Final Project:

Food Donation Management System

Course:

Software Design and Architecture

Members:

Salman Ahmed Khan (19k-1043)

Sadeem Sattar (19k-1102)

Abdullah Tilal Khan (19k-1103)

Dated: 22-May-2021

Term Project Deadline Announcement & Submission Guidelines

No need to submit working project and live hosting, just submit detailed documentation (as detailed as possible) project report and generated code and complete mockups (nonfunctional GUI in Balsamiq or any other tool or in HTML). At the time of proposal, you submitted few screens of the application, but now you need to submit all possible mockups depicting 20 core domain features. However, working software will be a big plus point. Following project deliverables are already announces on 24th Feb 2021 via slate:

Final Project Deliverables:

Submit following deliverables in final project submission, **[*]** means its optional **(OPTIONAL)**:
SRS **[*]** in IEEE format (https://web.cs.dal.ca/~hawkey/3130/srs_template-ieee.doc), ECBs, Robustness Diagrams, Timing Diagrams, Domain Model, Communication Diagram, ERD **[*]**, Project Flowchart, Sequence & Activity Diagrams, Network Diagram, Your own process model strategy for software development, Project Plan, Feasibility report, systems architecture, two tier, three tier and N tier architecture, Component, Object Diagram **[*]**, class diagram, Object communication, Generalization and inheritance, all types of UML associations, Associations, dual associations, self-associations, aggregation, composition, interface classes, realizations, dependencies, parameterized Classes, Enumerations, Concurrent objects , active objects, main architecture diagram, Layered architecture models, Use Case diagrams and their descriptions, Subsystem models, Sequence models, Data collection sequence **[*]**, Collaboration & State charts, Object interface specification, IDL, Analysis Classes, Deployment diagram, Homogenization process. Student need to submit the presentation, UML Generated Source Code, SQL scripts, UML project file (Papyrus), UML exported diagrams in PNG and project report with the realization of above-mentioned UML diagrams with detailed descriptions. Web services, Microservices or CORBA based architecture is encouraged. Student also need to demonstration of the working software **[*]**
Presentations will start from **19-05-2021.** I will call students for presentation according to their roll numbers or whatever sequence I decide, and if students don't appear for presentation in his / her turn he / she will lose some important marks.
All the projects must be submitted on **19-05-2021 at 11 AM on slate**. Late submissions will have negative impact on the marks.
Following are additional instructions for project submission:

1. Final Project Submission (MS Word Report + Generated CODE+SQL)
2. Follow all guidelines provided earlier in proposal submissions (24th Feb 2021). As a reference it's referred here again: Description of your term project proposal, document its features and development tasks and time line (Gantt chart), and estimate project development cost in terms of Rupees. Document name of group members (not more than 3. Project should be based on **Management Information System** such as hospital management system, inventory management, bank management system, transport management etc. Avoid creating online book store or other shopping carts. **Project should contain 20 core domain features (Note: login / logout/ dashboard is not core domain features).** For example, online flight booking is a core domain feature of Travel and Tour planning management system.
3. Projects will not be accepted after due date.
4. If you fail to submit the project on time, you may not be able to do it later on. Do not submit project on 11th hour, if in case connection is dropped you won't be able to submit it again. Project submission on emails after passing due time is not allowed.
5. If you upload the empty or corrupted archive, you will get zero marks. Hence double check before uploading.
6. Re-submission of project is not allowed, think, verify and double check everything before uploading.
7. Plagiarism, if detected, will result in zero marks.
8. Only Group leader is required to submit the project.
9. Project must be submitted via slate. Only one submission is allowed.
10. Final MS Word Project Report should contain the following.
    A. Cover page of report: Name, roll no, date of submission, **Live hosting URL** [*] with all the user name & passwords.
    B. After cover page, a printout of "this" project assignment page is required (1st page only).
    C. A detailed description of the project.
    D. Complete feature of your application. (minimum 20 core domain features. - login / logout are not domain features.
    E. There must be a **TOC** in the beginning and **References** in IEEE citation format in the end of the report. There is no formal format or template required of the report just make sure you cover all the aspects described above and submit it on time. Don't use font size greater than 10 in the report for normal paragraph, and font size 12 for headings and use single line spacing through the document. Generated Code should be submitted along with the project documentation.
    F. Refer to this PDF for UML guidelines. (user: alphapeeler , password: Letmein007)
    G. Work breakdown structure & Project Gantt Chart.
    H. Explain all the features of project by detailed textual descriptions and diagrams / screen shots.
    I. Setup instructions: Show step by step instructions on how to deploy / set-up the project after downloading. [*]
11. Submit the project after making a single zip archive of the project files. Following directories must be present in Zip file:
    A. MSWordDoc - > containing the above-mentioned report and project hosting URL **[*]** on the front page of report.
    B. Code - > Generated UML classes from papyrus (**MANDATORY)** + Complete Project source code [*]
    C. SQL -> Database SQL scripts with data of your project. [*]
    D. PPT -> Project Presentation
12. Naming convention of the zip file: **SDAProject-GroupLeaderNameRollNo-Section.Zip**

Table of Contents

Project Plan

Food Donation Management System

Description

The project Food Donation Management System emphasis on the saving of food in the form of donations and reduce the food waste. The project focuses on three types of the users which include Donor, Admin and NGOS. The Donor will be able to register with the system in order to publish the details of the donation. The NGOS will be required to register with the system to search for the required donation and receive it. The Admin of the system will determine the authentication of the donor and receiver according to the given registration details and control the management of the system.

Project Core Features:

The Project Core features are listed below:

Donor of the System:

1. Add food item description
2. Update food item description
3. View food item description
4. Delete food item description
5. View receipt
6. Add review to the receiver (NGO)
7. View ranking
8. View donation history
9. Add Profile
10. Update Profile
11. Submit order pickup confirmation of the receiver (NGO)
12. Submit registration details

Receiver of the System:

13. Search available donation
14. View the ranking of donor
15. Book the food
16. Add review to donor
17. View receipt
18. Add profile
19. Update profile

20. Submit order pickup confirmation of the donor
21. Submit registration details

Admin of the System:

22. Verify the details of donor
23. Verify the details of receiver (NGO)
24. Generate the receipt
25. Verify the receipts (for pickup)
26. Calculate per average
27. Update Accounts
28. Delete Accounts

Project Cost Estimation:

Developer Cost = Rs. 1000 per hour

Each day: 2 hours of working

Per day calculation: 1000*2*3 = Rs. 6000

Process Model Strategy

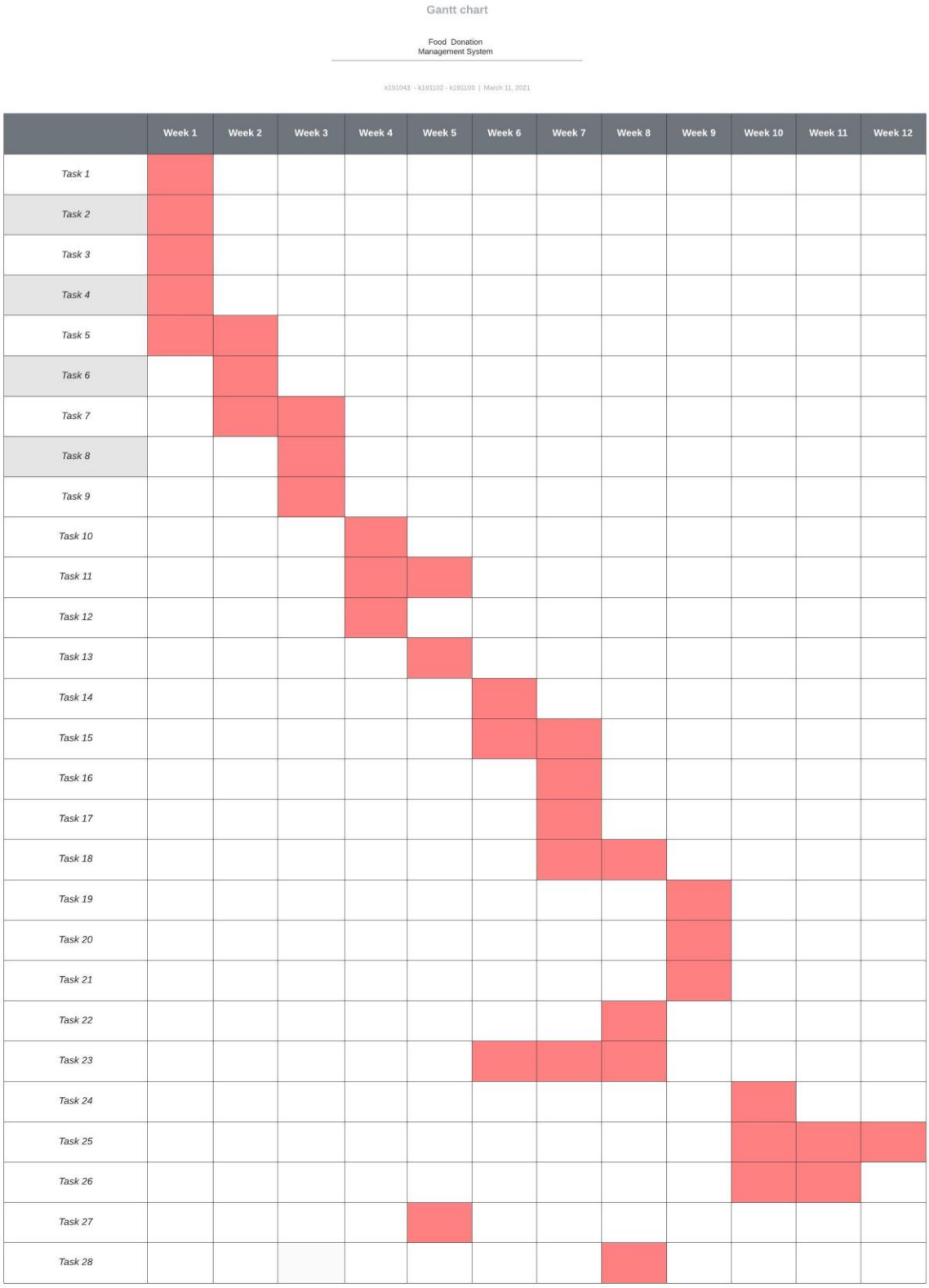
In this food donation management system we will use unified process model in our software development life cycle.

The system will be develop using Agile Development methodologies using Extreme Programming framework which is the subset of unified process model.

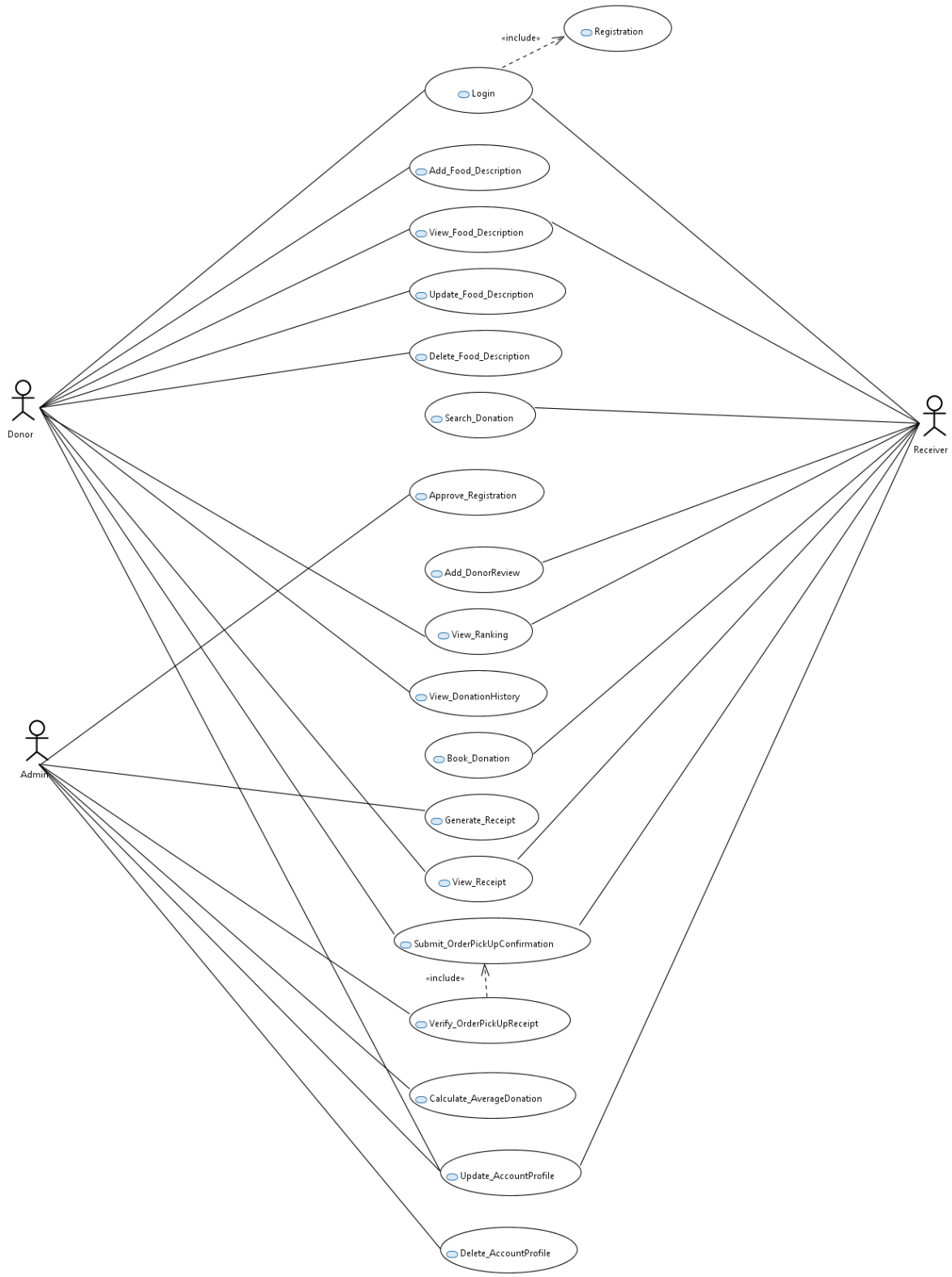
The advantages of using Extreme Programming

- Better for small projects.
- Stable development with continues testing
- Good for the projects with rapidly changing requirements.
- More reliable
- It minimizes time and save cost.

Gant Chart


Gantt chart
Food Donation
Management System
k191043 - k191102 - k191103 | March 11, 2021
Week 1
Week 2
Week 3
Week 4
Week 5
Week 6
Week 7
Week 8
Week 9
Week 10
Week 11
Week 12
Task 1
Task 2
Task 3
Task 4
Task 5
Task 6
Task 7
Task 8
Task 9
Task 10
Task 11
Task 12
Task 13
Task 14
Task 15
Task 16
Task 17
Task 18
Task 19
Task 20
Task 21
Task 22
Task 23
Task 24
Task 25
Task 26
Task 27
Task 28

Use Case Diagram



Fully Dressed Used Case

Use Case 1. Login.

- **Primary Actor:** Admin/ Receiver/ Donor.
- **Description:** This use case triggers when user want to enter into the system.
  Pre-condition: User must register first.

  Flow of Events:

  1. User visit site add name and password.
  2. Submit it after fill login form.

  Post-condition: User login successful.
  Assumption: User understand system language (English).

Use Case 2. Registration.

- **Primary Actor:** Receiver/ Donor.
- **Description:** This use case triggers when user want to register into the system.
  Pre-condition: User must have internet connection.

  Flow of Events:

  1. User visit site fill click registration button.
  2. add name and password etc.
  3. Submit it after fill login form.

  Post-condition: User registration is successful.
  Assumption: User understand system language (English).

Use Case 3. Update Profile.

- **Primary Actor:** Donor/Admin/Receiver.
  Description: The profile might change according to the user wish. Profile contain parameters like user name, address, phone number etc.
  Pre-condition: user must login into the system.
- **Flow of Events:**
  1. The userenters the system with some password.

2. The user makes the required changes.
3. The user saves the changes and logs out.

Alternative Flow of Events:

4. Some of the parameter might not need any change.
5. User might enter invalid password and need to go back.

Post-condition: A Profile will be displayed when the user enters the system.
Assumption: Admin is given the rights and privileges to enter the system and make the required changes.

Use Case 4. Add Food.

- **Primary Actor:** Donor.
- **Description:** This use case triggers when an item goes out of stock.
  Pre-condition: User must login into the system.
  Flow of Events:
  1. Donor Add food along with its description and quantity.
  2. Updates the data base accordingly.

  Post-condition: User add food successfully.
  Assumption: The Admin is given the rights and privileges to enter the system and make the required changes.

Use Case 5. Book Food.

- **Primary Actor:** Receiver.
- **Description:** This use case triggers when request for the food.
  Pre-condition: User must login into the system.
  Flow of Events:
  1. Request food form appear on screen.
  2. Receiver fill request form along with its description and quantity.
  3. Submit request.
  4. Updates the data base accordingly.
  5. Wait for the approval of form.

  Post-condition: Booking successful.
  Assumption: The Admin is given the rights and privileges to enter the system and make the required changes.

Use Case 6. View Ranking.

- **Primary Actor:** Donor.
- **Description:** This use case triggers user wanted to see its ranking.
  Pre-condition: User must login into the system.
  Flow of Events:
  1. User click view ranking button.
  2. Ranking Graph is shown to user.

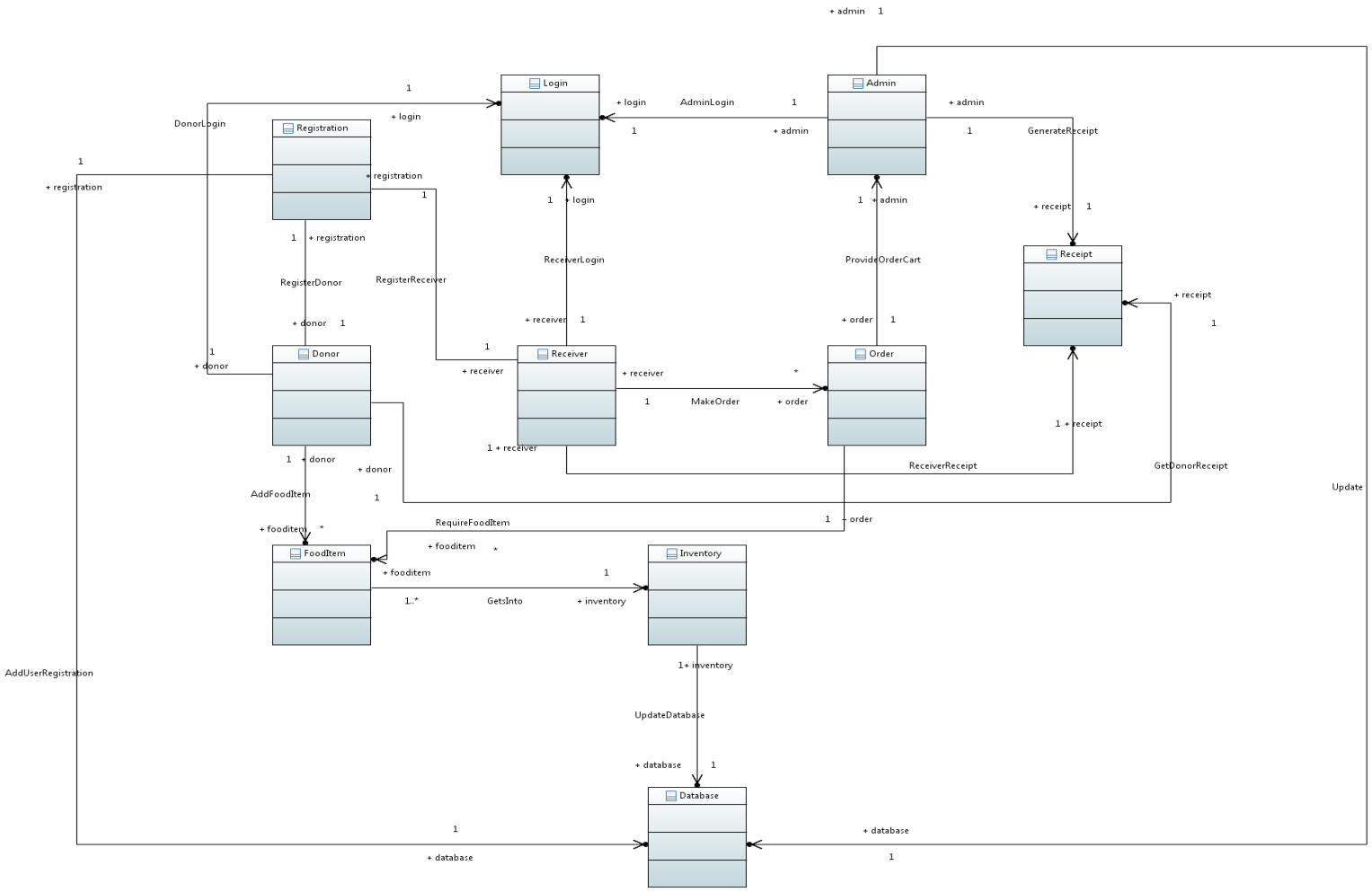
  Post-condition: User view ranking successfully.

Use Case 6. Approved Registration.

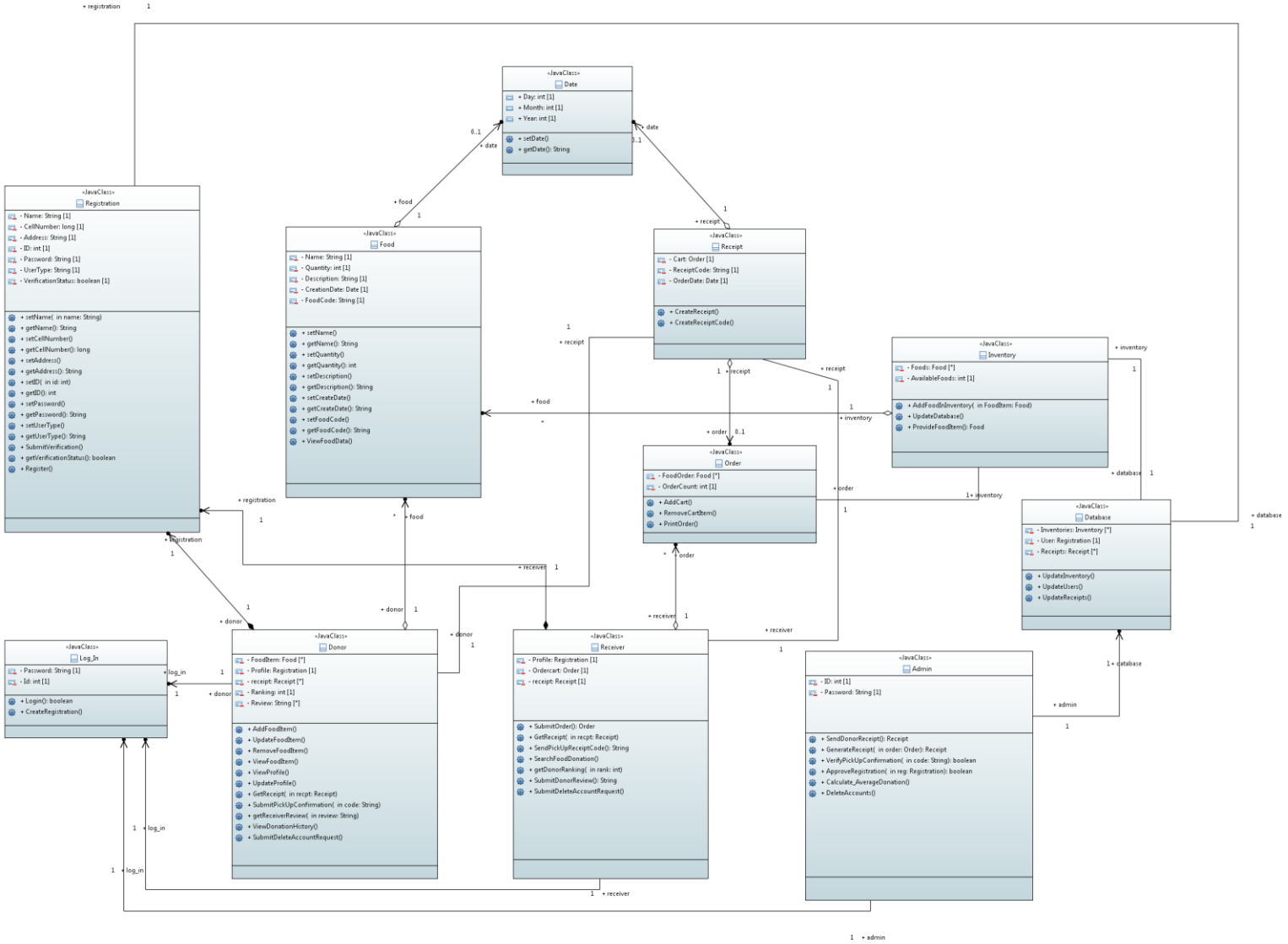
- **Primary Actor:** Admin.
- **Description:** This use case triggers when donor and receiver wanted to registration its self.
  Pre-condition: Admin must login into the system.
  Flow of Events:
  1. Admin click view registration list button.
  2. Check requirement and tick the text box accordingly.
  3. Save Data.
  4. Give Access to user.

