Megatesting Center Inc. Inventory Management Software (MCIMS)

MCIMS SYSTEMS DOCUMENTATION

Reviewed By: Miguel Carlo Guillermo, MM.
UNIVERSITY OF THE PHILIPPINES MINDANAO | UP MINDANAO, UP MINDANAO, TUGBOK, DAVAO CITY, 8000 DAVAO DEL SUR

About the Team

The completion of project won't be possible without the cooperation and commitment of the developer team to help Megatesting Center Inc. digitize their inventories. The team would also commend our Software Engineering professor, Miguel Carlo Guillermo MM., for his continuous support and sharing of wisdom throughout the development phase.

Below are the team members and their respective roles in the project:

Dory Anne Portillo – Author and Operational Manager

- Coordination and supervision, Human Resources, Communication and Strategic Input
- Email : drportillo@up.edu.ph

Aleimar Villabrille - Software Engineer and Scrum Master

- Design, development, testing and evaluation of the software
- Email : apvillabrille@up.edu.ph

Maria Angela Ramas - Backend Engineer

- Plans and develops utility to every frontend design
- Email : mrramas@up.edu.ph

Paula Ann Gabrielle Yap - Frontend Engineer

- Creates interaction and user experience
- Email : piyap@up.edu.ph

Jeselle Kimberly Carriedo - Database Analyst

- database design, migration, performance monitoring, security, troubleshooting, as well as backup and data recovery
- Email : jmcarriedo2@up.edu.ph

Mia Wednesday Faith Tom - Software Tester

- Investigates the codes for bugs/errors to ensure quality of the software
- Email : mbtom@up.edu.ph

Table of Contents

System Overview

MCIMS was designed to meet the needs of the Megatesting Center Inc. to digitize their inventory. The system contains four different accounts: manager, receiver, releaser and accounting accounts, each of which with different functionalities:

Receiver

Is responsible for receiving daily transactions. Inputs data from a contractor such sample type, contractor name etc. Receiver also needs daily received report and

Is responsible for receiving daily transactions. Inputs data from a contractor such sample type, contractor name etc. Receiver also needs daily received report and all transactions starts from receiver.

Accounting

Is responsible for daily sales reports. This account gives the Official Receipt and calculates the total payment of a certain transactions. On times that there are erroneous inputs from receiver such as misspelled name of a contractor, this account is also responsible for requesting edit to manager for later update and approval.

Releaser

Is responsible for releasing transactions. Once items are done tested releaser releases it and reports daily released transactions.

Manager

Tracks daily sales, daily released and daily received reports. Manager account also is responsible of approving edit requests from accounting.

Scenarios of Typical Usage

All items to be tested starts from receiving account. Data are inputted and once done will be sent to accounting account. Accounting now is responsible for handling payments of these items. Releaser now is responsible for releasing items that are done tested by testers. While such transaction is asynchronously happening, manager keeps tracks of daily activities of these three users.

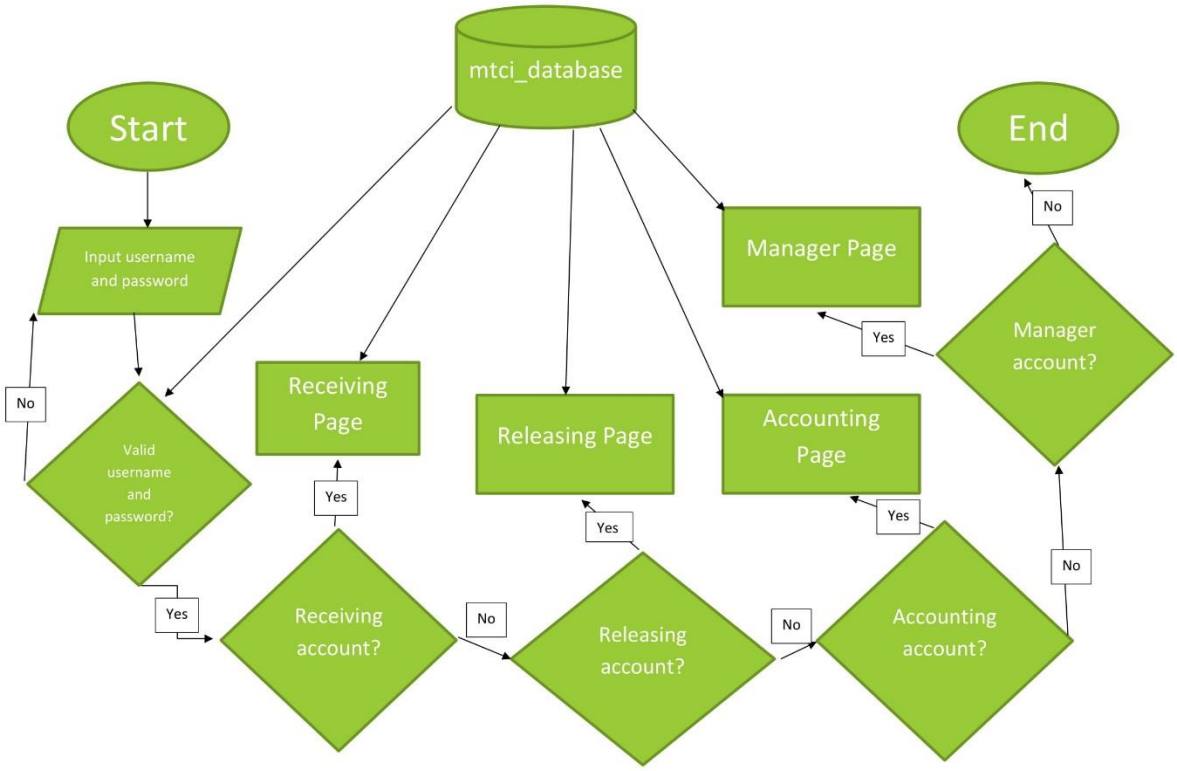
System Design

Hardware and Software Requirements

Table below discusses the hardware and software needed to create the software product:

