sustainability

MDPI

Article

Development of a Mobile Application System for Eco-Accounting

Hua Huang, Daizhong Su *, Wenjie Peng and You Wu

Advanced Design and Manufacturing Engineering Centre, School of Architecture, Design and the Built Environment, Nottingham Trent University, Nottingham NG1 4FQ, UK; hua.huang@ntu.ac.uk (H.H.); wenjie.peng@ntu.ac.uk (W.P.); you.wu@ntu.ac.uk (Y.W.)

* Correspondence: daizhong.su@ntu.ac.uk; Tel.: +44-(0)115-848-2306

Received: 3 November 2020; Accepted: 14 November 2020; Published: 19 November 2020

Abstract: Nowadays, eco-accounting is widely used in sustainable consumption and production. In order to incentivise consumers' sustainable consumption and enhance their environmental awareness, a novel mobile based eco-accounting infrastructure has been developed by this research. It applies the eco-credit values to incentivise the consumer's recycling activities and utilises the eco-cost values to record the consumer's footprint obtained through consumption. The infrastructure consists of four modules: the consumer's eco-account, eco-shopping, eco-recycling and eco-incentives. In order to implement the mobile eco-accounting infrastructure, multiple mobile technologies have been applied to develop the novel functions of the mobile app, including a new QR encryption algorithm, embedded Google maps, advanced Internet-based services and multi-language support. A case study is conducted by demonstrating the consumer's purchasing and recycling processes. It proves that the novel mobile application system has been successfully developed, which provides an effective support for the implementation and demonstration of the eco-accounting infrastructure.

Keywords: mobile app; eco-accounting; eco-shopping; recycling; eco-incentive; sustainable consumption; sustainability

1. Introduction

Sustainable consumption is an important agenda in relation to global climate change and the reduction of human beings' impact on the environment. As an instrument to measure sustainable consumption behaviours, eco-accounting is currently attracting great attention. Eco-accounting is used for the generation, analysis and use of financial and related non-financial information in order to effectively solve the problems encountered in the process of sustainable development [1,2]. As a tool to report sustainable impacts, it has been applied in ecological footprint measurements, e.g., water footprint [3–5] and carbon footprint [6,7], soil quality assessment [8], ecosystems [9] and energy efficiency [10]. Studies have revealed that eco-accounting is able to calculate environmental impacts from various aspects, however, incorporating lifecycle impact assessment methods into the eco-accounting framework would provide an extended eco-accounting solution from the perspective of products' life cycle. With this solution, the accounting will be able to cover major environmental impact categories through the product's business supply chain and to share the eco-accounting results among stakeholders, which would be one step further to support sustainable development in the circular economy context. Many scholars have applied the conventional eco-accounting merits for major stakeholders of supply chains in various scenarios such as eco-innovation for businesses [11–13], and decision making for policy implications [4,6]. However, few studies have accommodated the eco-accounting results for consumers and other stakeholders of the product supply chain.

Some studies proposed the use of eco-accounting results for consumers to participate in the decision-making process of sustainable consumption, e.g., using environmental and health indicators to indicate food nutritional footprint [14], affecting consumers' purchase behaviours by ecolabels indicating product ecological information [15–17]. These studies have proven the necessity of interacting with consumers for eco-accounting practices, but they addressed sustainable consumption only and focused on individual consumption phases, which means that the complete view of eco-accounting for the product supply chain is still missing.

With the wide application of mobile technologies, some mobile applications have been developed to provide mobile phone users with the eco-logical information of products and services to encourage their sustainable behaviours in various scenarios, such as estimating the users' greenhouse gas emissions with their paired bank transactions data [18,19]; presenting the product information (e.g., price, expiry date, quality indicators) to consumers while grocery shopping [20]; encouraging car drivers to reduce traffic and its consequent impacts on the environment, climate change and human health [21]; scheduling appointments for solid waste collection in order to reduce environmental burdens mainly linked to the disposal procedures, and then providing eco-feedback information, e.g., energy consumption, for householders [22]. Those mobile applications demonstrate a means of interacting with consumers to encourage sustainable behaviour and have proven to enable efficient sharing and communication with up to date eco-logical information for consumers. However, they address challenges in a sole scenario, for impacts from consumer regular consumption and recycling activities, and lack a systematic solution to accommodate consumers' recording, monitoring and retrieving behaviours for eco-logical impacts. Existing individual mobile applications for eco-accounting deal with only partial aspects of the eco-accounting, without comprehensive functions. For instance, only the functions of eco-recycling and eco-incentive are dealt with in the reGAIN app [23], from which customers cannot obtain products' ecological information, i.e., eco-point value, and eco-account function is also not developed for customers.

To overcome the shortcomings of the existing mobile applications for eco-accounting, a novel mobile application, CIRC4Life app, has been developed with an eco-accounting infrastructure, which uses eco-cost values to record consumers' footprints resulting from shopping. The app also uses eco-credit values to record and incentivise consumers' recycling activities and provides individual consumers with personal eco-accounts. The CIRC4Life app is an outcome of the CIRC4Life project supported by European Commission's H2020 programme [24] and is based on the experience of the authors' team from the Advanced Design and Manufacturing Engineering Centre (ADMEC) of Nottingham Trent University. The ADMEC has been actively involved in the research in eco-accounting. The myEcoCost project supported by the European Commission FP7 Eco-environment programme [25–27], of which the ADMEC is a core member, developed a method to calculate the eco-cost score for products' impact on the environment. The eco-cost scores were calculated based product carbon and material footprints. Based on the eco-cost method, a prototype of eco-accounting infrastructure was developed by Su and Peng [28,29], in which an eco-point method is proposed. With the prototype, a novel eco-accounting infrastructure is further developed by the CIRC4Life project.

In the following sections, the related concepts and overall architecture of the mobile application are introduced first, then the development technologies are reported, followed by the presentation of a case study to demonstrate the operation procedures and functions of the mobile application. Finally, the conclusions are given at the end of the paper.

2. Overview of the Mobile Application System for Eco-Accounting

2.1. The Eco-Accounting Concepts

The developed mobile application aims to implement the eco-accounting infrastructure proposed in the CIRC4Life project. In order to understand the functions of the mobile application, it is necessary to explain the following key concepts: eco-cost, eco-credit and eco-account.

2.1.1. Eco-Cost

Eco-cost is used to indicate a product's negative impact on the environment. Within the eco-point approach presented in the CIRC4Life project, each product is associated with an eco-cost value that is calculated by using a lifecycle impact assessment method. The higher the eco-cost value is, the more negative environmental impact the product has. The eco-cost value of a product is added to a consumer's eco-account after his/her purchase of the product. The aggregated value of eco-costs reflects the negative environmental impacts generated through the product's life cycles, which help the consumer select more sustainable products.

2.1.2. Eco-Credit

Eco-credit is adopted to credit the consumer's positive behaviour to recycle products, which means consumers can earn the eco-credit value via recycling the end-of-life products. As per how to calculate the eco-credit of a recycled products, the details of this can be found in the CIRC4Life deliverable [25]. Within the mobile application, eco-credit values consist of eco-credits awarded due to recycling, eco-credits spent and eco-credits balance. The eco-credits recycled is the eco-credit value that is obtained by the consumer recycling activities. The eco-credits spent refer to the eco-credit value spent by a consumer in eco-incentive activities, e.g., receiving a discount on purchasing products, redeeming for theatre or cinema tickets, donating eco-credits for tree planting. The eco-credits balance is the balance of the total eco-credits recycled and the total eco-credits spent.

2.1.3. Eco-Account

The eco-account of a consumer is used to record his/her eco-costs and eco-credits related to purchasing and recycling activities. It means that eco-account can enable consumers to record and track their daily eco-footprints on environment. An example of a consumer eco-account page is demonstrated in Table 1.

Table 1. An example of a consumer's eco-account page.

Date	**Activities**	**Eco- Costs**	**Eco-Credits**		
			Recycled	**Spent**	**Balance**
01/08/2020	Purchase a tablet	67	0	0	0
08/08/2020	Recycle an iPad	0	74	0	74
16/08/2020	Purchase a lamp with discount using eco-credits	26	0	10	64
20/08/2020	Exchange ticket with eco-credits	0	0	20	44
01/09/2020	Donate eco-credits for tree planting	0	0	10	34
	Total	93	74	40	

As Table 1 shows, the eco-cost total reflects the purchased products' accumulated negative impact on the environment, and the eco-credits balance represents consumers' current positive impact on the environment. By using mobile communication technologies, consumers can track their eco-credits, eco-costs and daily eco-footprints through their mobile phones.

2.2. System Architecture

Based on the eco-point approach and the eco-accounting infrastructure developed in Section 2.1, a novel eco-accounting infrastructure using multiple mobile technologies has been implemented, which is shown in Figure 1.

