January 27, 2017

Cupboard: Requirements Specification

Team 8:

Clare Doran, Abdul Ghani, Ucizi Mafeni, Luke Needham, Soumya Singh

Summary:

A web app built with the intention of helping reduce food waste.

Contents

1 INTRODUCTION

7.5 million tonnes of food goes to waste in 2015, 4.4 million tonnes of which is edible. This has a retail value of around £13 billion. This is associated with 19 million tonnes of carbon dioxide. emmisions, which is around a quarter of that emitted by road traffic in the UK[1]. Meanwhile, many people throughout the country struggle to put together enough money for food, with food banks becoming busier every year [2].

From an environmental, economical, and ethical point of view this is a travesty.

Cupboard aims to be a system that allows users to painlessly find and share food that would otherwise go to waste. A priority is speed and ease of use, as it is too easy to just throw away food. It will also allow users to arrange collection in a manner that does not force them to disclose their house address or any other details they wish to keep private.

The main sections of this document are:

- **Project Scope:** Here we give an overview about what our system will (and won't) do.
- **Domain analysis:** An analysis of similar products already in place.
- **Proposed Deliverables:** How we plan to undergo the project.
- **Identified, Risks, Assumptions, Dependencies and Constraints:** A discussion of things we must consider and keep in mind as we move forward.
- **Solution Requirements:** Planned functional and non-functional requirements.
- **Development Approach:** A brief overview of the technologies we plan to use in order to develop the product.

2 PROJECT SCOPE

The aim is to become a go-to everyday service that people will use 'by default' without even thinking about wasting food. This means users will be able to easily list items with a minimal amount of effort.

There will be a simple points-based system and a ranking system in order to encourage sharing of food and to reassure other users that the quality of food is likely to be good. When some food has been successfully exchanged both parties will get points and will be able to rate each other. If users wish to participate, there will be a simple public leaderboard.

Users can set their dietary requirements (allergies, religious etc.) and have results automatically filtered.

There will be a messaging system (in order to arrange collection) and a comments system (in order to view interest in the item and ask questions).

The aforementioned rankings, messaging, and comments systems are purely to facilitate the sharing of food. We wish to avoid any sort of 'social media' elements (such as friend requests, status updates etc.) in order to not lose sight of our main goal - the prevention of food waste.

3 DOMAIN ANALYSIS

The domain of our project encompasses online marketplaces and sites for purchasing food. There are a number of sites which already occupy these domains, the design of which we will draw from in designing our own site.

- eBay [3]: An online marketplace through which users can both list and purchase almost anything. When listing an item, users must provide at least one photo and information such as title, condition and type, as well as an optional description and additional photos.

  This is important as it forces users to provide enough information for buyers, and discourages fake listings. Users search for items by name, and can sort results in a number of ways, which is very useful, however it lacks more options for filtering results.

  Perhaps the best feature of the site is that every listing displays the seller's username, along with a score and the percentage of all the feedback the user has received which is positive. This allows buyers to gauge how trustworthy the seller is, and is a feature which we feel will work well in our system also.

- foodsharing.de [4]: A food sharing sharing site, currently only available in Germany. It reduces food waste by allowing users to list food they wish to give away, and then arrange a time and place for the volunteer "Lebensmittelretter Innen" to collect the food and deliver it to whoever wanted it. This method removes the risk of potentially dangerous or illegal substances being exchanged, as all items are checked by the trusted volunteers.

  However it introduces the problem of having to have a large number of volunteers willing to give up their time. The site itself is quite well designed, but navigation is sometimes difficult.

  Every item of food listed has its own page, but we found these pages to be lacking, listing only a very brief description most of the time. The listings are also all anonymous and there is no system of rating users, as in eBay for example, which would allow users to check how trustworthy the user providing the food is.

- Olio [5]: A mobile app with a strong UK presence. Users can share food by manually collecting/delivering it, or through "Drop Boxes" - local stores or cafés where users can drop off food they'd like to be shared.

  However the app has received many negative reviews from users frustrated that there is no food they can get near them, which is an issue our system is also likely to face, particularly soon after launch.

An interesting feature of the 2 existing food sharing apps listed above is that their search functions and catalogs are strongly based on location. They place a lot of emphasis on an interactive map to find items, contrary to eBay's method of having the user search for items by name, and then allowing the user to filter results, indicating perhaps that users are more concerned about how far they have to travel for food, rather than what that food is.

