

Software Design Description Document For INGAZ for printing service

Nura Mostafa , Nourhan Atef, Seif ElDein Mohamed, Omar Shereef
CEO:Tamer Abdelwahed

March 2020

GitHub: <https://github.com/OmarShereef/INGAZ-for-print>

Version	Date	Description
1.0	16-May-2020	First Submission
1.1	9-May-2020	Introduction.
1.2	10-May-2020	Context viewpoint
1.3	12 -May-202	Upload Class Diagram,Requirements Matrix
1.4	14 -May-202	Human Interface Design
2.0	13-June-2020	Second Submission,
2.1	21-May-2020	Updating Requirements Matrix
2.2	1-June-2020	updating Logical Diagram, and updating data description
2.3	3-June-2020	DefinitionsAndAcronyms.

Table 1: Document version history

Contents

1	Introduction	3
1.1	The Purpose	3
1.2	Scope	3
1.3	Overview	3
1.4	Definitions and Acronyms	3
2	Overview of the project	4
2.1	The Scope of the project	4
3	Goals and objectives	4
3.1	Timeline of the project	5
4	Context viewpoint	5
5	System Architecture	6
5.0.1	Logical Diagram	6
5.1	Viewpoint of the composition	7
5.2	Viewpoint of the Structure	8
5.3	Viewpoint of the Algorithm	9
5.4	Viewpoint of the Interaction	10
6	Data Design	11
6.1	Description of the Data	11
6.2	Description of the database design	12
7	Human Interface Design	12
7.1	User Interface	12
7.2	Screen Images	13
8	Matrix of the Requirements	15

1 Introduction

1.1 The Purpose

The purpose of this document is to provide a full description of how the Ingaz system works. The Ingaz system is an online web-based application system to provide a high-quality printing service to benefit our clients where this service is not available in Egypt. This software design document (SDD) will describe the aim of the system and its functionalities. In addition, the document will show all constraints on the system, all contraventional interface designs, and all diagrams that were needed to build the system.

1.2 Scope

The document targets the clients, employees, and companies administration which has a role in the companies business flow .

1.3 Overview

This document describes most of the system diagrams and architectures. it also previews how the system main functionalities work and how the user views and interacts with the software. The sections in this document gives a detailed description for the diagrams that help the developer developing the system.It includes the class diagrams , sequence diagrams and architecture diagrams.

1.4 Definitions and Acronyms

Term	Definition
MVC	Model,view,controller designing pattern.
Singleton design pattern	The Simple database design pattern that has one object instance by implementing such a class database.

2 Overview of the project

The system's main target is to provide a service that has only been available internationally to Egyptians. The system allows the client to create, customize, and print flyers, business cards, and notebooks with simple clicks that make this service all over Egypt easier. [2]

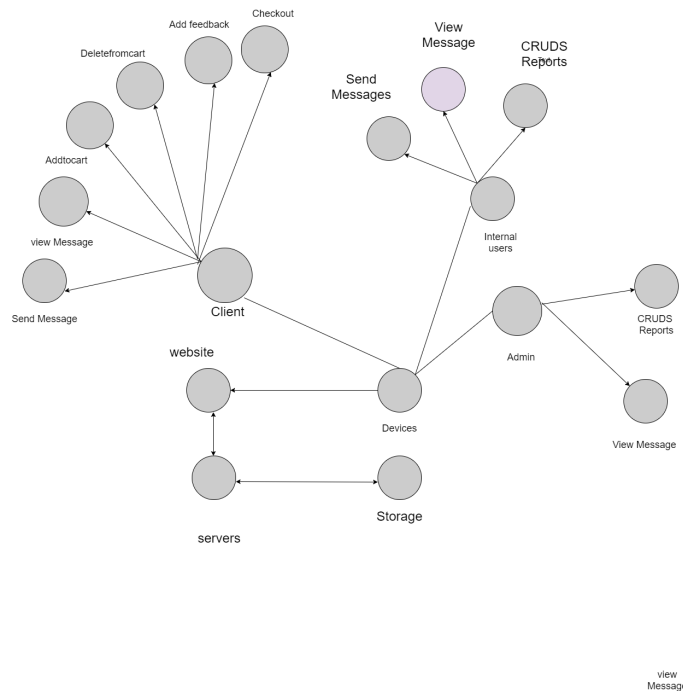


Figure 1: Project overview diagram
[4]

2.1 The Scope of the project

This system aims to provide a high quality printing service to benefit our clients. It is expected to present the clients' designs in the best way possible to help them reach their goals and visions.

