

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	3
2	Project Scope	3
3	Domain Analysis	3
4	Proposed Deliverables	3
5	Identified, Risks, Assumptions, Dependencies and Constraints	4
6	Solution Requirements	4
6.1	Functional Requirements	4
6.2	Non-Functional Requirements	5
7	Development Approach	5
7.1	Key Summary	5
7.2	Development Stack	5
8	References	5

1 INTRODUCTION

7 million tonnes of food goes to waste in the UK, more than half of which is edible (<https://www.lovefoodhatewaste.org/>). Meanwhile, many people throughout the country struggle to put together enough money for food, with food banks becoming busier every year (<https://www.trusselltrust.org/2015/11/18/uk-foodbank-use-still-at-record-levels-as-hunger-remains-major-concern-for-low-income-families/>).

Cupboard aims to be a system that allows users to painlessly find and share food that would otherwise go to waste. It should be quick and easy to 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.

2 PROJECT SCOPE

Cupboard will facilitate the sharing of food between users. It will do so in a manner that is as simple as possible, as we want to discourage people from taking the easy route of simply binning food they won't eat. There will be a simple points-based system and a ranking system in order to encourage sharing of food and 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).

3 DOMAIN ANALYSIS

There are a couple of examples of food sharing sites/apps such as:

- foodsharing.de (<https://foodsharing.de/>): A German food sharing site that allows users (just like our site) to trade food that is going out of date with other users to reduce food waste. However it is only available in Germany.
- Olio (<https://olioex.com/>): Mobile app with strong UK presence. A key distinguishing feature is "Drop Boxes". These are local stores/cafes where users can drop off food they'd like to be shared.
- SharingFood (<https://itunes.apple.com/us/app/sharing-food/id992111062?mt=8>): Mobile app based in Italy. Seems to be fairly new and so has a very small user base.

4 PROPOSED DELIVERABLES

(Gantt Chart Goes here)

5 IDENTIFIED, RISKS, ASSUMPTIONS, DEPENDENCIES AND CONSTRAINTS

- User Location: How much information about the user's location can we make publicly available without putting their security at risk?
- Assuming google maps should be able to find the addresses entered by the user. The location functionality is completely dependent on the Google Maps API.

6 SOLUTION REQUIREMENTS

6.1 FUNCTIONAL REQUIREMENTS

FR 1: User sign up

Users can sign up via email and eventually via social media. As a minimum, they will first enter a username, email and password – then be taken to a page where they can optionally set a profile picture, location, and any dietary requirements.

FR 2: User sign up

Users will be able to list an item of food. The minimum amount of information will be a title. Optionally, they may add a picture, location, description, expiry date, and set the dietary requirement flags. If the user has set their location, it will be automatically added.

FR 3: Messaging, comments, and collection of food

Messaging (private communication, to exchange exact addresses etc.) and comments (questions that others may be interested in, expressing interest) will be implemented. After food is collected, both parties will have their scores and ratings affected appropriately and the item will be moved to a separate 'history' database. (Items can also be deleted, and not moved to a users history).

FR 4: Search

Searching will be implemented in a dynamic fashion using AJAX. By default, items will be listed by location.

FR 5: Points and ratings

A simple points/ratings system will be implemented. When some food has been shared, the donor receives three points and the collector gets a single point. Each party can then rate their experience with the other out of 5. If the user wishes, they may be displayed on a public leaderboard.

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 KEY SUMMARY

We've decided to use PhpStorm as the main IDE. Ultimately we hope to host the Database (and possibly website) on AWS. We will mainly collaborate using Github and Slack (and occasionally email).

Though we'll all inevitably contribute to both sides of the site, we've decided to split the work as follows:

- **Design and Frontend:** Clare, Ucizi and Luke
- **Database and Backend:** Soumya and Abdul

7.2 DEVELOPMENT STACK

- PHP (Database and other backend features)
- HTML/CSS
- SQL
- Javascript (Client-side features)
- Bootstrap (Simplifies mobile compatibility)
- Git

8 REFERENCES
