Applied Information Technology And Computer Science Vol. 3 No. 2 (2022) 1214-1227


**AITCS**

Homepage: http://publisher.uthm.edu.my/periodicals/index.php/aitcs
e-ISSN :2773-5141

# Recyclable Waste Management System for Tzu Chi Recycling Center

**Liaw Jing Yin[1], Nureize Arbaiy[1*]**

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

DOI: https://doi.org/10.30880/aitcs.2022.03.02.074
Received 29 Julai 2022; Accepted 26 October 2022; Available online 30 November 2022

**Abstract**: A recycling collection center is an organization that collects recycled materials for the purpose of resale or conversion to other products. these centers are usually run by companies or individuals. However, there are several issues encountered in managing activities at recycling center including manual entry for data storage, manpower task management, and lack of a systematic system to support recycling activities. To support recycling activities at the case study location, Tzu Chi, a system was developed for administrators, volunteers, and drivers at this recycling center. The system allows users to keep records of recycled waste collected in a database. Administrators can retrieve records to generate reports, generate delivery orders, create payment invoices, and control task management. Volunteers can make new collected waste records into system and create delivery order, while driver can access to created delivery orders. An object-oriented approach was used for the system analysis, and the system development is guided by a prototype-based methodology. This system was developed with MAMP, PHP, Dreamweaver, and Sublime. Through this system, users at the recycling center can manage their recycling activities more efficiently especially the management of their operational data. The system is also expected to keep all records of recycling waste, the activities, and other related transaction record in the database.

**Keywords**: Recyclable waste, Database, Information management system

## 1. Introduction

Growth of municipal solid waste generation in Malaysia was increased with the population and urbanization growth [1]. Malaysia has been reliant on landfilling to dispose those municipal solid waste causing environment pollution around the landfill sites [2]. Since 1993, recycling activity is launched by Ministry of Housing and Local Government (MHLG) in tend to replace landfills. Recycling is a process of converting waste into new useful material or product [3]. Recycling process starts from collecting the recyclable wastes, sorting by types, processing into raw material, or remanufacturing into new products. A lot of recyclable wastes are produced from daily uses like paper, plastic bottle, glass bottles, cardboards and Tetra Pak carton, clothes, aluminiums, and tin cans. Meaghan [4] ever stated recycling can preserve natural resources as the needs of mining new material is reduced and save energy required by using recycled material.

**Corresponding author: nureize@uthm.edu.my*

To further understand the process of recycle waste management, this project had conducted a study case at the Tzu Chi Foundation from Kuching, Sarawak for their recycling events. The recycling event involved volunteers who responsible for waste collection and sorting at recycling points, drivers who delivery recyclable waste, and administrator who manage the recycling event. After several interviews with them, few problems have been identified like manual entries method in their activities. The data entries including data records of new collected waste, and unreliable payment invoice from buyer. Besides that, randomization of volunteers in task management also caused difficulties for Tzu Chi like uneven manpower shuffling and non-fixed task distribution. Another issue is the generation of reports by the current system. The current system does not have filtering functions such as filtering by date, location, type, or buyer.

Therefore, this project developed a system for organizations to manage their recycling events and business data. The system provides a data storage to keep the recyclable waste records. The system also keeps track of delivery orders for internal purpose like recyclable waste transportation or external purpose to deliver order to buyer. The system also allows the administrator to filter the records of recyclable waste and prepare a report for the Tzu Chi Foundation. The system users including administrator, volunteers, and driver from Tzu Chi. Administrator in charge of managing the recycling events, so they responsible for managing recyclable waste record, creating new delivery order to buyer, managing task management, and generating report. Compared to administrator, volunteer has less feature like manage latest recyclable waste record, view task schedule, and create delivery order for internal purpose. While driver only able to access to delivery order module to complete the pending delivery orders.

## 2. Related Work

Recycling centers were built to collect donations of recyclables from the public. Recycled waste will be separated by their volunteers before being sold to buyers or for other recycling purposes. The process starts from the collection of waste at a recycling center or recycling site. The recycling point is a temporary place for volunteers to collect recycled waste from the public. The difference between a recycling center and a recycling point is that the recycling center also serves as a storage place to store the recycling waste that is collected and opened on working days. Once volunteers receive new recyclable waste, they will sort the accumulated waste and dispose of the non -recyclable waste. Afterwards, volunteers from each recycling point will weigh the sorted recycling waste and record it manually on paper, then send it to the administrator for inclusion in their computer record system. However, the system is only in the form of a spreadsheet and there is no dedicated database. Currently, the current system records only the amount of waste, type and location accumulated. The system is managed by an administrator.

