

University of Buea

Faculty of Engineering and
Technology

Department of Computer
Engineering

Universite de Buea

Faculte d'ingenieurie et de
Technologie

Genie Informatique

CEF 440: INTERNET AND MOBILE PROGRAMMING

SYSTEM ANALYSIS AND DESIGN

- TENDONGFAC GEORGE NJIMO FE20A116
- DONGHO NONGNI GUIROLE FRANCIS FE20A230
- ARVELLA CHRISTIE PETTOUN TCHOUAKWE
FE20A011
- MBUNGAI MICHAEL BERNARD FOMO FE20A064
- NGUEMGNE FOTSO VINY ANGELE FE20A078

Supervisor:

Dr. NKEMENI Valery

TABLE OF CONTENTS

I. SYSTEM DESCRIPTION

II. SYSTEM ARCHITECTURE

II-a UML DIAGRAMS

1. Use Case Diagram
2. Class Diagram
3. Sequence Diagrams
4. Data Flow Diagram
5. Activity Diagram
6. Entity-Relationship Diagram

II-b USER INTERFACE DESIGN

- Administrator's Architecture
- User's Architecture

I- SYSTEM DESCRIPTION

ReUseEats, is a food waste management application developed to reduce food waste in the society. This hybrid application links those in need and those that have. The application gives the possibility for users to upload food items on the platform for a discount prize or for free. Interested users contact the seller and arrange a meet-point through the application by accessing the seller's profile.

Whenever a user uploads a new food item on the system, all others users are notified through via their email addresses and within the app as well. Also, users can report others users due to non-respect of the negotiation terms.

However, ReUseEats does not handle payment transactions of food items between users. The aim of this application is to link the users.

II- SYSTEM ARCHITECTURE

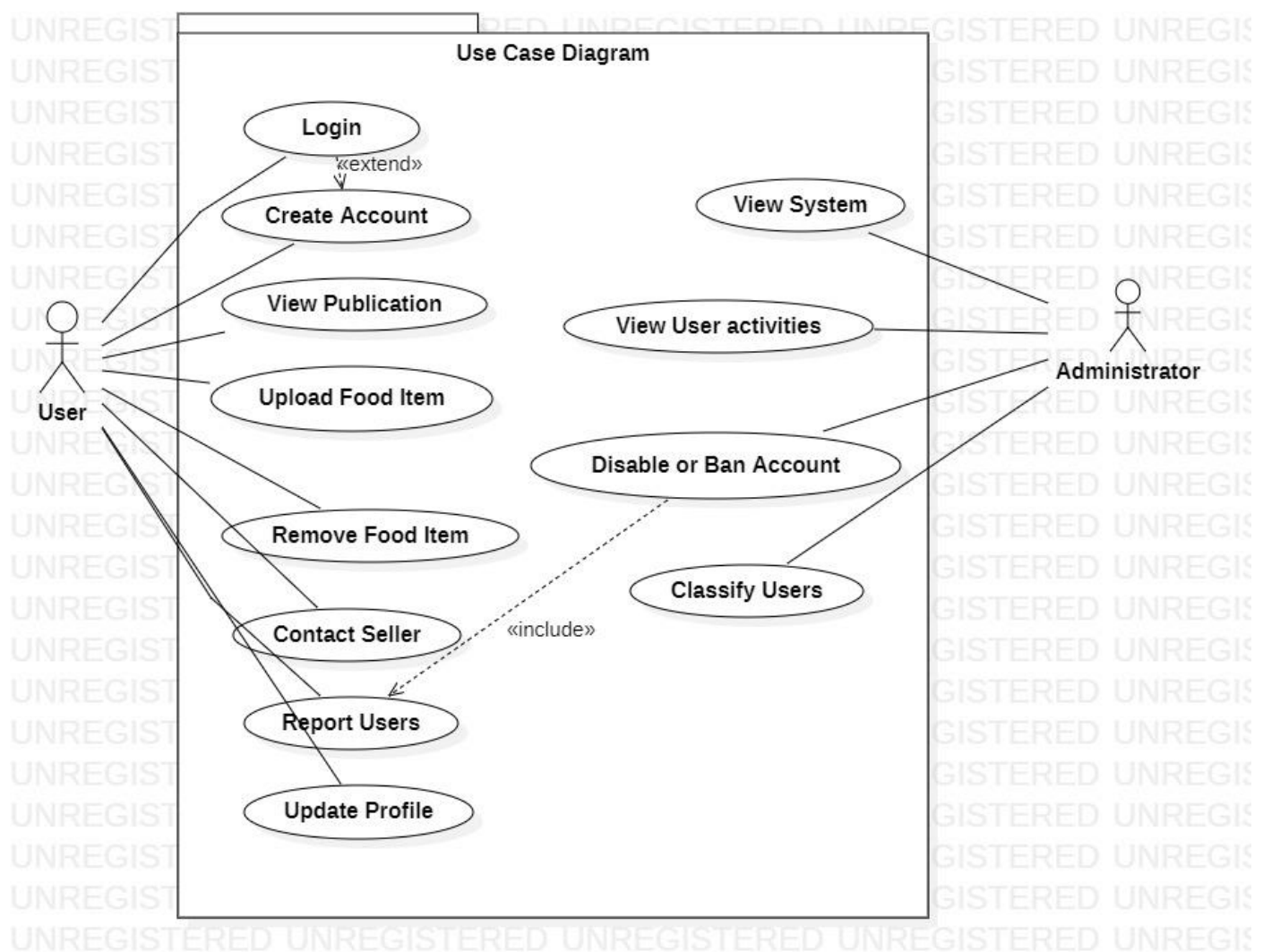
II-A UML DIAGRAMS

Using StarUml, we realized the following UmL diagrams.

1. USE CASE DIAGRAM

The main actors within the system are:

- A. User(client)
- B. Admin



Textual Description

a. Login

Description: Allows the user to access the system and interact with other users in the system.

Pre-condition: The user should have created an account before accessing the system through login.

Post-condition: After logging into the user's account, the user is taken to the home page.

Main Flow: create account, login.

b. Create account

Description: Allows the user to create a login and be a registered user.

Pre-conditions: This use case starts when the user accesses the system feature that enables him to create an account by entering information that is maintained in the user's account.

Post-conditions: After the account is created, the user logs in normally by entering the required credentials.

Main Flow: enter personal information, create account.

c. View food stuff

Description: The available food items from users are displayed for other users.

Pre-condition: The user should login first before the user can view food items.

Post-condition: after viewing the food items the user can choose and access the various profiles.

Main Flow: Enter login credentials, view food stuff.

d. Upload food stuff

Description: The user can upload food items to his/her profile for other users to view.

Pre-condition: The user should login first before he/she can upload food items to the system.

Post-condition: After uploading the food items, other users can view the food items.

Main Flow: enter login credentials, upload stuff.

e. Remove food stuff

Description: Here after uploading the food item, he/she can remove the food item from the system.

Pre-condition: The user should upload a food item.

Post-condition: Other users cannot view the removed food items.

Main Flow: enter login credentials, homepage, upload food stuff, remove food stuff.

f. Contact seller

Description: Other users can contact a user if interested with a particular food item on that user's profile through email or telephone number.

Pre-condition: A user should have a food item uploaded on his/her profile in the system.