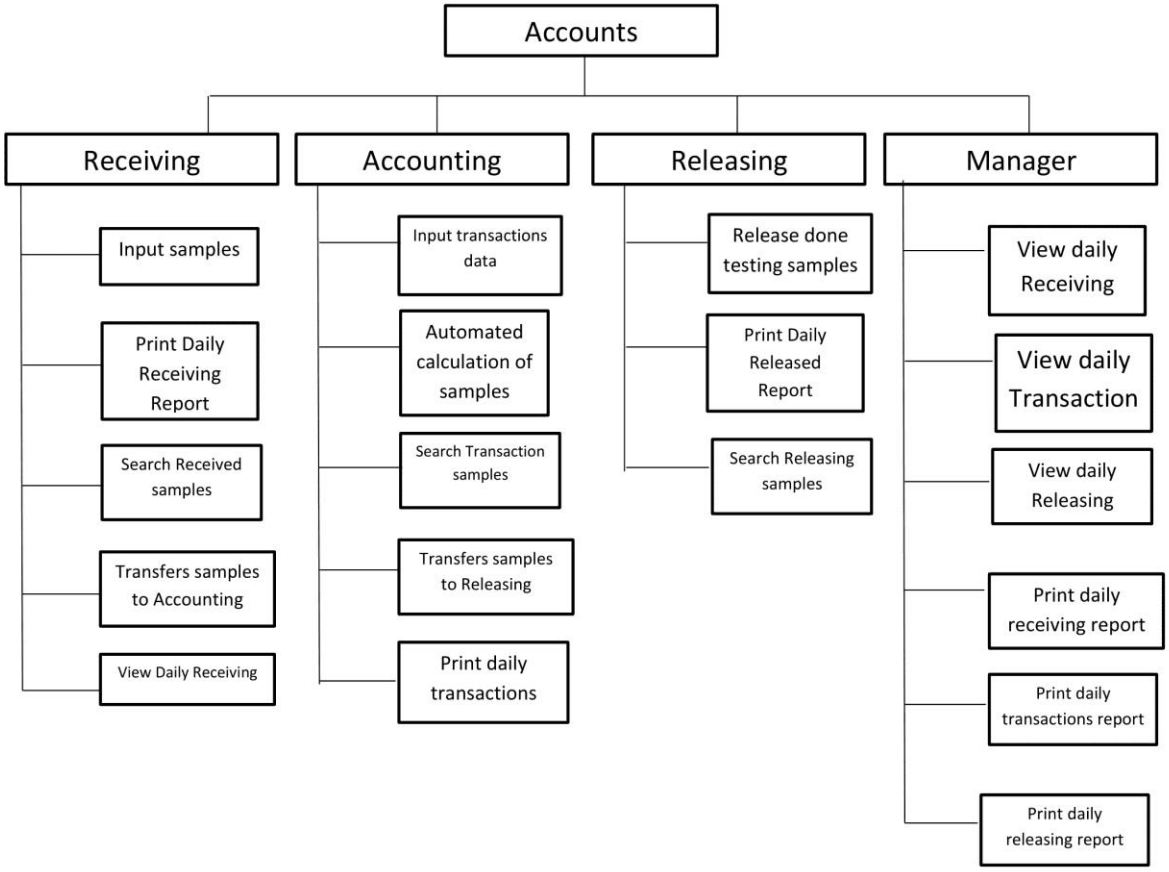
Hardware	Software
Pocket Wi-Fi (for internet connection purposes)	Text Editors (Visual Code, Notepad++, Sublime Text)
Personal Computers (Laptops	XAMPP and 000webhost server account

Control Flow of System



All accounts start from successfully logging in into the system. Various security is embedded in logging such as stripping slashes and SQL injection protection. Once successfully logged in, different users will be redirected to different account pages where controls of accessing and updating specific tables from database was given control of. Logging out will terminate logged in sessions and access of database.

Functions and Modules



Above image is various specific functions of different accounts of the system. Please note that there are also other functions that is inherent to the system that is not mentioned from above such as logging in, logging out, edit password, forgot password, and sending email for concerns in the system. Functions that are explicitly mentioned above are major tasks of a certain account.

