

Development of Information Management System for Food Donation Distribution

Thivyanath A/P Muniandy¹, Nureize Arbaiy^{1*}

¹Faculty of Computer Science and Information Technology,
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

DOI: <https://doi.org/10.30880/aitcs.2022.03.02.085>

Received 16 June 2022; Accepted 02 November 2022; Available online 30 November 2022

Abstract: This paper describes the development of a food donation distribution system that aims to improve the information management process for donation and distribution to those in need by utilizing information technology and databases. Non-governmental organizations (NGOs) currently maintain track of donations using Excel spreadsheets. It is highly susceptible to human error. Because spreadsheets are open to manipulation and are not secured, there is a risk of data corruption. As a result, a food donation distribution information management system is proposed to address the issues while also providing a means to improve the efficiency of the current management process. This system was created using Visual Studio Code, XAMPP, and the Microsoft Windows 10 operating system. The PHP programming language was used to create the system. The prototype technique is utilized to oversee project development implementation. The outcomes of the study will have a positive impact on commonly used procedures, particularly in the process of NGOs managing food donation information.

Keywords: Food Donation, Distribution System, Information Management System, Charity

1. Introduction

A charitable donation is a gift of cash or property made to a nonprofit organization to help it accomplish its goals for which the donor receives nothing of value in return [1]. A non-governmental organization (NGO) is a non-profit, citizen-led organization that operates independently of government yet may engage in worldwide charitable, developmental, or social missions [2]. Non-profit hunger relief organizations strive to counteract the negative effects of food insecurity by providing food and services to individuals in need. These organizations rely on the goodwill of donors, that provide cash or in-kind food donations, to help them meet their objectives.

Food distribution management to the needy is a type of distribution system that charities are able to keep track of, from donations made by donors to the needy that receives at the end [3]. Since the world is stricken by the covid-19 pandemic, impoverished people look for alternative ways to get their

necessities and thus, many non-profit organizations were introduced. Citizens came forward to help those in need by donating foods and other essentials to these charities. So, when hundreds of donations are made, any records in these charities can get lost and disorganized. It makes an information system necessary to manage the business records. Information systems are business applications on computers that perform data processing and consist of databases, application programs and manual procedures, and machines [4] Therefore, the food distribution management system is needed to keep the donation data organized and each process is conducted smoothly and efficiently.

Additionally, Kencahara Soup Kitchen, a food charity organization is having problems manually handling the data on Excel when the donations are made. Since spreadsheets are vulnerable to manipulation and are not adequately protected, there is a risk of data corruption. Another issue is that inputting data into the spreadsheet takes a long time because it is done by hand. For a variety of reasons, the storage and administration of these data is critical. Data storage efficiency, data retrieval, report generating, and decision making are examples of these. Hence, a food donation distribution information management system is presented to address the issues that arise while also providing an option for improving the efficiency of the current management process. The system will be designed so that each perishable and non-perishable food item, as well as other necessities, will have its own section. Further, for administrators and volunteers, the system will have two separate login and registration modules. This allows the administrator to log in and monitor any changes to the records. Next, the system will have a more user-friendly interface that will make the data entry process easier and reduce the risk of data errors

The remaining of this paper proceeds as follows. Section 2 presents the related works. Section 3 explains the research methodology; Section 4 presents and discusses the results of system analysis and finally Section 5 concludes the study.

2. Related Work

Food charities are a non-profit organization that collects and distributes food to those in need. Consumers receive food at no cost from food banks and smaller food pantries. Receiving, holding, storing, packaging, repackaging, and distributing food for consumption off-site are just some of the services provided by them. Moreover, they are part of a variety of organizations and programs that use donated food to feed the hungry, promote better eating, build community capacity, educate and train people, or preserve cultural eating habits. Food charities hope to enhance access to food for those in need by engaging in these activities. They offer a major amount of people's meals, especially children, making nutrition and food quality increasingly crucial aspects of the services they provide. Food charities are often run on a shoestring budget, rely on volunteers and contributions, and have limited ability to receive, store, and prepare various foods [5].

Charities may be found in almost every society, and their rise as an economic force is extensively documented. As a result, the charitable organizations' survival is dependent on contributors and funders. Nonetheless, NGOs occasionally provide a hand in such situations. In general, there are three sorts of NGOs: charities, voluntary welfare organizations, and public institutions. A registered charity of the NGO will be an organization with objectives and activities that fall into one of these four categories (poor relief, education advancement, religious advancement, and other community-beneficial purposes) [6].

Good data management is vital for any process nowadays. Data management is the activity of gathering, organizing, protecting, and preserving an organization's data for it to be evaluated for business decisions. So, it is important as it is an inventory of knowledge, skills, and methods usage, that helps to fulfil the wishes of projects performers and consumers. Data management is known for its planning processes, where creating and acknowledging of the schemes are necessary to perform the

goals of the project [7]. Thus, a good food donation distribution must have an effective planning scheme to guarantee the effective usage of the resources.