Post-condition: after a food item is no more available the user removes the concerned food item.

Main Flow: Enter login credentials, view food stuff, view user profile.

g. Report other users

Description: when a user does not follow the terms of the transaction, the other user can report that user.

Pre-condition: when a food item has been uploaded that is when other users can access the profile for a possible transaction.

Post-condition: the admin goes on to implement the necessary punishment.

Main Flow: enter login, enter homepage, enter user profile then access report.

h. View the system

Description: Here, the admin views the system.

Pre-condition: The admin must login in to the system.

Post-condition: view the various services the system offers.

Main Flow: enter login credentials, view the system.

i. View user activities

Description: The various activities of the users are viewed by the admin such as creating an account.

Pre-condition: Users should exist in the system.

Post-condition: The admin can have an estimate of the number of users, uploaded images and traffic in the system.

Main Flow: enter login credentials, then view the various activities through the admin's dashboard.

j. Barn/disable users

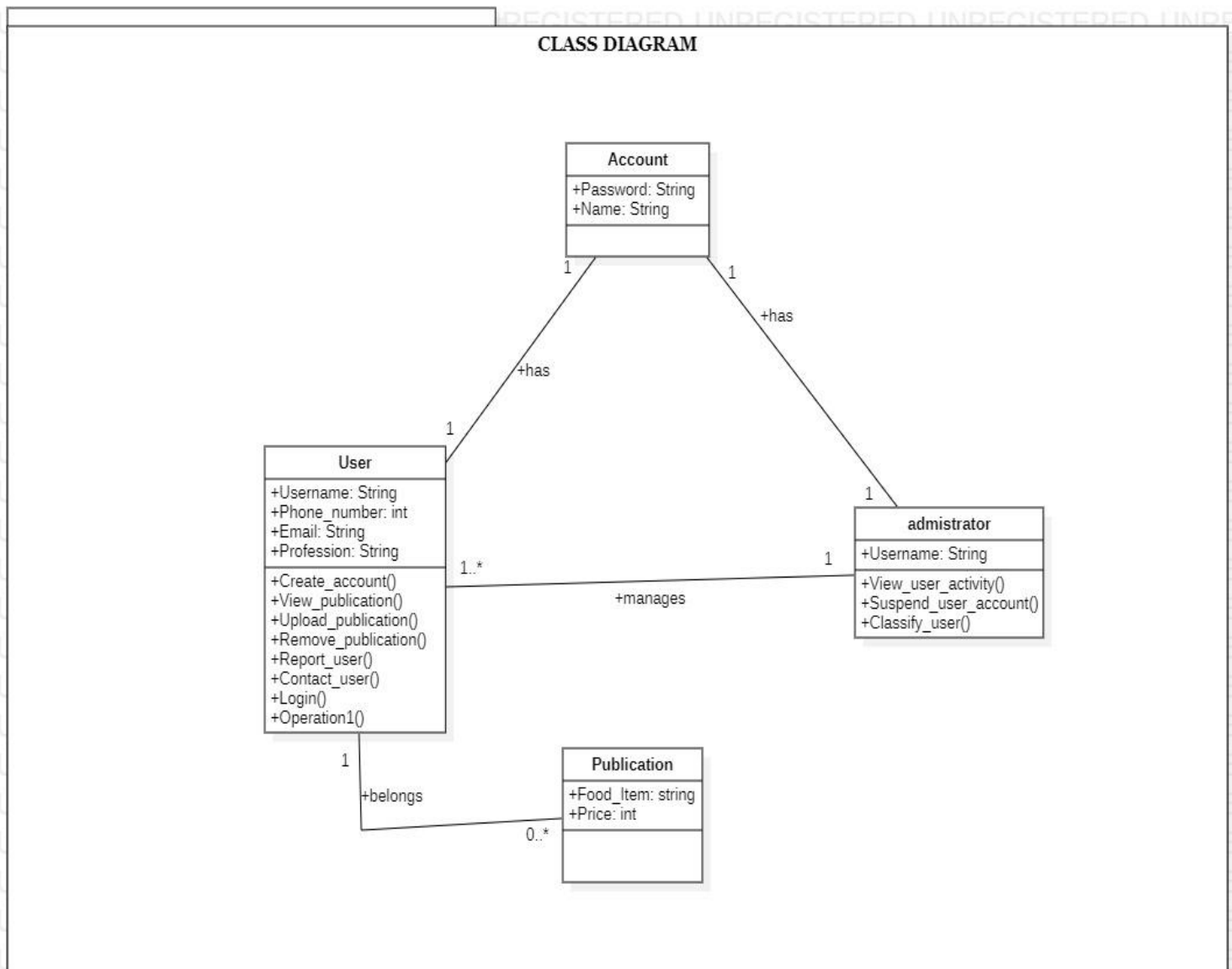
Description: The admin punishes users by disabling users or barning the user's account.

Pre-condition: users must have disobeyed transactional rules a certain number of times.

Post-condition: The concerned user won't be able to access his/her account for a period of time or definitely.

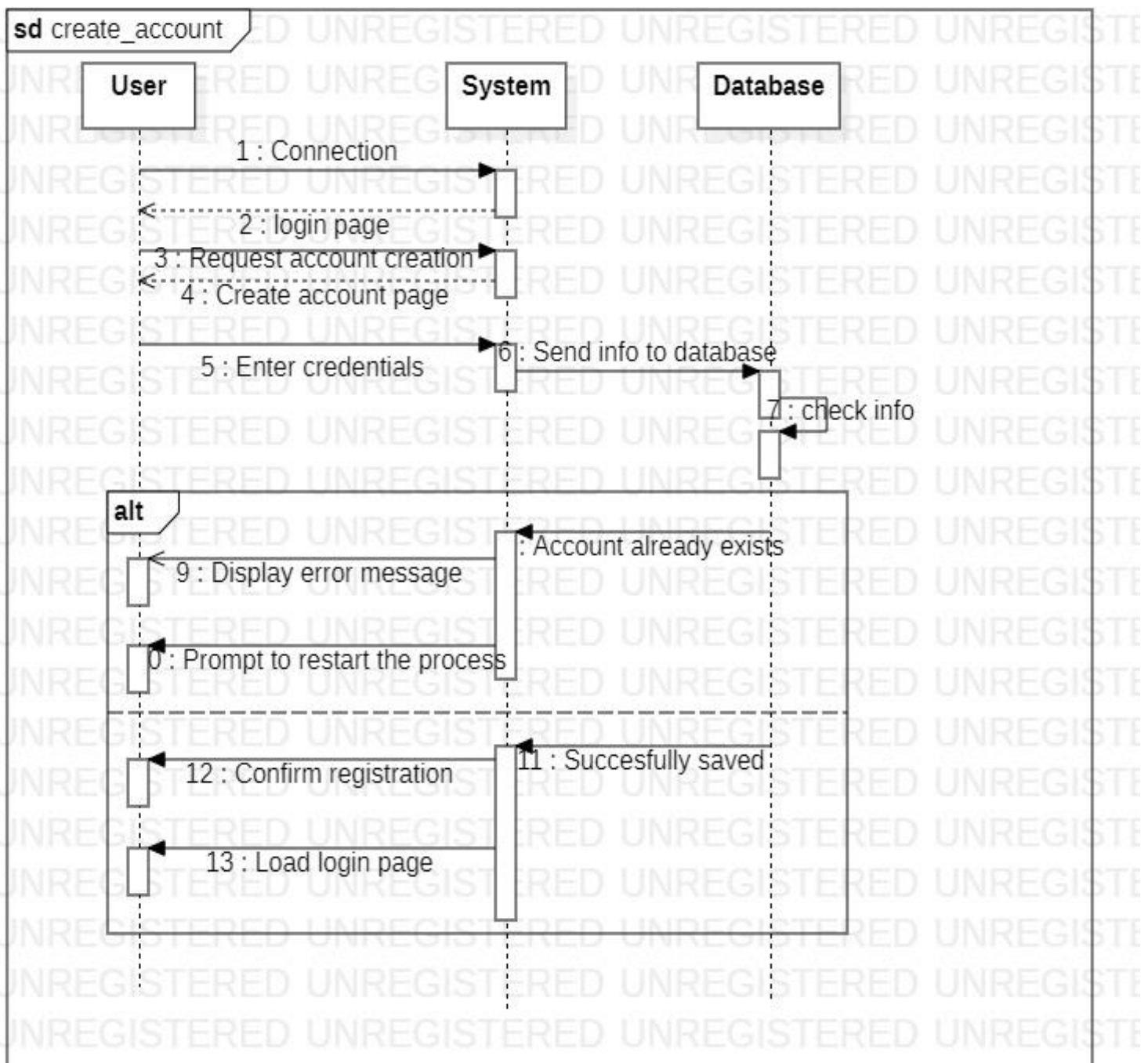
Main Flow: enters login credentials, enters the admin dashboard.

