Requirements

Group 17

Team Loading

Joel Warren
Charlie Wilson
Jake Keast
Hari Bhandari
Ishrit Thakur
Joshua Hills

Requirements 2. a)

Requirements elicitation and negotiation

Requirements elicitation was carried out in 4 main stages:

1. Requirements discovery and understanding
2. Requirements classification and organisation
3. Requirements prioritisation
4. Requirements documentation

To begin understanding requirements, we analysed the product brief for 'Piazza Panic', eliciting different requirements from this. We then conducted a meeting with our stakeholder to negotiate further requirements from vague parts of the product brief and the stakeholder themself.

Now that we had an unstructured collection of requirements, we then classified related requirements together in coherent groups, under the categories:

- Functional requirements
- Non-functional requirements
- User requirements

Based on our negotiations with the stakeholder and the product brief, we then prioritised the user requirements, under the priorities shall, should and may be satisfied.

We then tabulated all this information in the statement of user and system requirements. Each requirement type was presented in its own table, with the related description, priority and fit criteria, with both the functional and non-functional requirements tables also tracing the related user requirements.

Requirements presentation reasoning

The requirements are presented in this manner to clearly distinguish between the 3 types of requirements. We gave each requirement a unique ID, to allow clear traceability. Both the functional and non-functional requirements tables also have a column to trace which user requirements that requirement traces to, again to support traceability. ID's had a prefix of FR/NFR/UR to clearly identify which category the requirement falls under. Furthermore, we gave IDs meaningful names to improve readability, as "UR_GAME_MODE" for example requires less explanation than e.g "UR1.1". For subsequent assessment deliverables, this referencing system may need to be updated.

Research carried out

We studied the "Requirements Engineering" lecture provided by our lecturers, and also read chapter 4 of 'Software Engineering' (Sommerville, Ian), which inspired our requirements elicitation process.

Requirements 2. b)

User Requirements

ID	Description	Priority
UR_UX	System shall offer a simplistic and enjoyable user experience.	Shall
UR_COOK_SWITCH	System shall allow users to switch between cooks.	Shall
UR_COOK_CONTROL	System shall allow users to move the cook and interact with the environment.	Shall
UR_INVEST	System shall allow users to invest their earnings into more cooks/stations.	Shall
UR_GAME_MODE	System shall allow the user to select a gamemode.	Shall
UR_ORDER_RECIEVE	System shall allow users to make a dish based on a customer order.	Shall
UR_ORDER_COMPLETE	System shall allow users to complete/deliver an order to a customer.	Shall
UR_NEW_USER	System should help users with no prior experience.	Should

Functional Requirements

ID	Description	User Requirements
FR_IDLE	If no game is in progress and the user is idle, simple gameplay should run on screen.	UR_NEW_USER
FR_CUSTOMER_ARRIVE	Customers should arrive at certain intervals and place an order.	UR_ORDER_RECEIVE
FR_CUSTOMER_WAIT	Customers should wait for their order at the counter for a certain amount of time.	UR_ORDER_RECIEVE
FR_ORDER_SERVED	Once an order is made, the customer should be satisfied and leave.	UR_ORDER_COMPLETE
FR_ORDER_INCOMPLETE	If an order is not made on time, the user should lose reputation and the customer should leave.	UR_ORDER_COMPLETE
FR_CUSTOMER_GROUP	After a certain amount of time, customers should arrive in pairs/groups.	UR_ORDER_RECEIVE
FR_PATTIES_BURN	If a patty is not flipped on time, the patty will burn, requiring a new patty.	UR_COOK_CONTROL
FR_PANTRY	Cooks (controlled by the user) can interact with the pantry to obtain ingredients from ingredient stations.	UR_COOK_CONTROL
FR_COOKING_STATIONS	Cooks can interact with different cooking stations to perform the actions: chop, flip, grab (an ingredient) or place what they are carrying.	UR_COOK_CONTROL

Non-Functional Requirements

ID	Description	User Requirements	Fit Criteria
NFR_LIBRARY_LEGALITY	All public libraries used in development are legal to use.	N/A	No public library licence is broken.
NFR_COLOUR_USAGE	User interface should be black & white, not relying on colour.	UR_UX	The interfaces' colour is black & white.
NFR_INTERFACE_TEXT	All user facing text should be minimal and kept to plain english.	UR_UX	Any user facing text is in plain english.
NFR_BEGINNER_EXPERIENCE	Users with no prior experience should be able to play the game.	UR_NEW_USER	Game is playable by new players.
NFR_PRINTABLE_MANUAL	System shall be accompanied by a printable manual that details all its functions.	UR_NEW_USER	System manual is accessible and printable.
NFR_MAINTAINABILITY	The system should be maintainable and workable for another group	N/A	**

** Maintainability is difficult to determine fitness since it is based on the response of another group, and incorporates many areas. In general, it can be subjectively measured by areas such as traceability in requirements and architecture, as well as concise well-written code.