

Requirements

Assessment 2

Group 3:

Ben Howard <bh1219@york.ac.uk>
Cai Hughes <cabh500@york.ac.uk>
Harry Richardson <hr1040@york.ac.uk>
Ivan Ndahiro <in597@york.ac.uk>
James Sutton <jis509@york.ac.uk>
Yuzhao Liu <yl5164@york.ac.uk>

Group 6:

Igor Smollinski <is942@york.ac.uk>
Pranshu Dhungana <pd861@york.ac.uk>
Zack Tyler-Kyle <ztk500@york.ac.uk>
Phoebe Russell <pbr508@york.ac.uk>
Sanjna Srinivasan <ss3264@york.ac.uk>
Sam Savery <sgs527@york.ac.uk>
Ewan Hutcheson <@eh1776york.ac.uk>

Single Statement of Need: *The system shall allow a user to interact and control cooks to prepare and deliver dishes on virtual customers' demands.*

"You are to build a single-player game that requires managing the staff around a kitchen, who will be preparing various dishes requested by customers coming into the Piazza Restaurant."

Introduction (text highlighted in red indicates the changes made)

By following the initial brief set out for us, we've managed to set fundamental requirements that were necessary to provide for our game. After a meeting with our customer, we further understood what would be required to implement into the project. This is essential because seeking agreement on the requirements from our stakeholders avoids any misinterpretations and aligns with their expectations. This is the elicitation and analysis part of requirements engineering. We used "Software engineering" by Sommerville to help us plan our requirements in an ordered way.

Each user and software requirement has been recorded in the following tables with a unique but comprehensible ID that relates to the requirement. This was needed as we can refer to our requirements directly without confusion. This refers to the requirements specification part of requirements engineering which is the process of writing down the system and user requirements into a requirements document.

With each requirement, we made sure to write a fitting and clear description to best explain its purpose and why it must be implemented as part of our game. Communication of the requirements with everybody in the team is necessary in order to avoid confusions concerning features to be implemented within the game. Hence, the selection was a group process as we would each submit a requirement and decide together if it meets our brief and is needed for the game.

For the user requirements, we made sure to avoid any confusing technical terms and focused on what the players would be able to carry out and interact with in the game. Generally a user requirement describes the functional and nonfunctional requirements of a system in simple language, avoiding technical jargon so that it can be understood by non technical users.

Our functional requirements focus on what the game must do to follow the brief and meet our customer's expectations. These are the technical actions that the system will take in order to meet the user requirements.

The non-functional requirements focus on the quality of our game and the user's experience. In order to avoid potential issues in the future, we've specified a fit criteria to achieve for the game to hold our systems to a specific standard and provide an estimation on player's experience interacting with the game.

Requirements changing is an essential part of maintaining a software project. For phase 2, a new set of user requirements had emerged so we made sure to thoroughly read through the product brief again, and note down all the new user requirements needed to be implemented, i.e., adding new recipes, chefs, etc. We also looked into the three new sets

of user requirements posted on vle, to be implemented for phase 2. After gathering the new set of user requirements, we followed the same procedure for drafting the functional and non-functional requirements, and updated the tables with these.

