Requirements

This report evaluates the new requirements added to the game developed by “Lucky” Team 13. These new requirements were added to improve the functionality and entertainment of the game. The requirements were implemented based on research from the book “Software Engineering” by Ian Sommerville [2].

A significant addition to the game was the incorporation of pizza and jacket potato. To enable these recipes, new ingredients were added such as, rolled dough, sauce, cheese, and pizza toppings were added. Additionally, two new stations were added [3], a baking station for cooking pizza and an unlockable station that can only be unlocked by accumulating points and purchased in the shop.

To add more challenge to the game, a time limit was introduced for completing orders. Players must complete each order within the allocated time or risk losing reputation points. A total of three such losses results in an overall loss of reputation.

In addition, an Endless Mode was added, allowing players to play the game without a specific objective [4]. To simulate actual restaurant scenarios and make the game more engaging, a menu system, customer counter, and customer groups were added. In addition, five power-ups were introduced to assist players advance and progress levels more quickly, enhancing the entertainment of the game.

To improve the overall game experience, the existing code was edited. Chefs were made to remain at their respective stations, and carrying stacks were included instead of one item. Various customer intervals were incorporated, and the choice of difficulty levels was provided [5].

Lastly, save and continue requirements were added to the game, allowing players to resume their progress at a later time. It is noteworthy that some old requirements were retained because they turned out to be useful and crucial for the system to run smoothly. To ensure that the new requirements align with the original requirements, the current requirements were linked to the previous report. The original report can be accessed at [1].

The team’s commitment to quality control and effective project management ensured that the new requirements were seamlessly integrated into the existing system and met the expectations of all stakeholders.

Assessment 2 requirements:

User Requirements:

ID	Description	Priority
UR_ENDLESS_MODE	A constant amount of customers come until you lose.	Shall
UR_INVEST_MONEY	Can spend earnt money on upgrades	Shall
UR_UNLOCK	Can unlock more stations, recipes or staff	Shall
UR_SAVE_GAME	Save game	Shall
UR_CONTINUE_GAME	Continue Game	Shall
UR_CARRYING_STACK	The player may be able to carry a stack of items at once, instead of just one item at a time, to serve customers more efficiently	May
UR_CHALLENGE	The game should provide an appropriate level of challenge to the player to keep them engaged and motivated	Should
UR_REALISM	The game shall provide a realistic customer flow by varying the time intervals between customer arrivals	Shall
UR_EFFICIENCY	The player shall be able to organise tasks in a way that maximises their efficiency and reduces waiting times for customers	Shall
UR_CUSTOMISATION	The player shall be able to customise their task organisation to fit their personal playing style	Shall
UR_CUSTOMER_LEAVES	Customers may leave if they wait too long or if they receive poor service	Shall
UR_REPUATION	The game should track the player's reputation, which can be improved or damaged based on customer satisfaction	Shall

UR_SCENARIO_MODE	A mode in which the player must complete a series of levels or challenges with increasing difficulty	May
UR_LOSE_GAME	The game should have a way for the player to lose, such as by running out of money or failing to meet customer demands	Shall
UR_ENDLESS_SCORE	The game should keep track of the player's score as they serve customers, even in endless mode	May
UR_FOOD_PREP	The player must be able to prepare food items for customers, such as chopping vegetables or grilling meat	Shall
UR_RECIPE	The game should include multiple recipes for the player to learn and master	Shall
UR_INGREDIENT_DOUGH	The player must have access to ingredients to add cheese to pizzas	Shall
UR_INGREDIENT_BEANS	The player must have access to ingredients to make bean-based dishes, mainly the jacket potato	Shall
UR_INGREDIENT_SAUCE	The player must have access to ingredients to make pizza sauce	Shall
UR_INGREDIENT_CHEESE	The player must have access to ingredients to add cheese to pizzas	Shall
UR_PIZZA_TOPPINGS	The player must have access to pizza toppings to add to pizzas	Shall
UR_WIN	The game must be won when the user has finished serving all the customers	Shall
UR_CHEF	All chefs should be unique	Shall
UR_INGREDIENT_POTATO	The player must have access to ingredients to make potato-based dishes, mainly the jacket potato	Shall

UR_RECIPE_PIZZA	The game should include recipes specifically for making pizza, which may include different crust types, sauce options, and toppings	Shall
UR_STATION	The player must be able to manage different stations in the restaurant, such as the pizza oven, the prep area, and the fryer	Shall
UR_GAMEPLAY	The game should be fun and engaging to play, with a variety of tasks and challenges to keep the player interested	Should
UR_DIFFICULTY	The game should provide a range of difficulty levels to accommodate players of different skills levels	Should
UR_POWERUP	The player may be able to earn power ups or bonuses that can help them serve customers more efficiently or increase their score	May
UR_LEADERBOARD	The game should include a leaderboard where players can compare their scores and achievements with others	May
UR_FOOD_QUALITY	The quality of the food prepared by the player should affect customer satisfaction and reputation	Should
UR_MONEY	The game should allow the user to make money on the completion of an order based on the speed of completion	Shall
UR_TIME	The game should be be fast paced and the time to complete an order should not be too long	Shall
UR_CONVENTIONS	The game should follow standard conventions	May
UR_COLLISION	The game should not allow the chefs to phrase through each other	May
UR_INACTIVITY	The game should play a pre-recorded demo when the user is inactive for a given time	Shall

