

# Food Wastage Management

Artean Denisa Ioana

Universitatea de Vest din Timișoara  
Facultatea de Matematică și informatică  
Informatică Română  
Anul II

Proiect Individual  
Marinel Iordan

# Summary

<b>1</b>	<b>Abstract</b>	<b>1</b>
<b>2</b>	<b>Introducere</b>	<b>1</b>
<b>3</b>	<b>Potențiali utilizatori</b>	<b>1</b>
<b>4</b>	<b>Aplicații similare</b>	<b>1</b>
<b>5</b>	<b>Funcționalitate</b>	<b>2</b>
<b>6</b>	<b>Manualul utilizatorului</b>	<b>2</b>
<b>7</b>	<b>Documentație tehnică</b>	<b>6</b>
7.1	Codul aplicației [7]	6
7.2	Explicarea claselor și a metodelor	6
7.3	Baza de date	7
<b>8</b>	<b>Concluzii</b>	<b>9</b>
<b>9</b>	<b>Direcții viitoare</b>	<b>9</b>

## 1 Abstract

Acest document conține documentația aplicației Food Wastage Management, o aplicație mobile pentru telefoanele cu Android. Aplicația a fost realizată în Android Studio, folosind limbajul Java și are ca scop gestionarea mâncării rămase după un eveniment.

## 2 Introducere

Scopul aplicației este acela de a ne ajuta să gestionăm mâncarea rămasă după un eveniment sau o adunare, astfel, oamenii care organizează un eveniment, vor introduce adresa evenimentului și mâncarea rămasă.

## 3 Potențiali utilizatori

Aplicația se adresează tuturor persoanelor. Aplicația permite introducerea adresei și a mâncării rămase după un eveniment, de asemenea și a unui număr de telefon, acestea fiind adăugate într-o bază de date și apoi afișate în aplicație pentru a fi văzută de toți utilizatorii.

## 4 Aplicații similare

Există deja aplicații similare [12], pentru gestionarea de mâncare rămasă.

ResQ este o aplicație destinată restaurantelor, cafenelelor, hotelurilor etc. Aceștia oferă mâncare la preț redus, care altfel ar fi fost aruncată. De asemenea, această mâncare se poate comanda și plăti de pe aplicație, iar apoi ridicată de la restaurantul respectiv.

O altă aplicație similară cu ResQ este Karma, o aplicație multi-premiată pentru cumpărarea, la un preț mai mic, a mâncării din restaurante, cafenele etc.

