

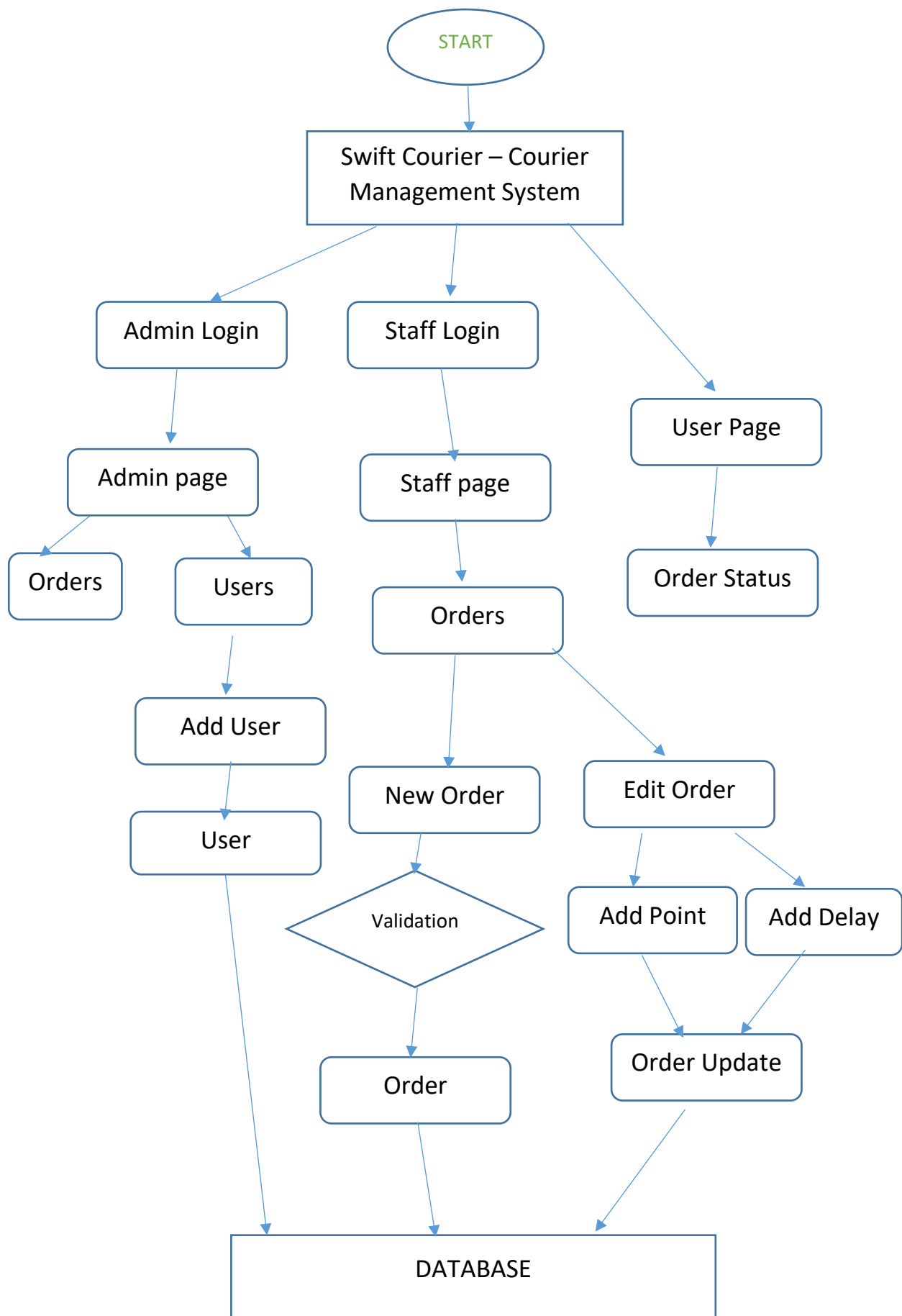
SYSTEM DESIGN

INDRODUCTION TO SYSTEM DESIGN

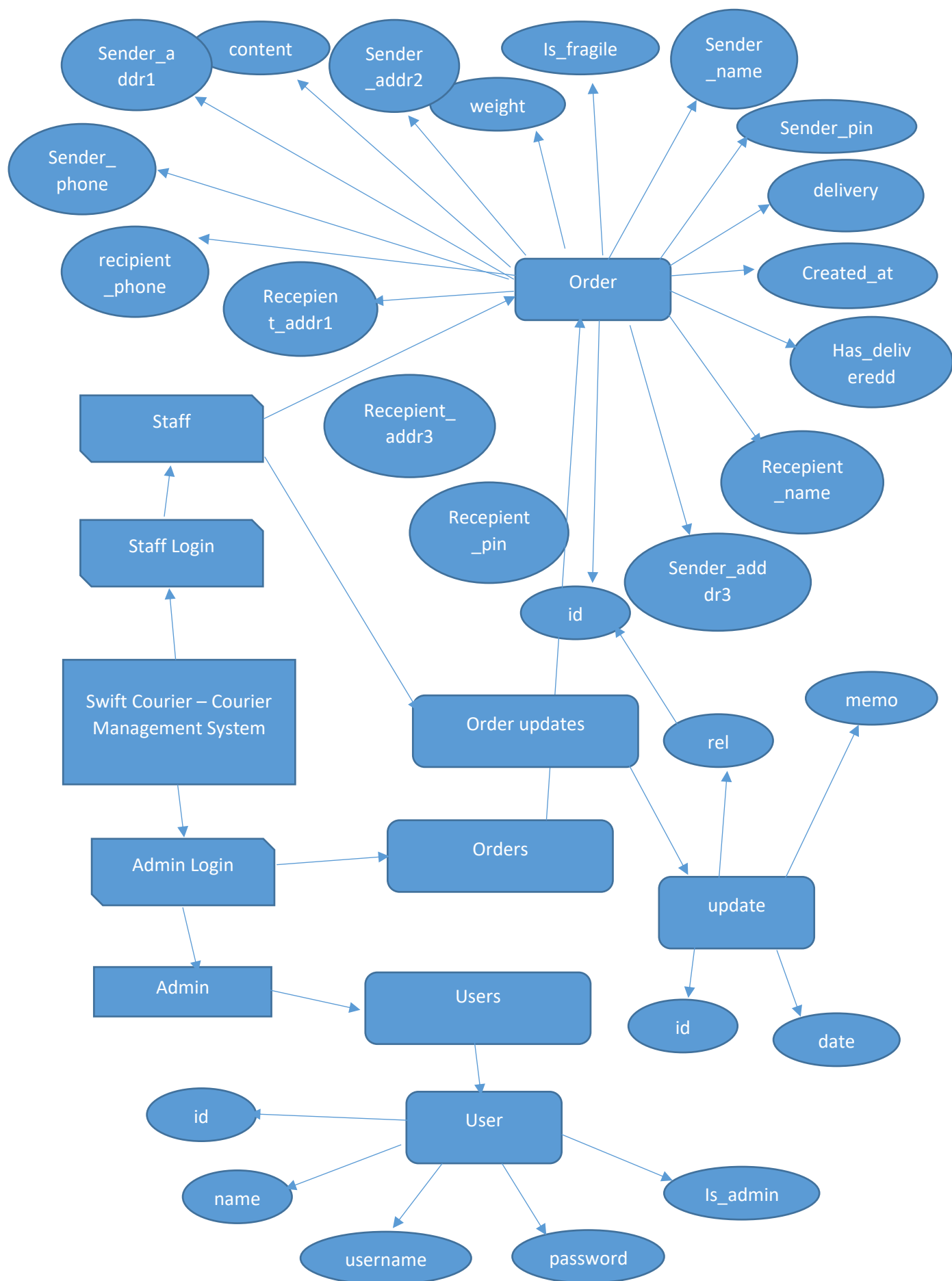
The design phase is the life cycle phase in which the detailed design of the selected system in the study phase is accomplished. In the design phase, the technical specifications are prepared for the performed of all allocated tasks. It also includes the construction of programs and programs testing. In the design phase, the first step is to determine the output is to produced and in what format.

Systems design is the process of defining elements of a system like modules, architecture, components and their interfaces and data for a system based on the specified requirements. It is the process of defining, developing and designing systems which satisfies the specific needs and requirements of a business or organization.

