

# Carbon Footprint Tracker

<b>EX1. Problem Statement</b>	<b>4</b>
1. Project Title	4
2. Project Definition	4
3. Expected Duration	4
4. Software Requirements	4
5. Hardware Requirements	4
6. Objective	4
7. Scope	5
8. Significance of the Problem	5
9. Advantages of the Project	5
10. Prerequisites	5
<b>EX2. Software Requirements Specification</b>	<b>6</b>
1. Introduction	6
1.1 Document Purpose	6
1.2 Product Scope	6
1.3 Abstract	6
1.4 Intended Audience and Document Overview	6
1.5 Definitions, Acronyms and Abbreviations	7
1.6 References and Acknowledgements	7
2. Overall Description	7
2.1 Product Perspective	7
2.2 Product Functionality	8
2.3 Users and Characteristics	8
2.4 User interface	8
2.5 Operating Environment	9
2.6 Hardware requirements	9
2.7 Design and Implementation constraints	9
2.8 System Flow Diagram	9
2.8 User Documentation	11
2.9 Assumptions and Dependencies	11
3. Specific Requirements	11
3.1 Functional Requirements	11
3.2 Sub Functionalities	11
3.3 Behavioural Requirements	12
4. Other Nonfunctional Requirements	12

4.1	Performance Requirements	12
4.2	Safety Requirements	12
4.3	Security Requirements	12
4.4	Software Quality Attributes	12
<b>EX3.</b>	<b>Use Case Model</b>	<b>14</b>
1.	Aim	14
2.	Notations	14
3.	Identification of Actors	14
4.	Identification of Scenarios	15
5.	Relating Use Cases	15
6.	Use Case Diagrams	17
7.	Fully Dressed Use Case Description	21
8.	Documentation	22
<b>EX4.</b>	<b>Domain Model and Class Diagram</b>	<b>23</b>
1.	Aim	23
2.	UML Notations for Domain Model and Class Diagram	23
3.	Identification of Classes	23
4.	Identification of Associations	24
5.	Domain Model Diagram	27
6.	Class Diagram	28
7.	Documentation	28
<b>EX5.</b>	<b>Interaction Diagrams</b>	<b>29</b>
1.	Sequence Diagram - Main Success Scenario	29
2.	Sequence Diagram - Each Use Case	29
3.	Collaboration Diagram - Main Success Scenario.	33
<b>EX6.</b>	<b>Interaction Diagrams</b>	<b>34</b>
1.	Main Success Scenario	34
2.	Each Use Case	34
<b>EX7.</b>	<b>State Machine Diagram</b>	<b>37</b>
<b>EX8.</b>	<b>Package, Component and Deployment Diagram</b>	<b>38</b>
1.	Package Diagram	38
2.	Component Diagram	39
3.	Deployment Diagram	40
<b>EX9.</b>	<b>Implementation of UI &amp; Domain Layers</b>	<b>41</b>
<b>EX10.</b>	<b>Implementation of Technical Service Layer</b>	<b>41</b>
<b>EX11.</b>	<b>Developing Test Plan &amp; Test Cases</b>	<b>42</b>
1.	Identification of Testing Scenarios	42
2.	Tabulation of Test Cases	42
<b>EX12.</b>	<b>Applying Design Patterns</b>	<b>46</b>
1.	Factory Pattern	46
2.	Strategy Pattern	46

3. Observer Pattern	47
<b>EX13. Test Cases After Refinement</b>	<b>48</b>
1. Identification of Testing Scenarios	48
2. Tabulation of Test Cases	48
<b>EX14. Refactoring</b>	<b>52</b>

## ***EX1. Problem Statement***

### **1. Project Title**

Web-based Carbon Footprint Tracker.

## 2. Project Definition

To build a system that enables the individuals in a community to track their carbon footprint. The users will be able to understand the amount of carbon their simple everyday activities generate. This system will compute comparative statistics in terms of personal goals as well as community goals for a sustainable lifestyle. This will allow users to make changes in their way of life by adopting more environment-friendly alternatives, and pave the way to carbon neutrality by offsetting their emissions to the highest standards.

## 3. Expected Duration

4 months.

## 4. Software Requirements

- **Operating System:** Windows 10, 11
- **Front End:** HTML, CSS, JavaScript
- **Back end :** MySQL Server

## 5. Hardware Requirements

- **Processor:** 1 gigahertz (GHz) or faster processor or SoC.
- **RAM:** 1 gigabyte (GB) for 32-bit or 2 GB for 64-bit
- **Hard disk space:** 20 GB for 64-bit OS

## 6. Objective

The users can track and calculate their carbon emissions. This will allow them to assess their contribution to global warming.

## 7. Scope

The solution should give complete information about personal and community level carbon emissions.

## 8. Significance of the Problem

It is well known that a large proportion of the Indian population is either unaware, or chooses to disregard the life threatening effects certain day to day tasks pose to the environment. Basic necessities such as electricity and transportation, involve the burning of fossil fuels which in turn result in huge carbon footprints. Other activities result in smaller amounts of carbon emissions that compound over time and cause significant damage to the environment.

There exist individuals in our communities that are aware of these issues but have no concrete means to implement change on a large scale. This project aims to make individuals understand how much carbon they produce on a daily basis based on their activities. This will help them realise where they stand in comparison to the safe levels prescribed by various government agencies and push communities to adopt a cleaner, greener and more sustainable lifestyle.

## 9. Advantages of the Project

- Carbon neutrality: allows users to become carbon neutral by offsetting their emissions to the highest standards and to support local communities.
- Carbon responsible management: Companies can develop their marketing and communication strategies around the environmental credentials
- Supply Chain Reporting: provides an easy to use and cost effective solution to meet environmental reporting needs of companies since there is a push for more sustainable practices worldwide.

## 10. Prerequisites

Users should have the latest versions of ios and android that support the application. They are expected to sign up using their mobile number. This will allow them to track their history of carbon emissions from their profile.

---

## ***EX2. Software Requirements Specification***

### 1. Introduction

This section provides an overview of the entire Software Requirements Specifications (SRS) document.

## 1.1 Document Purpose

The purpose of this document is to show the software requirements of the Carbon Footprint Tracker. The functionality and scope of this application is described in this SRS document.

## 1.2 Product Scope

The Carbon Footprint Tracker aims at helping users track their carbon footprint in a holistic and dynamic fashion.

The major benefits of the application are:

- It tracks the amount of carbon emitted by a user under various categories like travel, electricity and garbage production based on the data that they input
- It allows users to set goals and improve their carbon footprint over a period of time.
- It provides statistics that compare user emissions with that of their community to encourage the adoption of more sustainable means of living.

## 1.3 Abstract

To build a system that enables the individuals in a community to track their carbon footprint. The users will be able to understand the amount of carbon their simple everyday activities generate. This system will compute comparative statistics in terms of personal goals as well as community goals for a sustainable lifestyle. This will allow users to make changes in their way of life by adopting more environment-friendly alternatives, and pave the way to carbon neutrality by offsetting their emissions to the highest standards.

## 1.4 Intended Audience and Document Overview

The SRS document is intended for developers who will be implementing the system. Environmentalists and government agencies may benefit from the document to understand the working of the application before using it on a larger scale.

## 1.5 Definitions, Acronyms and Abbreviations

Carbon neutrality	Carbon neutrality means having a balance between emitting carbon and absorbing carbon from the atmosphere in carbon sinks
-------------------	---

Carbon footprint	A carbon footprint is the total amount of greenhouse gases (including carbon dioxide and methane) that are generated by our actions.
Carbon responsible management	Carbon management is an organised approach to gain the strategic advantages of CO2 emissions reductions.
Supply Chain management	Supply chain management is the practice of coordinating the various activities necessary to produce and deliver goods and services to a business's customers.

## 1.6 References and Acknowledgements

- Object-oriented and Classical Software Engineering by Stephen R. Schach
- Software Engineering: A Practitioner's Approach by Roger F. Pressman

## 2. Overall Description

This section highlights the prerequisites and the general factors that affect the products and its requirements. This information will aid in the better understanding of the specific requirements mentioned in the next section.

### 2.1 Product Perspective

Many human activities have caused widespread damage to the environment. These activities have impacts that include, but are not limited to, pollution, deforestation, enlargement of the ozone hole and so on. Most of these problems have a common root cause – carbon emissions. By reducing the carbon footprint of each individual, the harm to the environment can be significantly decreased. Hence, there is a pressing need for availing clean and green technology at the earliest.

The proposed application will provide the necessary means to track carbon emissions from daily activities and aid in achieving individual and community goals in sustainable development.

## 2.2 Product Functionality

This section provides the graspable functional overview of the end product. The product is expected to be providing the following functionalities

1. Register to the Carbon Footprint Tracker.
2. Login to the application.
3. Enter daily, weekly and monthly emission goals.
4. Enter daily actions under each category.
5. System computes carbon emissions for the day.
6. Comparative statistics over a period of time.
7. Progress with respect to personal goals.
8. Progress with respect to the community.
9. Special dashboard for environmentalists with community data statistics.
10. Special dashboard for government agencies.

## 2.3 Users and Characteristics

- **Individual users:** Members of a community who track their daily levels of carbon emission
- **Private organisations:** This group of users is composed of schools, colleges and corporations that encourage individuals to track their carbon footprint for a social cause.
- **Environmentalists:** They have access to the overall statistics and trends in the carbon footprint of individual users and communities.
- **Government organisations:** These are organisations that track the data of particular areas and constituent wards

## 2.4 User interface

The carbon footprint tracker must have an appealing and easy to navigate user interface. User interactions should be interruptible and re-doable. The technical internal details should be hidden from the end users. Factors like data consistency and reliability must be considered. Response time of each page contributes to the overall performance of the web app, hence the web application must have less response time.

## 2.5 Operating Environment

- Operating system: Latest versions of windows and macOS
- Language: Java, MySQL, HTML



- Software: - JDK/JRE- Eclipse IDE

## 2.6 Hardware requirements

**Processor:** 1 gigahertz (GHz) or faster processor or SoC.

**RAM:** 1 gigabyte (GB) for 32-bit or 2 GB for 64-bit

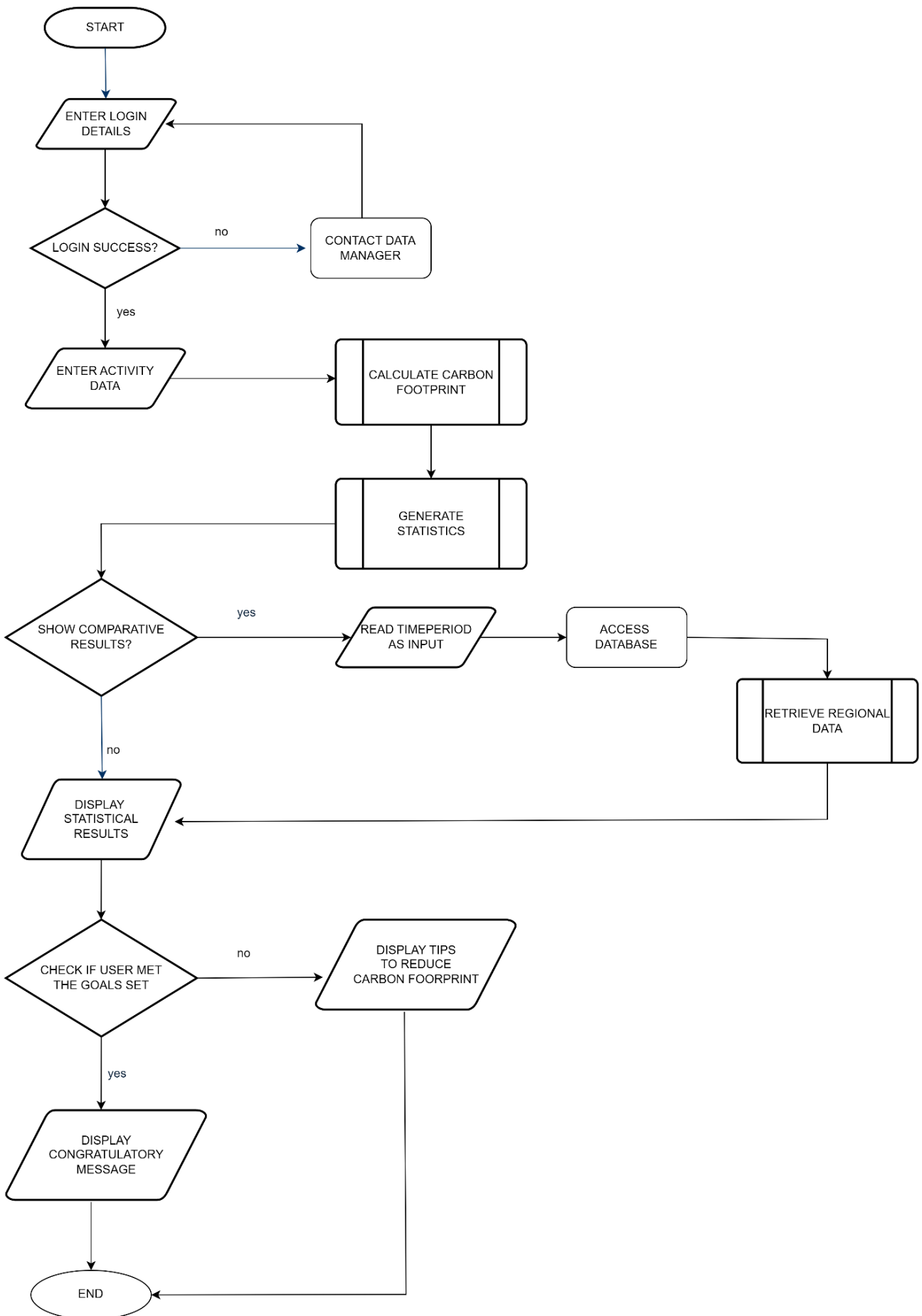
**Hard disk space:** 20 GB for 64-bit OS

## 2.7 Design and Implementation constraints

The carbon footprint tracker must be designed such that it is highly efficient and reliable. The data tracked and stored in the database should be consistent. The response time of the app must be less. The Internet is a necessity for using all the functionalities provided by the developer.

The system is developed in the Java Language and is supposed to work only on any operating system that has JDK or JRE installed on it.

## 2.8 System Flow Diagram



## **2.8 User Documentation**

It will provide specific guidelines to a user for using the Carbon Footprint Tracker application. The documentation will contain requirement-specific instructions for each category of users.

## **2.9 Assumptions and Dependencies**

It is assumed that the information entered by the users are accurate and not exaggerated. There are no means by which the system can confirm the validity of the data entered. This venture is built on the belief that the users are concerned about sustainability.