3 Goals and objectives

In terms of the issue of visiting the printing store, our system offers a delivery service that delivers all over Egypt [1]. It also less time consuming and will be effective for people in places that are far from the printing store.

3.1 Timeline of the project



Figure 2: Gantt chart diagram

4 Context viewpoint

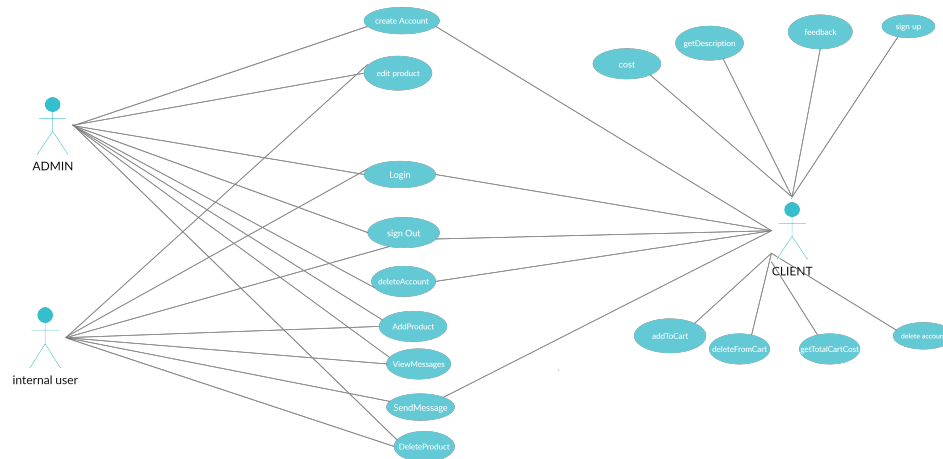


Figure 3: Use Case Diagram
[3]

5 System Architecture

5.0.1 Logical Diagram

In Inagz system, we utilize the MVC design which contains three sections are the model, the view, the controller. every one of this part is mindful to accomplish something and they rely upon one another. The Model contains only the unadulterated application data, it contains no method of reasoning delineating how to acquaint the data with a user. The View presents the model's data to the customer. The view acknowledges how to get to the model's data, anyway it doesn't have the foggiest thought what this data suggests or what the customer can do to control it. The Controller exists between the view and the model. In our system, there are 3 subsystems as the administrator in which he can do anything in the framework and the inside clients can CRUD and send and see messages to and from customers lastly the customer who can make their own item and send and see messages to

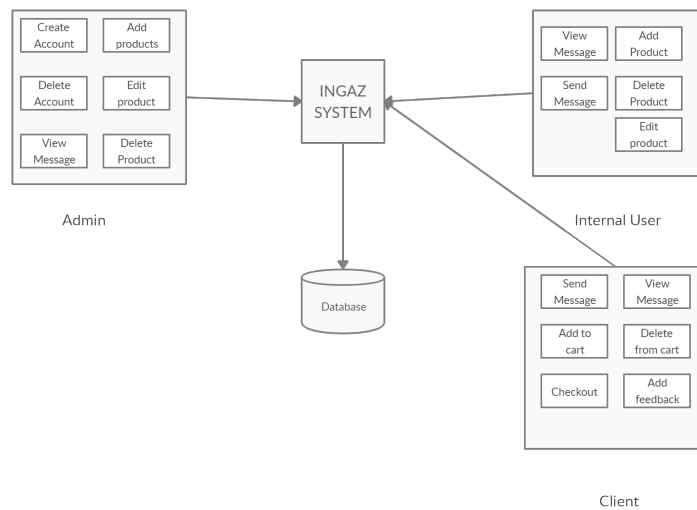


Figure 4: Logical diagram

5.1 Viewpoint of the composition

Firstly Admin can CRUD (Add,delete, edit) anything from the system and view the messages between the admin and the internal users. then internal users also can add , delete, edit products and send and view message to and from the client.Lastly Client customizes his own product and proceed to checkout to know the total cost of all in the cart.

