Requirements:

Team 14

Joshua Beaven (jb2825@york.ac.uk)

Calum Feenan (cf1218@york.ac.uk)

Harry Hillman (hth512@york.ac.uk)

Jonathon Snelgrove (js3336@york.ac.uk)

Omar Alhosani (oaa560@york.ac.uk)

Rayan Butt (rb1770@york.ac.uk)

Requirements:

2a)

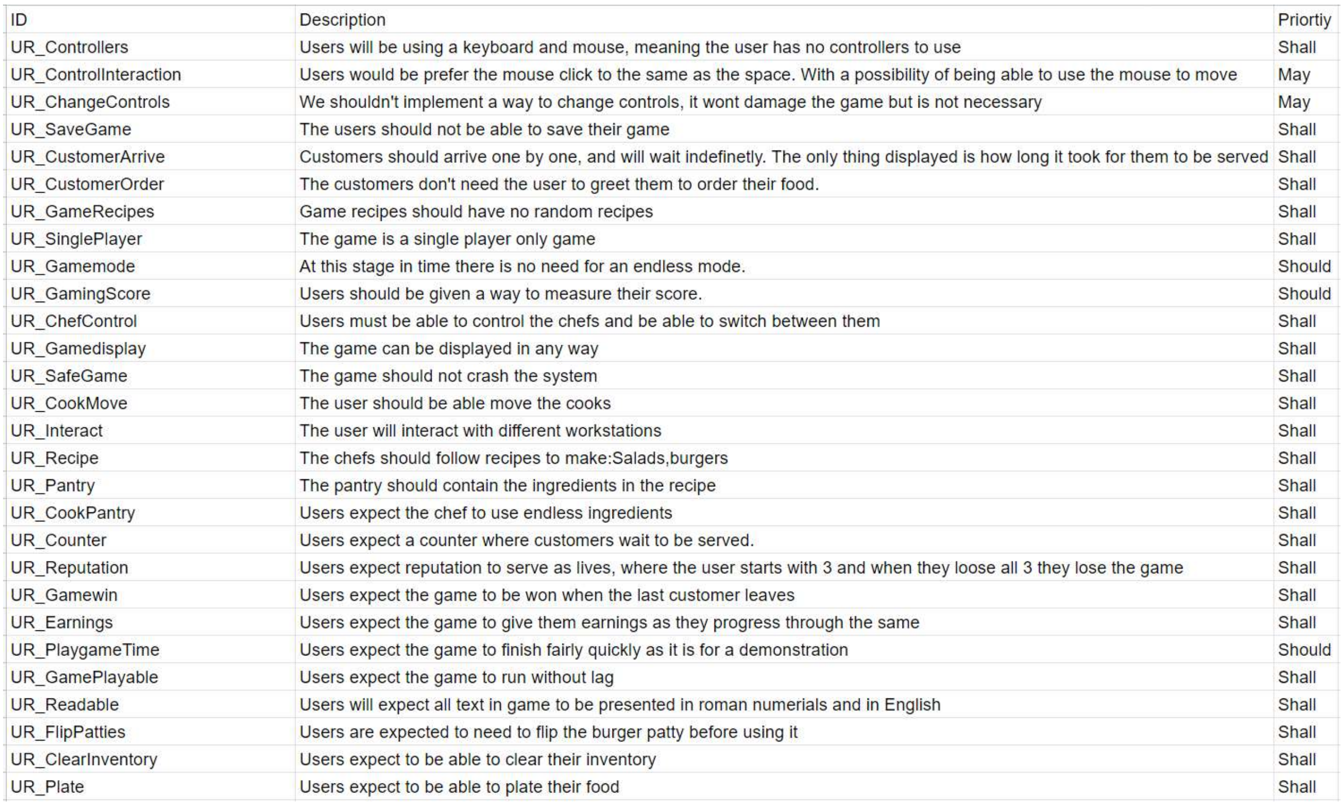
To first elicit our requirements, we first got a Single Statement of Need. This is a concise statement that told us what our goal was, in this case, to create a game called “piazza panic” where chefs would prepare set meals for customers. With our main customers, one being an individual and the other being the York University Communications Office, who plan on using the game on an open day. These requirements are for stage 1 of the project's development. Requirements for stage 2 are not noted down.

Once we understood the SSON, we moved on to understanding the User Requirements, we contacted and got into a meeting with our main customer(the individual) early on in the process to find out about the tasks that the users should be able to carry out using our system, and what the customer expects from the game. We noted down the customers' answers in an QnA sheet and categorised them into their respective requirement type (either a User Requirement or a Functional Requirement). As well as this, user requirements were gathered through the product brief our customer had given us.

