Requirements

“Lucky” Team 13

Team 13

Yuxin Wu
Bailey Findlay
Elizabeth Edwards
Chenxi Wu
Alex McRobie
Yihong Zhao

Requirements introduction

When given this project we had a brief outline of what this game should be as a whole. Obviously this isn't enough. Sure the chef has to cook a burger, but does he have to stay there the whole time it cooks? Can you swap to a different chef while the first chef cooks the burger? Obviously there are a lot of questions in this fashion so the first thing we had to do was set up a meeting with the customer we were assigned.

We had a team meeting to decide what questions we should ask the customer that weren't immediately obvious to us. We then formatted these questions into a document where we could enter the answers alongside. A small snippet of this can be seen to your right.

First Meeting

Question	Answer
Should interactions continue when you aren't controlling the chef/switch chef	When not controlling the chef, its action continues (in short).
Do we need to plan all requirements even ones we aren't doing	
How should the tutorial level be structured? (Show options considered)	Avoid texts for tutorials. So yes to having a tutorial, but make it intuitive (icons, etc.)

Once we were satisfied with all the answers we set out as a group to create a document where we combined all the requirements we could think of, while trying to link and sort them between user requirements, requirements and then sub-requirements, as shown below:

User Requirements	Sub-User Requirements	System Requirements
You will switch between three cooks, controlling one at a time by instructing them where to go, and helping them prepare each dish.	Switching between cooks	• Assign a key to swap between chefs (Can use the numbers 1 -3 or a defined hot swap key to cycle through them) • It needs to be clear which chef the player is currently controlling.

Now that we had created a rough document showing everything that could be added we had to reformat these requirements in a way that would make it easy to understand for the software engineers or anyone trying to get an overview of the project. When figuring out the best way to format this we looked into the lectures taught by Dimitris Kolovos. Explaining how we convert user requirements into system requirements, in the end leaving us with three tables; User requirements, Functional requirements and Non-Functional requirements, all of which are linked via the user requirement they are derived from.

Once we were happy with this we ended by adding a "Fit Criteria" to our non-functional requirements to illustrate the criteria in which the requirement needs to hit to be a success. After finishing this we had a simple, yet informative presentation of our requirements.

After taking over the project from the previous team, additional user requirements are added to better suit the increased scope of the game. In a similar process as the previous team, a document was created called 'ROUGH REQUIREMENTS' to better visualise the increase in requirements. This limited the requirements that may be accidentally forgotten when additional ones were added.

Requirements:

Endless mode
Change number of customers in scenario mode
Burning
Pizza (Get base) - (put Tomato on) - (Put cheese on) - (Bake it in bake station)
Jacket potato (pick up) - (cook) - (put Tuna on)
3 Reputation point loss implement after time limit
Endless mode customers may come in pairs / 3's
At the start, only some of the cooking stations will be available, and you will be able to invest some of your earnings to enable the other cooking stations, and call more kitchen staff back from leave.
Save game

Implement five special power ups that chefs can obtain (e.g. speed increase, shorter cooking times)

- Auto complete dish
- Shorter cooking timer
- No burning
- Speed increases
- Increase customer patience

- Implement support for different levels of difficulty in the game (e.g. easy, normal, hard)
- Implement facilities that allow players to save the state of the game at any point and resume a saved game later

This includes but is not limited to 'UR_Endless' to accommodate the endless mode and 'UR_Save' for the ability to save. Some user requirements with 'may' priority needed to be changed to 'shall' due to the change in requirements. An example would be 'UR_Investment'. Since the user will be able to spend money to unlock additional cooking stations.

UR_Endless	The user will be able to choose to play an endless mode where customers will keep arriving until they lose	Shall
UR_Customer_Aug	The user will be able to change the number of customers that arrive (default to 5)	Shall

Similarly, following the increase in user requirements, more functional requirements are added since the system will now need to be able to allow the increased user requirements to function. An example of this would be 'FR_Power_Ups' for the ability to acquire power ups that benefit the player.

The non-functional requirements largely remain the same except for some added user requirements. This is due to a lack of differences in the system's operational capabilities and constraints.