Post-condition: User get system access successfully.

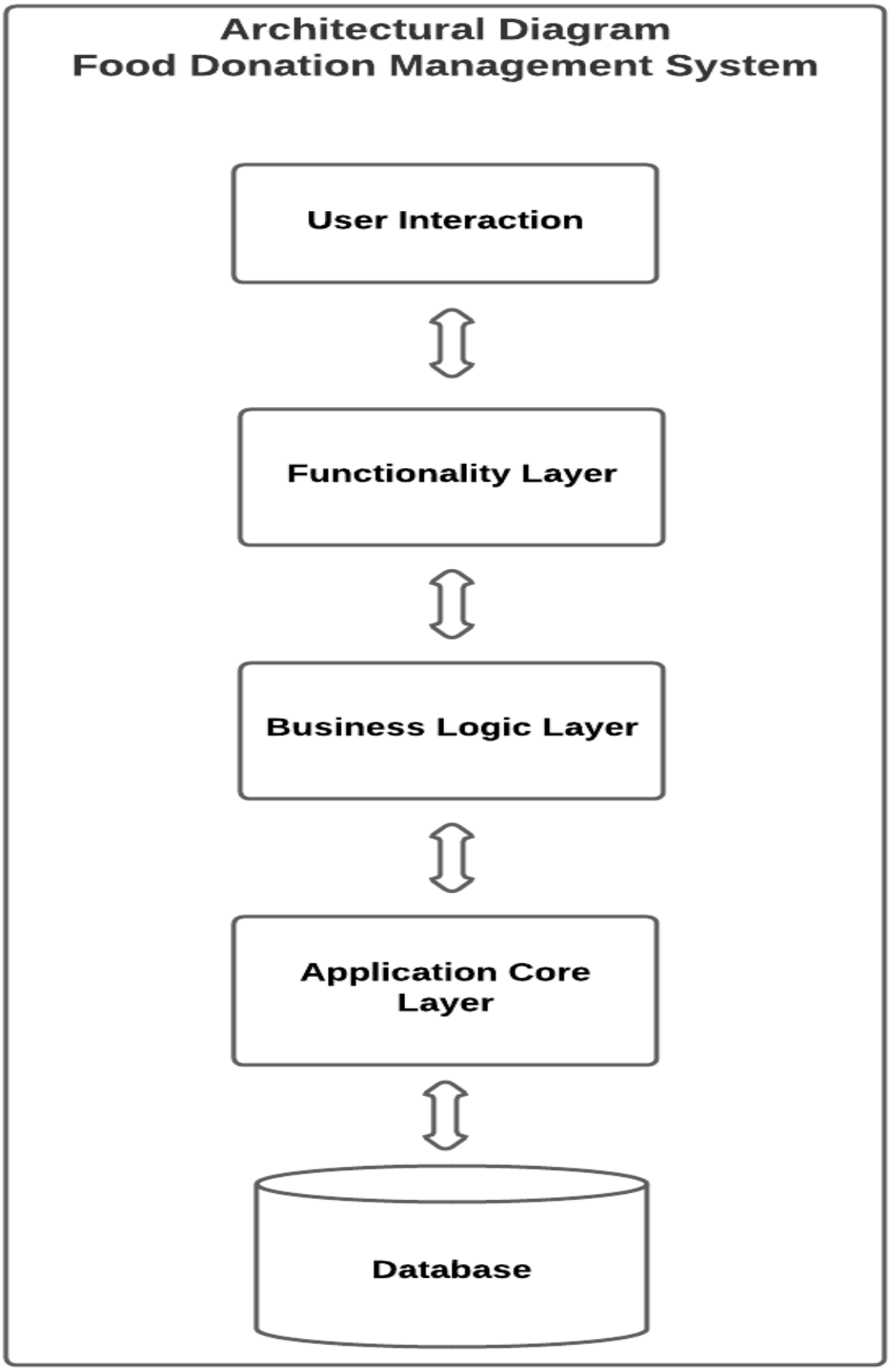
Domain Model



Class Diagram

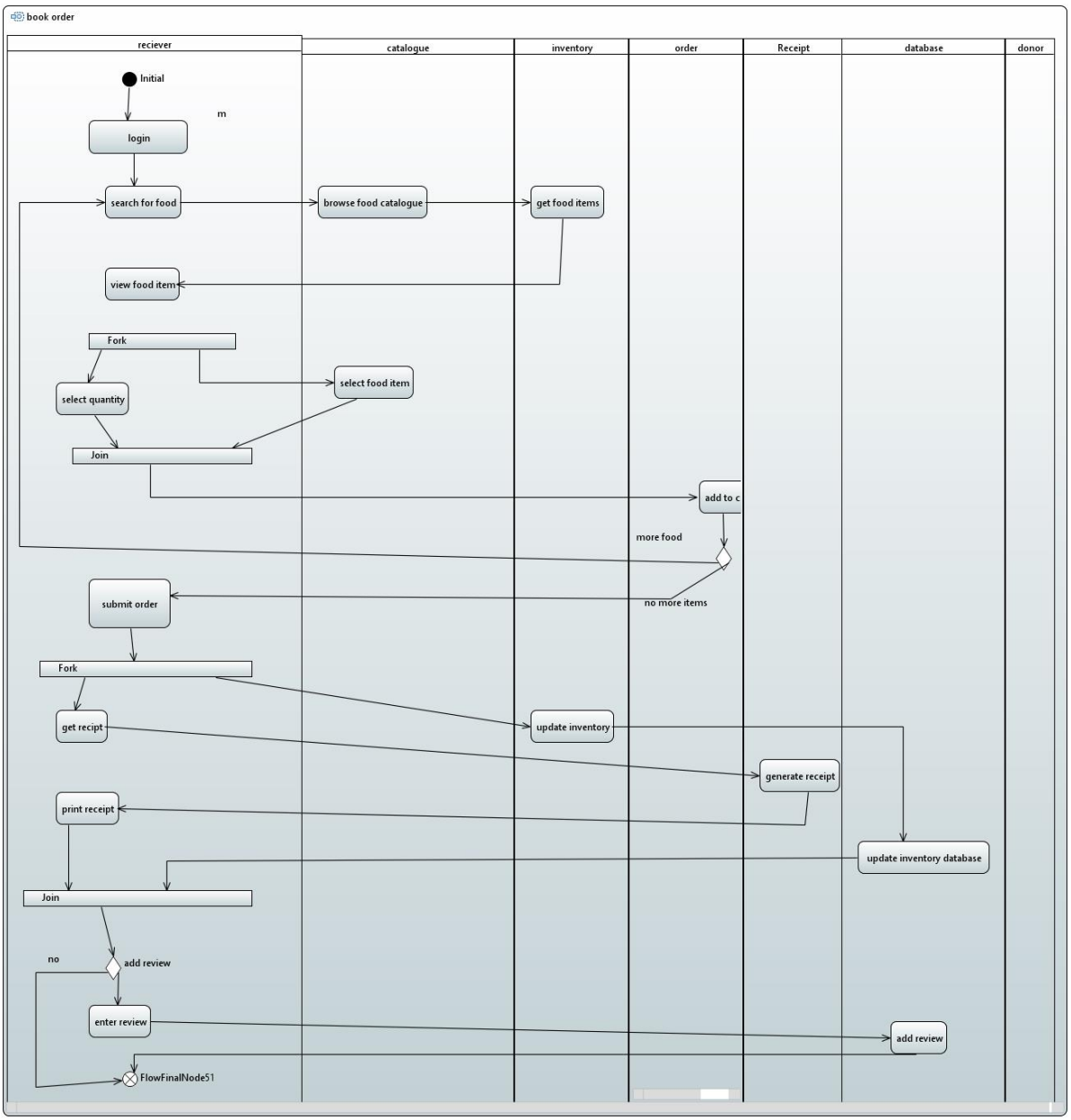


N-TIER Architecture Diagram:

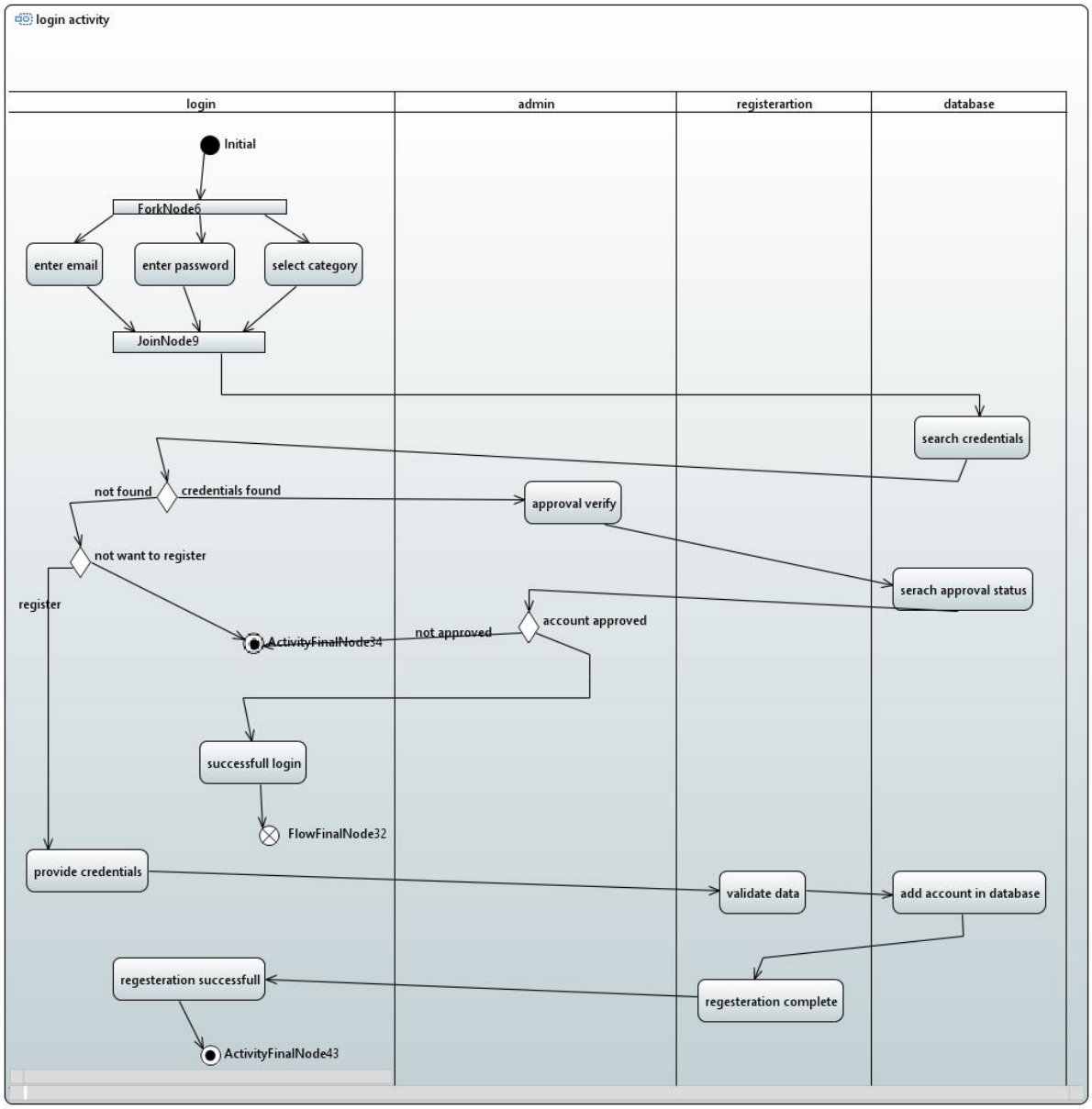


Activity Diagram

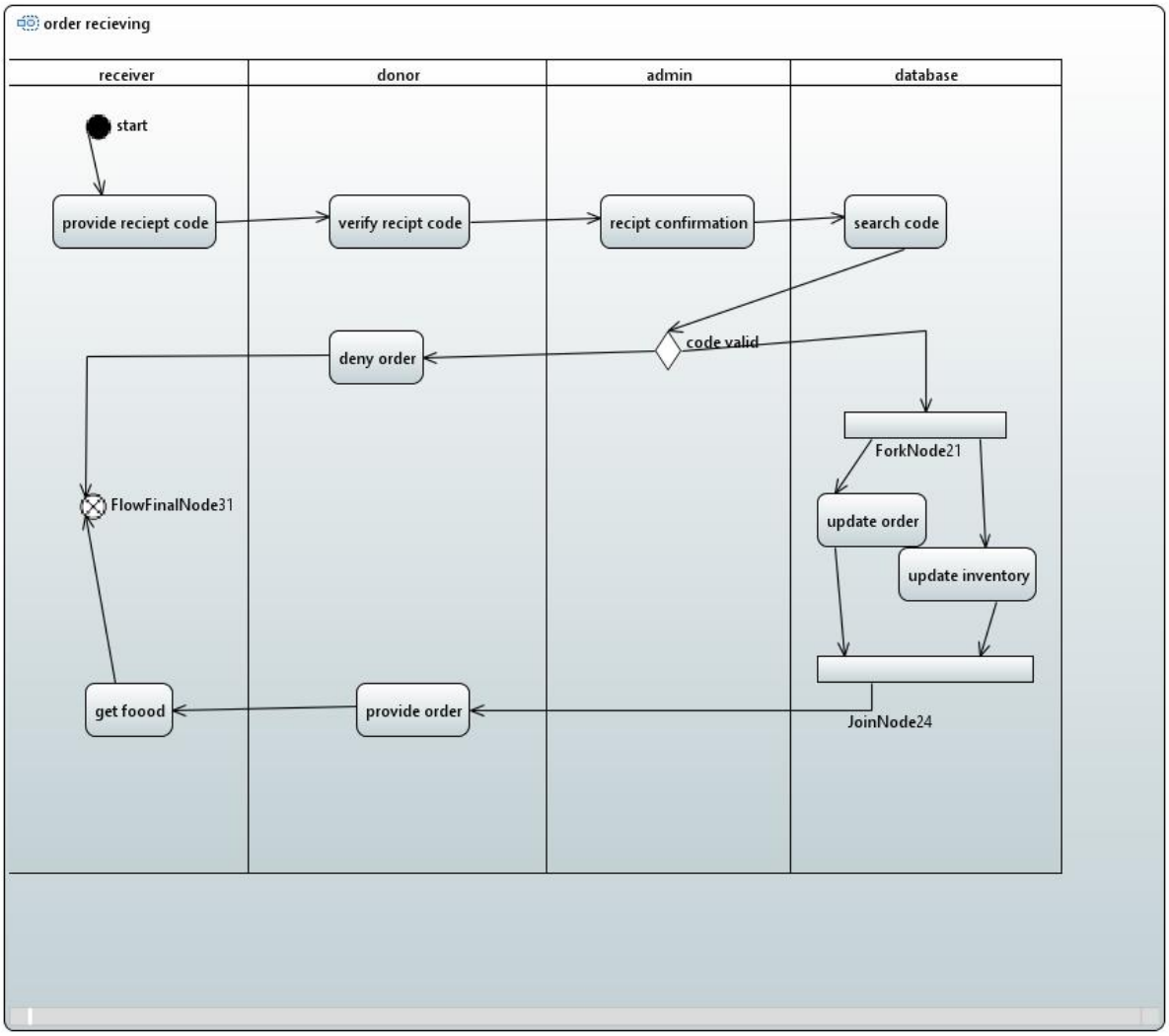
Add food to receive donation activity



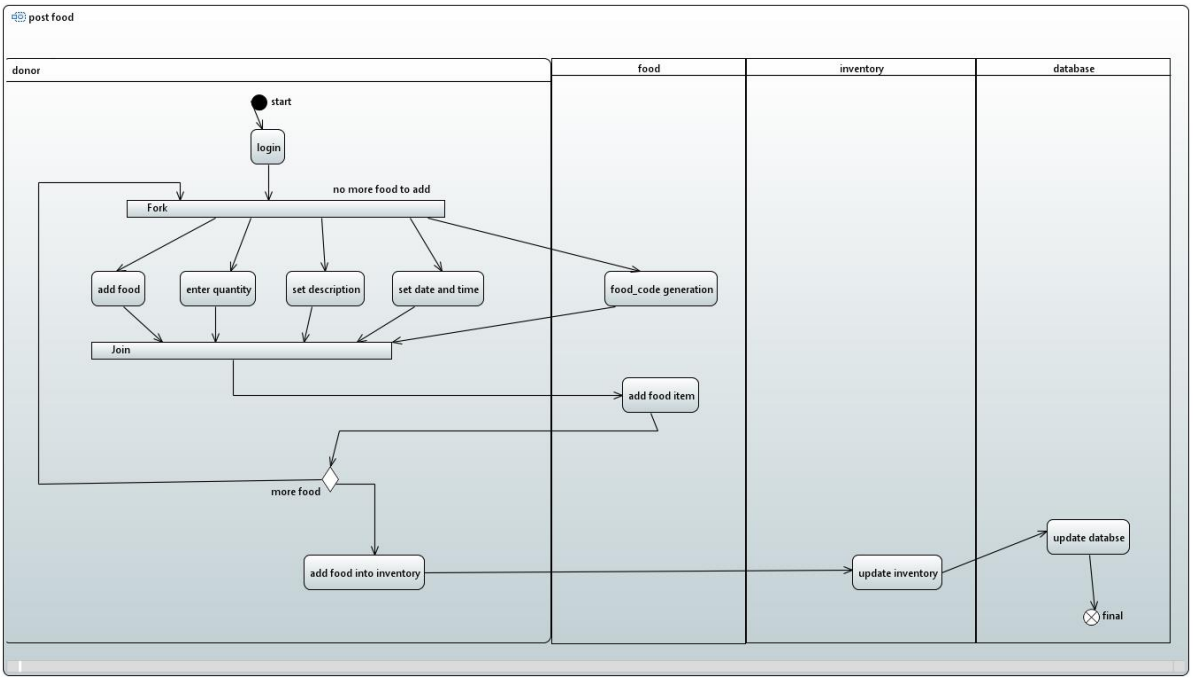
Login & Registration activity



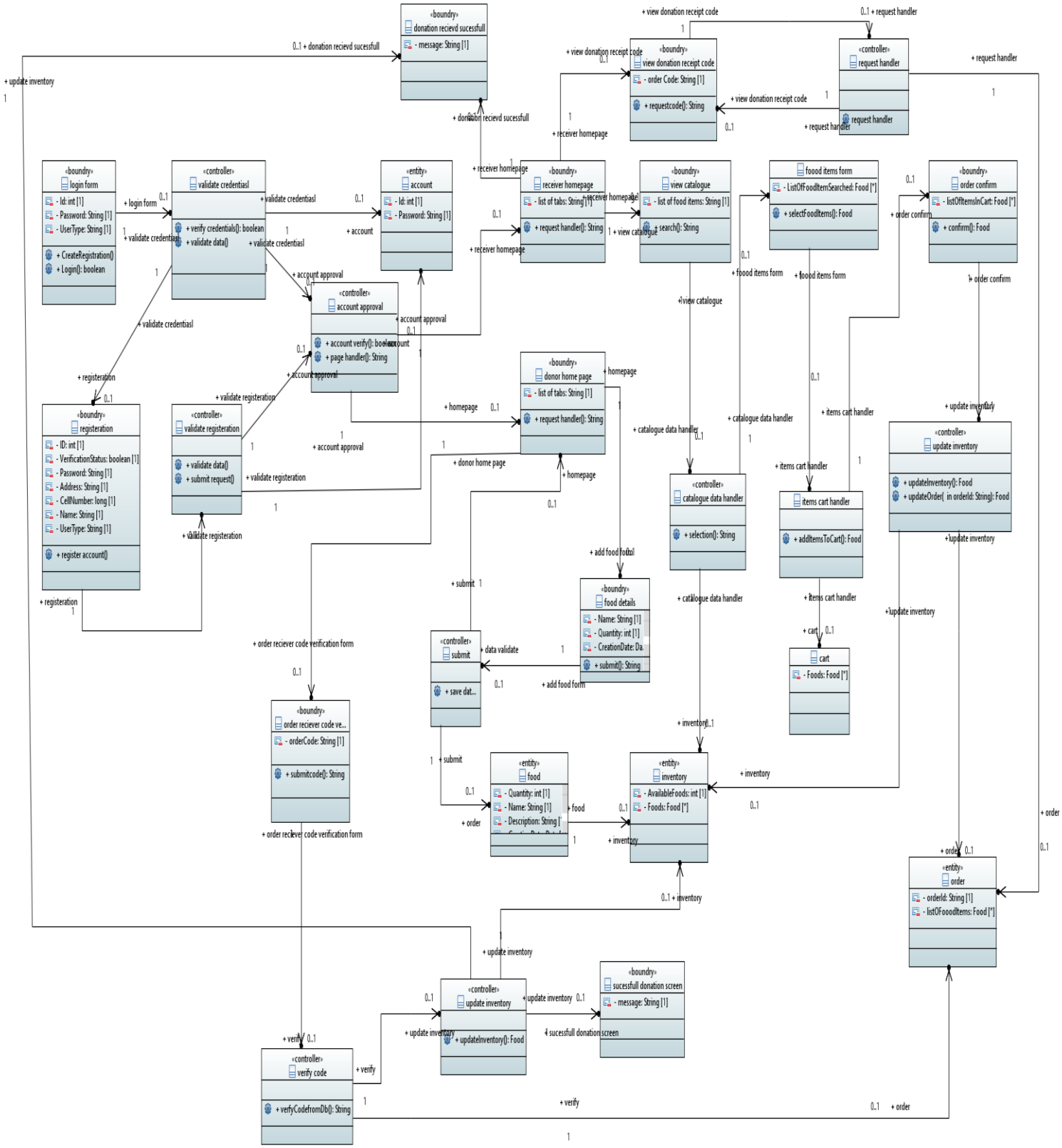
Donation pickup/receiving activity



Add excess food to donate activity

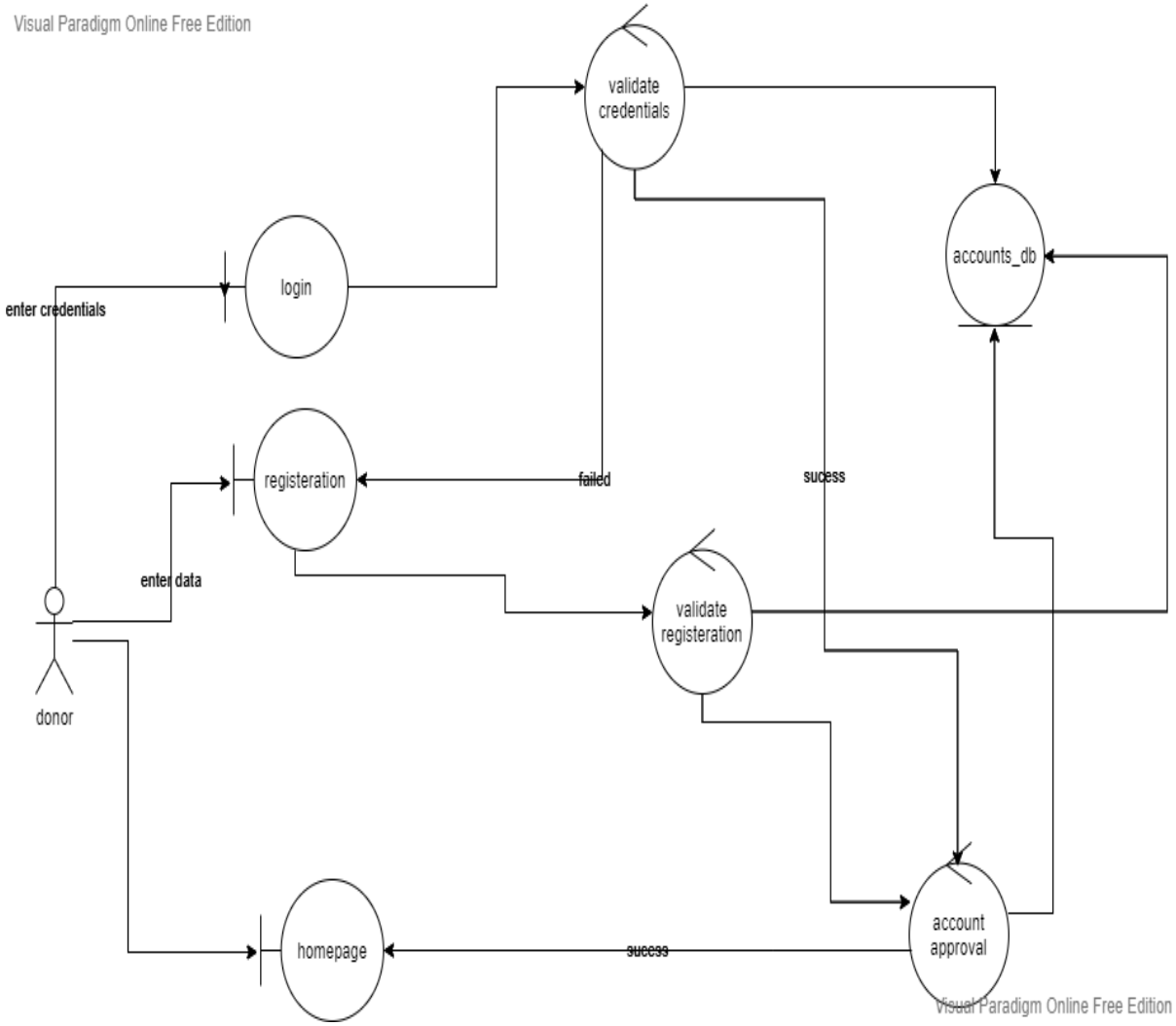


Entity Control Boundary (ECB)