Task management is a process to manage a task lifecycle, including task planning, estimation, and scheduling [5]. A task management system is used to support those activities with some extra features like track dependencies, milestones, and task priority. The common features of task management system including the plan and schedule for each task's activities, assigning task to members, prioritization (in term of hierarchy of task importance, difficulty, and time consumption), user permission control, and others. This task management method will be included in the new system. Looking at the implementation of processes in recycling centers at present, web -based systems and databases are needed to improve the existing processes. This is especially true for the storage, management, and analysis processes of business operations data. This will help organizations manage their business operations, task distribution and workforce more efficiently. Three similar systems were also investigated and compared with the proposed system. The results of this study are important to see the good features that can be implemented in a new system. **Table 1** summarizes the comparison.

**Table 1: System's Comparison**

| System / Features | Waste Logics | SoftExpert Waste | Routeware | Proposed System |
|---|---|---|---|---|
| Login and Registration | √ | √ | √ | √ |
| User | Administrator, driver, customer | Waste management team, driver | Officer, driver | Administrator, volunteer, driver |
| Delivery | Route planning, Delivery order with payment and signature | Track the waste movement | Real-time monitor and track the route | Records delivery order details. |
| Report | Analyses business performance | Form analytic report with filter | Not available | Generate report with filter |
| Task Management | X | X | X | √ |
| Invoice | √ | √ | X | √ |
| Platform | Cloud-based& Android application | Web-based application | Web-based& Android application | Web-based application |

## 3. Methodology

Prototyping model is a type of software development model consists of several preliminary version of prototypes that will be presented for customer to evaluate and find out the requirement is achieved or not. Once that prototype fulfilled the requirement and it will be upgraded with other features until a final product form [7]. This model is chosen for this project because prototypes are needed to satisfy the customer's needs. The prototyping model phases begin with initial requirement, iteration phases of design, constructing prototype and customer evaluation, then followed with implementation, and testing. The level of requirements details could be classified in to two types which is functional and nonfunctional requirements [8]. Functional requirement is defined as a main basic action or feature that must be included in a system to support the main driver system activities. Non-functional requirement is defined to present the tasks performance in term of the overall system [9]. In **Table 2** summarizes the functional requirements according to the study case, while **Table 3** summarizes the non-functional requirements.

**Table 2: Functional requirements.**

| No | Module | Description |
|---|---|---|
| 1 | Registration and Login Module | • Allow the new users to register new account before login.<br>• Allow the existing users to login with the id and password.<br>• Redirect the valid users to dashboard when successful login. |
| 2 | Recyclable Waste Records Module | • Allow the volunteer to add new record.<br>• Allow the volunteer to edit the newest record within the available time. |

| | | |
|---|---|---|
| | | • Allow the administrator to retrieve the recyclable waste records for checking the waste storage amount. |
| 3 | Task management module | • Allow the administrator to assign the manpower tasks evenly for each recycling centre and recycling points.<br>• Allow the volunteer to access to the add and edit records feature.<br>• Allow the volunteer to check their schedule. |
| 4 | Delivery Module | • Allow the administrator and volunteer to create new delivery order.<br>• Allow the driver access to the delivery order when transferring the recyclable waste.<br>• Allow the driver to record each delivered wastes' cost unit. |
| 5 | Invoices Module | • Allow to store the invoice histories from buyer.<br>• Administrators generate payment invoice with the delivery order |
| 6 | Report Module | • Administrator can generate report about their sales, collected recyclable wasted and account statement for accountant |

**Table 3: Non-functional requirements.**

| No | Requirement | Description |
|---|---|---|
| 1 | Performance | The system should be always usable |
| 2 | Operational | The loading time required for a website is no more than 1 minute |
| 3 | Security | The system should be user friendly |
| 4 | Cultural and political | The system should be able to work on any web browser |

User requirement analysis is essential to discover and list out the user requirements. The description of requirements might be the functions or settings that will support the system operation. The requirements could be in terms of organization, usability, or function mapping [10]. User requirements are listed in the **Table 4**. These are the functionality expectation from the users.

**Table 4: User requirements.**

| No | Requirement |
|---|---|
| 1 | All their users including administrator, volunteers and drivers must have an account with valid id and password. |
| 2 | The system should be able to keep the overall recyclable waste records. |
| 3 | Volunteer should be able to add new records when they received new recyclable waste. |
| 4 | Volunteer should be able to edit their added records in a time. |
| 5 | Administrator should be able to manage the recyclable waste records. |
| 6 | Administrator should be able to manage their volunteers' task schedule. |
| 7 | Administrator should be able to crate task list according to the date. |
| 8 | Administrator should be able to generate report |
| 9 | Administrator and volunteer should be able to create new delivery order for each recyclable waste transportation occurred. |

| | |
|---|---|
| 10 | Each delivery order must have included details like waste amount & type, delivery source & destination locations, and sender & receiver. |
| 11 | Driver should be responsible as the delivery order's sender. |
| 12 | Driver should confirm with each order's cost unit. |
| 13 | Driver should make sure the delivery order includes the recipient details. |
| 14 | Administrator should be able to generate the invoice for the completed delivery orders. |

Use case diagram from object-oriented approach is used to present the system requirements with use cases, identify the actors and show the interaction of actors with the use cases. **Figure 1** shows the use case diagram of the developed system with 3 actors: Volunteer, Driver and Administrator. There is total 6 use cases which are registration and login, waste record management, generate report, task management, manage delivery order and manage invoices.

