Black Box Testing: Test Cases:
In this document, we will outline and give the results of all black box tests which we carried out where automatic testing was not appropriate, giving the related requirement. We carried these tests out as the features were developed, supplementary to whichever automatic tests we were able to design in order to improve our test coverage and test the game more intuitively.

Requirement: UR_CONTROL_COOKS
To test this requirement, we will confirm that all three chefs are able to be selected, and the selected chef is able to be moved using the WASD keys. A passing result will allow for movement, and the selected chef sprite should be displayed in the top left corner.. An edge case is the case where no chef is selected: pressing WASD should have no effect unless a chef is selected.
Result: PASS. All chefs can be selected and moved. Erroneous inputs, such as incorrect keyboard characters, have no effect on the game state.

Requirement: UR_INGREDIENTS
In the final game, there are 6 ingredient stations, all of which must dispense the corresponding ingredient, with this ingredient being displayed in the chef's hand once dispensed. To pass this test, all stations must exhibit this correct behaviour, and all chefs must be capable of holding the ingredients.
Result: PASS. All ingredients are correctly dispensed

Requirement: UR_COOK_FOOD, UR_SERVE_FOOD
The recipe station must be able to convert ingredients into the appropriate recipe. This can be tested by ensuring that every ingredient combination corresponds to the correct recipe, and that all other combinations which do not result in a recipe have no effect. We will test this by systematically checking every recipe.
Result: PASS. All recipes are able to be successfully created and submitted. Erroneous inputs have no effect.

Requirement: UR_CUSTOMERS
To test this requirement, we must check that there is indeed a fixed number of customers. We will test this by playing through the game and ensuring that only 5 orders are received (with the exception of the newly implemented endless mode)
Result: PASS, only 5 orders were received.

Requirement: UR_FAILING_STEPS
It is a new requirement that the player shall be able to overcook ingredients which require the oven (patties, pizza bases, potatoes). To test this, we will cook all of these ingredients and ensure that the burning mechanic is triggered.
Result: PASS, all ingredients burnt after the specified time period.

Requirement: UR_WRONG_INGREDIENT
To implement this requirement, a clear button was implemented, which will remove whichever items the currently selected chef is holding. To test this, we tested that the correct action occurred both with ingredients and recipes.
Result: PASS, the clear button works for all ingredients and recipes.

Requirement: UR_CURRENCY
To test this requirement, we will ensure that the logic behind the currency system is correct (through unit testing) and test that the station upgrades are working by attempting the action, both with sufficient funds and without.
Result: PASS. When the user has enough coins for an upgrade, the appropriate station is unlocked, and if not then there is no action.

Requirement: UR_DIFFICULTY
To test this requirement, we must ensure that there are tangible differences between the three difficulty levels
Result: PASS In play testing, we did find the harder levels more difficult, which we were able to see as it took us longer to complete the later level.

Requirement: UR_POWERUPS
This is another new feature which we implemented in the second phase. It was important to test the logic behind the powerups, which we did using unit tests, and then test the activation of the powerups
Result: PASS, automated unit tests were passed and the deployment is also working

Requirement: UR_MODES
This is another new requirement, which we will test by ensuring that the menu works, and that the endless mode will end when reputation reaches 0, as opposed to when all recipes are completed.
Result: PASS, endless mode is correctly implemented.