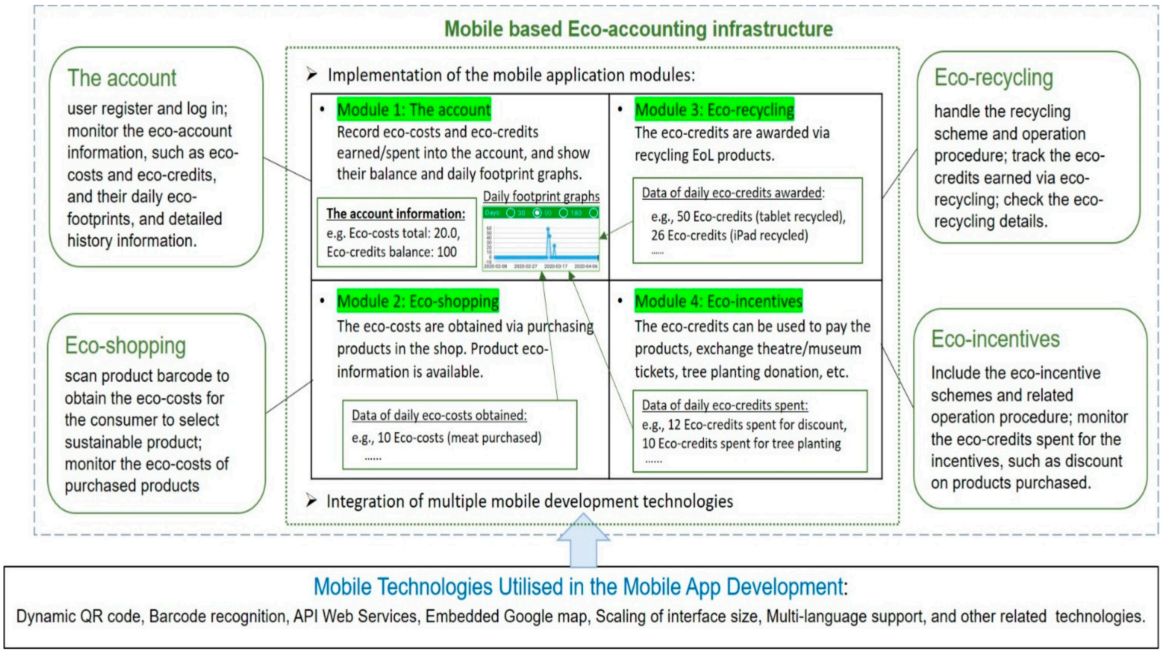

Mobile based Eco-accounting infrastructure
The account
user register and log in; monitor the eco-account information, such as eco-costs and eco-credits, and their daily eco-footprints, and detailed history information.
Eco-shopping
scan product barcode to obtain the eco-costs for the consumer to select sustainable product; monitor the eco-costs of purchased products
Implementation of the mobile application modules:
Module 1: The account
Record eco-costs and eco-credits earned/spent into the account, and show their balance and daily footprint graphs.
The account information:
e.g. Eco-costs total: 20.0, Eco-credits balance: 100
Daily footprint graphs
Module 3: Eco-recycling
The eco-credits are awarded via recycling EoL products.
Data of daily eco-credits awarded:
e.g., 50 Eco-credits (tablet recycled), 26 Eco-credits (iPad recycled)
Module 2: Eco-shopping
The eco-costs are obtained via purchasing products in the shop. Product eco-information is available.
Data of daily eco-costs obtained:
e.g., 10 Eco-costs (meat purchased)
Module 4: Eco-incentives
The eco-credits can be used to pay the products, exchange theatre/museum tickets, tree planting donation, etc.
Data of daily eco-credits spent:
e.g., 12 Eco-credits spent for discount, 10 Eco-credits spent for tree planting
Integration of multiple mobile development technologies
Eco-recycling
handle the recycling scheme and operation procedure; track the eco-credits earned via eco-recycling; check the eco-recycling details.
Eco-incentives
Include the eco-incentive schemes and related operation procedure; monitor the eco-credits spent for the incentives, such as discount on products purchased.
Mobile Technologies Utilised in the Mobile App Development:
Dynamic QR code, Barcode recognition, API Web Services, Embedded Google map, Scaling of interface size, Multi-language support, and other related technologies.

Figure 1. Overall architecture of the mobile application system for eco-accounting.

The mobile based eco-accounting infrastructure consists of four modules: the account, eco-shopping, eco-recycling and eco-incentive, which are detailed in the following sub-sections.

2.2.1. The Account Module

The account module provides four functions: user registration and login, user profile, eco-account information and eco-history detail. The user registration function allows consumers to create their eco-accounts in this system by filling their account name, email address and password. The user login function is used to validate consumers' account ID and ensure that only the registered user can log in the mobile application in order to utilise all the functions. In the user profile function, consumers can edit their personal profiles, change login password and email. The eco-account information function is used to monitor consumers' eco-cost total and eco-credit balance, and to show consumers' daily eco-footprints, which are presented by four ecological charts: eco-credits recycled chart, eco-credits spent chart, eco-credits balance chart and eco-costs chart. The eco-history detail function is designed to retrieve and list the consumers' history activities, e.g., eco-shopping activities, eco-recycling activities and eco-incentive activities.

2.2.2. Eco-Shopping Module

The eco-shopping module is designed for consumers to view the product eco-information when they purchase in stores via scanning the product barcode. It means that consumers can scan products' barcodes with their mobile phone to obtain products' eco-costs retrieved from remote database server to facilitate the selection of sustainable products and monitor the eco-costs of their purchased products.

2.2.3. Eco-Recycling Module

When consumers' products (e.g., tablet, iPad, mobile phone) reach the end-of-life, they can recycle them to obtain the corresponding eco-credits from their recycled products. In the eco-accounting system, the eco-recycling module is used to handle the recycling schemes and operation procedures (such as how to recycle electronic products and how to recycle organic waste), to provide the QR codes of consumers' recycling ID that are utilised to communicate with the intelligent recycling system (namely the intelligent bin) developed by CIRC4Life, and to track consumers' recycling history.

Through the above functions, consumers can earn eco-credits via their recycling activities and check all the recycling details with their mobile phones.

2.2.4. Eco-Incentive Module

In order to encourage consumers to recycle and reuse their products, an innovative incentive scheme is developed in the eco-accounting system, and consumers' eco-credits earned via recycling products can be awarded to their eco-account, which are redeemable for shopping discounts and theatre and cinema tickets. Therefore, the eco-incentive module is designed to guide consumers to use eco-incentive scheme with their eco-credits. This module provides functions for consumers to view and check all the history details of their eco-incentive activities.

3. System Implementation

To provide an effective means to implement the eco-accounting infrastructure, multiple mobile development technologies have been developed, including consumer ID verification based on the dynamic and secure QR code, barcode recognition, communication with the remote server, intelligent navigation via embedded Google maps, automatic scaling of user interfaces, multi-languages support for user interfaces and other related technologies, the details of which are given in the following sub-sections.

3.1. Consumer ID Verification Based on Dynamic and Secure QR Code

In CIRC4Life, the Near-Field-Communication (NFC) technology is used to validate a consumer's identity that is represented by the personal identification code. The NFC technology depends on the hardware of the mobile devices, so it is a high-cost solution for consumer ID verification. In this mobile application, the consumer ID refers to two kinds of IDs: recycling ID and shopping ID. The recycling ID is used to communicate with the intelligent recycling system developed by CIRC4Life project, and the shopping ID can be scanned by the scanner used in the eco-shopping or the eco-incentive system for initiating the communication with the remote server. To reduce the communication cost but ensure the security of consumers' private shopping IDs, the use of a dynamic QR code for consumer ID verification is adopted in this procedure. Thus, generating a dynamic and secure QR code according to a consumer's shopping ID is crucial to achieve the above function. Additionally, the shopping ID string used to generate QR code should be encrypted and obfuscated. Thereby, a novel algorithm for data obfuscation and encryption is proposed, as shown in Algorithm 1.

Algorithm 1: GenerateQRCodeString

```
Require (input parameter): The source qrcode string qrstr that needs to be encrypted and obfuscated
Procedure:
1: Generate a random integer between 1 and 10 as the encrypting key
2: If the qrstr is not null and its length is 52, then do
3:     Encrypt the qrstr with the encryption algorithm EncryptStr and the encrypting key
4:     Generate a new obfuscated qrcode string qrstr using the following rule:
5:     qrstr = getRandomstr(5)+qrstr.substring(0,12)+getRandomstr(8)+qrstr.substring(12,23)
6:             +getRandomstr(7)+qrstr.substring(23,37)+getRandomstr(10)+qrstr.substring(37,52)
7:             +getRandomstr(9)+String.valueOf(key);
8: End If
9: Return the new qrcode string qrstr that has been encrypted and obfuscated
```

In Algorithm 1, GenerateQRCodeString has an input parameter qrstr, which denotes the source qrcode string (i.e., shopping ID string) that needs to be encrypted and obfuscated. In Line 1, a random integer between 1 and 10 is generated as the encrypting key used in the encryption algorithm. In this mobile application, each consumer's shopping ID string always has 52 characters, which means that the length of the shopping

ID string is fixed. Therefore, it is necessary to check whether the string qrstr is not null and its length is 52. As shown in Lines 2 and 3, if the qrstr is not null and it has 52 characters, then the encryption EncryptStr will be called to encrypt the string qrstr using the encrypting key as its input parameter. The detail procedure of the encryption EncryptStr can be found in Algorithm 2. In Lines 4–7, the encrypted string qrstr is obfuscated by adding the encrypting key at the end of the string and inserting the random string with different length in different position of the encrypted string qrstr, where the function getRandomStr is used to obtain the random string with the length of the input number, and the function substring (start_index,end_index) is used to get the substring from start_index to end_index in the qrstr. In Line 9, the new qrcode string qrstr that has been encrypted and obfuscated is returned as the input parameter of the function GenerateQRCodeImg that is used to generate a QR code. The function GenerateQRCodeImg can be executed once the page for displaying the OR code of a consumer's shopping ID (namely QR code page) is created and loaded in the mobile application. The QR code page can be unlimitedly loaded, thus each QR code generated by the function GenerateQRCodeImg is different, which means each generated QR code is dynamic and secure.

Algorithm 2: EncryptStr

```
  Require (input parameter):
str —— the string that needs to be encrypted and obfuscated
key —— the number that is used to encrypt the string str
Procedure:
1: Define a string restr used for storing the encrypted string
2: If str is not null then do
3:    For each character c in str do
4:       If character c is a lowercase then do
5:          Replace character c with the key'th letter after c in the 26 lowercase alphabet
6:       End If
7:       If character c is a uppercase then do
8:          Replace character c with the key'th letter after c in the 26 uppercase alphabet
9:       End If
10:      Add character c to the string restr
11:    End For
12: End if
13: Return the string restr that has been encrypted using the encrypting key
```