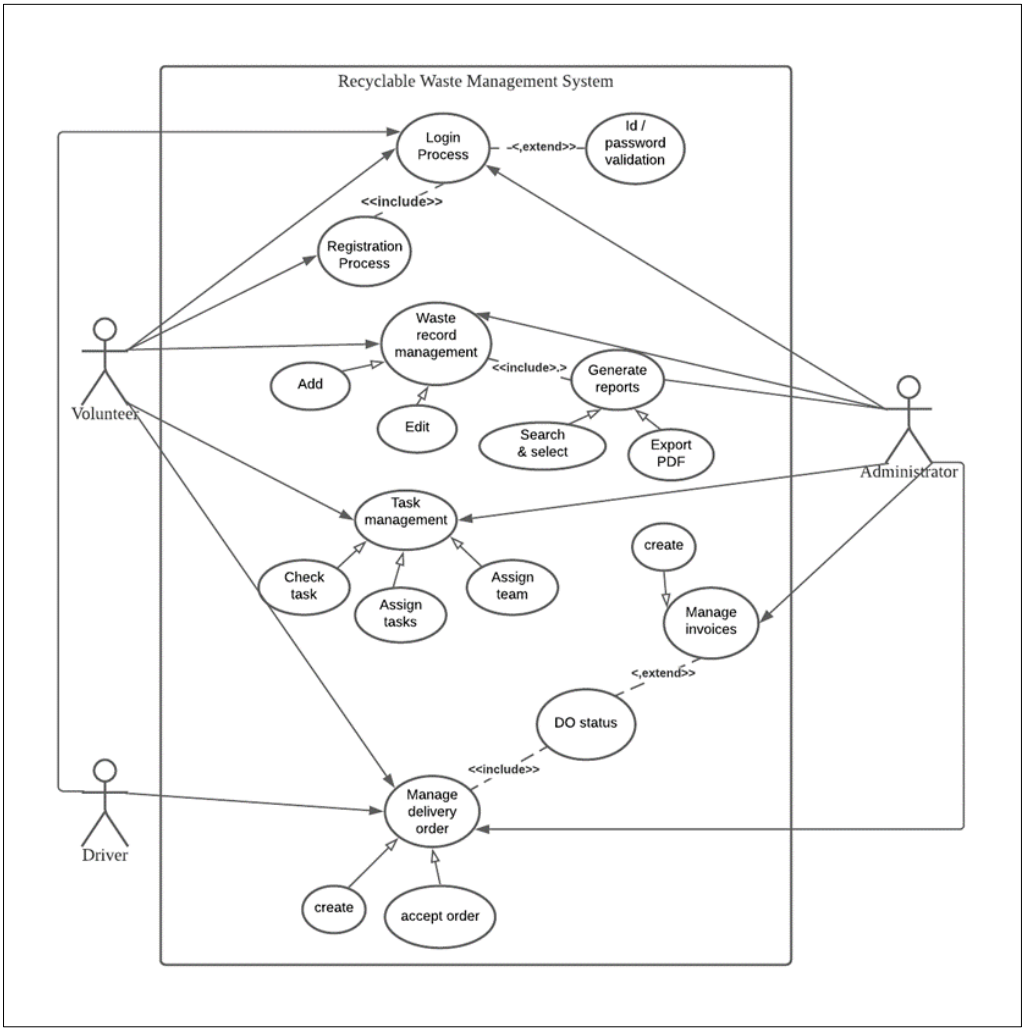


**Figure 1: Use case diagram**

Class diagram is an overview of system's attributes and functions that specified in classes and the relationship between the classes [11]. **Figure 2** shows the class diagram for the system with thirteen

classes namely are `tbl_user`, `Volunteers`, `Administrator`, `Driver`, `tbl_team`, `tbl_task`, `tbl_wasterecord`, `tbl_inventory`, `tbl_category`, `tbl_invoice`, `tbl_report`, `tbl_dolocation` and `tbl_deliveryorder`.