From these two, we then designed the appropriate system requirements. Which were further divided into: Functional requirements and Non-functional requirements. These system requirements were what requirements needed to be met in order to meet the user requirements, but were a more technical explanation of the user requirements and allowed us to implement it into code. Functional requirements are things the system must do, where if a user completes an action in the system, the system must give the correct respective response. Non-functional requirements act more as qualities that the system needs to have, that has to match the fit criteria.

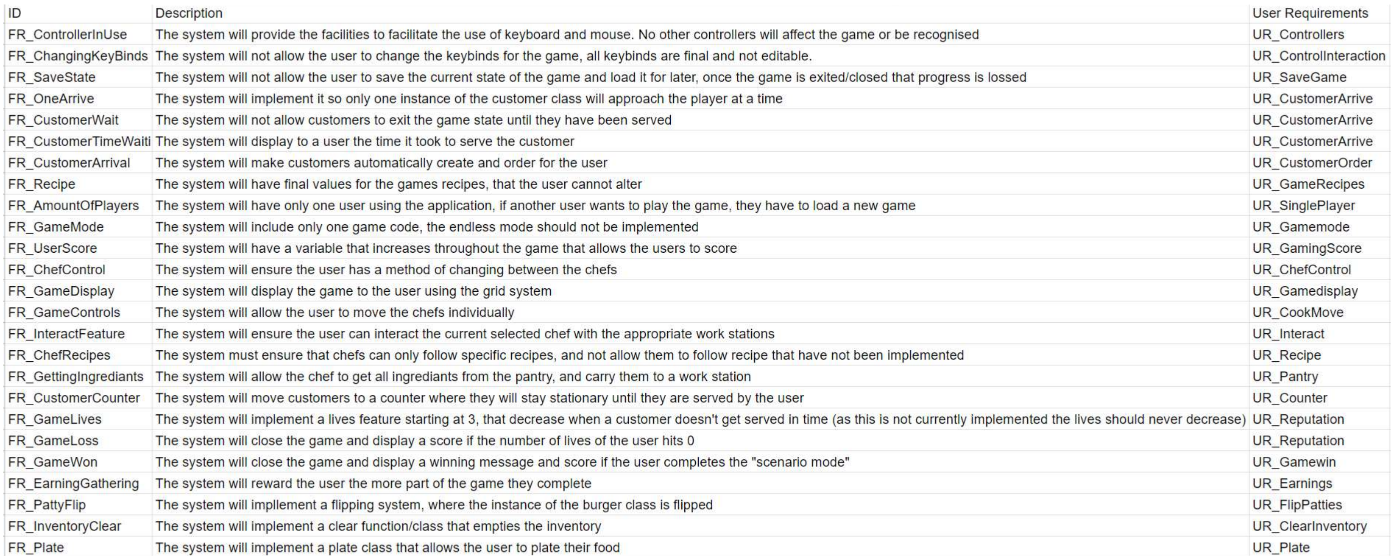
2b) Zooming in on the table is needed to see them

Table 1: User requirements:



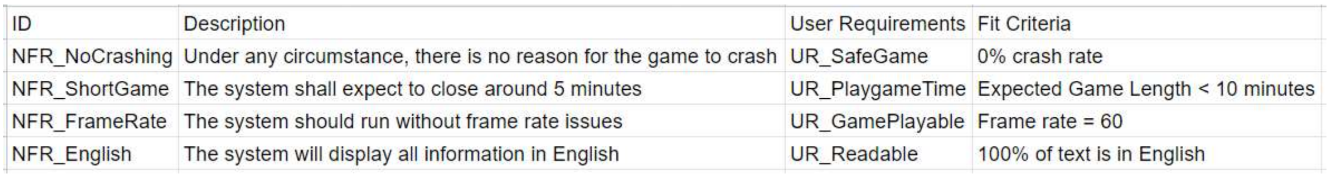
ID	Description	Priortiy
UR_Controllers	Users will be using a keyboard and mouse, meaning the user has no controllers to use	Shall
UR_ControlInteraction	Users would be prefer the mouse click to the same as the space. With a possibility of being able to use the mouse to move	May
UR_ChangeControls	We shouldn't implement a way to change controls, it wont damage the game but is not necessary	May
UR_SaveGame	The users should not be able to save their game	Shall
UR_CustomerArrive	Customers should arrive one by one, and will wait indefinetly. The only thing displayed is how long it took for them to be served	Shall
UR_CustomerOrder	The customers don't need the user to greet them to order their food.	Shall
UR_GameRecipes	Game recipes should have no random recipes	Shall
UR_SinglePlayer	The game is a single player only game	Shall
UR_Gamemode	At this stage in time there is no need for an endless mode.	Should
UR_GamingScore	Users should be given a way to measure their score.	Should
UR_ChefControl	Users must be able to control the chefs and be able to switch between them	Shall
UR_Gamedisplay	The game can be displayed in any way	Shall
UR_SafeGame	The game should not crash the system	Shall
UR_CookMove	The user should be able move the cooks	Shall
UR_Interact	The user will interact with different workstations	Shall
UR_Recipe	The chefs should follow recipes to make:Salads,burgers	Shall
UR_Pantry	The pantry should contain the ingredients in the recipe	Shall
UR_CookPantry	Users expect the chef to use endless ingredients	Shall
UR_Counter	Users expect a counter where customers wait to be served.	Shall
UR_Reputation	Users expect reputation to serve as lives, where the user starts with 3 and when they loose all 3 they lose the game	Shall
UR_Gamewin	Users expect the game to be won when the last customer leaves	Shall
UR_Earnings	Users expect the game to give them earnings as they progress through the same	Shall
UR_PlaygameTime	Users expect the game to finish fairly quickly as it is for a demonstration	Should
UR_GamePlayable	Users expect the game to run without lag	Shall
UR_Readable	Users will expect all text in game to be presented in roman numerials and in English	Shall
UR_FlipPatties	Users are expected to need to flip the burger patty before using it	Shall
UR_ClearInventory	Users expect to be able to clear their inventory	Shall
UR_Plate	Users expect to be able to plate their food	Shall