Database

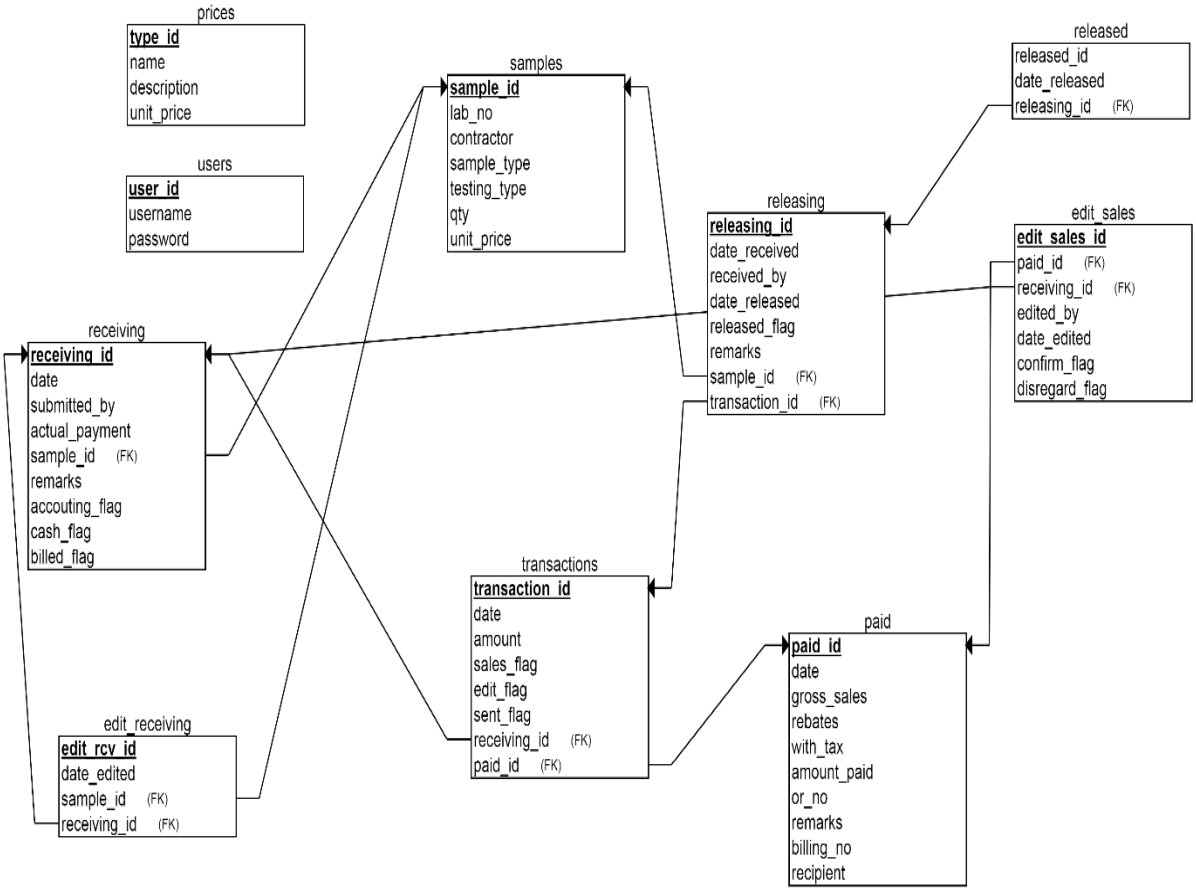


Figure 1 Relational Schema of MTCI Database

The database is normalized in 3NF form since the system demands heavy order entry type of needs. The database removes data duplication which ensures integrity of the database. The database consists of 10 tables. Each contains significant attributes needed in the system.

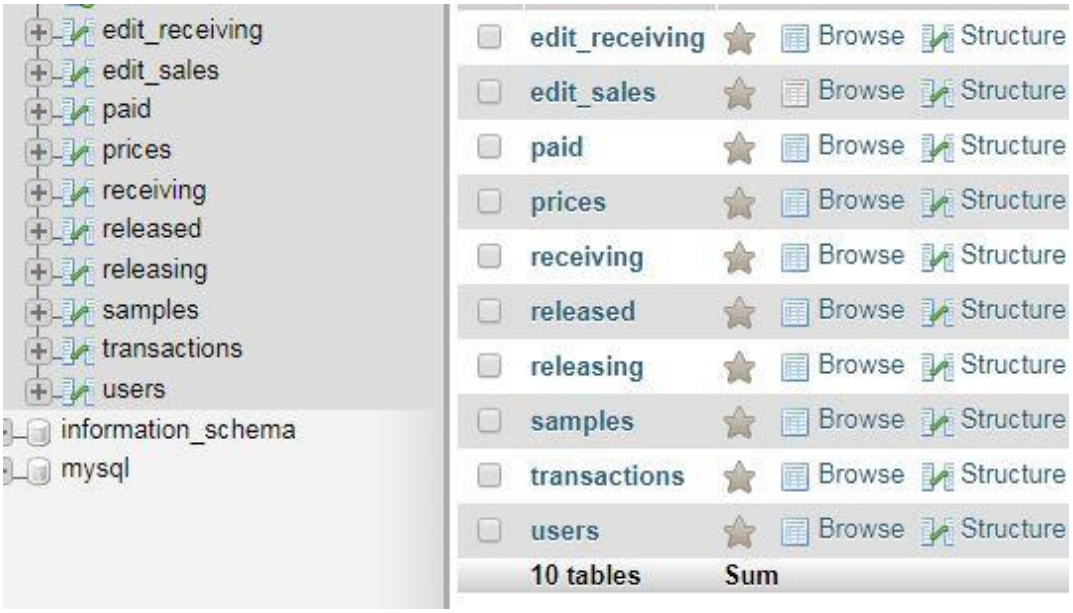


Figure 2 List of Tables of the MTCI database

Tables

- edit_receiving – this table stores all received items that are requested to be edited. Manager will see all these requests and will confirm which will change the value of confirm_flag attribute
- edit_sales – this table stores all transactions items that are requested to be edited. Manager will see all these requests and will confirm which will change the value of confirm_flag
- paid – this table consists all paid items that are ready to be released
- prices – this table consists of all samples prices provided by the company. This allows auto computation of transaction payments
- receiving – all received items that are still not transferred to accounting will be stored in this table
- released – this table consists of all released items that manager could view
- samples – this is stores attributes of a sample in a transaction
- transactions – this contains all required transaction attributes such as official receipt number, total payment etc.
- users – this table stores all user names and password of all accounts.

The database is stored in 000webhost phpMyAdmin a free webhosting site for development and experimentation of the system by the developer team.