# **3. Specific Requirements**

## **3.1 Functional Requirements**

- Allowing the users in a community to sign up using their phone number.
- Collecting data for each user from records of the activities entered by them.
- Estimating the carbon footprint for every individual using standards specified by the government.
- Enabling the user to view category wise carbon emissions.
- Providing statistics for tracking and comparison of carbon emissions over time.
- Displaying local and global statistics in a visually appealing and engaging manner.
- Promoting community awareness through notifications and suggestions based on user activity history.

## **3.2 Sub Functionalities**

- Collection of data  
Once the individual enters his/her activity data in numerical format, the system collects it and stores it in the database. The data is also used to calculate and store the carbon footprint of the individual.
- Retrieval of data  
The data and activity history of every individual, or a community of individuals, can be retrieved from the database by the appropriate users (environmentalists, organisation managers etc) who can then monitor the corresponding progress.
- Management of data  
Data maintenance and ensuring its consistency is done by the data manager, who is responsible for managing the system database and optimising it to produce timely results. The system offers the data manager appropriate rights to do so.

### **3.3 Behavioural Requirements**

- The LoginError use case enables a member to contact the required personnel automatically, in cases where the server might be busy or there are issues with respect to the system-database connectivity.
- When the user logs in his/her activity records, the system displays the corresponding carbon emissions generated, in a user-readable and visually appealing format.
- When the user wishes to view his carbon footprint over a period of time, the system responds with a summary of statistics over the requested period.

## **4. Other Nonfunctional Requirements**

### **4.1 Performance Requirements**

- Increase customer satisfaction.
- Reduce overall costs.
- The same resources can be used many times by a number of users.
- Presence of data backup. In case data is deleted, it can be recovered.
- The load time for user interface screens shall take no longer than two seconds.
- The log in information shall be verified within five seconds.
- Queries shall return results within five seconds

### **4.2 Safety Requirements**

- Backup, recovery and business continuity. The application ensures adequate backup of data as may be required by its operations.
- Both data and software should be backed up periodically.
- An off-site backup is necessary for recovery from major failure to ensure business continuity.

### **4.3 Security Requirements**

- The user information is only accessible by the user and the customer service personnel. Access to this information is protected by a user login screen that requires a valid user identification.
- If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying the operations of committed transactions from the backed up log, up to the time of failure.

### **4.4 Software Quality Attributes**

- **Standards Compliance:** There shall be consistency in variable names within the system. The graphical user interface shall have a consistent look and feel.
- **Availability:** The system shall be available 24\*7.
- **Maintainability:** The system will be developed in well known platforms that are easy to maintain.
- **Portability:** The system shall be designed to run on various popular environments.

The supreme goal of the application is focused on increasing responsible consumption and production of resources and products at the earliest possible juncture.

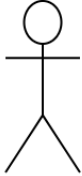
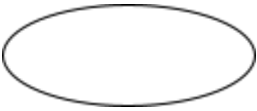



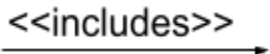
---

## **EX3. Use Case Model**

### **1. Aim**

Creation of an UML Use case model for the web-based Carbon Footprint Tracker.

### **2. Notations**

Diagram	What it denotes
	Actor
	Use Case
	Connection between Actor and Use Case
	Boundary of system
	Extend relationship between Use Cases (one UC calls Another under certain condition; think of if-then decision points)
	Include relationship between Use Cases (one UC must call another; e.g., Login UC includes User Authentication UC)

### **3. Identification of Actors**

- Individual users

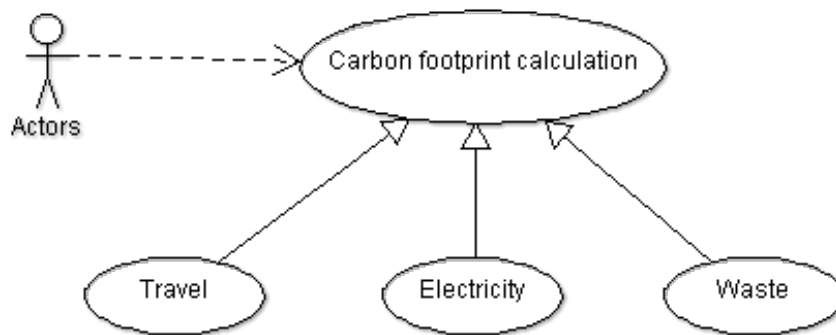
- They are individuals of a community that enter their activities to track their carbon footprint.
- Environmentalists
  - They are individuals who are associated with NGOs and have access to the overall community trends of their respective areas.
- Private organisations
  - They are commercial organisations, schools and colleges that wish to track their carbon emissions as an entity of individuals.
- Government organisations
  - They are organisations affiliated with the government that keep track of their constituent wards and their carbon emissions.
- Manager
  - Manages the database manually and has access to all accounts in case of issues.

## 4. Identification of Scenarios

- Success scenario 1: Users enter activity data and are able to view their accurate carbon footprint
- Success scenario 2: Environmentalists and government organisations are able to view the trends of their community accurately
- Failure scenario: Error occurs while logging in or with carbon footprint calculation
- Sub functions
  - Collection of data
  - Retrieval of data
  - Management of data

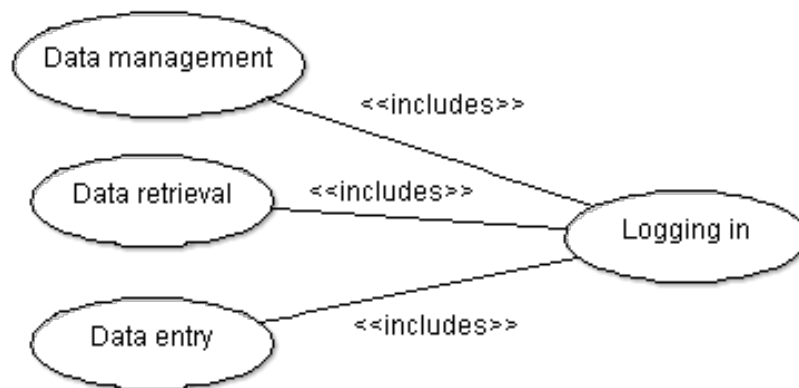
## 5. Relating Use Cases

- **Generalisation:**
  - Calculation of carbon footprint involves the calculation of all sub categories like travel, waste generation, electricity consumption, water consumption etc.
  - Hence, “carbon footprint calculation” is a generalisation of all sub categories.



- **Includes:**

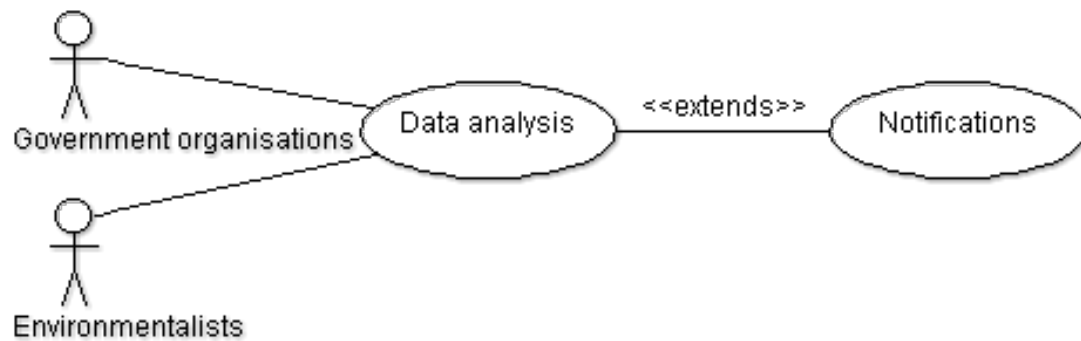
- The user must be logged in in order to enter their activity data.
- Any user functionality in the tracking system can be performed only after logging in.
- Hence, any functionality “includes” the act of logging in.



- **Extends:**

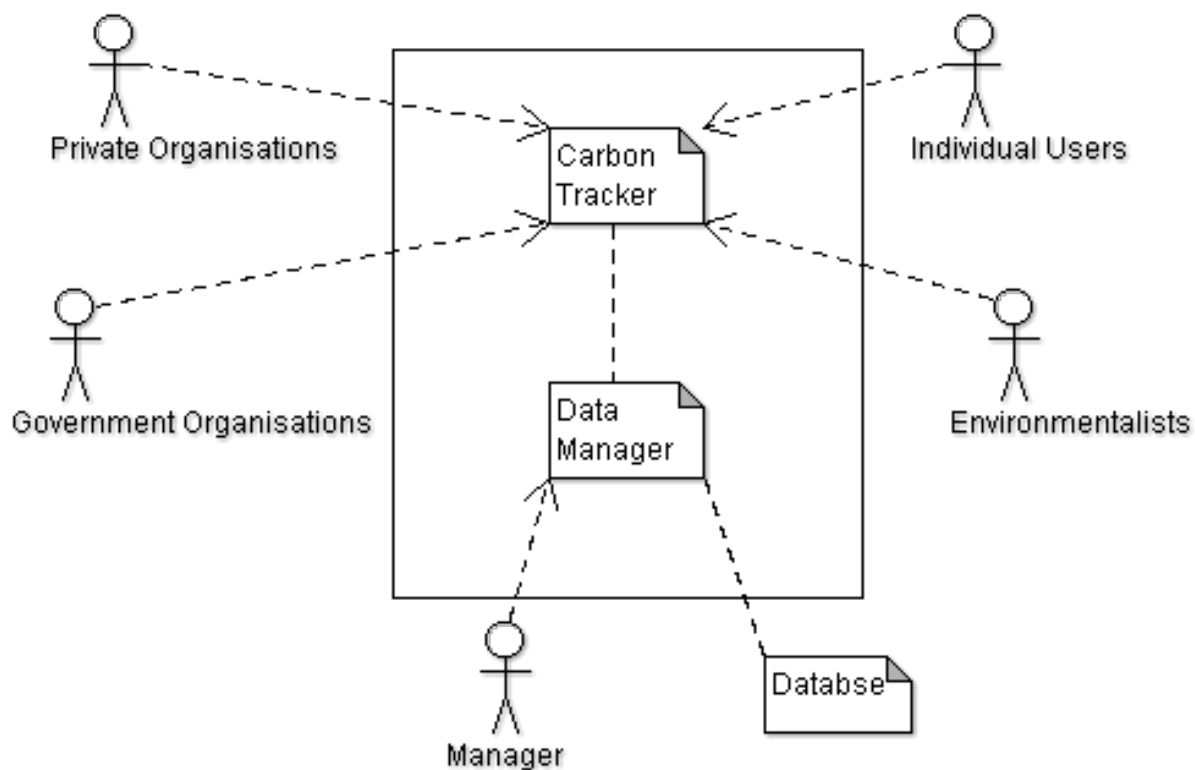
- Environmentalists and government organisations are frequently notified about the status of their community.
- The act of “notifying” extends the act of receiving notifications of communities in the case of certain actors.
- The activity of notification stands alone for the remaining actors.
- Thus, notification is a type of “extends” relationship.



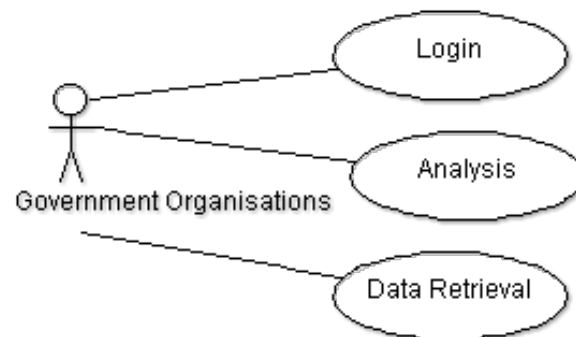
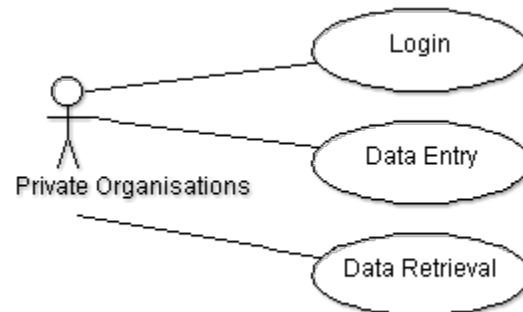
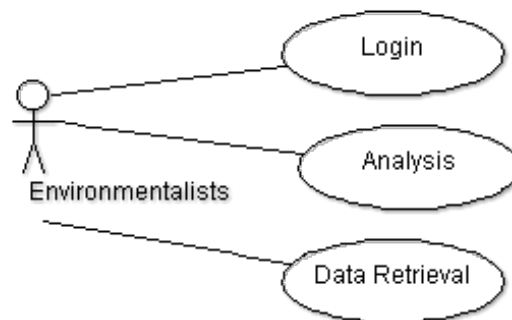
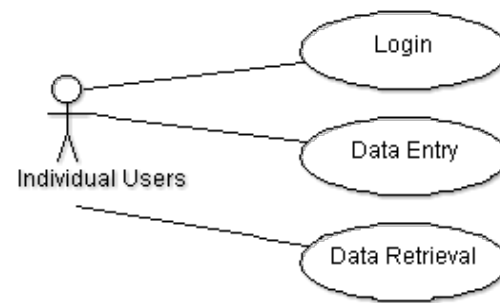


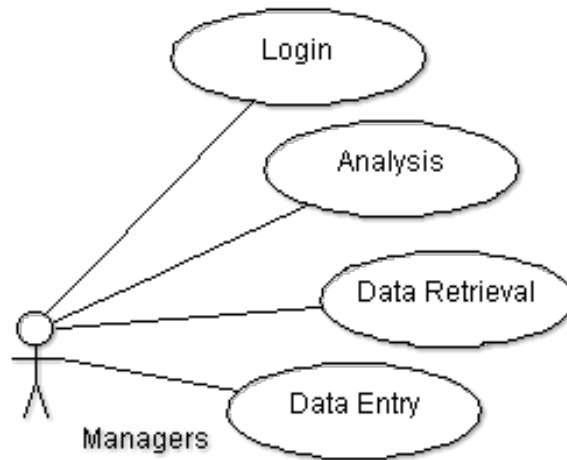
## 6. Use Case Diagrams

Overall use case diagram:

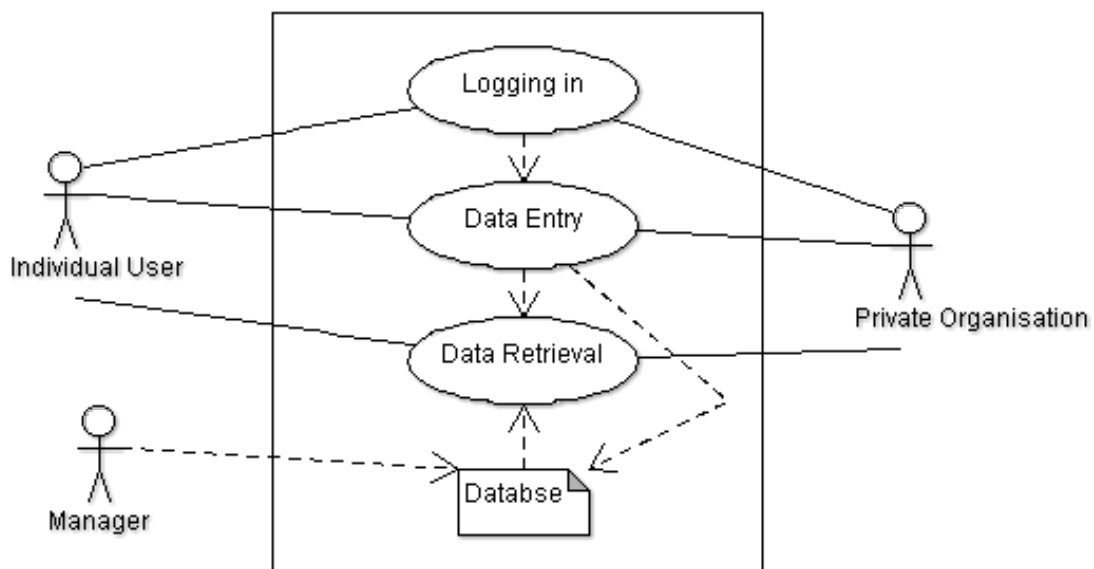


Actors and Use Case Sub Functions:

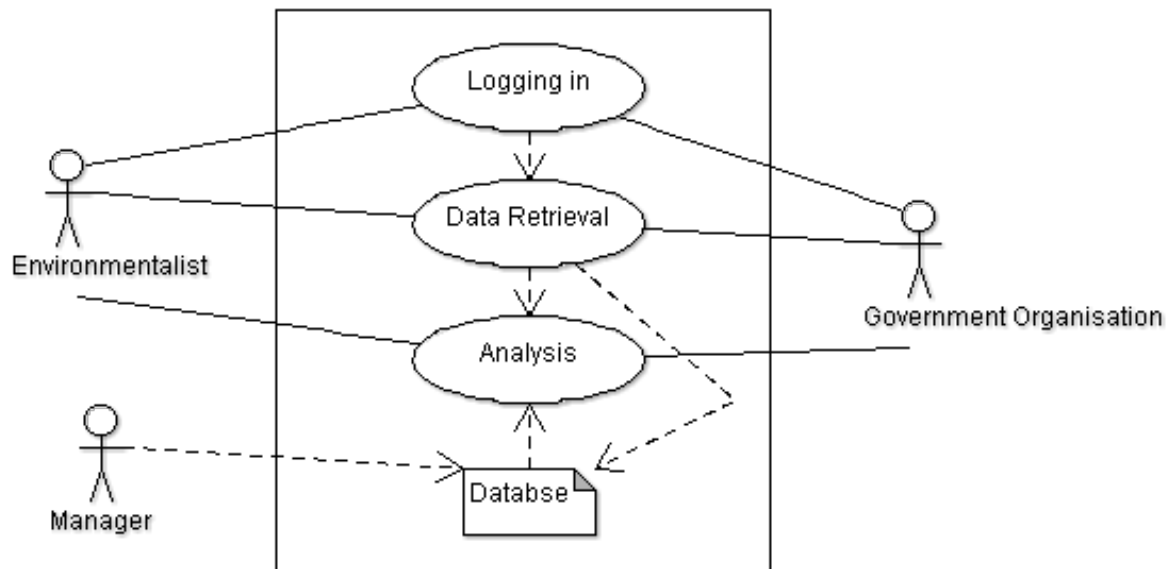




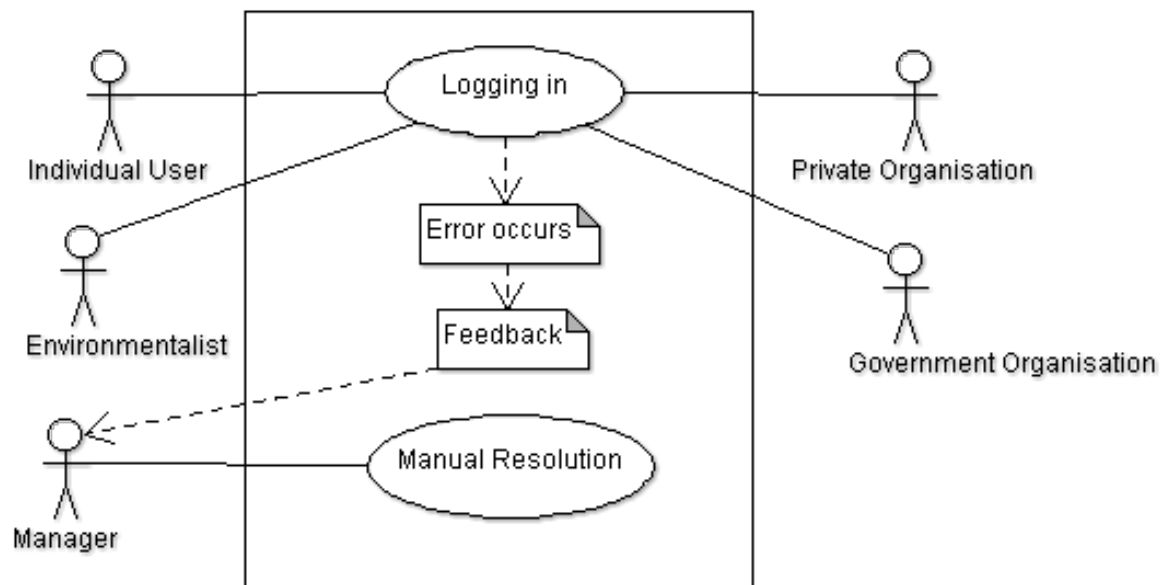
**Success scenario 1:**



**Success scenario 2:**



**Failure scenario:**



## 7. Fully Dressed Use Case Description

Topic	Description
<b>Scope</b>	Ideal web-based carbon footprint tracking
<b>Level</b>	User goal
<b>Primary Actor</b>	Individual user
<b>Stakeholders and Interests</b>	<ul style="list-style-type: none"> <li>- Individual User: wants fast, accurate results and comparison with personal and community goals.</li> <li>- Environmentalist: wants the status of their community and timely notifications regarding the same.</li> <li>- Private Organisation: wants its employees data in a single place and an overall cumulative status in terms of goal.</li> <li>- Government Organisation: wants the status of a set of areas and communities and notifications for the same.</li> </ul>
<b>Preconditions</b>	Only registered users can login
<b>Success Guarantee</b>	The carbon footprint calculated for every category is accurate. The analysis results sent are also accurate.
<b>Main success scenario</b>	<ol style="list-style-type: none"> <li>1. User logs in to the individual user account.</li> <li>2. User enters activity data under each category.</li> <li>3. This data is entered in the database.</li> <li>4. System computes the carbon footprint.</li> <li>5. System generates trends and goals at both the personal level and community level.</li> <li>6. These trends are retrieved from the database and the user can view them.</li> <li>7. User logs out.</li> </ol>
<b>Frequent Alternate scenario</b>	<ol style="list-style-type: none"> <li>1. User logs in to the special user account.</li> <li>2. User has access to a dashboard which displays the data of all the individual users and organisations under their jurisdiction.</li> <li>3. This data is retrieved from the database.</li> <li>4. Data analysis is performed.</li> <li>5. User logs out.</li> </ol>
<b>Failure Scenario</b>	<ol style="list-style-type: none"> <li>1. User logs in to their account.</li> <li>2. Error occurs (either login error or calculation error).</li> <li>3. User reports the error as part of feedback.</li> <li>4. Feedback is sent to the manager.</li> <li>5. Manager manually resolves the error.</li> </ol>
<b>Frequent Subfunction</b>	<ol style="list-style-type: none"> <li>1. Logging in.</li> </ol>

	<ol style="list-style-type: none"> <li>2. Data entry by actors.</li> <li>3. Data retrieval by actors.</li> <li>4. Analysis of retrieved data.</li> </ol>
--	--

## 8. Documentation

The use case diagram denotes the usage of some frequent subfunctions. Logging in must be performed by all users irrespective of their actor status. The primary actors are the individual users and private organisations that provide data for the database of the system. The secondary actors constitute the environmentalists and the government organisations that make decisions based on the analysis of the retrieved data. Finally the management side is catered to by the manager who has the rights to access the data of all types of users and manual access over the database.

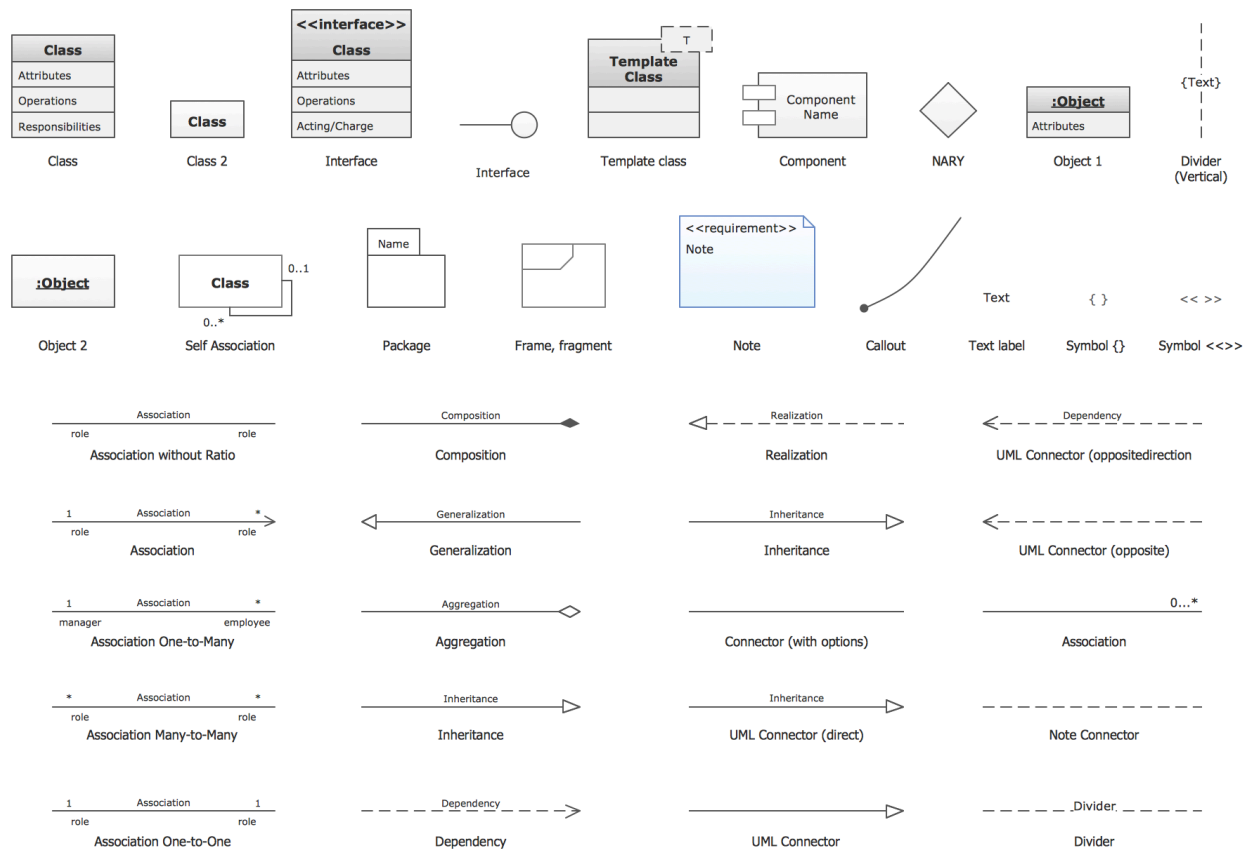
---

## EX4. Domain Model and Class Diagram

### 1. Aim

To identify UML Notations for Domain model diagram and Class diagram, and to draw the same for the proposed project Carbon Footprint Tracker.

### 2. UML Notations for Domain Model and Class Diagram



### 3. Identification of Classes

#### a. Conceptual Class Category List

Category	Conceptual Classes
System	System
Database	Database
Administrator	Administrator
Login types	Login, User, Authority
Types of users	Person, PvtOrg
Types of authority	Environmentalism, GovOrg
Carbon Footprint calculation	Carbon Footprint, Travel, Home, Food

## b. Identification of Noun Phrases

- The **System** data is stored in a **Database**.
- The **Administrator** has complete access to the **System**.
- In order to use the application the user must **Login** to the **System**.
- **Users** and **Authorities** can **Login**.
- The **Users** consist of **Persons** and **Private Organisations**.
- The **Authorities** consist of **Environmentalists** and **Government Organisations**.
- The **Carbon Footprint** is computed for **Travel**, **Home** and **Food** categories.
- **Persons** and **Private Organisations** enter **Carbon Footprint** data under **Travel**, **Home** and **Food** categories.
- **Environmentalists** and **Government Organisations** view **Carbon Footprint** data under **Travel**, **Home** and **Food** categories.
- The **Authorities** send feedback to the **Users** based on their **Carbon Footprint**.