Table 2: Functional requirements:



ID	Description	User Requirements
FR_ControllerInUse	The system will provide the facilities to facilitate the use of keyboard and mouse. No other controllers will affect the game or be recognised	UR_Controllers
FR_ChangingKeyBinds	The system will not allow the user to change the keybinds for the game, all keybinds are final and not editable.	UR_ControlInteraction
FR_SaveState	The system will not allow the user to save the current state of the game and load it for later, once the game is exited/closed that progress is lossed	UR_SaveGame
FR_OneArrive	The system will implement it so only one instance of the customer class will approach the player at a time	UR_CustomerArrive
FR_CustomerWait	The system will not allow customers to exit the game state until they have been served	UR_CustomerArrive
FR_CustomerTimeWaiti	The system will display to a user the time it took to serve the customer	UR_CustomerArrive
FR_CustomerArrival	The system will make customers automatically create and order for the user	UR_CustomerOrder
FR_Recipe	The system will have final values for the games recipes, that the user cannot alter	UR_GameRecipes
FR_AmountOfPlayers	The system will have only one user using the application, if another user wants to play the game, they have to load a new game	UR_SinglePlayer
FR_GameMode	The system will include only one game code, the endless mode should not be implemented	UR_Gamemode
FR_UserScore	The system will have a variable that increases throughout the game that allows the users to score	UR_GamingScore
FR_ChefControl	The system will ensure the user has a method of changing between the chefs	UR_ChefControl
FR_GameDisplay	The system will display the game to the user using the grid system	UR_Gamedisplay
FR_GameControls	The system will allow the user to move the chefs individually	UR_CookMove
FR_InteractFeature	The system will ensure the user can interact the current selected chef with the appropriate work stations	UR_Interact
FR_ChefRecipes	The system must ensure that chefs can only follow specific recipes, and not allow them to follow recipe that have not been implemented	UR_Recipe
FR_GettingIngrediants	The system will allow the chef to get all ingrediants from the pantry, and carry them to a work station	UR_Pantry
FR_CustomerCounter	The system will move customers to a counter where they will stay stationary until they are served by the user	UR_Counter
FR_GameLives	The system will implement a lives feature starting at 3, that decrease when a customer doesn't get served in time (as this is not currently implemented the lives should never decrease)	UR_Reputation
FR_GameLoss	The system will close the game and display a score if the number of lives of the user hits 0	UR_Reputation
FR_GameWon	The system will close the game and display a winning message and score if the user completes the "scenario mode"	UR_Gamewin
FR_EarningGathering	The system will reward the user the more part of the game they complete	UR_Earnings
FR_PattyFlip	The system will impllement a flipping system, where the instance of the burger class is flipped	UR_FlipPatties
FR_InventoryClear	The system will implement a clear function/class that empties the inventory	UR_ClearInventory
FR_Plate	The system will implement a plate class that allows the user to plate their food	UR_Plate

Table 3: Non-Functional Requirements:



ID	Description	User Requirements	Fit Criteria
NFR_NoCrashing	Under any circumstance, there is no reason for the game to crash	UR_SafeGame	0% crash rate
NFR_ShortGame	The system shall expect to close around 5 minutes	UR_PlaygameTime	Expected Game Length < 10 minutes
NFR_FrameRate	The system should run without frame rate issues	UR_GamePlayable	Frame rate = 60
NFR_English	The system will display all information in English	UR_Readable	100% of text is in English