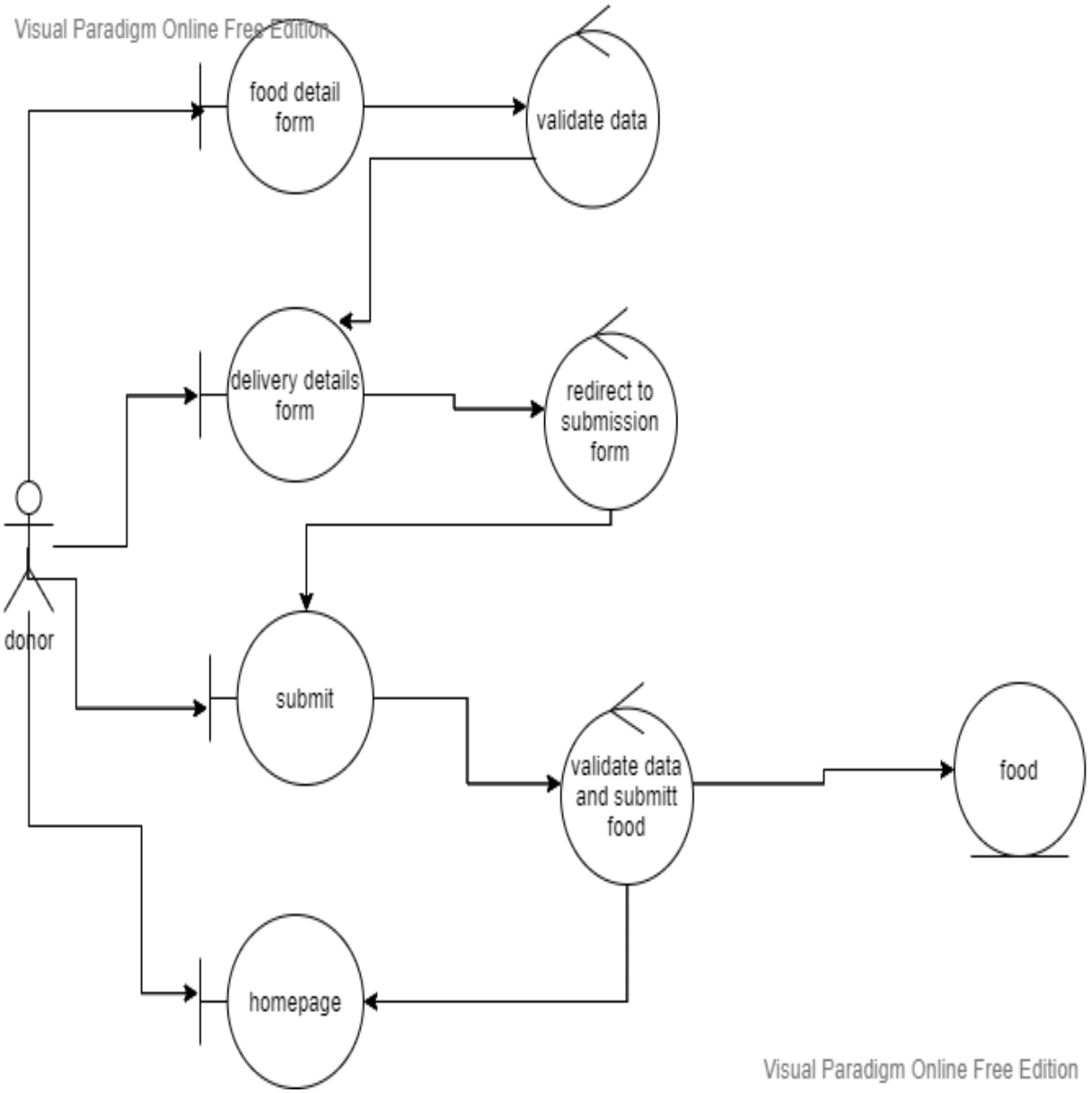


Robustness Diagram

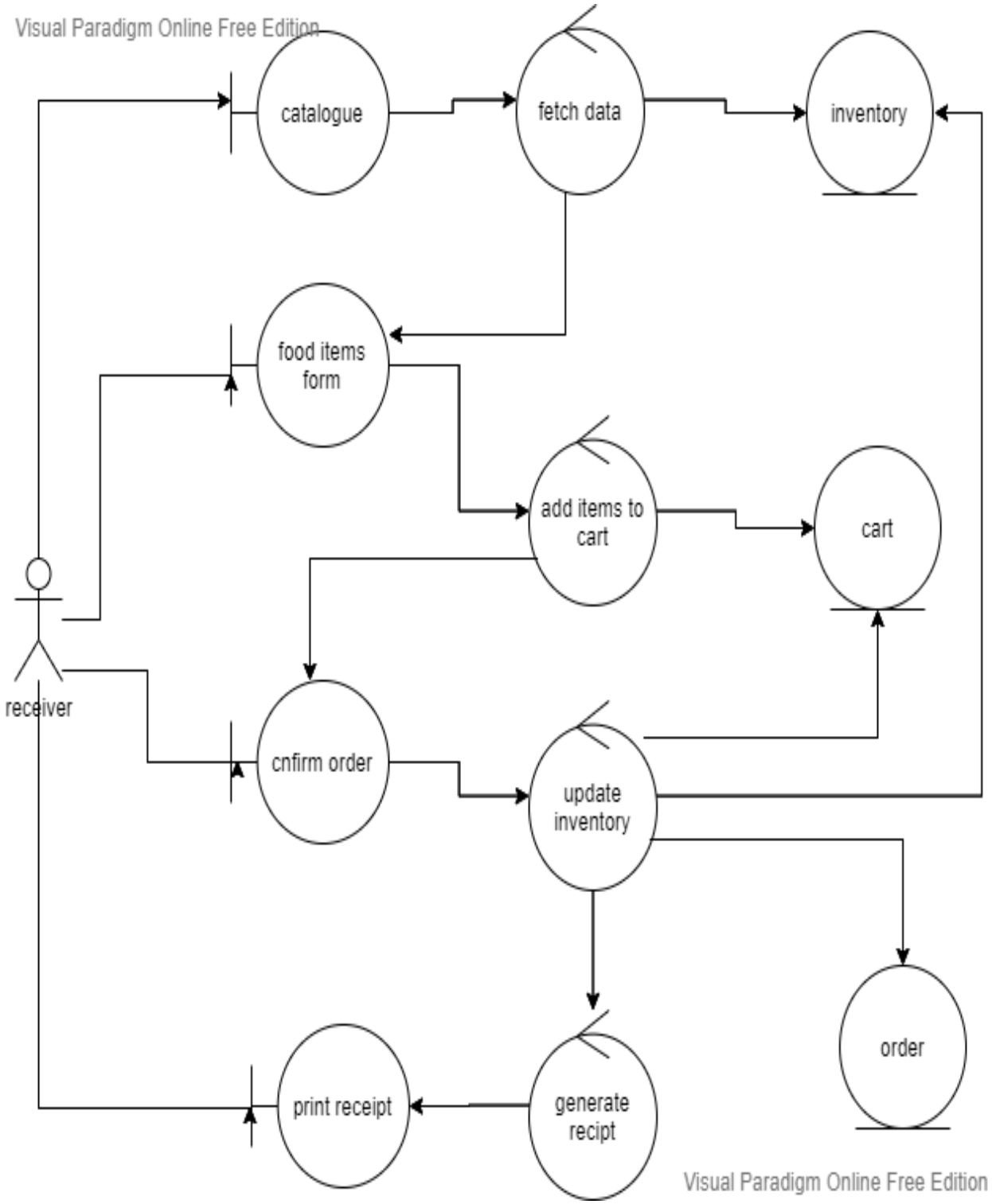
Login-Registration



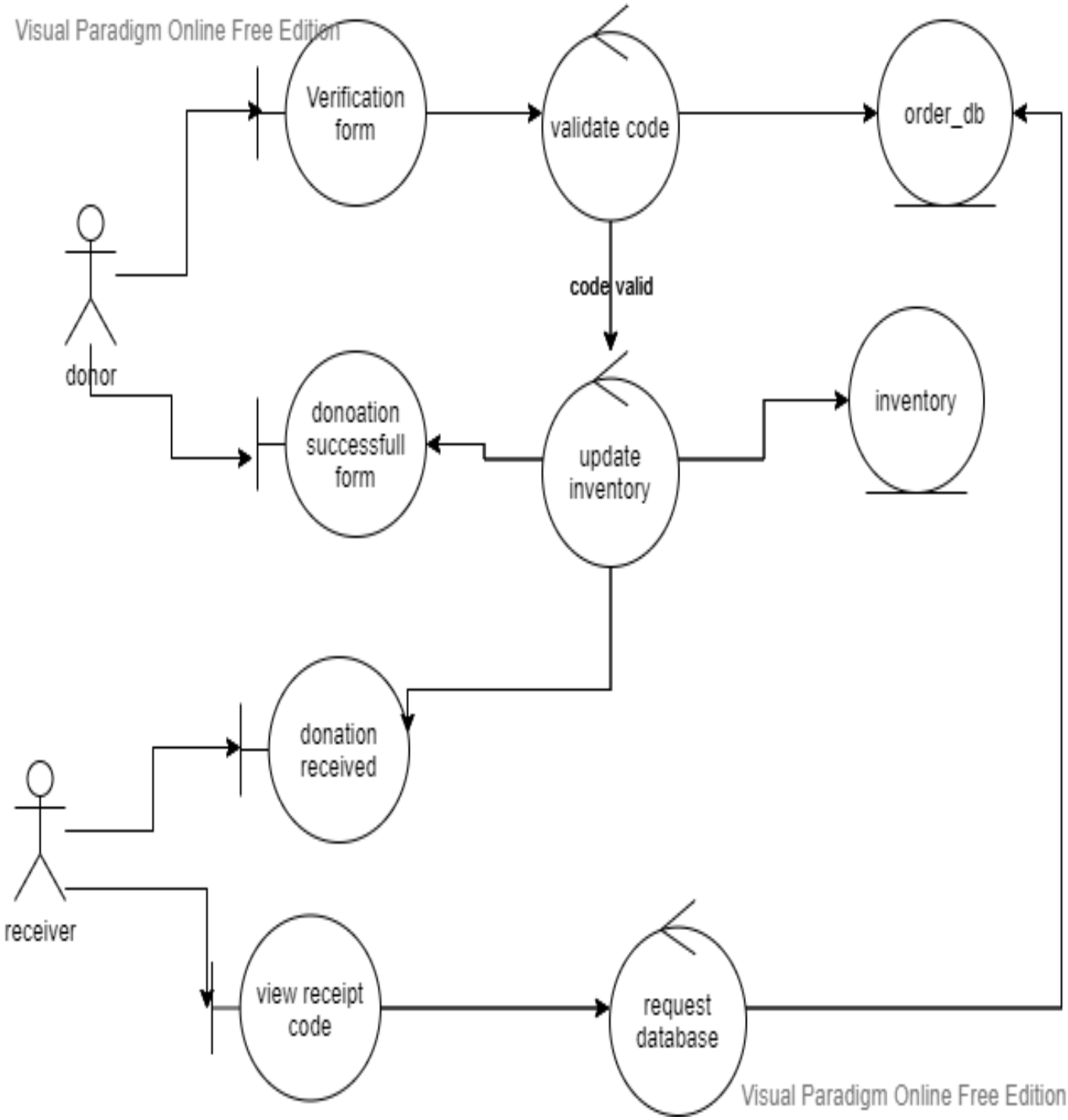
Post Food Item



Book Order

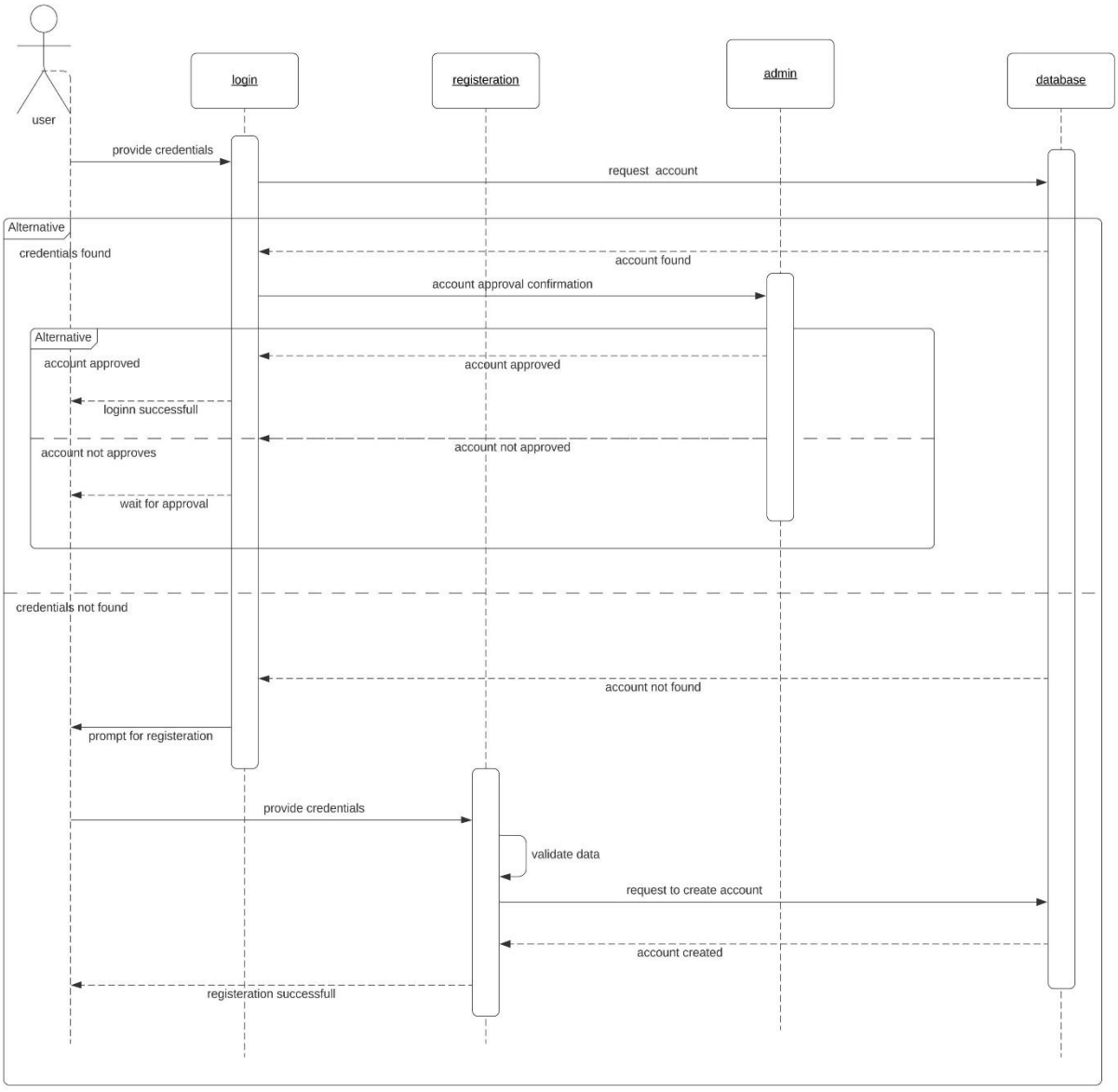

Visual Paradigm Online Free Edition
catalogue
fetch data
inventory
food items form
add items to cart
cart
receiver
cnfirm order
update inventory
order
print receipt
generate recipt
Visual Paradigm Online Free Edition