DATA FLOW DIAGRAM



ER DIAGRAM



DATABASE TABLES

User Table (users)

Column Name	Data Type
id	int
name	varchar
username	varchar
password	varchar
Is_admin	bool

Order Table (orders)

Column Name	Data Type
id	int
content	varchar
weight	Int
amount	Int
Is_fragile	Bool
Sender_name	varchar
Sender_phone	Varchar
Sender_addr1	Varchar
Sender_addr2	Varchar
Sender_addr3	Varchar
Sender_pin	Varchar
recipient_name	Varchar
recipient _phone	Varchar
recipient _addr1	Varchar
recipient _addr2	Varchar
recipient _addr3	Varchar
recipient _pin	Varchar
created_at	datetime
delivery	datetime
has_delivered	bool

Order Updates Table (order_updates)

Column Name	Data Type
id	int
date	datetime
memo	varchar
rel	int

CODE

Main Form – Form1.vb

Imports MySql.Data.MySqlClient

Public Class Form1

Public DbConnection As MySqlConnection

Public LoggedInusername As String

Public SelectedId As Integer

Public AddAdminPage As AddAdmin

Public AdminLoginPage As AdminLogin

Public AdminPage As Admin

Public StaffLoginPage As StaffLogin

Public StaffPage As Staff

Public UserPage As User

Public UserOrderDetailsPage As UserOrderDetails

Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load

InitDbConnection()

LoadInitialPage()

End Sub

Private Sub InitDbConnection()

Dim ConnectionString =

"server=127.0.0.1;uid=root;pwd=admin;database=swiftcourier"

DbConnection = New MySqlConnection With {

.ConnectionString = ConnectionString

}

' Create tables

DbHelper.CreateTables()

End Sub

Private Sub LoadInitialPage()

UserModule.IsUsersExists()

If UserModule.UserExists = True Then

LoadUserPage()

Else

LoadAddAdminPage()

End If

End Sub

Public Sub LoadAddAdminPage()

AddAdminPage = New AddAdmin With {

.TopLevel = False,

.FormBorderStyle = FormBorderStyle.None

}

AppPanel.Controls.Clear()

AppPanel.Controls.Add(AddAdminPage)

AddAdminPage.Show()

End Sub

Public Sub LoadAdminLoginPage()

AdminLoginPage = New AdminLogin With {

.TopLevel = False,

.FormBorderStyle = FormBorderStyle.None

}

```

        AppPanel.Controls.Clear()
        AppPanel.Controls.Add(AdminLoginPage)
        AdminLoginPage.Show()
    End Sub

    Public Sub LoadAdminPage()
        AdminPage = New Admin With {
            .TopLevel = False,
            .FormBorderStyle = FormBorderStyle.None
        }

        AppPanel.Controls.Clear()
        AppPanel.Controls.Add(AdminPage)
        AdminPage.Show()
    End Sub

    Public Sub LoadStaffLoginPage()
        StaffLoginPage = New StaffLogin With {
            .TopLevel = False,
            .FormBorderStyle = FormBorderStyle.None
        }

        AppPanel.Controls.Clear()
        AppPanel.Controls.Add(StaffLoginPage)
        StaffLoginPage.Show()
    End Sub

    Public Sub LoadStaffPage()
        StaffPage = New Staff With {
            .TopLevel = False,
            .FormBorderStyle = FormBorderStyle.None
        }

        AppPanel.Controls.Clear()
        AppPanel.Controls.Add(StaffPage)
        StaffPage.Show()
    End Sub

    Public Sub LoadUserPage()
        UserPage = New User With {
            .TopLevel = False,
            .FormBorderStyle = FormBorderStyle.None
        }

        AppPanel.Controls.Clear()
        AppPanel.Controls.Add(UserPage)
        UserPage.Show()
    End Sub

    Public Sub LoadUserDetailsPage()
        UserOrderDetailsPage = New UserOrderDetails With {
            .TopLevel = False,
            .FormBorderStyle = FormBorderStyle.None
        }

        AppPanel.Controls.Clear()
        AppPanel.Controls.Add(UserOrderDetailsPage)
        UserOrderDetailsPage.Show()
    End Sub
End Class