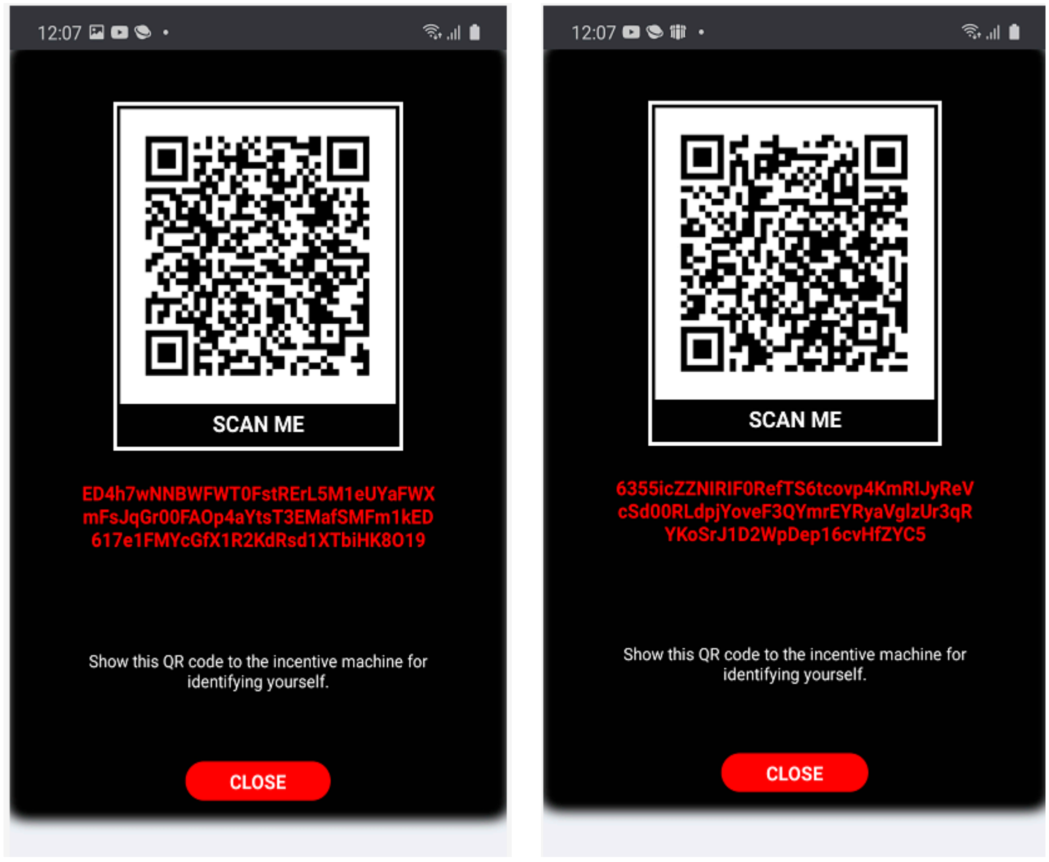
Using Algorithms 1 and 2, an example of the dynamic and secure OR code for a consumer's shopping ID is generated, as shown in Figure 2. Two QR codes in Figure 2 both refer to the same shopping ID that has been encrypted and obfuscated. They can be scanned by the shopping/incentive machines, but the string obtained from the above QR code is obfuscated and encrypted. Thus, it needs to be decrypted and de-obfuscated in the client's side programme of the shopping/incentive machines. The decryption algorithm is offered in Algorithm 3, and the de-obfuscating procedure can be easily implemented according to the rule presented in Algorithm 1 for data obfuscation and encryption.

Algorithm 3: DecryptStr

```
  Require (input parameter):
str—the string that needs to be de-obfuscated and decrypted.
key—the decrypting number that can be obtained in str
Procedure:
1: Define a string restr used for storing the decrypted string;
2: If str is not null then do
3:       For each character c in str do
4:          If character c is a lowercase then do
5:             Replace c with the key'th letter before c in the 26 lowercase alphabet
```

Algorithm 3: DecryptStr

```
6:          End If
7:          If character $c$ is a uppercase then do
8:             Replace $c$ with the key'th letter before $c$ in the 26 uppercase alphabet
9:          End If
10:         Add character $c$ to the string *restr*
11: End For
12: End if
13: Return the string *restr* that has been decrypted using the decrypting key
```



ED4h7wNNBWFWT0FstRErL5M1eUYaFWX
mFsJqGr00FAOp4aYtsT3EMafSMFm1kED
617e1FMYcGfX1R2KdRsd1XTbiHK8O19

Show this QR code to the incentive machine for identifying yourself.

6355icZZNIRIF0RefTS6tcovp4KmRIJyReV
cSd00RLdpjYoveF3QYmrEYRyaVglzUr3qR
YKoSrJ1D2WpDep16cvHfZYC5

Show this QR code to the incentive machine for identifying yourself.

Figure 2. The example of two QR codes for the same shopping ID.

3.2. Barcode Recognition

In order to retrieve the product information, the product ID should be obtained first. In the mobile application, the product ID is represented by a string, which is usually complex and difficult to remember. To simplify the communication between consumers and products, the GS1-128 barcode is adopted to represent the product ID. A detailed description of the GS1-128 barcode can be found on its webpage (it can be accessed by the URL: https://www.gs1-128.info) [30]. Obviously, scanning and recognising barcode is crucial to obtain the product ID. Currently, there are many methods to implement a barcode scanning and recognising program. For example, using Java Programming in Android Studio, adopting the API of Baidu AR or EasyAR, or making use of the Unity's AR SDK. In this mobile application, java programming language within Android Studio is applied to implement the function of barcode recognition. The detailed processes of scanning and recognising barcode are illustrated in Figure 3.

As shown in Figure 3, the implementation of barcode recognition consists of the following steps. Firstly, the mobile device's camera should be initialised and opened to scan the barcode (i.e., the product's GS1-128 barcode). Once the image of barcode is successfully obtained by calling the camera API of the mobile device, the obtained image is pre-processed and transformed to be a grayscale image. Then, the grayscale image is decoded through the decode analysis. Finally, the related information contained in the barcode, such as product ID, number or text, can be retrieved from the network or the remote database server according to the results obtained in the previous steps. Based on the barcode recognition process, the java programming language is used to develop the barcode recognition function in Android Studio. The interface of scanning barcode also has been designed. and implemented (see Figure 4).

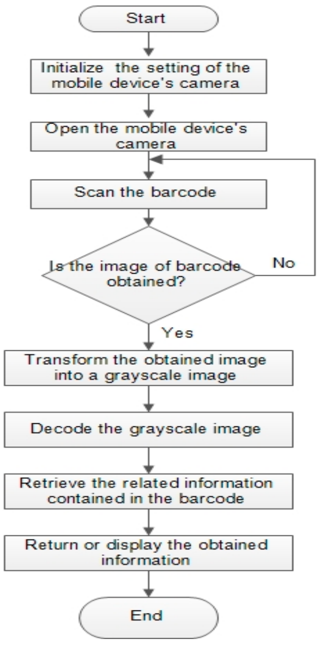

Start
Initialize the setting of the mobile device's camera
Open the mobile device's camera
Scan the barcode
Is the image of barcode obtained?
No
Yes
Transform the obtained image into a grayscale image
Decode the grayscale image
Retrieve the related information contained in the barcode
Return or display the obtained information
End

Figure 3. The process of barcode recognition.

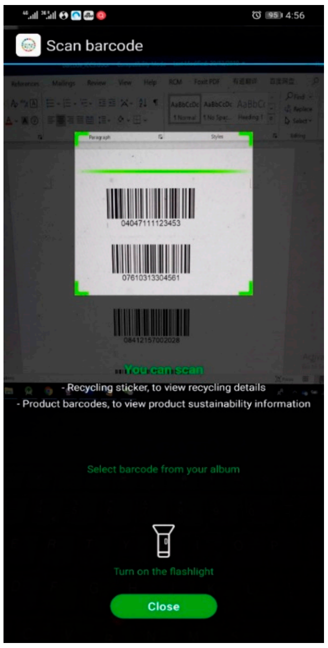

4:56
Scan barcode
You can scan
- Recycling sticker, to view recycling details
- Product barcodes, to view product sustainability information
Select barcode from your album
Turn on the flashlight
Close

Figure 4. The interface of scanning product barcode.

3.3. Communication with the Remote Server

In the mobile infrastructure based eco-accounting system, all resources and data related to consumers, products and history detail of eco-activities are stored in the database of the remote servers. To access or update the resources and data, a set of common application programming interfaces (APIs) are utilised to enable the client application to connect with the remote server over the Internet. In this research, Web Services technique is adopted to build the secure and encrypted transmission between the server and the mobile client. When consumers log in or operate the mobile client, the mobile application can initialise a connection to the remote server by sending a (GET/POST) request calling the APIs deployed in the remote server. Then the web server can handle the request through the application services to access/fetch data from the central database. After the request completion, the application services can transform the returned data into JSON messages, then the web server sends the JSON messages to response the mobile client. The process is demonstrated in Figure 5.

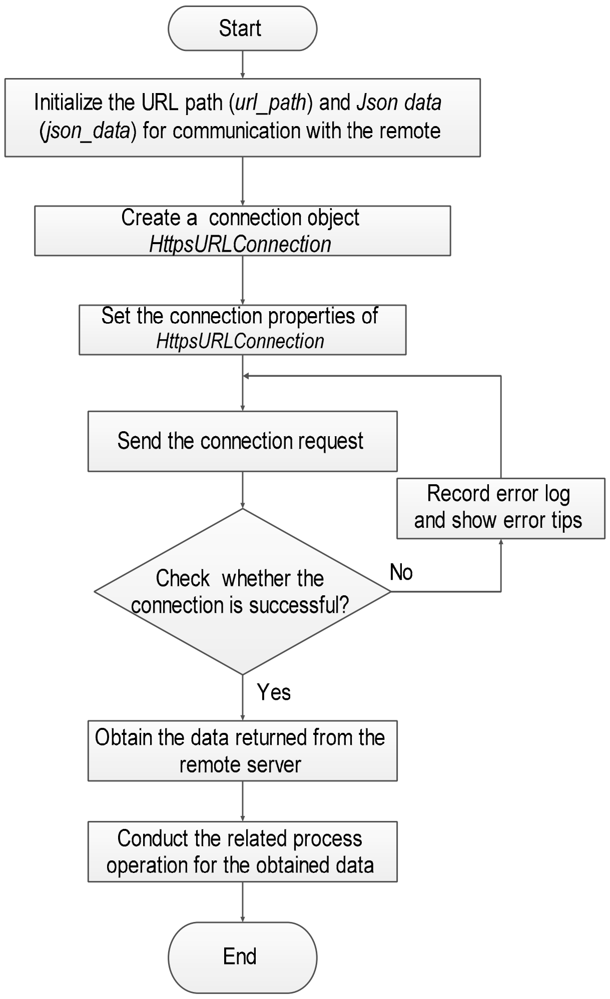

Start
Initialize the URL path (*url_path*) and *Json data* (*json_data*) for communication with the remote
Create a connection object *HttpsURLConnection*
Set the connection properties of *HttpsURLConnection*
Send the connection request
Record error log and show error tips
Check whether the connection is successful?
No
Yes
Obtain the data returned from the remote server
Conduct the related process operation for the obtained data
End

Figure 5. The process of the communication with the remote server.