Web-based system have seen a huge increase in popularity in recent years due to its many advantages. Consequently, web-based system plays an important role in this case study. Web-based system is known to provide access to real-time information. For example, the current method in recording donation inventory is done manually using papers and spreadsheet. Thus web-based system that offers online reports and digital dashboards can replace paper reports, which might take days to compile and deliver. This can improve the quality and the speed of decision making. Therefore, the new system developed in this project uses web-based technology to improve the functionality of the existing system and optimize the advantages provided by this web technology.

The study of related works is essential for developing a good system. The system assessment might supply developers with related information that will be extremely valuable while constructing the system without faults. Three additional inventory systems were considered for comparison. Inventory Storage Management App, Simple Inventory Control, and Pocket Inventory Manager are some of the existing systems. Thorough observation was carried out on similar systems, there are some differences and similarities identified. **Table 1** shows a clear comparison between similar systems and systems that will be developed.

Table 1: Comparison analysis among similar systems

System Features	Inventory Storage Management App	Simple Inventory Control	Pocket Inventory Manager	Food Donation Distribution Information Management System
Login and Registration Module	No	No	No	Yes
Inventory Module	Yes	Yes	Yes	Yes
Report generator	No	Yes	No	No
Internet Requirement	No	Yes	No	No
Platform	Web-based	Web-based & Android application	Android application	Web-based

Based on **Table 1**, all programs have various types of functional modules but have the same module which is inventory. All of the system is capable of listing out every item that is in the registry for users to look through. Meanwhile, Food Donation Distribution Information Management System is the only program that has a login and registration module where the other systems do not. Additionally, Simple Inventory Control needs an internet coverage in order for the system to run whereas the other system runs under no internet connection. Finally, Food Donation Distribution Information Management System and Inventory Storage Management App is a web-based system while Pocket Inventory Manager runs on an android application. Simple Inventory Control can run on both on the web and on an android app.

3. Methodology

Prototyping is a software development method where after each iteration, the prototype is presented to the user for evaluation. Users will test and evaluate the prototype and determine whether the prototype is developed according to their requirement or need to do some modification. The modification of the prototype will depend on the feedback given by the tester [8]. The phases used in this model are planning phase, prototyping phase which includes requirements analysis, system design, developing prototype, testing and review and finally implementation phase.

3.1 Planning Phase

The representative of the charity, Senior Public Relations Executive, Khong Jien Howe is interviewed via email to discuss the module, functions and other requirement that is needed in the system as well as the problems that is caused when using the present system. This is to ensure a quality system is produced. The method to gather this information is via a questionnaire through email.

3.2 Analysis Phase

In the analysis phase, the product and equipment necessary for the proper completion of the project are dissected based on the requirements. Functional, non-functional requirements, flowchart, use case and class diagram are produced in this phase. The functional requirements are shown in **Table 1**, whereas the non-functional requirements are shown in **Table 2**. **Table 3** lists the user requirements.

Table 1: Functional Requirements of The Developed System

No	Module	Description
1.	Register and Login Module	<ul style="list-style-type: none"> The system should allow administrator to register new users with respective email and passwords. The system should allow user to login into the system using registered email and password. The system should only allow a user to log in as their respective roles with a valid email and password. The system should redirect user to that respective main menu upon successful login.
2.	Inventory Management Module	<ul style="list-style-type: none"> The system should display all the items in the inventory. The system should allow users to add new item into the inventory list, modify, categorize and delete the items The system should allow the items to be sorted alphabetically, date added and expiring items to be on the top of the list.
3.	Donor Module	<ul style="list-style-type: none"> The system should display all the existing donors with their respective information. The system should allow users to add new data into the donor module, modify and delete the data. The system should allow all the donor's profile to be sorted alphabetically
4.	Donation Management Module	<ul style="list-style-type: none"> The system should allow user to contribute new data to the food recipient module, as well as change, categorize, and remove existing data. The system should allow all the food recipient's profile to be sorted alphabetically or chronologically. The system should allow user to key in data whether the food recipient has successfully received their food The system should allow user to print out a receipt that contains all the item that is received by a food recipient.

