**Online Pharmacy Portal (Group 1)**

**Deliverables 5 Report**

**For**

**Online Pharmacy Portal – MedAnytime**

**Group 1**

**Version 1.0 approved**

**Prepared by – Soujanya Akella**

**Gayatri Sravani Akella**

**Nikithanjali Dodla**

**Chahnitha Chandu**

**University of North Texas, Denton**

**18th November 2019**

**Table of Contents**

1. Requirements and specifications ………………………………………….. 3
2. UML design ……………………………………………………………….. 3
3. Test cases ………………………………………………………………….. 5
4. Unit Testing and System testing …………………………………………... 6
5. Workload Division ………………………………………………………… 8
6. User manual and compilation instructions ………………………………… 8
7. Initial requirements and current scenario comparison …………………….. 12
8. Member contribution table ………………………………………………… 13
9. **Requirements and specifications**

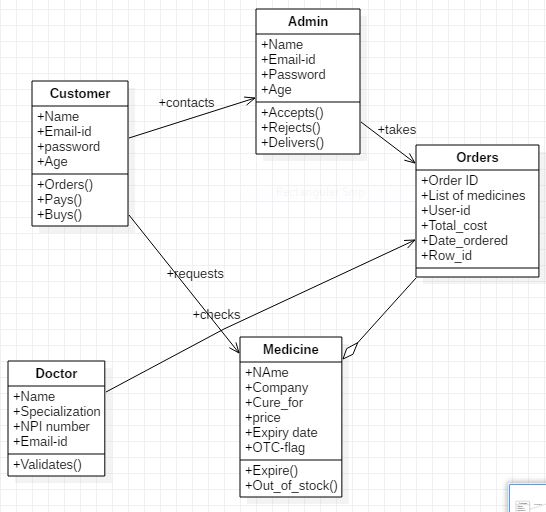
The requirements in this phase of development are such that we complete all the requirements promised in the initial requirement specification document to the best of our capabilities. For more specifics on the requirements, we have succeeded to implement, refer section 7.

The requirements implemented in the third phase of development for MedAnytime are,

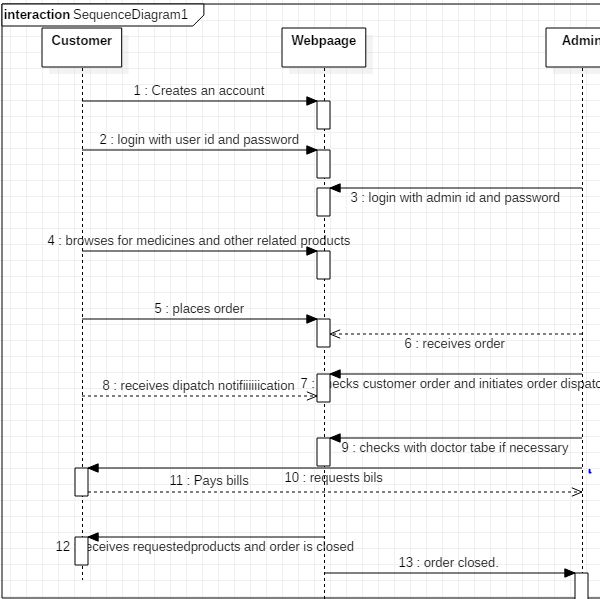
* Payment gateway for all transactions on user side.
* Emails sent to user every time there is any crucial change or action made.
* Unit testing
* System testing

These requirements are not the ones mentioned in deliverables 5. They had to be changed because our requirements were improperly mentioned in the second report and they did not come in the order of development to facilitate iterative development model.