2. CLASS DIAGRAM

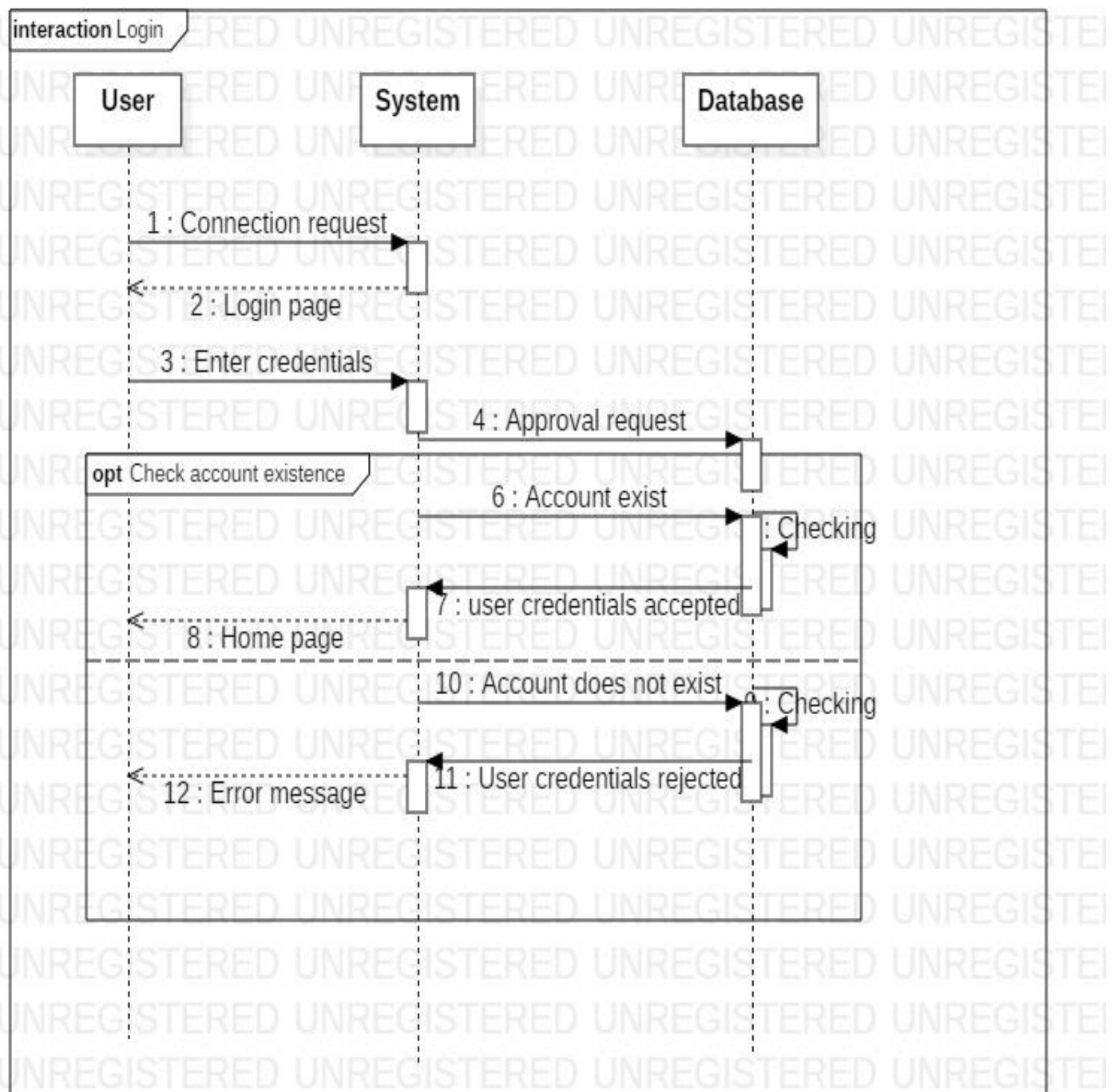


3. SEQUENCE DIAGRAM

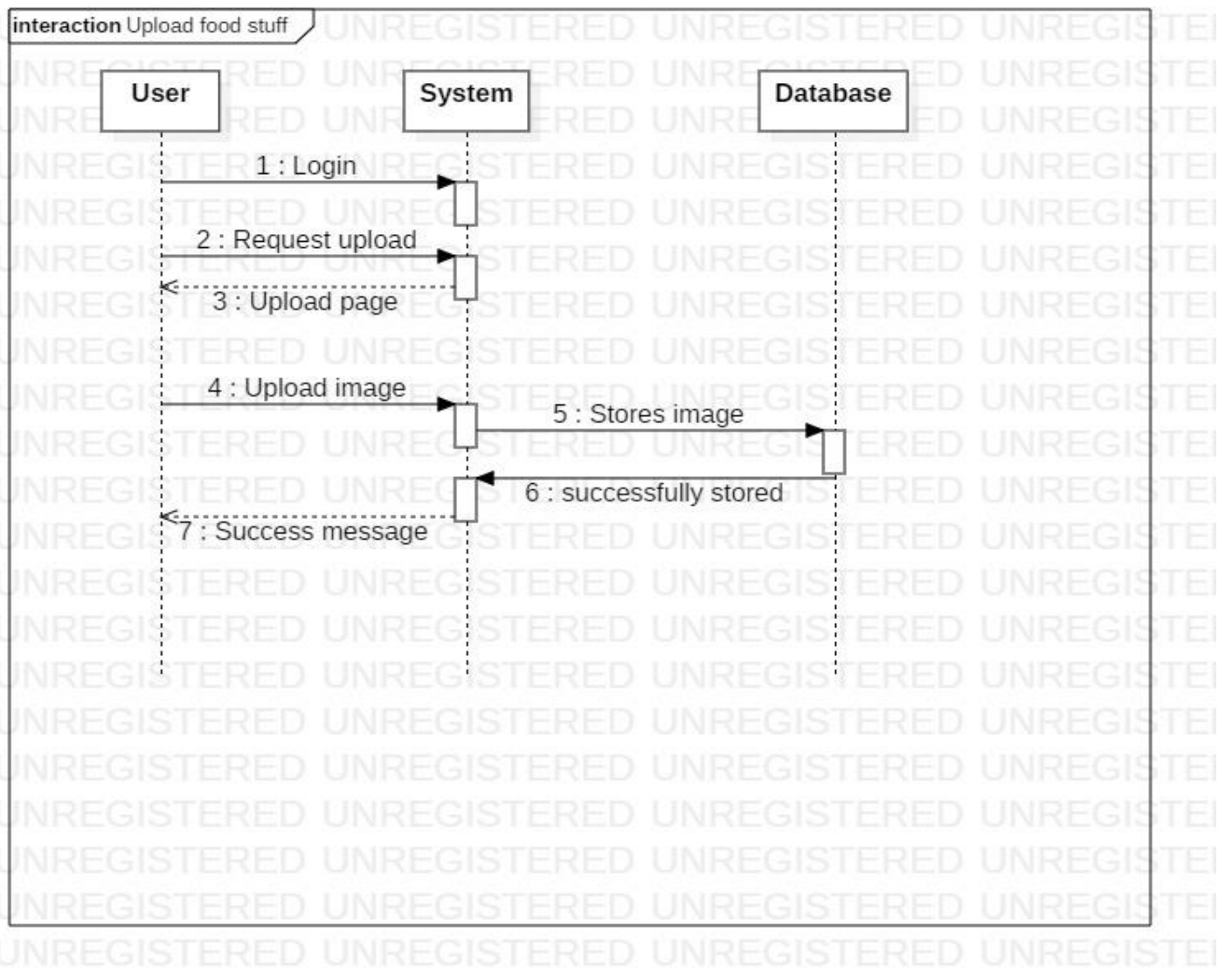
➤ Create account