Receive Order

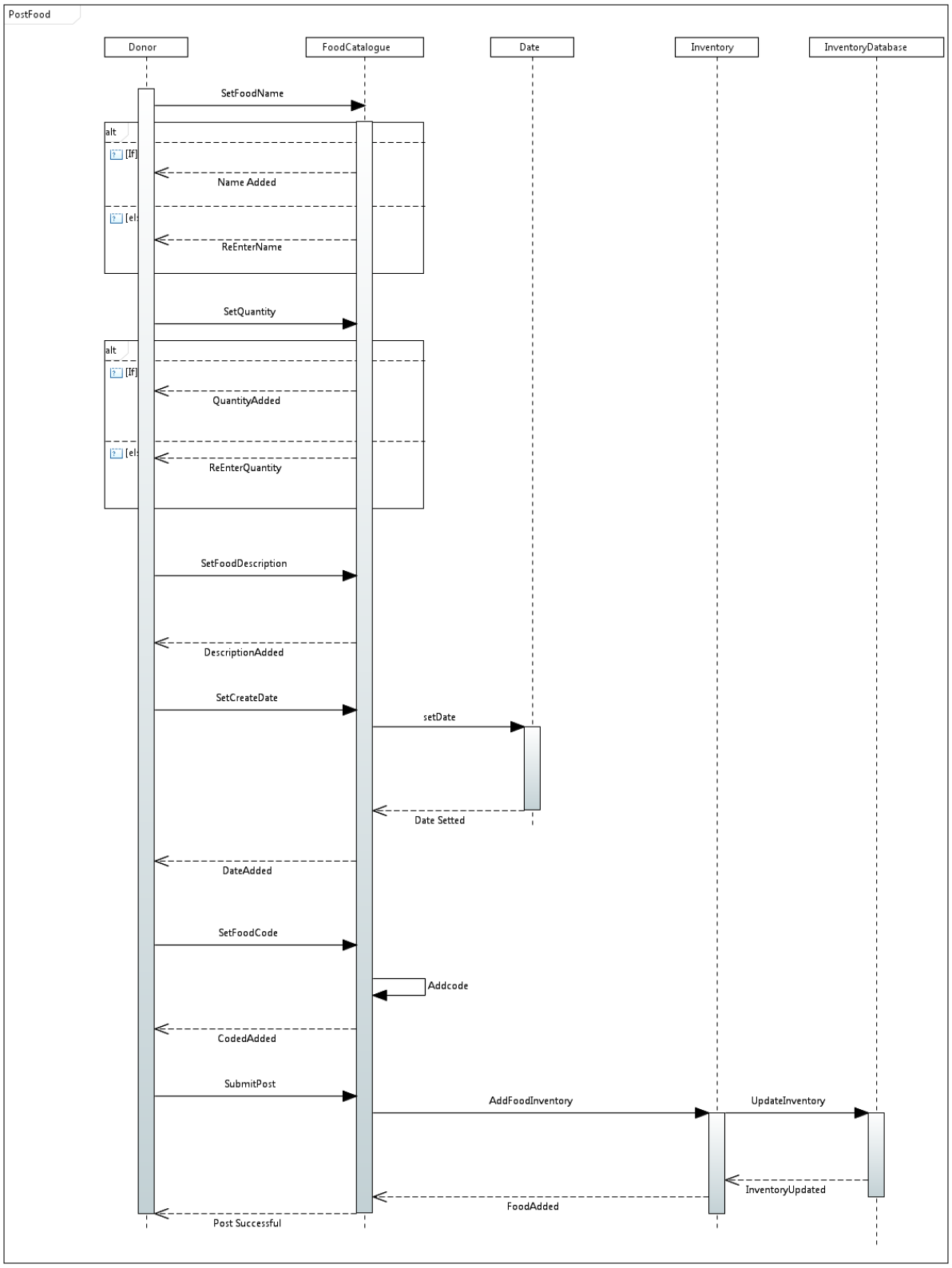


Sequence Diagram

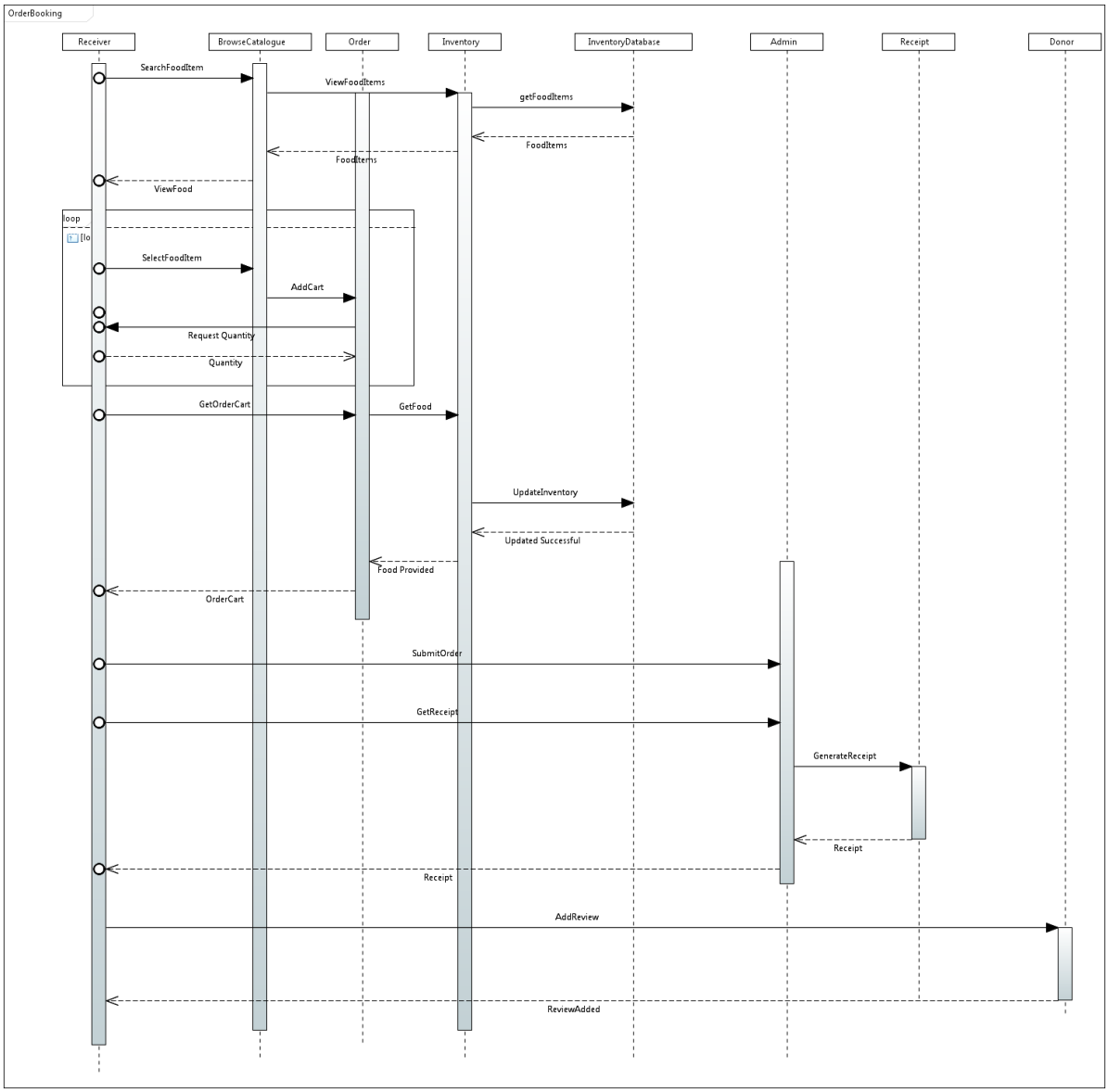
Login-Registration



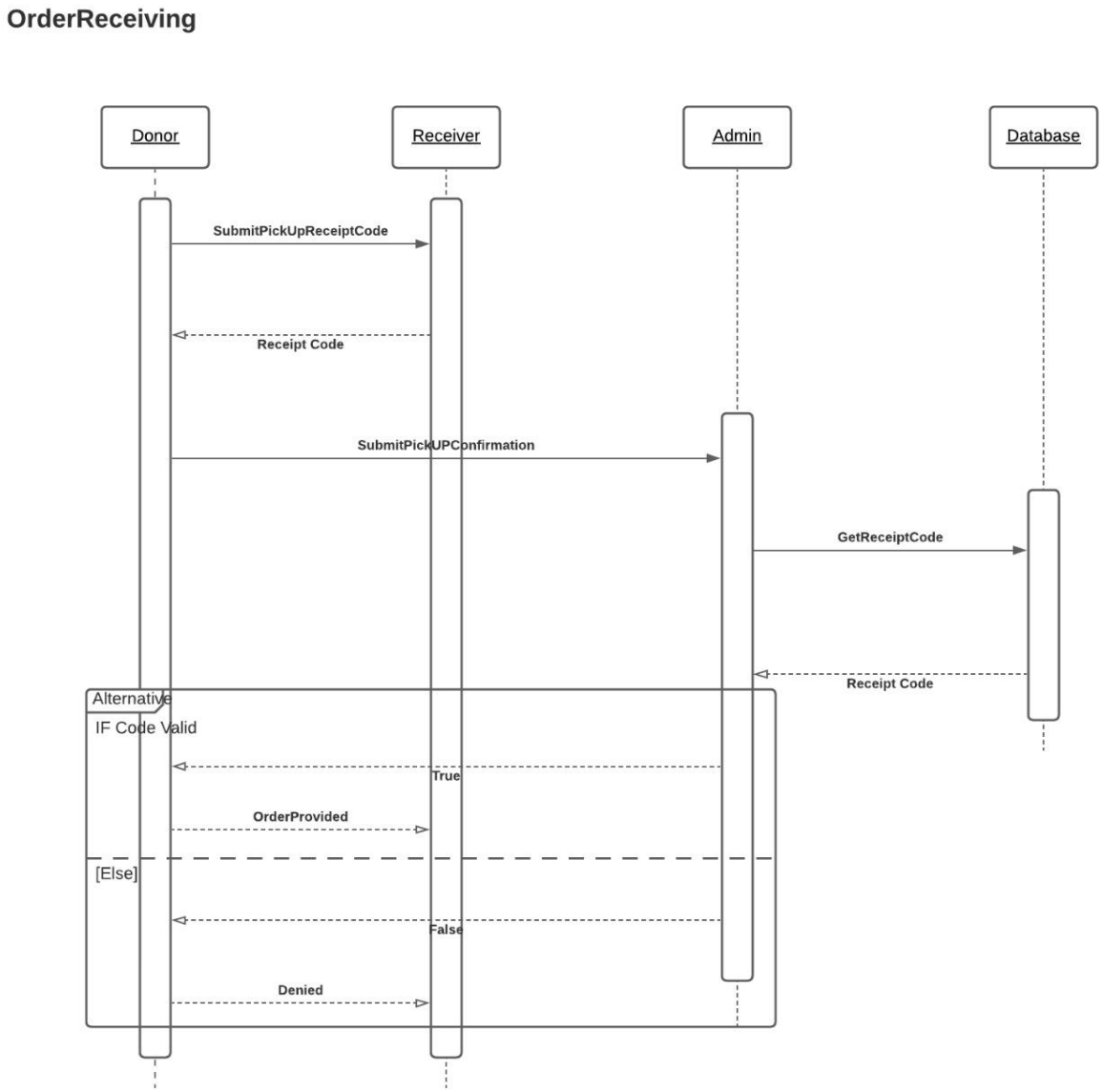
Post Food Item



Book Order

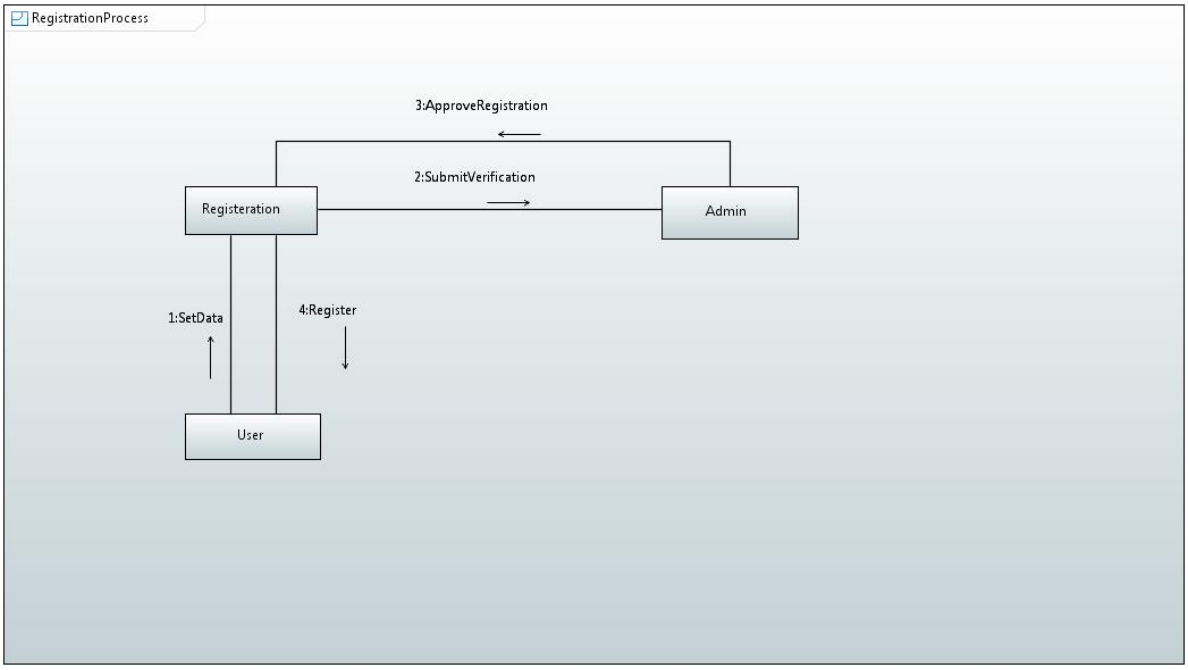

OrderBooking
Receiver
BrowseCatalogue
Order
Inventory
InventoryDatabase
Admin
Receipt
Donor
SearchFoodItem
ViewFoodItems
getFoodItems
FoodItems
FoodItems
ViewFood
loop
SelectFoodItem
AddCart
Request Quantity
Quantity
GetOrderCart
GetFood
UpdateInventory
Updated Successful
Food Provided
OrderCart
SubmitOrder
GetReceipt
GenerateReceipt
Receipt
Receipt
AddReview
ReviewAdded

Receive Order

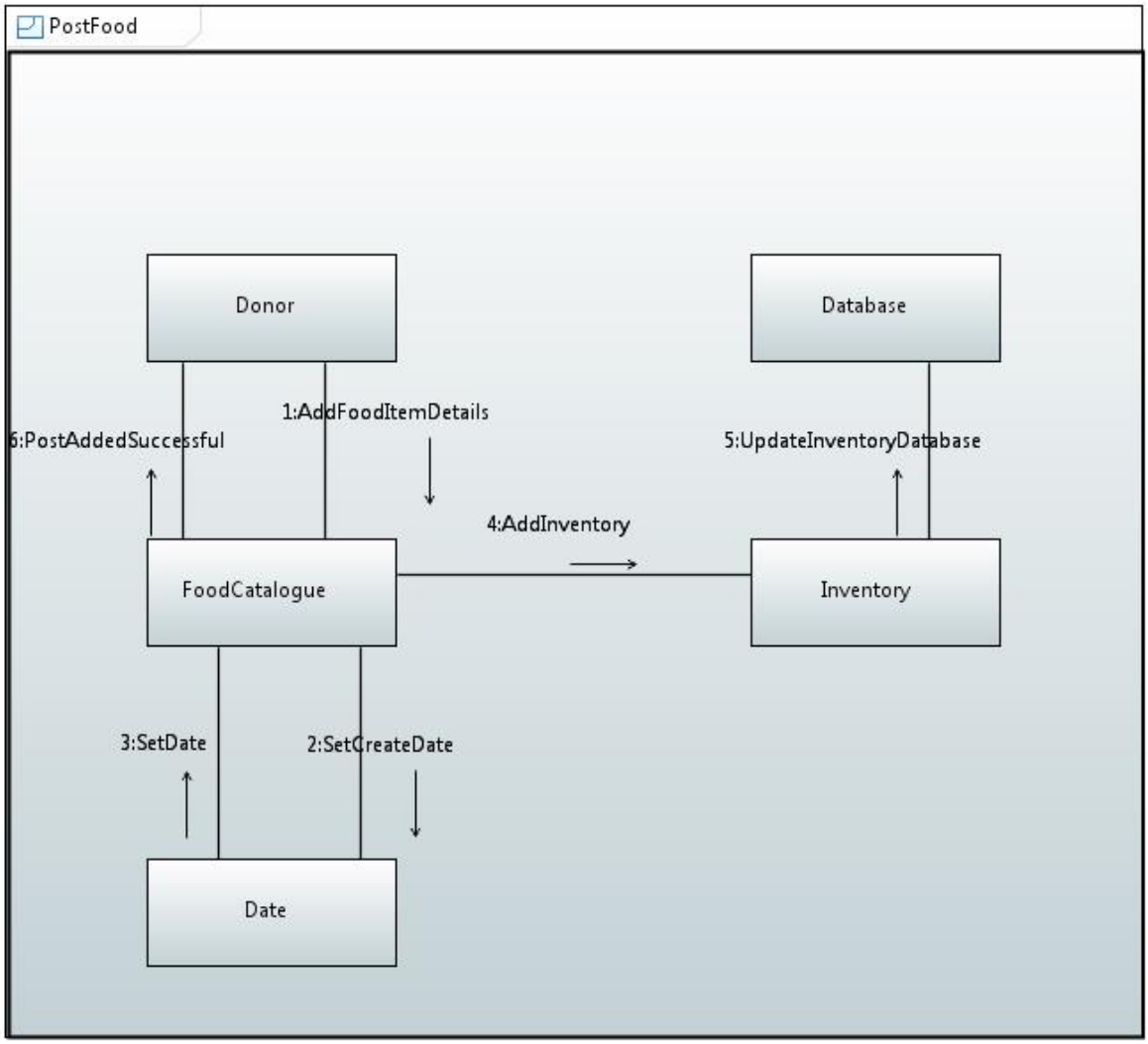


Collaboration Diagram

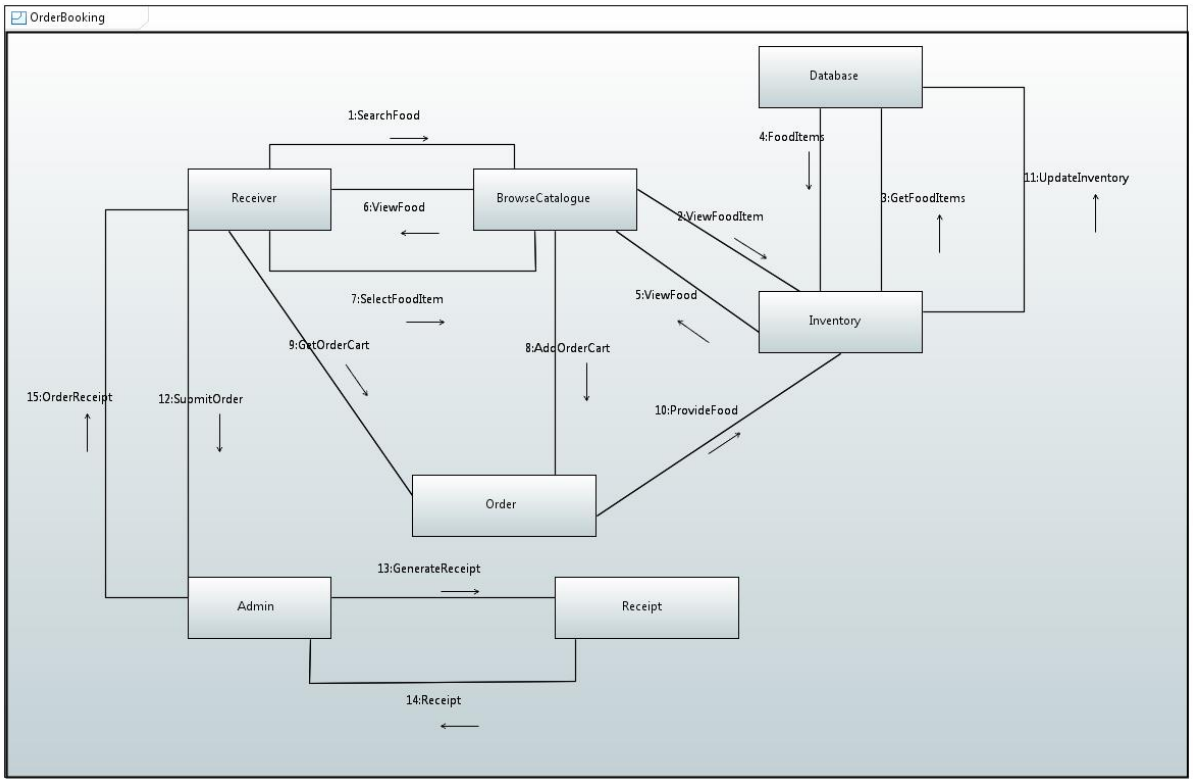
Registration



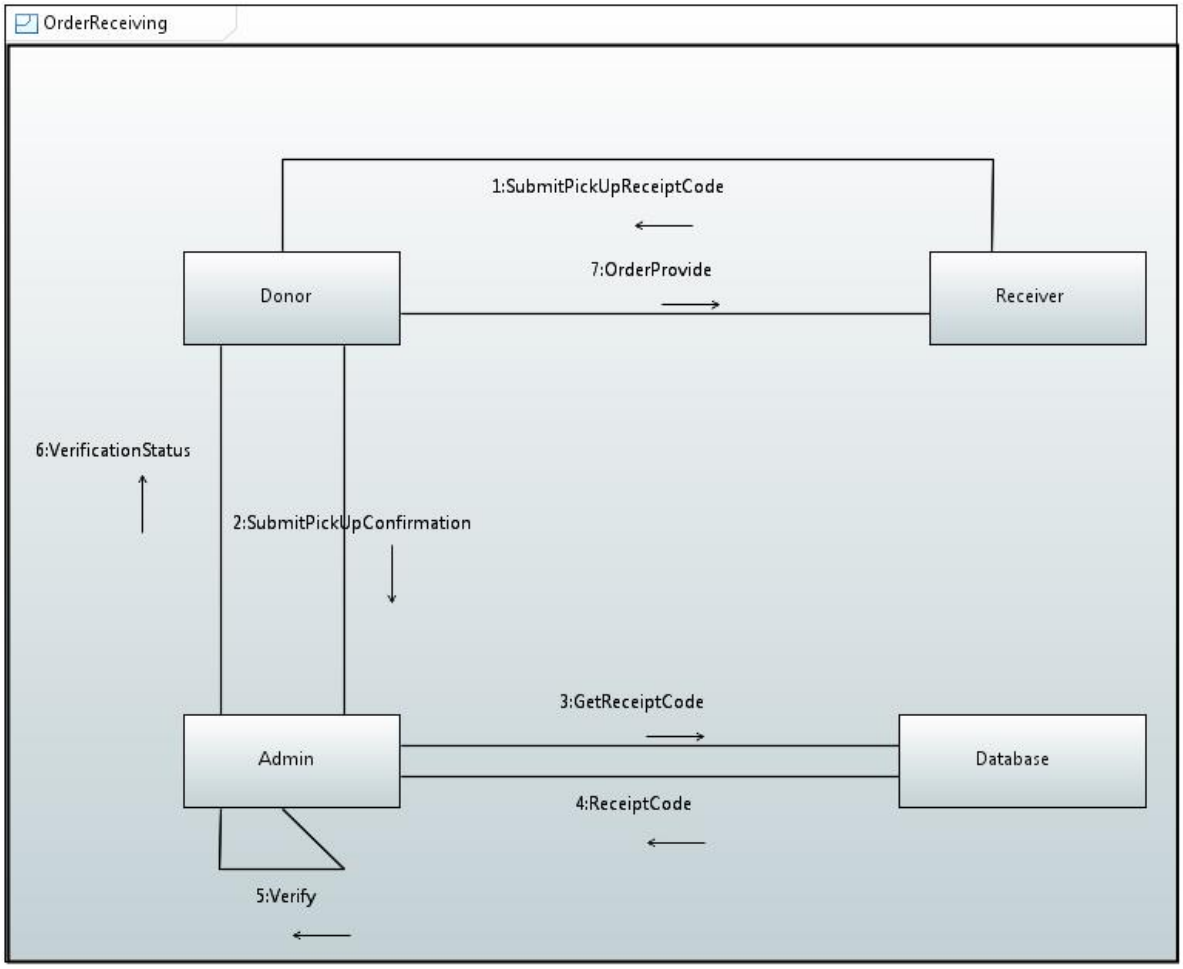
Post Food Item



Book Order

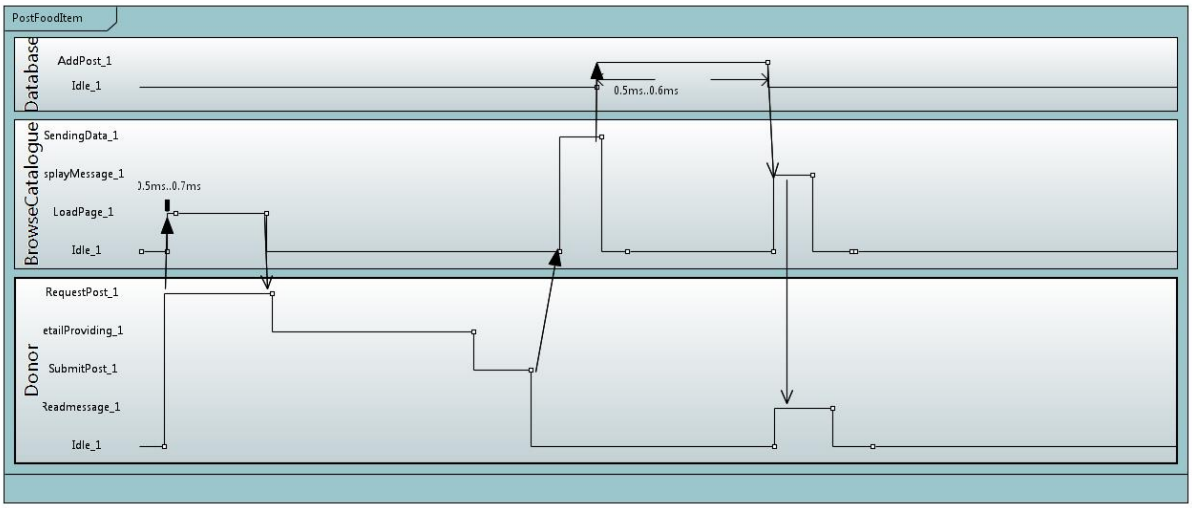

OrderBooking
Database
1:SearchFood
4:FoodItems
11:UpdateInventory
Receiver
BrowseCatalogue
6:ViewFood
2:ViewFoodItem
3:GetFoodItems
7:SelectFoodItem
5:ViewFood
Inventory
9:GetOrderCart
8:AddOrderCart
15:OrderReceipt
12:SubmitOrder
10:ProvideFood
Order
13:GenerateReceipt
Admin
Receipt
14:Receipt