```

Add Admin Page – AddAdmin.vb

```
Public Class AddAdmin
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
        Dim Name = NameInput.Text
        Dim Username = UsernameInput.Text
        Dim Password = PasswordInput.Text

        AdminModule.AddAdmin(Name, Username, Password)
    End Sub
End Class
```

Admin Login Page – AdminLogin.vb

```
Public Class AdminLogin
    Private Sub GoToStaffLoginButton_Click(sender As Object, e As EventArgs)
Handles GoToStaffLoginButton.Click
        Form1.LoadStaffLoginPage()
    End Sub

    Private Sub LoginButton_Click(sender As Object, e As EventArgs) Handles
LoginButton.Click
        Dim Username = UsernameInput.Text
        Dim Password = PasswordInput.Text

        AdminModule.LoginAdmin(Username, Password)
    End Sub
End Class
```

Admin Page – Admin.vb

```
Public Class Admin
    Dim DashboardPage As AdminDashboard

    Dim OrdersPage As AdminOrders

    Dim UserManagementPage As AdminUserManagement
    Dim AddUserPage As AdminAddUser
    Private Sub Admin_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        LoadOrdersPage()
    End Sub

    Private Sub LoadDashboardPage()
        DashboardPage = New AdminDashboard

        AdminAppPanel.Controls.Clear()
        AdminAppPanel.Controls.Add(DashboardPage)
    End Sub

    Public Sub LoadOrdersPage()
        OrdersPage = New AdminOrders

        AdminAppPanel.Controls.Clear()
        AdminAppPanel.Controls.Add(OrdersPage)
    End Sub

    Public Sub LoadUserManagementPage()
        UserManagementPage = New AdminUserManagement
```

```

        AdminAppPanel.Controls.Clear()
        AdminAppPanel.Controls.Add(UserManagementPage)
    End Sub

    Public Sub LoadAddUserPage()
        AddUserPage = New AdminAddUser

        AdminAppPanel.Controls.Clear()
        AdminAppPanel.Controls.Add(AddUserPage)
    End Sub

    Private Sub DashboardLink_Click(sender As Object, e As EventArgs) Handles
DashboardLink.Click
        LoadDashboardPage()
    End Sub

    Private Sub OrdersLink_Click(sender As Object, e As EventArgs) Handles
OrdersLink.Click
        LoadOrdersPage()
    End Sub

    Private Sub UsersLink_Click(sender As Object, e As EventArgs) Handles
UsersLink.Click
        LoadUserManagementPage()
    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
        Form1.LoadUserPage()
    End Sub
End Class

```

Admin Orders Page – AdminOrders.vb

```

Public Class AdminOrders
    Private Sub AdminOrders_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
        RenderOrderDataInListView()
    End Sub

    Private Sub RenderOrderDataInListView()
        OrderModule.GetOrderData()

        Dim CurrentItem = 0
        For Each Order As OrderModelMini In OrderModule.Orders
            OrdersList.Items.Add(Order.Id)
            OrdersList.Items(CurrentItem).SubItems.Add(Order.SName)
            CurrentItem += 1
        Next

        CurrentItem = 0
        For Each Order As OrderModelMini In OrderModule.Orders
            OrdersList.Items(CurrentItem).SubItems.Add(Order.RName)
            CurrentItem += 1
        Next

        CurrentItem = 0
        For Each Order As OrderModelMini In OrderModule.Orders
            OrdersList.Items(CurrentItem).SubItems.Add(Order.RPhone)
            CurrentItem += 1
        Next
    End Sub
End Class

```

```

        CurrentItem = 0
        For Each Order As OrderModelMini In OrderModule.Orders
            OrdersList.Items(CurrentItem).SubItems.Add(Order.RPin)
            CurrentItem += 1
        Next

        CurrentItem = 0
        For Each Order As OrderModelMini In OrderModule.Orders
            OrdersList.Items(CurrentItem).SubItems.Add(Order.Weight)
            CurrentItem += 1
        Next

        CurrentItem = 0
        For Each Order As OrderModelMini In OrderModule.Orders
            If Order.Delivered Then
                OrdersList.Items(CurrentItem).SubItems.Add("Completed")
            Else
                OrdersList.Items(CurrentItem).SubItems.Add(Order.Delivery.ToString("dd-MM-yy"))
            End If
            CurrentItem += 1
        Next
    End Sub

End Class