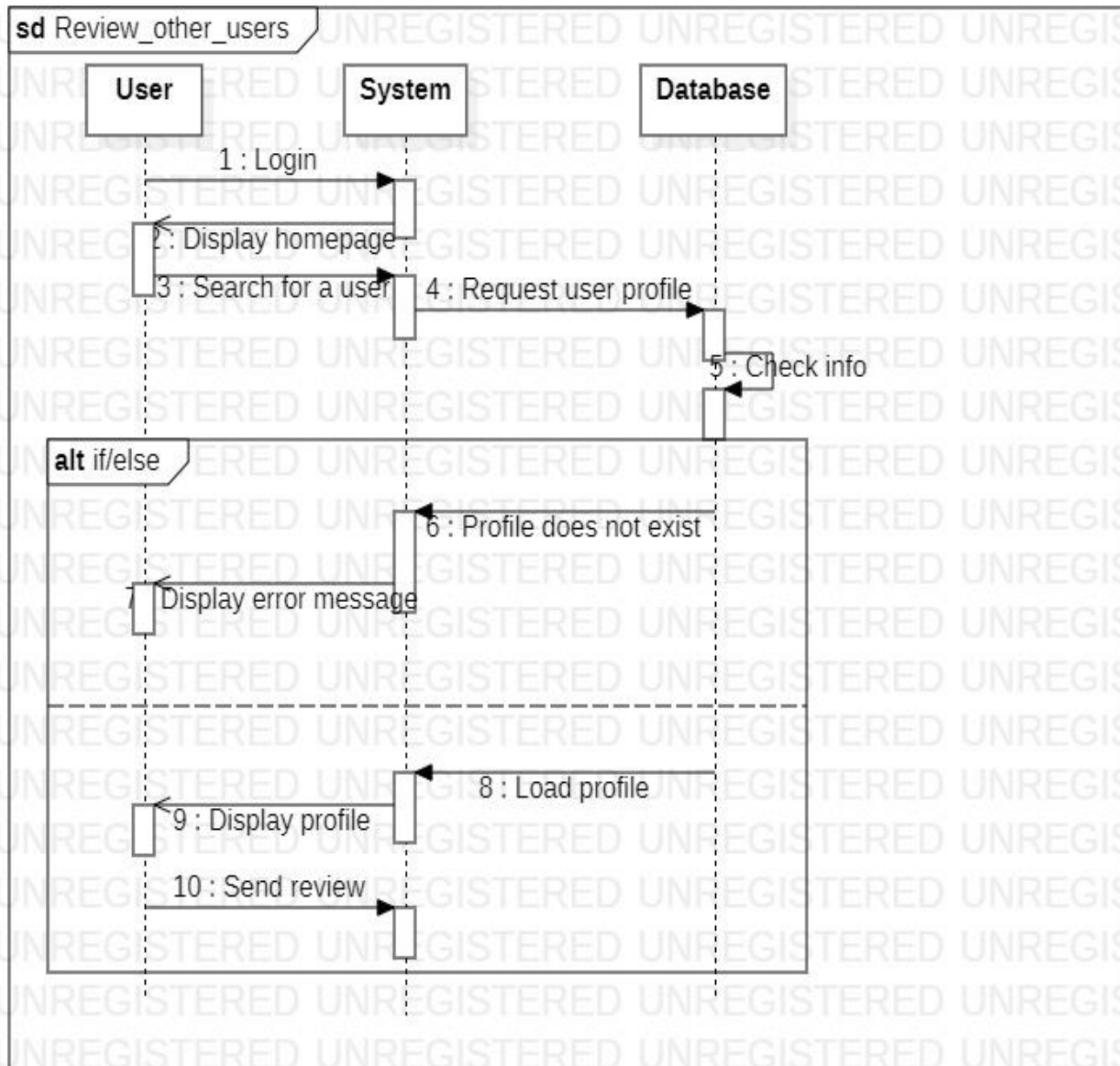
➤ Login



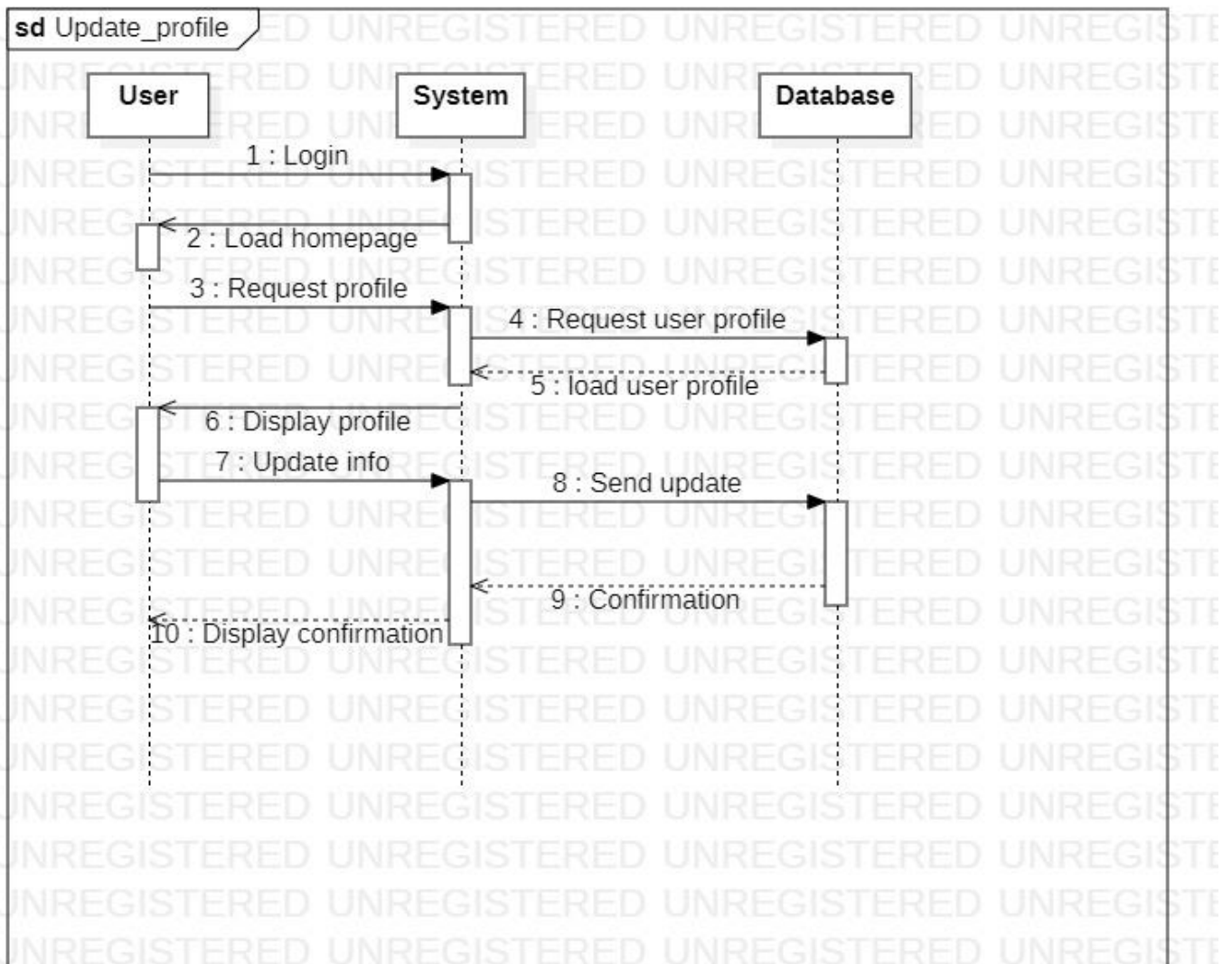
➤ Upload food Item



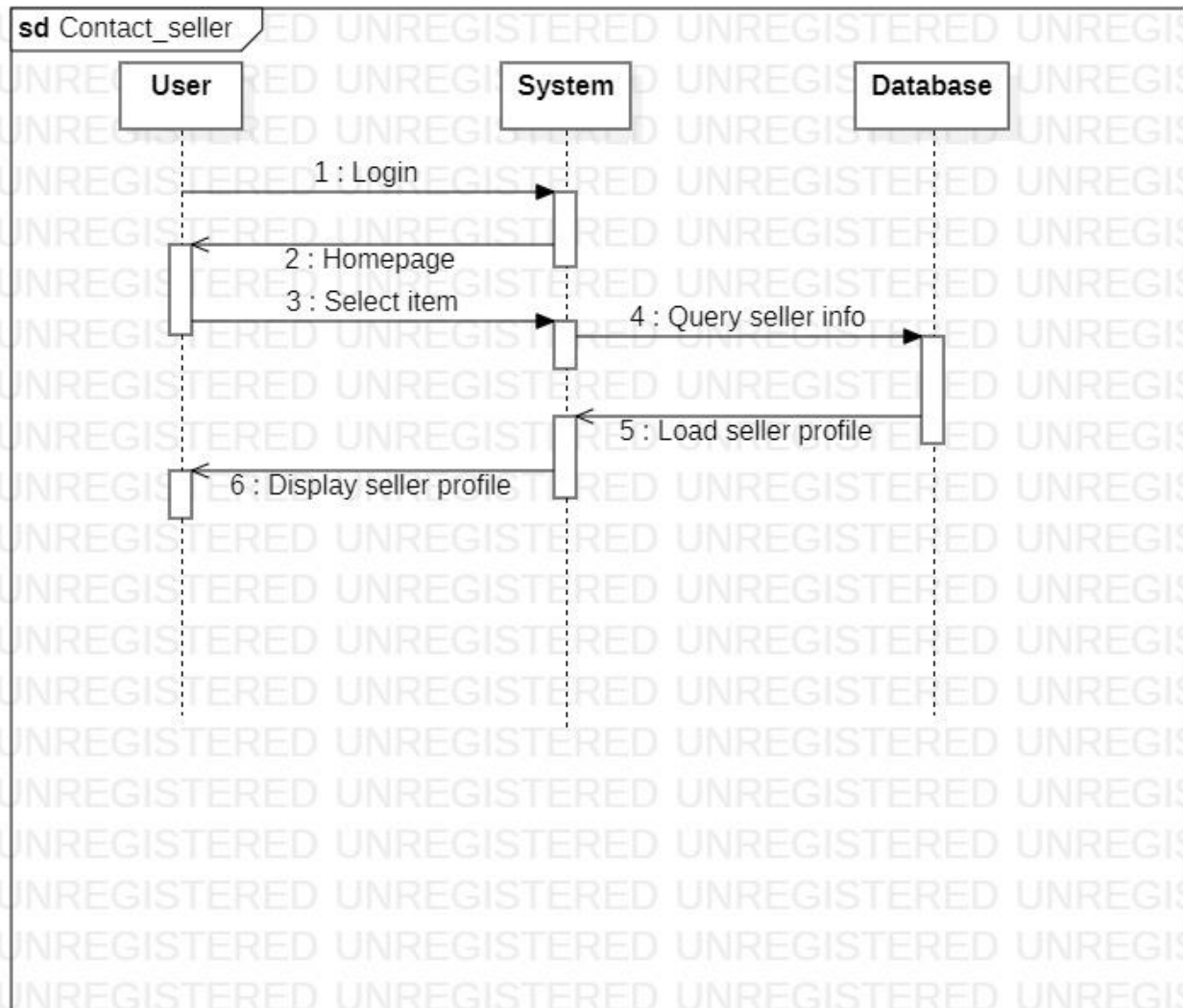
➤ Review Other Users