## 5 Funcționalitate

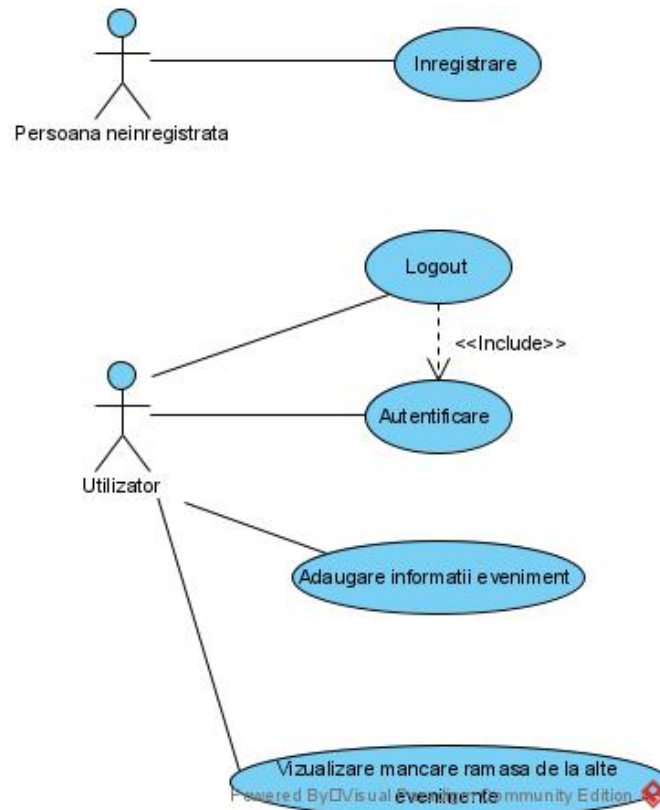
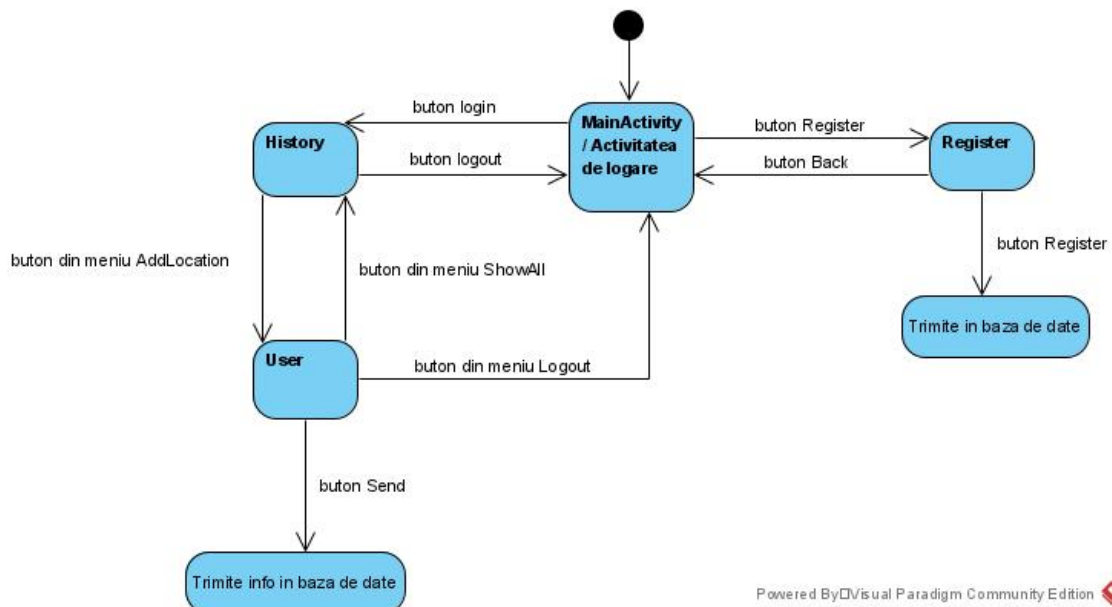
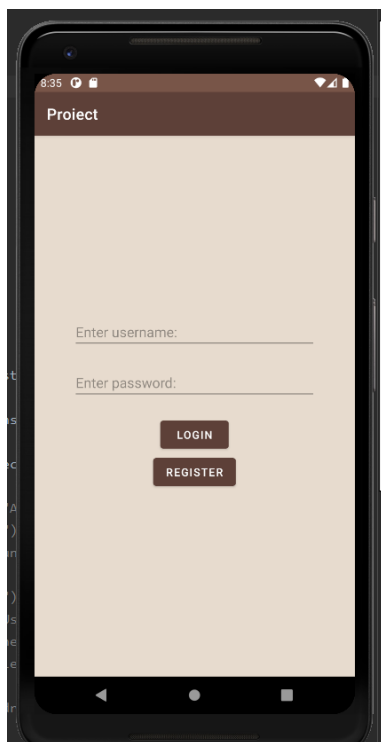


Diagrama de stări:



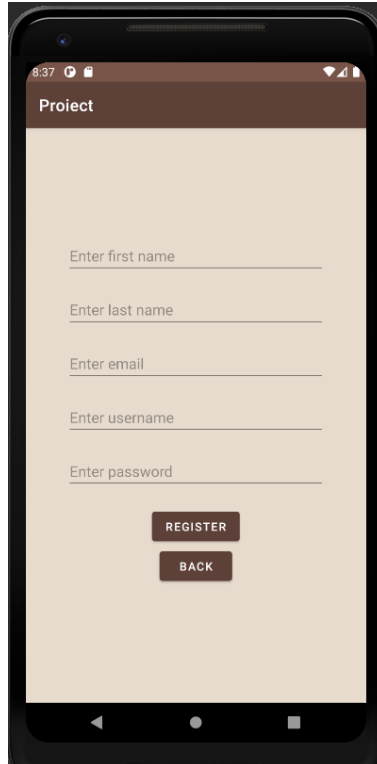
## 6 Manualul utilizatorului

La desciderea aplicației este afișată activitatea principală.