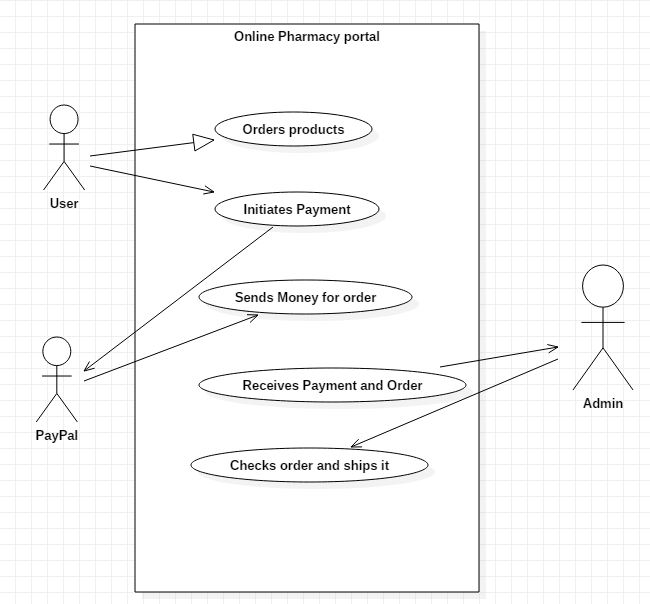
1. **UML Design**
   1. **Class Diagram**

****

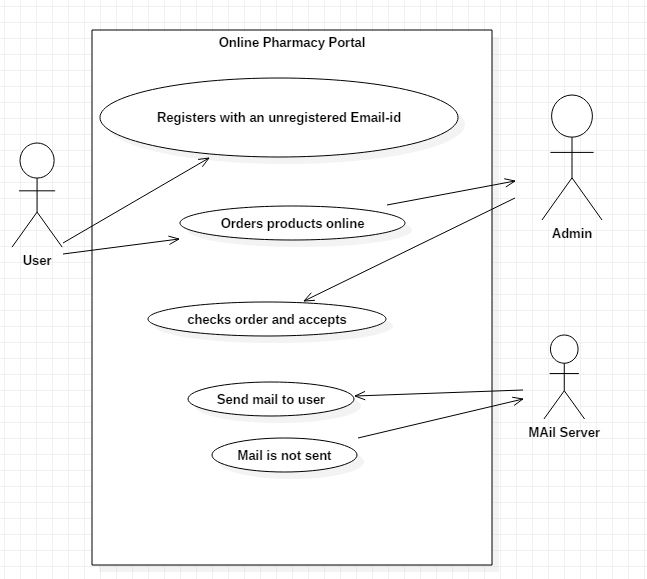
* 1. **Sequence Diagram**

****

* 1. **Use case (normal case)**

****

* 1. **Use case (Error case)**

****

1. **Test Cases**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **S.No** | **Requirement** | **Test priority** | **Test steps & Pre-conditions** | **Expected Result** | **Obtained Result** | **Success / Failure** | **Tester** |
| 1 | User sends order | Medium | User should have an account and products in cart. | User should receive a mail that an order has been placed. | User has received an email. | Success | Gayatri |
| 2 | Admin finds a prescription to be invalid and declines an order | High | User should have placed an order with a prescription. Admin declined. | User should receive a mail that his/her order is rejected. | User has received an email | Success | Soujanya |
| 3 | User enters shipping address and clicks on save | High | User should have an order ready for checkout. | User should be asked for PayPal account details | PayPal account details are requested | Sucess | Chahnitha |
| 4 | User enters PayPal details | High | User should check out an order and PayPal account has sufficient balance | The payment should be received by admin | Admin has received a payment. | Success | Nikitha |
| 5 | User enters PayPal details | Medium | User should check out an order and there is an error in payment. | The payment should be cancelled and order must remain in cart | Payment is Cancelled | Success | Gayatri |

1. **Unit Testing and System Testing**
   1. **Unit Testing**

**Test for payment**

<?php

require\_once('success.php');

use PHPUnit\Framework\TestCase;

class TestPayment extends TestCase

{

protected function setUp()

{

parent::setUp();

$\_GET = array();

}

public function tearDown(){ }

public function testpay()

{

$tpay = new paymentSuccess;

$\_GET = array(

'amt' => '8.00',

'cc' => 'USD',

'item\_name' => 'zzzquil',

'st' => 'Completed',

'tx' => '0U2290575H3589605',

'submit' => 1

);

$\_SESSION ['id'] = 'mpalkar@gmail.com';

$\_REQUEST = 'POST';

$this->assertTrue($tpay->testPaymentSuccess($\_GET,$\_SESSION, $\_REQUEST) == 1);

}

}

?>

**Test for Mailing**

<?php

require\_once('success.php');

use PHPUnit\Framework\TestCase;

class TestMail extends TestCase

{

protected function setUp()

{

parent::setUp();

$\_GET = array();

}

public function tearDown(){ }

public function testpay()