As shown in Figure 5, at the starting stage of the communication, two input parameters: url_path and json_data are should be initialised. The parameter url_path denotes the absolute URL path of the target API web service, and the parameter json_data is the data with json format used to communicate the remote server, such as the user access token, the key word (i.e., product id, order id) for retrieving information from the remote server and so on. Then, a connection object HttpsURLConnection is firstly created by using the URL object with url_path. In addition, the connection properties of the connection object HttpsURLConnection, i.e., the request type, request attributes, connection timeout and reading timeout, should be set. Last, it is necessary to check whether the connection is successful. If the connection is successful, the data will be returned from the remote server, the corresponding operations for the obtained data will be launched and the Android programme will record the error log and show the error message to consumers. The detail programme code of implementing the communication with the remote server is offered in Appendix A.

3.4. Intelligent Navigation via Embedded Google Maps

It is necessary to implement a function that is used to automatically guide consumers to find the nearby available intelligent bins for eco-recycling or locate the nearby service point for eco-shopping or eco-incentive activities. Google Maps is a web mapping service, which offers satellite imagery, aerial photography, street maps, 360° interactive panoramic views of streets (Street View), real-time traffic conditions and route planning for traveling by foot, car, bicycle or transportation. Therefore, Google Maps are embedded into the mobile application with the Maps SDK for Android. The SDK automatically handles access to Google Maps servers, map display, location service and navigation, and response to user operation gestures, but how to deploy the Google Maps SDK in the Android programme and how to call the related APIs of the Google Maps SDK still need to be developed. The processes of loading Google Map are illustrated in Figure 6.

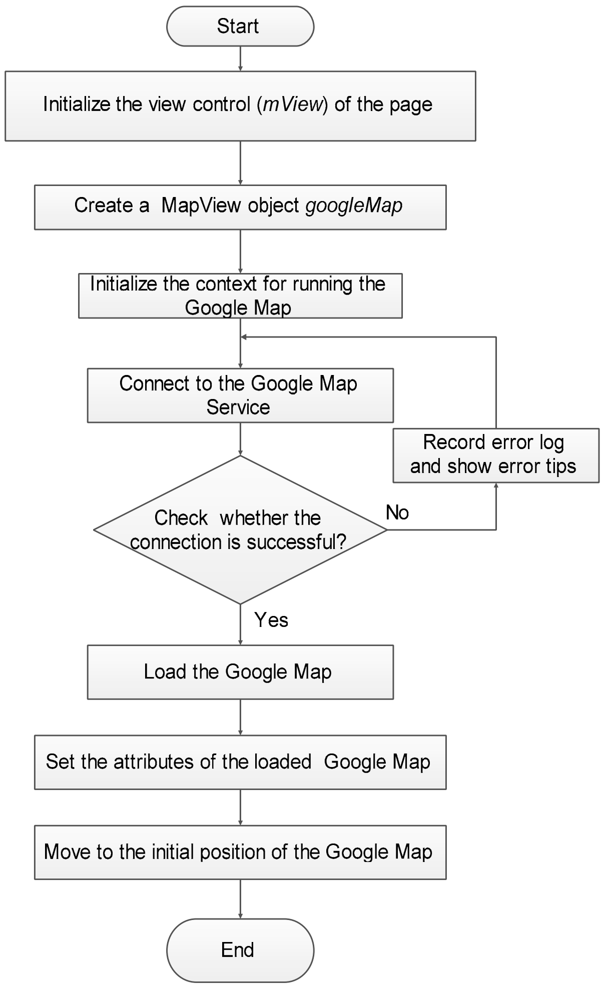

Start
Initialize the view control (*mView*) of the page
Create a MapView object *googleMap*
Initialize the context for running the Google Map
Connect to the Google Map Service
Record error log and show error tips
Check whether the connection is successful?
No
Yes
Load the Google Map
Set the attributes of the loaded Google Map
Move to the initial position of the Google Map
End

Figure 6. The process of loading Google Map.

As shown in Figure 6, the view control (mView) of the page that needs to load Google Map is firstly initialised and the map view control googleMap is created. Then, using the map view control googleMap, the procedures of running Google Map are initialised. After the initialisation is completed, the Google Map Service will be connected. After a successful connection, the setting of the loaded Google Map will be conducted in the following steps. First, the location object will be created according to the latitude and longitude of the initial position. Then, the minimum zoom size, maximum zoom size and the related operation attributes of the loaded Google Map will be set. At last, the viewpoint of the uploaded Google Map will be moved to the location of the initial position, and a tag will be added to the specified latitude and longitude. Based on the above processes (the programme code of loading Google Map can be found in Appendix B), the Google Map can be embedded to implement the intelligent navigation in the mobile application. For example, the interfaces of eco-recycling and eco-incentives with embedded Google maps are shown in Figures 7 and 8. If the connection is unsuccessful, the error log will be recorded and the error messages will be shown for users.

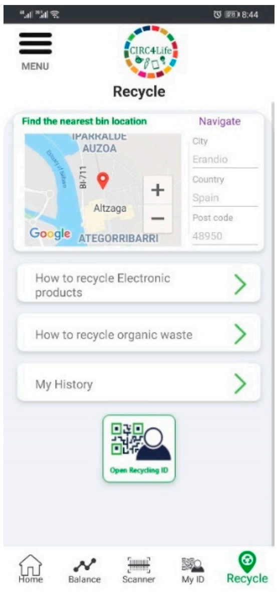


Figure 7. The interface of eco-recycling.

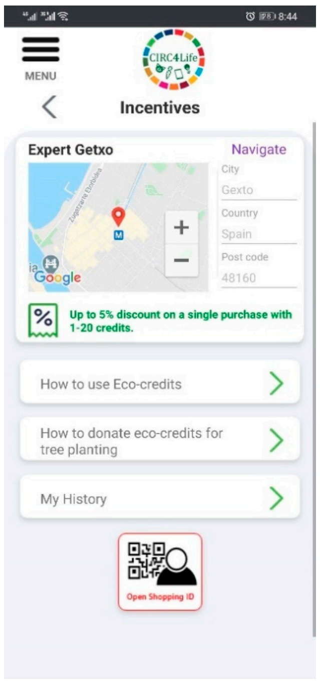


Figure 8. The interface of eco-incentives.

3.5. Automatic Scaling of User Interfaces

The mobile application is developed based on the Android platform, which is designed to run on different types of mobile devices (e.g., phones, tablets). In order to ensure the mobile application is functionally displayed on different mobile devices with different screen sizes, it should provide a flexible user interface that adapts to different screen configurations. The mobile application provides a dynamic application framework in which static files storing configuration-specific application resources (such as different XML layouts and images for different screen sizes) can be built. Then, the appropriate resources can be loaded based on users' device configuration. In other words, user interfaces in the mobile application are designed to support different screen sizes and different pixel densities. Two adaptation methods for improving user interfaces' compatibility are explained in the following sub-sections.

3.5.1. The Adaptation for Different Screen Sizes

The user interface layouts in the mobile application are designed to respond smartly for different screen sizes, instead of defining them with fixed dimensions. The following techniques have been used to support user interfaces for responding different screen sizes.

- Provide the flexible layouts for user interfaces by using view dimensions.

To provide a responsive layout for different screen sizes, the ConstraintLayout is used as the base layout, because ConstraintLayout allows programmers to specify the position and size for each view according to spatial relationships with other views in the layout. In this way, the views in the mobile application can move and stretch together with the change of the screen sizes. In addition, to ensure that the designed layouts are flexible and suitable to different screen sizes, "wrap_ content" and "match_parent" are used for the width and height of most view components, instead of hard-coded sizes, where "wrap_content" enables the view to set its size to whatever is necessary to fit the view content, and "match_parent" makes the view expand to as much as possible within the parent view.

- Provide alternative user interface layouts according to different screen configuration.

To offer a good user experience on multiple mobile devices, alternative layout resources are provided to optimise the user interface design for certain screen sizes. In the mobile application, screen-specific layouts are provided by creating additional res/layout/directories (the screen configuration that requires a different layout), and then the screen configuration qualifiers are offered to the corresponding layout directory names (i.e., the layout-w600dp for screens that have 600dp of available width). These configuration qualifiers represent the visible screen space available for user interfaces. The combination of the "smallest width" and "available width" qualifiers is used to support the size variations. In addition, fragments are used to extract the user interfaces logic into separate components in the mobile application.

3.5.2. The Adaptation for Different Pixel Densities

Different mobile device screens also have different pixel sizes. Therefore, to ensure that the mobile application can properly scale the loaded images, it is necessary to develop user interfaces for supporting different pixel densities. In the mobile application, resolution-independent units of measurements are used, and alternative bitmap resources for each pixel density are provided. The detailed descriptions are presented below.

- Use Resolution-Independent Units of Measurements.

To adapt the visible size of user interfaces on screens with different densities, the user interfaces in the mobile application use density-independent pixels (dp) as the unit of measurement. One dp is a virtual pixel unit that is roughly equal to one pixel on a medium-density screen (160 dpi is the "baseline" density). The mobile application can translate this value to the appropriate number of real pixels for each other density. For example, if a view is defined to be "300 px" wide, it will appear much larger on the device on the left. Thus, "300 dp" is used to ensure that it appears the same size on both screens. In addition, when defining text sizes, scalable pixels (sp) is used as the unit. The sp unit is the same size as dp, but it resizes based on the user's preferred text size.

- Provide alternative bitmap resources for each pixel density.

To provide good graphical qualities on mobile devices with different pixel densities for user interfaces, multiple versions of bitmap (image) should be provided, each version representing each type of density bucket. There are several density buckets available for use (see Table 2). Table 2 describes different available configuration qualifiers and what screen types that they ae applicable to. To create alternative bitmap drawables for different densities in the mobile application, the 3:4:6:8:12:16 scaling ratio among the six primary densities are utilised, and the generated images files are stored in the appropriate subdirectory under res/(drawable-xxxhdpi/ or drawable-xxhdpi/ or drawable-xhdpi/ or drawable-hdpi/ or drawable-mdpi/). The mobile application will automatically pick the correct one based on the pixel density of the device.

Table 2. Configuration qualifiers for different pixel densities.