Table 2: Non-Functional Requirements of The Developed System

No	Requirements	Description
1.	Performance	The system should be usable at all times
2.	Compatibility	The system should function well when other applications are running
3.	Security	The system should be protected to prevent unauthorized access
4.	Reliability	The system should operate well even after extensive use

Table 3: User requirements of The Developed System

No.	User Requirements
1.	All users must be able to enter a valid id and password to enter the system
2.	Administrators should be able to create new accounts for new Volunteers.
3.	Administrators and Volunteers should be able to see all the items in the item inventory
4.	Administrators and Volunteers should be able to add, modify and delete the items in the inventory
5.	Administrators and Volunteers should be able to sort the inventory list alphabetically, date added and expiring items to be on the top of the list.
6.	Administrators and Volunteers should be able to see all the existing donors with their respective information
7.	Administrators and Volunteers should be able to add new data into the donor module, modify and delete the data
8.	Administrators and Volunteers should be able to sort all the donor's profile alphabetically or by the date added.
9.	Administrators and Volunteers should be able to see all the existing food recipients that have been registered in the system
10.	Administrators and Volunteers should be able to contribute new data to the donation management module, as well as change and remove existing data.
11.	Administrators and Volunteer should be able to sort all the food recipient's profile alphabetically
12.	Administrators and Volunteer should be able to key in data whether the food recipient has successfully received their food.
13.	Administrators should be able print out donation data.

Figure 1 showcases the correlation between the users of the system, volunteer and administrator with the system modules. There are a total of four modules.

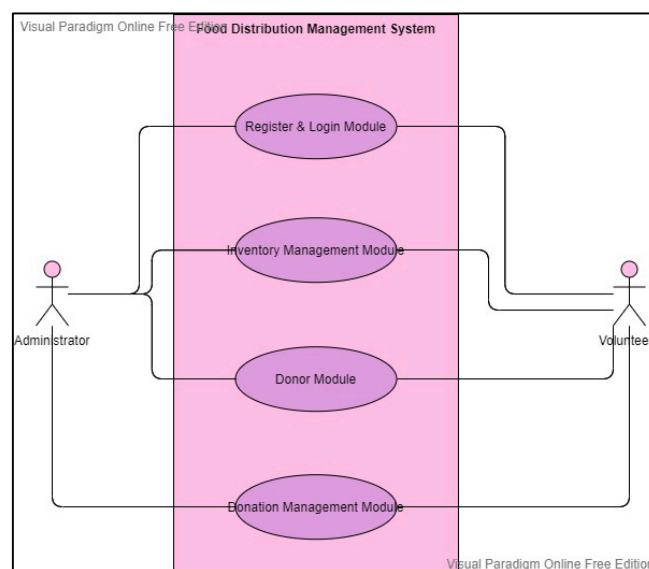


Figure 1: Use Case Diagram of the system

Figure 2 shows the class diagram for the system where there are a total of seven classes each with their corresponding attributes and operations. **Figure 3** shows the flowchart of the system. Based on **Figure 3**, once a user has a valid login account, they can proceed to login into the system and if the authentication is successful, user will be redirected to dashboard or else receive an error message and go through the process again. Then, user will proceed to inventory management where they can make the decision to either add, delete or modify item to be recorded into the inventory. If they do, they will then proceed to the end of the flowchart. Users can option to proceed from inventory management module to the donor or donation management module where the same process will take place till they end to the last point of the system.