{

$mailCheck = new paymentSuccess;

$\_POST = array(

'tNum' => '5dd372f865268'

);

$\_SESSION['id'] = 'aarthisingh19399@gmail.com';

$this->assertTrue($mailCheck->testSendMail($\_POST,$\_SESSION) == 1);

}

}

?>

* 1. **System Testing**

System testing is a type of testing in the Software development process that is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements. In our project, we perform all of the essential tests also as part of system testing. The essentials tests are integration testing and functional testing. We have put together all the components for the webpage and checked their working together individually in multiple systems. In our observation, the page seems to be working well. All the promised functionalities are still intact. None of the components have performed differently in the integrated system.

1. **Workload Division**

All the work was equally divided and performed by all team members. All of them put in equal efforts. The tasks performed by each respective member are mentioned below:

* PayPal Integration (By Gayatri)
* User account payments to admin (By Soujanya)
* Email Integration (By Nikitha)
* Email sent to user for order related activities (By Chahnitha)
* Unit Testing (Gayatri & Nikitha)
* System Testing (Soujanya & Chahnitha)
* Documentation & Presentation (Gayatri, Soujanya, Nikitha and Chahnitha)

1. **User Manual and Compilation Instructions**

The compilation and installation instructions still remain the same in this phase as well.

Just for the sake of ease, I will mention them again. The installation process for this webpage requires us to have some software on the system in which we want to run the page on. Here, we make an assumption that the user has already downloaded the project folder onto his/her system before starting the installation process.

* Firstly, we need a working version of the XAMPP control panel (we have used XAMPP 3.2.4).
* Next we need to start apache service and the MySql service in the XAMPP control panel.
* Please place the downloaded project folder in the following folder on your system.

**C: /xampp/htdocs**

* Now, open a compatible web browser, preferably Google chrome. Now type the url: localhost:**port\_number/SwEngg/loginnewphp**
* Now, if all the components are working correctly, the login page should be displayed. This means, the webpage is now ready and available for use.
* If not, the user might want to check his mysql connection or any other necessary component that might not be running correctly. (In most cases, it is displayed on the browser itself that some component is not working correctly).
* Admin credential- **Username:** [Medanytime@gmail.com](mailto:Medanytime@gmail.com)

**Password:** Gaya3

* User credentials-  **Username:** [mgrey@gmail.com](mailto:mgrey@gmail.com)

**Password:**  Meredith

Instructions to run test cases: There is a separate folder called ‘Test project’. To be able to run the test cases, the user should first install ‘composer’ in this folder and run the test cases in this folder using command line. The command for the same is ‘ **.\vendor\bin\phpunit TestFileName.php**’

These are the various test cases implemented in this phase. They can be modified to test the different aspects of the respective functionalities.

TestMail (To test the email functionality when all arguments are proper)  
TmailNegative (To test the email functionality when tracking number is not found)  
TestPayment (Test the payment functionality when all arguments are proper)

TestPaymentNegative (test the payment functionality when it doesn’t work)

checkDocDetailsTestNegative (To check if doctor is registered with us)

checkDocDetailsTestPositive (To check if doctor is registered with us)

productupdateTest (Tests the product update functionality)

productAddTest (tests the product addition functionality)

loginTest (tests the login functionality)

generateTrackingNumberTest (Tests if a tracking number is generated)

productAddTestNegative (tests the product addition functionality)

TestAddToCartNegative (Test if products are added into the cart)