Receive Order

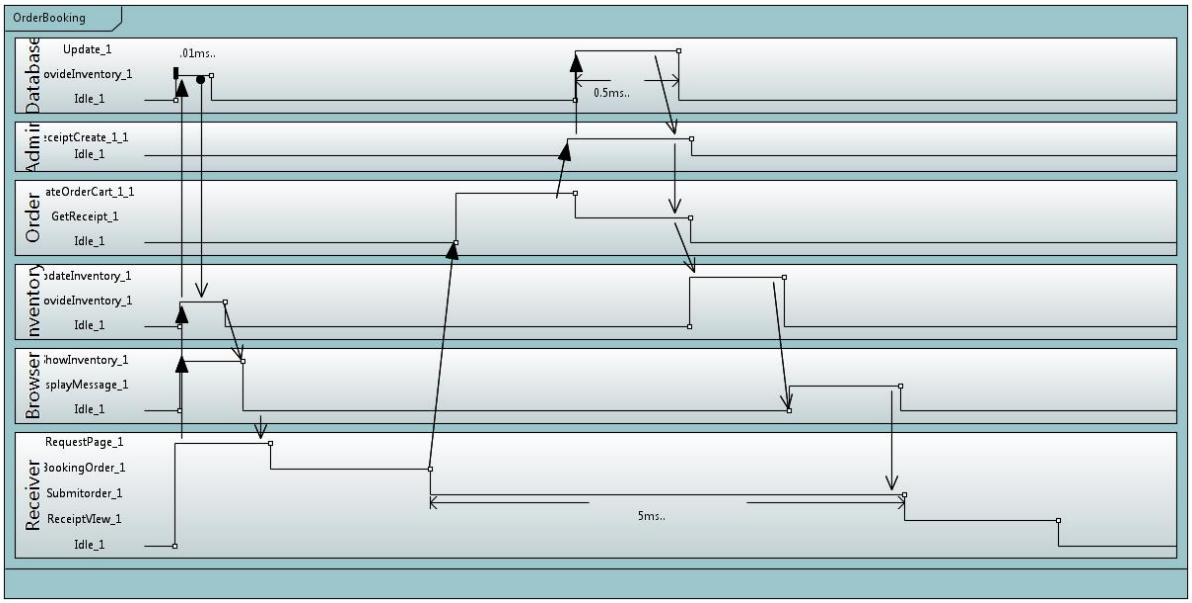


Timing Diagram

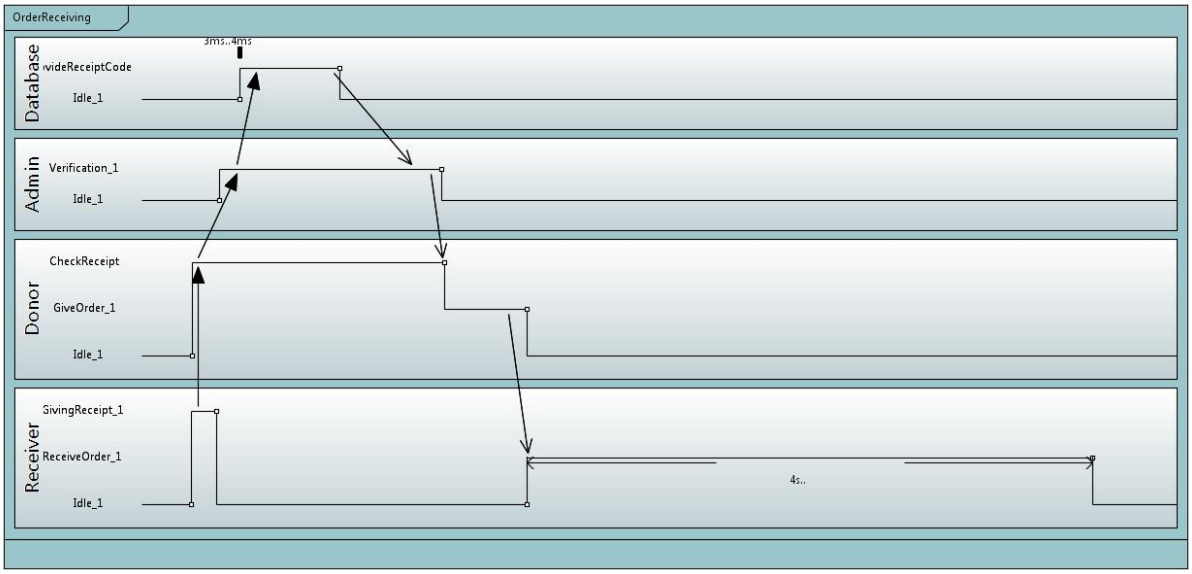
Post Food Item



Book Order

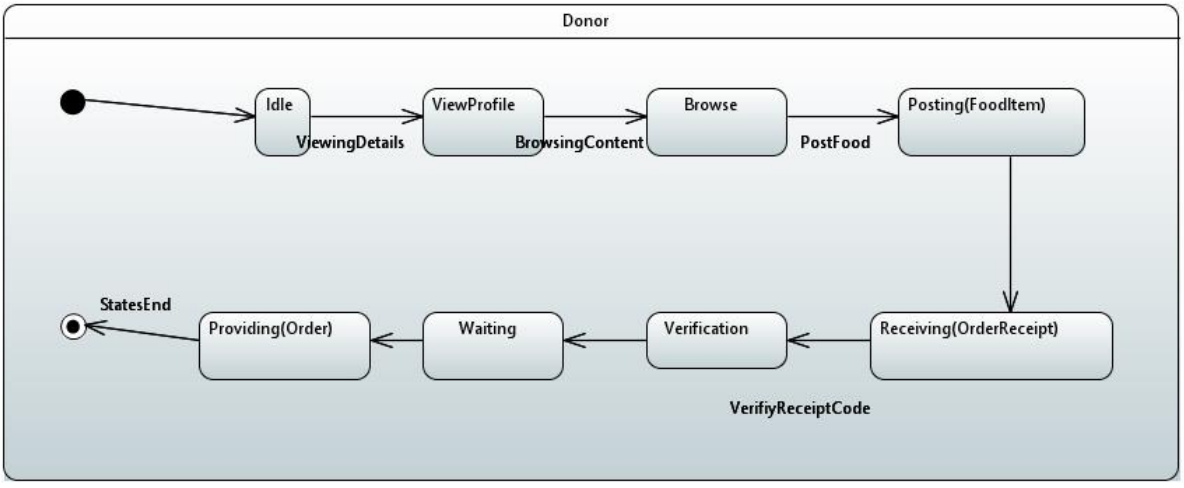


Receive Order

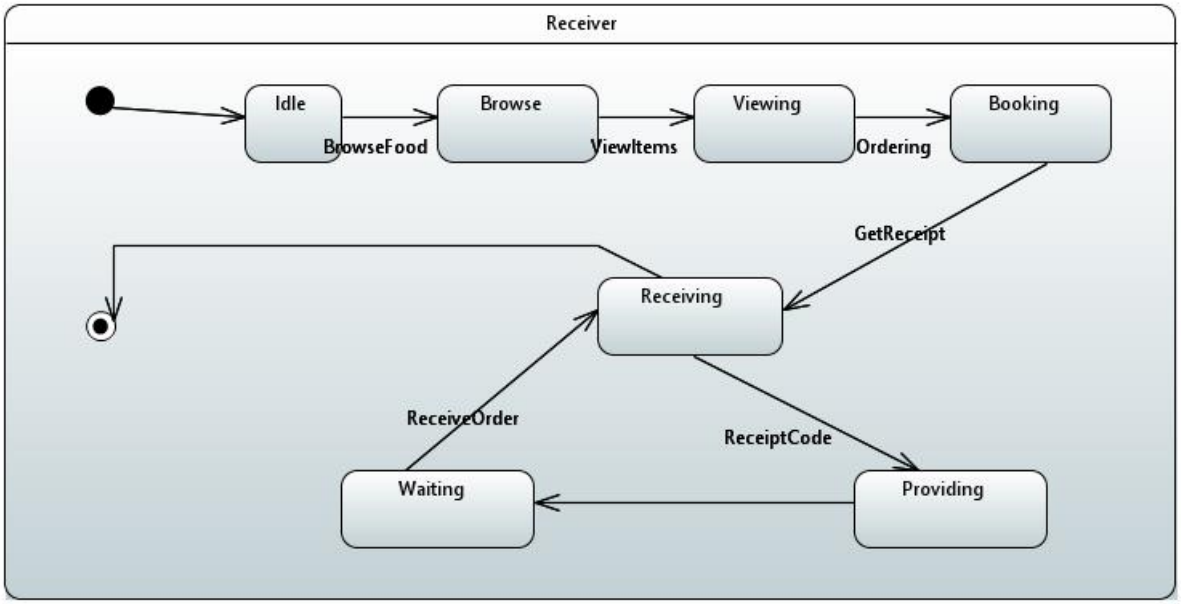


StateChart Diagram

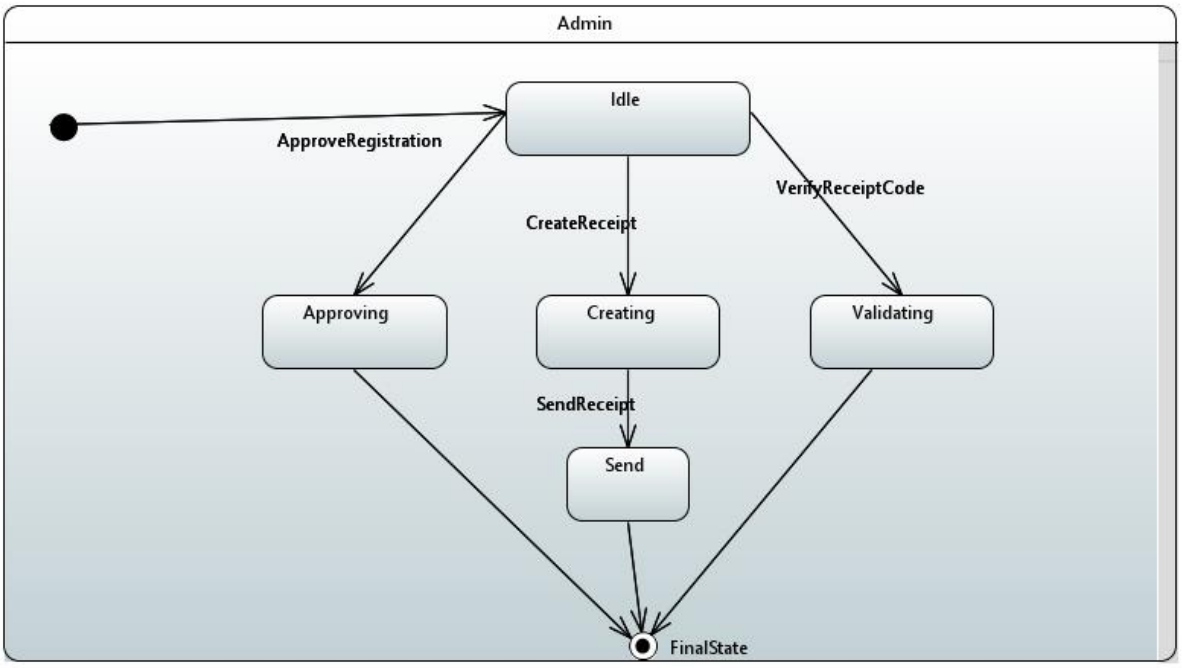
Donor States



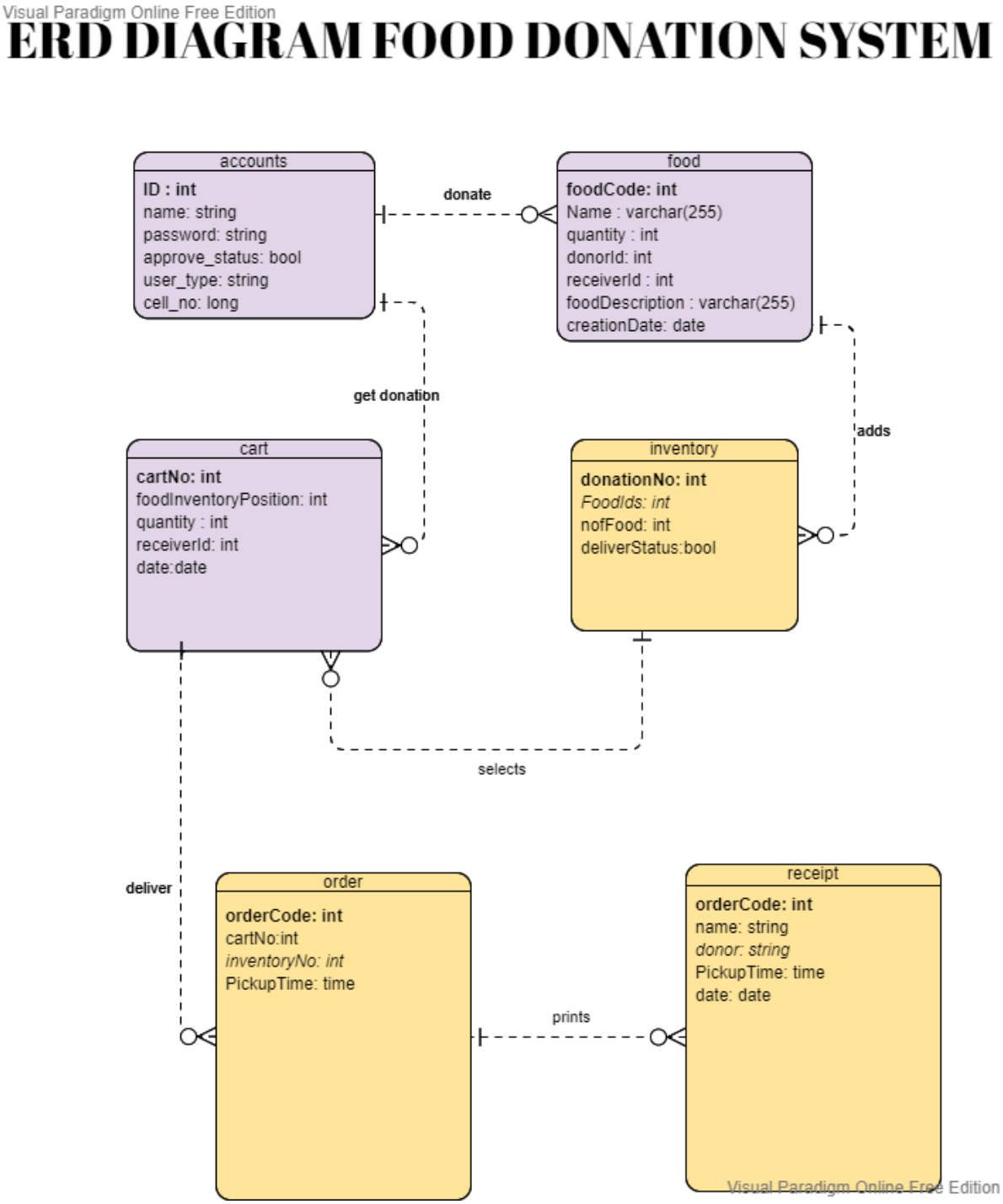
Receiver States



Admin States

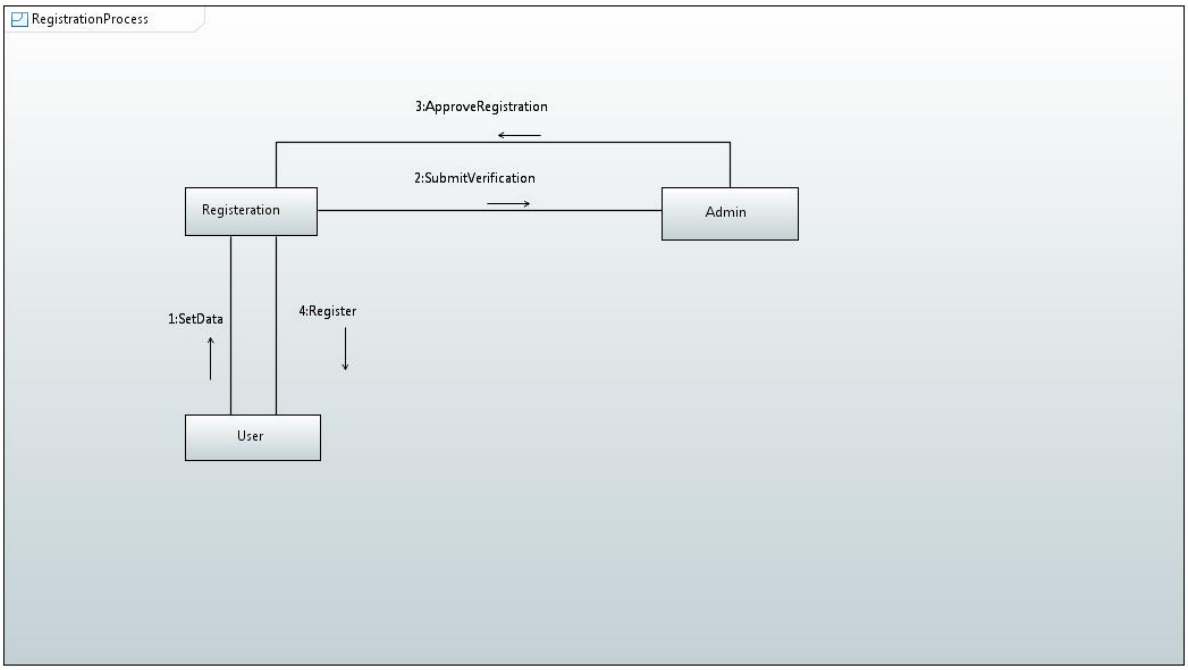


ERD Diagram

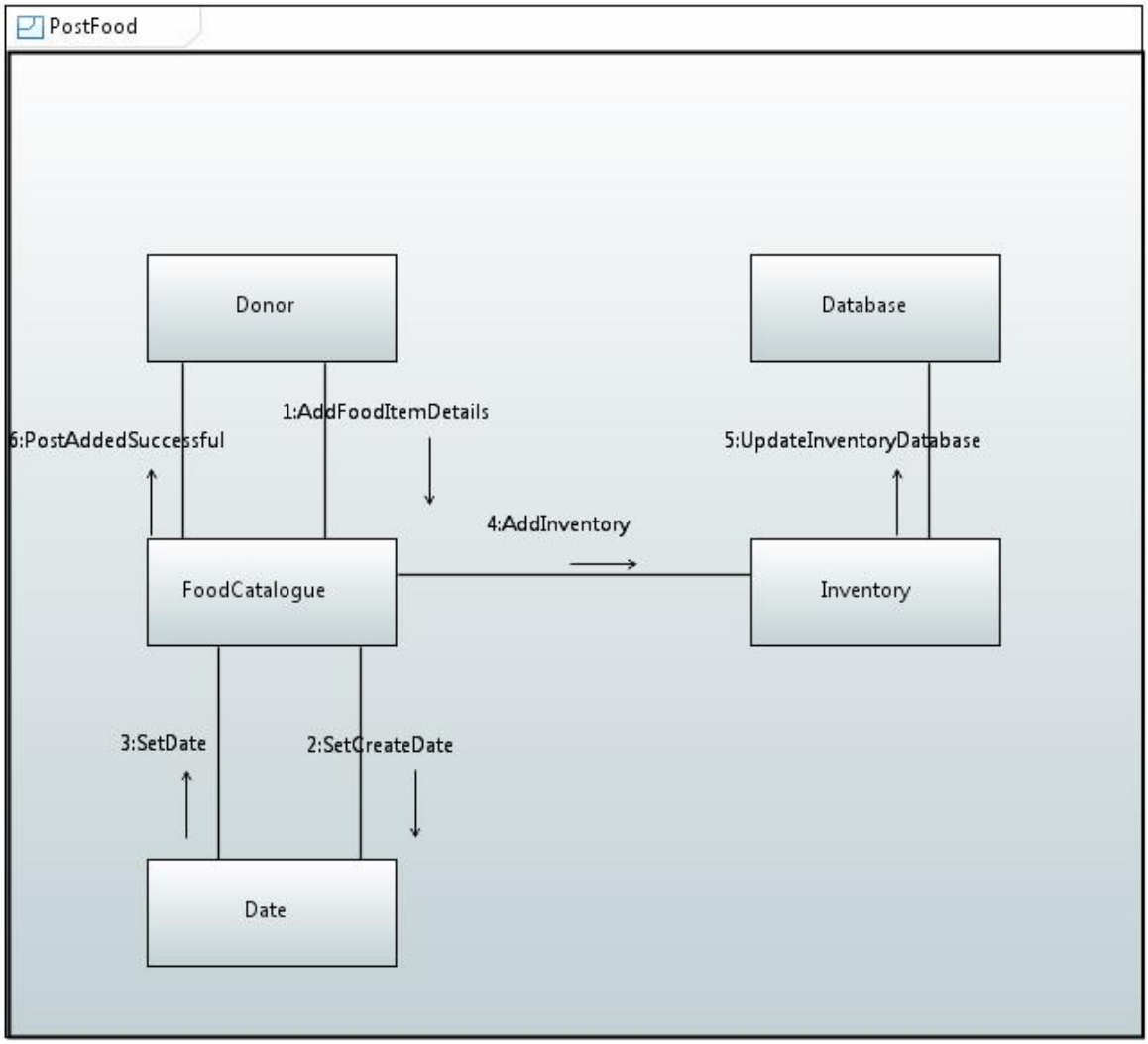

Visual Paradigm Online Free Edition
ERD DIAGRAM FOOD DONATION SYSTEM
accounts
ID : int
name: string
password: string
approve_status: bool
user_type: string
cell_no: long
donate
food
foodCode: int
Name : varchar(255)
quantity : int
donorId: int
receiverId : int
foodDescription : varchar(255)
creationDate: date
get donation
adds
cart
cartNo: int
foodInventoryPosition: int
quantity : int
receiverId: int
date:date
inventory
donationNo: int
FoodIds: int
nofFood: int
deliverStatus:bool
selects
deliver
order
orderCode: int
cartNo:int
inventoryNo: int
PickupTime: time
prints
receipt
orderCode: int
name: string
donor: string
PickupTime: time
date: date
Visual Paradigm Online Free Edition