**Register:** Pentru folosirea aplicației va trebui creat un cont. Pentru crearea contului se va apăsa butonul Register, care deschide activitatea respectivă pentru crearea contului. Aici vor trebui completate următoarele câmpuri:

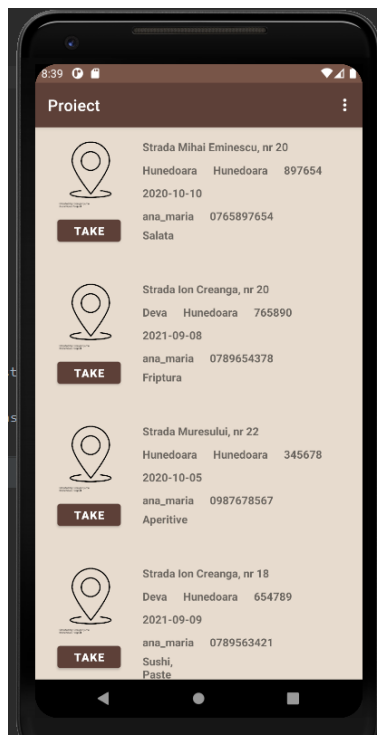
- First Name
- Last Name
- Email
- Username
- Password

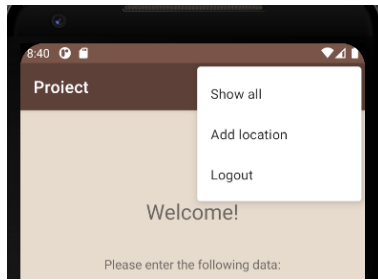


După completarea acestor câmpuri se va apăsa butonul Send, buton care înregistrează utilizatorul, și se va afișa un mesaj. Butonul Back ne va întoarce la ecranul principal.

**Login:** Dacă utilizatorul are un cont, acesta se poate loga și folosi aplicația. Dacă datele introduse corespund unui cont existent, atunci logarea se va realiza și se va deschide o activitate nouă. Dacă datele introduse nu sunt corespunzătoare unui cont existent, utilizatorul va fi informat.

În prima activitate se pot vizualiza toate locațiile introduse de către utilizatori.





În partea de sus se află un mic meniu [11] care conține:

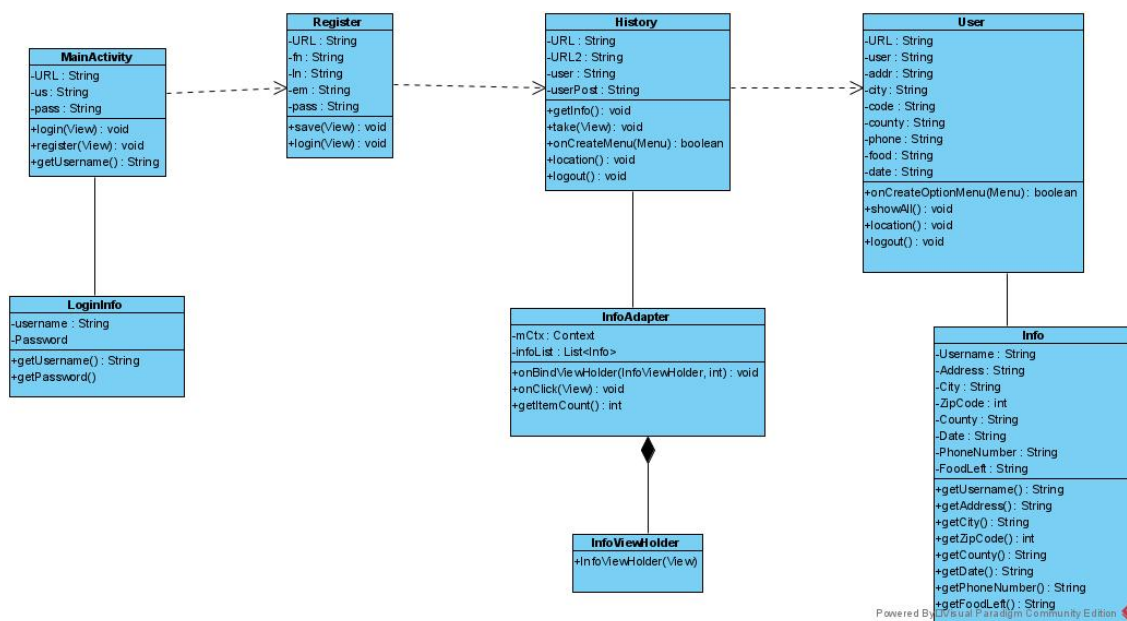
- Show all: ne duce la activitatea unde sunt afișate toate locațiile introduse de utilizatori.
- Add location: ne duce la o activitate unde se introduc informațiile despre eveniment