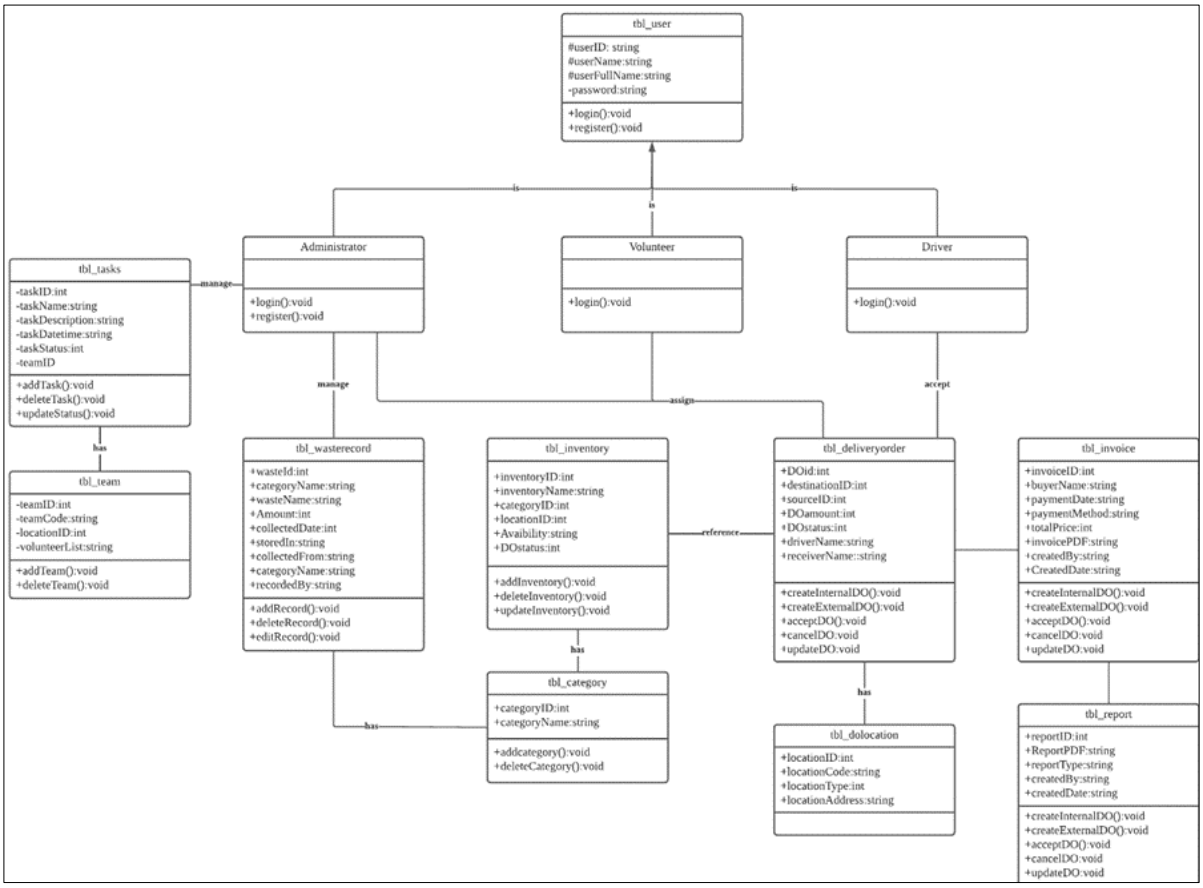


**Figure 2: Class diagram**

Flowcharts are used to visualize system processes by breaking them down into multiple parts and arranging them in a logical manner. **Figure 3** depicts the operation of a recycling waste management system. Volunteers enter new records into the system after collecting waste at a recycling center or recycling site, and this is how progress is made. Volunteers can place new delivery orders (internal use) to carry recycled waste collected from the recycling site back to the recycling center. The administrator must first inspect the waste stock before contacting the buyer.

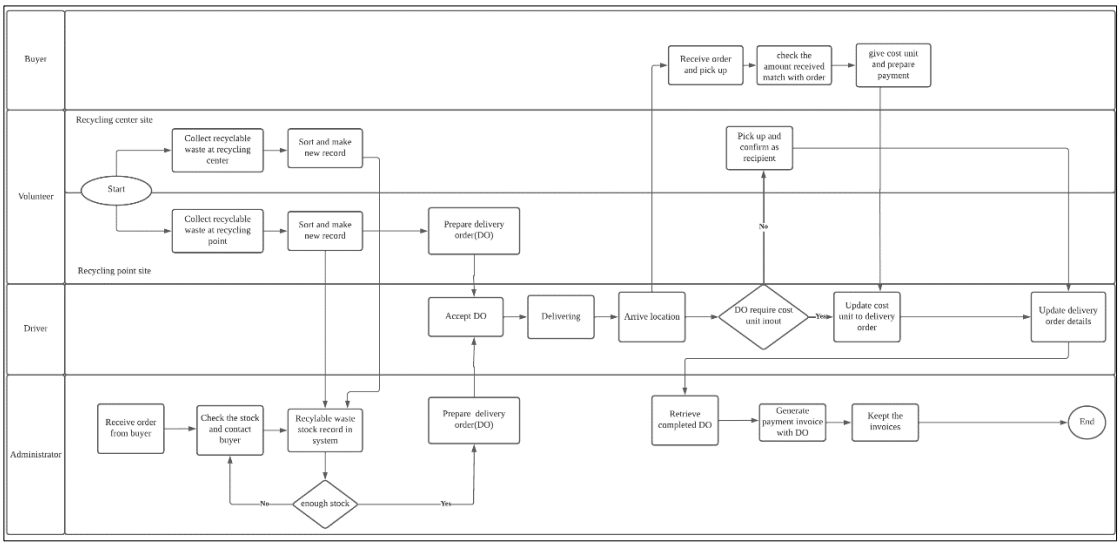


**Figure 3: Flowchart**

They can only complete the transaction and place a new delivery order (external purpose) to ship the recycled waste to the buyer if they have enough stock. A delivery order will be given to the driver. To fulfil an order for internal use, the driver only needs to submit the recipient's details. Before fulfilling the delivery order for external purpose shipments, the driver must update the order information in the system, such as cost units. The administrator saves the customer's payment invoice by creating an invoice record referring the completed delivery order after receiving payment from the buyer.