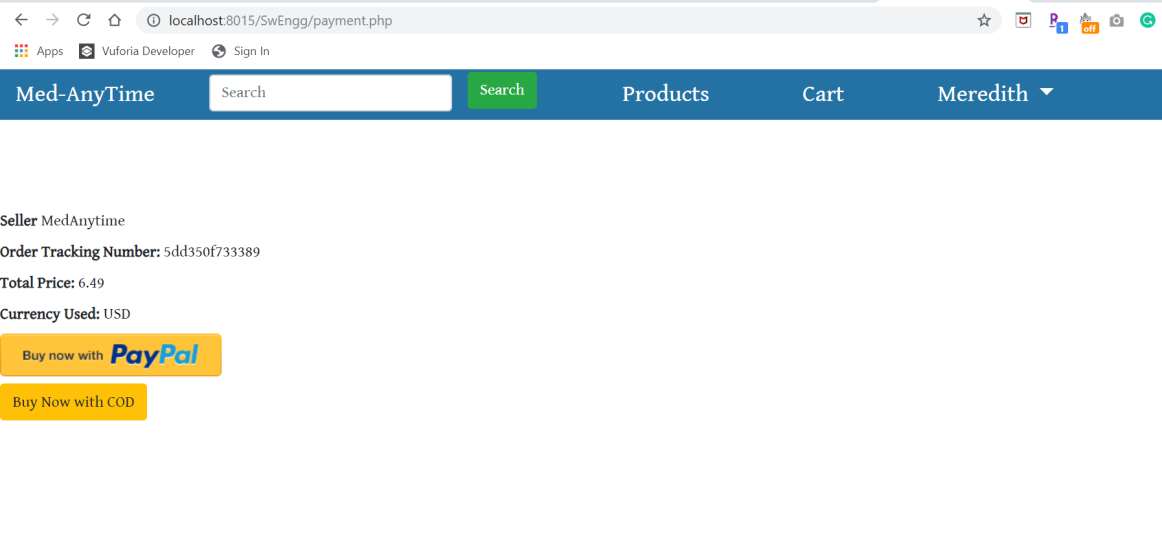
TestAddToCartPositive (Test if products are added into the cart)

registrationTest (Tests the registration functionality)