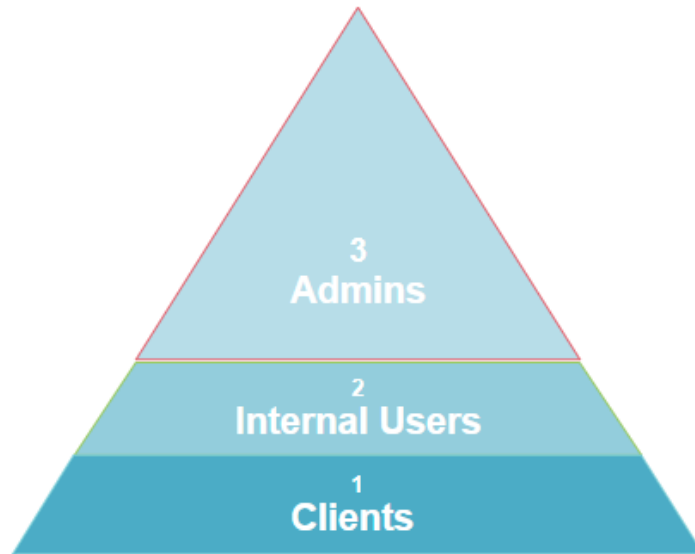


Figure 5: Hierarchical Diagram
[5]