Communication Diagram

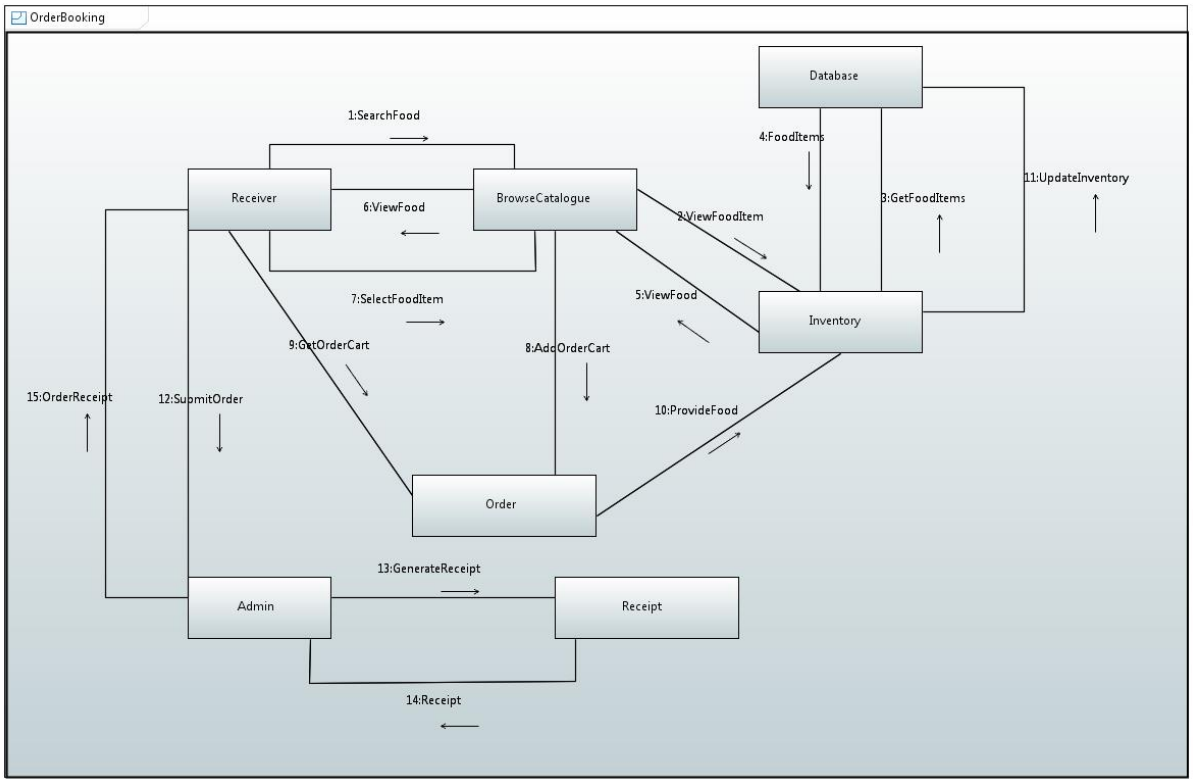
Registration



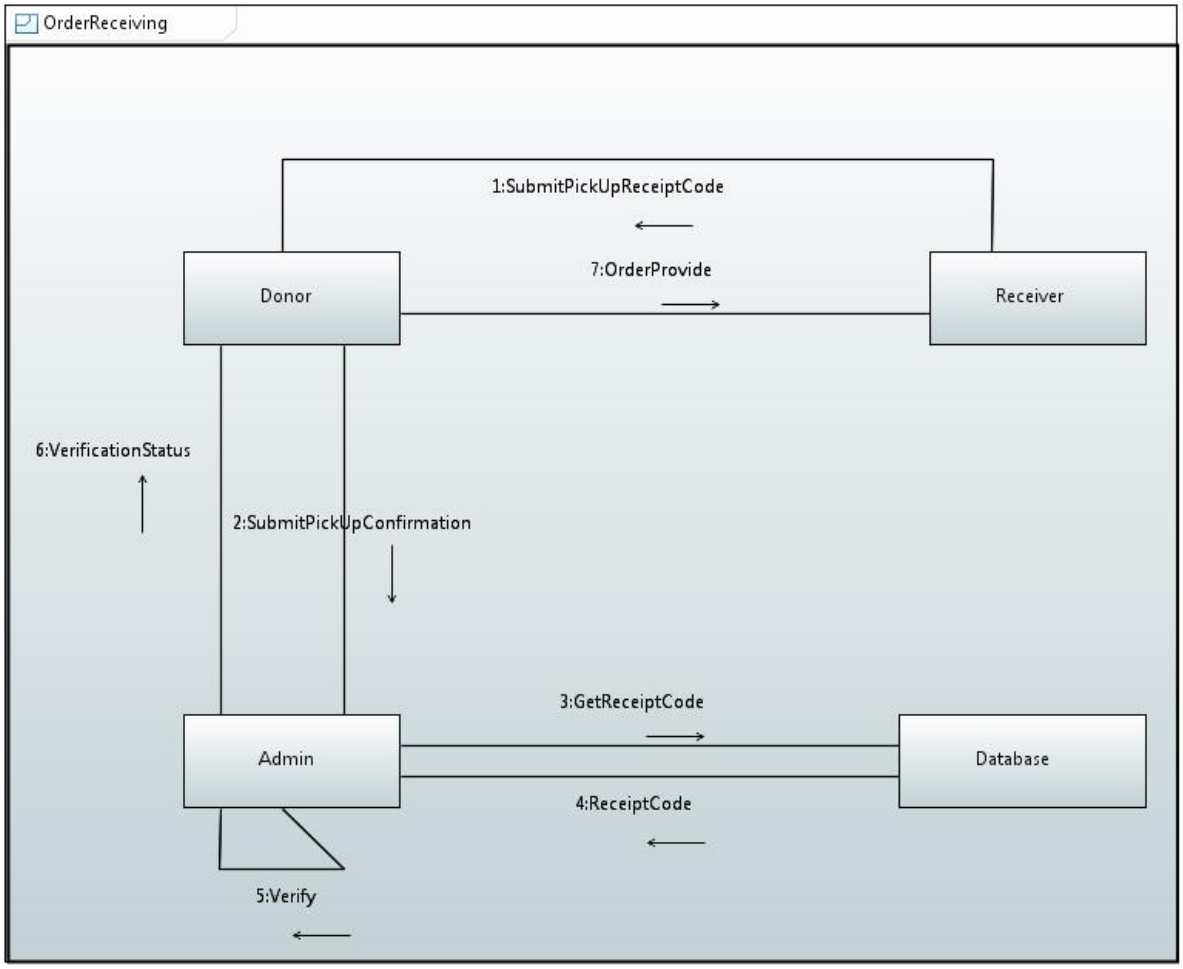
Post Food Item



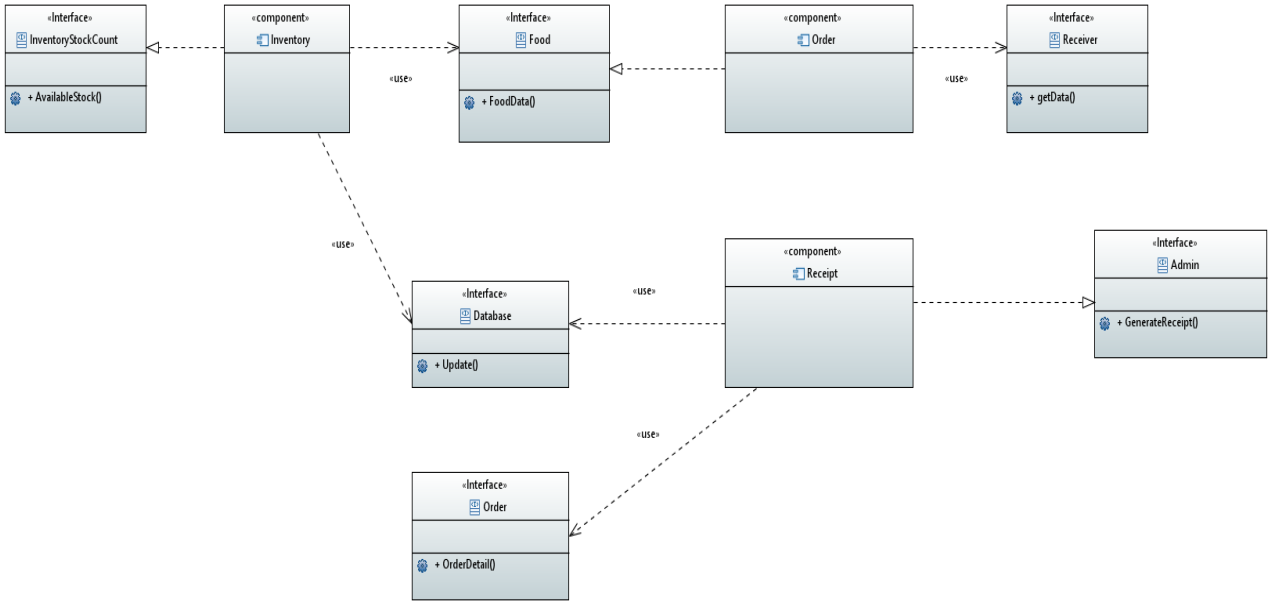
Book Order

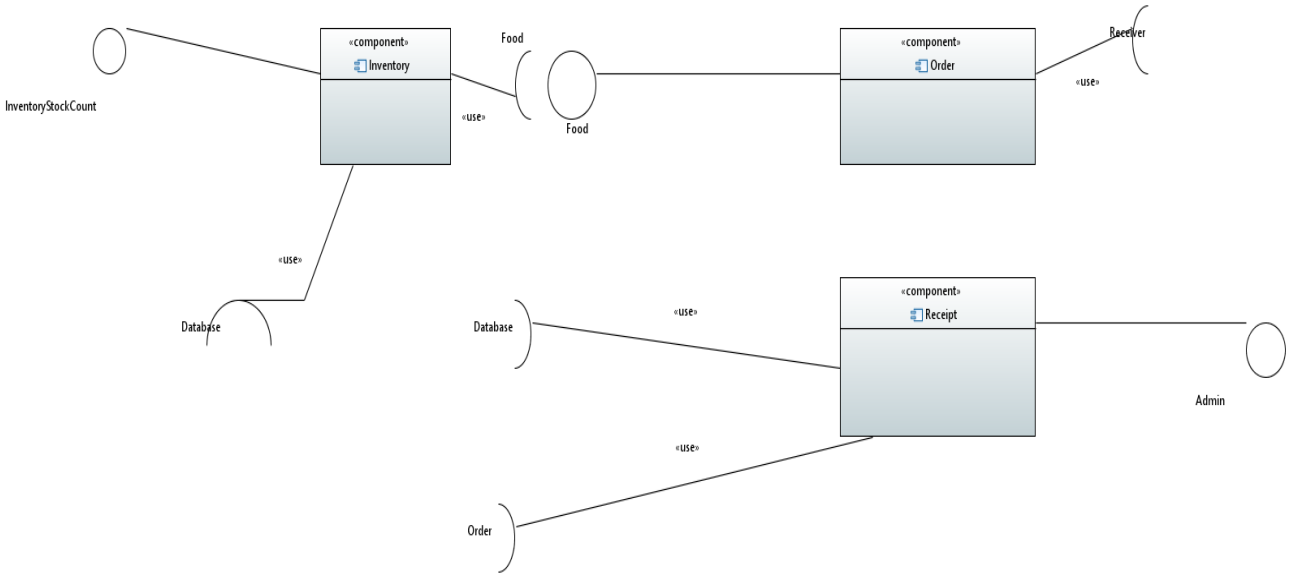

OrderBooking
Database
1:SearchFood
4:FoodItems
11:UpdateInventory
Receiver
BrowseCatalogue
6:ViewFood
3:GetFoodItems
2:ViewFoodItem
7:SelectFoodItem
5:ViewFood
Inventory
9:GetOrderCart
8:AddOrderCart
15:OrderReceipt
12:SubmitOrder
10:ProvideFood
Order
13:GenerateReceipt
Admin
Receipt
14:Receipt