User Requirements		
ID	**Description**	**Priority**
UR_Timer	The user should be given a visual time representation relating to the singular task or order they are completing.	Shall
UR_Gameplay	The user should be able to play the game as intended	Shall
UR_UX	The system will offer clean graphics, showing clearly what each station is, along with clearly labelled menus and buttons.	May
UR_Chef_Swap	The user will be allowed to swap between chefs, even whilst one is completing a task.	Shall
UR_Chef_Move	The user can move the chefs freely around the kitchen	Shall
UR_Chef_Interact	The chefs will be able to interact with any object in front of them (e.g. – Chopping board)	Shall
UR_Gamemodes	The user will be able to choose between two game modes; Scenario and Endless	Shall
UR_Reputation	The user will be able to lose reputation points if customers are not served	Shall
UR_Recipe	The user will make different recipes, requiring interaction with different stations	Shall
UR_Customer_Counter	There will be a counter where customers will visibly give their orders and then collect them from the user within a given time frame	Shall
UR_Investment	Users will be able to invest their earnings to unlock more chefs and stations	Shall
UR_Loss	The user will lose the game if all reputation points are lost	Shall
UR_Win	The user will win if they server all customers without losing their reputation points	Shall
UR_Endless	The user will be able to choose to play an endless mode where customers will keep arriving until they lose	Shall
UR_Customer_Aug	The user will be able to change the number of customers that arrive (default to 5)	Shall

UR_Save	The user can save the game	Shall
UR_Burning	The user will be able to burn the food if it was left to cook for too long	Shall

Functional Requirements		
ID	**Description**	**User Requirements**
FR_Controls	There will be a simple control scheme in place for the user to move the chef. The control scheme will include an interact button and a button that will cycle through the controllable chefs.	UR_Chef_Move, UR_Chef_Swap, UR_Chef_Interact, UR_Gameplay
FR_MainMenu	The menu will have a gamemode selection, where the user can configure the game, along with a settings page to change the volume and possibly controls.	UR_Gamemodes, UR_UX
FR_Gamemode	The user will have the option to pick between endless and “scenario” mode. If the user selects scenarios they will be able select how many customers they will need to serve (defaulting to 5).	UR_Gamemodes UR_Endless UR_Gameplay
FR_Alert	The system will notify the user when something has happened (Eg. reputation lost, chef change)	UR_Reputation UR_Chef_Swap
FR_Recipes	The system will provide the user several recipes	UR_Recipe UR_Gameplay
FR_End	The system will recognise when the player has lost/won the game and terminated the session	UR_Loss UR_Win
FR_Collision	The chefs will collide with the and not be able to move closer when in a certain range of an object	UR_Chef_Move UR_Gameplay
FR_Interact	When the chef is within the set range of a station or object, such as the counter, they will be able to use the interact key to use said station or object.	UR_Chef_Interact, UR_Recipe

Functional Requirements		
ID	**Description**	**User Requirements**
FR_Invalid_Interact	If the chef does not have the required ingredients for the station they intend to use they will not be able to interact with the station.	UR_Chef_Interact, UR_Recipe
FR_Reputation	The user will start with 3 reputation points, if they fail to complete and order in the time given they lose a point.	UR_Loss, UR_Timer, UR_Customer_Counter UR_Reputation
FR_Power_Ups	The user will be able to acquire some five special powers	UR_Investment

Non-Functional Requirements			
ID	**Description**	**User Requirement**	**Fit Criteria**
NFR_reliable_gameplay	The game should rarely crash. This should have no permanent effect on the game's operation	All UR	The system should be available 99% of the time.
NFR_bugfixing	Any game breaking bugs will be fixed and updated	All UR	Developers should be able to identify and repair errors
NFR_loadtime	The game should load within a reasonable time frame	All UR	Should be <30 seconds
NFR_lag	The game's lag should be non-existent or at least not noticeable.	All UR	As long as any lag does not affect gameplay it is acceptable.
NFR_instruction_set	The game will include an easily understandable description of the aim of the game and how to operate it	UR_UX	The system will contain clear objectives with instructions
NFR_save_corruption	If a save file becomes corrupted, the other	UR_Save	One malfunctioned save file should not

	files and the game itself will not be affected		impact other save files or the game itself.
NFR_control_set	The game will use simple controls which can be understood by anyone who can use a mouse and keyboard	UR_UX UR_CHEF_MOVE	The system should be simple enough for anyone who can use a mouse and keyboard to control
NFR_tutorial	The game will contain a tutorial and a reminder screen of the controls	UR_UX	Users should be given a clear instruction set available at all times
NFR_options	The hotkeys will be changeable to make the game easier to use	UR_UX	users can change controls
NFR_map_design	Map design will include high contrast and clearly distinguishable designs for interactable objects to make it easy to navigate	UR_UX UR_Customer_ Counter	users are given a clean non cluttered map with items being given visual cues
NFR_audio_visual_compatibility	Any audio cues will also have a visual cue	UR_UX	the game shall be operable by those with hearing impairment
NFR_visual_focus	Pictures will be used to represent things instead of words as much as possible	UR_UX	Provide an objective game by showing users instead of explaining
NFR_standard_controls	Hotkeys and controls will be similar to /the same as other controls for similar games to match the standard	UR_UX UR_Gameplay	users have a familiar and relative control set