

## Requirements

Team 16

CatepillaDevelopment

Yousif Al-Rufaye

Joe Fuller

William Gracie-Langrick

Jack Hardy

Bailey Uniacke

Ben Young

## **Introduction**

The requirements detailed in this document were mainly elicited from the brief written by the customer.

We have refined all the requirements into two separate categories: user requirements - what the user wants to be able to do within the system/game and system requirements - what and how the system will deliver user requirements.

Each requirement is also separated into one of three types: a functional requirement - an action the system must perform in order to perform a useful function; a non-functional requirement - qualities or behaviours of the system that are often critical to system performance; or a constraint requirement - global issues that affect the system and its requirements.

Some additional requirements and designs were added following the meeting with the customer. This includes the introduction of a demo mode. Inspired by some classic arcade games, in which if the game is idle for an extended period of time, a pre-recorded gameplay video will be displayed to give any potential players an idea of how the game would be played.

Another design choice that was changed involves accessibility. Taking into consideration those who may be colour blind, we have decided that instead of colours, the chefs should alternatively be differentiated with the shape of their hats, making the game more accessible for those who need it.

In keeping with standard conventions, a collision system will be established so the player characters will not be able to phrase through each other and will instead 'collide' and be made to walk past each other. Also in line with standard conventions, a simple control scheme will be established so the game will be playable even for those who aren't familiar with video games.

---

At the start of the assessment 2 phase of the project, three new requirements - power-ups, difficulty settings, and save games - were given. In order to clarify these, we conducted a new customer meeting. Once the requirements were clear, these were added to the user requirements and functional requirements table. As well as this, features from the initial brief that were not required in the assessment 1 phase are introduced for the assessment 2 phase, such as new recipes and the introduction of a third chef. Since these build on top of previous requirements, there are no new rows for these to the tables below.

Some non-functional requirements have also been added, based on requirements we included in the assessment 1 document that weren't present here, and due to feedback received from the assessment.

## User Requirements

<u>Requirement ID</u>	<u>Description</u>	<u>Risk and assumptions (if relevant)</u>	<u>Priority</u>
UR_WIN	The game must be won when the user has finished serving all the customers	If the level is made too difficult it could be impossible to win	Shall
UR_LOSS	The game is lost when all reputation points are lost		Shall
UR_PLAYABLE	The game should be playable and enjoyable for the cohort, customer and The University of York Communications Office	The game will run smoothly and as intended	Shall
UR_MONEY	The game should allow the user to make money on the completion of an order based on the speed of completion	The user may not know about the mechanic	Shall
UR_CHEF	All chefs should be unique through the shape of their hat.	The chefs are easily differentiated from each other	Shall
UR_RECIPE	All recipes should be unique		Shall
UR_KITCHEN_UNIT	The kitchen units should have different functions. E.g chopping board or frying pan.	The user may not know how to use the unit.	Shall
UR_TIME	The game should be fast paced and the time to complete an order should not be too long	The game could become too easy and not be fun to play	Shall
UR_CONVENTIONS	The game should follow standard conventions	The conventions could be followed blindly and be counter productive	May
UR_COLLISION	The game should not allow the chefs to phrase through each other	Excessive collisions may lead to frustration	May
UR_INACTIVITY	The game should play a pre-recorded demo when the user is inactive for a given time	Assumes the user may be inactive	Shall
UR_POWERUPS	The game should offer five	Power-ups shouldn't	Shall

	power-ups to improve the chefs' performances	make the game too easy	
UR_DIFFICULTY	The game should support different difficulty settings		Shall
UR_SAVE	The game should provide the option to save game progress, and load these saves		Shall
UR_MODES	The game should include options for two modes - scenario and endless		Shall
UR_MULTIPLE_CUSTOMERS	The game should sometimes give the player multiple customers at once to prepare food for		Shall