We feel our system should combine both of these functions, with a search bar and filtering methods like eBay, but with results sorted by distance from the user.

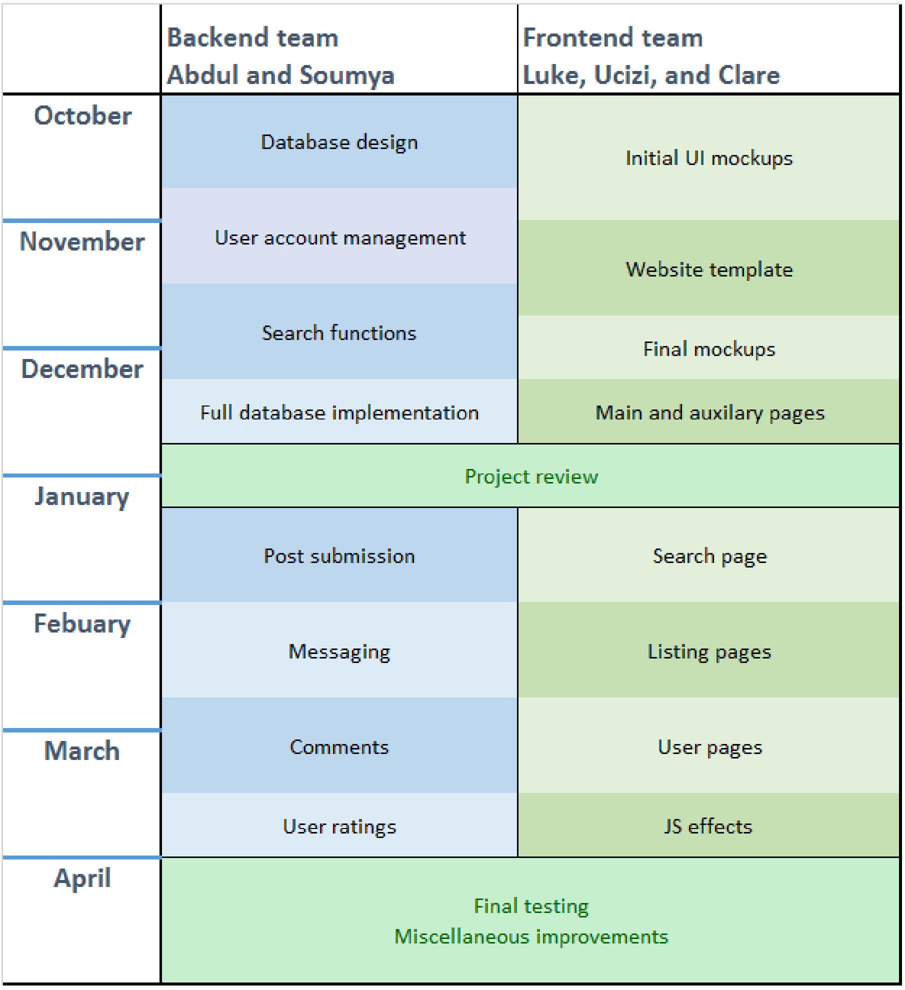
4 PROPOSED DELIVERABLES

4.1 DIVISION OF TASKS

Though all members shall inevitably contribute to both sides of the task, the major components of the project have been split as follows:

Task	Subtask	Team Member
Frontend	Website Layout Wireframe	Luke
	Logo and Branding Design	Clare
	HTML/CSS/Bootstrap	Ucizi
	Dynamic Features (Javascript and JQuery)	Ucizi
Backend	User Account — Registration and Login	Abdul
	Search	Soumya
	Listings/Posts	Soumya
	Messaging	Abdul
	Comments	Soumya
	User Ratings and Scores	Abdul
	Database Design and Implementation	Abdul & Soumya

4.2 MILESTONES AND DEADLINES



	Backend team Abdul and Soumya	Frontend team Luke, Ucizi, and Clare
October	Database design	Initial UI mockups
	User account management	
November		Website template
	Search functions	
December		Final mockups
	Full database implementation	Main and auxilary pages
January	Project review	
	Post submission	Search page
Febuary	Messaging	Listing pages
March	Comments	User pages
	User ratings	JS effects
April	Final testing Miscellaneous improvements	