```

Admin Users Page – AdminUserManagement.vb

```

Public Class AdminUserManagement
    Private Sub AdminUserManagement_Load(sender As Object, e As EventArgs)
        Handles MyBase.Load
            UserModule.LoadUserData()
            Dim CurrentItem = 0
            For Each User As UserModel In UserModule.Users
                UsersListView.Items.Add(User.Id)
                UsersListView.Items(CurrentItem).SubItems.Add(User.Name)
                CurrentItem += 1
            Next

            CurrentItem = 0
            For Each User As UserModel In UserModule.Users
                UsersListView.Items(CurrentItem).SubItems.Add(User.Username)
                CurrentItem += 1
            Next

            CurrentItem = 0
            For Each User As UserModel In UserModule.Users
                If User.IsAdmin Then
                    UsersListView.Items(CurrentItem).SubItems.Add("Admin")
                Else
                    UsersListView.Items(CurrentItem).SubItems.Add("Staff")
                End If
                CurrentItem += 1
            Next
        End Sub

        Private Sub GoToAddUserPageButton_Click(sender As Object, e As EventArgs)
            Handles GoToAddUserPageButton.Click
                Form1.AdminPage.LoadAddUserPage()
            End Sub
    End Class

```

```

    Private Sub UsersListView_Click(sender As Object, e As EventArgs) Handles
UsersListView.Click
        If (UsersListView.SelectedItems.Count > 0) Then
            MessageBox.Show(UsersListView.SelectedItems(0).SubItems(1).Text)
        End If
    End Sub
End Class

```

Admin Add User Page – AdminAddUser.vb

```

Public Class AdminAddUser
    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles
Button2.Click
        Dim Name = NameInput.Text
        Dim Username = UsernameInput.Text
        Dim Password = PasswordInput.Text
        Dim IsAdmin As Boolean

        If (IsAdminInput.Text = "Admin") Then
            IsAdmin = True
        Else
            IsAdmin = False
        End If

        UserModule.AddUser(Name, Username, Password, IsAdmin)
    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
        Form1.AdminPage.LoadUserManagementPage()
    End Sub
End Class

```

Staff Login Page – StaffLogin.vb

```

Public Class StaffLogin
    Private Sub GoToAdminLoginButton_Click(sender As Object, e As EventArgs)
Handles GoToAdminLoginButton.Click
        Form1.LoadAdminLoginPage()
    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
        Dim Username = UsernameInput.Text
        Dim Password = PasswordInput.Text

        StaffModule.StaffLogin(Username, Password)
    End Sub

    Private Sub GoToUserPageButton_Click(sender As Object, e As EventArgs)
Handles GoToUserPageButton.Click
        Form1.LoadUserPage()
    End Sub
End Class

```

Staff Page – Staff.vb

```

Public Class Staff

```



```

Dim DashboardPage As StaffDashboard

Public SelectedOrderId As Integer
Dim OrdersPage As StaffOrders
Dim NewOrderPage As StaffNewOrder
Dim EditOrderPage As StaffEditOrder
Dim AddPointPage As StaffAddPoint
Dim AddDelayPage As StaffAddDelay
Private Sub Staff_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    LoadOrdersPage()
End Sub

Private Sub LoadDashboardPage()
    DashboardPage = New StaffDashboard

    StaffAppPanel.Controls.Clear()
    StaffAppPanel.Controls.Add(DashboardPage)
End Sub

Public Sub LoadOrdersPage()
    OrdersPage = New StaffOrders

    StaffAppPanel.Controls.Clear()
    StaffAppPanel.Controls.Add(OrdersPage)
End Sub

Public Sub LoadEditOrderPage()
    EditOrderPage = New StaffEditOrder

    StaffAppPanel.Controls.Clear()
    StaffAppPanel.Controls.Add(EditOrderPage)
End Sub
Public Sub LoadAddPointPage()
    AddPointPage = New StaffAddPoint

    StaffAppPanel.Controls.Clear()
    StaffAppPanel.Controls.Add(AddPointPage)
End Sub
Public Sub LoadAddDelayPage()
    AddDelayPage = New StaffAddDelay

    StaffAppPanel.Controls.Clear()
    StaffAppPanel.Controls.Add(AddDelayPage)
End Sub
Public Sub LoadNewOrderPage()
    NewOrderPage = New StaffNewOrder

    StaffAppPanel.Controls.Clear()
    StaffAppPanel.Controls.Add(NewOrderPage)
End Sub

Private Sub OrdersLink_Click(sender As Object, e As EventArgs) Handles
OrdersLink.Click
    LoadOrdersPage()
End Sub

Private Sub DashboardLink_Click(sender As Object, e As EventArgs)
    LoadDashboardPage()
End Sub

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Form1.LoadUserPage()