 A screenshot of a mobile application interface. At the top, there's a dark brown header with the word 'Project' in white. Below the header, the text 'Welcome!' is centered. Underneath, it says 'Please enter the following data:'. There are seven input fields stacked vertically: 'Address', 'City', 'Zip Code', 'County', 'Date of the Event (yyyy/mm/dd)', 'Phone Number', and 'Food Left'. At the bottom of the form, there is a dark brown button with the word 'SEND' in white.

- Logout: ne duce la activitatea principală de login.

## 7 Documentație tehnică

### 7.1 Codul aplicației [7]



Toate activitățile extind clasa AppCompatActivity și conțin metoda onCreate().

### 7.2 Explicarea claselor și a metodelor

1. Clasa MainActivity [8] este folosită pentru afișarea primei activități a aplicației. Aici se găsesc două câmpuri pentru introducerea de username și parolă și două butoane (Login și Register). Login realizează logarea în cazul în care persoana are un cont existent, iar în caz contrar afișează un mesaj. Butonul Register va trimite persoana la activitatea Register.
2. Clasa Register [8] este folosită pentru crearea contului. Aici se găsesc 5 câmpuri pentru introducerea numelui de familie, prenumelui, email, username și parolă, și două butoane Register și Back. Butonul Register trimite datele introduse în baza de date și se va afișa un mesaj pentru succes și unul pentru eroare. Butonul Back va trimite persoana la activitatea MainActivity.
3. Clasa History [3] [5] [2] [4] [6] [9] este folosită pentru afișarea locațiilor introduse de utilizatori. Pentru aceasta s-a folosit RecyclerView și CardView.
4. Clasa InfoAdapter [3] [5] [2] [4] [6] este folosită pentru RecyclerView, pentru afișarea informațiilor în cardView. În această clasă se creează și leagă ViewHolder, un itemView plasat în RecyclerView. Clasa conține clasa InfoHolder care găsește TextView-urile (findViewById()) din cardView, metoda onCreateViewHolder, care accesează layout-ul ce conține TextView-urile, metoda onBindViewHolder, care adaugă textul în TextView, și metoda getItemCount, care returnează numărul de elemente ținute de adapter.
5. Clasa Info este folosită pentru a ține date despre eveniment. Membrii clasei sunt:
  - Username
  - Address
  - City
  - ZipCode
  - County
  - Date
  - PhoneNumber

- FoodLeft

Clasa conține un constructor cu membrii enumerați mai sus și metode de get().

6. Clasa LoginInfo este folosit] pentru a ține datele utilizatorului la logare. Membrii clasei sunt:

- username
- Password

Clasa conține un constructor cu membrii enumerați mai sus și metode de get().

### 7.3 Baza de date

Ca bază de date de folosește MySQL, phpMyAdmin, aceasta fiind o bază de date globală.