FIGURE 1: PROJECT GANTT CHART

Date	Task
30/10/16	Database Design Initial UI Mockups
15/11/16	User Account Management Website Template
16/12/16	Search Functions Final Mockups Full Database Implementation Main and auxiliary pages
31/01/17	Post Submission Search Page
24/02/17	Messaging Listing Pages
16/03/17	Comments User Pages
31/03/17	User Ratings JS Effects

TABLE 1: HARD DEADLINES

5 IDENTIFIED RISKS, ASSUMPTIONS, DEPENDENCIES AND CONSTRAINTS

5.1 RISKS

- It is possible that users may exploit the rating system, through the use of fake reviews and ratings, to boost their score and rating, and therefore trick other users into believing they are trustworthy when this is not the case.
- Users may exploit the system to list and distribute non-food items, and particularly nefarious users may even use it to distribute illegal substances.
- Users may (knowingly or otherwise) distribute food which is dangerous to consume, such as if it has gone off.
- Users may lure others, under the guise of giving them food, into potentially perilous predicaments.
- If the system's sole database goes down the whole site will be useless, since all the data is stored in a single place.
- It is possible that the system, particularly its databases, could be hacked. This could lead to user's private data, particularly sensitive information such as address and email address, being leaked and abused.

Should any of these risks occur, or should anything else unfortunate occur to anyone, but particularly minors, through use of the system, we may be held accountable. This in itself is also a risk. To mitigate this, we will ensure that we mitigate all other risks as much as possible, and force users to sign agreements waiving all responsibility from us. The ways we mitigate these risks are detailed in Functional Requirements.

5.2 ASSUMPTIONS & DEPENDENCIES

- We assume Google maps will be able to find the vast majority of addresses entered by the user. This location functionality is thus completely dependent on the Google Maps API.
- We assume that when a user searches for something they will find a decent number of results. This is dependent on enough users listing items on the system, otherwise, the system will be useless to people wanting to receive food.

5.3 CONSTRAINTS

- We need to ensure that no user's private information, including address and email address, is unnecessarily stored or made public, in order to avoid putting their security at risk.
- Due to the risk involved in regarding strangers meeting to collect items, we will restrict our app to users who are 18 years and older
- In order to allow it to run quickly even on devices with poor internet, the system will be limited in the number of performance-detrimental luxuries it can have.
- Due to a lack of time and relative lack of experience, the system will likely not be as attractive or efficient as it would be ideally.

6 SOLUTION REQUIREMENTS

6.1 FUNCTIONAL REQUIREMENTS

N.B. Due to the highly time-sensitive nature of the project, any low priority requirements (and thier linked dependencies) may not be implemented. As such, all low priority requirements are inherently unessential to the system, and it will remain fully operational with or without them.

FR 1: "About Us" page

Description	All users should have access to a page that contains information about the web app. This includes: • A short description about the purpose of the app • Contact information • Terms and Conditions of use
Priority	Medium
Dependencies	None
Expected results	User able to access the page. Page contains all the information stated in the description.
Exception Handling	

FR 2: Error-Reporting System

Description	Users should be able to report any errors they notice to the webmaster. Each error report should have: • A title, which is a brief description of the error (max. 50 chars) • A more detailed description of the error (max. 500 chars)
Priority	Medium
Dependencies	FR 1
Expected results	User can successfully send error report
Exception Handling	**User cannot send error report:** use email on "About us" page to contact webmaster

FR 3: User Feedback System

Description	Users should be able to provide feedback regarding their experience using the web app to the webmaster. This should be done in the form of a feedback form, which should have the following fields: • A title, which is a brief description of the feedback report(max. 50 chars) • The main feedback report (max. 1000 chars)
Priority	Low
Dependencies	FR 2 (Error Reporting)
Expected results	User's can successfully send feedback report
Exception Handling	**User unable to send feedback form** : report error to webmaster

FR 4: User sign up

Description	The User should be able to create an account by following the registration process. Registration will involve: • Getting the user registration details (see FR 5). • Getting the user to accept that they are at least 18 years old. • Getting the user to agree to the Terms and Conditions of use (see FR 1).
Priority	High
Dependencies	FR 2, FR 5 (Get user registration details), FR 1 (About us page)
Expected results	Following account creation, the user will be able to login using their provided username and password
Exception Handling	**User unable to create account:** report error to webmaster