The designs are created according to the requirements information gathered at analysis phase. The designs will ensure the executions of the construction prototype in next phase. System architecture, database design and user interface design are explained in the following. System architecture is used to provide an overview of system concepts including system outlines, behaviors, and component relationships in a visual presentation. **Figure 4** is the system architecture that can be accessed by system users from different sites. Users include administrators, volunteers, and drivers. The system also requires a database to store records.

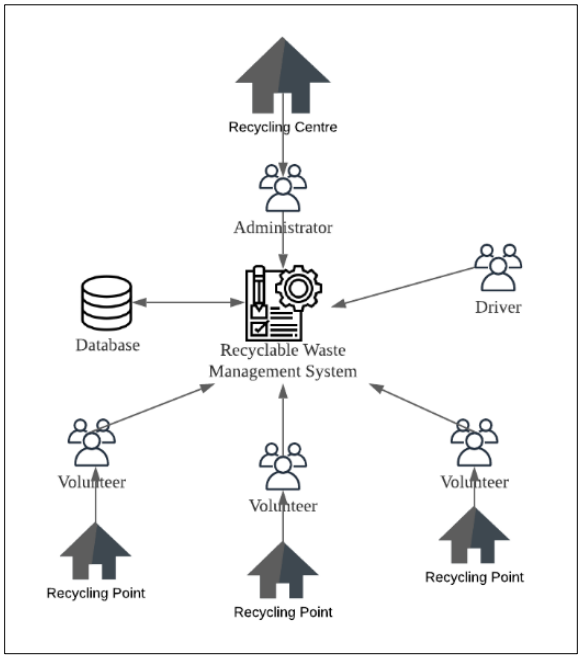


**Figure 4: System Architecture**

Database schema is listed in the following:

i. `tbl_user (userID, username, userFullName, password)`
ii. `tbl_team (teamID, teamCode, locationID, volunteerList)`
iii. `tbl_task (taskID, taskName, taskDescription, taskDatetime, teamID, taskStatus)`
iv. `tbl_wastecategory (categoryID, categoryName)`
v. `tbl_wasteinventory (inventoryID, inventoryName, categoryID, locationID, Avaibility, DOstatus)`
vi. `tbl_wasterecord (wasteId, categoryName, wasteName, Amount, collectedDate, storedIn, collectedFrom, recordedBy)`
vii. `tbl_deliveryorder (DOid, DOtype, destinationID, sourceID, DOitem, DOamount, DOdate, DOstatus, driverName, receiverName, unitCost)`
viii. `tbl_dolocation (locationID, locationCode, locationType, locationAddress)`
ix. `tbl_invoice (InvoiceID, buyerName, paymentDate, paymentMethod, totalPrice, invoicePDF, CreateDate, createBy)`
x. `tbl_report (reportID, reportType, reportPDF, createDate, createBy)`

**Figure 5** to **Figure 8** show interface design for waste record management, task management, delivery order, invoice PDF preview and report generation.

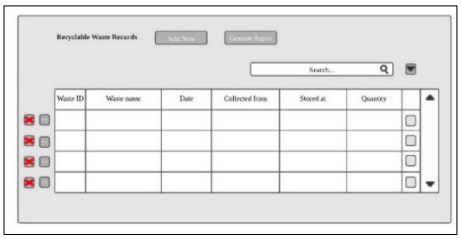


**Figure 5: Waste records management design**

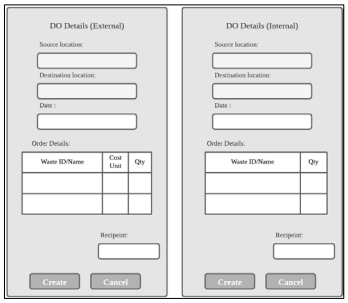


**Figure 6: Delivery order forms design**

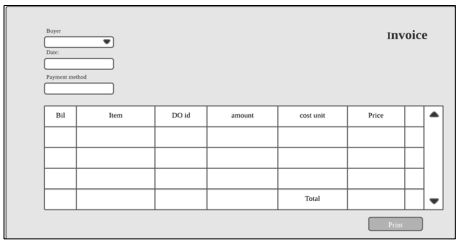


**Figure 7: Invoice preview design**

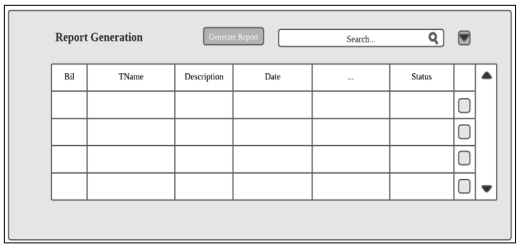


**Figure 8: Report generation page design**

## 4. System Implementation and Testing

To implement this project, system modules development is using several software like MAMP, Adobe Dreamweaver 2021 and sublime text. The coding language used are HTML and CSS for developing the website. For the system operation, the programming language mainly using the PHP language and some Java script for certain function.