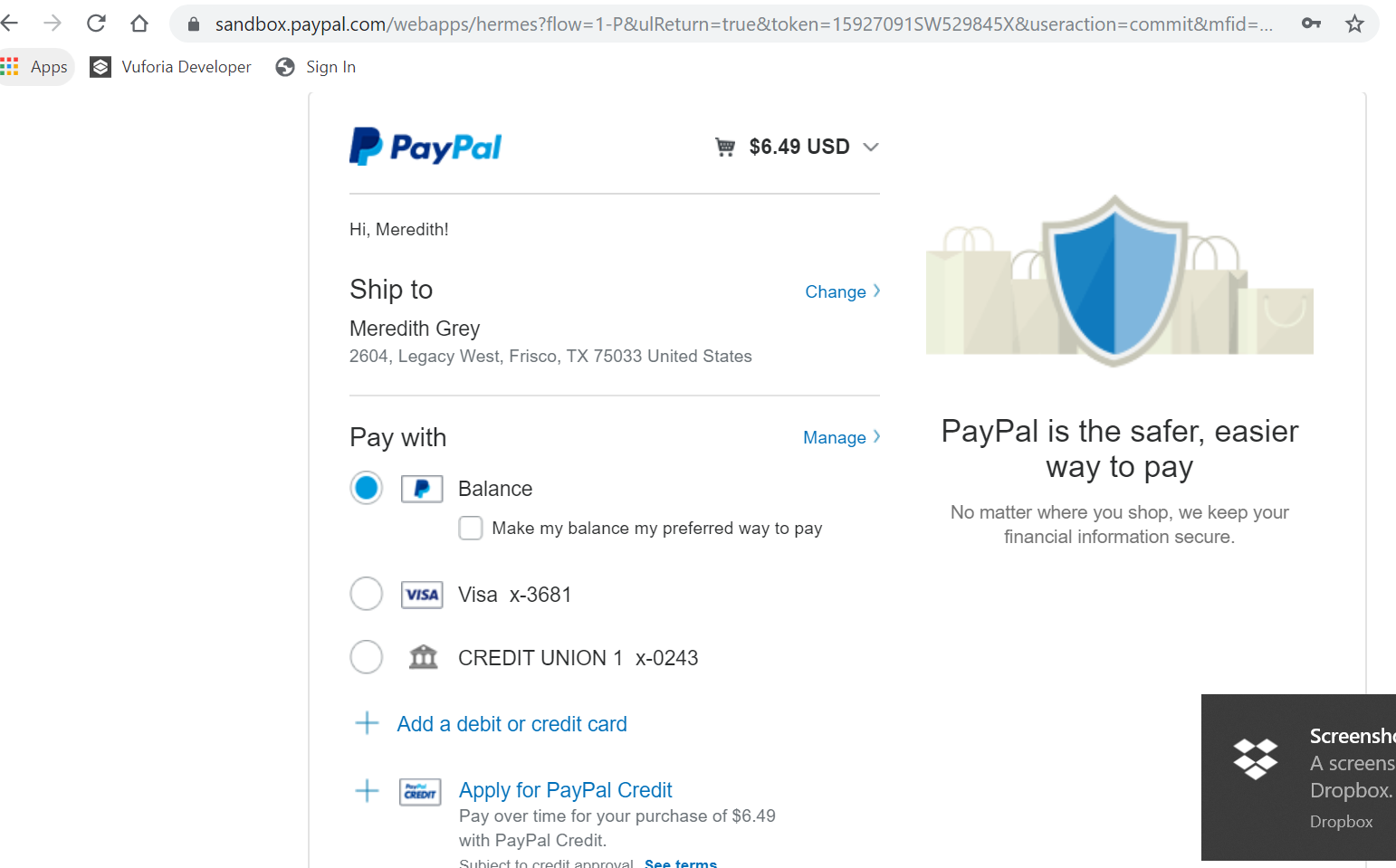
**User Instructions:**

The user instructions are only a continuation of what was given in the earlier reports. So far, we have given instructions on how to register, login to their respective user pages. User can then search for products by product name or illness. Next there were instructions for adding products to their cart, checking out to complete order and entering shipping address. On the admin side, we have seen instructions on how to search, add/delete or update products, and check orders and reject them if feels necessary.