FR 5: Get User Registration Details

Description	During sign-up, the user should provide the following details in order to successfully complete the sign-up process: • username (must be unique) • email address • password • physical address • post code (must be valid) In addition, the user should also be given the option to fill in their Dietary Requirements and Allergy Information. However, it is not mandatory to fill in these details in order to complete registration.
Priority	High
Dependencies	FR 2, FR 6 (Validate User Details), FR 11 (Dietary Requirements), FR 12 (Allergy Information)
Expected results	User inputs all valid fields and proceeds to complete registration
Exception Handling	**User unable to fill in registration details:** report error to webmaster **User inputs invalid details:** should be validated as described in FR 6

FR 6: Validate Registration Details

Description	Information in the registration form should be validated as follows: username: checked against database to see if its unique email-address: adheres to general email structure (<someuser>@<somehost>.<com/net...>) password: minimum length of 7 characters, alphanumeric, contains one symbol from valid symbol list (refer to glossary) postcode: is actual valid postcode (check in UK postcode directory) If there exists an invalid input on submission, the form must be rejected and the user must be notified of their error (i.e highlight incorrect field and place error message next to it)
Priority	High
Dependencies	FR 2
Expected results	Validation system correctly highlights any invalid fields
Exception Handling	**Validation system doesn't meet specified requirements:** report error to webmaster

FR 7: Account Activation

Description	Once the user has provided the registration details, an email with an account activation link should be sent to the provided email address. The user should then be able to follow the activation link in order to activate their account.
Priority	Low
Dependencies	FR 2
Expected results	After the user follows the activation link, they should be able to login to their account using the login details they provided during registration
Exception Handling	**Incorrect email address provided:** the user should have the option to correct the provided email address. **Activation link doesn't work:** the user should have the option to request another activation link. **Activation system doesn't work:** report error to webmaster

FR 8: Password Reset Facility

Description	If a registered user forgets their password, they should be able to request a password reset link. They should provide a valid email address within the request. If the provided email address is linked to an account on the system, a password reset link should be sent to that email. Upon following the password reset link, the user should be able to set a new password.
Priority	Low
Dependencies	FR 6
Expected results	User able to successfully reset password
Exception Handling	**User unable to reset password:** report error to webmaster. **Email doesn't meet validation standards set in FR 6:** Prevent submission of password reset request. Notify user that invalid email was entered.

FR 9: User Dashboard

Description	Following login, all registered users should have access to a personal dashboard where they can do the following: • edit their personal details • edit their current dietary requirements • edit their allergy information • view their user score and rating
Priority	High
Dependencies	FR 10 (Edit Profile), FR 11, FR 12, FR 13 (User Score), FR 15 (User Rating)
Expected results	User is able to access and edit all personal information from the dashboard
Exception Handling	**Registered and logged-in user can't access dashboard:** notify webmaster of error

FR 10: Edit Profile

Description	A user should be able to change their personal details this includes: • their username • their email address • their password • their dietary requirements • their allergy information All changes to personal details will be validated under the standards mentioned in FR 6. With regards to dietary requirements and allergy information, users will be able to select whichever categories listed in FR 11 and FR 12 that personally apply to them.
Priority	High
Dependencies	FR 6, FR 11, FR 12
Expected results	User able to successfully change personal details.
Exception Handling	**User unable to save changes:** Report error to webmaster **New details don't satisfy validation standards:** Invalid fields must be highlighted with hint (as to why field is Invalid) shown

FR 11: Dietary Requirements

Description	Dietary requirements should be split into the following categories: • Halal • Kosher • Vegan • Vegetarian There should be an additional category group called "other" where users can specify a dietary requirement not listed in the default categories.
Priority	High
Dependencies	FR 2
Expected results	Dietary requirements meet specified requirements.
Exception Handling	**Dietary requirements don't meet specification:** report error to webmaster.

FR 12: Allergy Information

Description	Allergies should be split into the following categories: • Peanuts • Gluten • Lactose • Soy There should be an additional category group called "other" where users can specify any allergies not listed in the default categories.
Priority	High
Dependencies	FR 2
Expected results	Allergy information meets specified requirements.
Exception Handling	**Allergy information doesn't meet specification:** report error to webmaster.