## 4.1 Registration and Login Module

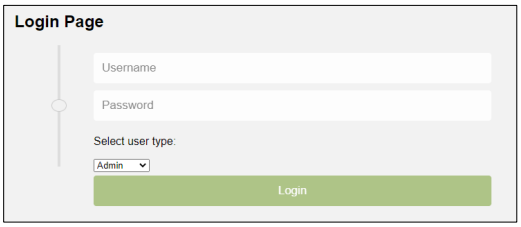


**Figure 9: Login interface**

**Figure 10: Login code segment**

**Figure 9** shows the login process will require the username, password, and user role. The system directs the user to their homepage according to their user role type. For loop is used to authenticate the user validation and switch case will direct the user to homepage according to their user role as shown in **Figure 10**. **Table 5** show the test case for the Login and Registration module, the four-test case's purpose is to verify whether the administrator is allowed to register for new users, all user able to login into the system, and whether the system will restrict login if an incorrect credentials is entered.

**Table 5: Test Case for Account Registration and Login Module.**

| Test Case ID | Description | Expected Result | Actual | Pass/Fail |
|---|---|---|---|---|
| M1-1 | To check whether administrator can register for new user | The administrator should be able to create for an account | The administrator has successfully created for an account | Pass |
| M1-2 | To check whether user 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 password, username or role is entered | The system should restrict login when an incorrect password, username or role has been entered | The system restricted the login when an incorrect or no credentials has been entered | Pass |
| M1-4 | To check whether the user login to their corresponding role homepage | The system should direct the user to their home page according to their role | The system restricted the login when an incorrect role selected even though the username and password is correct | Pass |

## 4.2 Recyclable waste records Module

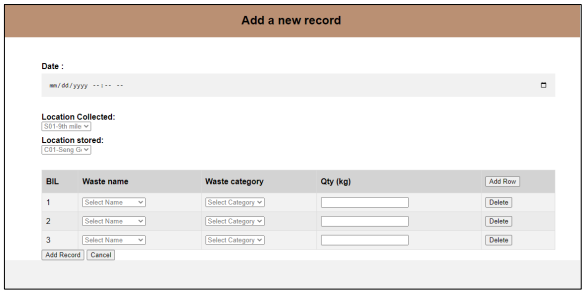


**Figure 11: Add waste record form**

**Figure 11** shows the add new records form for the administrator and volunteer to add new collected waste records into database. Dynamic table is used to the user to key in multiple rows of waste records and insert to database at the same time. Each row of the table also has 2 drop boxes for user to select waste name and type, the drop box selection options were retrieved from inventory and category database respectively. **Table 6** is the passed test cases that verify whether the user could manage the recyclable waste records with this module.

**Table 6: Test Case for Recyclable Waste Records Module**

| Test Case ID | Description | Expected Result | Actual | Pass/Fail |
|---|---|---|---|---|
| M2-1 | To check whether user can add new record | The user should be able to add new records | The user has successfully added new records | Pass |
| M2-2 | To check whether user can edit the record | The user should be able to edit the record | The user has successfully edit the record | Pass |
| M2-3 | To check whether the administrator able to view the created records | The system should display the records list from the database | The user can view the records in table form | Pass |
| M2-4 | To check whether the administrator able to check the waste amount in the storage | The system should display the waste amount in different storage | The waste total amount is recorded in waste inventory and user can view it | Pass |

## 4.3 Task Management Module

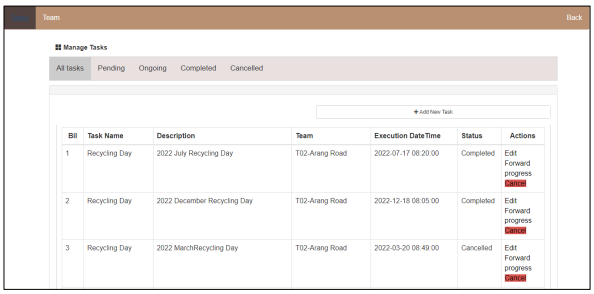


**Figure 12: Task management interface**

```
links" onclick="openTab(event, 'all')" id="defaultOpen">All tasks
links" onclick="openTab(event, 'pending')">Pending</button>
links" onclick="openTab(event, 'ongoing')">Ongoing</button>
nks" onclick="openTab(event, 'completed')">Completed</button>
nks" onclick="openTab(event, 'cancelled')">Cancelled</button>
```

**Table 13: Code segment for task management.**

Task management is using different tabs to presenting task progress. The task progress starting from pending to ongoing and finally completed. Cancelled is also one of the task statuses. Every time the administrator creates new task it will be set as pending status. The task status can be forwarded by clicking 'forward progress' button in **Figure 12**. **Figure 13** shows the code for different task progress tabs. Table 7 is the passed test case to verify whether user can manage their created task and view the task progress.

