Functional Requirements

Requirement ID	**Description**	**Risk and assumptions (if relevant)**	**User Requirement**
FR_MAIN_MENU	The system requires a main menu window. The system should then provide buttons for load game, endless mode and scenario mode. The system should then allow the user to select the difficulty.		UR_CONVENTION S
FR_LOAD_GAME	The system should allow the user to load the last game played.		UR_SAVE_GAME
FR_ORDER_WAVES	In both scenario mode and endless mode, the system should display the order tickets in waves, the number of tickets per wave depends on the difficulty chosen and how long they have lasted. New tickets only come in after the user has completed all the tickets of the previous wave Each order ticket should display the instructions.		UR_DIFFICULTY
FR_DIFFICULTY	In easy mode the tickets should predominantly come in waves of 1 and 2 . For medium, the order tickets should quickly progress to 2 tickets per wave with a few 3s. For hard mode there should be a few 2s but mainly 3 tickets per wave. In easy mode the system should only display the order tickets for salads and burgers, for medium, the system should also display the order ticket for jacket potato and for hard mode the system should display the order tickets for salads, burgers, jacket potatoes and pizza. In endless mode, the time limit to serve all dishes should slowly decrease as the game progresses	The order tickets may not fit on the screen.	UR_DIFFICULTY
FR_ORDER_TIME_LIMIT	The system should provide a timer at the bottom of every ticket, which shows how much time the user has to serve a customer. The amount of time the user has should be different for each recipe.	The time limit to serve a customer should be reasonable	UR_TIME
FR_EXCEEDED_TIME_LIMIT	If the user doesn't serve a customer in time the system should notify the user that they have failed and the system should remove the order ticket.		UR_TIME
FR_RP_LOSS_INFORM_USER	The system should notify the user when they have lost a reputation point	The user may miss the notification	UR_PLAYABLE UR_CONVENTIONS

FR_GAME_OVER	In scenario mode, if the user completed all the orders with at least 1 rep point left, the system should tell the user that they won. If the user loses all 3 rep points before completing all the orders the system should tell the user that they lost. In endless mode the system should display the number of customers served and their high score. In both game modes the game over screen should also display the user's earnings from that particular game.		UR_WIN UR_LOSS UR_CONVENTION S
FR_CUSTOMERS_SERVED	The system should provide a counter for the number of customers served in endless mode		UR_MONEY
FR_MONEY	The system should allow the user to earn money based on the speed of order completion. During a game, only the total earnings are displayed; the money earned during each game should be displayed at the end.	The user may not understand the mechanic	UR_MONEY UR_CONVENTIONS UR_PLAYABLE
FR_UNLOCKABLES	Instead of a separate window, the system should display the unlockable content on the map(plates, frying, chopping, baking) but with a lock/greyed out. The system should also allow the user to unlock extra chefs. The prices should be displayed below each unlockable item.	Relying on colour change for notifications may be an issue for some users.	UR_MONEY
FR_SPEND_MONEY	The system should allow the user to spend money, which should remove the lock and deduct the price from their total earnings.		UR_MONEY
FR_ACTION_TIME_LIMIT	Once a chef starts a cooking action a progress bar should appear above their head which indicates how long the cooking action will take.The system should allow the user to switch back to a chef that is performing a cooking action, however the chef won't be able to move until the cooking action has been completed		UR_TIME
FR_QTE	For baking and frying cooking actions, a small progress bar should appear, which indicates how long the user has to complete the quick time event. The system should allow the user to complete the QTE by first selecting the correct chef, then pressing the appropriate key which removes the progress bar.	The 2 different progress bars may confuse the user.	UR_PREP_STEP
FR_PREP_FAILED	If the user doesn't complete the QTE in time, then the system should notify the user that they have failed the cooking action, and that they will have to bin the item and start again.	The user may miss the notification	UR_PREP_STEP
FR_INGREDIENTS	The system should provide ingredients for pizza and jacket potatoes. Orders for pizza and jacket potato only come in after the		UR_RECIPE

	user unlocks the baking station.		
FR_PIZZA_RECIPE	The system should provide the recipe for pizza : rolling dough, adding sauce, cheese and baking the pizza. The system should allow the user to assemble the pizza on the plate and then pick up the uncooked pizza as 1 item before putting it in the oven.		UR_RECIPE
FR_JACKET_POTATO_RECIPE	The system should provide the recipe for jacket potato: Bake the potato, cook the beans, add the beans to the potato, and add cheese. The system should allow the user to cook the beans.		UR_RECIPE
FR_POWER_UPS	The system should display powerups on the HUD, which can be activated using the spacebar. Power ups are based on how many orders you complete; every 2 orders a new power up should pop up. There should also be a timer which indicates how long the powerup will last. The 5 power ups are : Increase speed of chef, 2x money, burn proof, Order time freeze, 2x cooking speed. The user can only activate 1 power up at a time.		UR_POWER_UP
FR_PAUSE_MENU	The system should allow the user to pause the game, this should open a menu which provides a button for instructions/controls, an exit button, resume button, and a save game button. Ensure the menu displays which user is currently playing.		UR_CONVENTIONS
FR_SAVE_STATE	The system should allow the user to pause and save their game, so their progress is stored when the game is not running. Each saved game should be stored under the name/profile of the user. If the user completes a game, then the saved state is removed.		UR_SAVE_GAME
FR_AUTO_SAVE	The system should save the game state every 10 seconds.		UR_SAVE_GAME
FR_DEMO	The system should play previously recorded gameplay if the game is left inactive for a given amount of time.		UR_INACTIVITY UR_CONVENTIONS

Non-Functional Requirements

Requirement ID	Description	User Requirement	Fit Criteria
NFR_RESPONSIVE	The system should respond quickly to user input from the mouse and keyboard	UR_PLAYABLE	The response time should be <1 second after input
NFR_COOKING_TIME	The recipes should be cooked in a reasonable timeframe	UR_TIME	Each recipe should take x amount of time to make
NFR_AVAILABLE	The game should run and not crash	UR_PLAYABLE	Uptime: 99%
NFR_FRAME_RATE	The frame rate should not drop during expected usage	UR_USER_EXPERIENCE	Minimum frame rate of 20fps
NFR_TIMING	In scenario mode the game should take 5-10 minutes.	UR_USER_EXPERIENCE	5≤ mean time ≤ 10
NFR_LOAD_TIME	The system should load quickly	UR_USER_EXPERIENCE	Maximum time to load should be 20 seconds.
NFR_GAME_LENGTH	The recipes should be cooked in a reasonable time frame	UR_TIME	Each recipe should take x amount of time to make
NFR_ACCESSIBLE	There should be the possibility to add accessibility features in the future	UR_PLAYABLE UR_CONVENTIONS	Features should be added in < 1 month after the game is made
NFR_DESIGN	The system should be colourful and engaging.	UR_USER_EXPERIENCE	80% of testers would describe the system as colourful and engaging.
NFR_NON_VIOLENT	The game should not be violent.	UR_USER_EXPERIENCE	80% of testers would not consider the system to be violent.