FR 13: User Score

Description	All Users will have a personal score, starting from 0, which will increase as the User gains 'points'. Points will be earned for every trade, with 10 points being given to the provider, and 1 point being given to the receiver.
Priority	Low
Dependencies	FR 2
Expected results	User is awarded points as specified above.
Exception Handling	**User notices error in current score:** Report error to webmaster

FR 14: Score Leaderboard

Description	The User Score Leaderboard should display the scores of all users in descending order (i.e highest score first). The Leaderboard should be visible to all users. When a registered user views the Leaderboard, their position on the Leaderboard should be clearly highlighted so its easy to find. There should also be a shortcut that allows that allows the user to instantly view their position on the Leaderboard.
Priority	Low
Dependencies	FR 2, FR 13
Expected results	Leaderboard displays all user scores. All user scores displayed in descending order
Exception Handling	**Multiple users are tied with the same points:** They all get the same position number, and the next lowest ranked user is given a position equal to the number of users with a higher score, plus one. e.g. if there are 4 users tied at 3rd, the next user will 8th.

FR 15: User Ratings

Description	All Users will have a personal rating from 0 to 5, calculated as the average of all the ratings received from other users. Whenever a user either provides or collects an item, the other user involved in the trade will rate them out of 5, and this will change their average rating accordingly. The publisher of the listing will be able to rate the collector and vice versa once the item is "inactive". For the publisher this should be done from the listing history page (see FR 26). For the collector, this should be done from the order history page (see FR 31)
Priority	Low
Dependencies	FR 2, FR 26 (User Listing Page), FR 31 (Orders Page)
Expected results	User has personal rating and it is correct. User able to rate other user involved in any trade.
Exception Handling	**User unable to give rating:** Report error to webmaster **User notices irregularity in current rating:** Report possible error to webmaster

FR 16: Listing

Description	A listing is the core element of the web catalogue. All listings should contain the following attributes: • The name of seller, along with their score and rating (immutable). • A main title describing the item (max. 50 chars) • A longer description of the item (optional; max. 200 chars). • The date the listing was added (immutable) • An expiry date for the food item. • A list of Allergen and Dietary Requirements the item conforms to (at least one must be selected). • The postcode from which the item can be collected • A listing status (i.e. Active/Inactive) • 1 primary photo (Will be displayed with listing in public catalogue [FR 22]) • Up to 3 secondary photos (Will displayed within listing page [FR 20])
Priority	High
Dependencies	FR 2, FR 22 (Item Catalogue), FR 17 (Listing Status), FR 18(Active Listing), FR 19 (Inactive Listing), FR 20
Expected results	All listing items conform to the structure mentioned in the description
Exception Handling	

FR 17: Listing Status

Description	At any particular moment in time, all listings should have one (and only one) of the following statuses: **Active:** The listing item is available for collection. (see FR 18 For more details) **Inactive:** The listing item is no longer available for collection (see FR 19 for more details.)
Priority	High
Dependencies	FR 2, FR 18, FR 19
Expected results	All listings are either Active or Inactive
Exception Handling	**Listings don't meet status specification:** report error to webmaster

FR 18: Active Listing

Description	All active listings are visible on the public catalogue. Any registered user (except the publisher) should be able to make an offer to commit to collecting a listed item. Any item should have a maximum of one offer. If another user has already offered to receive an item, it will remain Active, but will become unavailable, so no other user will be able to place another offer.
Priority	High
Dependencies	FR 2, FR 22, FR 29 (Commit to collect)
Expected results	All active listings meet the specifications in the description above.
Exception Handling	**Active Listing doesn't meet specification standards:** report error to webmaster.

FR 19: Inactive Listing

Description	Inactive Listings are not visible on the public catalogue. A listing item can only be made inactive by a by the publisher manually setting it to inactive. Once an listing item is made inactive, It will no longer be visible on the public catalogue. It will also remain permanently inactive (i.e. the user cannot reset it to active).
Priority	High
Dependencies	FR 2, FR 22
Expected results	All inactive listings meet the specifications in the description above.
Exception Handling	**Active Listing doesn't meet specification standards:** report error to webmaster.

FR 20: Listings Page

Description	All listings will have a specific page, which will contain all the listing information (see FR 16). When on this page users shall have the option to commit to collecting the item, or to add it to their watchlist.
Priority	High
Dependencies	FR 2, FR 16
Expected results	All listings have a listing page. All listing information present on page as described above.
Exception Handling	**Listing page doesn't meet specification standards:** report error to webmaster.