**Table 7: Test Case for Task Management Module**

| Test Case ID | Description | Expected Result | Actual | Pass/Fail |
|---|---|---|---|---|
| M3-1 | To check whether administrator assign volunteers into a team. | The administrator should be able to create team with responsible recycling site and volunteers | The team successfully created | Pass |
| M3-2 | To check whether administrator can assign task for the team | The administrator should assign task for the team | The administrator has successfully assigned task for the team | Pass |
| M3-3 | To check whether the volunteers can check their responsible team and assigned task. | The volunteers should be able to check their responsible team and assigned task. | The volunteers successfully to check their responsible team and assigned task. | Pass |
| M3-4 | To check whether the system able to track the task progress | The system should display the task progress. | The system successfully displays the task progress. | Pass |
| M3-5 | To check whether the volunteer and administrator can update the task progress status. | The volunteer and administrator should be able to update their task progress status. | The volunteer and administrator success to update their task progress status. | Pass |

## 4.4 Delivery Module

```
for ($b=0; $b<count($_POST["wName"]); $b++){
$sql2="UPDATE
            tbl_wasteinventory i
        SET
            i.Avaibility =(i.Avaibility - '$wAmount[$b]')
        WHERE
            i.inventoryName = '$iName[$b]'";

}
```

**Figure 14: Code segment to update waste inventory**

**Table 5: Delivery progress flow**

| Delivery type | Delivery status progress |
|---|---|
| Internal | Waiting for delivery (2)→ In-transit (3) →Arrived recycle centre (4) |
| External | Waiting for delivery (2)→ In-transit (3) →Accepted by buyer (5) → invoiced (8) |

Same as the task management, delivery module also consists of several status and the progress flow are shown in **Table 5**. As the delivery order is created depend on the waste inventory stock amount. Therefore, whenever a new delivery order is created, the waste inventory should be updated by deducting the amount that will deliver out. **Figure 14** shows the part of code for updating the waste inventory table. To whether the system able to support the users' delivery order progress, test cases is conducted, and the results are listed in **Table 8**.

**Table 8: Test Case for Delivery Order Module**

| Test Case ID | Description | Expected Result | Actual | Pass/Fail |
|---|---|---|---|---|
| M4-1 | To check whether administrator can assign create new delivery order | The administrator should be able to create new delivery order with the collected waste stock. | The administrator success to create new delivery order with the collected waste stock. | Pass |
| M4-2 | To check whether the driver can accept delivery orders and complete it | The driver should be able to complete the accepted delivery orders | The driver success to complete the accepted delivery orders | Pass |
| M4-3 | To check whether the system able to track the delivery order progress | The system should be able to display the delivery order progress. | The system display the delivery order progress in different tabs. | Pass |
| M4-4 | To check whether the driver can update the delivery order, | The driver should be able to update the delivery order details, | The driver able to update the delivery order details. | Pass |

## 4.5 Invoice Module

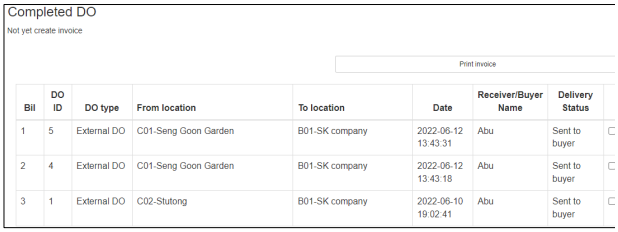


**Figure 15: Payment invoice PDF preview**

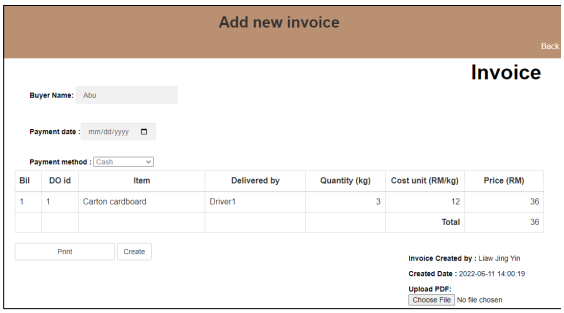


**Figure 16: invoice preview**

To generate the invoice with the completed delivery order information, the completed delivery order table consist of check box in each row as shown in **Figure 15**. At first the administrator needs to select the checkbox, and then clicks the print invoice button to proceed to invoice preview as shown in **Figure 16**. **Table 9** show the test case for the Invoice Module The purpose of this test is to verify whether the system able to help administrator to generate softcopy invoice and keep it in invoices histories.