```

```
End Sub
End Class
```

Staff Orders Page – StaffOrders.vb

```
Public Class StaffOrders
    Private Sub NewOrderButton_Click(sender As Object, e As EventArgs) Handles
NewOrderButton.Click
        Form1.StaffPage.LoadNewOrderPage()
    End Sub

    Private Sub StaffOrders_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
        RenderOrderDataInListView()
    End Sub

    Private Sub RenderOrderDataInListView()
        OrderModule.GetOrderData()

        Dim CurrentItem = 0
        For Each Order As OrderModelMini In OrderModule.Orders
            OrdersList.Items.Add(Order.Id)
            OrdersList.Items(CurrentItem).SubItems.Add(Order.SName)
            CurrentItem += 1
        Next

        CurrentItem = 0
        For Each Order As OrderModelMini In OrderModule.Orders
            OrdersList.Items(CurrentItem).SubItems.Add(Order.RName)
            CurrentItem += 1
        Next

        CurrentItem = 0
        For Each Order As OrderModelMini In OrderModule.Orders
            OrdersList.Items(CurrentItem).SubItems.Add(Order.RPhone)
            CurrentItem += 1
        Next

        CurrentItem = 0
        For Each Order As OrderModelMini In OrderModule.Orders
            OrdersList.Items(CurrentItem).SubItems.Add(Order.RPin)
            CurrentItem += 1
        Next

        CurrentItem = 0
        For Each Order As OrderModelMini In OrderModule.Orders
            OrdersList.Items(CurrentItem).SubItems.Add(Order.Weight)
            CurrentItem += 1
        Next

        CurrentItem = 0
        For Each Order As OrderModelMini In OrderModule.Orders
            If Order.Delivered Then
                OrdersList.Items(CurrentItem).SubItems.Add("Completed")
            Else
                OrdersList.Items(CurrentItem).SubItems.Add(Order.Delivery.ToString("dd-MM-yy"))
            End If
            CurrentItem += 1
        Next
    End Sub
End Class
```

```

    Private Sub OrdersList_SelectedIndexChanged(sender As Object, e As EventArgs)
Handles OrdersList.SelectedIndexChanged
    If (OrdersList.SelectedItems.Count > 0) Then
        Form1.StaffPage.SelectedOrderId =
OrdersList.SelectedItems(0).SubItems(0).Text
        Form1.StaffPage.LoadEditOrderPage()
    End If
End Sub
End Class