FR 21: Comments Board

Description	All listing pages should have a comments board, where any registered user can post a comment related to the listing. All comments will be visibile to all users. All comments will be sorted by time of posting (ascending order i.e. most recent comment first)
Priority	Low
Dependencies	FR 2, FR 20
Expected results	Registered users able to comment on listing page. All comments correctly sorted by time
Exception Handling	

FR 22: Item Catalogue

Description	The item catalogue is an index of all active listings. Any user should be able to explore the item catalogue. They should also be able to to apply all of the search filters and sorting options mentioned in FR 25 and FR 24
Priority	High
Dependencies	FR 2 FR 25 (Search Filters), FR 24 (Sorting Search Results)
Expected results	Item Catalogue available to all users. Users able to implement filters and sorting as described.
Exception Handling	

FR 23: Search

Description	Users will be able to search for items in the complete catalogue of items, based on the item's name. Sorting and filtering of search results will be done as described in FR 25 and FR 24. After performing a search, the user will be shown a page containing all the Active item listings matching the search. Each item displayed will show a preview of the full listing, with the item's primary photo, name and description, as well as the distance the item is from the user. Upon selecting a result, the user will be taken to the listing's specific page.
Priority	High
Dependencies	FR 2 FR 25, FR 24,
Expected results	User can perform search. All filters and sorting options function as specified. All results displayed as specified.
Exception Handling	**User provides empty search query:** return entire item catalogue. **Search query returns no results:** notify user that no results were found.

FR 24: Sorting Search Results

Description	By default, all search results will be sorted by distance from the Source (location of the listing), to the Destination (location of the user performing the search) of the listed item where: • The source location is always the collection postcode stated on the item's Listing page. • The destination is, by default, the postcode provided during registration. The user will be given option to change the destination to the current GPS location of their device. Alternatively, users can sort search results by Expiry date. This can be in either ascending order (i.e. soonest date first), or descending order (i.e. furthest date first).
Priority	High
Dependencies	FR 2, FR 22
Expected results	User can sort results by criteria mentioned above. All results are correctly sorted.
Exception Handling	**User unable to sort results:** report error to webmaster **Results not sorted as expected:** report error to webmaster

FR 25: Search Filters

Description	Users should be able to filter results by: • Dietary Requirements • Allergen Additionally the user should have the option to filter out "Active" but committed items
Priority	High
Dependencies	FR 2, FR 11, FR 12 FR 22
Expected results	User able to filter results by criteria mentioned above
Exception Handling	**User unable to filter results:** report error to webmaster.

FR 26: User's Past & Current Listings Page

Description	Any Logged-in user should have access to a personal listings page, from where they can do the following: • view their current listings (all active listings) • view their listing history (all inactive listings) • add new listings • edit current listings
Priority	High
Dependencies	FR 2, FR 16, FR 27 (Add Listing), FR 28 (Edit Listing)
Expected results	User can perform all actions listed above.
Exception Handling	**User unable to perform any of the actions:** report error to webmaster.

FR 27: Add Listing

Description	Any logged in user should be able to add a new listing to the item catalogue. For a listing to be successfully added to the catalogue, It must conform to the specification of a valid listing item (see FR 16). All newly added listings will be given an "Active" status.
Priority	High
Dependencies	FR 2, FR 16 FR 22
Expected results	User able to add listing. Listing has "Active" status
Exception Handling	**User unable to add listing:** Report error to webmaster.

FR 28: Edit Listing

Description	All registered users should be able to edit any of their active listings. Users should be able to modify the following attributes: • Listing Title • Listing Description • Expiry Date • Dietary Requirements and Allergy Information • Collection Postcode • Listing photo(s) Additionally the user should be able to set the listing to "Inactive". As described by FR 19, once set to "Inactive", the user cannot revert the status to "Active", and will be unable to perform any of the modifications listed above.
Priority	High
Dependencies	FR 2, FR 16
Expected results	User able to edit listing. Changes successfully saved after edit complete.
Exception Handling	**Edited field(s) don't meet specifiation standards:** prevent the edit, notify user of invalid fields.