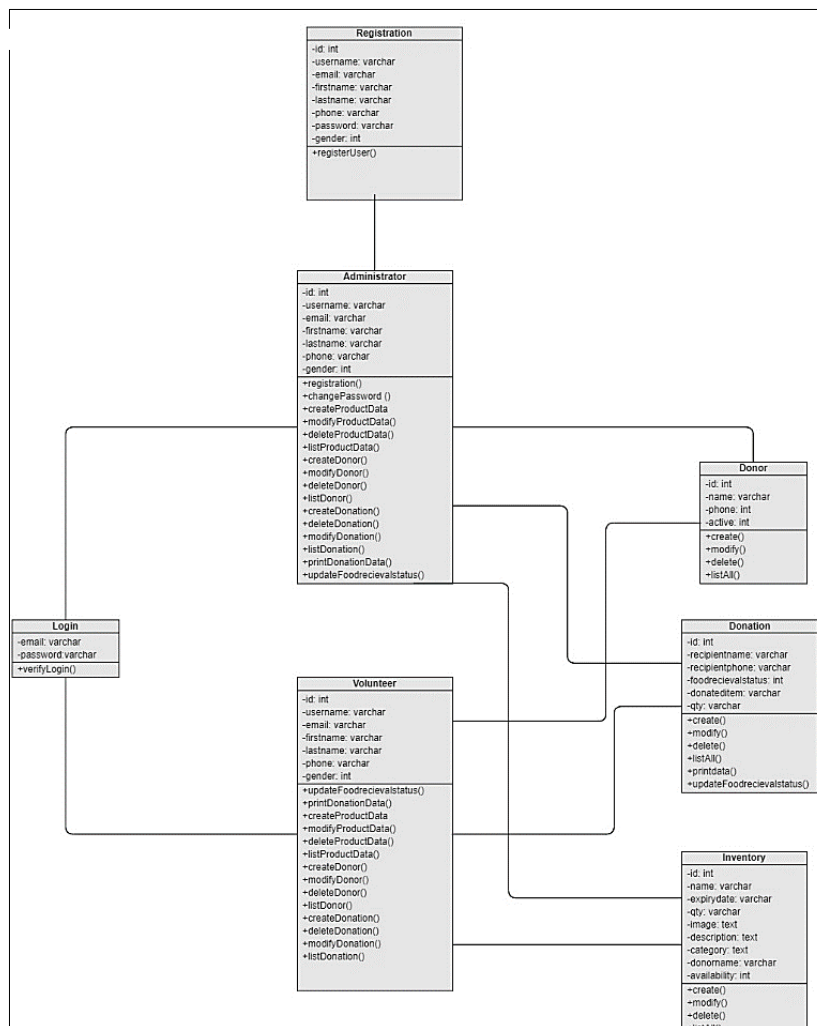


Figure 2: Class Diagram of the system

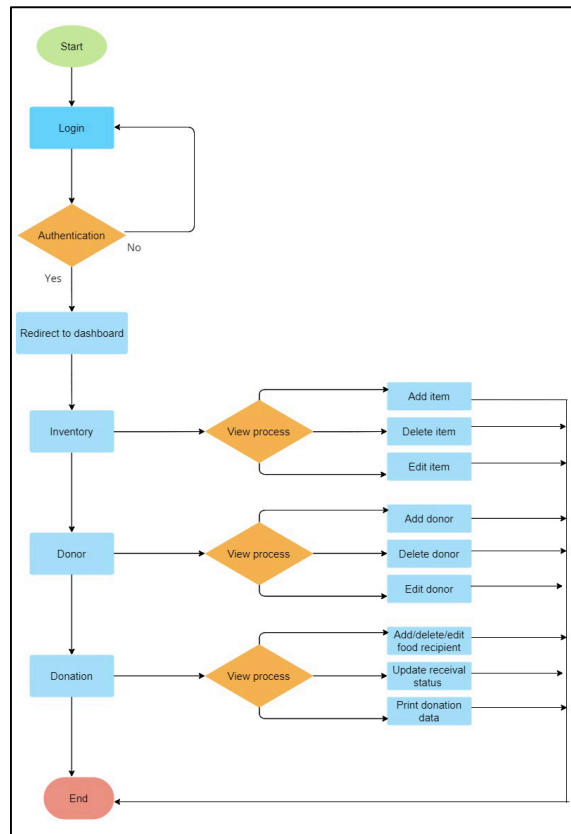


Figure 3: Flowchart of the system

3.3 Design Phase

The overall of system design is produced in this phase. Data dictionaries and database schema will be produced. The user interface of the system will be sketched quickly throughout the design process. Moreover, the physical characteristics of the system are specified, and a detailed design is prepared.

Figure 4 shows the system architecture. User will go through the web interface where the modules are settled, which are the registration & login module, inventory management module, donor module and finally donation management module. The modules' data are linked to database where all the data is ultimately stored.

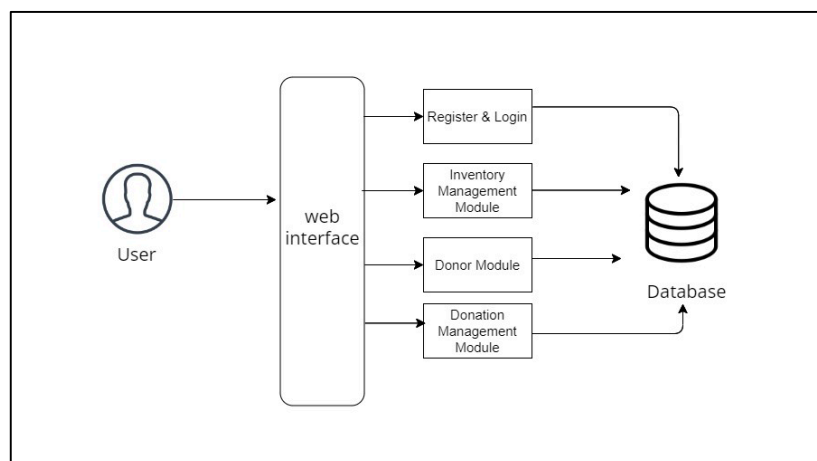


Figure 4: System architecture

Relational schema for this system is listed in the following:

- i. Administrator (id, username, email, firstname, lastname, phone, gender)
- ii. Inventory (id, name, expirydate, qty, image, description, category, donorname, availability).
- iii. Donor (id, name, phone, active)
- iv. Donation (id, recipientname, recipientphone, foodrecievalstatus, donateditem, qty)
- v. Volunteer (id, username, email, firstname, lastname, phone, gender)
- vi. Registration (id, username, email, firstname, lastname, phone, gender, password)
- vii. Login(email, password)

Figure 5 shows donation inventory interface where it will list all the items donated in a systematic list. The table will contain the name of the item, quantity available in stock, expiration date, if the item is perishable or non-perishable or if the item is not food and the date the item was added to inventory. The interface will consist of add and delete buttons as well as sort buttons by drop down where users can sort the list alphabetically or by date added. Added search bar for users to search easily by entering any keyword. Finally, the number of items in the inventory will be displayed at the top of the page.

Figure 6 shows food recipient where it will display food recipients. Profiles with photos can also be viewed. For easier navigation between profile pages, numbering is included at the bottom of the page. The interface will include add and delete buttons, as well as drop-down sorting that allows users to sort the list alphabetically or by date added. A search bar has been added to allow users to easily find anything by entering any keyword. Finally, at the top of the page, the total number of food recipients will be displayed.

Figure 7 shows food receival status interface where it will be displayed with name, age and a toggle button to indicate whether a particular food recipient has received their food. Profiles with photos can be viewed as well. Numbering is included at the bottom of the page for easy navigation across profile pages. A search bar has been introduced to allow users to quickly find anything by entering any keyword. Finally, the total number of food recipients who have received their food will be presented at the top of the page.

#	Name	Quantity	Exp. Date	Perishable/No...	Date Added
1	Can sardines	3	24/5/2022	Non-perishable	13/12/2021
2	Jar peanut b...	4	2/5/2022	Non-perishable	21/2/2021
3	Jar Pasta sa...	3	18/7/2023	Non-perishable	2/5/2022
4	Pantene sha...	6	13/7/2025	Non-food	2/3/2021

Figure 5: Donation Inventory User Interface

Figure 6: Food Recipient User Interface

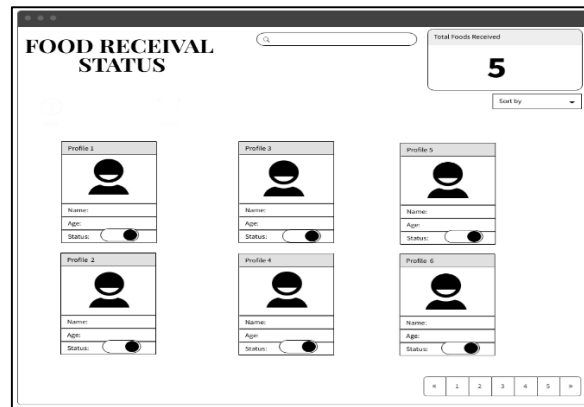


Figure 7: Food Receival Status User Interface

3.4 Development Phase

PHP programming language will be used to write the system's code. Database will be set up and linked to the user interface of the system. The system's prototype will be developed according to the information gathered requirement analysis phase and system design phase. The sub activity is to develop the prototype system based on previous phases.

3.5 Testing Phase

Testing will commence when the prototype is complete. With appropriate technique, the system's features are tested. Additionally, client will give feedback to review what the system lacks and how it feels. Thus, the project will return to requirement analysis phase to create a new prototype based on the client's review or if any errors are found within the system.

3.6 Implementation Phase

In this final phase, the final system is developed based on the final prototype and tested as a whole. The client will test the system and it is ready to use.

4. Results and Analysis

System implementation process and system testing is taken place for food donation distribution information system and the results are recorded in this section.

4.1 System Implementation

System implementation involves all the processes such as source code of the system development.

4.1.1 Registration and Login module

Figure 8 and **Figure 9** shows show the server-side coding and the user interface of the account registration. This is where administrators create new account for volunteers. **Figure 10** and **Figure 11** show the server-side coding and the user interface for logging into the system. Each user is only capable of logging into the system if they have valid credentials.

```

23 <?php if($this->session->flashdata('success')): >
24 <div class="alert alert-success alert-dismissible" role="alert">
25 <button type="button" class="close" data-dismiss="alert" aria-label="Close"><span aria-hidden="true">✕</span></button>
26 <?php echo $this->session->flashdata('success'); >
27 </div>
28 <?php elseif($this->session->flashdata('error')): >
29 <div class="alert alert-error alert-dismissible" role="alert">
30 <button type="button" class="close" data-dismiss="alert" aria-label="Close"><span aria-hidden="true">✕</span></button>
31 <?php echo $this->session->flashdata('error'); >
32 </div>
33 <?php endif; >
34
35 <div class="box">
36 <div class="box-header">
37 <h3 class="box-title">Add Volunteers</h3>
38 </div>
39 <form role="form" action="{php base_url('users/create')}> " method="post">
40 <div class="box-body">