Density Qualifier	Description
ldpi	Resources for low-density (ldpi) screens (~120 dpi).
mdpi	Resources for medium-density (mdpi) screens (~160 dpi). (the baseline density)
hdpi	Resources for high-density (hdpi) screens (~240 dpi).
xhdpi	Resources for extra-high-density (xhdpi) screens (~320 dpi).
xxhdpi	Resources for extra-extra-high-density (xxhdpi) screens (~480 dpi).
xxxhdpi	Resources for extra-extra-extra-high-density (xxxhdpi) uses (~640 dpi).
nodpi	Resources for all densities. These are density-independent resources. The system does not scale resources tagged with this qualifier, regardless of the current screen's density.
tvdpi	Resources for screens somewhere between mdpi and hdpi; approximately 213 dpi. This is not considered a "primary" density group. It is mostly intended for televisions and most apps shouldn't need it—providing mdpi and hdpi resources is sufficient for most apps and the system will scale them as appropriate. If you find it necessary to provide tvdpi resources, you should size them at a factor of 1.33*mdpi. For example, a 100 px × 100 px image for mdpi screens should be 133 px × 133 px for tvdpi.

Based on the above methods, the automatic scaling of user interfaces can be implemented in order to adapt user interfaces for different screen sizes and different pixel densities. To ensure the user interfaces correctly scale on a variety of screen sizes, the Android Emulator is used to emulate multiple screen sizes to test the mobile application.

3.6. Multi-Language Support for User Interfaces

To facilitate consumers from countries with different languages (i.e., English, Spanish, Basque, etc.) to use the mobile application, it is necessary to implement multi-language support for user interfaces. To address this, the mobile application should include resources that can be translated to the languages of the current locale (namely language-specific resources). Thus, different locales can be supported by using the resources directory in the mobile application. These resource directories (specified to one language) are used to create a res/directory in the top level of the project. These res/directories are subdirectories for various resource types. There are also a few default files such as res/values/strings.xml, which hold the default string values that are represented by English language (en locale). In the mobile application, there are some different resources files for different languages as shown in Table 3. In Table 3, for all string values in the Spanish language, the corresponding resource subdirectory (es locale) is res/values-es/strings.xml, and the res/values-bs/strings.xml is the resource subdirectory for all string values in the Basque language (bs locale).

Table 3. The resource list for different languages.

Language	Locale Name	Resource Subdirectory	Example of Texts
English	En	res/values/strings.xml	<string name="eco_account"> Eco-account </string> <string name="eco_shopping">Eco-shopping </string> <string name="eco_recycling">Eco-recycling </string> <string name="eco_recycling"> Eco-incentive </string> <string name="product_inf">Product Detail Information </string> . . . . . .
Spanish	Es	res/values-es/strings.xml	<string name="eco_account"> Eco-cuenta </string> <string name="eco_shopping"> Eco-compra </string> <string name="eco_recycling"> Eco-reciclaje </string> <string name="eco_incentive"> Eco-incentivo </string> <string name="product_inf"> Información detallada del producto </string> . . . . . .
Basque	Bs	res/values-bs/strings.xml	<string name="eco_account"> Eko-kontuaren </string> <string name="eco_shopping">Eko-erosketa </string> <string name="eco_recycling">Eko-birziklatzea </string> <string name="eco_incentive">Eko-sustagarria </string> <string name="product_inf"> Produktuari buruzko informazio zehaztua </string> . . . . . .

Using the above resources, the current mobile application version supports three languages: English, Spanish and Basque (see Figure 9). Thereby consumers can switch user interfaces supporting three languages in the mobile application.

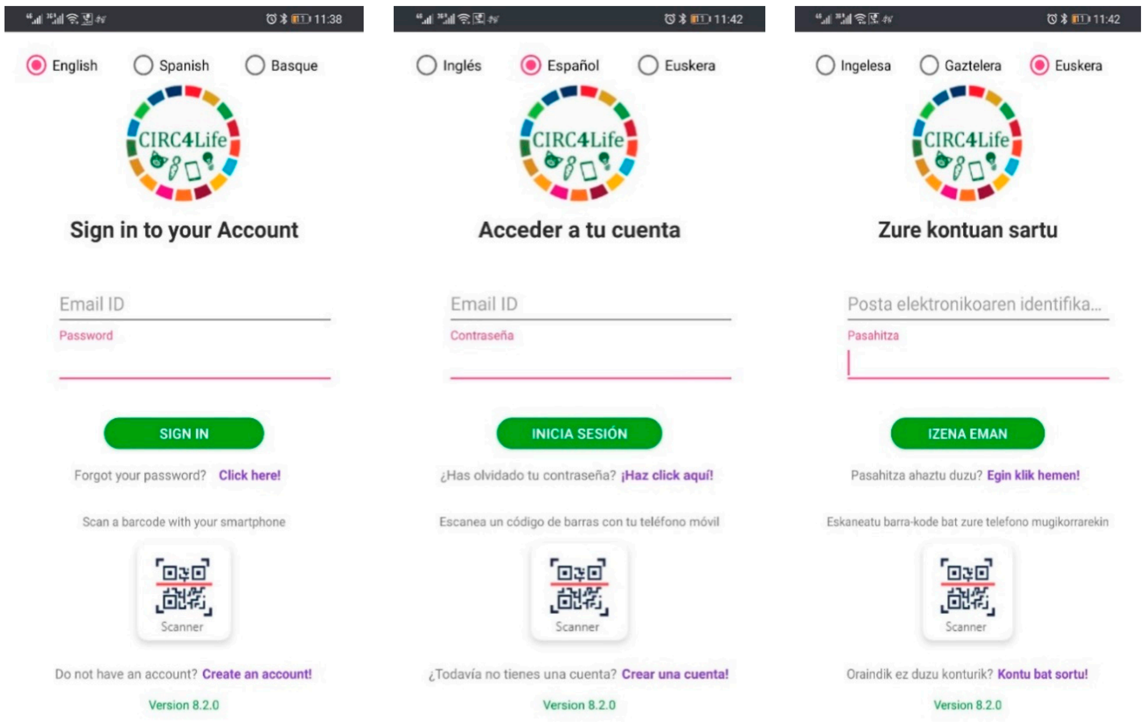

[✓ English] Spanish Basque
CIRC4Life
Sign in to your Account
Email ID
Password
SIGN IN
Forgot your password? Click here!
Scan a barcode with your smartphone
Scanner
Do not have an account? Create an account!
Version 8.2.0
Inglés [✓ Español] Euskera
CIRC4Life
Acceder a tu cuenta
Email ID
Contraseña
INICIA SESIÓN
¿Has olvidado tu contraseña? ¡Haz click aquí!
Escanea un código de barras con tu teléfono móvil
Scanner
¿Todavía no tienes una cuenta? Crear una cuenta!
Version 8.2.0
Ingelesa Gaztelera [✓ Euskera]
CIRC4Life
Zure kontuan sartu
Posta elektronikoaren identifika...
Pasahitza
IZENA EMAN
Pasahitza ahaztu duzu? Egin klik hemen!
Eskaneatu barra-kode bat zure telefono mugikorrarekin
Scanner
Oraindik ez duzu konturik? Kontu bat sortu!
Version 8.2.0

Figure 9. An example of the user interfaces supporting three languages (English, Spanish and Basque).

By using the Android Studio development platform and the above integrated mobile development technologies, the mobile application of the designed eco-accounting system has been developed and tested, which is now available in the Google Play Store.

4. Case Study

The developed mobile application has been successfully applied to support and demonstrate the proposed circular economy approaches in CIRC4Life project. A Samsung smartphone is used to demonstrate the main functions of the CIRC4Life app. The processes of applying CIRC4Life app can be divided into four phases: user register and login; scan products' barcode for obtaining eco-information to purchase products (eco-shopping); recycle products to earn eco-credits (eco-recycling); spend eco-credits to get discount on purchasing products or donate eco-credits for tree planting (eco-incentives), which are detailed in the following sections.

4.1. User Register and Login

After the CIRC4Life app installation, the user needs to first register an account. The user registration function allows consumers to create a new account by submitting the requested registration information (e.g., email ID, first name, last name and password) (see Figure 10). Provided that the registration is successful, an account related to the registered email ID will be created as a unique identifier of the user at the time of registration. Then the consumer can use the email ID to log in. Once the user's login name and password are successfully validated, the home page is shown on the mobile phone (see Figure 11).

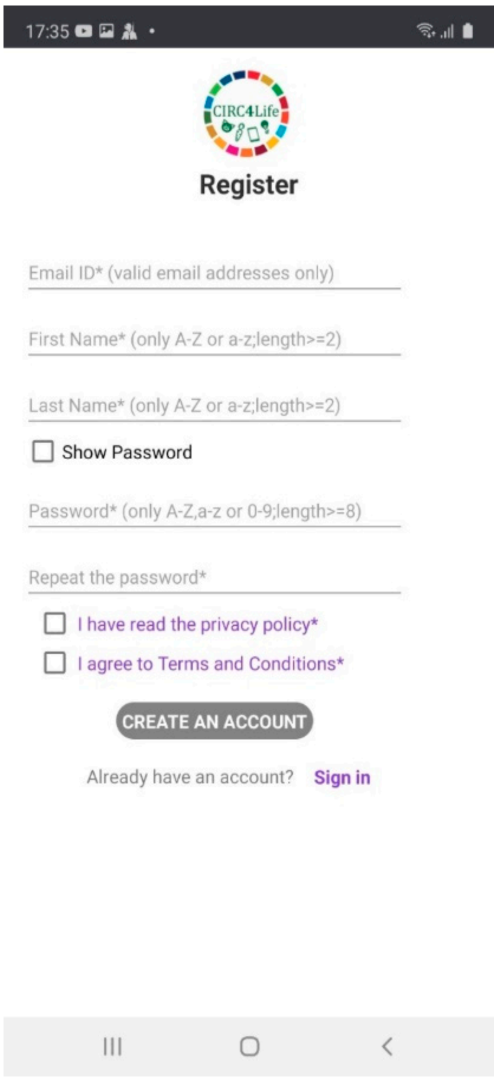

17:35
CIRC4Life
Register
Email ID* (valid email addresses only)
First Name* (only A-Z or a-z;length>=2)
Last Name* (only A-Z or a-z;length>=2)
Show Password
Password* (only A-Z,a-z or 0-9;length>=8)
Repeat the password*
I have read the privacy policy*
I agree to Terms and Conditions*
CREATE AN ACCOUNT
Already have an account? Sign in

Figure 10. The interface of user registration.