Entity Relationship Diagram

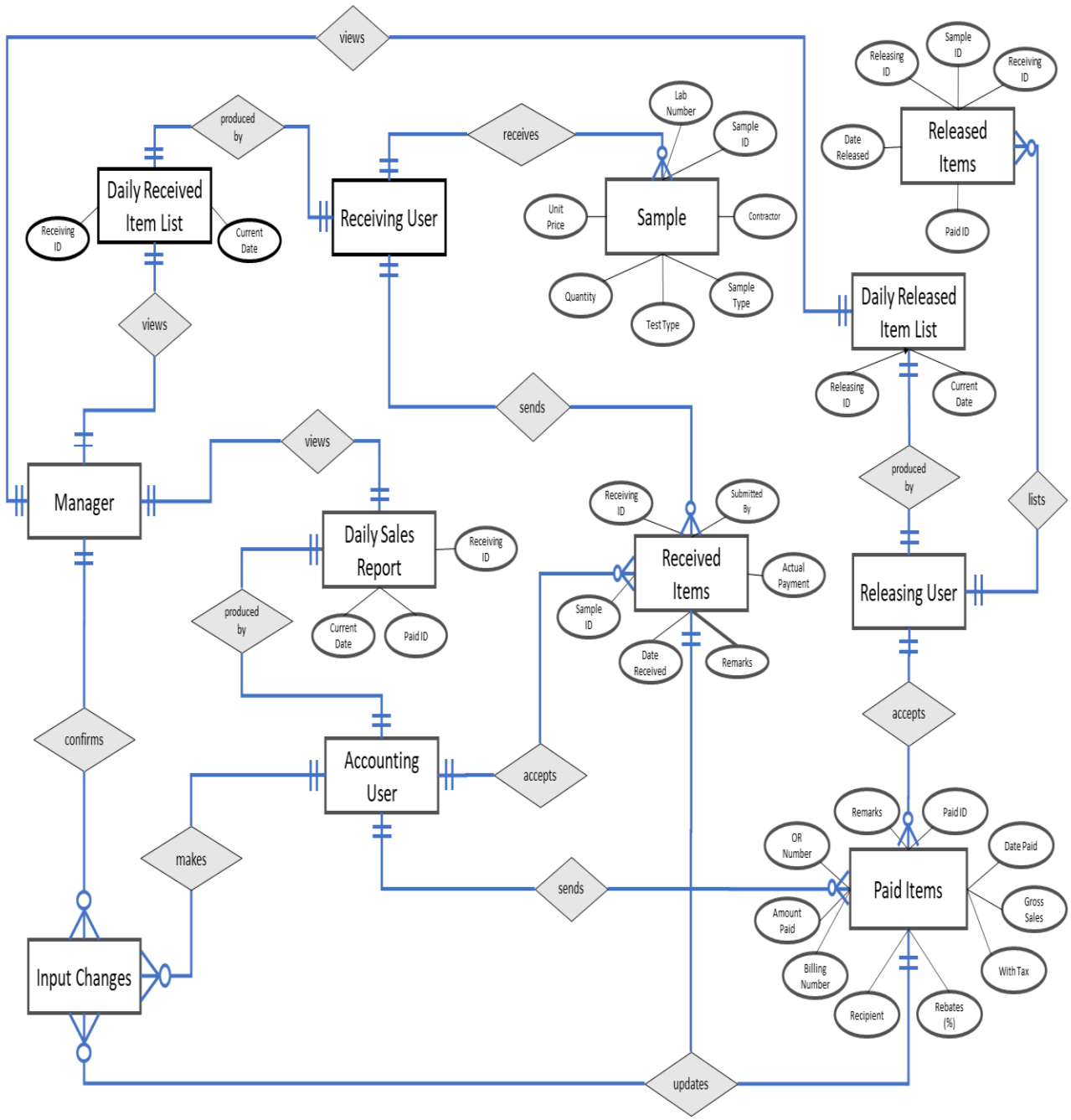


Figure 3 MCIMS Entity Relationship Diagram

Abstract representation through ERD shown by Figure 3 will guide future analysis and improvement of the system.

Frontend Design

For minimalist and faster load content, the team decided not to use any frameworks. The frontend engineer drafted all layouts and designed majority of pages. The team decided to use green and yellow as primary theme colour of pages which based on the company logo colour of Megatesting Center Inc.

CSS Files

Name ▼	Size	Date	Permissions
Edit_Password_CSS.css	3.7 kB	2018-05-14 07:03:00	-rw-r--r--
Etc.css	3.9 kB	2018-05-16 17:58:00	-rw-r--r--
FAQ.css	4.5 kB	2018-05-20 23:46:00	-rw-r--r--
Forgot_PW_CSS.css	2.1 kB	2018-04-07 08:04:00	-rw-r--r--
Help_CSS.css	2.0 kB	2018-04-07 08:04:00	-rw-r--r--
Main_CSS.css	4.7 kB	2018-05-20 23:47:00	-rw-r--r--
Manager.css	3.3 kB	2018-05-17 11:20:00	-rw-r--r--
Receiving.css	3.7 kB	2018-05-14 03:46:00	-rw-r--r--
Releasing.css	2.9 kB	2018-04-27 13:23:00	-rw-r--r--

Figure 4 CSS Files of MCIMS

Figure 4 shows majority CSS files used by the system which could be found in CSS folder. For faster code analysis and editing, the frontend engineer named each CSS files to their explicit usage. Accounting CSS is separated in a different folder since it demands different layout and design.

Images

Name ▼	Size	Date	Permissions
Aleimar.jpg	27.9 kB	2018-05-20 08:45:00	-rw-r--r--
avatar.png	5.9 kB	2018-04-07 08:04:00	-rw-r--r--
bg.jpg	273.5 kB	2018-05-14 05:58:00	-rw-r--r--
Dory.jpg	33.5 kB	2018-05-16 17:12:00	-rw-r--r--
Kim.jpg	106.4 kB	2018-05-20 23:43:00	-rw-r--r--
logo.png	10.3 kB	2018-05-14 05:58:00	-rw-r--r--
megatesting-bg.png	1.3 MB	2018-05-17 13:07:00	-rw-r--r--
Mia.jpg	454.8 kB	2018-05-16 17:12:00	-rw-r--r--
Paula.jpg	43.1 kB	2018-05-16 17:12:00	-rw-r--r--
Ria.jpg	139.7 kB	2018-05-16 17:12:00	-rw-r--r--

Figure 5 Images Files used by MCIMS

Figure 5 shows all images used by the systems. This includes logo of Megatesting Center Inc. developers' photos, background image used in certain pages, and company's banner. All these images could be found in the Images folder and is used by various frontends of the system.

Backend Files

The following subsections briefly discusses some major backend files used by the system:

Logging in

Figure 6 demonstrates how a valid username and password where redirected to their respective pages/directories. Alert by JavaScript will show once username or password are invalid.

```
{
    $_SESSION['login_user']=$username;
    /*we save the the username as this Session's login  user */
    //user is the manager
    if($username == "mtcidavao_manager")
    {
        header('Location: manager/managerdailyreceiveIndex.php');
        exit();
    }
    //user is the receiving
    else if($username == "mtcidavao_receiver")
    {
        header('Location: receiving/');
        exit();
    }
    //user is the accounting
    else if($username == "mtcidavao_accounting"){
        header('Location: accounting/');
        exit();
    }
   //user is the releasing
   else if ($username == "mtcidavao_releaser")
   {
      header('Location: releasing/');
      exit();
   }
}
//invalid user or password
else
{ ?>
    <script type="text/javascript">
        if (confirm("Invalid username/password! ")) {
            window.location.href = "index.html";
```