În baza de date se află două tabele, USERS și INFO. Tabelul USERS conține datele introduse de fiecare utilizator la crearea unui cont, iar tabelul INFO conține datele fiecărei locații introduse de către utilizator.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/> 1	<b>Id</b> 🔑	int(11)			No	None		AUTO_INCREMENT
<input type="checkbox"/> 2	<b>FirstName</b>	text	utf8mb4_general_ci		No	None		
<input type="checkbox"/> 3	<b>LastName</b>	text	utf8mb4_general_ci		No	None		
<input type="checkbox"/> 4	<b>Email</b> 📧	varchar(300)	utf8mb4_general_ci		No	None		
<input type="checkbox"/> 5	<b>Username</b> 📧	varchar(100)	utf8mb4_general_ci		No	None		
<input type="checkbox"/> 6	<b>Password</b>	text	utf8mb4_general_ci		No	None		

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/> 1	<b>Id</b> 🔑	int(11)			No	None		AUTO_INCREMENT
<input type="checkbox"/> 2	<b>Username</b>	varchar(200)	utf8mb4_general_ci		No	None		
<input type="checkbox"/> 3	<b>Address</b>	text	utf8mb4_general_ci		No	None		
<input type="checkbox"/> 4	<b>City</b>	text	utf8mb4_general_ci		No	None		
<input type="checkbox"/> 5	<b>ZipCode</b>	int(11)			No	None		
<input type="checkbox"/> 6	<b>County</b>	text	utf8mb4_general_ci		No	None		
<input type="checkbox"/> 7	<b>Date</b>	date			No	None		
<input type="checkbox"/> 8	<b>PhoneNumber</b>	text	utf8mb4_general_ci		No	None		
<input type="checkbox"/> 9	<b>FoodLeft</b>	text	utf8mb4_general_ci		No	None		

Pentru conectarea la baza de date și introducerea datelor în tabele s-a folosit php [8].

```

connect.php x login.php x validate.php x register.php x info.php x getInfo.php x take.ph
1  <?php
2  $server = "localhost";
3  $username = "root";
4  $password = "";
5  $database = "proiect_pdm";
6  $connect = new mysqli($server, $username, $password, $database);
7  if($connect->connect_error){
8      die("Connection failed: ". $connect->connect_error);
9  }
10 >

```



```

1 <?php
2 function validate($data) {
3     $data = trim($data);
4     $data = stripslashes($data);
5     $data = htmlspecialchars($data);
6     return $data;
7 }
8 ?>

```

Se validează datele: se șterg spațiile din ambele părți ale șirului, se șterg backslashes și se convertesc unele caractere predefinite în entități HTML.

```

1 <?php
2 if(isset($_POST['FirstName']) && isset($_POST['LastName']) && isset($_POST['Email']) && isset($_POST['Username']) && isset($_POST['Password'])) {
3     require_once "connect.php";
4     require_once "validate.php";
5     $FirstName = validate($_POST['FirstName']);
6     $LastName = validate($_POST['LastName']);
7     $Email = validate($_POST['Email']);
8     $Username = validate($_POST['Username']);
9     $Password = validate($_POST['Password']);
10    $sql = "insert into users values('', '$FirstName', '$LastName', '$Email', '$Username', '". md5($Password) . "')";
11    if(!$connect->query($sql)) {
12        echo "failure";
13    } else {
14        echo "success";
15    }
16 }
17 ?>

```

```

1 <?php
2 if(isset($_POST['Username']) && isset($_POST['Password'])) {
3     require_once "connect.php";
4     require_once "validate.php";
5     $Username = validate($_POST['Username']);
6     $Password = validate($_POST['Password']);
7     $sql = "select * from users where Username='$Username' and Password='". md5($Password) . "'";
8     $result = $connect->query($sql);
9     if($result->num_rows > 0) {
10        echo "success";
11    } else {
12        echo "failure";
13    }
14 }
15 ?>

