

Engineering 1 Group Assessment 2
Change Report
Cohort 3 Group 23

Team Members:

Harry Draper
Seif Hussein
Tikhon Likhachev
Thomas Maalderink
Joshua McKean
Sebastian Armstrong

Summary of Change Tracking

In order to keep track of changes to the Assessment 1 deliverables, we used a combination of methods and tools to ensure that our descriptions of the changes were accurate. We created a new Discord channel to update the other team members of any changes that were made to the documents and to ensure we remembered each change to add them to the report. We used this in combination with Google Docs' built in change tracking feature to verify the changes so that we would have an accurate description and report of the changes that were made.

To keep track of changes to the code, we communicated with other members of the team frequently to ensure that everyone was aware of the updates made. Additionally, we used GitHub to verify any changes we reported to make sure that the information on the change report was accurate.

Requirements

Introduction:

- No changes made as there is relevant research done by the previous team which is helped our team to generate the new requirements

How requirements were elicited and negotiated:

- The method of acquiring requirements was not needed in assessment 2 because the information from the original requirements collected by the previous team was sufficient, therefore, this was left unchanged.

User requirements:

- No changes were made to the layout of the requirements because our team found them clear enough to be used again.

- 16 New user requirements were added to the table and 1 original was modified along with their descriptions:

- Updated the UR_COOKS Requirement from 2 cooks to 3.
- Added the UR_MODE_SELECT user requirement to allow the selection of the game mode.
- Added the UR_MODE_ENDLESS user requirement for the endless mode
- Added UR_EARNING where cooks earn money.
- Added UR_UNLOCK for investing in cooking stations
- Added UR_GAME_SAVE so users can save game state
- Added UR_GAME_LOAD so users can load saved games
- Added UR_GAME_DIFFICULTY so users can select difficulty
- Added UR_REPUTATION so cooks can gain and lose reputation
- Added UR_HARD_MODE so users can play hard mode
- Added UR_NORMAL_MODE so users can play normal mode

- Added UR_EASY_MODE so users can play easy mode
- Added UR_POWER_UPS so users can use power ups
- Added UR_SCORE_POWER to increase score by 1
- Added UR_MONEY_POWER to increase money by 5
- Added UR_REPUTATION_POWER to increase reputation by 1
- Added UR_CUSTOMER_POWER to decrease customer spawn rate for a period of time
- UR_ORDER_POWER to fulfil an order without the need to cook it

System requirements – Functional:

- 17 New user requirements were added to the table and 3 original was modified along with their descriptions:

- Updated the FR_COOKS Requirement from 2 cooks to 3.
- Updated FR_RECIPE to include all the recipes
- Added the FR_MODE_SELECTION user requirement to allow the selection of the game mode.
- Added the FR_MODE_ENDLESS user requirement for the endless mode
- Added FR_EARNING where cooks earn money.
- Added FR_UNLOCK for investing in cooking stations
- Added FR_GAME_SAVE so users can save game state
- Added FR_GAME_LOAD so users can load saved games
- Added FR_DIFFICULTY_SELECTION so users can select difficulty
- Added FR_REPUTATION so cooks can gain and lose reputation
- Added FR_HARD_MODE so users can play hard mode
- Added FR_NORMAL_MODE so users can play normal mode
- Added FR_EASY_MODE so users can play easy mode
- Added FR_POWER_UPS so users can use power ups to help user
- Added FR_SCORE_POWER to increase score by 1
- Added FR_MONEY_POWER to increase money by 5
- Added FR_REPUTATION_POWER to increase reputation by 1
- Added FR_CUSTOMER_POWER to decrease customer spawn rate for a period of time to make the game easier for the user.
- FR_ORDER_POWER to fulfil an order without the need to cook it.

System requirements – Non-Functional:

- 2 new non-functional system requirements have been added to the table with their descriptions, risks and user requirements. These are:
- NFR_DIFFICULTY for the user to be able to tell they are playing different difficulties
- NFR_LOAD for the game not to crash while loading saved games

Architecture Changes

Software Architecture:

- Mostly unchanged apart from specifying that a plantUML plugin for IntelliJ was used to generate the new class diagrams for assessment 2.

Abstract Representation:

- No changes were made to the UML component diagrams as they still accurately represented the classes for the UI because there was no changes needed in the corresponding classes

Concrete Representation:

- The original UML class diagrams displaying the inheritance and relationships between classes were replaced by a new class diagram which covers the new classes and parts of the implementation not covered by the original ones. This covers each part of the project including: Game, Stations, Maths, Physics and UI.

Abstract Architecture Justification:

- No changes were made as the previous team's justification and architecture was the same as our teams.

Concrete Architecture Justification:

- No changes were made as the previous team's justification and architecture was the same as our teams.

Justification relating to Requirements:

- Added the following to how each requirement was satisfied in the concrete architecture:
 - **UR_HARD_MODE**
 - **UR_NORMAL_MODE**
 - **UR_EASY_MODE**
 - **UR_GAME_DIFFICULTY**
 - **UR_GAME_SAVE**
 - **UR_GAME_LOAD**
 - **UR_EARNING**
 - **UR_UNLOCK**
 - **UR_POWER_UPS**
 - **UR_SCORE_POWER**
 - **UR_MONEY_POWER**
 - **UR_REPUTATION_POWER**
 - **UR_CUSTOMER_POWER**
 - **UR_ORDER_POWER**
 - **UR_MODE_SCENARIO**
 - **UR_MODE_ENDLESS**

Method Selection & Planning Changes

Methods and Tools:

- Most of this is unchanged as most of the previous teams methods and tools are the ones our team used, although we didn't use Jekyll for the website in Assessment 1 but we did use it for assessment 2 for continuity.
- However we specified that we used plantUML in IntelliJ for some of the diagrams

Team Organisation Approach:

- This section has been changed to represent how the team works and progresses the project.
- Roles and responsibilities were assigned to each team member to reflect each person's work toward the project.
- The organisation through discord is entirely unchanged as discord was used by our team also to set up meetings and continue working.

Project Planning:

- The old Gantt chart has been removed and replaced by a new one relevant to assessment 2 so that the tasks and requirements were clearer.
- The review of the base plan was changed, but the use of agile development methodology is unchanged.
- Key tasks were changed so they would be relevant to assessment 2

Risk Assessment & Mitigation Changes

Introduction:

- Unchanged as it aligns with how our team would introduce the risk assessment.

Risk Management Process:

- Unchanged our team agrees with the previous teams risks and justifications

Risk Mitigation:

- Unchanged as our team will be using the same process to mitigate risks

Risk Table/Risk Monitoring

- Changed the owners of each risk to a member of our team.
- Added one risk where the team runs out of time to complete the project