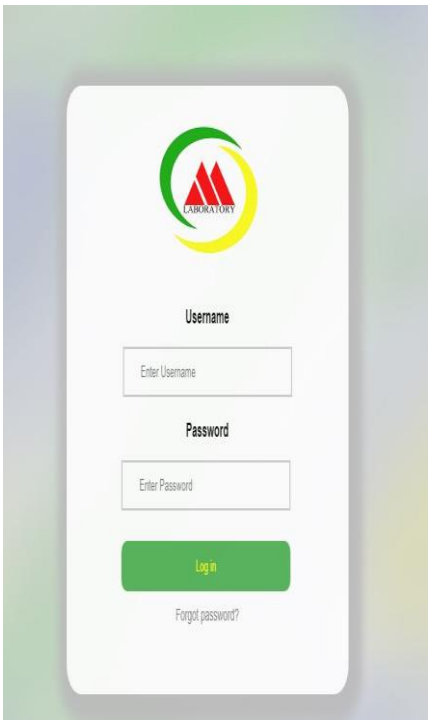


Figure 6 logging.php sample codes and system snippet

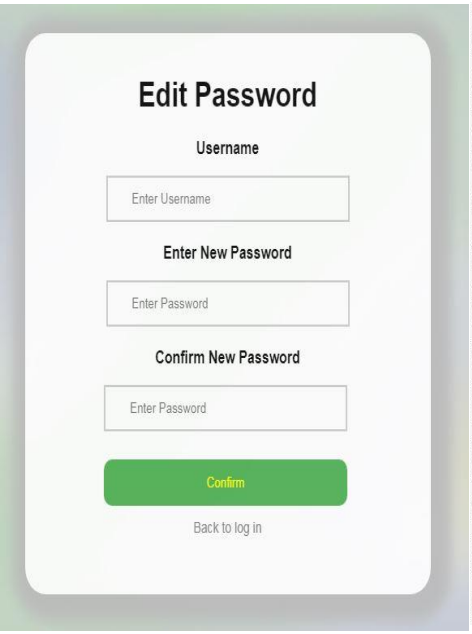
Logging out

Figure 7 shows how logout.php destroy all save sessions so users can't redirect back to their pages unless through re-logging in.

```php
<?php
session_start();
session_unset();
session_destroy();

include 'index.html';
?>
```

Figure 7 logout.php code




```php
$uname=stripslashes($uname);
$newpsw=stripslashes($newpsw);
$conpsw=stripslashes($conpsw);

if (isset($_POST['submit']))
{
    if(($uname == $username) && ($newpsw == $conpsw))
    {
        $sql = "UPDATE users SET password='$newpsw' WHERE username = '$username'";

        if (!mysqli_query($link, $sql))
        {
            echo "Error editing password: " . mysqli_error($link);
            exit();
        }
        header('Location: index.html');     //back to index
    }

    //alert user
    else
    {?>
        <script type="text/javascript">
            if (confirm("Invalid username/password! ")) {
                window.location.href = "index.html";
            } else {
            window.location.href = "index.html";
            }
        </script>
    <?php
    }
}
?>
```

Figure 8 editpassIndex.php code and system snippet

Edit Password

Figure 8 shows how editpassIndex.php file updates passwords in the database. It then redirects user to logging in page to test new password.

Daily Receive Report


```php
while($row=mysqli_fetch_array($result, MYSQLI_ASSOC)){
    $attributes[]=array(
        'lab_no'            => $row['lab_no'],
        'contractor'        => $row['contractor'],
        'sample_type'       => $row['sample_type'],
        'qty'               => $row['qty'],
        'unit_price'        => $row['unit_price'],
        'actual_payment'    => $row['actual_payment'],
        'submitted_by'      => $row['submitted_by'],
        'remarks'           => $row['remarks']);
}

foreach($attributes as $current){
    $i=1;
    $fill=!$fill;
    $pdf->SetFills(array($fill,$fill,$fill,$fill,$fill,$fill,$fill,$fill,$fill));
    $pdf->Row(array(
            $current['lab_no'],
            $current['contractor'],
            $current['sample_type'],
            $current['qty'],
            $current['unit_price'],
            $current['actual_payment'],
            $current['submitted_by'],
            $current['remarks']));
    if($i%19==0){
        $pdf->AddPage();
        $fill=true;
    }
    $i++;
}
```

Figure 9 DailyRecievingPDF.php code and system snippet

Figure 9 shows how DailyRecevingPDF.php file retrieves all attributes needed to print in the pdf file. It is accompanied by DailyReceivingPDF-Attributes.php file which generates the PDF file.

Daily Sales Report


```php
while($row=mysqli_fetch_array($result, MYSQLI_ASSOC)){
    $attributes[]=array(
        'lab_no'            => $row['lab_no'],
        'contractor'        => $row['contractor'],
        'test'              => $row['testing_type'],
        'qty'               => $row['qty'],
        'unit_price'        => $row['unit_price'],
        'amount'            => $row['amount'],
        'gross_sales'       => $row['gross_sales'],
        'rebates'           => $row['rebates'],
        'w_tax'             => $row['with_tax'],
        'amount_pd'         => $row['amount_paid'],
        'or_no'             => $row['or_no'],
        'billing_no'        => $row['billing_no'],
        'remarks'           => $row['remarks']
    );
}

$length=count($attributes);

$oldCont="";
$flag=0;
$j=1;
```

Figure 10 DailySalesPDF.php code and system snippet

Figure 10 demonstrates how DailySalesPDF.php file retrieves all attributes needed for printing. This process is accompanied by DailySalesPDF-Attributes which generates the PDF file.

```php
while($row=mysqli_fetch_array($result, MYSQLI_ASSOC)){
    $attributes[]=array(
        'date_received'     => $row['date_received'],
        'lab_no'            => $row['lab_no'],
        'contractor'        => $row['contractor'],
        'received_by'       => $row['received_by'],
        'date_released'     => $row['date_released']
    );
}

foreach($attributes as $current){
    $fill=!$fill;
    $pdf->SetFills(array($fill,$fill,$fill,$fill,$fill));
    $pdf->Row(array($current['date_received'], $current['lab_no'], $current['contractor'],
    $current['received_by'], $current['date_released']));
}
$date=date('Y-m-d');
$pdf->SetTitle('Daily Releasing Report');
//This outputs the pdf file to the browser
$pdf->SetDisplayMode('real');
$pdf->output('I', 'MTCI-DVO_DailyReleasingReport'.$date.'.pdf');
```

Daily Released Report

Figure 11 shows how DailyReleasedPDF.php fetches all attributes needed to print in daily released report. This is accompanied by DailyReleasingPDF-Attributes.php file which generates the PDF file.

Figure 11 DailyReleasedPDF.php code and system snippets

The following subsections discusses files that are specific to each account.

