Software Requirements Specification Document

OurFood, Team #30
201501066 Moin Hussain Moti
201525094 Bakhtiyar Syed
201501246 Nitin John Raj
201525214 Konigari Rachna

Brief problem statement

OUR FOOD builds an effective end-to-end supply chain starting from scientifically assessing the quality of the crops from farmer, processing the crops and then delivering it to bulk buyers.

Project Details:

1. Administrator interface to input/upload crop details, quality parameters, farmer details, buyer details, logistics etc.

2. Assess the cost of the crop based on the inputs from scientific equipment, crop parameters logistics & history data.

3. Interface for buyers to request bulk orders.

4. Interface for farmers to provide crop details.

5. Analytics: Suggest the best location and logistics based on buyers request. Analyze and build reports based on parameters viz. crop, season, order size, logistics.

System requirements

Languages : Java, XML, JQuery(JavaScript), HTML, CSS

Tools : Spring MVC Framework for developing the web application.

Hibernate to integrate the database.

Thymeleaf for UI Design.

Collaboration Tools : Apache Maven, Tomcat, Spring Tool Suite (STS), Eclipse IDE/IntelliJ IDEA, Visual Paradigm, WPS Office

It should run on all platform that supports java runtime environment.

Users profile

Farmers:

Any form of access to the ourfood.

OurFood Admin:

Admin of the OurFood Systems.

Buyers:

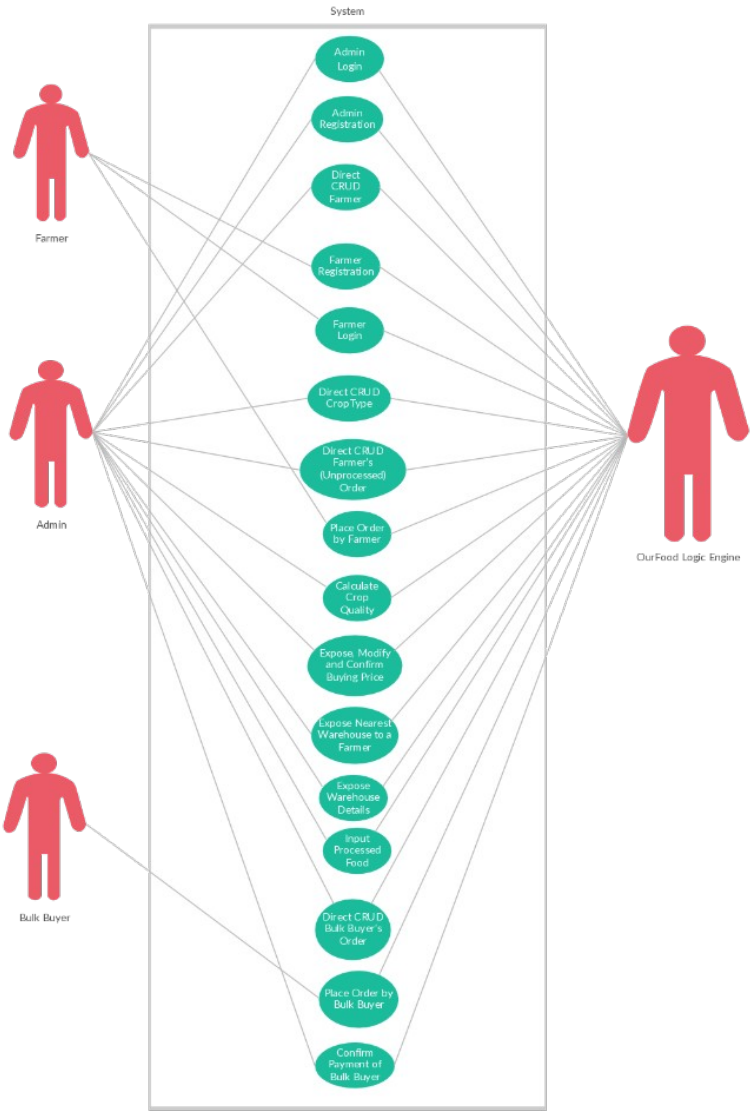
Wholesaler of processes crops

Interested in bulk buys.