## 4. Identification of Associations

### a. Association Category list

Description	Association
The System data is stored in a Database.	System - Database
The Administrator has complete access to the System.	System - Administrator
In order to use the application the System must be Logged into.	System - Login



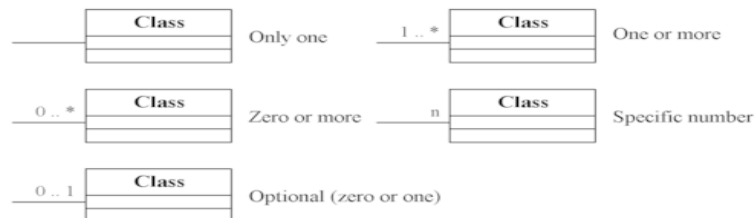
Users and Authorities can Login.	Login - User Login - Authority
The Users consist of Persons and Private Organisations.	User - Person User - PvtOrg
The Authorities consist of Environmentalists and Government Organisations.	Authority - Environmentalist Authority - GovOrg
The Carbon Footprint is computed for Travel, Home and Food categories.	Carbon Footprint - Travel Carbon Footprint - Home Carbon Footprint - Food
Persons and Private Organisations enter Carbon Footprint data under categories.	Carbon Footprint - Person Carbon Footprint - PvtOrg
Environmentalists and Government Organisations view Carbon Footprint data categories.	Carbon Footprint - Environmentalist Carbon Footprint - GovOrg
The Authorities send feedback to the Users based on their Carbon Footprint.	User - Authority

## b. Definition of Associations and their notations

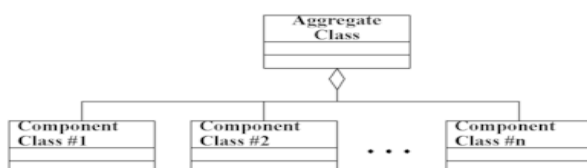
### Association between classes



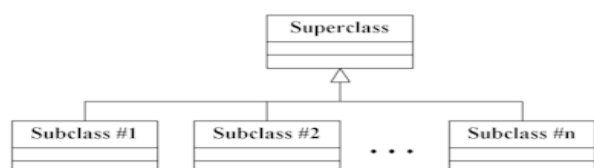
### Association Cardinality



### Aggregation between classes



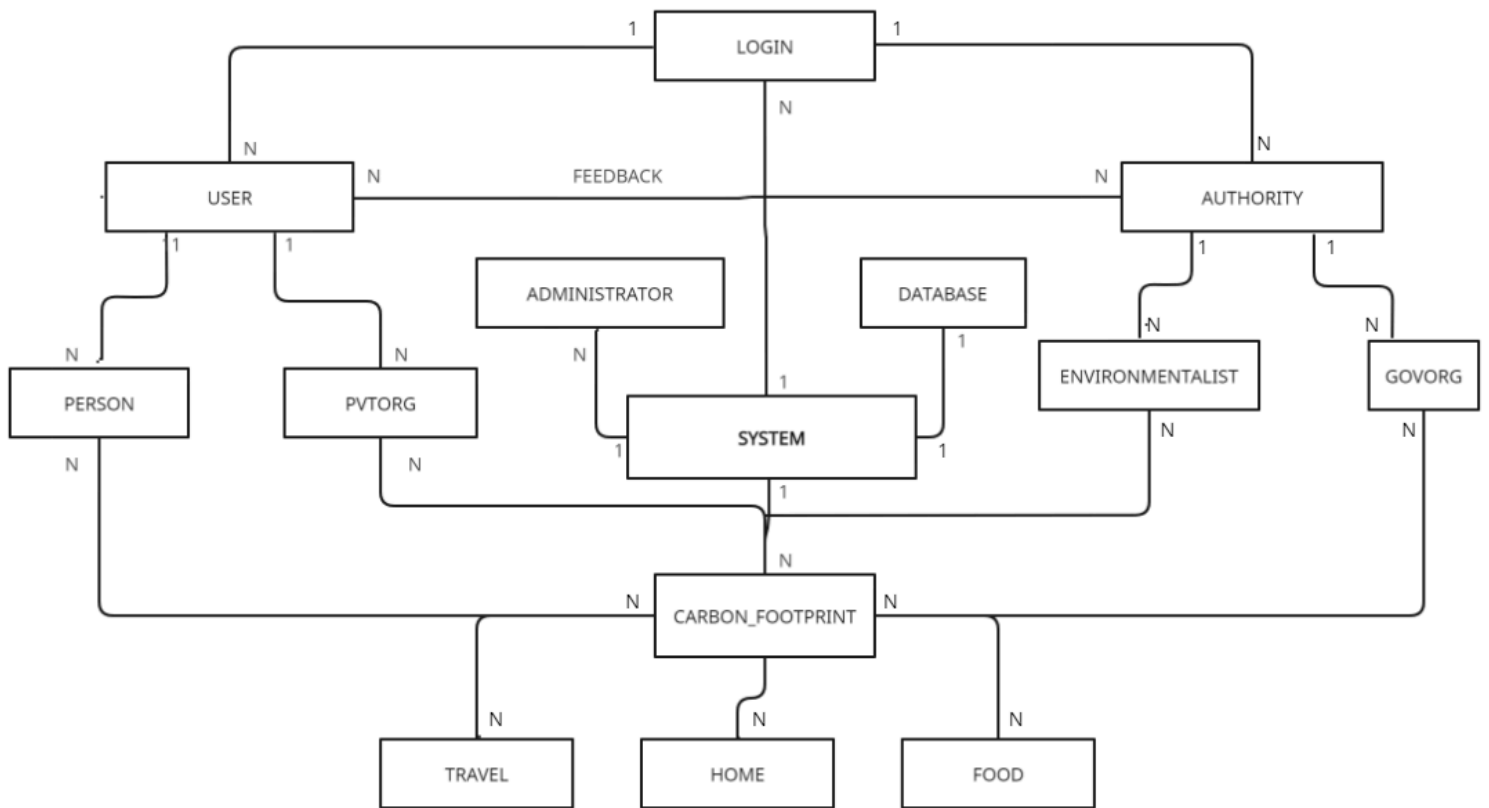
### Class inheritance (subtyping of classes)



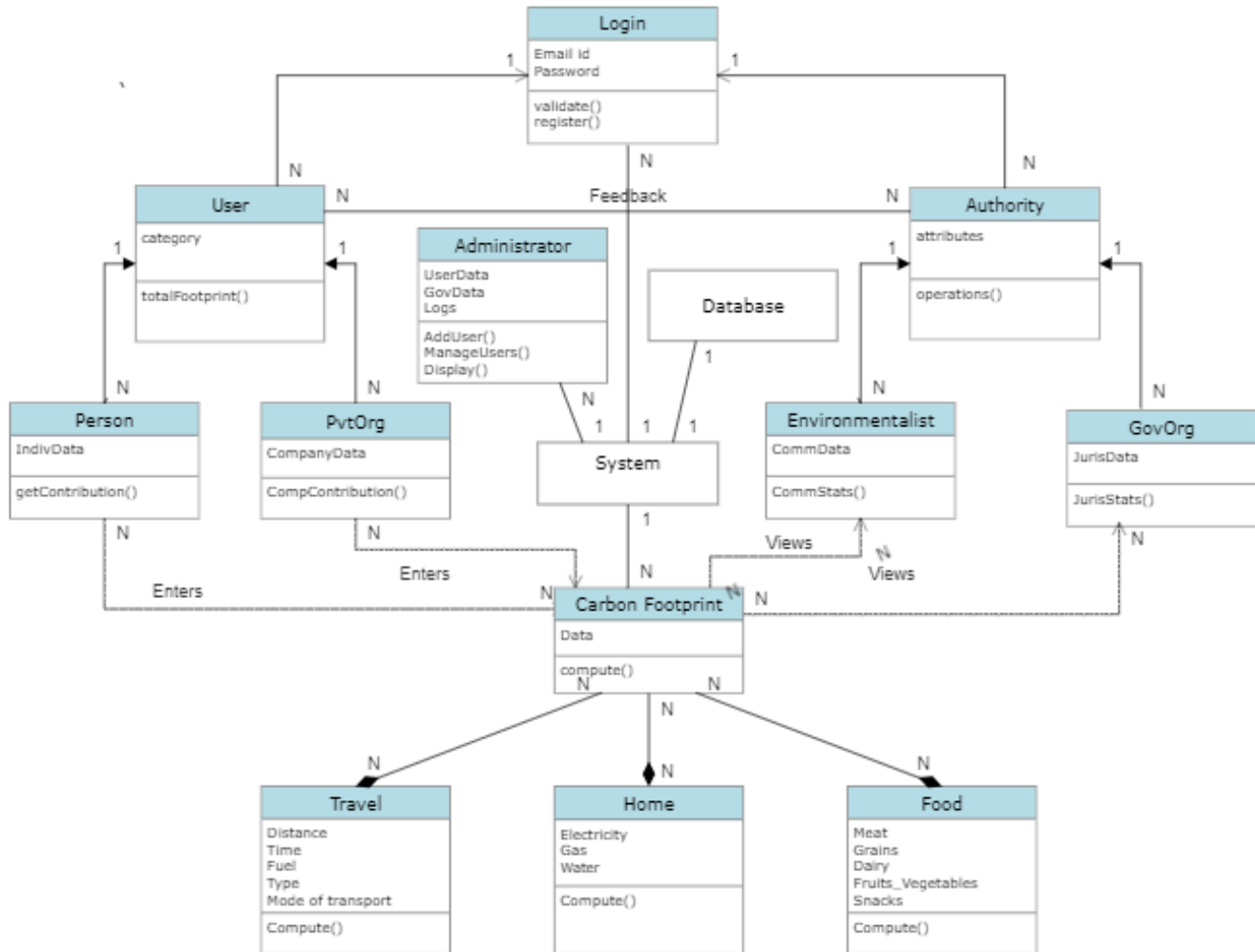
### c. Multiplicity based associations

Class 1	Class 2	Multiplicity Factor
System	Database	1 ... 1
System	Administrator	1 .. N
System	Login	1 .. N
System	Carbon Footprint	1 .. N
Login	User	1 .. N
Login	Authority	1 .. N
User	Person	1 .. N
User	PvtOrg	1 .. N
Authority	Environmentalism	1 .. N
Authority	GovOrg	1 .. N
Person	Carbon Footprint	N .. N
PvtOrg	Carbon Footprint	N .. N
Environmentalism	Carbon Footprint	N .. N
GovOrg	Carbon Footprint	N .. N
Carbon Footprint	Travel	N .. N
Carbon Footprint	Home	N .. N
Carbon Footprint	Food	N .. N
User	Authority	N .. N

## 5. Domain Model Diagram



## 6. Class Diagram



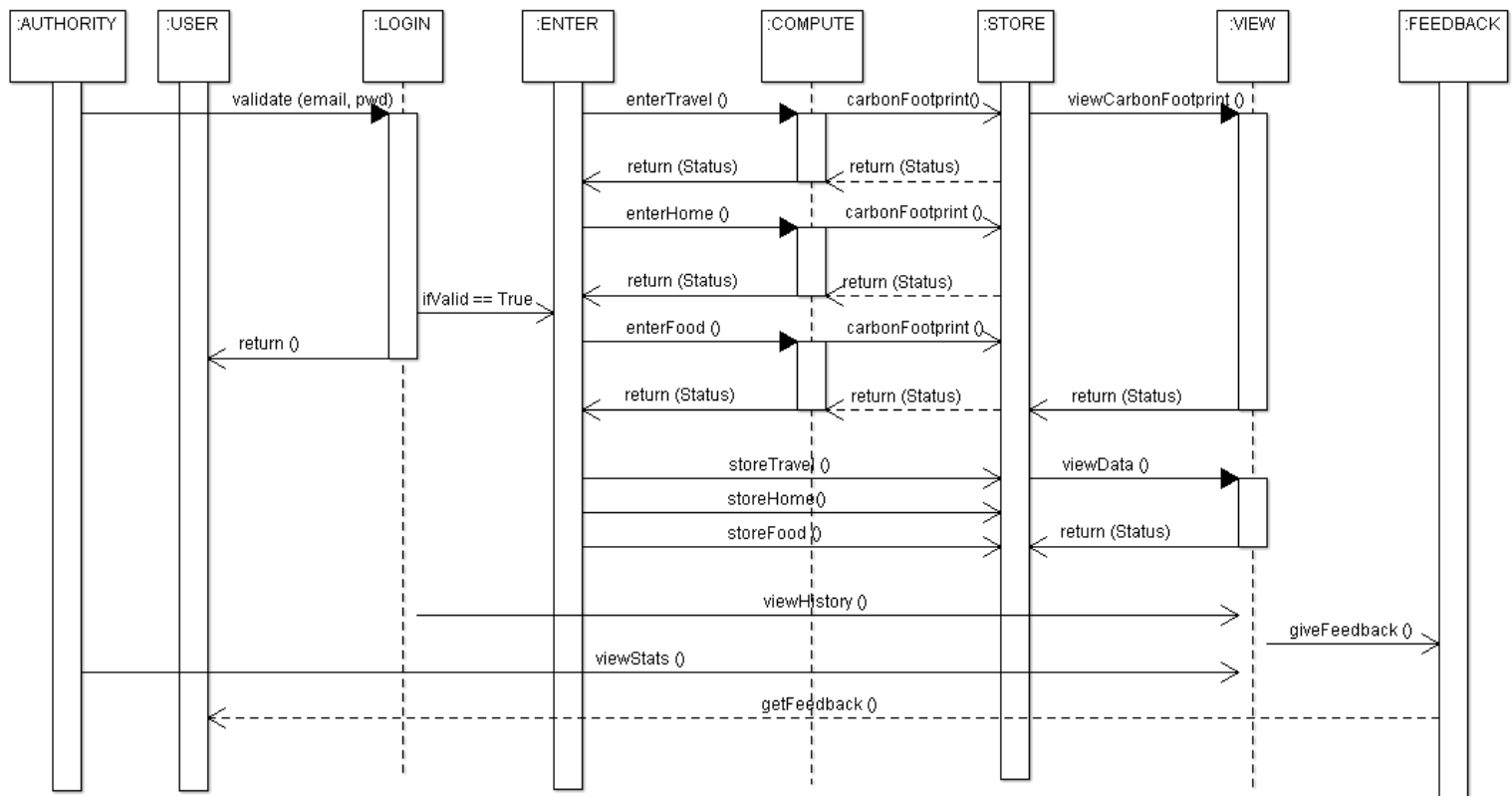
## 7. Documentation

The conceptual classes, associations and attributes necessary for drawing the domain model and the class diagram were identified in a sequential manner. The UML notations for the same have been figured, and the representational diagram drawn. The database stores all the data entered by the various users over a period of time. The carbon footprint is calculated based on the travel, home and food activity data entered by the user. The general users consisting of individual persons and private organisations enter their data, while authorities such as environmentalists and government organisations view the trends and statistics charted based on user data. The authorities also give feedback and set rules and standards for the users to follow, based on analysis performed on the carbon footprint data.

