Requirements

Group 30		Triple 10

Team Members:

Kelvin Chen, Amy Cross, Amber Gange, Robin Graham, Riko Puusepp, Labib Zabaneh

User and System Requirements Introduction

Using the product brief, we initially discussed fundamental components that the software must have to fulfil customer demands. Once refined, these became the user requirements, each of which were then expanded to define the system requirements to detail how the user requirements would be met. Then, an interview was undertaken with the customer to determine other non-functional requirements that were not outlined in the original brief, as well as to answer any queries we had regarding the brief.

Having acquired any additional requirements from the client, we reorganised each requirement into either the user, functional and non-functional requirements table. This was done as a group so that each member could have their input on whether the requirement was necessary, and to ensure that the requirements were as detailed as possible. The requirements were also given IDs so that they could easily be referred back to throughout the project.

User Requirements

ID	Description
UR_MENU	The system shall provide a menu for the player to begin a scenario and which is returned to once complete
UR_COOK_ CONTROLS	The system shall allow the player to switch between 2 cooks to help prepare dishes
UR_CUSTOMER_ ORDER	The system shall allow customers to arrive at the counter and demand a dish
UR_GAME_MAP	The system shall display a game map of a kitchen consisting of work stations
UR_PREP	The system shall allow the player to carry out a series of tasks to fulfil orders
UR_TIMER	The system shall keep track of how long it takes for a player to complete the scenario
UR_PANTRY	The system shall allow the player to access ingredients via a pantry
UR_ACCESSIBILITY	The system should be accessible for all users
UR_ENJOYABILITY	The system should be engaging and enjoyable

Functional Requirements

ID	Description	User Requirements
FR_MENU	The system shall display a main menu that is initially loaded when the program is started	UR_MENU
FR_START	An option shall be provided to begin the scenario	UR_MENU
FR_CONTROL_COOK	The player shall not be able to control both cooks at one time	UR_COOK_ CONTROLS
FR_SHOW_COOK	The system shall highlight which cook is currently being controlled by the player	UR_COOK_ CONTROLS
FR_CHANGE_COOK	The system shall allow the player to begin a task with one cook, then switch to the other without affecting the continuation of said task	UR_COOK_ CONTROLS
FR_ORDER	The system shall limit each customer to a single order of either a salad or burger	UR_CUSTOMER_ ORDER

FR_ARRIVAL	The customers shall arrive one by one at random intervals and wait in a queue indefinitely	UR_CUSTOMER_ORDER
FR_IDLE	The player is never left idle	UR_CUSTOMER_ORDER
FR_ACCEPT_ORDER	The player shall physically accept a customer order	UR_CUSTOMER_ORDER
FR_NUM_CUSTOMERS	The system shall keep track of the number of customers arrived to ensure no more than 5 arrive	UR_CUSTOMER_ORDER
FR_DISPLAY_ORDER	The system shall display current customer orders for the player to easily see	UR_CUSTOMER_ORDER
FR_RECIPES	The system shall allow the player to access the recipes for different dishes in some way	UR_CUSTOMER_ORDER
FR_MULTI_ITEM	The player shall be able to handle multiple ingredients at one time at a single workstation	UR_GAME_MAP
FR_INGREDIENTS_LIMIT	The player shall only be allowed to place ingredients that they are currently carrying on a workstation	UR_GAME_MAP
FR_STEP_VALIDATION	Each preparation step shall require input from the player and the correct ingredient(s)	UR_PREP
FR_COOK_BUSY	A cook shall carry out a preparation step for a given period of time, making this cook inoperable until the task is complete	UR_PREP
FR_DISH_VALIDATION	The system shall provide verification of a dish when served to check if correct	UR_PREP
FR_TIMER	The system shall display a timer on the screen of the player’s progress	UR_TIMER
FR_INGREDIENTS_INFINITE	The system shall allow the player an infinite number of each ingredient	UR_PANTRY
FR_DISCARD	The system shall allow the player to discard any ingredients mistakenly accessed	UR_PANTRY
FR_INVENTORY	The system shall limit the player to how many ingredients can be stored at once	UR_PANTRY
FR_PANTRY	The player shall choose which ingredients to take from the pantry	UR_PANTRY

Non-Functional Requirements

ID	Description	User Requirements	Fit Criteria
NFR_SCENARIO_TIME	The scenario should have a fast paced rhythm	UR_TIMER	The scenario will take around 5 to 6 minutes
NFR_PLATFORMS	The product shall be accessible on a range of platforms	UR_ACCESSIBILITY	The system will support Windows/Linux
NFR_SCALABILITY	The product shall be scalable to accommodate reasonable screen sizes	UR_ACCESSIBILITY	720p/ 1080p resolution
NFR_CONTROL_SCHEME	The user shall be able to select their preferred control scheme	UR_ACCESSIBILITY	The system implements at least 2 control schemes
NFR_SPEED	The system shall minimise run time and memory usage	UR_ENJOYABILITY	<20 milliseconds
NFR_INSTRUCTIONS	The system shall be simplistic and provide instructions to support the player	UR_ENJOYABILITY	Having read the instructions 95% players will understand the controls and aim of the game