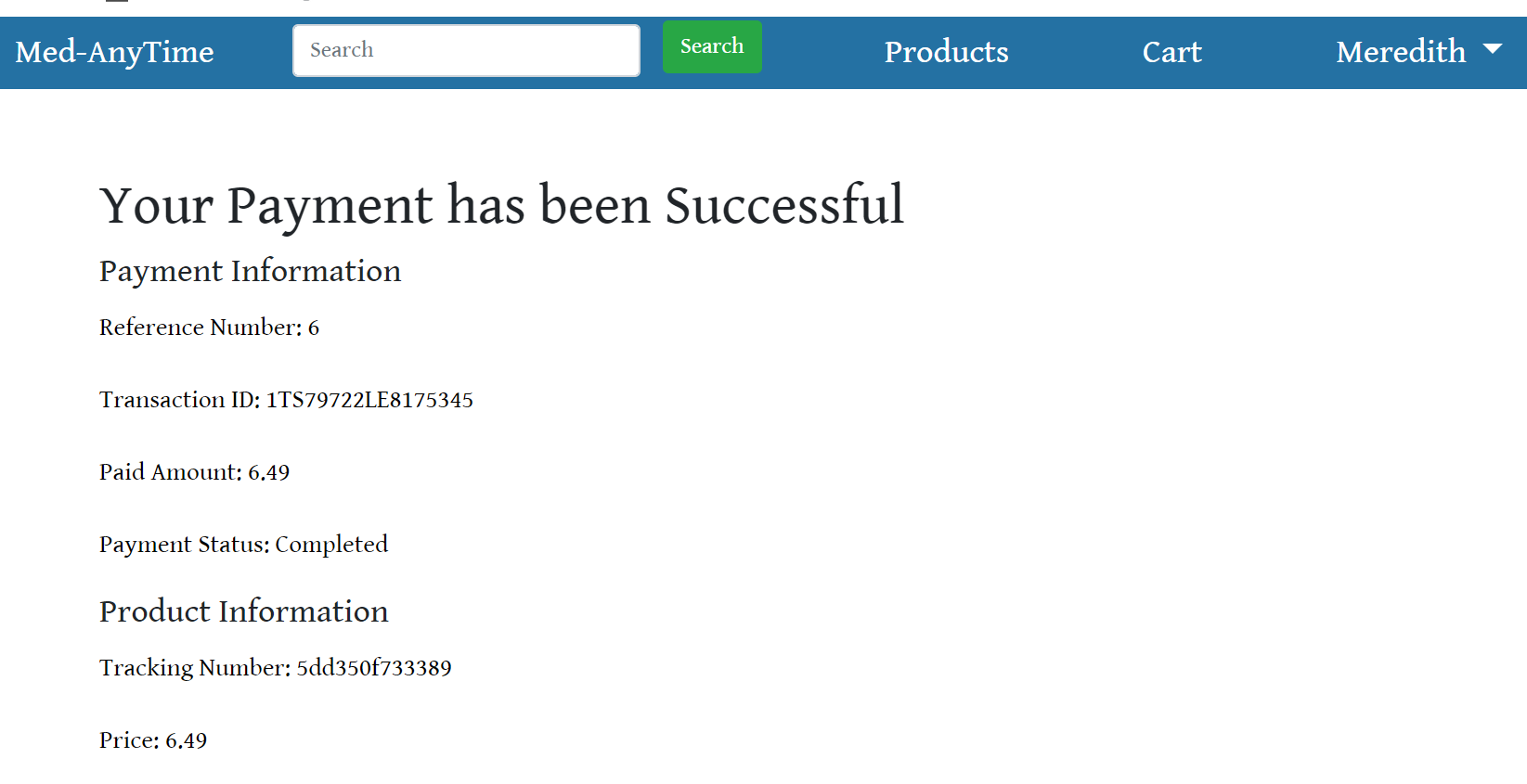
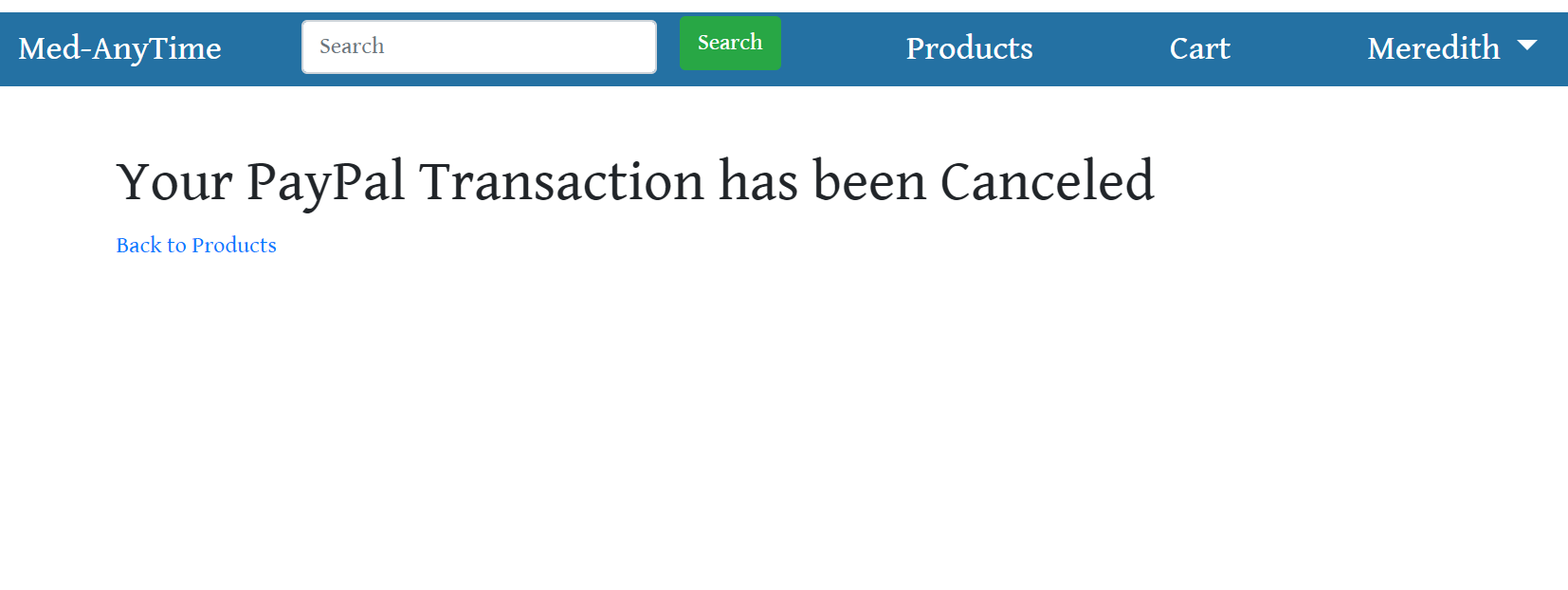
Now, we only have further instructions on how user can add payment details. As mentioned earlier, payment is done through PayPal. After the user enters address, and gives doctor details (if needed), they will be redirected to a page where they are asked for their PayPal account details. Even now, the user can choose to order and pay for the products with COD (Cash On Delivery). In this case, they will have to pay for their order when it is delivered to them.