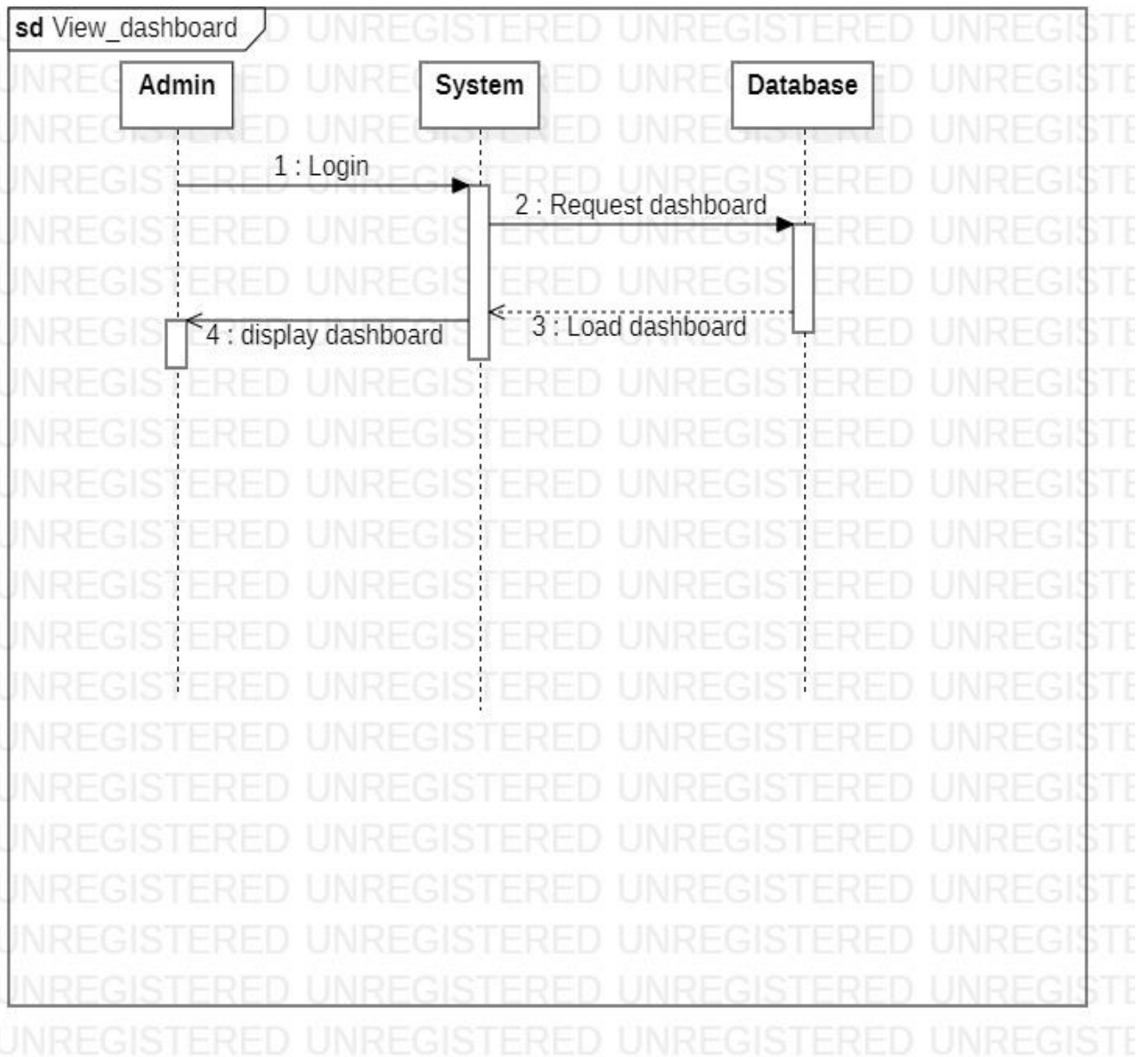
➤ Update profile



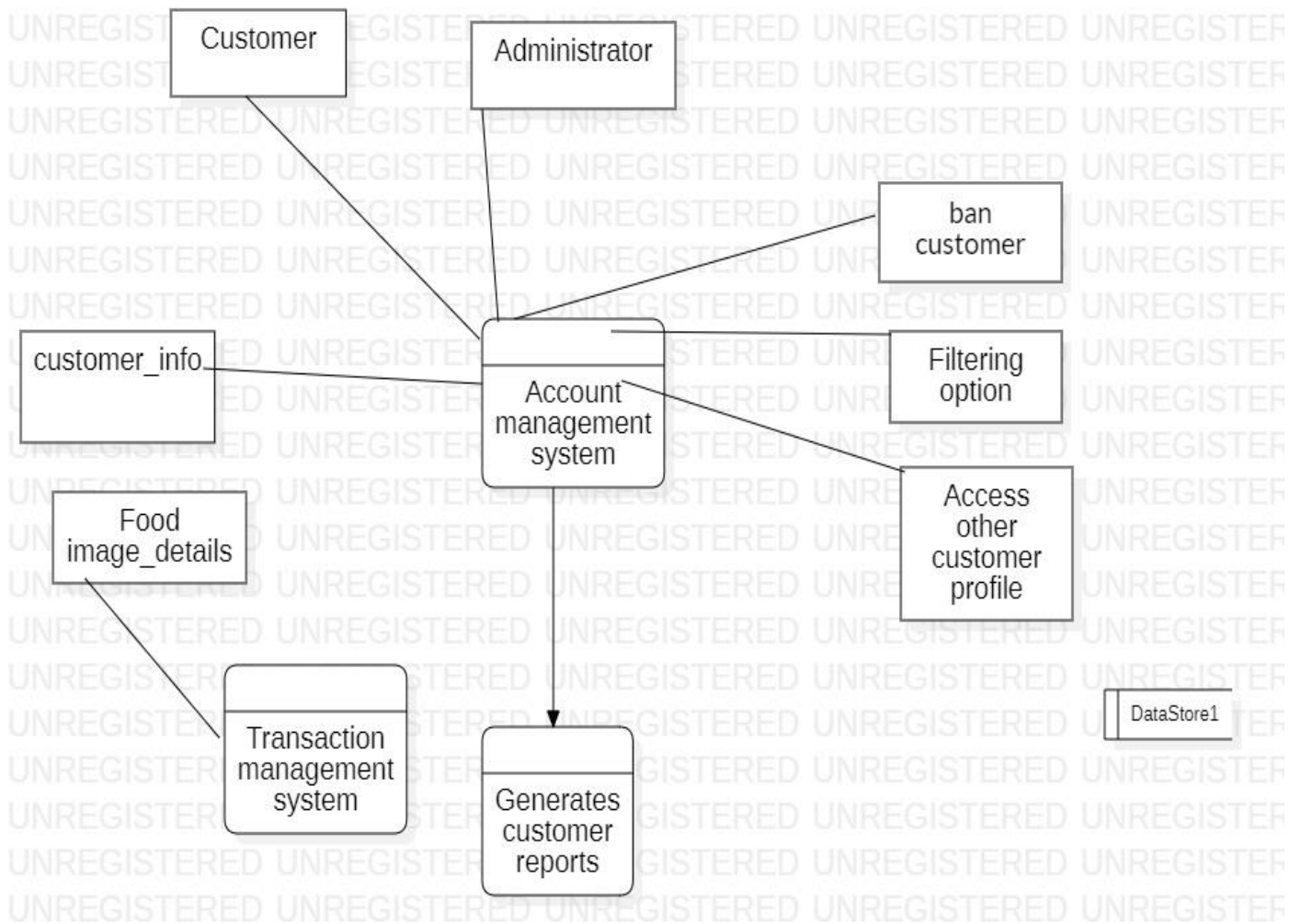
➤ Contact Seller



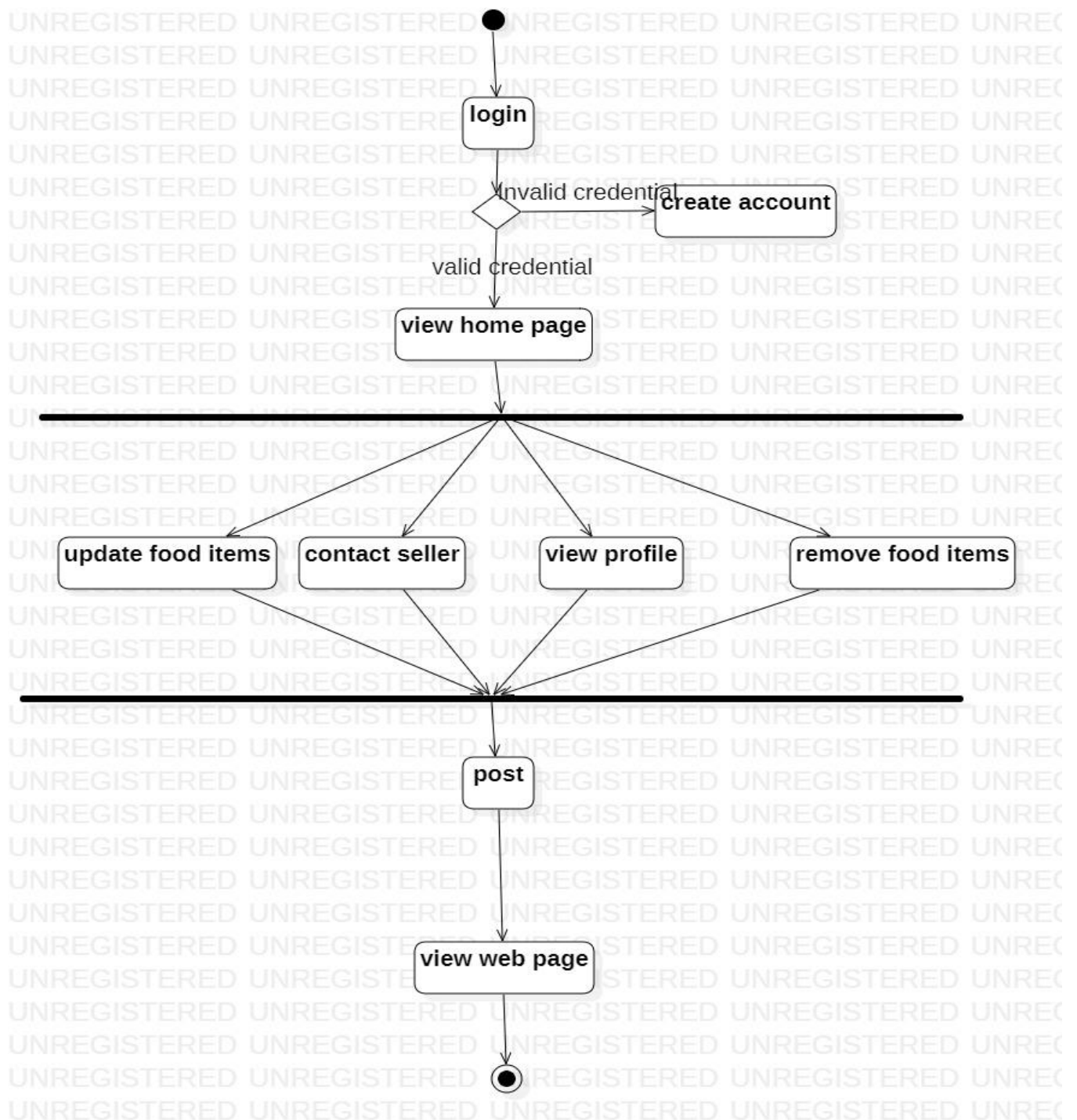