```

Figure 8: User registration source code

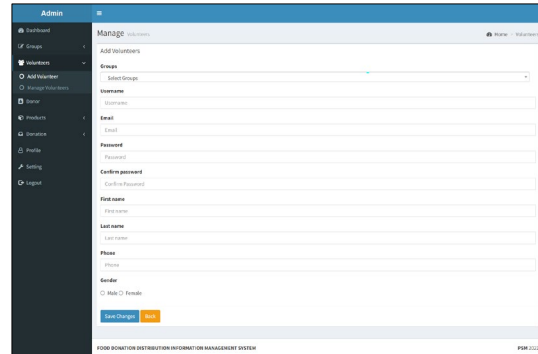


Figure 9: User registration

```

48 <form action="{php echo base_url('auth/login')}> " method="post">
49 <div class="form-group has-feedback">
50 <input type="email" class="form-control" name="email" id="email" placeholder="Email" autocomplete="off">
51 <span class="glyphicon glyphicon-envelope form-control-feedback"></span>
52 </div>
53 <div class="form-group has-feedback">
54 <input type="password" class="form-control" name="password" id="password" placeholder="Password" autocomplete="off">
55 <span class="glyphicon glyphicon-lock form-control-feedback"></span>
56 </div>
57 <div class="row">
58 <div class="col-xs-8">
59 <div class="checkbox icheck">
60 <input type="checkbox"> Remember Me
61 </div>
62 </div>
63 </div>
64 <div class="col-xs-4">
65 <input type="submit" class="btn btn-primary btn-block btn-flat">Sign In</button>
66 </div>
67 </div>
68 </div>
69 </div>
70 </div>
71 </div>
72 </form>

```

Figure 10: User login source code

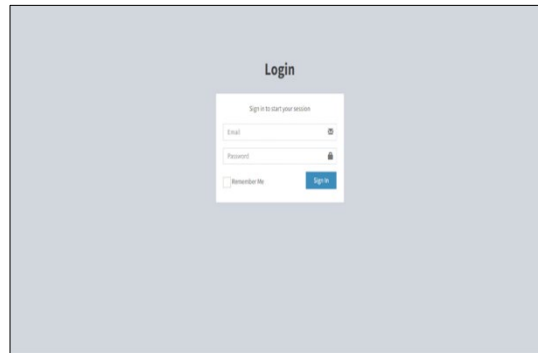


Figure 11: Login

4.1.2 Inventory module

Figure 12 and Figure 13 shows show the server-side coding and the user interface for inventory module. This where any product data that has been donated by donors are recorded into the inventory with their respective particulars.

```

38 public function fetchProductData()
39 {
40 $result = array('data' => array());
41
42 $data = $this->model_products->getProductData();
43
44 foreach ($data as $key => $value) {
45
46 $store_data = $this->model_stores->getStoreData($value['store_id']);
47 // button
48 $buttons = "";
49 if(in_array('updateProduct', $this->permissions)) {
50 $buttons .= "<a href='{base_url('products/update')}>".$value['id']."'> class='fa fa-pencil'</a>";
51 }
52
53 if(in_array('deleteProduct', $this->permissions)) {
54 $buttons .= "<button type='button' class='btn btn-default' onclick='removeFunc('{base_url('products/delete')}>".$value['id']."'> data-toggle='modal' data-target='#deleteProduct'>";
55 }
56
57 $img = '';
58
59 $availability = ($value['availability'] == 1) ? '<span class="label label-success">Active</span>' : '<span class="label label-warning">Low</span>';
60
61 $qty_status = "";
62 if($value['qty'] <= 30) {
63 $qty_status = '<span class="label label-warning">Low</span>';
64 } else if($value['qty'] <= 0) {
65 $qty_status = '<span class="label label-danger">Out of stock</span>';
66 }
67 }