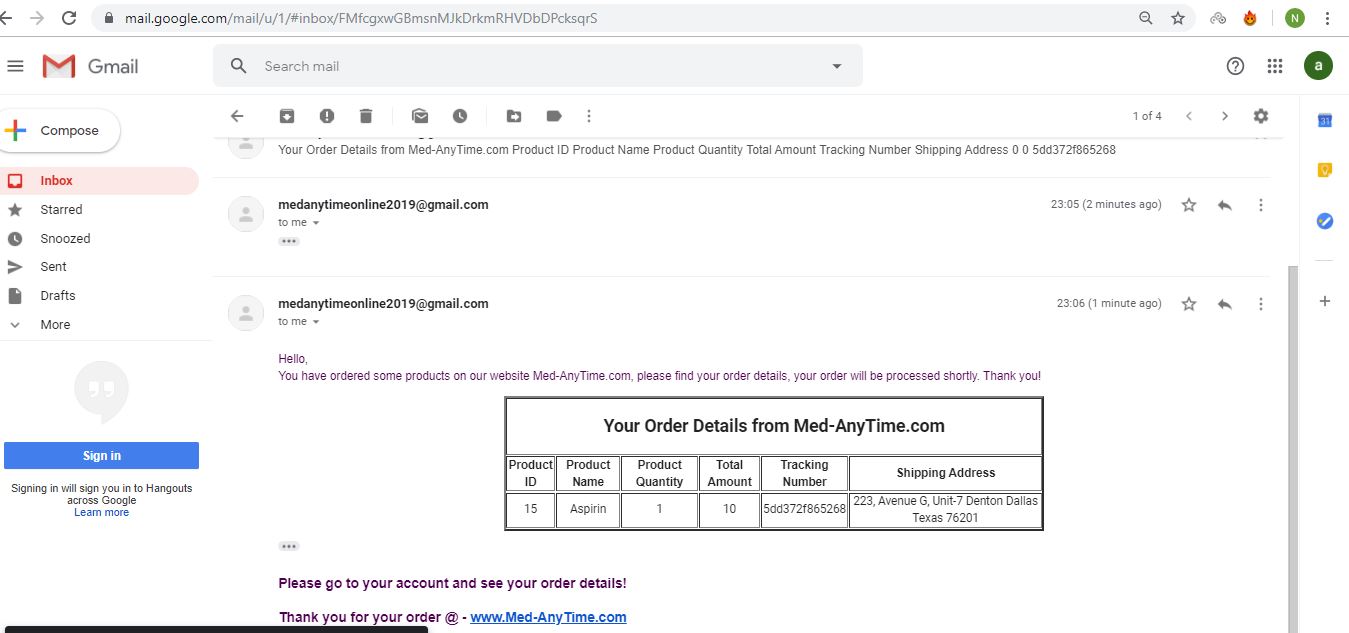
Later, the amount to be paid is directly cut form their account and is sent to the seller, who in our case is the admin. In that case, user will see a screen which says payment is successful.