➤ View Dashboard



4. DATA FLOW DIAGRAM

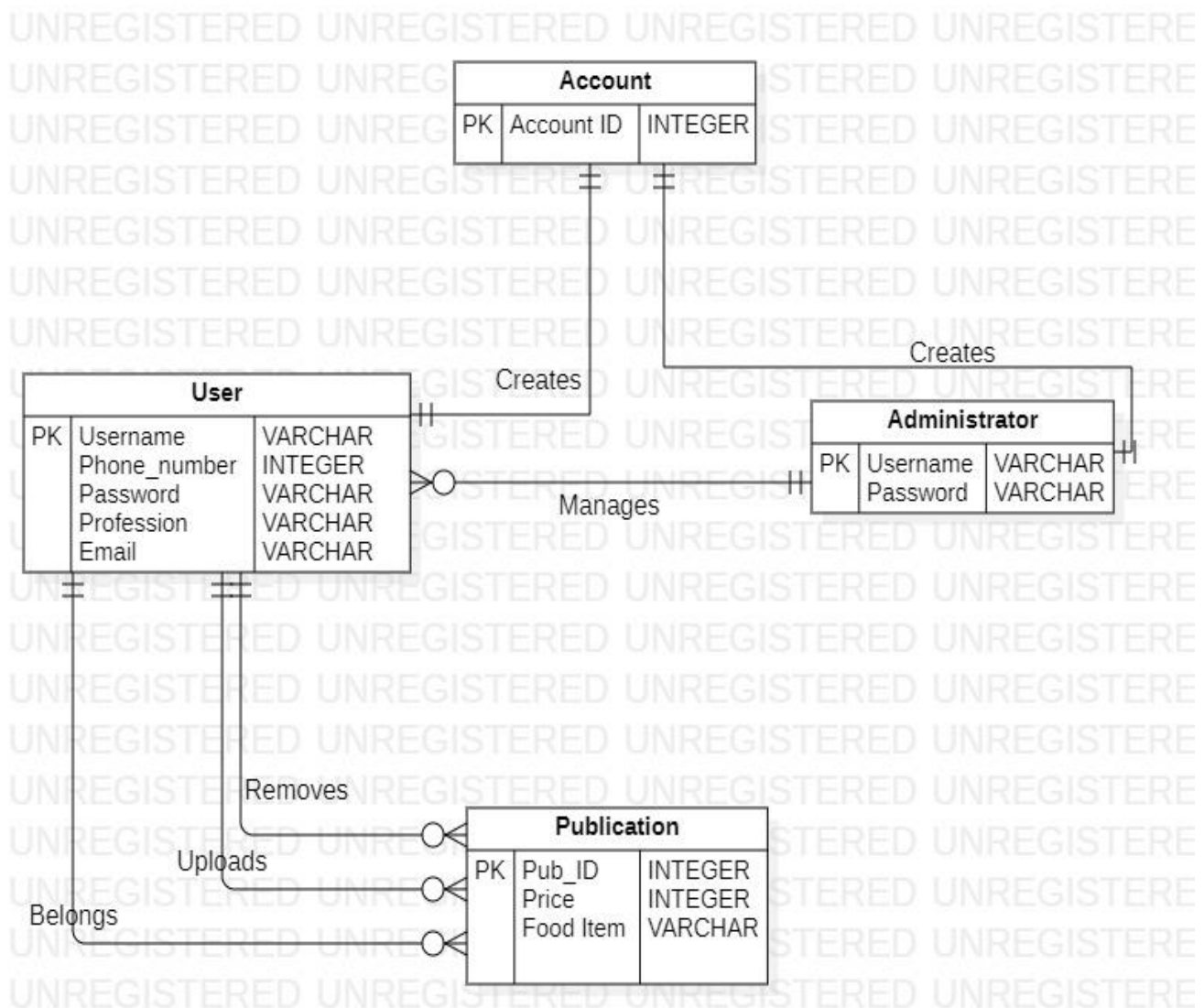


5. ACTIVITY DIAGRAM



6. ENTITY-RELATIONSHIP DIAGRAM