5.2 Viewpoint of the Structure

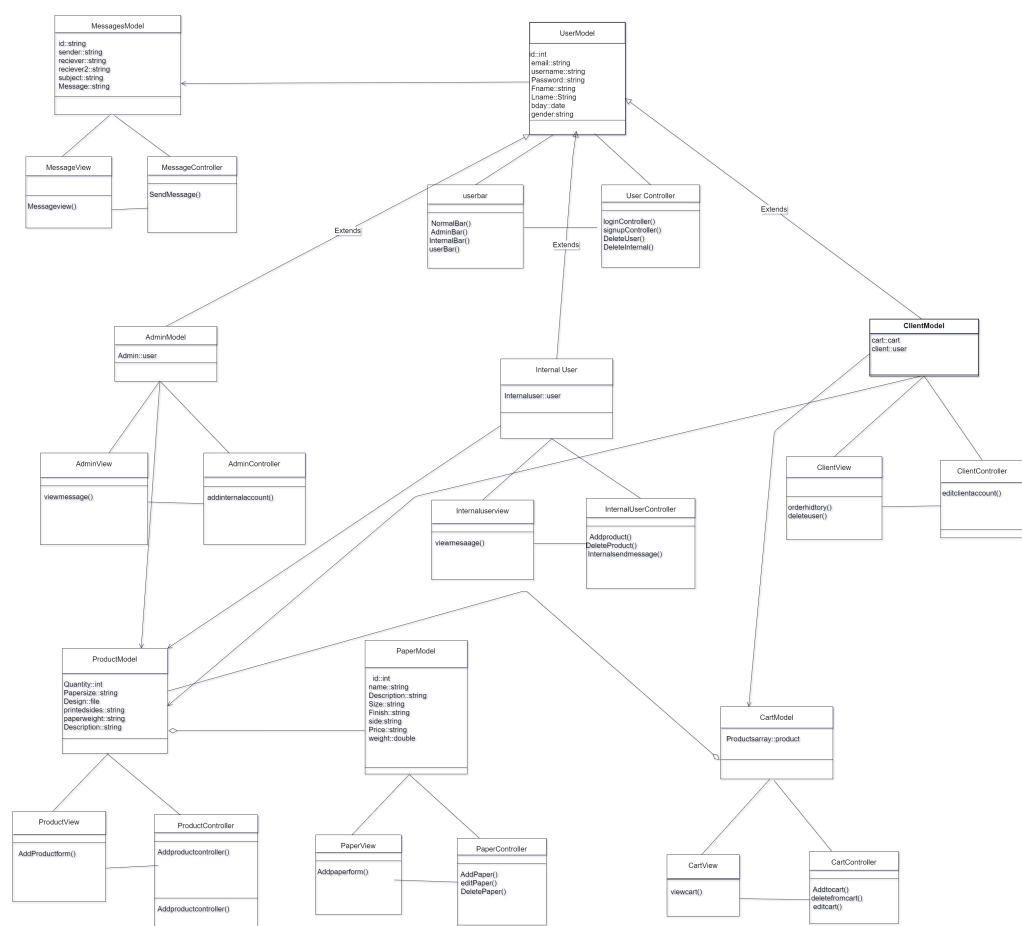


Figure 6: Class Diagram

5.3 Viewpoint of the Algorithm

```
public function editclientaccount()
{
    $id = $_SESSION['id'];
    $username = $_SESSION['username'];
    $password = $_SESSION['password'];
    $Fname = $_SESSION['Fname'];
    $Lname = $_SESSION['Lname'];
    echo $id;
    echo $username;
    $this->model->editacc($id,$username,$password,$Fname,$Lname);
}
```

Figure 7: Edit account function

```
function getUsers()
{
    $sql = "SELECT id,username,password,Fname,Lname FROM users WHERE access='user' ";
    $dbh = new Dbh();
    $result = $dbh->query($sql);
    $users = array();
    while($row=mysqli_fetch_assoc($result))
    {
        $user = new client();
        $user->setusername($row['username']);
        $user->setpassword($row['password']);
        $user->setFname($row['Fname']);
        $user->setLname($row['Lname']);
    }
    $users[] = $user;
    return $users;
}
```

Figure 8: GetUsers function

```

public function login($username,$password)
{
    $username = $_POST["username"];
    $password = $_POST["password"];
    $sql="Select *from users where username ='$username' and password='$password'";
    $dbh = new Dbh();
    $result = $dbh->query($sql);
    $row=$dbh->fetchRow();
    if (empty($row))
    {
        if($row['access']=='admin')
        {
            $_SESSION['access']='admin';
        }
        if($row['access']=='internal')
        {
            $_SESSION['access']='internal';}
        if($row['access']=='user')
        {
            $_SESSION['access']='user';}
        echo "login successfully";
    }
    else
    {
        echo "Invalid Email or Password";
        $_SESSION['id']=$row['id'];
        $_SESSION['email']=$row['email'];
        $_SESSION['username']=$row['username'];
        $_SESSION['password']=$row['password'];
        $_SESSION['fname']=$row['fname'];
        $_SESSION['lname']=$row['lname'];
        $_SESSION['birthday']=$row['bday'];
        $_SESSION['gender']=$row['gender'];
        echo"<pre>";
        print_r($sql);
        echo"</pre>";
        echo"<pre>";
        print_r($row);
        echo"</pre>";
    }
}

```

Figure 9: Login function

```

function signup($email,$username,$password,$fname,$lname,$birthday,$gender)
{
    $password=password_hash($_POST['password'], PASSWORD_BCRYPT);
    $sql="insert into users(email,username,password,fname,lname,bday,gender,access) values('".$_POST['email']."','".$_POST['username']."','".$password."','".$_POST['fname']."','".$_POST['lname']."','".$_POST['bday']."','".$_POST['gender']."'
    ','user')";
    $sql1 = "SELECT id, email, username FROM users WHERE email ='".$_POST['email']."' or username='".$_POST['username']."'";
    $dbh = new Dbh();
    $dbh2=new Dbh();
    if($dbh2->query($sql1) == true)
    {
        while($row=$dbh2->fetchRow())
        {
            if($row['email']==$_POST['email'])
            {
                echo "<script>alert('Email Already Taken');</script>";
            }
            if($row['username']==$_POST['username'])
            {
                echo "<script>alert('Username Already Taken');</script>";
            }
        }
    }
    if($dbh->query($sql) == true)
    {
        echo "Signup successfully";
    }
}
}