## EX5. Interaction Diagrams

### 1. Sequence Diagram - Main Success Scenario

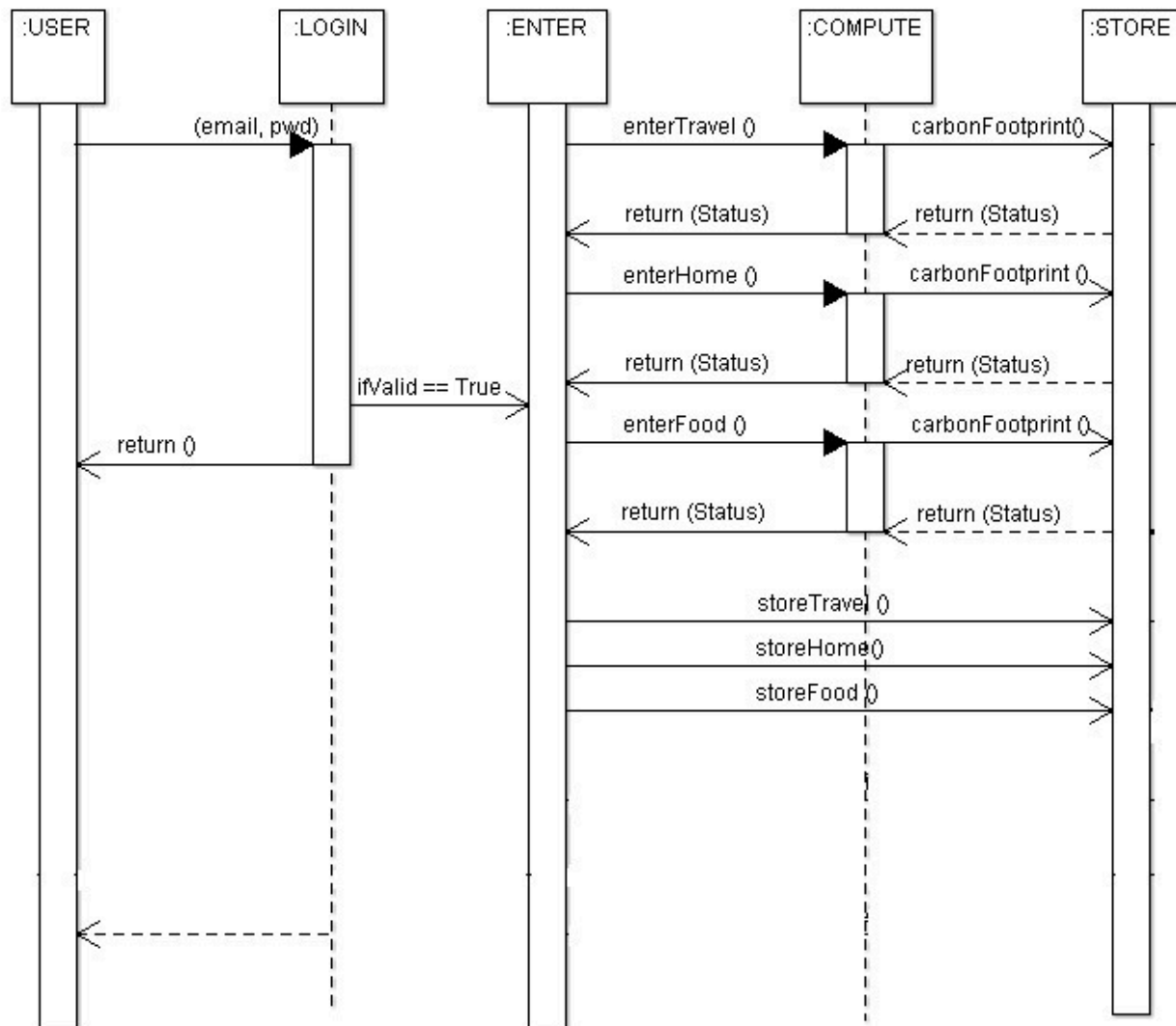
A Sequence diagram for the main success scenario in the use case diagram.



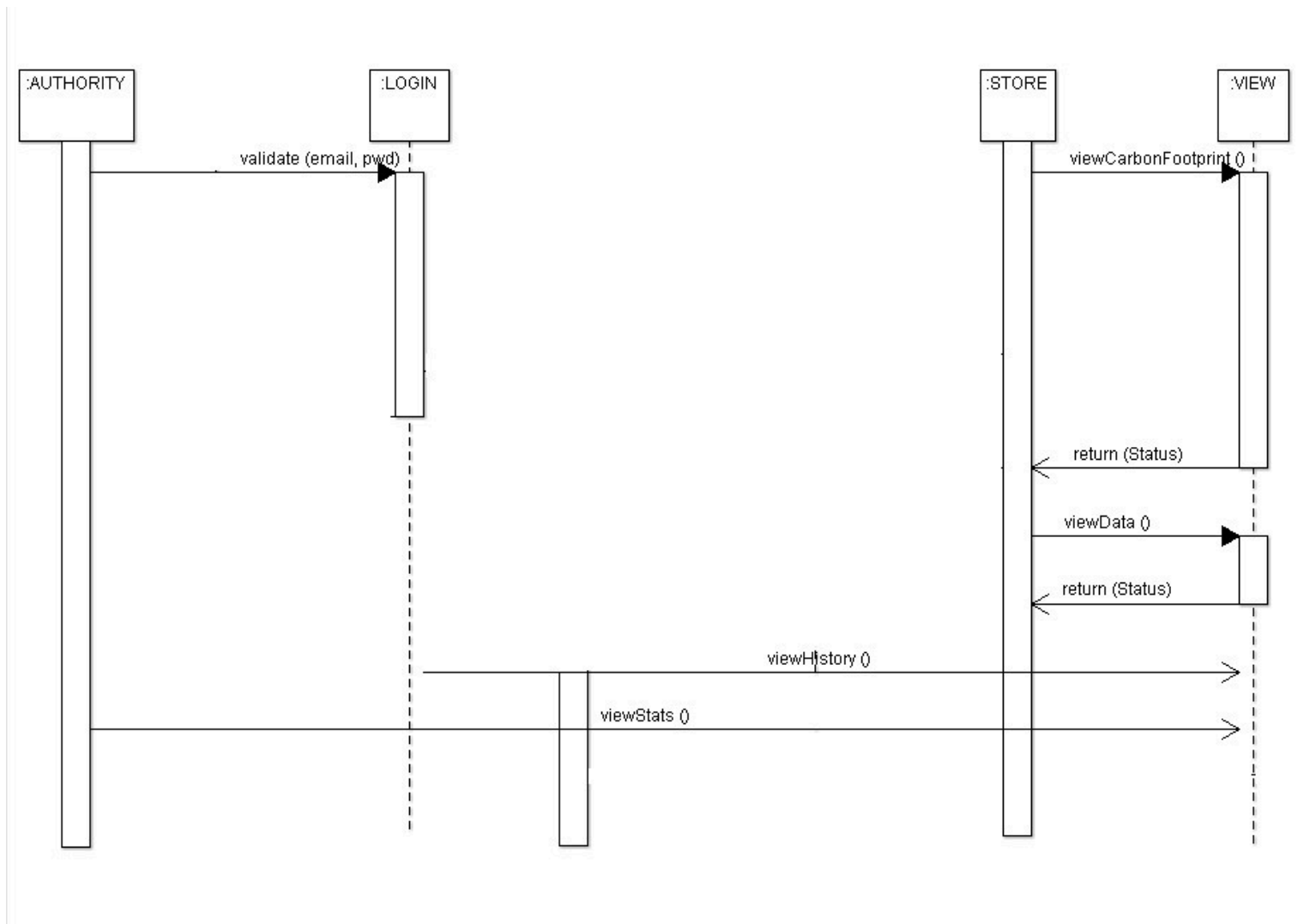
### 2. Sequence Diagram - Each Use Case

A Sequence diagram for each function (Use Case)

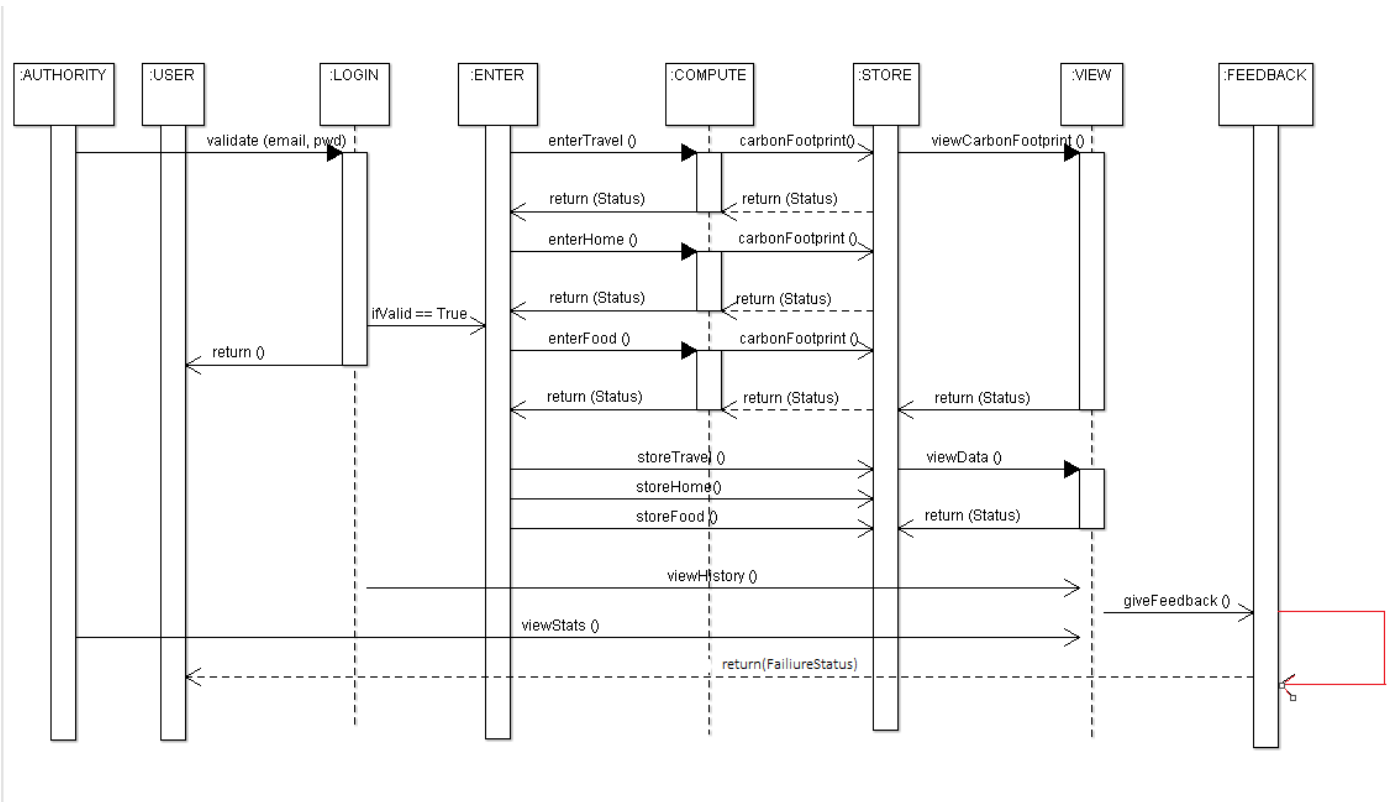
1. User enters activity data using which the system computed the Carbon Footprint.



**2. Authority views the carbon footprint statistics of all Users under their jurisdiction.**



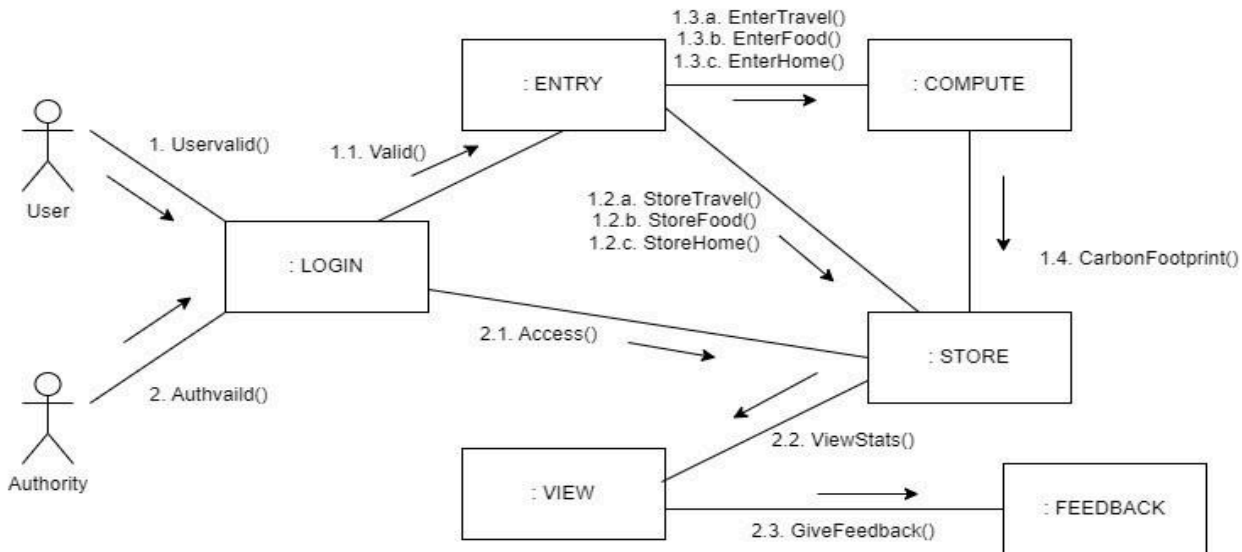
### 3. Failure scenario where there is an issue with relaying the Authority feedback to the User.





### 3. Collaboration Diagram - Main Success Scenario.

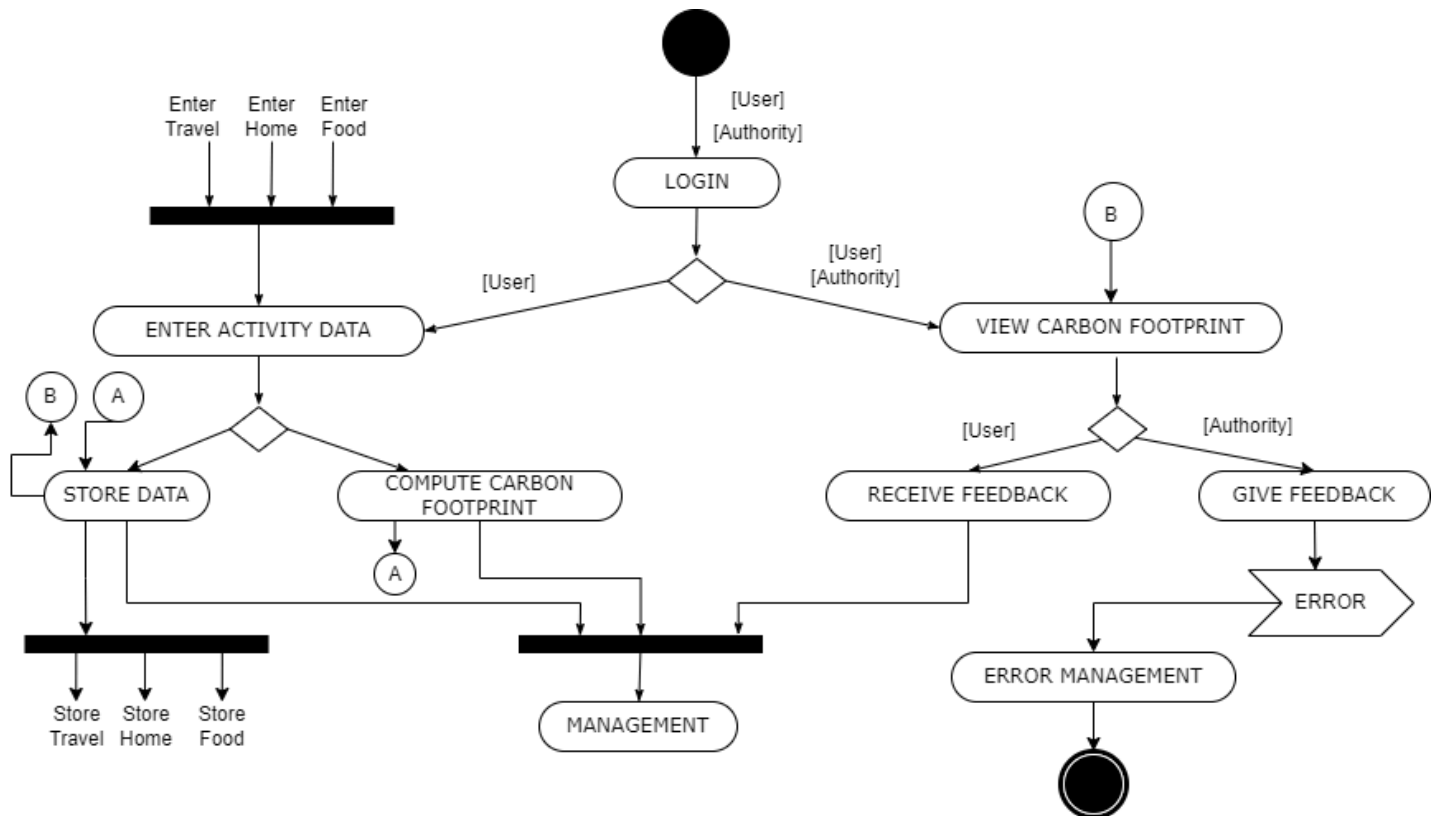
Collaboration diagram for the main success scenario in the use case diagram.