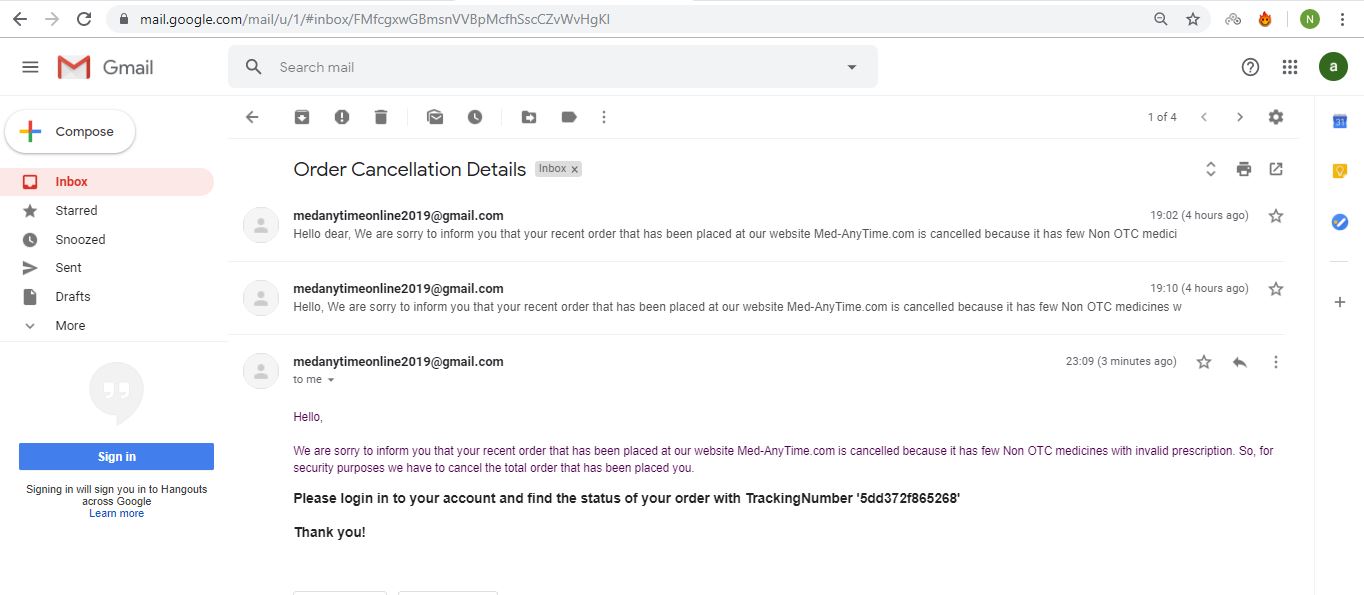
Suppose the user attempts to pay with a PayPal account and incurs a payment error, the payment is cancelled and the product they tried to order remains in the cart.

There is nothing to be done to by the user to get the emails. It is an automated process. The user just needs to make sure they have an active email-ID on their account. An email is sent to the user in either of two situations. First if there is any order placed from their account successfully. In this case, the mail simply says that we have orders from your account on MedAnytime. After this, if gives a description of products ordered.



Otherwise, the user receives a mail when the admin declines an order that was placed from their respective account. In this case, the mail says that their order contains non-OTC items that were ordered with an invalid prescription. So their order was declined.



1. **Initial requirements and current scenario**

Initially we had planned on developing a web page that allows its users to shop for pharmaceutical products online while still sticking to the laws of drug trade.

We have succeeded in developing the page that has separate pages for users and the admin. It displays and variety of products and medicines being sold. It asks for a prescription whenever a medicine is ordered and also payment is processed as soon as the order is placed. Payment is done through PayPal only. On the admin side, admin is able to add, update and delete products, view orders and reject them if something is found to be dodgy. Emails are sent to user whenever required about any changes in their accounts or certain important notifications.

The application we have come up with right now is a mere prototype. It will still not work in real-time as the whole application is working on Xampp and is on the localhost. Because we are using Xampp, and hosting this on the localhost, the product changes and orders will reflect in the same system only. So that is the first limitation of our project. There are a few other issues that are currently not yet resolved that we will work on in the future. These issues include refunding the bill total to the customer when their order is either cancelled or rejected. Also, tracking of each individual order is still out of scope for the project.

Other developments that we can make in this application are: Adding a ‘forgot password’ button for the users in case they forget their passwords. Also, facilitate more than one admin to be able to manage the page because, currently it is too much work for one person.

So, I would say that our project is about 95% successful seeing as how we have implemented almost everything we promised initially.

1. **Member Contribution Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Member Name** | **Contribution Description** | **Overall contribution** | **Note**  **(If applicable)** |
| Gayatri Sravani Akella | Owner of the project repository. Creation of database, development of inventory page, Database manager for admin account, Cart functionality for user. PayPal integration for payments. Testing and documentation. | 25% |  |
| Soujanya Akella | Development of login page and registration page, Validations for registering users, products add/delete in admin page, Development of page that asks for address and doc details at checkout from cart, generation of tracking number. PayPal integration for payments. Documentation and testing | 25% |  |
| Chahnitha Chandu | Configuration of Xampp server, Development of admin page, Page to page navigation, search functionality for admin and user. Testing and documentation. | 25% |  |
| Nikithanjali Dodla | Development of user page, products display in user page, Database changes whenever admin makes any changes, Notify admin when order is placed by user, orders page in user side. Testing and documentation | 25% |  |