```

Figure 10: Signup function

5.4 Viewpoint of the Interaction

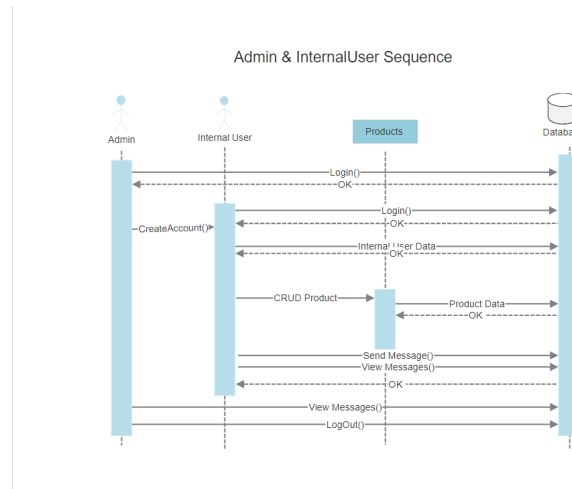


Figure 11: Admin Internal User diagram

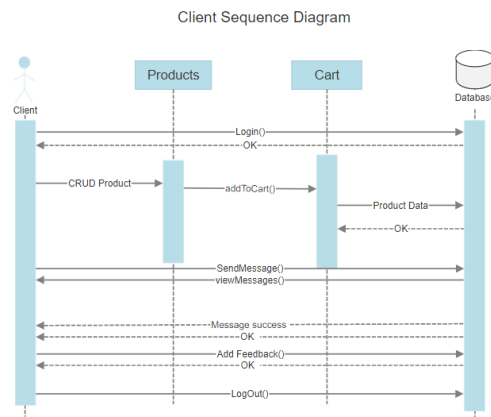


Figure 12: Client diagram

6 Data Design

6.1 Description of the Data

Before Creating this framework, the printing office was paper based, take subtleties with papers. This framework permits them to play out these activities with web application utilizing forms. The framework can include numerous clients. Database contains 5 tables. Tables clients and items are the primary tables which fundamentally rest of tables acquire from them. Likewise we use (Bcrypt) as hashing strategy to hash passwords for the security of the client.

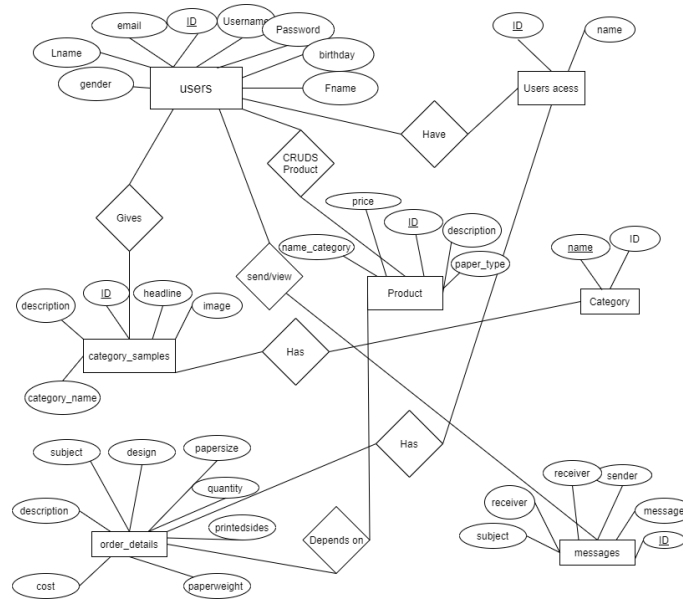


Figure 13: ER Diagram

6.2 Description of the database design

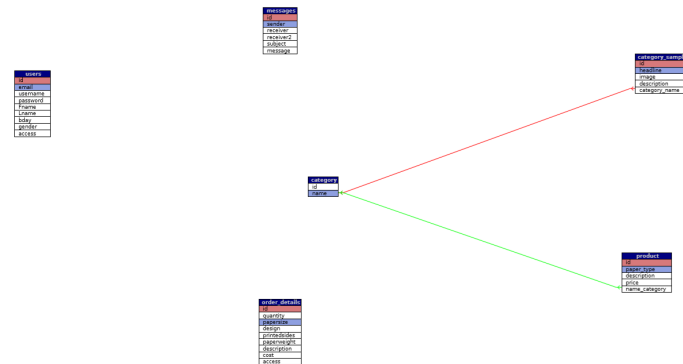


Figure 14: Database design Diagram

7 Human Interface Design

7.1 User Interface

In our Ingaz system when a client opens the web page he is automatically redirected to the homepage. He is then required to choose the product that he wishes to print. He will then start to customize the product in terms of his chosen design, the style, the size and type along with several other characteristics if required. Then the client will chose the requested delivery time from a chart with the prices depending on the delivery time. Click on "add to basket", at this stage the client has two options whether to go back and shop again for other products or to proceed to the checkout page. At the checkout stage the client is required to either log in or sign up into the website to finalise his purchase transaction and get a receipt.