```

Figure 12: Recording product data source code

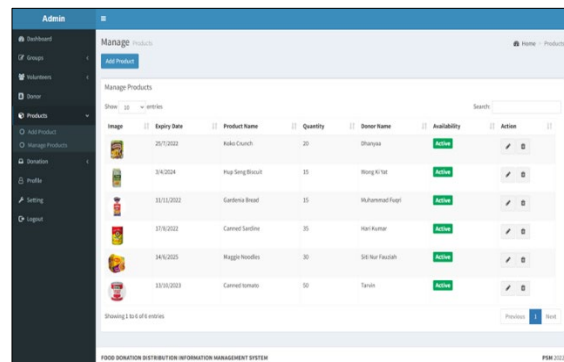


Figure 13: Listed product data

4.1.3 Donor module

Figure 14 and Figure 15 shows show the server-side coding and the user interface for donor module. This where donor data is recorded with their own particulars.

```

75 <?php if(in_array('createstore', $user_permission)): ?>
76 <!-- create brand modal -->
77 <div class="modal fade" tabindex="-1" role="dialog" id="addModal">
78 <div class="modal-dialog" role="document">
79 <div class="modal-content">
80 <div class="modal-header">
81 <button type="button" class="close" data-dismiss="modal" aria-label="Close"><span aria-hidden="true"></span></button>
82 <h4 class="modal-title">Add Donor</h4>
83 </div>
84 <div class="modal-body">
85 <form role="form" action="{?php echo base_url('stores/create')} ?" method="post" id="createform">
86 <div class="form-group">
87 <div class="form-group">
88 <input type="text" class="form-control" id="store_name" name="store_name" placeholder="Enter Donor name" autocomplete="off">
89 </div>
90 <div class="form-group">
91 <input type="text" class="form-control" id="store_name" name="store_name" placeholder="Enter Donor name" autocomplete="off">
92 </div>
93 <div class="form-group">
94 <input type="text" class="form-control" id="store_name" name="store_name" placeholder="Enter Donor name" autocomplete="off">
95 </div>
96 <div class="form-control" id="store_name" name="store_name" placeholder="Enter Donor name" autocomplete="off">
97 </div>
98 </div>
99 </div>
100 </div>

```

Figure 14: Creating a new donor source code

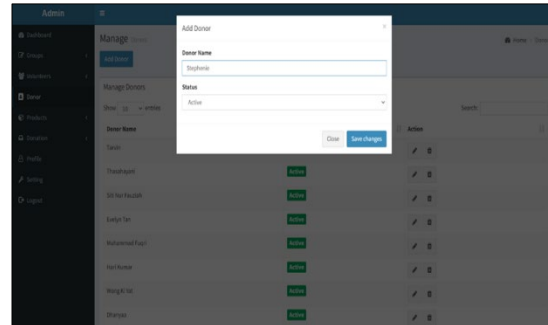


Figure 15: Adding new donor

4.1.4 Donation management module

Figure 16 and Figure 17 shows show the server-side coding and the user interface for donation management module. This where donation data is recorded where user can key in the food recipient information, update food receipt status and finally print out that information.

```

37 <?php if(in_array('createorder', $user_permission)): ?>
38 <!-- create order modal -->
39 <div class="modal fade" id="addOrderModal" role="dialog" style="display: none;">
40 <div class="modal-dialog" style="width: 50%; margin: auto;">
41 <div class="modal-content" style="background-color: #f9f9f9; border: 1px solid #ccc; padding: 10px;">
42 <div class="modal-header" style="background-color: #f9f9f9; border: 1px solid #ccc; padding: 5px 5px 0 5px;">
43 <h4 style="margin: 0; font-size: 1.2em; font-weight: normal;">Add Donation</h4>
44 <div class="modal-body" style="padding: 0 5px 5px 5px;">
45 <div class="form-group" style="border-bottom: 1px solid #ccc; padding: 5px 0 5px 5px;">
46 <input type="text" class="form-control" style="width: 95%; border: none; border-bottom: 1px solid #ccc;" value="">
47 </div>
48 <div class="form-group" style="border-bottom: 1px solid #ccc; padding: 5px 0 5px 5px;">
49 <input type="text" class="form-control" style="width: 95%; border: none; border-bottom: 1px solid #ccc;" value="">
50 </div>
51 <div class="form-group" style="border-bottom: 1px solid #ccc; padding: 5px 0 5px 5px;">
52 <input type="text" class="form-control" style="width: 95%; border: none; border-bottom: 1px solid #ccc;" value="">
53 </div>
54 <div class="form-group" style="border-bottom: 1px solid #ccc; padding: 5px 0 5px 5px;">
55 <input type="text" class="form-control" style="width: 95%; border: none; border-bottom: 1px solid #ccc;" value="">
56 </div>
57 <div style="padding: 5px 0 5px 5px;">
58 <input type="button" class="btn btn-primary" value="Add Donation" style="width: 100%; border: none; border-radius: 4px; font-weight: normal; text-decoration: none; color: white; background-color: #337ab7; padding: 5px 10px; font-size: 0.9em;">
59 </div>
60 </div>
61 </div>
62 </div>
63 </div>