Receive Order

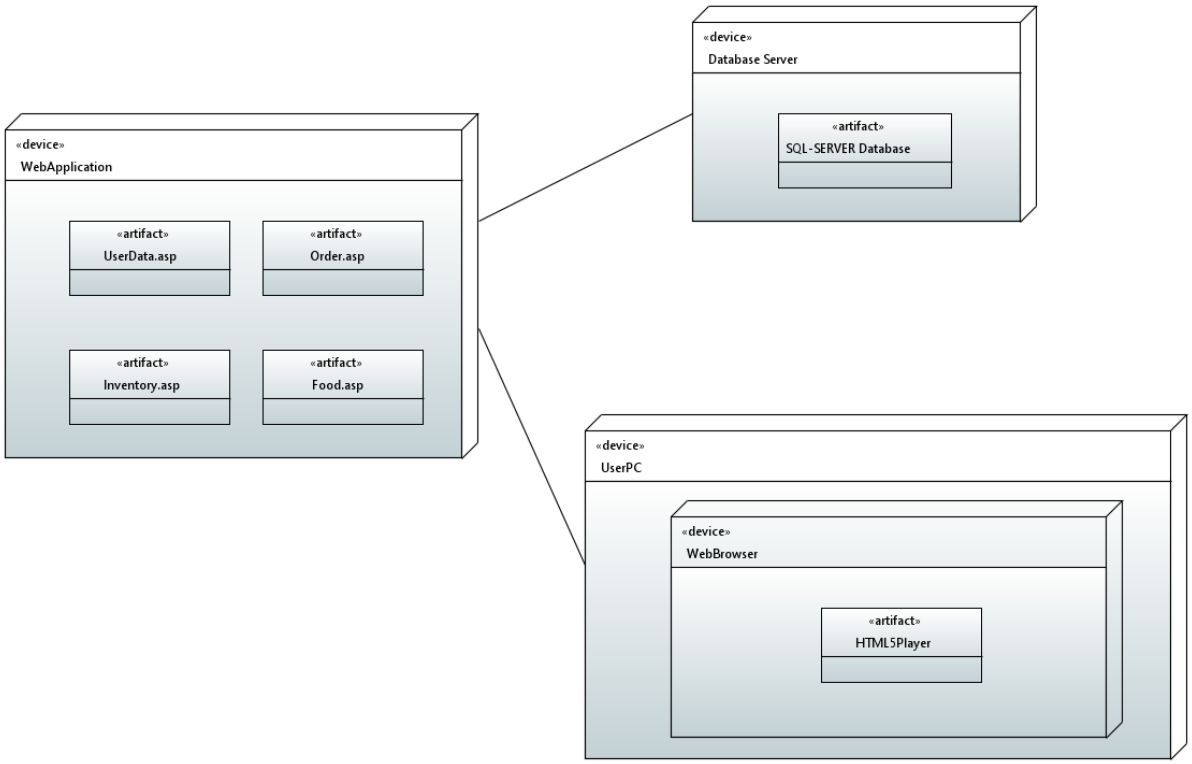


Component Diagram

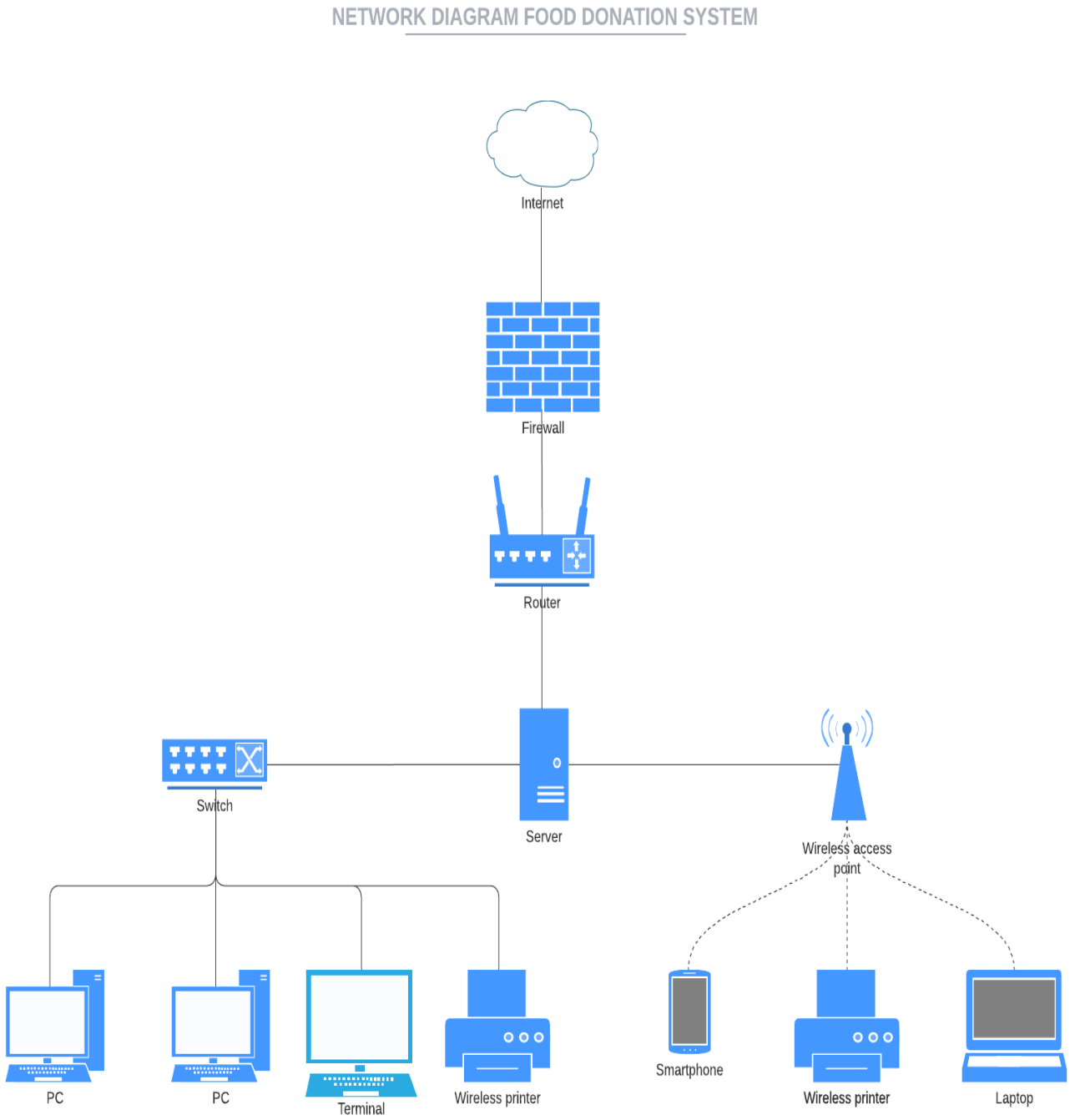




Deployment Diagram

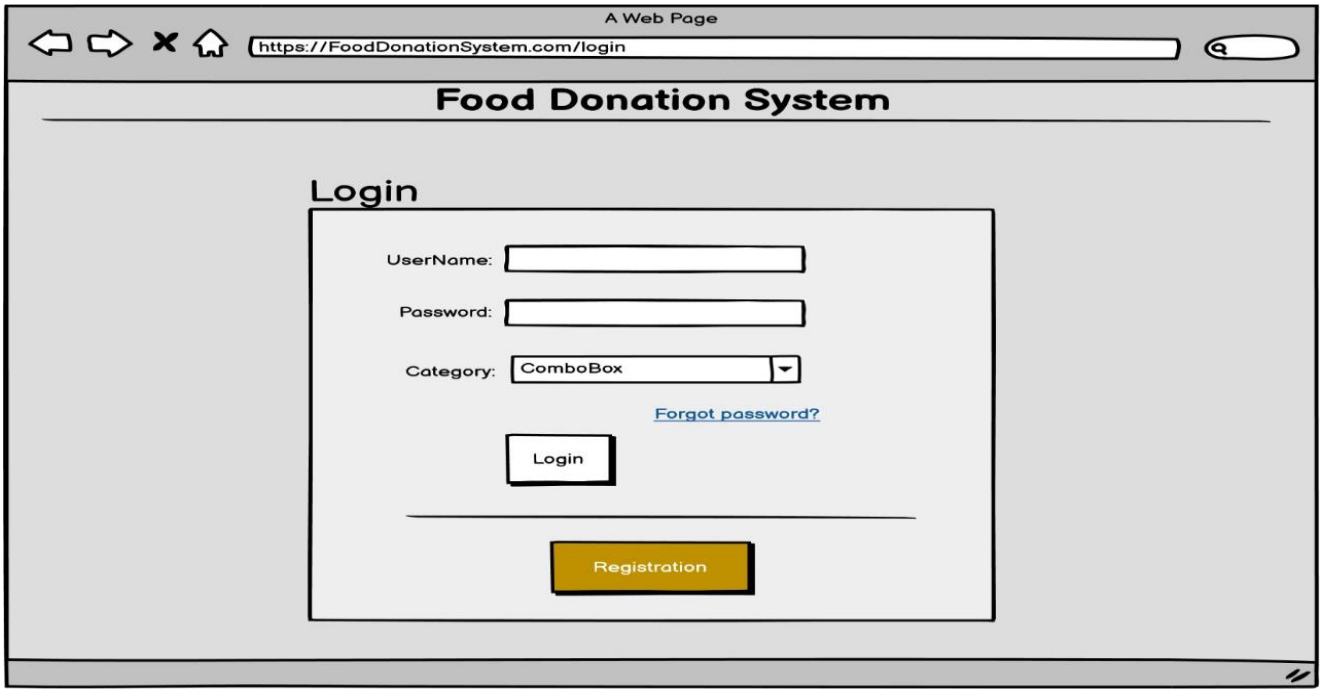


Network Diagram

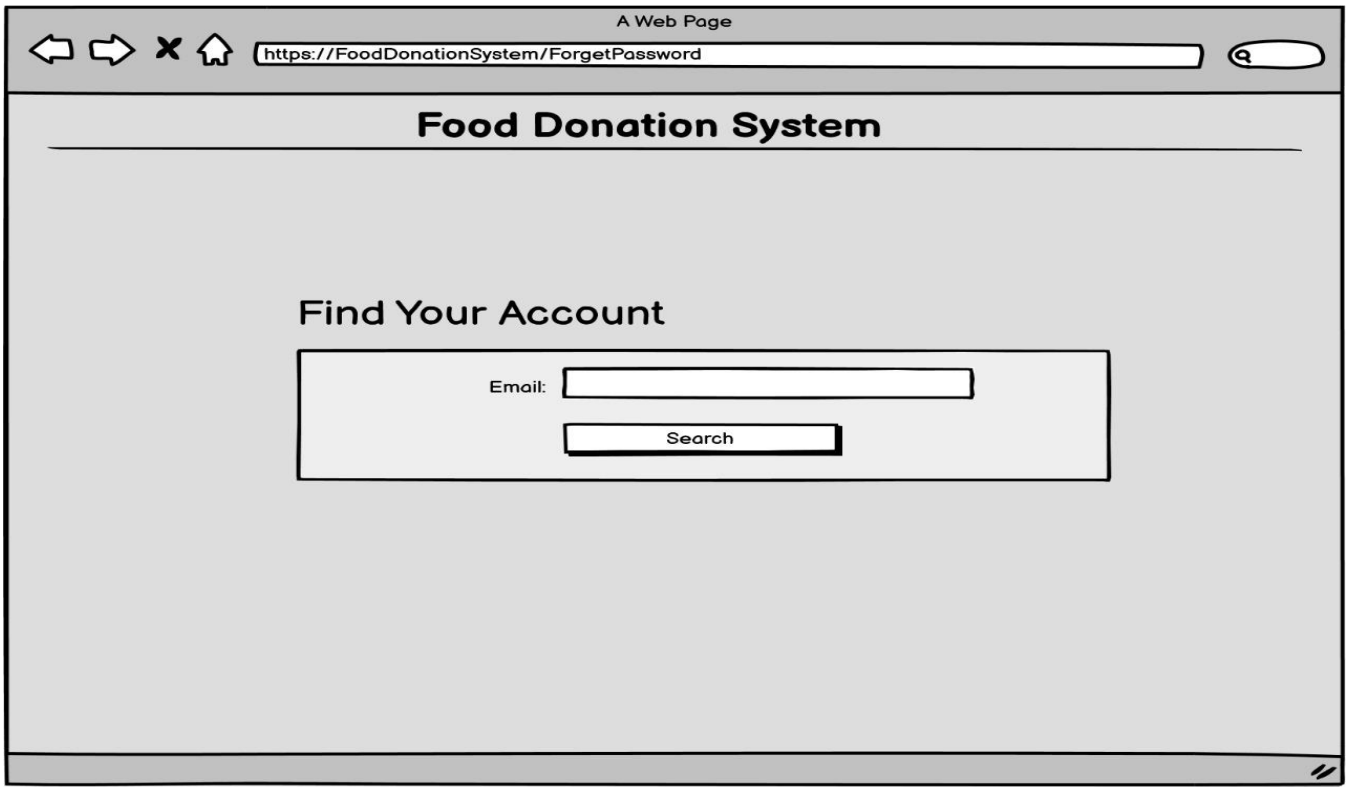


MockUp Images

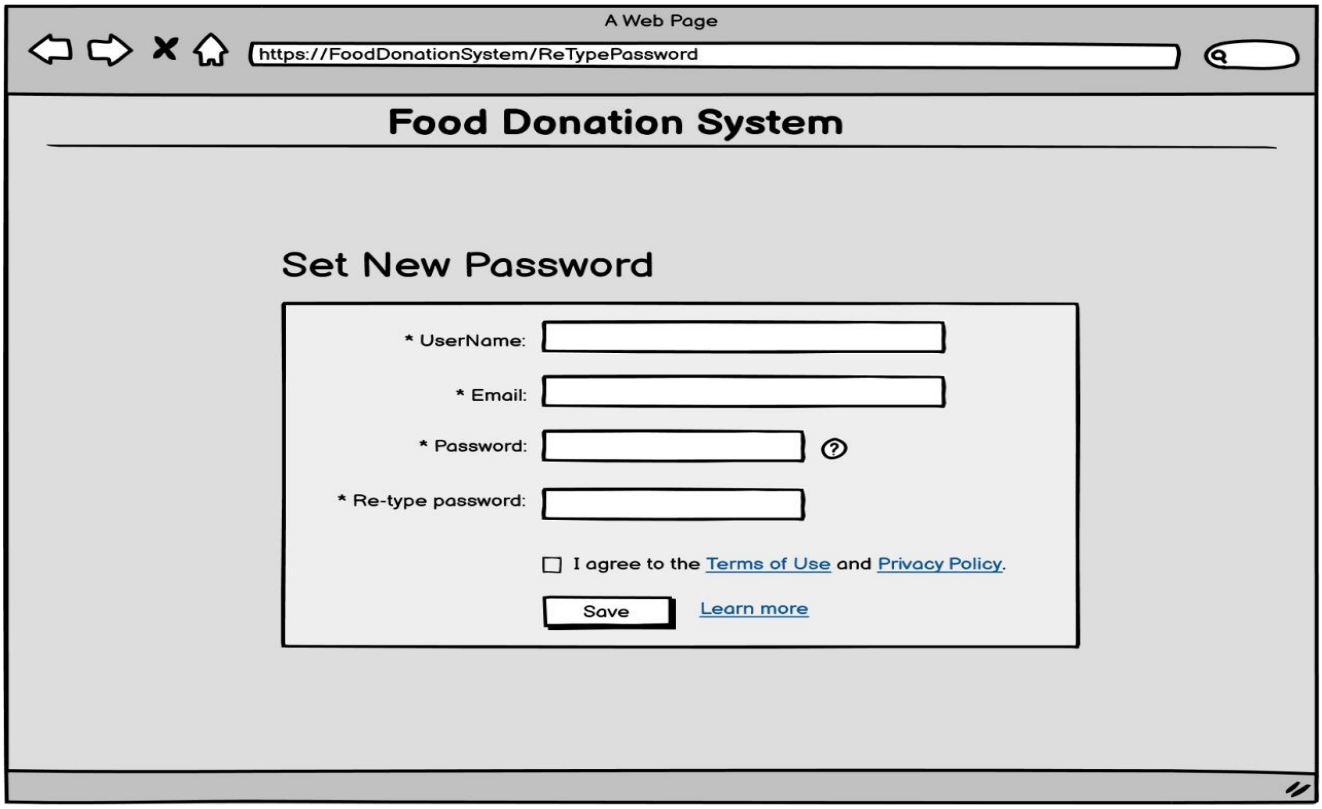
Login System



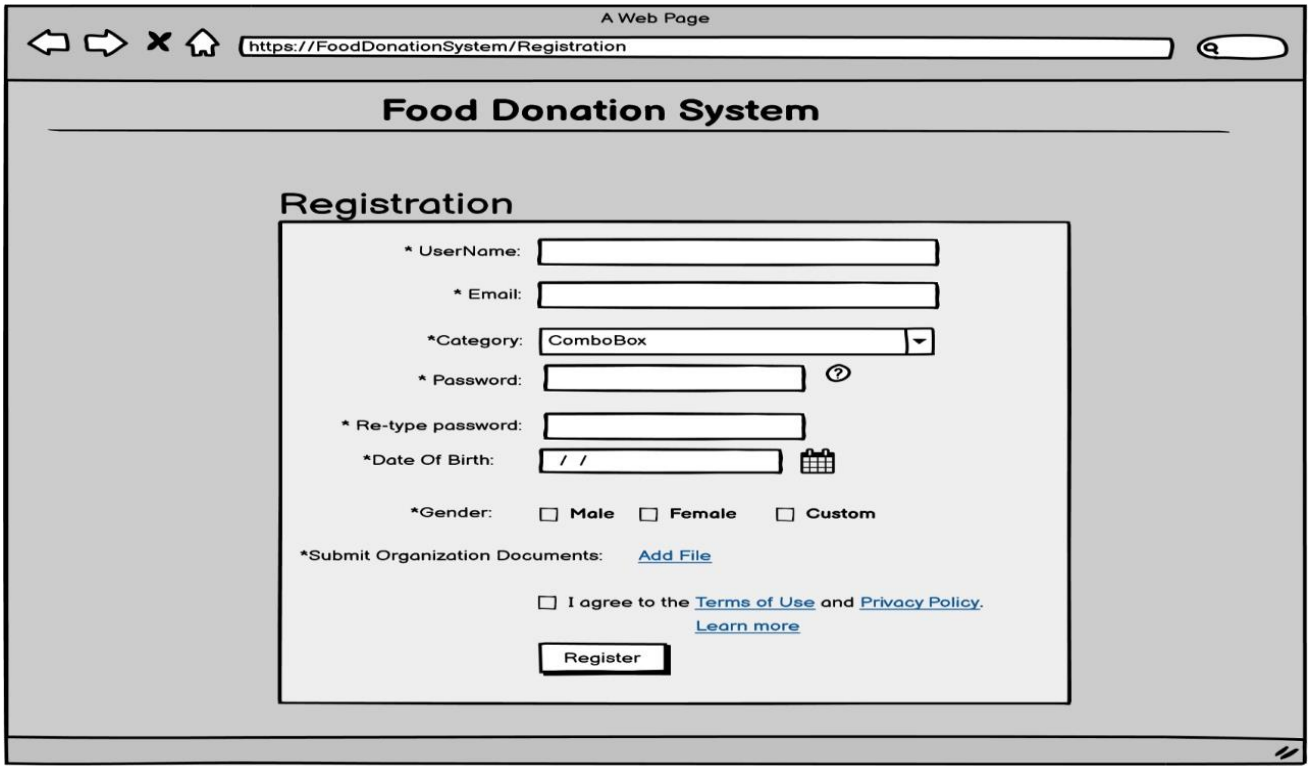
Forget Password



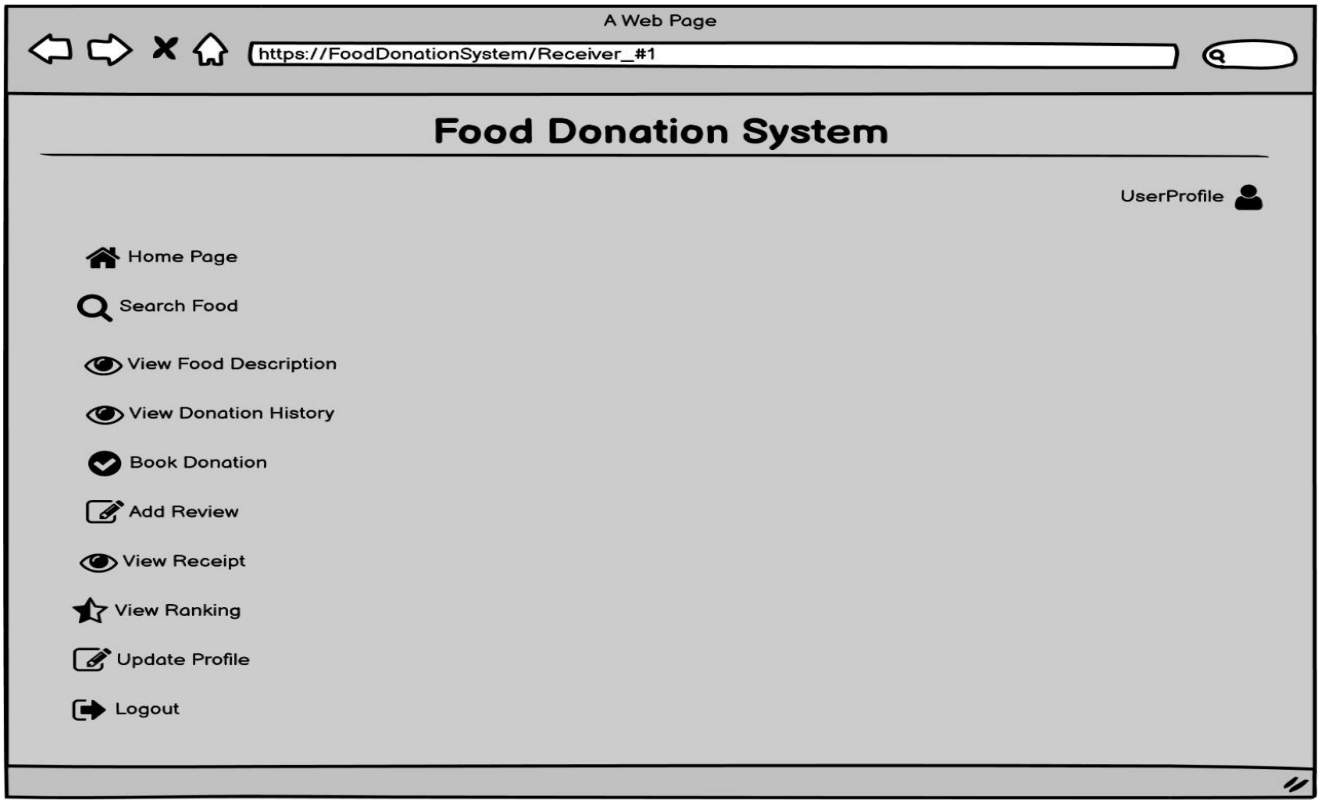
Set New Password



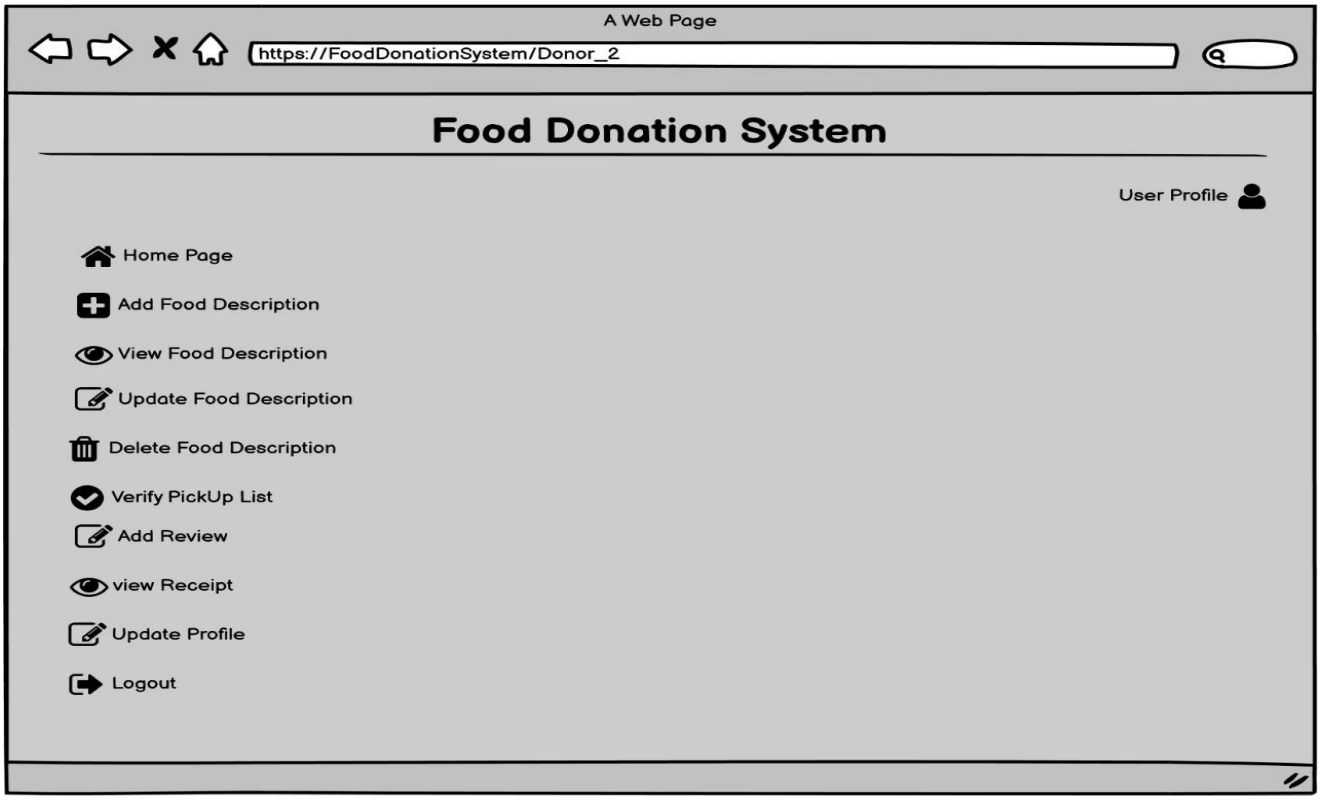
Register User



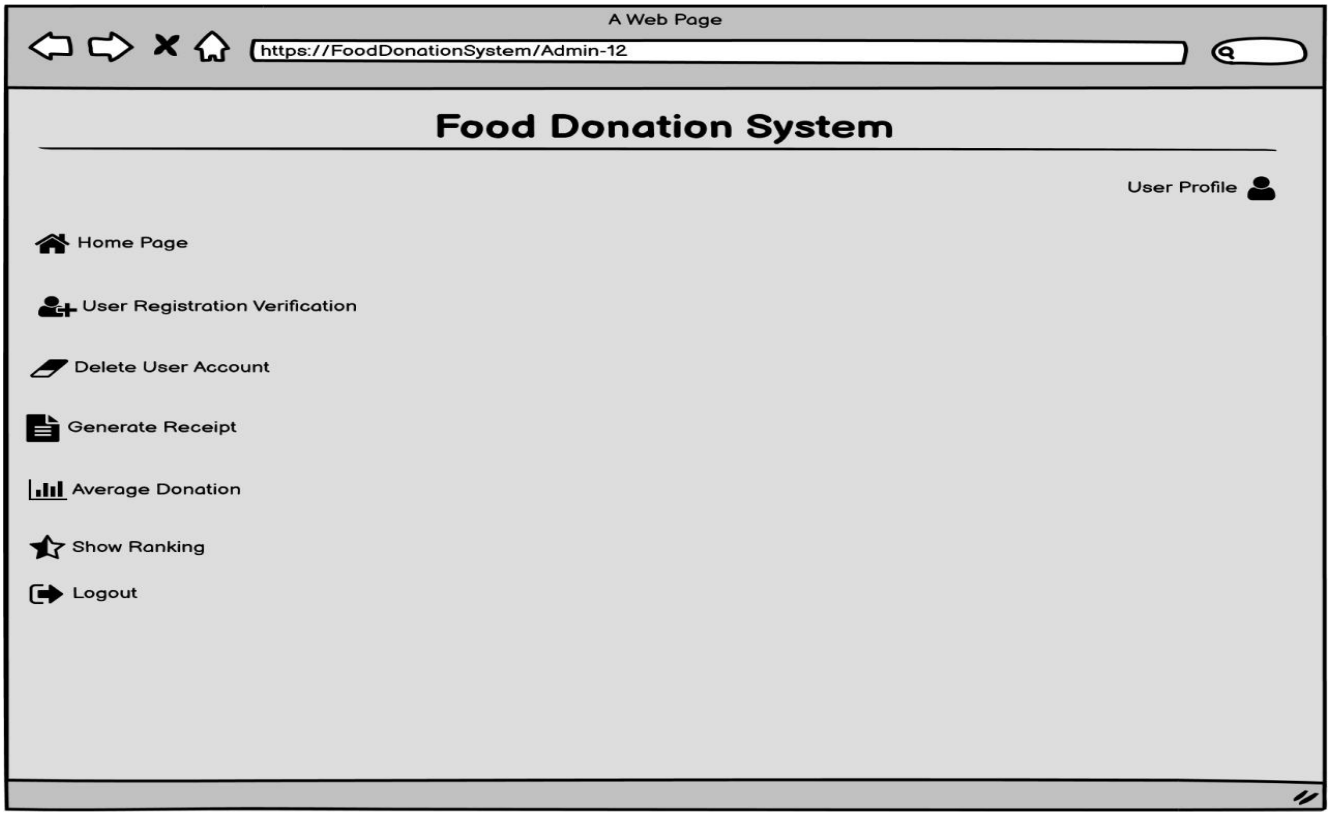
Receiver Home Page



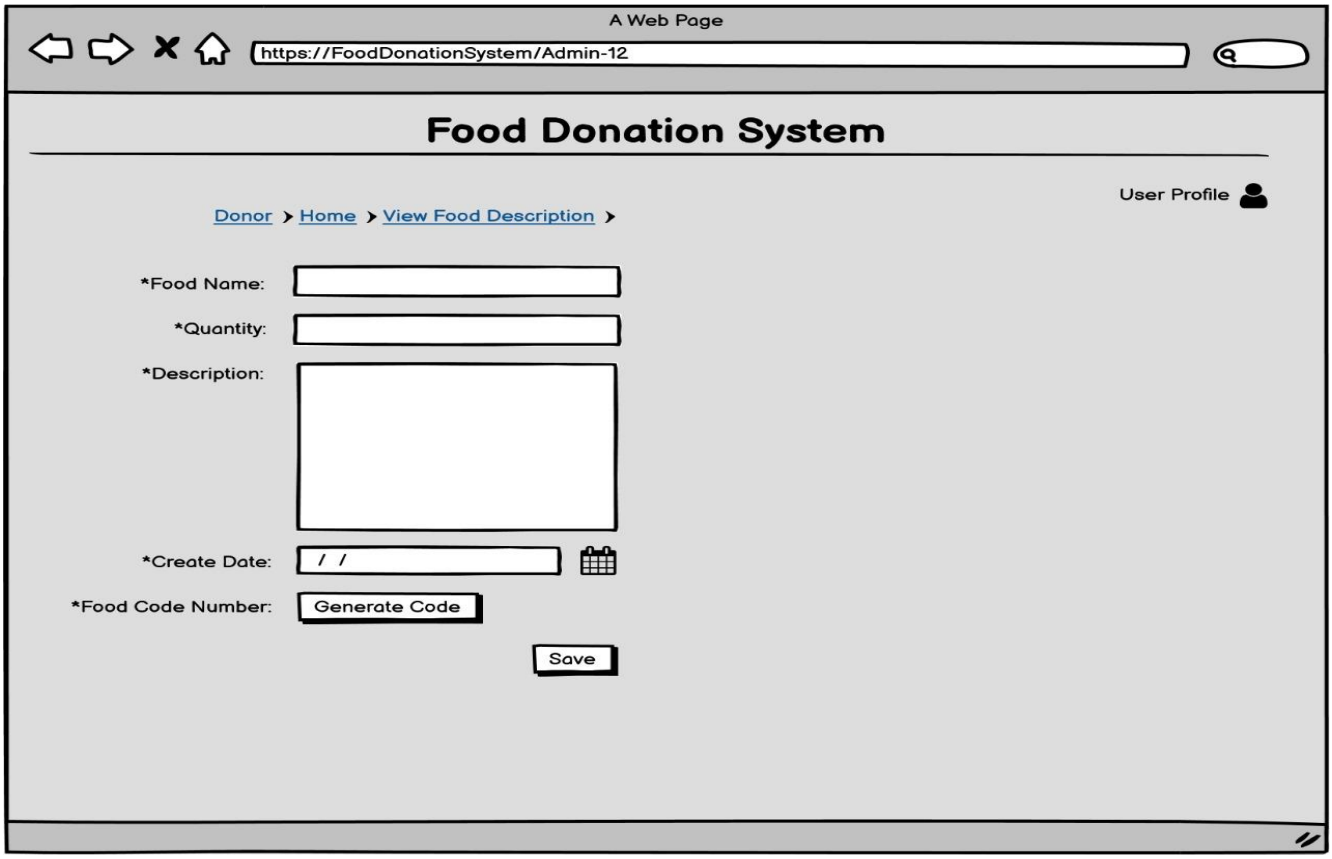