## EX6. Interaction Diagrams

### 1. Main Success Scenario

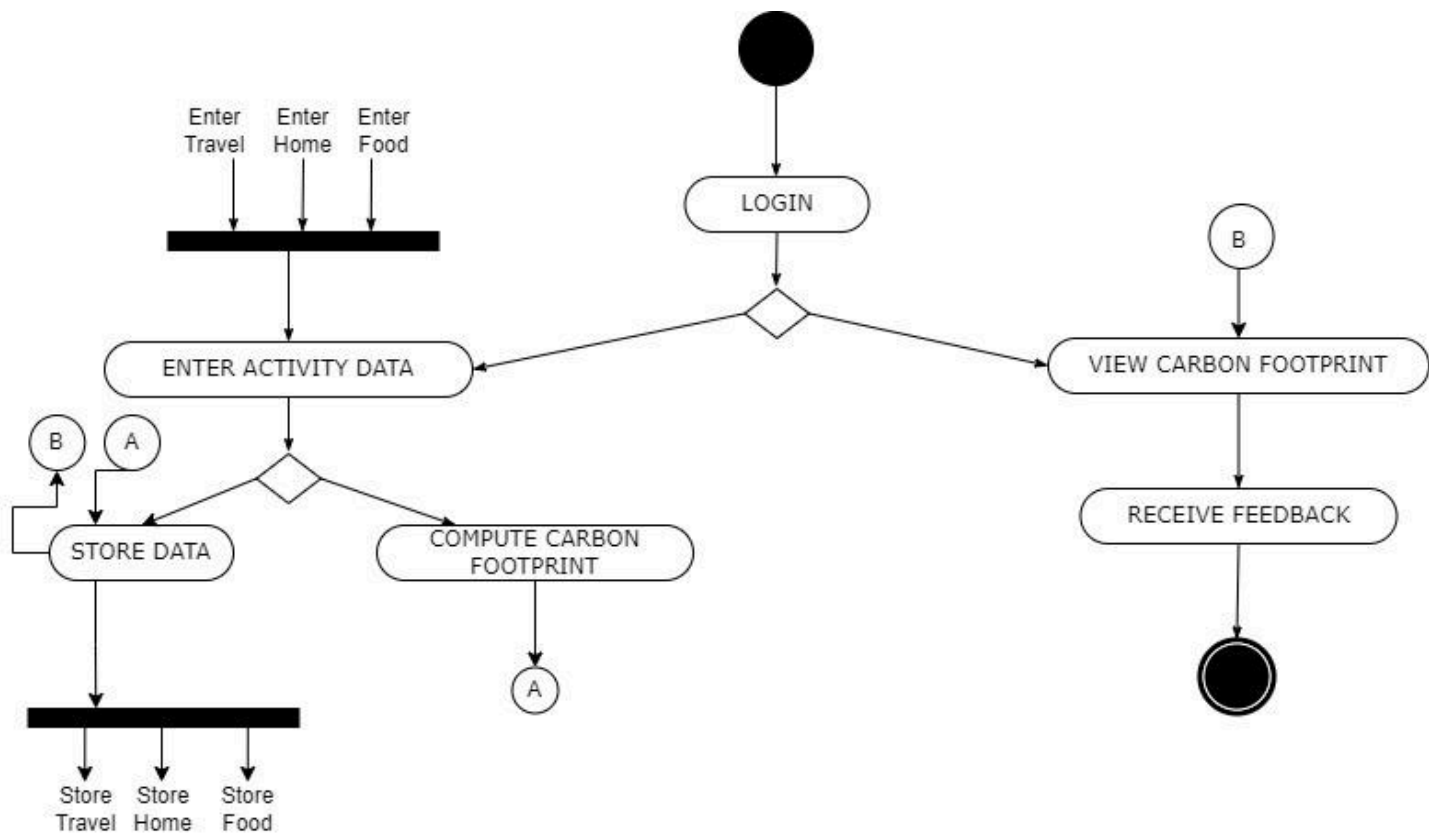
An Activity diagram for the main success scenario in the use case diagram.



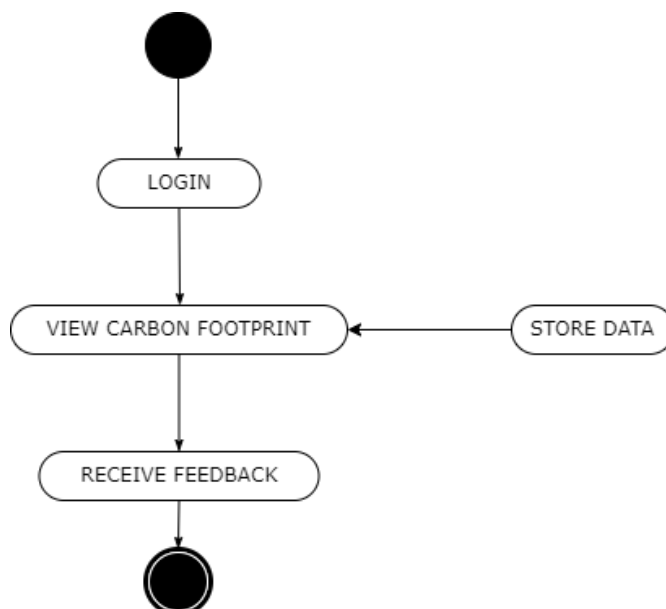
### 2. Each Use Case

An Activity diagram for each function (Use Case)

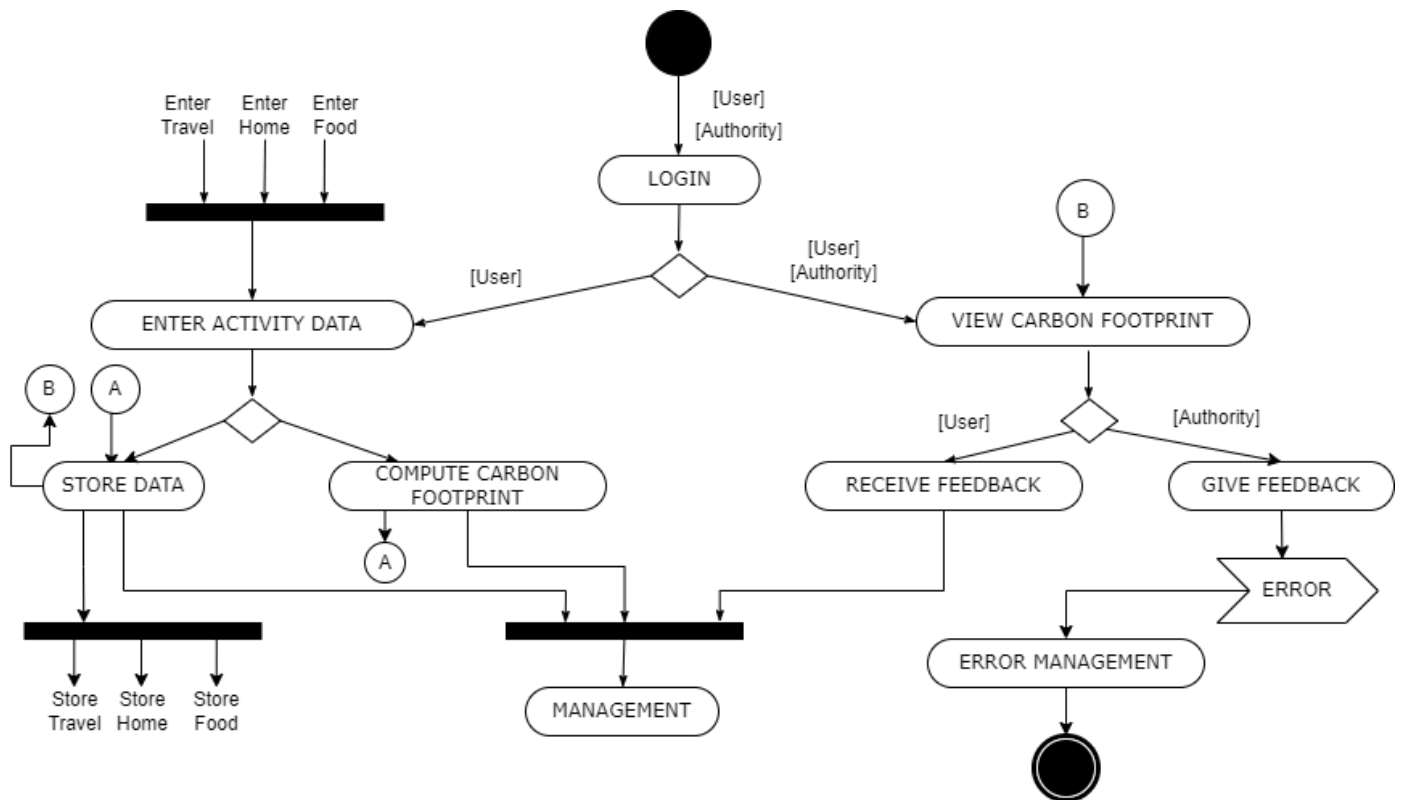
1. User enters activity data using which the system computed the Carbon Footprint.



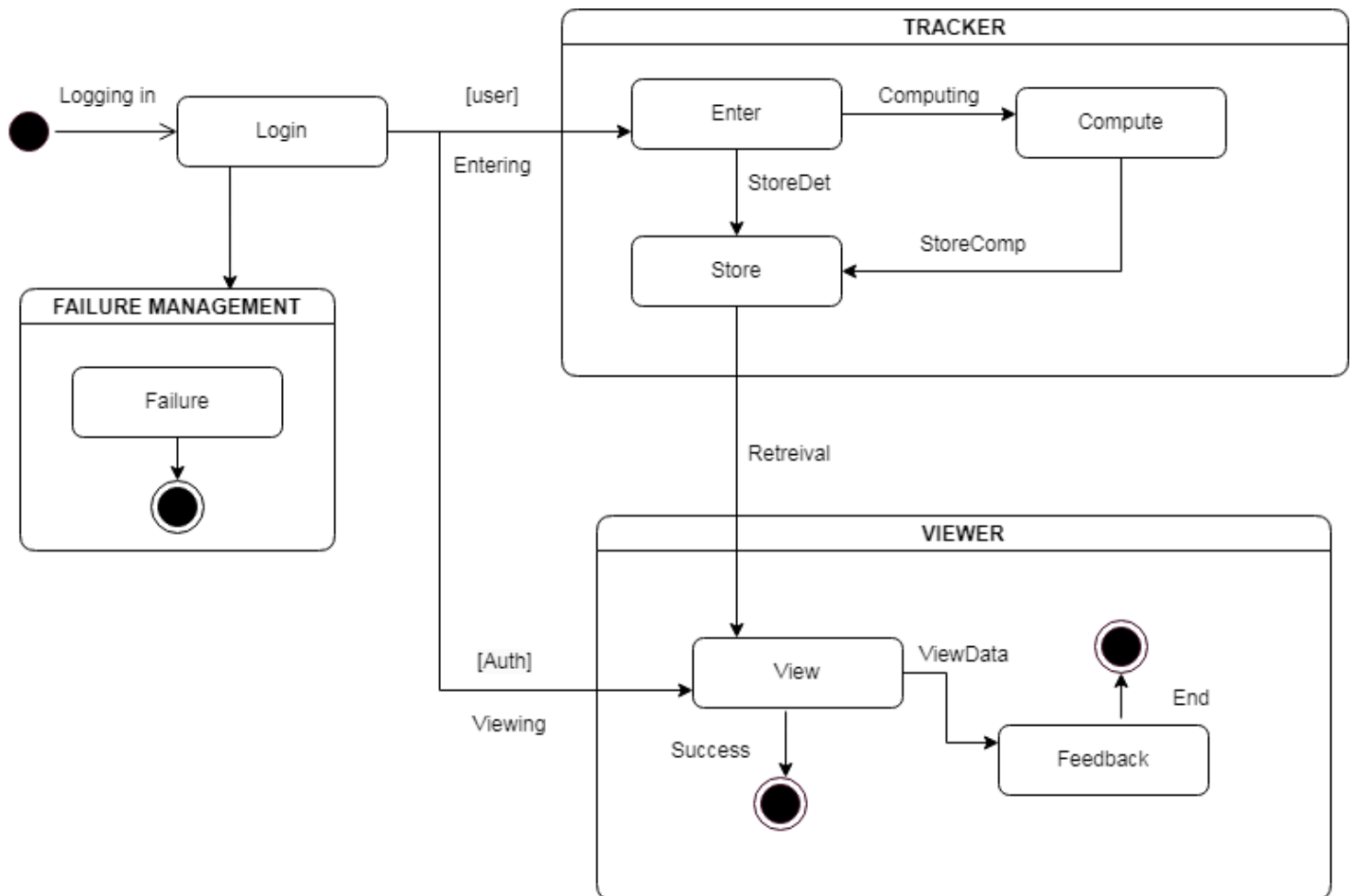
2. Authority views the carbon footprint statistics of all Users under their jurisdiction.