7.2 Screen Images

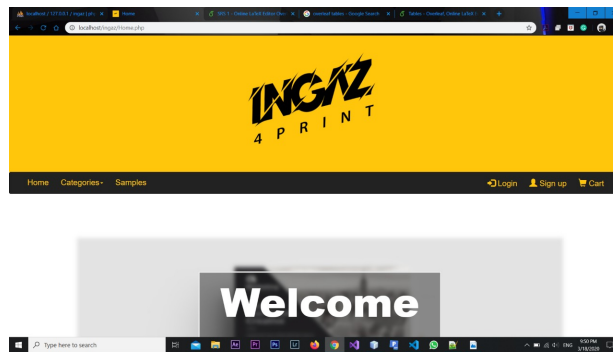


Figure 15: home page

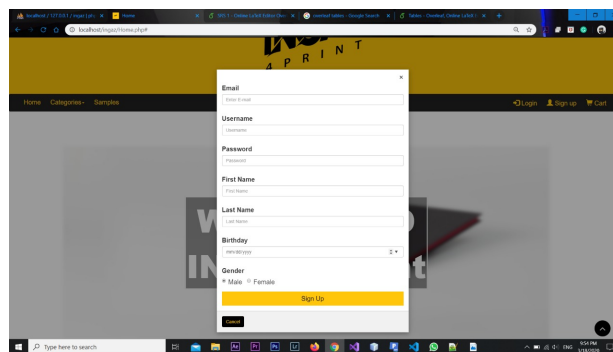


Figure 16: Signup form

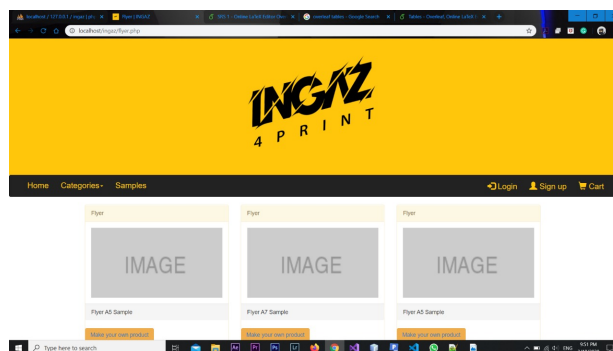


Figure 17: Product categories page

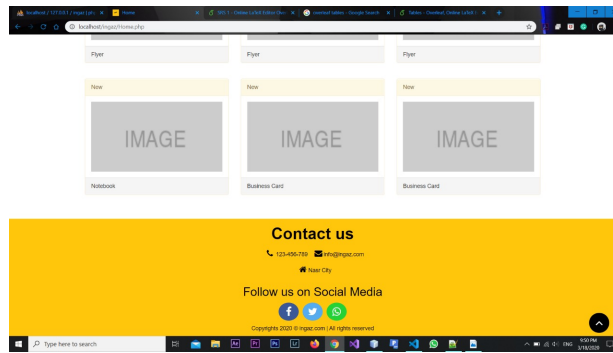


Figure 18: Product categories page continued

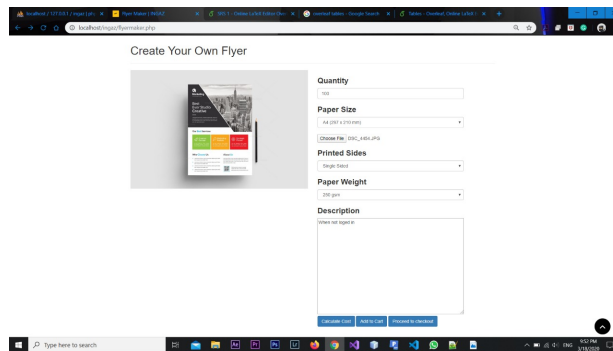


Figure 19: Product customization page (Not logged in)