```

Figure 14: Creating a new donation source code

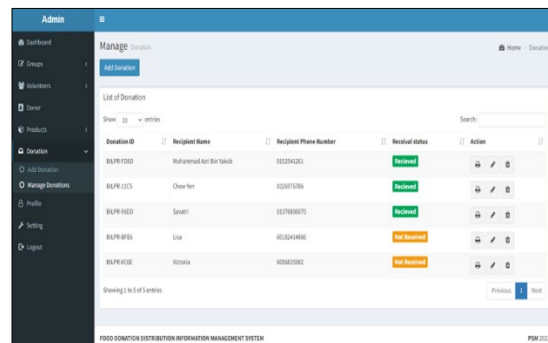


Figure 15: Listed donation data

4.2 Testing

A test is carried out to assess the functionality of each module. A User Acceptance Test (UAT) method is utilized to perform testing. The purpose of this test is to verify whether the volunteers are allowed to register for an account, login into the system, and whether the system will restrict login if an incorrect credentials is entered. Users conducted several tests throughout the registration and login procedure, as shown in Table 4

Table 4: Test case of registration and login module

No	Description	Expected	Actual	Result
M1-1	To check whether volunteer can register for an account	The user should be able to create for an account	The user has successfully created for an account	Pass
M1-2	To check whether an administrator or volunteer can login into the system	The user should be able to login into the system	The user has successfully logged into the system	Pass

M1-3	To check whether the system will restrict login whenever a wrong credential is entered	The system should restrict login when an incorrect credentials has been entered	The system restricted the login when an incorrect or no credentials has been entered	Pass
------	--	---	--	------

Table 5 show the test case for Inventory Management module. There are total of 3 test case for this module. The purpose of this test is to verify whether the administrators or volunteers are allowed to create, delete and edit products into the inventory database.

Table 5: Test case of inventory management module

No	Description	Expected	Actual	Result
M2-1	To check whether user can add new products into inventory	The user should be able to create new product data	The user has successfully recorded product data into the inventory	Pass
M2-2	To check whether user can delete products from the inventory	The user should be able to delete product data	The user has successfully deleted product data from the inventory	Pass
M2-3	To check whether user can edit the product data from inventory	The user should be able to edit product data	The user has successfully edited product data from the inventory	Pass

Table 6 show the test case for Donor module. There are total of 3 test case for this module. The purpose of this test is to verify whether the administrators or volunteers will successfully be able to key in donor details into the database.

Table 6: Test case of donor module

No	Description	Expected	Actual	Result
M3-1	To check whether user can add new donor into database	The user should be able to create new donor data	The user has successfully recorded donor data into the database	Pass
M3-2	To check whether user can delete donor data from database	The user should be able to delete donor data	The user has successfully deleted donor data from the database	Pass
M3-3	To check whether user can edit donor's data from the database	The user should be able to edit donor data	The user has successfully edited donor data from the database	Pass

Table 7 show the test case for Donation Management module. There are total of 3 test case for this module. The purpose of this test is to verify whether the administrators or volunteers will successfully be able to key in donation details into the database

Table 7: Test case of donation management module

No	Description	Expected	Actual	Result
M4-1	To check whether user can add, delete and edit food recipient into database	The user should be able to create new, delete and edit food recipient info into the database	The user has successfully recorded food recipient data into the database	Pass
M4-2	To check whether user can update receival status for food recipient	The user should be able to update receival status for food recipient	The user has successfully update receival status for food recipient	Pass
M4-3	To check whether user can print out individual donation data	The user should be able to print out individual donation data	The user has successfully print out individual donation data	Pass

5. Conclusion

In conclusion, this project will have a positive impact on the procedures commonly used especially in the process of food donation information management among NGOs.

Acknowledgment

I would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

References

- [1] Kagan, J. (2021, August 14). "What is a charitable donation?" Investopedia. <https://www.investopedia.com/terms/c/charitabledonation.asp>.
- [2] Desai, V. (2014). 8 "The role of non-governmental organizations (NGOs). In *The companion to development studies*" (pp. 590-594). Routledge
- [3] Ali, A. F. M., Ibrahim, M. F., & Aziz, M. R. A. (2018). "Food and Non-Food Expenditure Trends Among the Poor and Needy in Kelantan, Malaysia." *Journal of Entrepreneurship and Business*, 6(1), 59-78
- [4] Hasselbring, W. (2000). *Information system integration. Communications of the ACM*, 43(6), 32-38.
- [5] MacGregor, E. R. (2016). *Industry Food Donation Guidelines Part 3: Processor, Caterer/Restaurant and Retailer Specific Information*.
- [6] Suhaily Hasnan, Zubaidah Zainal Abidin, Maslinawati Mohamad, & Norashikin Kamarudin. (2012, September). *Issues, challenges and the way forward for charitable organizations in Malaysia*.
- [7] Jūratė Stankevičienė, Natalija Šedžiuvienė, Jolita Vveinhardt, & Deimantė Mitkuvienė. (2007). *Data Management in Modern Organization: the Aspect of Project Designing*.
- [8] Sarah Lewis (2019). "Prototyping Model". TechTarget.