FR 29: Commit to collect

Description	A user should be able commit to collecting an item that has an "Active" status (given that no other has committed to that item). Once that user has committed to collecting the item it will be added to their orders page (see FR 31), and a line of communication should be opened between the user and publisher via the messaging facility (see FR 30)
Priority	High
Dependencies	FR 2, FR 16, FR 30 (Messaging), FR 31
Expected results	User able to commit to active item. User cannot commit inactive item.
Exception Handling	**Another user has already committed to the item:** Any users who subsequently view the shouldn't should not have the option to commit to the same item.

FR 30: Messaging

Description	All registered users should have access to a messaging inbox, which will display all their existing messaging threads. All message threads will be sorted by the time of the last recieved/sent message of that particular thread. Each thread should be unique to a listing item and will be initiated whenever a user commits to collect an item.
Priority	High
Dependencies	FR 2, FR 16, FR 29
Expected results	Messaging facility operates as specified in the description above.
Exception Handling	**Messaging facility fails to meet specification criteria:** report error to webmaster.

FR 31: Orders Page

Description	All registered users should have access to an orders page. This will contain: **Current orders:** Listings which the user has committed to buying and currently have an "Active" status. A listing should be added to the user's current orders once he commits to collecting the item. **Order history:** Listings which the user once committed to buying, and now have an "Inactive" status. The user should be able to rate the publisher once the listing is made inactive.
Priority	High
Dependencies	FR 2, FR 16
Expected results	User can access orders page. Orders correctly categorised in "Current" and "History".
Exception Handling	**User has no orders:** Orders Page will be empty.

FR 32: Watch-list

Description	All users should have a watch-list, to which they can add any item from the catalogue which they are interested in but do not yet wish to commit to receive. Any item added to the watchlist should be removed if: • The user commits to the item. • The item becomes inactive The user should also have the option to manually remove an item from the watchlist.
Priority	Low
Dependencies	FR 2 FR 22
Expected results	Users can add and remove items from watch-list
Exception Handling	**Watch-list not functioning as specified:** report errot to webmaster.

6.2 NON-FUNCTIONAL REQUIREMENTS

- Items will be stored in only one location and so will be consistent.
- Animations and transitions will be simple and fluent in order to keep the interface responsive.
- All functionalities will be supported on all devices, apart from wearables.

7 DEVELOPMENT APPROACH

7.1 DEVELOPMENT STRATEGY

We plan to execute the project using the agile method of development.

We first design the entire project, which includes wireframes, layouts, color schemes, database structure, soft and hard deadlines, milestones etc. We then implement our plan on the frontend and backend simultaneously, with each task being completed as sprints and tested as it is completed. The frontend and the backend are then combined and tested as a whole, which has been elaborated later under Testing Strategy.

7.2 DEVELOPMENT STACK

This lists all kinds of tools, programming languages, frameworks, technologies etc. that will be used in the development of the project:

- **PHP** - Database and other Backend Features
- **MySQL** - Data Storage
- **HTML/CSS**- Frontend
- **Material Bootstrap** - Responsive and attractive design using Google's Material guidelines
- **Javascript/JQuery** - Dynamic Client-Side Features
- **AJAX** - Communication between components
- **Integrated Development Environment (IDE)** - PHP Storm
- **Amazon Web Services (AWS)** - Web hosting - AWS offers a suite of cloud-computing services.

7.3 TESTING STRATEGY

The individual components are tested by the developer as soon as they are developed, along with some external testing by another team member. Once all components are integrated and the project is ready, it is tested rigorously as a whole for all possible combinations of inputs, security breach attempts, display issues, compatibility, faults and loopholes.

7.4 COLLABORATION

This lists all media used by the team members for collaboration and discussion:

- **Github** - Version Control
- **Slack** - Messaging app and website for teams
- **Email** - Mailing list provided by university
- **Group Practical** - The weekly group practical gives times for short meetings.

8 SUMMARY

Our system will be reliable, attractive and above all easy to use. This will ensure that, instead of simply throwing away excess food, people will be more inclined to use our system with little extra effort and reduce food waste, while at the same time helping others.

References

[1] WRAP UP http://www.wrap.org.uk

[2] Trussell Trust https://www.trusselltrust.org

[3] eBay http://ebay.co.uk

[4] Foodsharing https://foodsharing.de/

[5] Olio https://olioex.com/