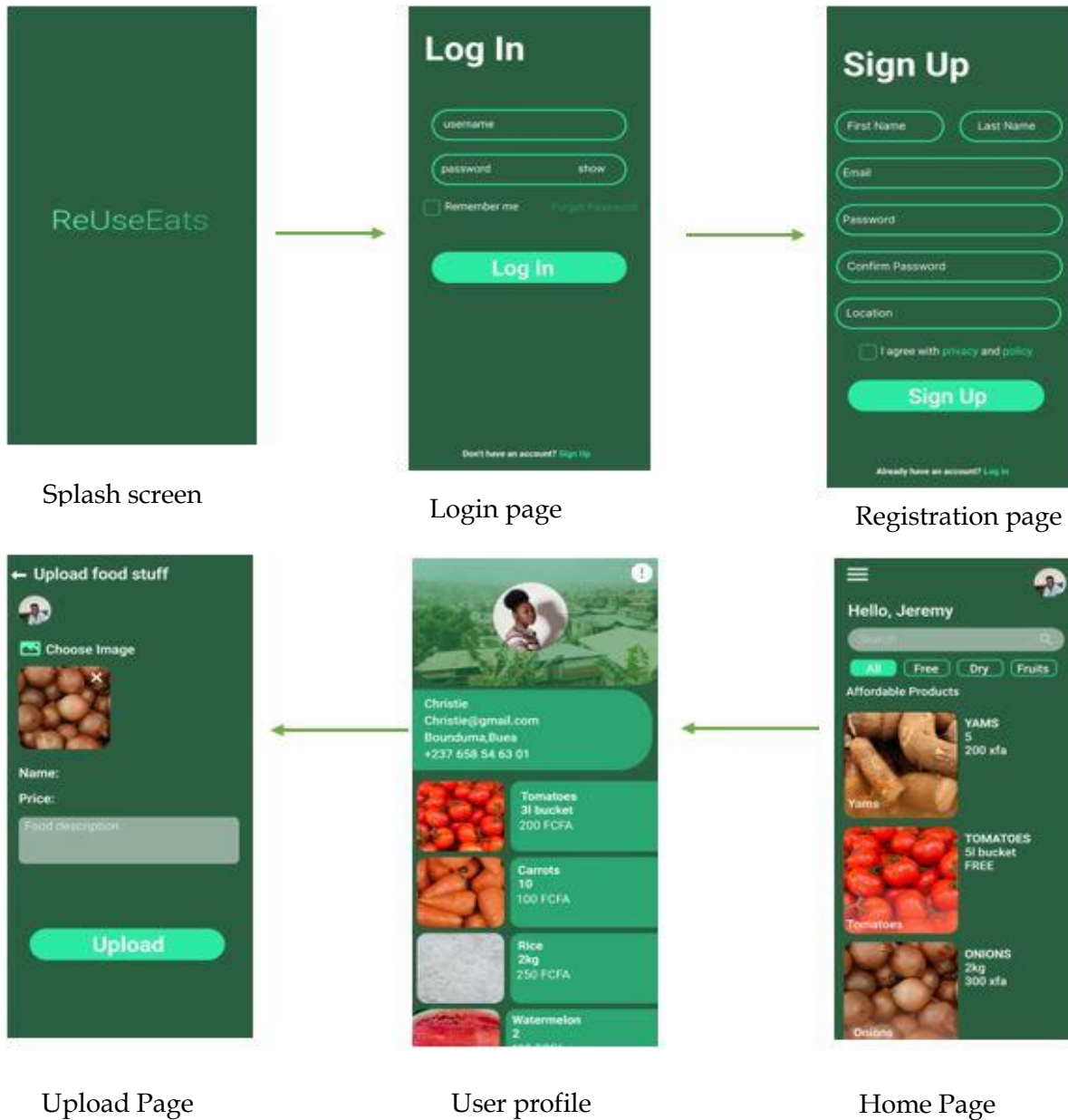
This diagram describes the relational database structure.



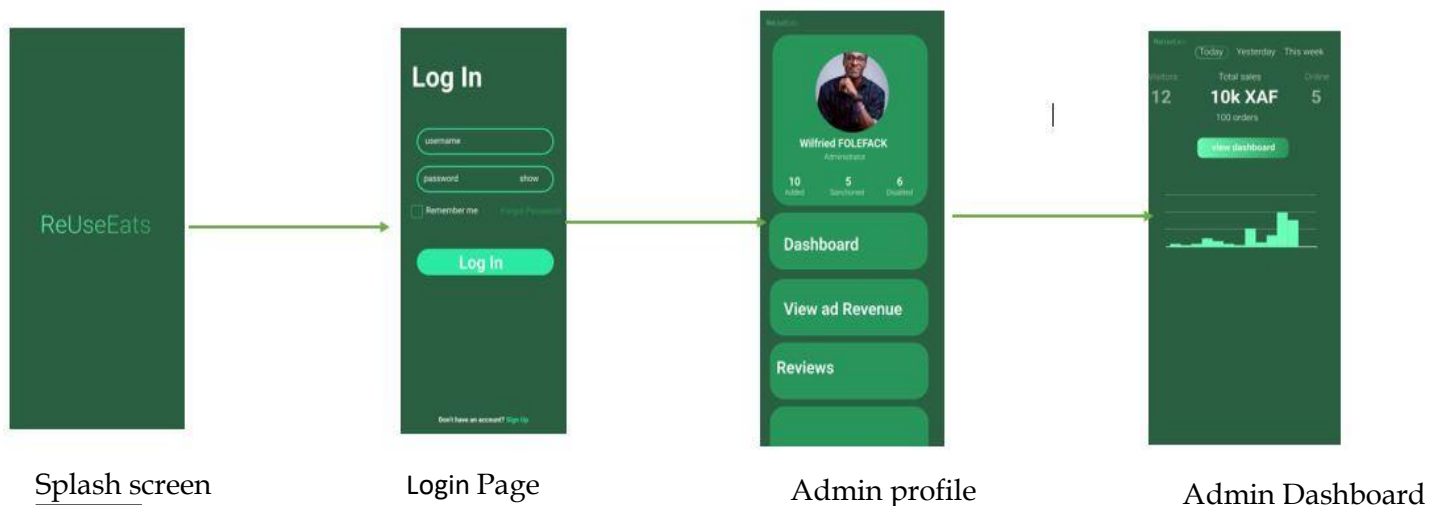
II-B USER INTERFACE DESIGN

Figma was used to design all the interfaces.

Below is the architecture of our app.



User's architecture



Administrator's architecture