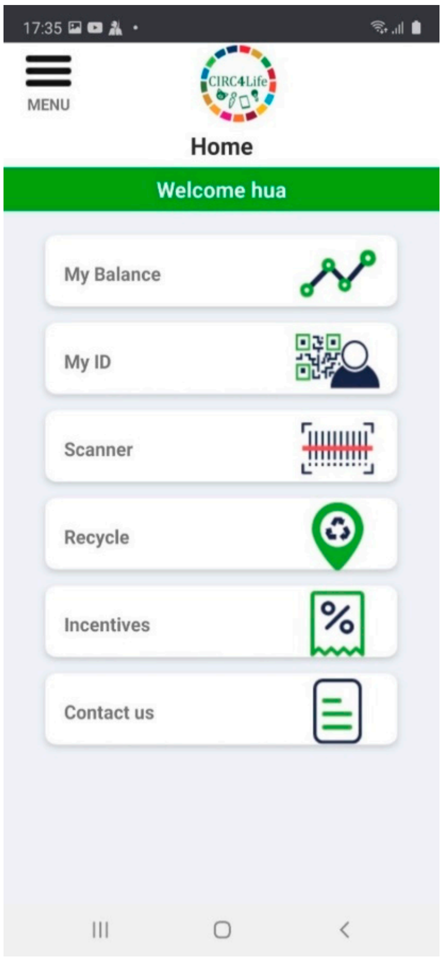


Figure 11. The home page of the CIRC4Life app.

Figure 11 shows the main menus of all functions in the mobile application. For example, the "My Balance" button is used to open the interface of eco-account information, the "Recycle" and "Incentives" buttons are linked to the interfaces of eco-recycling and eco-incentives, respectively.

4.2. Scan Products' Barcode for Obtaining Their Eco-Information to Purchase Products (Eco-Shopping)

To facilitate consumers to select environmental products with low eco-cost value, consumers are able to view the sustainable information of products, such as eco-costs and sustainable production information. Consumers can click the "Scanner" button (see Figure 11) to trigger the phone camera. The phone camera is enabled to scan the barcode (i.e., GS1-128 barcode) attached to a product (i.e., sausage) in order to get its product ID, as shown in Figure 12. Then, the obtained product ID, as an input parameter, is used to retrieve the product's detailed information from the remote database of ICT platform via REST-based web service for reading product data. Once the corresponding product information is successfully obtained, the mobile application can properly display the related eco-information on the screen as shown in Figure 13.

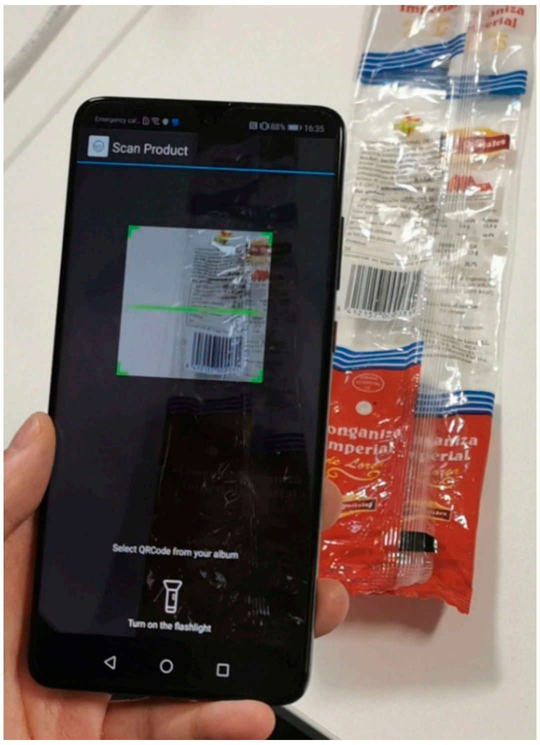
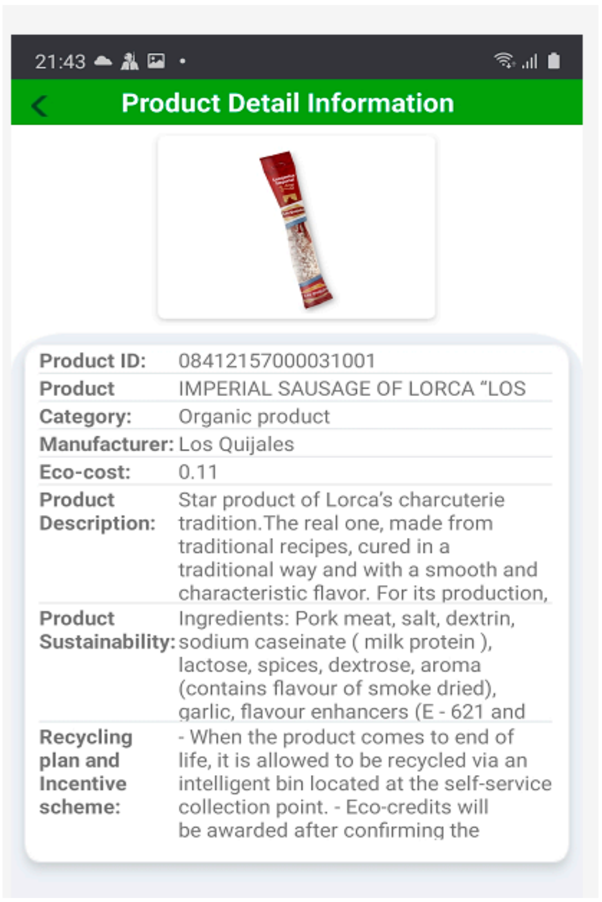


Figure 12. Scan the barcode of a product.


21:43

Product Detail Information

Product ID:	08412157000031001
Product	IMPERIAL SAUSAGE OF LORCA "LOS
Category:	Organic product
Manufacturer:	Los Quijales
Eco-cost:	0.11
Product Description:	Star product of Lorca's charcuterie tradition.The real one, made from traditional recipes, cured in a traditional way and with a smooth and characteristic flavor. For its production,
Product Sustainability:	Ingredients: Pork meat, salt, dextrin, sodium caseinate (milk protein), lactose, spices, dextrose, aroma (contains flavour of smoke dried), garlic, flavour enhancers (E - 621 and
Recycling plan and Incentive scheme:	- When the product comes to end of life, it is allowed to be recycled via an intelligent bin located at the self-service collection point. - Eco-credits will be awarded after confirming the

Figure 13. The eco-information of a product.

4.3. Recycle Products to Earn Eco-Credits (Eco-Recycling)

As shown in Figure 7 of Section 3.4, the mobile application can be used for consumers to recycle electronic products and organic waste (which is the biodegradable waste of animal or vegetable origin produced in households). By clicking the "How to recycle electronic products" and "How to recycle organic waste" buttons, consumers can view the process explanations of recycling electronic products and organic waste, respectively, and then bring EoL electronic products or organic waste to the intelligent bin for eco-recycling. Figure 14 shows the intelligent bin system for recycling EoL products, including a scanner, a printer, status buttons and a bin door. The consumer scans the QR code displayed on the mobile phone using the bin scanner, and then inputs the status of products. Subsequently, the bin prints out a barcode label, which is attached to the product in order to record the eco-credits into the consumer's eco-account. After all the operation procedures through the mobile application and the intelligent bin system, the recycling company (i.e., Indumenta, a project partner in Spain from the CIRC4Life project) will evaluate the recycled products in order to award the eco-credits. Then the eco-credits associated with the recycled products are allocated to consumers' eco-account, and consumers can check the earned eco-credits and view the history of eco-recycling activities as shown in Figure 15.

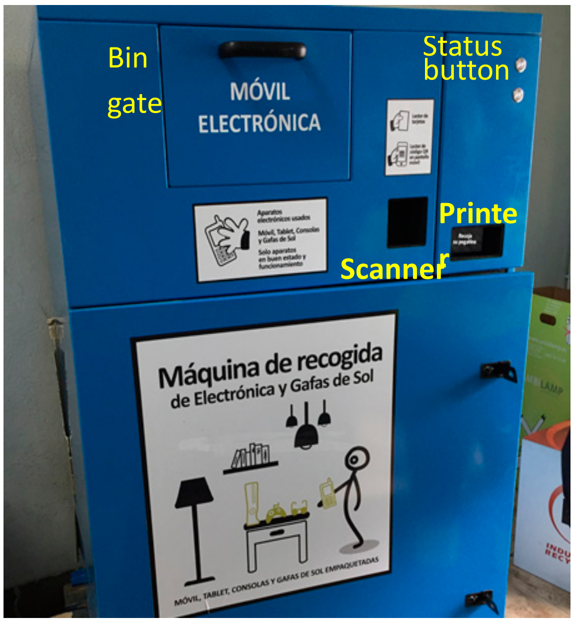
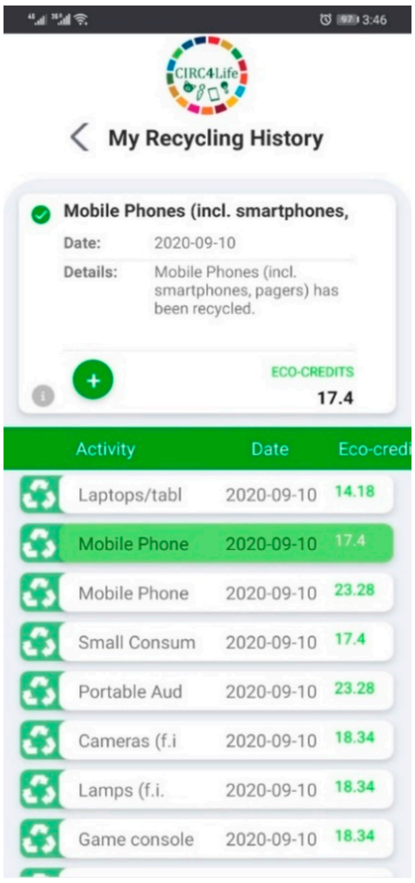


Figure 14. The Intelligent bin system.