Donor Home Page



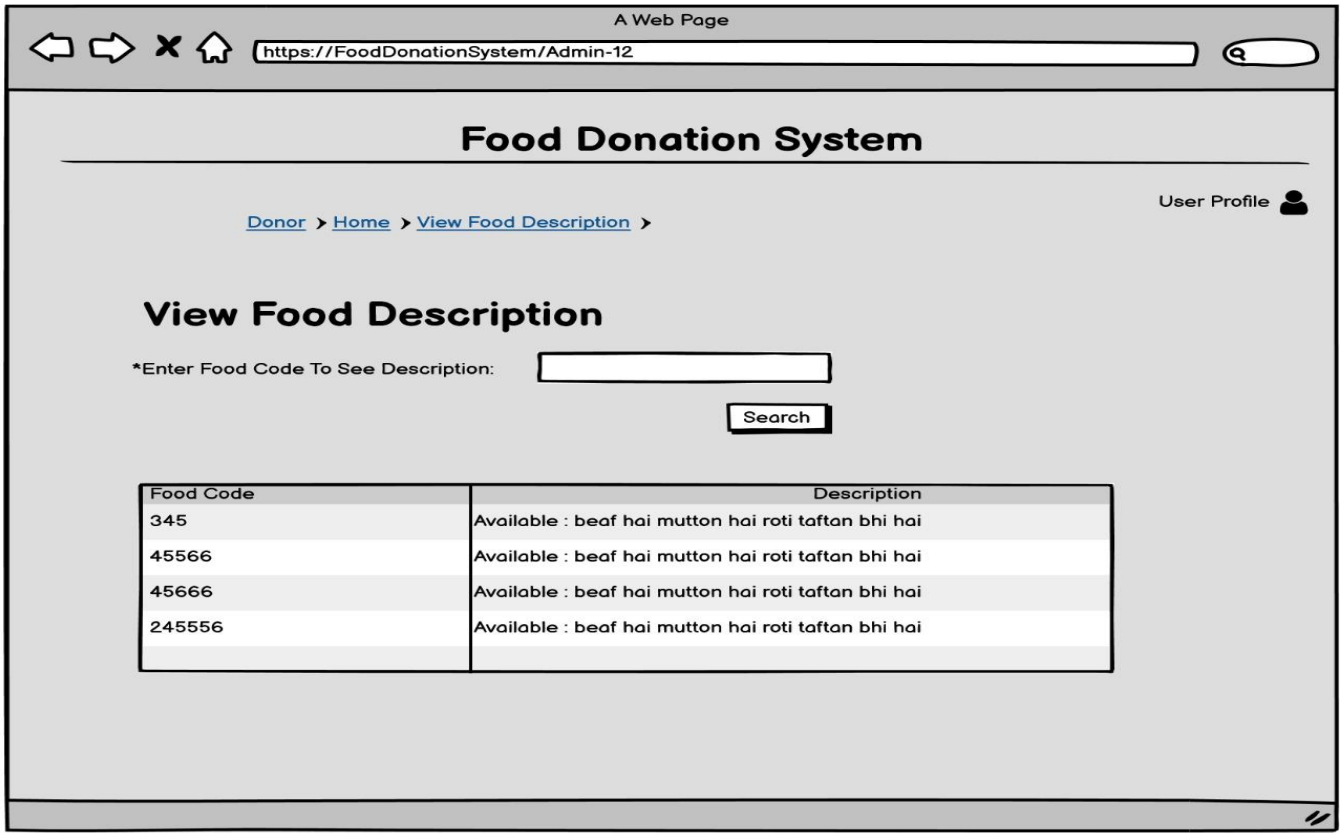
Admin Home Page



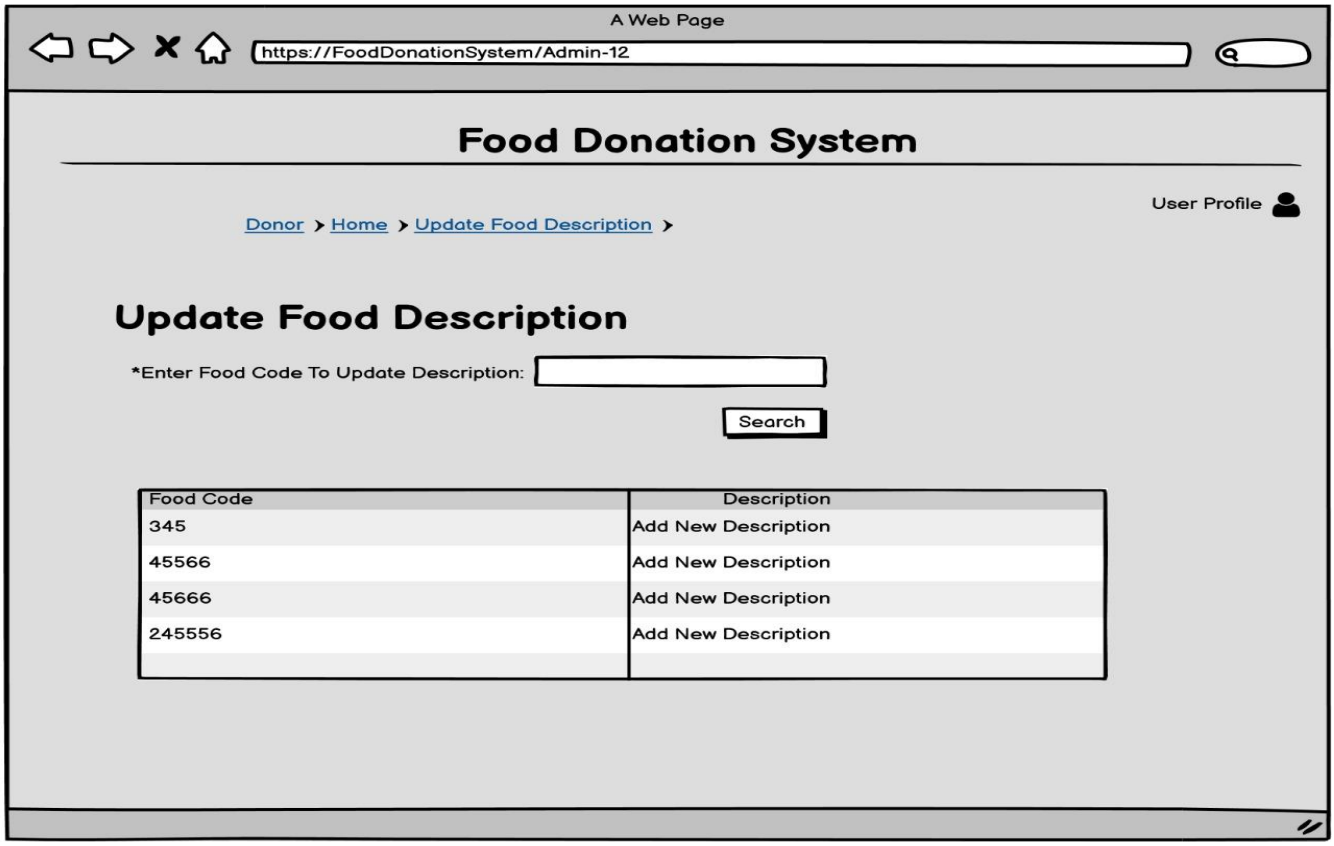
Post Food Item



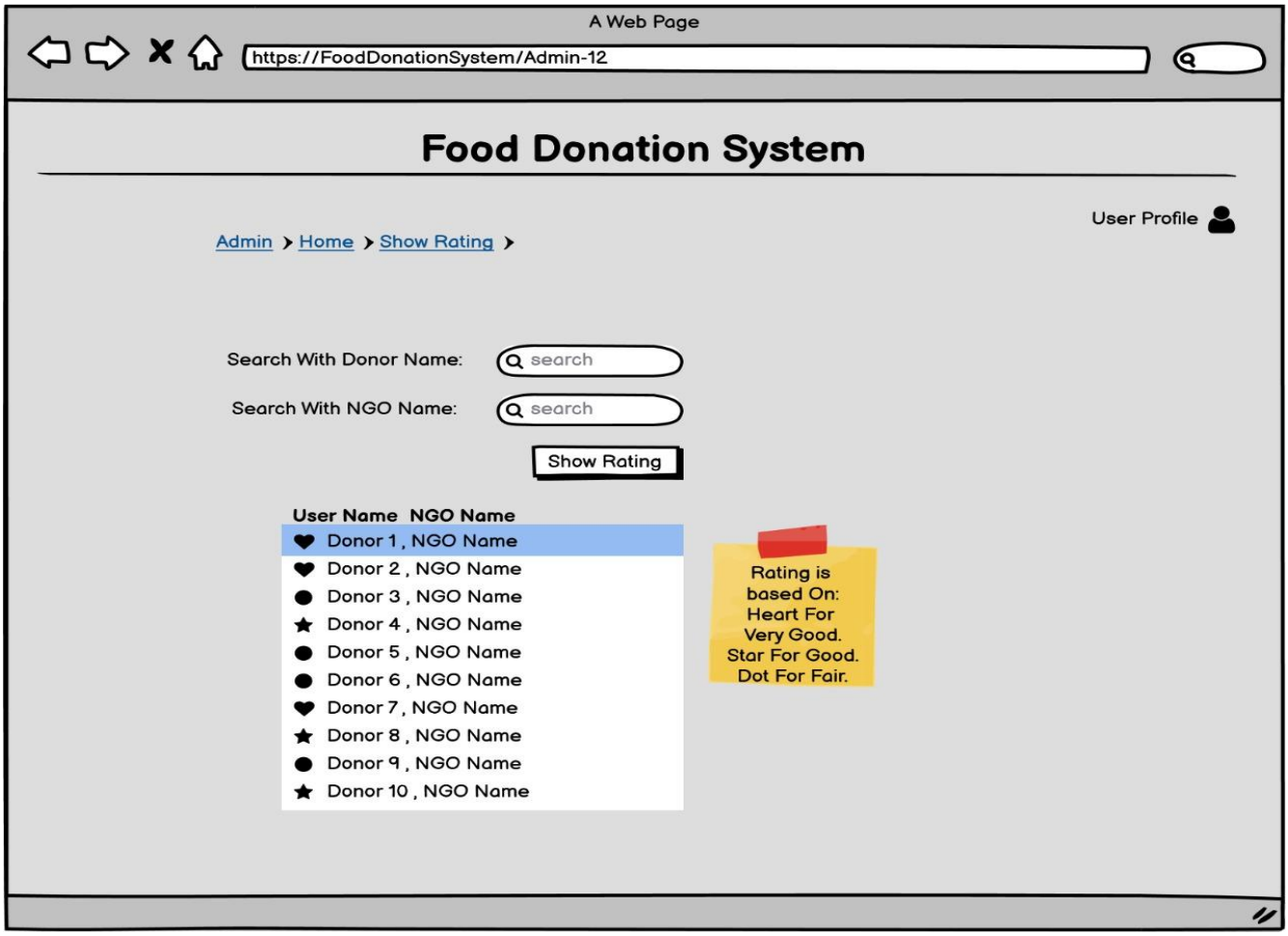
View Food Item



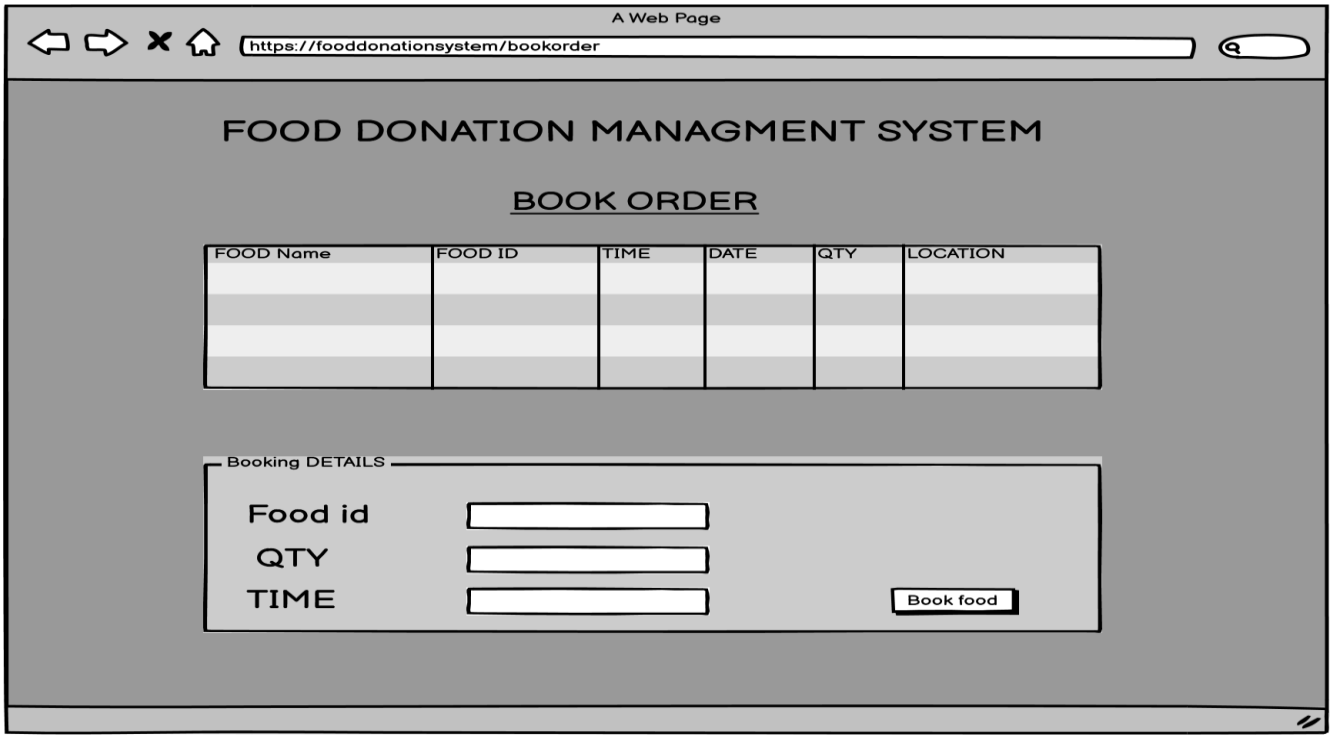
Update Food Description



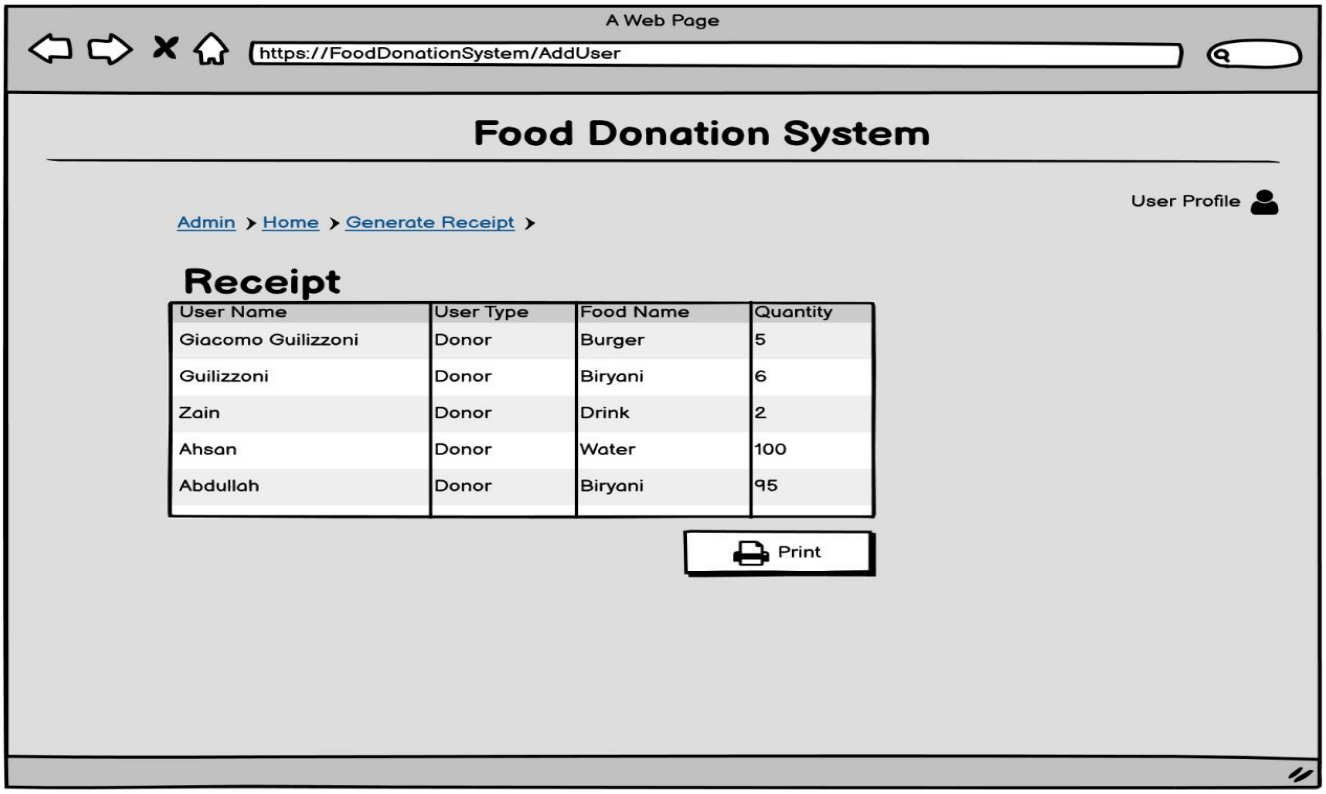
Search Profiles



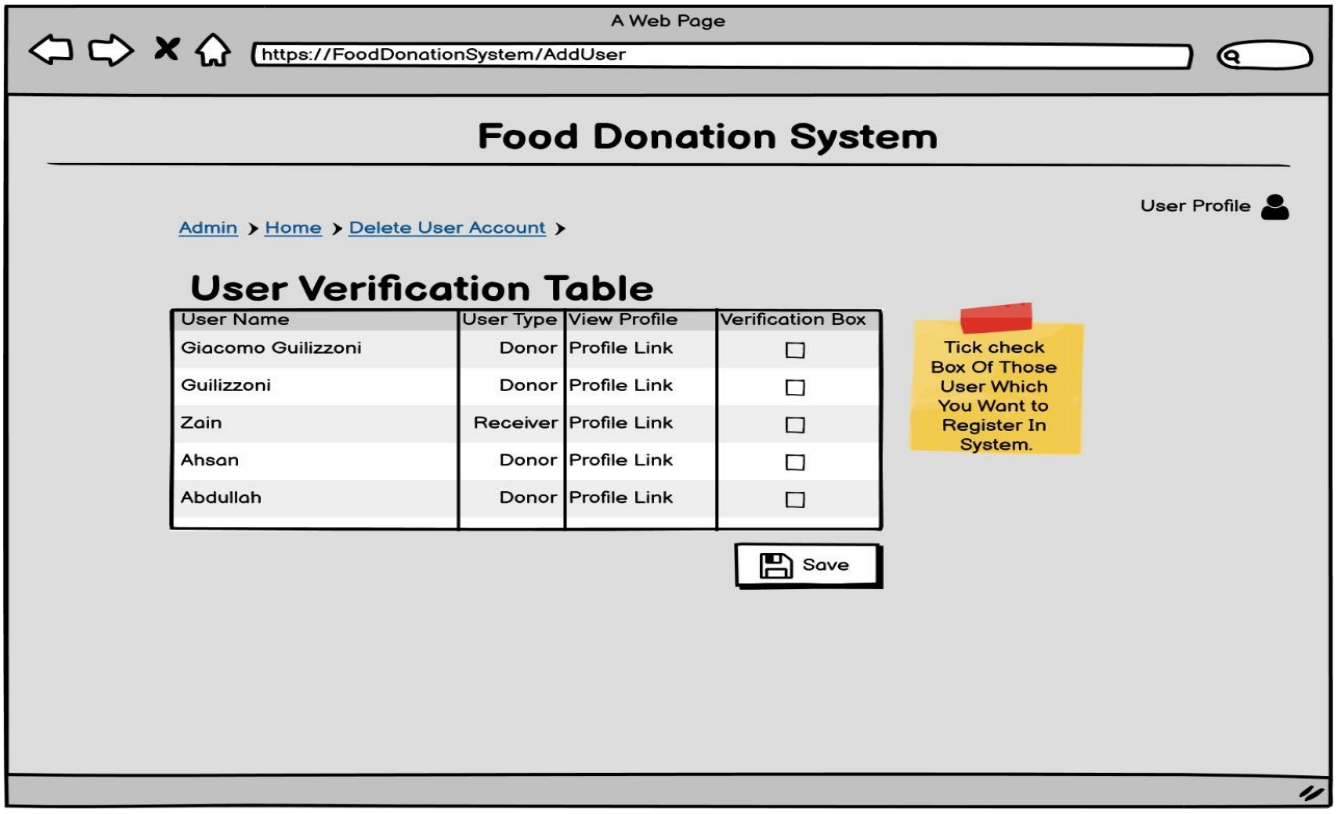
Booking Order



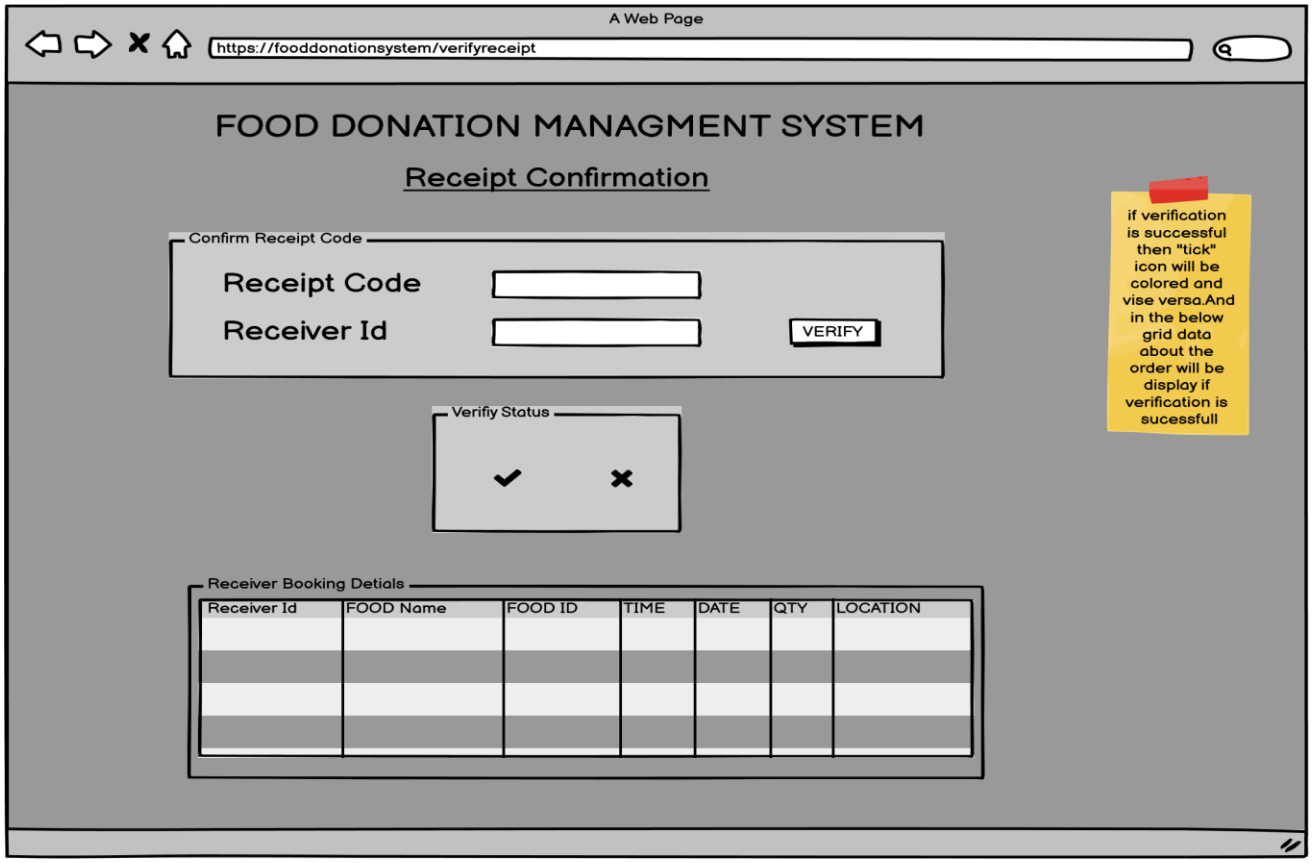
Generating Receipt



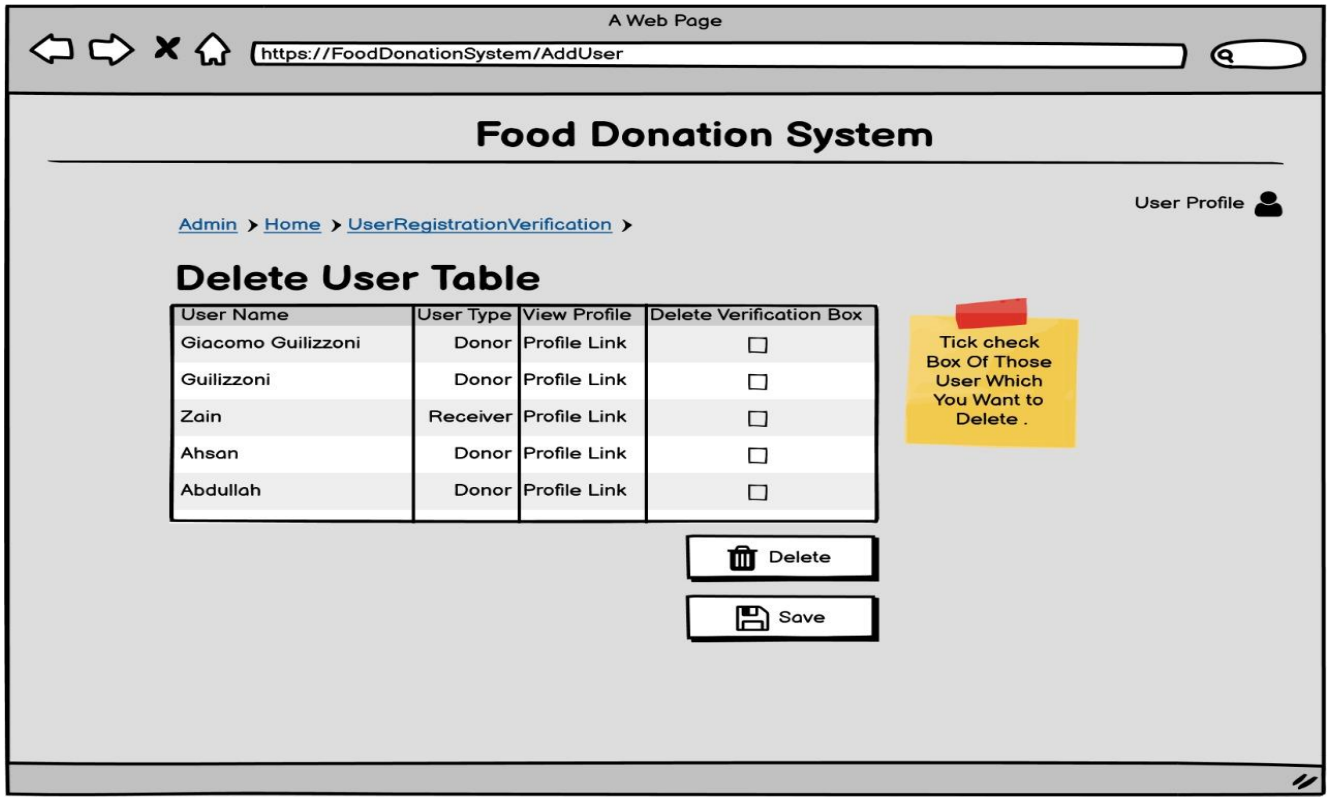
User Verification



Verify Receipt



Delete User Accounts



Calculate Average Donation