Functional

High priority:

* Users should be able to add products to the household fridge. The fridge will be a list of all products bought by any person in the house.
* When a user takes a product from the fridge, they have the “+” option, that indicates they took one portion of the product
* A user can register in the system by providing its email
* A user is able to authenticate in the system by providing his username and password.
* For every product in the fridge, a user can see its name, price, initial number of portions, how many portions are left, and who bought it.
* A user can see credits owed, total credits they have, and a list of things they bought

Medium priority:

* A user is able to create a new house
* A user is able to join a house
* A user is able to leave a house
* Whenever a new user requests to join a house, another member of that household should approve his request.
* If a user has minus fifty or less credits, a red flag will appear notifying him he should do the groceries.
* Red flagged user cannot mark anything as eaten in the system, until he gets credits, which can only be done by buying food.
* A product should be deleted from the system/list of products when the quantity reaches 0 portions.
* In order to fix misclicks, users have a “-” option to indicate the subtraction of a portion.

Low priority:

* The user that bought the groceries for the household can edit the information about the product they acquired, regarding the name and price of a certain product, plus how many portions they bought per product.
* If the user wants to change its password he can either provide his previous password or use the “forgot my password” functionality. He will get an email with a link to reset it
* If a user marks a product as spoiled or its expiry date has passed, the remaining portions of those products will then disappear from the list of available ones and its cost will be subtracted evenly from all of the members of the household.
* A user can reset all transactions and items in the fridge for the whole house.
* When a group of house members are eating or cooking together, one member can add all of the used products and indicate which roommates joined him for dining. The costs will be then split evenly.

Non-functional

* The test suite achieve 80% code coverage using LOC, as the metric
* The system needs to be written in Java 11.
* The system should be developed with Spring Boot and Gradle.
* The system should be modular, scalable and follow REST API structure.

User case diagram

