall have multiple game screens, such as the main menu and the the pause screen	shall

Functional Requirements Table

ID	Description	User requirements
FR_SWITCH_CHEF	The User should be able to swap between chefs	UR_CHEF
FR_MOVE	User should be able to move chefs using the keys “W”,”A”,”S”,”D” for moving the chef up,left,down and left respectively	UR_CHEF
FR_INTERACTION	The system takes the input “SPACE” and detects whether an interaction between a chef, counter and item is possible. If it is, the interaction is performed.	UR_STATION
FR_CHEF_COLLISIONS	If a player attempts to walk into an area of the map which they should not be able to walk through, the controls should be blocked and the chef should touch the object or area blocking them.	UR_CHEF
FR_FOOD_FAIL	If players fail the preparation stage, they will have to repeat the step	UR_STATION
FR_USE_STATION	The user should be able to use stations and start preparation of the items on them	UR_STATION

FR_COUNTER	The system should allow chef to drop top food item on counter	UR_STATION
FR_PREPARE	After interacting with a station, there is a time limit until the user can move while the interaction is taking place.	UR_STATION
FR_COMPLETE_PREP	When preparation is complete, user is able to receive the prepared food item	UR_STATION
FR_GET_FOOD	The user shall be able to pick up food from ingredient stations and hold onto three items on a stack handled by the system	UR_GAMEPLAY
FR_PUT_FOOD_DOWN	The user should be allowed to put down items from their stack onto surfaces in the game	UR_GAMEPLAY
FR_STACK_DISPLAY	The system should display graphics that show the current state of the stack of items held by the player	UR_GRAPHICS
FR_ASSEMBLE	User should be able to assemble a dish from items in the game	UR_GAMEPLAY
FR_SERVE	The users shall be able to serve a customer at a serving station with a completed recipe	UR_GAMEPLAY
FR_WIN	User should be able to complete a scenario	UR_GAMEPLAY
FR_ORDERS	Set number of orders should be displayed for the user to complete	UR_GAMEPLAY
FR_SERVE_CHECK	The system will check if served food is a valid order. If it is, the customer is served and the player receives money. If the order is wrong, the error will be displayed to the user through a notification made by the system	UR_GAMEPLAY
FR_SERVED_COMP	The system compares the number of dishes served and compares it to the amount	UR_GAMEPLAY

	required to complete the game	
FR_POWER_UPS	The user should be able to collect power ups that can be used to gain bonuses. One such powerup must be a cup of tea to calm customers down in order to gain more time to serve them	UR_GAMEPLAY
FR_REPUTATION_POINTS	If a user runs out of time when serving a customer, they will lose a reputation point. If they run out of reputation points, they will lose. Users will start with three	UR_GAMEPLAY
FR_EARN_MONEY	If a user serves the customer, they will earn money (gold)	UR_GAMEPLAY
FR_SPEND_MONEY	Users can spend money (gold) on new chefs or new stations in order to improve their productivity	UR_GAMEPLAY
FR_CUSTOMER_ARRIVE	Customers should arrive and wait at the serving stations	UR_GAMEPLAY
FR_CUSTOMER_WAIT	Customers should wait for a limited time after ordering before leaving. This will be tied to the difficulty	UR_GAMEPLAY
FR_CUSTOMER_GROUPS	The customers should either come into the restaurant on their own, with another customer or in a group of three	UR_GAMEPLAY
FR_CUSTOMER_LOSE	The user should lose reputation if a customer's patience timer runs out.	UR_GAMEPLAY
FR_DIFFICULTY	The user shall be able to pick from three different difficulty levels, either easy, normal or hard	UR_GAMEPLAY
FR_ITEM_INTERACTIONS	The Map shall have an area where the user can go to with certain items in order to “interact”. Types of interaction should be things like taking a patty to an oven to cook it.	UR_GAMEPLAY

FR_ITEMS_EFFECTS	When interacting with an item and environmental object, the system should display a graphic (such as a progress bar) to show this taking place	UR_GRAPHICS
FR_ITEM_CHANGE	When an interaction finishes, the item in the interaction should change in accordance to the interaction.	UR_ITEMS
FR_ORDERS	The system should display the customers orders on screen. This will allow for users to quickly be able to tell what they need to make.	UR_GRAPHICS
FR_ENDLESS_MODE	This mode will continuously allow customers to order food and will keep going as long as the user doesn't run out of reputation foods. The game will keep track of the amount of customers served.	UR_MODE
FR_SCENARIO_MODE	This mode will allow users to configure the amount of customers that appear (default to 5). If a player serves all these customers whilst having at least one reputation point, they win. Else, they will lose.	UR_MODE
FR_SAVE_SLOT	The system should have one save slot that the user caan return to and pick up their game from	UR_SAVE
FR_TIMER	The system has a timer from beginning of scenario to when the win condition is fulfilled	UR_TIMER
FR_PAUSE	System stops the gameplay while the pause menu is displayed. The game should not continue until unpaused.	UR_PAUSE
FR_QUIT	You should be able to quit the game from the main menu, win, lose and pause screen	UR_MENU
FR_SAVE	The player should be able to save the game and keep progress in that save	UR_SAVE

FR_LOAD	The player should be able to load saved progress once in the game	UR_SAVE

Non-functional Requirements Table

ID	Description	User requirements	Fit Criteria
NFR_OPERATE	The system should be able to be operated by most people	UR_TUTORIAL	Users should know controls within 10 seconds.
NFR_SCREEN	The system should work on all screen sizes	UR_GRAPHIC	The game should work on the average screen size as provided by the Steam Hardware & Software Survey: March 2023 **[2]**
NFR_GRAPHIC	The graphics should accurately convey things to the user	UR_GRAPHIC	Users should be able to tell where any chef is, where the currently controlled chef is, what areas are able to be walked on, what food item is on top of stack, where each station is,how much time has elapsed and what the current orders are.
NFR_DEVICE	The game should not contain memory leaks and hence be able to played all day	UR_DEVICE	The game will be available for up to 8 hours at a time.

NFR_PLATFORM	The game should be playable on current versions of windows and Mac	UR_DEVICE	The game runs on Windows 10, Windows 11 and the current Mac OS version.