Functional Requirements:

ID	Description	User Requirements
FR_CUSTOMER_GROUPS	The system shall increase the odds of customers arriving in pairs or groups of 3 as the game progresses	UR_CUSTOMER_ARRIVAL
FR_CUSTOMER_TIMES_OUT	The customer shall leave after waiting too long and a reputation point shall be deducted	UR_CUSTOMER_LEAVES UR_REPUTATION
FR_CUSTOMER_INTERVALS	The system shall randomly generate a time interval between customer arrivals, with some intervals shorter or longer than others, to add minor challenge to the game	UR_CUSTOMER_ARRIVAL UR_CHALLENGE
FR_SCENARIO_LOSE	The system shall award the player with a loss if rep points reach 0	UR_SCENARIO_MODE UR_LOSE_GAME
FR_ENDLESS_MODE	The system shall infinitely send more customers to be served in increasing frequency	UR_ENDLESS_MODE
FR_ENDLESS_SCORE	The system shall keep track of the amount of customers served and the highest number served in an endless run	UR_ENDLESS_MODE UR_ENDLESS_SCORE
FR_ENDLESS_END	The system shall end the endless mode when rep points reaches 0	UR_ENDLESS_MODE UR_LOSE_GAME
FR_PREP_FAIL	The system shall require a prep step is repeated should it be failed	UR_FOOD_PREP
FR_RECIPE_PIZZA	The system shall require that a pizza is made by through multiple steps	UR_RECIPE, UR_INGREDIENT_DOUGH, UR_INGREDIENT_SAUCE, UR_INGREDIENT_CHEESE, UR_PIZZA_TOPPINGS
FR_RECIPE_JACKET_POTATO	The system shall require that a jacket potato is made by through multiple steps	UR_RECIPE, UR_INGREDIENT_POTATO, UR_INGREDIENT_CHEESE, UR_INGREDIENT_BEANS

FR_INGREDIENT_DOUGH	The system shall provide a dough ingredient for the pizza recipe	UR_RECIPE_PIZZA
FR_INGREDIENT_SAUCE	The system shall provide a sauce ingredient for the pizza and jacket potato recipe	UR_RECIPE_PIZZA
FR_INGREDIENT_CHEESE	The system shall provide a cheese ingredient for the pizza and jacket potato	UR_RECIPE_PIZZA, UR_RECIPE_JACKET_POTATO
FR_PIZZA_TOPPINGS	The system shall allow for various pizza toppings to be added	UR_RECIPE_PIZZA
FR_INGREDIENT_POTATO	The system shall provide a potato ingredient for the jacket potato recipe	UR_RECIPE_JACKET_POTATO
FR_INGREDIENT_BEANS	The system shall provide a beans ingredient for the jacket potato recipe	UR_RECIPE_JACKET_POTATO
FR_STATIONS	The system shall provide cooking stations for the cooking stations	UR_STATION
FR_TIME_LIMIT	The system shall deduct reputation points if orders take too long to be served	UR_REPUTATION, UR_GAMEPLAY
FR_ENDLESS_DIFFCULTY	The system shall adjust the frequency of customer groups and order times based on the chosen difficulty	UR_ENDLESS_MODE, UR_DIFFICULTY
FR_MENU	The system shall provide a menu for customers to choose from	UR_GAMEPLAY
FR_BURGER_FLIPPING	The system shall allow the player to flip burgers	UR_GAMEPLAY
FR_CUSTOMER_COUNTER	The system shall display the number of customers served	UR_GAMEPLAY
FR_MONEY	The system shall provide in game money	UR_GAMEPLAY UR_INVEST_MONEY UR_MONEY
FR_SPEED_POWERUP	The system shall provide a powerup that increases movement speed which can be bought with money	UR_POWERUP UR_INVEST_MONEY UR_MONEY