```

```

1 <?php
2 if(isset($_POST['Username']) && isset($_POST['Address']) && isset($_POST['City']) && isset($_POST['ZipCode']) && isset($_POST['County']) && isset($_POST['Date'])
3 && isset($_POST['PhoneNumber']) && isset($_POST['FoodLeft'])) {
4     require_once "connect.php";
5     require_once "validate.php";
6     $Username = validate($_POST['Username']);
7     $Address = validate($_POST['Address']);
8     $City = validate($_POST['City']);
9     $ZipCode = validate($_POST['ZipCode']);
10    $County = validate($_POST['County']);
11    $Date = validate($_POST['Date']);
12    $PhoneNumber = validate($_POST['PhoneNumber']);
13    $FoodLeft = validate($_POST['FoodLeft']);
14    $sql = "insert into info values('', '$Username', '$Address', '$City', '$ZipCode', '$County', '$Date', '$PhoneNumber', '$FoodLeft')";
15    if(!$connect->query($sql)) {
16        echo "failure";
17    } else {
18        echo "success";
19    }
20 }
21 ?>

```



```

1  <?php
2  include('connect.php');
3  $stmt = $connect->prepare("Select Username, Address, City, ZipCode, County, Date, PhoneNumber, FoodLeft from info order by Id DESC");
4  $stmt->execute();
5  $stmt->bind_result($Username, $Address, $City, $ZipCode, $County, $Date, $PhoneNumber, $FoodLeft);
6
7  $info = array();
8
9
10 while($stmt->fetch()) {
11
12     $temp = array();
13     $temp['Username'] = $Username;
14     $temp['Address'] = $Address;
15     $temp['City'] = $City;
16     $temp['ZipCode'] = $ZipCode;
17     $temp['County'] = $County;
18     $temp['Date'] = $Date;
19     $temp['PhoneNumber'] = $PhoneNumber;
20     $temp['FoodLeft'] = $FoodLeft;
21     array_push($info, $temp);
22 }
23
24 echo json_encode($info);
25
26 ?>

```

## 8 Concluzii

Aplicația creată este folosită pentru a gestiona mâncarea rămasă de la evenimente. Ea a fost dezvoltată în Android Studio, folosind limbajul Java, și este o aplicație destinată tuturor oamenilor.

## 9 Direcții viitoare

În momentul de față, doar o parte din aplicație este terminată, nefiind încă finalizată. Pe viitor aș vrea să mai adaug următoarele facilități

- Crearea unui buton Take, care trimite în baza de date informații despre userul care a rezervat/luat mâncarea de la evenimentul respectiv, și informațiile evenimentului, iar după apăsarea butonului, acesta se va dezactiva.
- Crearea unei hărți unde se vor pune pin-uri cu locațiile introduse
- Un istoric unde se vor putea vizualiza toate locațiile adăugate de userul respectiv
- Crearea unei notificări atunci când user introduce o locație nouă.

## References

- [1] Developers Android  
<https://developer.android.com/>
- [2] RecyclerView and CardView - Simplified Coding  
<https://www.simplifiedcoding.net/android-recyclerview-cardview-tutorial/>
- [3] Retrieve data mysql - Simplified Coding  
<https://www.simplifiedcoding.net/retrieve-data-mysql-database-android/>
- [4] RecyclerView and Volley - Fetching Data from MySQL Database – Simplified Coding  
<https://www.youtube.com/watch?v=Yw7Lx9wqyGs>
- [5] Android, PHP And Mysql Tutorial - Retrieving Data From Mysql Database Using Volley Library – Tech With Walter  
<https://www.youtube.com/watch?v=C4DwG2jXS4t> = 131s
- [6] GeeksForGeeks - Volley library  
<https://www.geeksforgeeks.org/volley-library-in-android/>
- [7] Codul aplicației  
<https://github.com/DenisaArtean/FoodWastageManagement/>
- [8] Android Login and Register with PHP MySQL and Volley Library - Sandip Bhattacharya  
<https://www.youtube.com/watch?v=GKyEJmCoK5s>
- [9] Simple RecyclerView with CardView Tutorials in Android Studio - SmallAcademy  
<https://www.youtube.com/watch?v=PEofA7Z8eQ>
- [10] The Noun Project for icon  
<https://thenounproject.com/search/?q=locationi=304559>
- [11] How to Implement menu in Android Activity with Options Menu - Lemubit Academy  
<https://www.youtube.com/watch?v=HmedyRcKOzc>
- [12] Similar apps  
<https://apiumhub.com/tech-blog-barcelona/top-food-waste-apps/>