User Requirements

ID	Description	Priority
UR_SCENARIO_MODE	In Scenario Mode the game should run until all customer demands have been met, or the player has lost all reputation points.	Shall
UR_ENDLESS_MODE	In Endless Mode the game should run until the player either quits the game or all reputation points are lost.	Shall
UR_CONTROL_SYSTEM	The game shall allow the cooks to move between stations by clicking or pressing direction keys and switch between cooks by pressing a key.	Shall
UR_ITEMS	Each cook should be able to interact with and carry items retrieved from the pantry. These should be transferable to cooking stations.	Shall
UR_DEMANDS	Each customer shall give a demand, which will require a recipe to be made	Shall
UR_RECIPES	The game shall have recipes for salads and burgers, initially.	Shall
UR_REPUTATION_POINTS	The game shall have 3 reputation points that act as lives for the player.	May
UR_COOKING_STATIONS	The game shall have a fixed number of stations to prepare & cook the ingredients for the customer.	Shall
UR_ITEM_STATION	Users should have a station to leave their item	Shall
UR_PANTRY_STATION	The game shall have a pantry for the cooks to collect ingredients from.	Shall
UR_COUNTER	The Finished meals may be brought to the counter to be served to customers	Shall
UR_UX	The game shall offer a pleasant user experience.	May
UR_TOOLTIP	A user shall be shown a prompt to indicate how to play the game (which controls do what).	May
UR_PLATFORM	The game should be run at least two platforms Windows, linux, mac (pick 2)	Shall
UR_TUTORIAL	The game should have tutorial to guide players	Shall
UR_LEADERBOARD	There should be multiple leaderboards for different difficulties	Shall
UR_AESTHETICS	The game should not be 3D and violence is not allowed	Shall

UR_ACCESSIBILITY	There should be two different menus, one for main menu and in-game menu	Shall
UR_RESOLUTION	The game resolution should scale well on all manner of screens- laptops, TVs, projector screens etc	Shall
UR_FRAMERATE	The game should perform at 60fps consistently.	Shall
UR_POWER_UPS	The game should have any 5 special power ups of personal choice that the chef can gain during play that makes gameplay easier and more engaging.	Shall
UR_SAVE_STATE	The game should allow the player to save the progress of their game at any point in time, and be able to resume the progress of the saved game later on.	Shall
UR_SUPPORT_DIFFICULTY	The game should allow the player to choose what difficulty (Easy, Normal, Hard) the game should run at. Allowing players to set the level of challenge of the game according to their preferences.	Shall
UR_LOCKED_RESOURCES	The game should allow players to invest their earnings to unlock certain cooking stations. They can also use it to call back an extra cook from leave as a third playable character.	Shall

Functional Requirements

ID	Description	User Requirement
FR_COOK_CONTROLLER	The system shall allow the user to control and switch the cooks.	UR_COOKS
FR_SETTINGS_MENU	The system shall allow the user to change attributes of the game through a settings menu.	UR_ACCESSIBILITY
FR_COOK_ACTIONS	The system shall allow the user to set actions for the cooks by means of interactions between customers or stations.	UR_COOKS
FR_DEMANDS	A demand should appear on screen up until it has expired / has been accepted.	UR_DEMANDS
FR_SALAD	The system will allow the user to combine cut lettuce, tomatoes and onions to make a salad	UR_RECIPES
FR_BURGER	The system will be allow the user to combine fried patties and toasted buns to make a burger	UR_RECIPES
FR_JACKET_POTATO	The system will allow the user to combine baked potatoes and cheese to make jacket potato	UR_RECIPES

ID	Description	User Requirement
FR_PIZZA	The system will allow the user to combine kneaded flour, tomatoes and cheese to make pizza.	UR_RECIPES
FR_ACTION	The system should let users interact with the cooking stations in order to accomplish the actions required to make the recipes such as cutting, frying, baking, etc	UR_COOKING_STATIONS
FR_RESTART_MENU	The system shall give the user the ability to restart or exit the game when necessary.	UR_ACCESSIBILITY
FR_MAIN_MENU	The user shall be greeted by a main menu when they run the program which will allow them to access other sub-menus and play the game.	UR_ACCESSIBILITY
FR_HIGH_SCORE	The user shall be able to see and compare the high score(s).	UR_LEADERBOARD
FR_PREPARE_STAGE	The user should take control of the staff member at certain points in time. If they failed, they have to repeat the step	UR_RECIPES
FR_REPUTATION_DECREMENT	The reputation should decrement if the player failed to satisfy customers' orders within a certain time.	UR_REPUTATION_POINTS
FR_ITEM_DESCRIPTION	Each item will be described in text to avoid confusion of the design	UR_ITEMS
FR_COOKS	The system should allow the player to control 5 cooks at a time. When a cook is interacting with a station the cook should not be able to move during the action time.	UR_CONTROL_SYSTEM
FR_NUMBER_OF_CUSTOMERS	The game shall support a fixed number of customers in the scenario mode, and an unlimited number of customers in the endless mode until the player loses the game, or quits the game.	UR_DEMAND
FR_PLATFORM_AVAILABILITY	The game should be able to run on the systems of the majority of players and not crash, in the specified desirable OS (At least 2 of them)..	UR_PLATFORM
FR_POWERUP_GENERATION	The user has a 70% chance of generating power ups when they make a full recipe, and only a 33% of obtaining power ups if performing only actions, i.e. turning flour to dough.	UR_POWERUPS
FR_INCREASE_SPEED	This powerup will enable the chef to increase their speed by 2 times. Chances of generation is 30% on interactions with stations.	UR_POWERUP Sorder