Receiving

Figure 12 which displays how receiver.php retrieves data and transmits it to index.php found in receiving folder

```php
//submit & update to 'samples' table
$query1 = "INSERT INTO samples (lab_no, contractor, sample_type, testing_type, qty, unit_price)
    VALUES('$lab_no', '$contractor', '$samp_type', '$test_type', '$quantity', '$unitprice')";
//execute query
mysqli_query($con, $query1) or die("Error : " .mysqli_error($con));

//submit & update to 'daily_receiving' table
$date = date('Y-m-d H:i:s');
$last_id = mysqli_insert_id($con);
$actualPayment = $quantity * $unitprice;
$accounting_flag = '0';

$query2 = "INSERT INTO receiving (date, sample_id, actual_payment, submitted_by, accounting_flag, remarks)
    VALUES('$date', '$last_id', '$actualPayment', '$submitted_by', '$accounting_flag', '$remarks')";

//execute query
mysqli_query($con, $query2) or die("Error : " .mysqli_error($con));

$receiving_id = mysqli_insert_id($con);
$query3 = "INSERT INTO transactions (date, receiving_id, amount, paid_id, sales_flag, edit_flag, sent_flag)
VALUES (CURDATE(), '$receiving_id', '$actualPayment', 0, 0, 0, 0)";

//execute query
mysqli_query($con, $query3) or die("Error : " .mysqli_error($con));

header('Location: index.php');
```

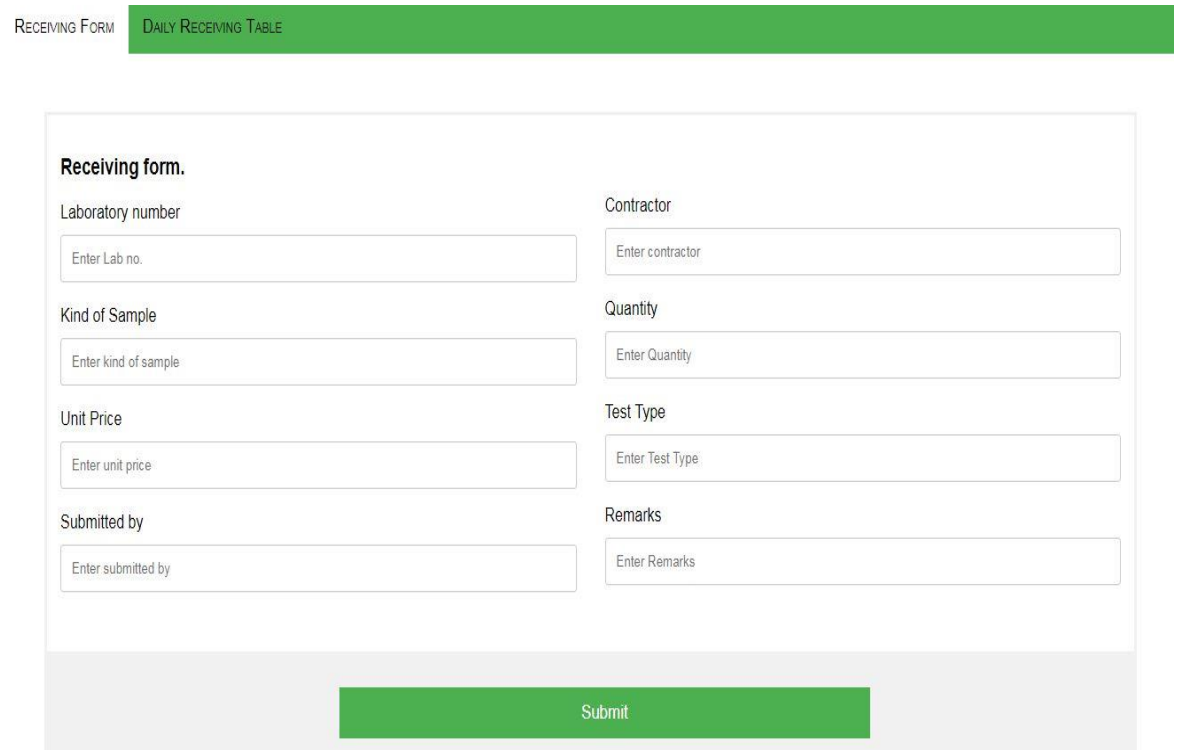


Figure 12 receiver.php code and system snippet

Accounting

```
30      $query="SELECT receiving.receiving_id, receiving.date, samples.sample_id, samples.lab_no,
31              samples.contractor, samples.sample_type, samples.testing_type, samples.qty, samples.unit_price,
32              receiving.submitted_by, receiving.remarks, receiving.actual_payment
33              FROM transactions, receiving, samples
34              WHERE transactions.receiving_id=receiving.receiving_id
35                AND transactions.date=receiving.date
36                AND receiving.sample_id=samples.sample_id
37                AND transactions.sales_flag=0";
38      $result=mysqli_query($con, $query) or die("Error : " .mysqli_error($con));
39
40      while($row=mysqli_fetch_array($result, MYSQLI_ASSOC))
41      {
42          $attributes[]=array('id' => $row['receiving_id'], 'date' => $row['date'], 'sample_id' => $row['sample_id'],
43           'lab_no' => $row['lab_no'], 'contractor' => $row['contractor'], 'sample_type' => $row['sample_type'],
44            'test' => $row['testing_type'], 'qty' => $row['qty'], 'unit_price' => $row['unit_price'],
45            'submitted_by' => $row['submitted_by'], 'remarks' => $row['remarks'], 'actual_payment' => $row['actual_payment']);
46      }
47  ?>
```

TRANSACTIONS | SEARCH RESULTS — Date | Lab Number | Contractor — Search... — Submit