Feature requirements (described using use cases)

No.	User Case Name	Description	Release
UC-1	Admin Login	As an admin, I can log into the system to perform certain actions.	R1
UC-2	Admin Registration	As a super-admin, I can register other admins.	R2
UC-3	Direct CRUD Farmer	As an admin, after logging in, I can CRUD a Farmer to the database.	R1
UC-4	Farmer Registration	As a farmer, I can enter details in a predefined UI to contact the system and thereby register.	R2
UC-5	Farmer Login	As a farmer, I can enter details in a predefined UI to contact the system and thereby login to perform certain actions.	R2
UC-6	Direct CRUD CropType	As an admin, after logging in, I can CRUD the names and types of crops so that the farmer can sell us those.	R1
UC-7	Direct CRUD Farmer's (Unprocessed) Order	As an admin, after logging in, I can CRUD an order for any farmer who has contacted me offline.	R1
UC-8	Place Order by Farmer	As a farmer, I can place an order to OurFood via a predefined UI to request OurFood to buy the crop.	R2
UC-9	Calculate Crop Quality	As an admin, I should be able to view the quality of a farmer's crop as evaluated by the OurFood personnel, calculated from the data they upload on the server.	R1
UC-10	Expose, Modify and Confirm Buying Price	As a farmer or an admin, I can view the amount that OurFood is willing to pay a farmer for his crops, and receive confirmation when the amount has been paid. Amount can be negotiated offline and changed by OurFood admins.	R1

UC-11	Expose Nearest Warehouse to a Farmer	As an OurFood admin, I can view the nearest warehouses to a specified farmer’s location.	R1
UC-12	Expose Warehouse Details	As an OurFood admin, I can view details of a warehouse, like its maximum capacity, the raw materials already present, etc.	R2
UC-13	Input Processed Food	As an OurFood admin, I can manually input the amount of a processed good that has been manufactured in a warehouse.	R1
UC-14	Direct CRUD Bulk Buyer’s Order	As an OurFood admin, I can enter details of a bulk buyer’s order	R1
UC-15	Place Order by Bulk Buyer	As a bulk buyer, I should be able to view what products are available, their prices to be delivered to me, and I should be able to place an order accrdingly (through a predefined UI).	R2
UC-16	Confirm Payment of Bulk Buyer	As an admin, I should be able to view if a bulk buyer’s order has been paid for, and thus know when to begin delivery.	R2
UC-17	New Buying Interface for Buyers	Now Users/Buyers can buy products from a new improved interface	R2
UC-18	New Entity Product	Introduction of a new entity product which categorizes buyer orders	R2
UC-19	Auto-location for Buyers	Auto Location feature for buyers to get their location	R2



Source: http://creately.com/diagram/isyfp0kl1/OurFood

Use case description

Use Case Number:	*UC-1*
Use Case Name:	*Admin Login*
Overview:	*The admin is a person in OurFood with who can interact with the system we set into place. Different admins may have different access to the data we expose.*
Actors:	*Administrator*
Pre condition:	*Login request made to system*
Flow:	*1. System receives a login request with username and password.* *2. System checks if the data is valid.* *3. System logs the admin in if it is, else returns a 'login failed'.*
Post Condition:	*The admin is logged in.*

Use Case Number:	*UC-2*
Use Case Name:	*Admin Registration*
Overview:	*The admin is a person in OurFood with who can interact with the system we set into place. Different admins may have different access to the data we expose. The superadmin can register other admins.*
Actors:	*Super administrator*
Pre condition:	*Admin register request made to system*
Flow:	*1. System receives an admin register request with username, password, details.* *2. System checks if the data is valid.* *3. System saves the new administrator in the database.*
Post Condition:	-

Use Case Number:	*UC-3*
Use Case Name:	*Direct CRUD Farmer*
Overview:	*Farmers are the people who sell OurFood their crops. This use case allows an OurFood admin to create a Farmer account to track all transactions with the Farmer.*
Actors:	*Administrator*
Pre condition:	*Farmer register request made to system*
Flow:	*1. System receives a farmer register request with username, password and other details.* *2. System checks if the data is valid.* *3. System stores the farmer in the database.*
Post Condition:	-

Use Case Number:	*UC-4*
Use Case Name:	*Farmer Registration*
Overview:	*Farmers are the people who sell OurFood their crops. This use case allows a Farmer to create a Farmer account to track all his transactions with the OurFood, place orders, etc.*
Actors:	*Farmer*
Pre condition:	*Farmer register request made to system*
Flow:	*1. System receives a farmer register request with username, password and other details.* *2. System checks if the data is valid.* *3. System stores the farmer in the database.*
Post Condition:	-

Use Case Number:	*UC-5*
Use Case Name:	*Farmer Login*
Overview:	*Farmers are the people who sell OurFood their crops. This use case allows a Farmer to login to perform certain actions.*
Actors:	*Farmer*
Pre condition:	*Farmer login request made to system*
Flow:	*1. System receives a farmer login request with username and password.* *2. System checks if the data is valid.* *3. System logs the farmer in.*
Post Condition:	*Farmer is logged in.*