### 3. Failure scenario where there is an issue with relaying the Authority feedback to the User.

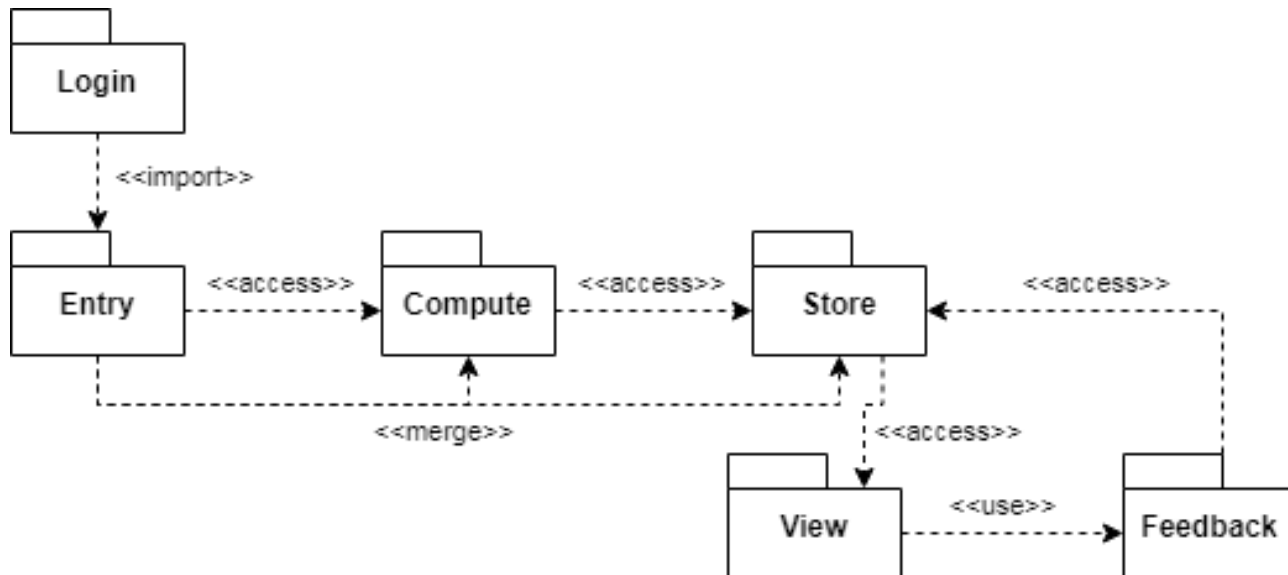


## EX7. State Machine Diagram

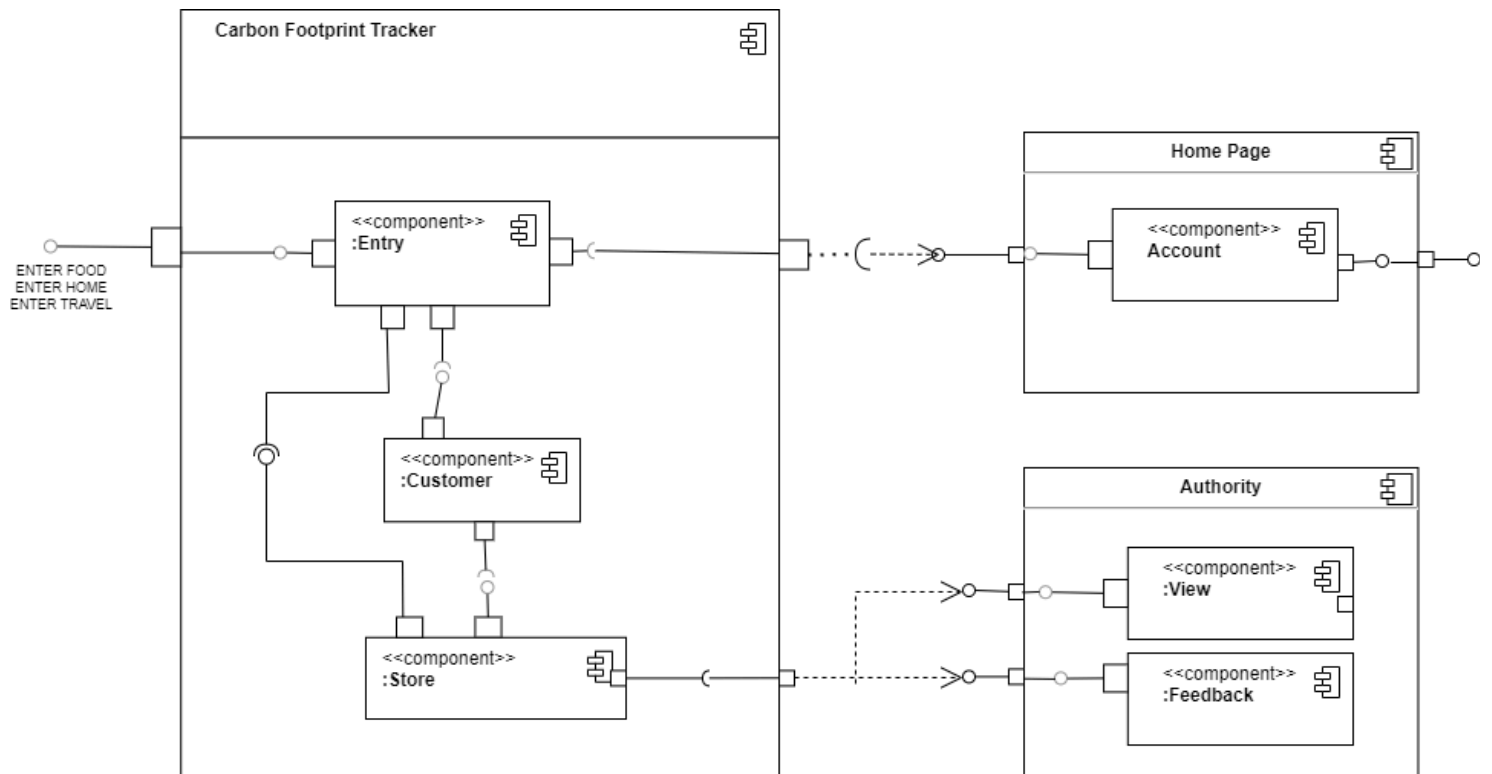


## **EX8. Package, Component and Deployment Diagram**

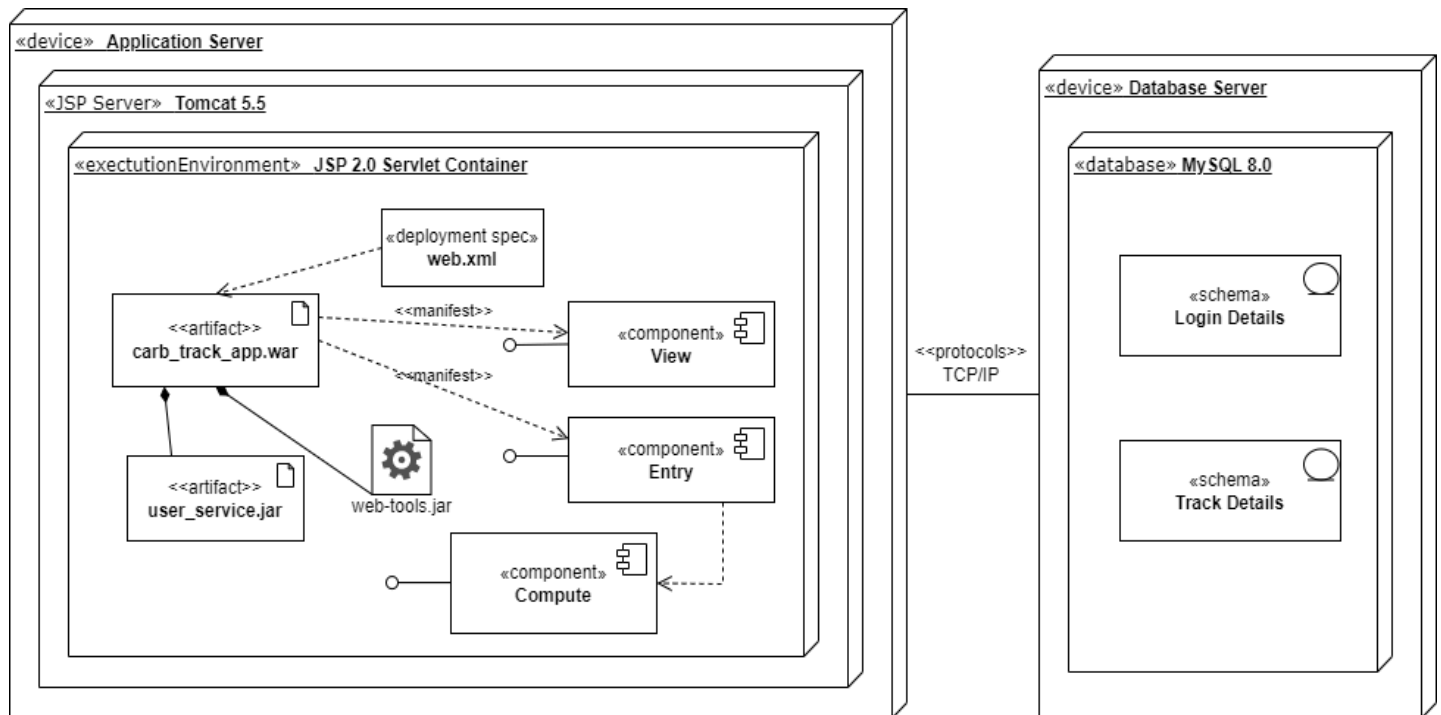
### **1. Package Diagram**



## 2. Component Diagram



### 3. Deployment Diagram





## ***EX9. Implementation of UI & Domain Layers***

This can be found as a .zip file in lms

---

## ***EX10. Implementation of Technical Service Layer***

This can be found as a .zip file in lms

---

## **EX11. Developing Test Plan & Test Cases**

### **1. Identification of Testing Scenarios**

1. Check if user can be authenticated and logged in
2. Check if admin can be authenticated and logged in
3. Check if new user can register successfully
4. Check if Home Details page can be accessed
5. Check if values for Home can be submitted
6. Check if Food Details page can be accessed
7. Check if values for Food can be submitted
8. Check if Travel Details page can be accessed
9. Check if values for Travel can be submitted
10. Check if Statistics page can be viewed

### **2. Tabulation of Test Cases**

SNO	TEST SCENARIO	TESTING STEPS	TEST DATA	EXPECTED OUTCOME	ACTUAL OUTCOME	PASS /FAIL
T01	User authentication	Enter login credentials like user ID and password, submit the credentials	UserID: mia Password: 1	Invalid (incorrect password)	Valid	Fail
T02	User authentication	Enter login credentials like user ID and password, submit the credentials	UserID: mia Password: 123	Valid	Valid	Pass
T03	Admin authentication	Enter login credentials like Admin ID and password, submit the credentials	UserID: ria Password: 981	Invalid (Invalid Admin ID)	Valid	Fail
T04	Admin authentication	Enter login credentials like user id and password, submit the credentials	UserID: a001 Password: 981	Valid	Valid	Pass
T05	New Registration	Click the register button, and enter a	UserID: new Name: Ken	Invalid details	Invalid	Pass

		new user id, name, age, address and new password. Submit the form	Age: -1 Address: NYC Password: abc	(Invalid Age)		
T06	New Registration	Click the register button, and enter a new user id, name, age, address and new password. Submit the form	UserID: new Name: Ken Age: 21 Address: NYC Password: abc	Valid Details	Valid	Pass
T07	Accessing Home Details page	Navigate to Home Details page from the Dashboard.	Category: Seed requested page: unavailable	Invalid (broken link for requested page)	Valid	Fail
T08	Accessing Home Details page	Navigate to Home Details page from the Dashboard.	Category: Seed requested page: available	Valid	Valid	Pass
T09	Entering Home Details	Enter Home details such as amount of Water, Electricity and Gas used, at a particular moment of data entry. Press Submit.	Water: -1 units Gas: 50 units Electricity 2 units	Invalid Entry	Valid	Fail
T10	Entering Home Details	Enter Home details such as amount of Water, Electricity and Gas used, at a particular moment of data entry. Press Submit.	Water: 50 units Gas: 50 units Electricity 2 units	Valid	Valid	Pass
T11	Accessing Food Details page	Navigate to the Food Details page from the Dashboard.	Category: Seed requested page: unavailable	Invalid (broken link for requested page)	Invalid	Pass
T12	Accessing Food Details page	Navigate to the Food Details page from the Dashboard.	Category: Seed requested page: available	Valid	Valid	Pass
T13	Entering Food Details	Enter Food details such as amount of	Meat: 10 units Dairy: 10 units	Valid Entry	Valid	Pass

		meat, grains, dairy, snacks, fruits and vegetables, at a particular moment of data entry. Press Submit.	Fruits: 20 units			
T14	Entering Food Details	Enter Food details such as amount of meat, grains, dairy, snacks, fruits and vegetables, at a particular moment of data entry. Press Submit.	Meat: -1 units Dairy: -1 units Fruits: 20 units	Invalid Entry	Invalid	Pass
T15	Accessing Travel Details page	Accessing Home Details page	Category: Seed requested page: unavailable	Invalid (broken link for requested page)	Valid	Fail
T16	Accessing Travel Details page	Navigate to Travel Details page from the Dashboard.	Category: Seed requested page: available	Valid	Valid	Pass
T17	Entering Travel Details	Enter Travel details such as selecting mode of transport (car/bus/cycle/air/train ) and distance travelled, at a particular moment of data entry. Press Submit.	Mode: Bus Distance: 23 kms	Valid Entry	Valid	Pass
T18	Entering Travel Details	Enter Travel details such as selecting mode of transport (car/bus/cycle/air/train ) and distance travelled, at a particular moment of data entry. Press Submit.	Distance: 23 kms	Invalid Entry (No mode selected!)	Invalid	Pass
T19	Viewing Stats Page	Accessing the View Stats button from the Home/Food/Travel Details page and	Category: Seed requested page: available	Invalid (Displays wrong statistics)	Valid	Fail