```

Staff New Order Page – StaffNewOrder.vb

```

Public Class StaffNewOrder
    Dim CalculatedAmount As Integer
    Private Sub GoBackOrdersButton_Click(sender As Object, e As EventArgs)
Handles GoBackOrdersButton.Click
        Form1.StaffPage.LoadOrdersPage()
    End Sub

    Private Sub CreateOrderButton_Click(sender As Object, e As EventArgs) Handles
CreateOrderButton.Click

        Validation.IsValidName(SNInput.Text)
        If Validation.IsNameInvalid Then
            MessageBox.Show("Sender name : " + Validation.ErrMessage)
            Exit Sub
        End If

        Validation.IsValidPhoneNumber(SPINInput.Text)
        If Validation.IsPhoneNumberInvalid Then
            MessageBox.Show("Sender phone : " + Validation.ErrMessage)

            Exit Sub
        End If

        Validation.IsValidPinNumber(SPINInput.Text)
        If (Validation.IsPinNumberInvalid) Then
            MessageBox.Show("Sender PIN : " + Validation.ErrMessage)
            Exit Sub
        End If

        Validation.IsValidName(RNInput.Text)
        If Validation.IsNameInvalid Then
            MessageBox.Show("Receipient name : " + Validation.ErrMessage)
            Exit Sub
        End If

        Validation.IsValidPhoneNumber(RPINInput.Text)
        If Validation.IsPhoneNumberInvalid Then
            MessageBox.Show("Receipient phone : " + Validation.ErrMessage)

            Exit Sub
        End If

        Validation.IsValidPinNumber(RPINInput.Text)
        If (Validation.IsPinNumberInvalid) Then
            MessageBox.Show("Receipient PIN : " + Validation.ErrMessage)
            Exit Sub
        End If

        SubmitNewOrder()
    End Sub

```

```

Private Sub SubmitNewOrder()
    Dim IsFragile = False

    If (FragileInput.Text = "Yes") Then
        IsFragile = True
    End If

    CalculateAmount(WeightInput.Value, IsFragile)
    Dim Amount = CalculatedAmount

    Dim CreatedAt = Date.Now()
    Dim DeliveryDate = DateAdd("d", DaysInput.Value, CreatedAt)

    Dim NewOrder = New OrderModel With {
        .Content = ContentInput.Text,
        .Weight = WeightInput.Value,
        .IsFragile = IsFragile,
        .Amount = Amount,
        .SName = SNameInput.Text,
        .SPhone = SPInput.Text,
        .SAddr1 = SA1Input.Text,
        .SAddr2 = SA2Input.Text,
        .SAddr3 = SA3Input.Text,
        .SPin = SPINInput.Text,
        .Rname = RNInput.Text,
        .RPhone = RPInput.Text,
        .RAddr1 = RA1Input.Text,
        .RAddr2 = RA2Input.Text,
        .RAddr3 = RA3Input.Text,
        .RPin = RPINInput.Text,
        .CreatedAt = CreatedAt,
        .Delivery = DeliveryDate,
        .HasDelivered = False
    }

    OrderModule.CraeteOrder(NewOrder)
End Sub

Private Sub CalculateAmount(Weight As Integer, ISFragile As Boolean)
    Dim Amount As Integer

    If (ISFragile) Then
        Amount = 100 + Weight * 3
    Else
        Amount = 70 + Weight * 2
    End If

    CalculatedAmount = Amount
End Sub

Private Sub CheckPriceButton_Click(sender As Object, e As EventArgs) Handles
CheckPriceButton.Click
    Dim IsFragile = False

    If (FragileInput.Text = "Yes") Then
        IsFragile = True
    End If

    CalculateAmount(WeightInput.Value, IsFragile)
    PriceLabel.Text = CalculatedAmount
End Sub
End Class

```

Staff Edit Order Page – StaffEditOrder.vb

```
Public Class StaffEditOrder
    Private Sub StaffEditOrder_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
        IdLabel.Text = Form1.StaffPage.SelectedOrderId

        OrderModule.GetOrderById(Form1.StaffPage.SelectedOrderId)

        OrderUpdatesModule.GetUpdatesById(Form1.StaffPage.SelectedOrderId)

        OrderUpdatesModule.GetUpdatesById(Form1.StaffPage.SelectedOrderId)

        IdLabel.Text = Form1.SelectedId

        UpdatesList.Items.Add(OrderModule.Order.CreatedAt.ToString("dd-MM-yy"))
        UpdatesList.Items(0).SubItems.Add("Dispatched")
        Dim CurrentItem = 1
        For Each Update As OrderUpdatesModel In OrderUpdatesModule.Updates
            UpdatesList.Items.Add(Update.Date.ToString("dd-MM-yy"))
            UpdatesList.Items(CurrentItem).SubItems.Add(Update.Memo)
            CurrentItem += 1
        Next

        If (OrderModule.Order.Delivered = True) Then
            UpdatesList.Items.Add(OrderModule.Order.Delivery.ToString("dd-MM-yy"))
            UpdatesList.Items(CurrentItem).SubItems.Add("Delivered")
        Else
            UpdatesList.Items.Add(OrderModule.Order.Delivery.ToString("dd-MM-yy"))
            UpdatesList.Items(CurrentItem).SubItems.Add("Expected delivery")
        End If
    End Sub

    Private Sub AddPointButton_Click(sender As Object, e As EventArgs) Handles
AddPointButton.Click
        Form1.StaffPage.LoadAddPointPage()
    End Sub

    Private Sub AddDelayButton_Click(sender As Object, e As EventArgs) Handles
AddDelayButton.Click
        Form1.StaffPage.LoadAddDelayPage()
    End Sub

    Private Sub GoBackButton_Click(sender As Object, e As EventArgs) Handles
GoBackButton.Click
        Form1.StaffPage.LoadOrdersPage()
    End Sub

    Private Sub MarkDeliverButton_Click(sender As Object, e As EventArgs) Handles
MarkDeliverButton.Click
        OrderModule.MarkAsDelivered(Form1.StaffPage.SelectedOrderId, Date.Now())
    End Sub
End Class
```