### **System Requirements - Functional**

<b><u>Requirement ID</u></b>	<b><u>Description</u></b>	<b><u>Risks and Assumptions</u></b>	<b><u>User Requirements</u></b>
FR_FINISH	The system must recognise when all of the customers have been served		UR_WIN UR_LOSS
FR_MOVEMENT	The user must be able to move the chef		UR_PLAYABLE
FR_CONTROLS	The controls decided should be accessible and easily understood even by people who aren't familiar with videogames	Some standard control conventions may not be applicable to our game	UR_PLAYABLE UR_CONVENTIONS
FR_SWITCH	The system must allow the user to switch chefs		UR_PLAYABLE
FR_INTERACT	The system must let the user interact with the kitchen		UR_KITCHEN_UNIT UR_PLAYABLE
FR_SWITCH_INFORMATION_USER	The system will notify the user when they have changed chef	The user may miss the notification	UR_PLAYABLE UR_CONVENTIONS
FR_REPUTATION_LOSS_INFORMATION_USER	The system will notify the user when they have lost a reputation	The user may miss the notification	UR_PLAYABLE UR_CONVENTIONS

	point		
FR_RECIPE	The system will show the user what recipe they are currently making	The instructions may not be clear enough	UR_RECIPE UR_PLAYABLE UR_CONVENTIONS
FR_CONTINUE_ACTION	The system will allow the non-controlled chef to continue their previous action. Eg, continue chopping vegetables	The user may not know about this feature	UR_PLAYABLE UR_CHEF
FR_DEMO	The system will play a previously recorded gameplay that serves as the tutorial if left inactive for a given amount of time	Assumes there may not always be a user playing the game	UR_INACTIVITY
FR_MONEY	The system will allow the user to earn money based on the speed of order completion	The user may not understand the mechanic	UR_MONEY UR_CONVENTIONS UR_PLAYABLE
FR_COLLISION	The system will not allow the chefs to phrase through each other		UR_COLLISION UR_PLAYABLE
FR_POWERUPS	The system should have power-ups that can be picked up and activated by chefs	Activation of powerup will require a new button press	UR_POWERUPS UR_CONVENTIONS
FR_DIFFICULTY	The main menu should offer a range of difficulties for the player to select		UR_DIFFICULTY
FR_SAVE	There should be an option at all times for the user to save their game to return to later. It should save as much information as possible, such as player positions and currently held ingredients	Assuming that as many features as possible getting saved is reasonable with the technology	UR_SAVE
FR_LOAD	When starting a game, the system should offer the option to load a save-game		UR_SAVE

FR_INVESTMENT	The system should offer the possibility of buying new cooking stations, or new members of staff		UR_MONEY
FR_CUSTOMERS	The system should allow for multiple orders to be given at any one time		UR_MULTIPLE_CUSTOMERS

### **System Requirements - Non Functional**

<b><u>Requirement ID</u></b>	<b><u>Description</u></b>	<b><u>User Requirement</u></b>	<b><u>Fit Criteria</u></b>	<b><u>Risks and Assumptions</u></b>
NFG_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	Assuming the input is valid
NFR_GAME_LENGTH	The recipes should be cooked in a reasonable timeframe	UR_TIME	Each recipe should take x amount of time to make	Assuming the player is active in that time
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_AVAILABLE	The game should run and not crash	UR_PLAYABLE	Uptime: 99% during game time	We have the coding ability to make this happen
NFR_DISTINCT_ENTITIES	All entities (chefs or food items) should be easily distinct from each other without relying on colour/text	UR_PLAYABLE UR_CONVENTIONS	Chefs have slightly different models	Assuming that the previous team's sprite textures are easy to edit
NFR_TIMER_ACCURACY	The timer should be	UR_TIME	Should be in seconds (with a	

	accurate so that events can sync properly		few decimal places to display milliseconds).	
NFR_AESTHETIC	The style of the game should be joyful and colourful	UR_PLAYABLE UR_CONVENTIONS	Textures should be colourful	

### **Constraint Requirements**

<b><u>Requirement ID</u></b>	<b><u>Description</u></b>
CR_PROJECT_FINANCE	Financial constraints - we don't want to be spending any money on any tools or otherwise.
CR_PROJECT_LEGAL	Legally we cannot use any tools, libraries etc without proper use (for example crediting libraries that require it, or free-use like LibGDX).
CR_DESIGN_PURPOSE	The purpose of the game is to be displayed and played on an open-day monitor, we should gather requirements and design the game for its purpose.
CR_DESIGN_TECHNOLOGY	The game should be able to run on most if not all machines and we should prioritise minimising input delay / lag and minimising glitches / bugs.
CR_TIME	The game (due to its purpose) needs to be easily accessible and also relatively fast-paced, it should be easy to pick up and should only be 5-6 minutes long.