FR_COOKING_TIME_POWER UP	The system shall provide a powerup that increases movement speed which can be bought with money	UR_POWERUP UR_INVEST_MONEY UR_MONEY
FR_INSTACOOK_POWERUP	The system shall provide a powerup that allows instant cooking for 10 seconds which can be bought with money	UR_POWERUP UR_INVEST_MONEY UR_MONEY
FR_NO_BURN_POWERUP	The system shall provide a powerup that prevents burning during preparations which can be bought with money	UR_POWERUP UR_INVEST_MONEY UR_MONEY
FR_UNLOCK_ALL_POWERU P	The system shall provide a powerup that unlocks all stations or provides an extra chef	UR_POWERUP UR_INVEST_MONEY UR_MONEY
FR_REPUTATION	The player shall start with 3 reputation points	UR_REPUTATION
FR_LEADERBOARD	The system shall keep track of the highest scores and display a leaderboard table at the end of the game	UR_LEADERBOARD
FR_UNLOCK	The system shall provide the means to unlock more stations, recipes and staff	UR_UNLOCK UR_INVEST_MONEY UR_MONEY
FR_CARRYING_STACK	The system shall allow the player to carry a stack of items at once, instead of just one time at a time, to speed up serving customers	UR_CARRYING_STACK
FR_ORGANISE_BETTER	The system shall allow the player to organise the order in which tasks are completed to optimise their efficiency and serve customers more quickly	UR_EFFICIENCY UR_CUSTOMISATION
FR_CHEF_STATION	The system shall require the player to ensure the chef remains at the station until the food item has been fully prepared to prevent overcooking or burning.	UR_FOOD_QUALITY UR_REALISM
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 to the chef	UR_PLAYABLE

FR_CONTROLS	The controls decided should be accessible and easily understood even by people who aren't familiar with videogames	UR_PLAYABLE UR_CONVENTION
FR_SWITCH	The system must allow the use to switch chefs	UR_PLAYABLE
FR_INTERACT	The system must let the user interact with the kitchen	UR_STATION UR_PLAYABLE
FR_CONTINUE_ACTION	The system will allow the non-controlled chef to continue their previous action. Eg, continue chopping	UR_PLAYABLE UR_CHEF
FR_DEMO	The system will play a previously recorded gameplay that serves are the tutorial if left inactive for a given amount of time	UR_INACTIVITY
FR_COLLISION	The system will not allow the chefs to phrase through each other	UR_COLLISION UR_PLAYABLE

Non-Functional Requirements:

ID	Description	User Requirement	Fit Criteria
NFR_UNLOCK	The system shall be secure	UR_UNLOCK	The system shall provide a secure authentication mechanism for buying more staff based off the in-game currency
NFR_TIME_LIMIT	The system shall be time-sensitive	UR_REPUTATION UR_GAMEPLAY	The system shall have a time limit for completing orders
NFR_SCENARIO_ENDLESS	The system shall be scalable	UR_ENDLESS_MODE	The system will support an appropriate number of customers
NFR_MENU	The system shall be user-friendly	UR_GAMEPLAY	The system shall have a user-friendly menu that lists available items and options

NFR_STATIONS	The system will be modular	UR_STATIONS	The system shall have separate stations for baking and unlocking the shop, which must be modular and secure
NFR_POWERUPS	The system will be efficient	UR_POWERUP UR_INVEST_MONEY UR_MONEY	The system shall have five power-ups that increase speed, reduce cooking time, instacook for 10 seconds, no burn, and unlock all stations or extra chefs.
NFR_DIFFICULTY	The system will be adjustable	UR_DIFFICULTY	The system shall have different difficulty levels that change the odds of groups and order times
NFR_SAVE_GAME	The system will be persistent	UR_SAVE_GAME	The system shall allow users to save their progress
NFR_REPUTATION	The system will be accountable	UR_REPUTATION	The system shall deduct three reputation points for taking too long to complete an order, and all reputation points for losing overall.
NFR_CONTINUE_GAME	The system will be resumable	UR_CONTINUE_GAME	The system shall allow users to continue their progress
NFR_RESPONSIVE	The system will be responsive	UR_PLAYABLE	The response time should be less than one second after input
NFR_AVAILABLE	The system will be able run and not crash	UR_PLAYABLE	Uptime: 99% during game time

Constraint Requirements:

ID	Description
CON_PROGRAMMING_LANGUAGE	The system must be written in Java (version 11).
CON_OS	The system must be able to run on all desktop operating systems: Linux, Windows and MacOS

CON_SYS_REQUIREMENTS	The game must be able to run on any modern desktop
CON_RESOLUTION	The game must adapt to any screen size

References:

[1] “Requirements v2”, Team13ENG1, 2022-2023. [Online].Available: https://team13eng1.github.io/files/assessment/Requirements%20v2.pdf. [Accessed: May 1,2023].

[2] Somerville, I.(2016). Requirements Engineering. In Software Engineering (10th ed., pp. 81-118). Pearson.

[3] Somerville, I.(2016). Requirements Elicitation. In Software Engineering (10th ed., pp. 119-150). Pearson.

[4] Somerville, I.(2016). Requirements Specification. In Software Engineering (10th ed., pp. 151-188). Pearson.

[5] Somerville, I.(2016). Requirements Validation. In Software Engineering (10th ed., pp. 189-226). Pearson.