4.4. Spend Eco-Credits to Get Discount on Purchasing Products or Donate for Planting Trees (Eco-Incentives)

As shown in Figure 8 of Section 3.4, consumers can use the obtained eco-credits to get a discount on purchasing products in stores or donate their eco-credits for a sustainable environment action. For example, consumers can redeem their earned eco-credits to plant trees in a green area of their cities, which is performed based on the designed eco-scheme. By clicking the "How to use eco-credits" button and the "How to donate eco-credits for tree planting" button in Figure 8, consumers can view the process explanations of getting incentives and donating eco-credits. The incentive system scans the consumer's identity through a barcode scanner, calculates the eco-credits equivalent to the discount of purchased products, and then prints the receipt showing the eco-account information (seed Figure 16). Once eco-credits are redeemed, the spent eco-credits are taken away from consumers' eco-account. Then, consumers can check the eco-credits spent and view the history of their eco-incentive activities (i.e., spending eco-credits to get discount for purchasing products or donating eco-credits for planting trees) by the mobile application (see Figure 17).



My Recycling History

Mobile Phones (incl. smartphones,

Date: 2020-09-10

Details: Mobile Phones (incl. smartphones, pagers) has been recycled.

ECO-CREDITS 17.4

Activity	Date	Eco-credi
Laptops/tabl	2020-09-10	14.18
Mobile Phone	2020-09-10	17.4
Mobile Phone	2020-09-10	23.28
Small Consum	2020-09-10	17.4
Portable Aud	2020-09-10	23.28
Cameras (f.i	2020-09-10	18.34
Lamps (f.i.	2020-09-10	18.34
Game console	2020-09-10	18.34

Figure 15. The interface of eco-recycling history.

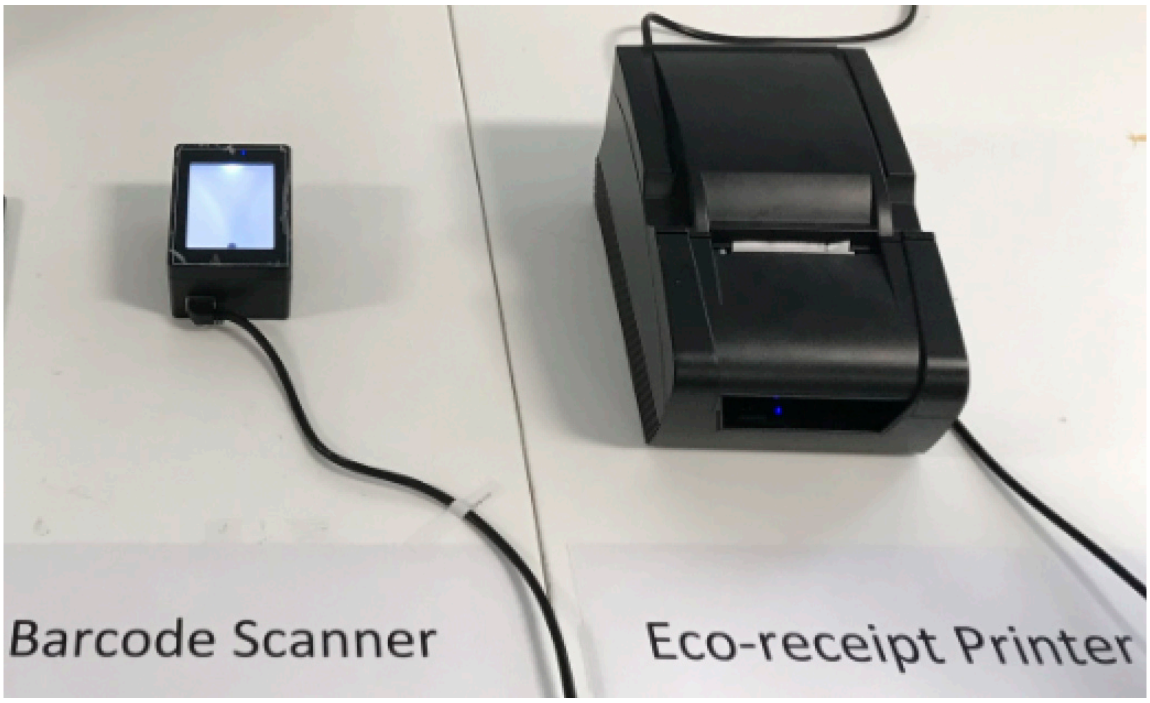
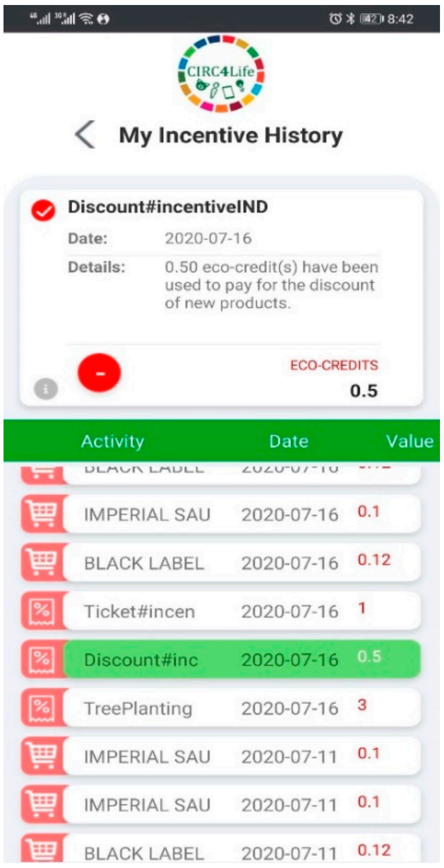


Figure 16. Eco-incentive devices.



8:42

CIRC4Life

My Incentive History

Discount#incentiveIND

Date: 2020-07-16

Details: 0.50 eco-credit(s) have been used to pay for the discount of new products.

ECO-CREDITS
0.5

Activity	Date	Value
IMPERIAL SAU	2020-07-16	0.1
BLACK LABEL	2020-07-16	0.12
Ticket#incen	2020-07-16	1
Discount#inc	2020-07-16	0.5
TreePlanting	2020-07-16	3
IMPERIAL SAU	2020-07-11	0.1
IMPERIAL SAU	2020-07-11	0.1
BLACK LABEL	2020-07-11	0.12

Figure 17. The interface of eco-incentive history.

5. Conclusions

To enhance our community's awareness and implementation of sustainability, this research has developed a mobile application, namely the CIRC4life app, which implements the eco-cost/eco-credit approach and the eco-accounting infrastructure. This paper introduces the concepts and system structure of the application, and presents the implementation of multiple mobile techniques to achieve the requested functions, followed by a case study to demonstrate the utilisation of the mobile App. In summary, this research has made the following contributions to knowledge:

- The mobile application enables sustainable consumption, which covers eco-shopping, recycling, eco-incentives and consumer eco-accounting. In comparison with existing mobile applications, the CIRC4Life app is more comprehensive and provides a systematic solution to accommodate the recording, monitoring and retrieving behaviours for eco-logical impacts. With the CIRC4Life app, the consumers' sustainable consumption can be enhanced.
- Multiple mobile development technologies, such as consumer ID verification based on dynamic and secure QR code, barcode recognition, communication with the remote server, intelligent navigation via embedded Google maps, automatic scaling of user interfaces and multi-language support for user interfaces, are utilised to enhance user experience.
- A case study is conducted, which demonstrates the procedure, effect of applying this mobile application, and proves the merits of the research. The case study confirmed that the four modules, i.e., eco-account, eco-shopping, eco-recycling and eco-incentive, have been successfully integrated, which is a novel advantage of this research.

The current version of the circ4life App only supports Android phones, and the user interfaces only support three languages: English, Spanish and Basque. In the future work, the mobile application will be developed to support the iOS operating system and more languages.

Author Contributions: Conceptualization, H.H., D.S., W.P. and Y.W.; methodology, H.H., D.S. and W.P.; software, H.H.; validation, H.H., D.S., W.P. and Y.W.; formal analysis, H.H., D.S., W.P. and Y.W.; investigation, H.H., D.S. and Y.W.; resources, H.H., D.S. and Y.W.; data curation, H.H., W.P. and Y.W.; writing—original draft preparation, H.H.; writing—review and editing, H.H., D.S., W.P. and Y.W.; visualization, H.H., D.S., W.P. and Y.W.; supervision, D.S., W.P. and Y.W.; project administration, D.S., W.P. and Y.W.; funding acquisition, D.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by European Commission H2020 Circular Economy Programme, grant agreement number 776503.

Acknowledgments: The authors acknowledge the support received from CIRC4Life project consortium members and, in particular, the ICT platform developed by WP4 team led by ICCS, traceability techniques provided by EECC, case studies conducted by ALIA, Indumetal Recycling and Recyclia, and demonstration and living lab supports provided by RISE, LAU, MMM and other partners.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. The Programme Code of the Communication with the Remote Server Using the API Web Service

Function name: CommunicationWithRemoteServer
Input parameter:
url_path ---- the absolute URL path of the target API web service
json_data ---- the data with json format used to communicate with the remote server
Output parameter: a json message object

```
public void CommunicateWithRemoteServer(String url_path, String json_data)
{     // The method is used to communicate with the remote server
      Thread t = new Thread(){ // Create a sub-thread t for connecting server
public void run()
{
         try
{
             URL url = new URL(url_path);    //Encapsulate the url_path as a URL object
             //Create a connection object to connect the remote server
             HttpsURLConnection conn = (HttpsURLConnection) url.openConnection();
             conn.setRequestMethod("POST");//Set the request type as 'POST' pattern
             // Add two lines of attributes for the POST request
             conn.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");
             conn.setRequestProperty("Content-Length", json_data.length() + "");
conn.setConnectTimeout(5000); //Set connection timeout
             conn.setReadTimeout(5000); //Set reading timeout
             conn.setDoOutput(true); // Set to open the output stream
             OutputStream os = conn.getOutputStream();//Get the output stream
             os.write(json_data.getBytes());//Use the output stream to submit json_data to the server
             if(conn.getResponseCode() == HttpsURLConnection.HTTP_OK)
{   // If the request is sent and processed successfully.
                InputStream is = conn.getInputStream();//Get the input stream in the connection object
                String text = StreamUtil.getTextFromStream(is);// Get the json-text in the input stream
                Message msg = handler.obtainMessage();//Create a message object
                msg.obj = text;   //Sets the json-data carried by the message object
                handler.sendMessage(msg);//Send the message to the message queue of the main thread
              }
              else   // If the request is not sent or processed unsuccessfully.
              {    //Record the error log
                   Log.e("Error","Fail to connect remote server! errcode:"+conn.getResponseCode());
               }
             } catch (Exception e) {
                     e.printStackTrace();     //Print the tracking inforamtion
                }
            }
}
      };
      //Start the sub-thread t
      t.start();
}
```