Use Case Number:	*UC-6*
Use Case Name:	*Direct CRUD CropType*
Overview:	*Crops are of various predefined types. No farmer can sell a crop that is not predefined by an admin.* *Eg: wheat, rice, etc.*
Actors:	*Administrator*
Pre condition:	*CropType register request made to system*
Flow:	*1. System receives a crop-type register request with crop name and other details.* *2. System stores the crop-type in the database.*
Post Condition:	-

Use Case Number:	*UC-7*
Use Case Name:	*Direct CRUD Farmer's Order*
Overview:	*When a farmer is ready to sell his crops, he can contact OurFood offline. The admin then generates an order and passes it to the system.*
Actors:	*Administrator*
Pre condition:	*CropType register request made to system*
Flow:	*1. System receives a crop-type register request with crop name and other details.* *2. System stores the crop-type in the database.*
Post Condition:	-

Use Case Number:	*UC-8*
Use Case Name:	*Place Order by Farmer*
Overview:	*When a farmer is ready to sell his crops, he place an order through a predefined UI. The UI is out of the scope of of our project, so we will just be handling the data request.*
Actors:	*Administrator*
Pre condition:	*CropType register request made to system*
Flow:	*1. System receives a crop-type register request with crop name and other details.* *2. System stores the crop-type in the database.*
Post Condition:	-

Use Case Number:	*UC-9*
Use Case Name:	*Calculate Crop Quality*
Overview:	*When the OurFood personnel assess the farmer's crops, they measure certain parameter's ad put it on the OurFood server. Those parameters are used to process unprocessed orders made by farmers, and calculate attributes like crop quality.*
Actors:	*OurFood Server*
Pre condition:	*Server sends signal indicating that OurFood personnel has uploaded crop parameters.*
Flow:	*1. The parameters are taken and used to calculate the crop quality.* *2. The estimating price is calculated from these parameters.*
Post Condition:	-

Use Case Number:	*UC-10*
Use Case Name:	*Expose, Modify and Confirm Buying Price*
Overview:	*Exposes the estimated price to the admin, who presents it to the farmer. Negotiations can be made, and when the deal is closed, the buying price is confirmed.*
Actors:	*Administrator*
Pre condition:	*Price request is issued to the system*
Flow:	*1. The system receives the price view request and returns the price.* *2. The system receives the modify price request and changes the price accordingly.* *3. The system receives the confirm price request and confirms the price has been accepted.*
Post Condition:	*When confirmed, the price has been fixed*

Use Case Number:	*UC-11*
Use Case Name:	*Expose Nearest Warehouse to Farmer's Location*
Overview:	*Shows all warehouses near the farmer, and the cost of transporting the crop to the said warehouses.*
Actors:	*Administrator*
Pre condition:	*View warehouses request is issued by the administrator.*
Flow:	*1. The system receives the view warehouses request with data like the farmer;s location.* *2. The system calculates the cost it might take to transport the crops to the nearby warehouses.* *3. The system displays the data.*
Post Condition:	-

Use Case Number:	*UC-12*
Use Case Name:	*Expose Warehouse Details*
Overview:	*Exposes the details of a requested warehouse, like its maximum capacity, crops, etc.*
Actors:	*Administrator*
Pre condition:	*View warehouse details request is issued to the system*
Flow:	*1. The system receives the view warehouse details request.* *2. The system shows the warehouse details.*
Post Condition:	-

Use Case Number:	*UC-13*
Use Case Name:	*Input Processed Goods*
Overview:	*An admin can input how much of each processed good has been manufactured.*
Actors:	*Administrator*
Pre condition:	*The admin enters the data.*
Flow:	*The sytem accepts the data and makes the required database changes.*
Post Condition:	-

Use Case Number:	*UC-14*
Use Case Name:	*Direct CRUD Buyer's Order*
Overview:	*Bulk buyers place orders to buy processed goods. These can be placed offline, in which case a human admin needs to enter the order into the system.*
Actors:	*Administrator*
Pre condition:	*Admin enters the data.*
Flow:	*The data is processed by the system and the order is stored.*
Post Condition:	-

Use Case Number:	*UC-15*
Use Case Name:	*Place Order by Bulk Buyer*
Overview:	*Bulk buyers place orders to buy processed goods. These can be placed through a predefined UI, which is not in the scope of this project to design.*
Actors:	*Bulk Buyer*
Pre condition:	*The bulk buyer places an order online.*
Flow:	*The data is processed by the system and the order is stored.*
Post Condition:	-

Use Case Number:	*UC-15*
Use Case Name:	*Confirm Payment of Bulk Buyer*
Overview:	*After a bulk buyer has paid, delivery needs to be done. Thus, we need to confirm the payment of the bulk buyer to start this.*
Actors:	*Administrator*
Pre condition:	*The bulk buyer pays*
Flow:	*The payment is shown to be confirmed to the admin.*
Post Condition:	-