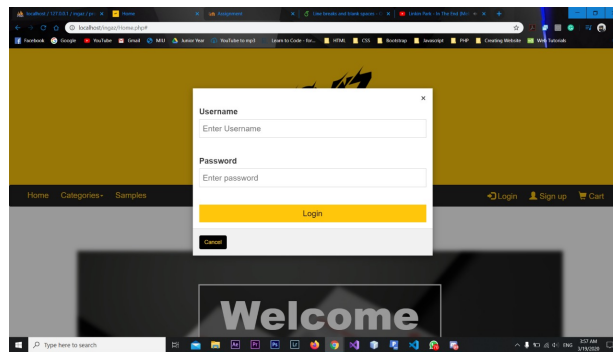


Figure 20: Login form

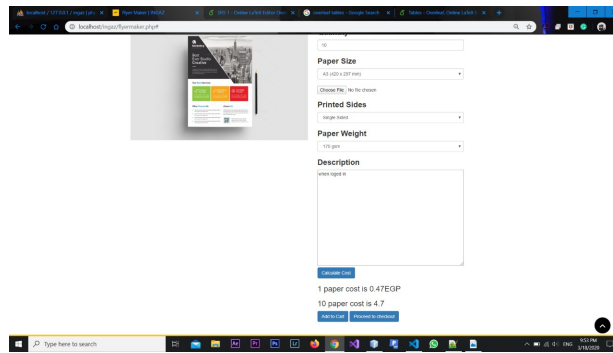


Figure 21: Product customization page (Logged in)

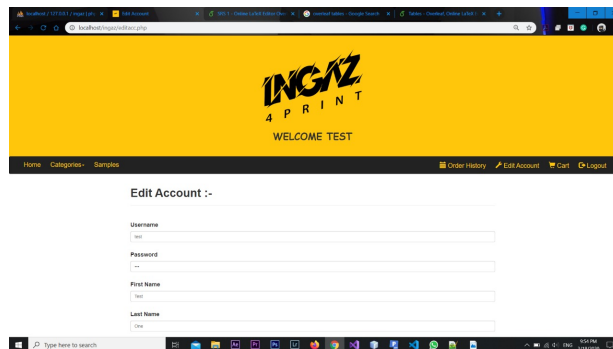


Figure 22: Edit Account page

8 Matrix of the Requirements

Requirement ID	Requirement Name	Requirement Description	Status
INGAZ01	Signup	Clients sign up to buy products.	completed
INGAZ02	Login	Users login to profile with username and password .	completed
INGAZ03	Logout	users logout from their profile .	completed
INGAZ04	AddAccount	The admin creates account for internal users.	completed
INGAZ05	DeleteAccount	Admin can delete internal users and client accounts but clients can delete their accounts.	completed
INGAZ06	editAccount	Admin can edit internal users and client accounts but clients can edit their accounts.	completed
INGAZ07	SendMessage	clients and internal users can send messages to each other.	completed
INGAZ08	ViewMessage	client:: view the messages sent from internal users internal users and Admin:: can view all messages .	completed
INGAZ09	AddFeedback	Clients should be able to add their feedback of the web-site and it's services.	In progress
INGAZ10	AddProduct	The internal user and admin can add product.	completed
INGAZ11	EditProductName	Admin and internal users can edit products which can change the description or the name of it.	completed
INGAZ12	DeleteProduct	The internal users and admin can delete products.	completed
INGAZ13	AddToCart	Clients save the product to their cart.	completed
INGAZ14	DeleteFromCart	clients can delete from cart.	completed
INGAZ15	AddPaper	Admin and internal users can add papers which insert the paper description	completed
INGAZ16	EditPaper	Admin and internal users can edit papers which can change the dpaper description .	completed
INGAZ17	DeletePaper	Admin and internal users can delete papers .	completed
INGAZ18	insertCategory	The internal users and admin can insert new category .	completed
INGAZ19	deleteCategory	The internal users and admin can delete category.	completed
INGAZ20	editCategory	The internal users and admin can edit category.	completed

Table 2: Requirements Matrix table

References

- [1] Aron Priest Andy Smith. *Online Printing Services by Solopress*. 1999. URL: <https://www.solopress.com/>.
- [2] Aldrin Bhunu. “An investigation into the adoption and implementation of an online business network platform for the printing industry and the associated benefits”. In: (2018).
- [3] Cinergix. *CREATELY APP*. 2008. URL: <https://app.creately.com/diagram/H3rTrFI5GQU/edit>.
- [4] Oliver Zahrt Joachim Seibert Martin Seibert. *DRAW.IO FOR DIAGRAMS*. 1994. URL: <https://app.diagrams.net/>.
- [5] *SMART DRAW*. 1994. URL: <https://cloud.smartdraw.com/>.