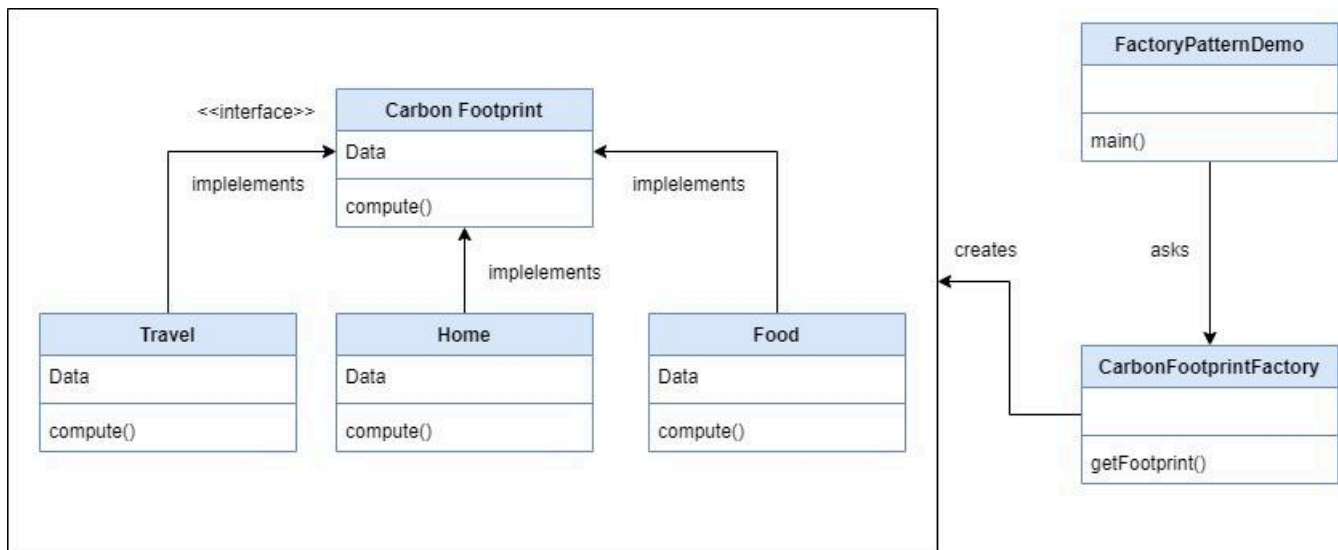
		viewing today's statistics.				
T20	Viewing Stats Page	Accessing the View Stats button from the Home/Food/Travel Details page and viewing today's statistics.	Category: Seed requested page: available	Valid	Valid	Pass

---

## EX12. Applying Design Patterns

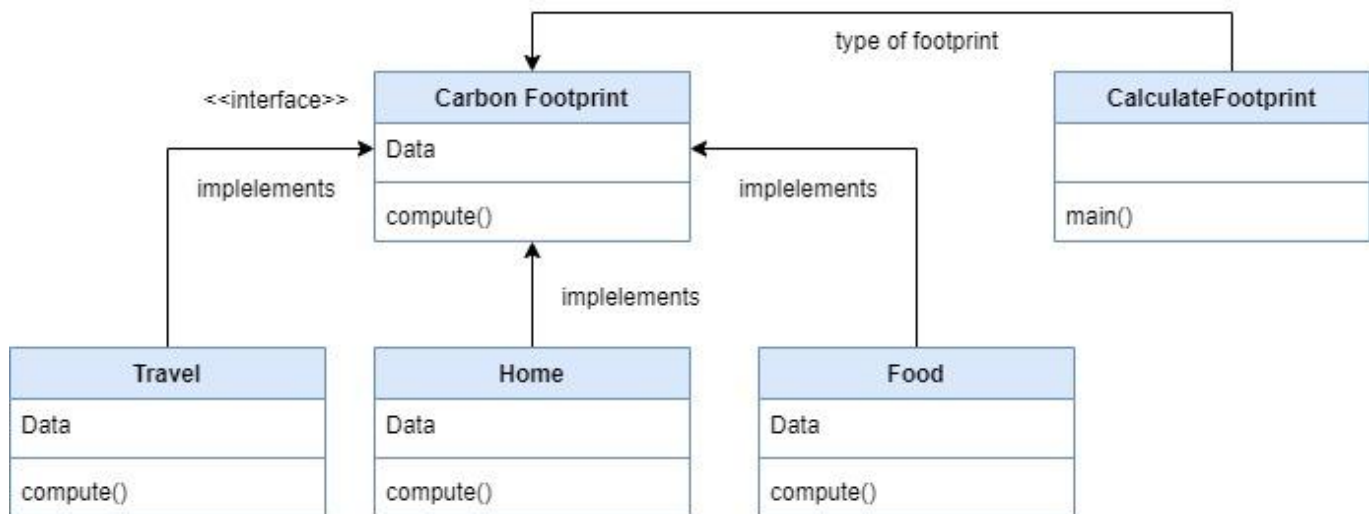
### 1. Factory Pattern

The Factory Method pattern can be used to create different types of carbon footprint calculators depending on the type of activity. For example, a calculator for travel will be different from a calculator for food consumption emissions and home activity emissions. This will allow the system to be modular and easily extensible.



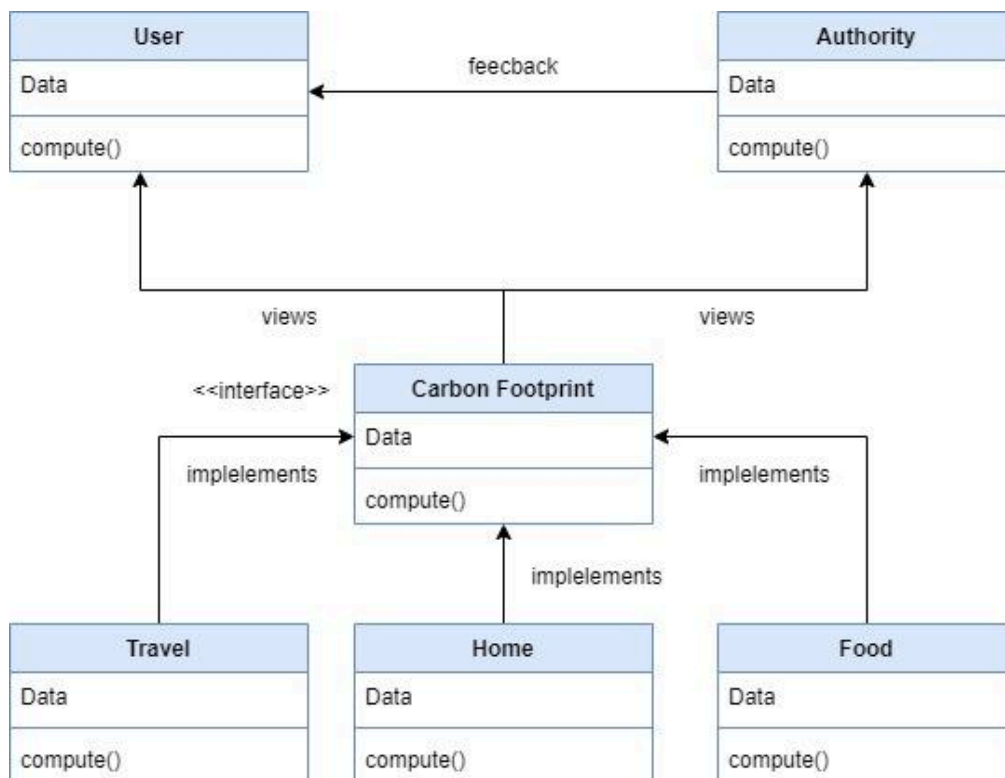
### 2. Strategy Pattern

The Strategy pattern can be used to implement different carbon footprint calculation strategies based on various factors like the type of activity. This will allow the system to accurately calculate the carbon footprint of different activities and provide users with more accurate data.



### 3. Observer Pattern

The Observer pattern can be used to keep track of changes in the carbon footprint of a user or an organisation. Whenever there is a change in the carbon footprint, the observers will be notified, allowing them to take appropriate actions. (feedback system).



## **EX13. Test Cases After Refinement**

### **1. Identification of Testing Scenarios**

11. Check if user can be authenticated and logged in
12. Check if admin can be authenticated and logged in
13. Check if new user can register successfully
14. Check if Home Details page can be accessed
15. Check if values for Home can be submitted
16. Check if Food Details page can be accessed
17. Check if values for Food can be submitted
18. Check if Travel Details page can be accessed
19. Check if values for Travel can be submitted
20. Check if Statistics page can be viewed
21. Check if Dashboard page can be revisited from the other pages

### **2. Tabulation of Test Cases**

SNO	TEST SCENARIO	TESTING STEPS	TEST DATA	EXPECTED OUTCOME	ACTUAL OUTCOME	PASS /FAIL
T01	User authentication	Enter login credentials like user ID and password, submit the credentials	UserID: mia Password: 1	Invalid (incorrect password)	Invalid	Pass
T02	User authentication	Enter login credentials like user ID and password, submit the credentials	UserID: mia Password: 123	Valid	Valid	Pass
T03	Admin authentication	Enter login credentials like Admin ID and password, submit the credentials	UserID: ria Password: 981	Invalid (Invalid Admin ID)	Invalid	Pass
T04	Admin authentication	Enter login credentials like user id and password, submit the credentials	UserID: a001 Password: 981	Valid	Valid	Pass



T05	New Registration	Click the register button, and enter a new user id, name, age, address and new password. Submit the form	UserID: new Name: Ken Age: -1 Address: NYC Password: abc	Invalid details (Invalid Age)	Invalid	Pass
T06	New Registration	Click the register button, and enter a new user id, name, age, address and new password. Submit the form	UserID: new Name: Ken Age: 21 Address: NYC Password: abc	Valid Details	Valid	Pass
T07	Accessing Home Details page	Navigate to Home Details page from the Dashboard.	Category: Seed requested page: unavailable	Invalid (broken link for requested page)	Invalid	Pass
T08	Accessing Home Details page	Navigate to Home Details page from the Dashboard.	Category: Seed requested page: available	Valid	Valid	Pass
T09	Entering Home Details	Enter Home details such as amount of Water, Electricity and Gas used, at a particular moment of data entry. Press Submit.	Water: -1 units Gas: 50 units Electricity 2 units	Invalid Entry	Invalid	Pass
T10	Entering Home Details	Enter Home details such as amount of Water, Electricity and Gas used, at a particular moment of data entry. Press Submit.	Water: 50 units Gas: 50 units Electricity 2 units	Valid	Valid	Pass
T11	Accessing Food Details page	Navigate to the Food Details page from the Dashboard.	Category: Seed requested page: unavailable	Invalid (broken link for requested page)	Invalid	Pass
T12	Accessing Food Details page	Navigate to the Food Details page from the Dashboard.	Category: Seed requested page: available	Valid	Valid	Pass

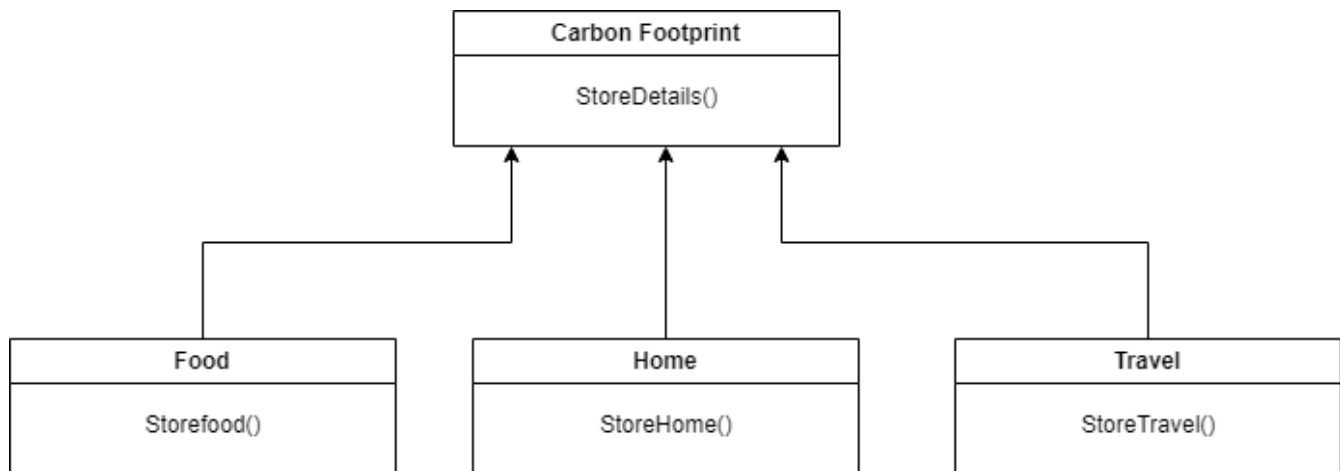
T13	Entering Food Details	Enter Food details such as amount of meat, grains, dairy, snacks, fruits and vegetables, at a particular moment of data entry. Press Submit.	Meat: 10 units Dairy: 10 units Fruits: 20 units	Valid Entry	Valid	Pass
T14	Entering Food Details	Enter Food details such as amount of meat, grains, dairy, snacks, fruits and vegetables, at a particular moment of data entry. Press Submit.	Meat: -1 units Dairy: -1 units Fruits: 20 units	Invalid Entry	Invalid	Pass
T15	Accessing Travel Details page	Accessing Home Details page	Category: Seed requested page: unavailable	Invalid (broken link for requested page)	Invalid	Pass
T16	Accessing Travel Details page	Navigate to Travel Details page from the Dashboard.	Category: Seed requested page: available	Valid	Valid	Pass
T17	Entering Travel Details	Enter Travel details such as selecting mode of transport (car/bus/cycle/air/train) and distance travelled, at a particular moment of data entry. Press Submit.	Mode: Bus Distance: 23 kms	Valid Entry	Valid	Pass
T18	Entering Travel Details	Enter Travel details such as selecting mode of transport (car/bus/cycle/air/train) and distance travelled, at a particular moment of data entry. Press Submit.	Distance: 23 kms	Invalid Entry (No mode selected!)	Invalid	Pass
T19	Viewing Stats Page	Accessing the View Stats button from the	Category: Seed requested page:	Invalid (Displays	Invalid	Pass

		Home/Food/Travel Details page and viewing today's statistics.	available	wrong statistics)		
T20	Viewing Stats Page	Accessing the View Stats button from the Home/Food/Travel Details page and viewing today's statistics.	Category: Seed requested page: available	Valid	Valid	Pass
T21	Going back to Dashboard	Going back to Dashboard from Stats Page	Category: Seed requested page: available	Valid	Valid	Pass
T22	Going back to Dashboard	Going back to Dashboard from Home Page	Category: Seed requested page: available	Valid	Valid	Pass
T23	Going back to Dashboard	Going back to Dashboard from Food Page	Category: Seed requested page: available	Valid	Valid	Pass
T24	Going back to Dashboard	Going back to Dashboard from Travel Page	Category: Seed requested page: available	Valid	Valid	Pass

## EX14. Refactoring

Refactoring can be used to refine the functionalities and simplify the code. It enables the developer to easily update and modify the existing functionalities. Refactoring helps in maintaining the code in the long run.

Here, the separate functions like StoreFood(), StoreHome() and storeTravel() can be combined and delivered as a single functionality storeDetails()



In the same way, separate functions like EnterFood(), EnterHome() and EnterTravel() can be combined and delivered as a single functionality EnterDetails()