Appendix B. The Programme Code of Loading Google Map

Function name: LoadGoogleMap
Input parameter:
mView ----- the view control of the page that needs to load Google Map.
Output parameter: None

```
public void LoadGoogleMap(View mView)
{
    MapView googleMap = (MapView) mView.findViewById(R.id.googlemapView); //Obtain the Google
Map View control
//Prepare for loading Google Map Service
    googleMap.onCreate();
    googleMap.onResume();
    try {
        MapsInitialiser.initialise(mContext); //Initialise the context for Google Map Service
    } catch (Exception e) {
        e.printStackTrace();
    }
//Connect to the Google Map Service, and obtain the response code as the value of responseCode
    int responseCode = GooglePlayServicesUtil.isGooglePlayServicesAvailable(mContext);
    if(responseCode!=ConnectionResult.SUCCESS) //If the connection with the Google Map Service is
unsuccessful.
{
GooglePlayServicesUtil.getErrorDialog(responseCode,0).show();   //Show the error tips
    }
else //If the connection with the Google Map Service is successful.
{
        double lat = 43.305628; //Define the latitude of the initial position
        double lng = -2.974852; //Define the longitude of the initial position
        LatLng appointLoc = new LatLng(lat, lng);//Create the location object according to the latitude and
longitude of the initial position
        googleMap.setMinZoomPreference(1f);//Set the minimum zoom size of the uploaded Google Map
        googleMap.setMaxZoomPreference(20f);//Set the maximum zoom size of the uploaded Google Map
//Set the related operation attributes of the uploaded Google Map
        googleMap.getUiSettings().setAllGesturesEnabled(true);
        googleMap.getUiSettings().setZoomControlsEnabled(true);
        googleMap.getUiSettings().setZoomGesturesEnabled(true);
        googleMap.getUiSettings().setMyLocationButtonEnabled(true);
//Move the Google Map to the location of the initial position
        googleMap.moveCamera(CameraUpdateFactory.newLatLngZoom(appointLoc,14.0f));
// Add a tag to the specified latitude and longitude
        googleMap.addMarker(new MarkerOptions().position(new LatLng(lat, lng)).title("Marker")
                .draggable(true)
                .zIndex(2)
                .icon(BitmapDescriptorFactory.fromResource(R.drawable.startmark)));
    }
}
```

References

1. Orbach, T.; Liedtke, C. *Eco-management Accounting in Germany: Concepts and Practical Implementation. Final Report for the Project Management Accounting and Environmental Management: Towards the Sustainable Enterprise. A Study of Operational and Material Flow Analysis, Particular, Wuppertal Papers*; Wuppertal Institute for Climate, Environment and EnergyDivision for Material Flows and Structural Change: Wuppertal, Germany, 1998.
2. Zhou, Z.; Ou, J.; Li, S. Ecological Accounting: A Research Review and Conceptual Framework. *J. Environ. Prot.* **2016**, *7*, 643–655. [CrossRef]
3. Zhao, R.; He, H.; Zhang, N. Regional water footprint assessment: A case study of Leshan City. *Sustainability* **2015**, *7*, 16532–16547. [CrossRef]

4. Wang, C.; Shi, G.; Wei, Y.; Western, A.W.; Zheng, H.; Zhao, Y. Balancing rural household livelihood and regional ecological footprint in water source areas of the South-to-North Water Diversion Project. *Sustainability* **2017**, *9*, 1393. [CrossRef]
5. Zhang, P.; Deng, M.; Long, A.; Deng, X.; Wang, H.; Hai, Y.; Wang, J.; Liu, Y. Coupling analysis of social-economic water consumption and its effects on the arid environments in Xinjiang of China based on the water and ecological footprints. *J. Arid Land* **2020**, *12*, 73–89. [CrossRef]
6. Li, Y.; Bao, L.; Li, W.; Deng, H. Inventory and policy reduction potential of greenhouse gas and pollutant emissions of road transportation industry in China. *Sustainability* **2016**, *8*, 1218. [CrossRef]
7. Li, Y.; Wang, Y.; He, Q.; Yang, Y. Calculation and evaluation of carbon footprint in mulberry production: A case of haining in China. *Int. J. Environ. Res. Public Health* **2020**, *17*, 1339. [CrossRef] [PubMed]
8. Wu, C.; Liu, Q.; Ma, G.; Liu, G.; Yu, F.; Huang, C.; Zhao, Z.; Liang, L. A study of the spatial difference of the soil quality of the Mun River basin during the rainy season. *Sustainability* **2019**, *11*, 3423. [CrossRef]
9. Scholz, T.; Hof, A.; Schmitt, T. Cooling effects and regulating ecosystem services provided by urban trees-Novel analysis approaches using urban tree cadastre data. *Sustainability* **2018**, *10*, 712. [CrossRef]
10. Sequeira, T.; Santos, M. Education and energy intensity: Simple economic modelling and preliminary empirical results. *Sustainability* **2018**, *10*, 2625. [CrossRef]
11. Von, G.; Justus von, G.; Klaus, W.; Stephan, W.; Marie-Sophie, W.; Robert, M. A global collaborative accounting network to calculate the resource use of products and services. In Proceedings of the World Resources Forum, Davos, Switzerland, 6–9 October 2013.
12. Liedtke, C.; Bienge, K.; Wiesen, K.; Teubler, J.; Greiff, K.; Lettenmeier, M.; Rohn, H. Resource use in the production and consumption system-the MIPS approach. *Resources* **2014**, *3*, 544–574. [CrossRef]
13. Neto, G.O.; Chaves, L.E.C.; Pinto, L.; Santana, J.C.C.; Amorim, M.; Rodrigues, M. Economic, Environmental and Social Benefits of Adoption of Pyrolysis Process of Tires: A Feasible and Ecofriendly Mode to Reduce the Impacts of Scrap Tires in Brazil. *Sustainability* **2019**, *11*, 2076. [CrossRef]
14. Lukas, M.; Rohn, H.; Lettenmeier, M.; Liedtke, C.; Wiesen, K. The nutritional footprint—Integrated methodology using environmental and health indicators to indicate potential for absolute reduction of natural resource use in the field of food and nutrition. *J. Clean. Prod.* **2016**, *132*, 161–170. [CrossRef]
15. Kimura, A.; Wada, Y.; Tsuzuki, D.; Goto, S.-I.; Cai, N.; Dan, I. Consumer valuation of packaged foods. Interactive effects of amount and accessibility of information. *Appetite* **2008**, *51*, 628–634. [CrossRef] [PubMed]
16. Kimura, A.; Wada, Y.; Kamada, A.; Masuda, T.; Okamoto, M.; Goto, S.-I.; Tsuzuki, D.; Cai, D.; Oka, T.; Dan, I. Interactive effects of carbon footprint information and its accessibility on value and subjective qualities of food products. *Appetite* **2010**, *55*, 271–278. [CrossRef] [PubMed]
17. Schnettler, B.; Pardo, S.; Miranda, H.; Lobos, G.; Mora González, M.; Adasme, C. Attributes that define preferences for cheese in Southern Chile: Do consumers value information about the carbon footprint? *Revista Cientifica de la Facultad de Ciencias Veterinarias de la Universidad del Zulia* **2015**, *25*, 402–411.
18. Andersson, D. A novel approach to calculate individuals carbon footprints using financial transaction data—App development and design. *J. Clean. Prod.* **2020**, *256*, 120396. [CrossRef]
19. Barendregt, W.; Biørn-Hansen, A.; Andersson, D. Users' experiences with the use of transaction data to estimate consumption-based emissions in a carbon calculator. *Sustainability* **2020**, *12*, 7777. [CrossRef]
20. Fagerstrøm, A.; Eriksson, N.; Sigurdsson, V. Investigating the impact of Internet of Things services from a smartphone app on grocery shopping. *J. Retail. Consum. Serv.* **2020**, *52*, 101927. [CrossRef]
21. Cellina, F.; Castri, R.; Simão, J.V.; Granato, P. Co-creating app-based policy measures for mobility behavior change: A trigger for novel governance practices at the urban level. *Sustain. Cities Soc.* **2020**, *53*, 101911. [CrossRef]
22. Gu, F.; Zhang, W.; Guo, J.; Hall, P. Exploring "Internet+Recycling": Mass balance and life cycle assessment of a waste management system associated with a mobile application. *Sci. Total Environ.* **2019**, *649*, 172–185. [CrossRef] [PubMed]
23. reGAIN app (n.d). Available online: https://play.google.com/store/apps/details?id=com.regainapp&hl=en (accessed on 23 October 2020).
24. Golsteijn, L. Updated Impact Assessment Methodology ReCiPe2016. Available online: https://simapro.com/2017/updated-impact-assessment-methodology-recipe-2016/ (accessed on 2 July 2018).
25. CIRC4Life (n.d.), CIRC4Life: An Circular Economy Approach for Lifecycles of Products and Services. Available online: http://www.circ4life.eu (accessed on 23 October 2020).

26. CIRC4Life Project Deliverable 2.4, Eco-Credits Method Final Definition. Available online: https://25cd04c9-5fc8-4b44-8c3c-9ad39fc8bbac.usrfiles.com/ugd/25cd04_4f1880e7455146dc9b4e6f4c7e184c24.pdf (accessed on 2 November 2020).
27. Report on Information Logistics Systems Development and Resulting Systems and Processes. Available online: https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5c8d17a8e&appId=PPGMS (accessed on 3 September 2020).
28. Su, D.; Peng, W. *Eco-Accounting Infrastructure, in Sustainable Product Development: Tools, Methods and Examples*; Springer International Publishing: Cham, Switzerland, 2020; pp. 73–84. [CrossRef]
29. Peng, W.; Wu, Y.; Su, D. Application of Information and Communication Technologies for Eco-Accounting. In *Sustainable Product Development: Tools, Methods and Examples*; Su, D., Ed.; Springer: Berlin/Heidelberg, Germany, 2020; pp. 85–126.
30. Resources and Education. Available online: https://www.gs1.org/standards/barcodes/1d-general-distribution (accessed on 20 September 2019).