ID	Description	User Requirement
FR_CLEAR_ORDER	This powerup will allow the user to clear out all the current existing orders from the screen. Chance of generation is 10% on interactions with stations.	UR_POWERUPS
FR_INSTANT_ACTION_GENERATION	This powerup will enable the chefs to instantly generate the item, skipping the action timer. Chance of generation is 30% on interactions with stations.	UR_POWERUPS
FR_INCREASE_REPUTATION_POINT	This powerup when generated will automatically add back one lost reputation point. Chance of generation is 10% on interactions with stations.	UR_POWERUPS
FR_ADD_POINTS	This powerup when generated will add 250 points to the user's original score. Chance of generation is 20% on interactions with stations.	UR_POWERUPS
FR_EASY_MODE	The easy mode has a maximum of 5 customers in scenario mode, and generates customers ~25% slower in both scenario and endless mode. Customers can only appear individually.	UR_SUPPORT_DIFFICULTY
FR_MODERATE_MODE	The moderate mode has a maximum of 10 customers in scenario mode, and generates customers at the default rate in both scenario and endless mode. Customers may spawn in pairs.	UR_SUPPORT_DIFFICULTY
FR_HARD_MODE	The hard mode has a maximum of 15 customers in scenario mode and generates customers ~50% faster in both scenario and endless mode. Customers may spawn in groups.	UR_SUPPORT_DIFFICULTY
FR_LOCKED_RESOURCE	The game will have certain greyed out stations which the player will be able to buy later on using their earned score from serving orders successfully. The player will also be able to unlock an additional chef this way.	UR_LOCKED_RESOURCES
FR_SAVE_STATE	The system will use the LibGDX preferences library to save the data of the game under a file, so the user can load it once they wish to resume the game.	UR_SAVE_STATE

Non-Functional Requirements

ID	Description	User Requirements	Fit Criteria
NFR_OPERABILITY	The system shall be operable by any new player once they have seen the tutorial.	UR_UX	The tutorial should not contain any technical jargon.

ID	Description	User Requirements	Fit Criteria
NFR_DOCUMENTATION	The system shall be accompanied by detailed instructions on how to set up and play the game.	UR_UX	99% of players should be able to access these instructions and understand how to play
NFR_TOOLTIP	The user shall be able to understand how to play the game based on the tooltips, which will contain no technical jargon	UR_TOOLTIP	95% of users will understand the gameplay based off of tooltips
NFR_RESPONSIVENESS	The game shall respond quickly to user input	UR_FRAMERATE, UR_UX	The game will respond within 70 milliseconds to user input
NFR_TIMER_PRECISION	The in-game timer should be precise	UR_DEMANDS	The timer error margin should be <1 milliseconds.
NFR_AVAILABILITY	The game should be in playable state, i.e. the chances of the game crashing should be lower.	UR_UX	The game should be in the playable state for > 99% of the time it is open as stated by the Steam Hardware Report.
NFR_FRAMERATE	The game should offer a stable framerate	UR_UX	The game should have a performance of maximum 60 fps
NFR_POWERUP_TIME	The game has a limited duration for which certain power ups are generated.	UR_POWERUPS	The maximum duration of any given relevant powerup is 15 seconds.