CURRENT DATE: 2018-05-23

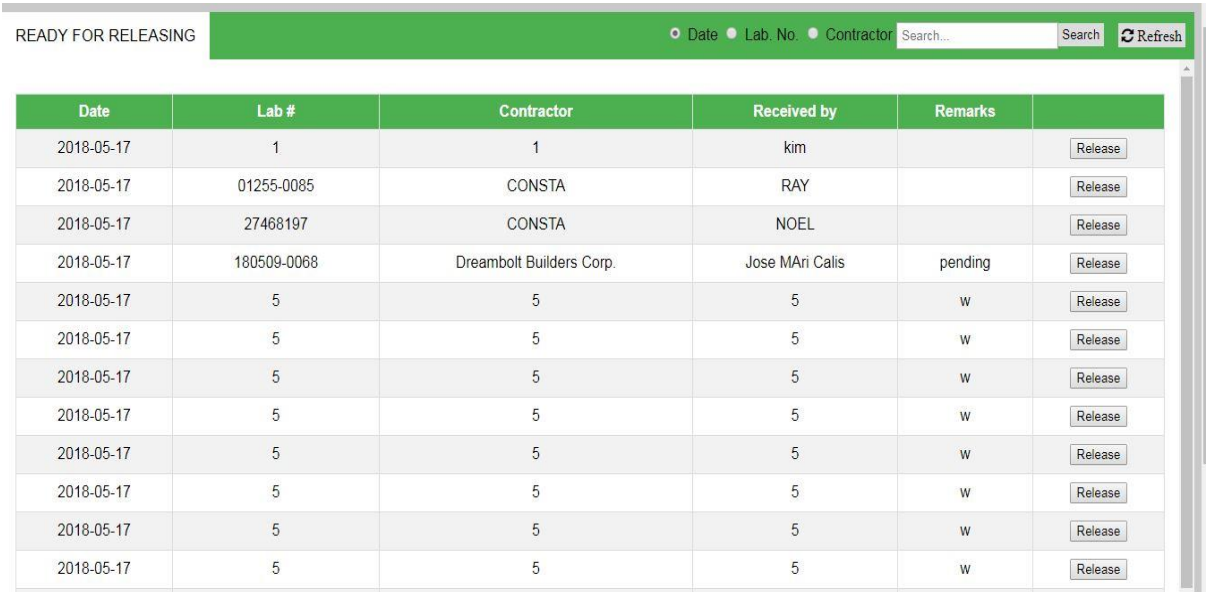
Date Received	Lab #	Contractor	Kind of Sample	Type of Test	Actual Payment	Paid	
2018-05-17	180509-0070	Disney Builders	CHB 6	flexural	1050.00		
2018-05-17	01255-0085	Consta	CEMENT	strength	1500.00		
2018-05-17	12	12	12	12	144.00		
2018-05-17	1231231242441	234234234	12313124	2342434	2890999168460352.00		
2018-05-17	23	5	5	5	25.00		
2018-05-17	5	5	5	5	25.00		
2018-05-17	5	5	5	5	25.00		
2018-05-17	5	5	5	5	25.00		
2018-05-17	5	5	5	5	25.00		
2018-05-17	5	5	5	5	25.00		
2018-05-17	5	5	5	5	25.00		

Figure 13 index.php of accounting

Figure 13 shows code snippets of index.php file of accounting folder. It demonstrates how the file retrieves all data from database to be display in accounting user page. Accounting folder consists of subfolder shown on figure on right which has its own purpose. Search folders contains files that generate search results and payment folder consists of files that calculates payment of transaction.

CSS		2018-05-16 16:38:00	drwxr-xr-x
edit		2018-05-16 22:11:00	drwxr-xr-x
Images		2018-05-17 14:13:00	drwxr-xr-x
payment		2018-04-30 03:06:00	drwxr-xr-x
search		2018-05-11 13:36:00	drwxr-xr-x
About.html	5.0 kB	2018-05-16 16:36:00	-rw-r--r--
choose.php	0.4 kB	2018-05-13 21:55:00	-rw-r--r--
index.php	6.7 kB	2018-05-19 16:34:00	-rw-r--r--

Releasing


READY FOR RELEASING

Date ○ Lab. No. ○ Contractor Search... Search Refresh

Date	Lab #	Contractor	Received by	Remarks	
2018-05-17	1	1	kim		Release
2018-05-17	01255-0085	CONSTA	RAY		Release
2018-05-17	27468197	CONSTA	NOEL		Release
2018-05-17	180509-0068	Dreambolt Builders Corp.	Jose MAri Calis	pending	Release
2018-05-17	5	5	5	w	Release
2018-05-17	5	5	5	w	Release
2018-05-17	5	5	5	w	Release
2018-05-17	5	5	5	w	Release
2018-05-17	5	5	5	w	Release
2018-05-17	5	5	5	w	Release
2018-05-17	5	5	5	w	Release
2018-05-17	5	5	5	w	Release

Figure 14 shows code snippets from index.php file of receiving folder. This is where it redirects ones a user successfully logged in as releasing account. Code snippets demonstrate how data are being fetch in the system.

```
20  if(!mysqli_select_db($con, "id5312484_mtci_davao")){
21      echo "Unable to locate the desired database.";
22      exit();
23  }
24
25  /* table operations begin */
26  $tquery= "SELECT releasing_id, date_received, samples.lab_no,
27              samples.contractor, received_by, date_released,    release_flag, remarks
28              FROM releasing
29              INNER JOIN samples ON releasing.sample_id = samples.sample_id
30              WHERE releasing.release_flag=0";
31  $result=mysqli_query($con, $tquery) or die("Error : " .mysqli_error($con));
32  $attributes=array();
33
34  while($row = mysqli_fetch_array($result, MYSQLI_ASSOC)){
35      $attributes[] = array('releasing_id' => $row['releasing_id'], 'date_received' => $row['date_received'],
36                  'lab_no' => $row['lab_no'], 'contractor' => $row['contractor'],
37                  'received_by' => $row['received_by'], 'date_released' => $row['date_released'],
38                  'release_flag' => $row['release_flag'], 'remarks' => $row['remarks']);
39  }
40  /* table operations end */
41
42  ?>
```

Figure 14 index.php of releasing

Manager

Manager contains 3 various indexes for daily receive, daily transaction and daily released. The following files are important since manager needs to track daily activities of the system.

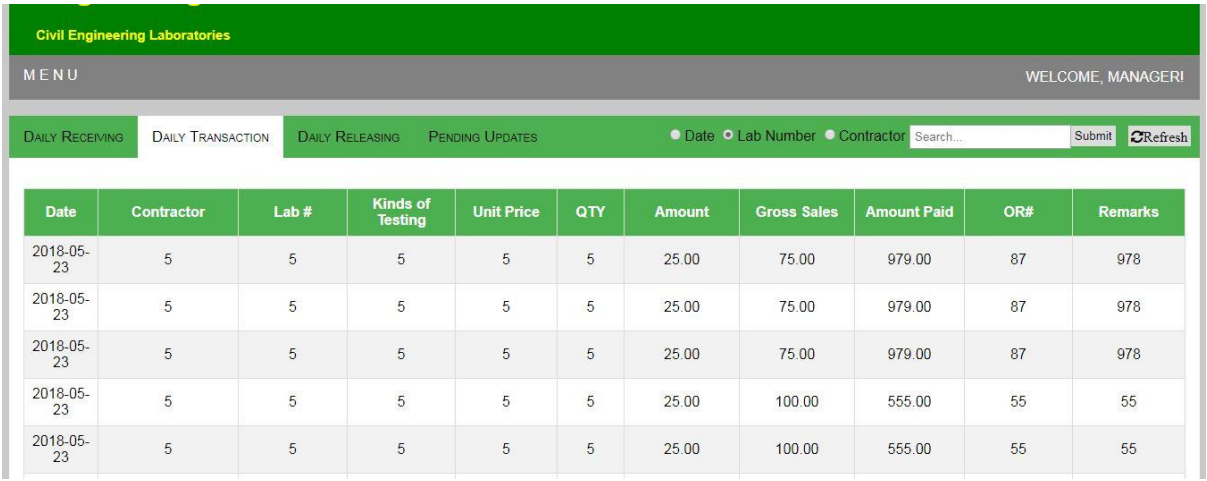
Daily Receiving


```
30  //query for getting the attributes of user
31  $query="SELECT receiving.date, samples.lab_no, samples.contractor,
32  samples.sample_type, samples.qty, samples.testing_type, samples.unit_price,
33  receiving.actual_payment, receiving.submitted_by, receiving.remarks FROM receiving,
34  samples WHERE receiving.sample_id=samples.sample_id AND receiving.date=CURDATE()";
```