**Table 9: Test Case for Invoice Module**

| Test Case ID | Description | Expected Result | Actual | Pass/Fail |
|---|---|---|---|---|
| M5-1 | To check whether administrator generate invoice from completed external delivery orders | The administrator should be able to retrieve completed DO to generate invoice | The administrator able to retrieve completed DO to generate invoice | Pass |
| M5-2 | To check whether administrator can export the invoice created. | The administrator should be able to export out the invoice created. | The administrator success to print out the invoice in PDF. | Pass |

| | | | | |
|---|---|---|---|---|
| M5-3 | To check whether the administrator can save the created invoice into invoice histories | The system should have invoice database to keep the invoice records. | The system consist of invoice table in the database. | Pass |

## 4.6 Report Module

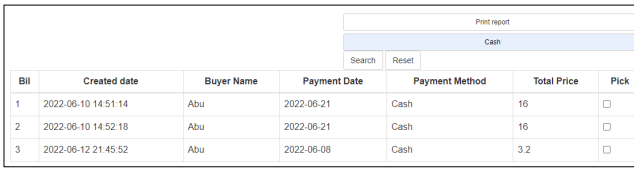


**Figure 17: Searching feature**


**Figure 18: Code segment for searching**

Same as invoice module, report generation also uses check box to select the report content. In this module searching feature is added for the administrator to make searching as shown in **Figure 17**. **Figure 18** is the code segment for searching module. **Table 10** show the test case for report module and these test cases are to verify the system help the administrator generate report.

**Table 10: Test Case for Report Module**

| Test Case ID | Description | Expected Result | Actual | Pass/Fail |
|---|---|---|---|---|
| M6-1 | To check whether administrator generate report with different purpose | The administrator should be able to generate report for sales and collected recycle waste | The administrator able to generate report for sales and collected recycle waste in PDF | Pass |
| M6-2 | To check whether administrator can export the report created. | The administrator should be able to export out the report created. | The administrator success to print out the report in PDF. | Pass |
| M6-3 | To check whether the system provide support function to customise report. | The administrator should be able to customise their report. | The administrator can select and search for the records before generate report. | Pass |

## 5. Conclusion

In conclusion, this method was created for the recycling facility at Tzu Chi. Several modules are suggested and created as the solution to their concerns after analyzing the difficulties they had with the recycling process data management. There are three types of users including administrator, volunteers, and driver. Manual entries problem will be solved by waste record module that allow the volunteers make records in database. They are also able to manage the delivery order more systematic with actual records kept in the database. Other than delivery order, buyer's payment invoices also could have softcopy records save in the database. After administrator confirmed with the payment amount and they can generate and save the invoice with delivery order as reference. Task management's unbalance

manpower issue also can be handled by administrator in this system. As the volunteers are separated into groups and it is easier for administrator to assign responsible recycling point or tasks for each group.

**Acknowledgment**

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

**References**

[1] Fauziah, S. H., Simon, C., & Agamuthu, P. (2004). Municipal solid waste management in Malaysia-Possibility of improvement. Malaysian Journal of Science, 23(2), 61-70.

[2] Badgie, D., Samah, M. A. A., Manaf, L. A., & Muda, A. B. (2012). Assessment of Municipal Solid Waste Composition in Malaysia: Management, Practice, and Challenges. Polish Journal of Environmental Studies, 21(3).

[3] Jereme, Innocent & Chamhuri, Siwar & Alam, Md. Mahmudul. (2014). Waste Recycling in Malaysia: Transition from Developing to Developed Country. 4. 1–14.

[4] Meaghan (2021). Top 10 Items That Should Always Be Recycled Retrieved from: https://www.lrsrecycles.com/top-10-items-that-should-always-be-recycled/

[5] David Robins (2016). Introduction to Task Management Software. Retrieved from: https://www.binfire.com/blog/introduction-task-management-tutorial/

[6] Gintaras Baltusevičius (2020). What is Report Generator. Retrieved from: https://whatagraph.com/blog/articles/report-generator

[7] Neha T (2020). Prototyping Model, Phases of Prototyping. Retrieved from: https://binaryterms.com/prototyping-model.html

[8] Summers, J. D., Joshi, S., & Morkos, B. (2014, August). Requirements evolution: Relating functional and non-functional requirement change on student project success. In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (Vol. 46346, p. V003T04A002). American Society of Mechanical Engineers.

[9] Alsaleh, S., & Haron, H. (2016). The Most Important Functional and Non-Functional Requirements of Knowledge Sharing System at Public Academic Institutions: A Case Study. Lecture Notes on Software Engineering, 4(2), 157.

[10] Maguire, M., & Bevan, N. (2002, August). User requirements analysis. In IFIP World Computer Congress, TC 13 (pp. 133-148). Springer, Boston, MA.

[11] Al-Fedaghi, S. (2017). Diagramming the class diagram: toward a unified modeling methodology. arXiv preprint arXiv:1710.00202.