Figure 15 managerdailyreceiveindex.php code snippet

Figure 15 shows the query for data retrieval in daily receive table of manager. Manager is only concerned with daily received that is why the query checks whether the date of received is the current date.

Daily Transactions



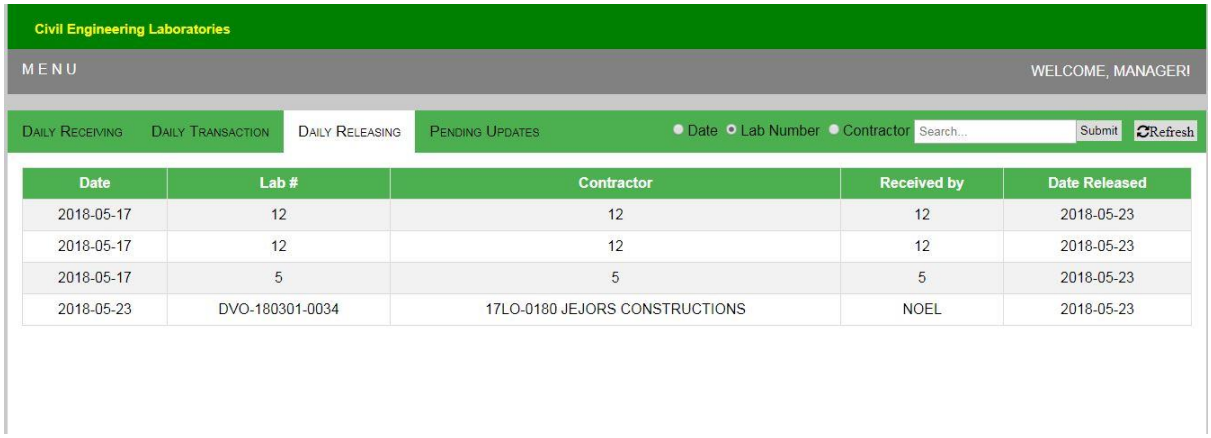
Date	Contractor	Lab #	Kinds of Testing	Unit Price	QTY	Amount	Gross Sales	Amount Paid	OR#	Remarks
2018-05-23	5	5	5	5	5	25.00	75.00	979.00	87	978
2018-05-23	5	5	5	5	5	25.00	75.00	979.00	87	978
2018-05-23	5	5	5	5	5	25.00	75.00	979.00	87	978
2018-05-23	5	5	5	5	5	25.00	100.00	555.00	55	55
2018-05-23	5	5	5	5	5	25.00	100.00	555.00	55	55

```
31  //query for getting the attributes of user
32  $query="SELECT samples.lab_no, samples.contractor, samples.testing_type,samples.qty,
33   samples.unit_price, transactions.amount, paid.gross_sales, paid.rebates,
34   paid.with_tax, paid.amount_paid, paid.or_no, paid.billing_no, paid.remarks
35      FROM transactions, samples, receiving, paid
36      WHERE paid.date=CURDATE() AND transactions.receiving_id=receiving.receiving_id
37       AND receiving.sample_id=samples.sample_id
38      AND transactions.paid_id=paid.paid_id AND transactions.sales_flag=1
39       ORDER BY transactions.paid_id ASC";
40
41  $result=mysqli_query($link,$query);     //executes the query
42
43  if(!$result)
44  {
45      echo "Error fetching attributes " . mysqli_error($link);
46      exit();
47  }
```

Figure 16 managerdailytransactionIndex.php code snippet

Figure 16 shows the query for data retrieval in daily sales table of manager. Manager is only concerned with daily transactions that is why the query checks whether the paid date is equal to the current date.

Daily Released




```
31  //query for getting the attributes of user
32  $query="SELECT releasing_id, date_received, samples.lab_no, samples.contractor,
33  received_by, date_released,    release_flag
34              FROM releasing
35              INNER JOIN samples ON releasing.sample_id = samples.sample_id
36              WHERE releasing.release_flag=1 AND date_released=CURDATE() ";
37
38  $result=mysqli_query($link,$query);     //executes the query
39
40  if(!$result)
41  {
42      echo "Error fetching attributes " . mysqli_error($link);
43      exit();
44  }
```

Figure 17 dailyreleasingIndex.php

Figure 17 shows the query for data retrieval in daily released table of manager. Manager is only concerned with daily released that is why the query checks whether the date released is the current date.

Software Limitations

Since the software is implemented on a cloud-based server using 000webhost as an online webserver, slower system loading is expected an is totally dependent of the internet service provider. The system can also access when there exists an internet connection.

Maintenance Guide

Few pointers to guide future developers in maintaining the system.

Database

Due to limited capacity of database, database should be regularly checked and maintain. One way to handle errors with regards to maximum capacity is eliminate too old transactions or transactions that are no longer needed by the system. This could be done through database inspection. Another way if elimination is not possible is to avail higher database capacity of the host server.

Connectivity

Make sure computers are connected to an internet service provider and has fast internet speed to fully maximize the usage of the system.

Future Improvement and Innovations

Localization

The software could be implemented in a local server. This demands servers and other necessary architectures but is more efficient because it does not solely rely on the internet connection.

Multi-Branching of the System

Since Megatesting Center Inc. consists of multiple branches nationwide, an upgraded management inventory system that could handle all branches would be a better implementation. This will help the CEO of the company track branches activity without visiting the actual branch.