Staff Add Order Point – StaffAddPoint.vb

```
Public Class StaffAddPoint
    Private Sub GoBackButton_Click(sender As Object, e As EventArgs) Handles
GoBackButton.Click
```

```

        Form1.StaffPage.LoadEditOrderPage()
    End Sub

    Private Sub AddPointButton_Click(sender As Object, e As EventArgs) Handles
AddPointButton.Click
        Dim Point = PointInput.Text

        OrderModule.GetOrderById(Form1.StaffPage.SelectedOrderId)
        OrderUpdatesModule.AddPointById(Form1.StaffPage.SelectedOrderId,
Date.Now(), Point)
    End Sub
End Class

```

Staff Add Order Delay – StaffAddDelay.vb

```

Public Class StaffAddDelay
    Private Sub GoBackButton_Click(sender As Object, e As EventArgs) Handles
GoBackButton.Click
        Form1.StaffPage.LoadEditOrderPage()
    End Sub

    Private Sub AddPointButton_Click(sender As Object, e As EventArgs) Handles
AddPointButton.Click
        Dim Memo = DaysInput.Value.ToString() + " days delay at " +
PointInput.Text

        OrderModule.GetOrderById(Form1.StaffPage.SelectedOrderId)

        Dim CurrentDate = Date.Now()
        Dim NewDate = DateAdd("d", DaysInput.Value, OrderModule.Order.Delivery)
        OrderUpdatesModule.AddDelayById(Form1.StaffPage.SelectedOrderId,
CurrentDate, Memo, NewDate)
    End Sub
End Class

```

User Page – User.vb

```

Public Class User
    Private Sub GoToStaffLoginButton_Click(sender As Object, e As EventArgs)
Handles GoToStaffLoginButton.Click
        Form1.LoadStaffLoginPage()
    End Sub

    Private Sub CheckStatusButton_Click(sender As Object, e As EventArgs) Handles
CheckStatusButton.Click
        Dim Id = IdInput.Text

        If (Id < 1) Then
            MessageBox.Show("Please enter a valid ortder id")
        Else
            OrderModule.GetOrderById(Id)
            If (OrderModule.OrderExists) Then
                Form1.SelectedId = Id
                Form1.LoadUserDetailsPage()
            Else
                MessageBox.Show("Please enter a valid ortder id")
            End If
        End If
    End Sub
End Class

```

```

        Private Sub User_Load(sender As Object, e As EventArgs) Handles MyBase.Load
            End Sub
        End Class

```

User Order Details Page – UserOrderDetails.vb

```
Public Class UserOrderDetails
```

```

    Private Sub UserOrderDetails_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
        OrderModule.GetOrderById(Form1.SelectedId)

        OrderUpdatesModule.GetUpdatesById(Form1.SelectedId)

        OrderUpdatesModule.GetUpdatesById(Form1.SelectedId)

        IDLabel.Text = Form1.SelectedId

        UpdatesList.Items.Add(OrderModule.Order.CreatedAt.ToString("dd-MM-
yy"))
        UpdatesList.Items(0).SubItems.Add("Dispatched")
        Dim CurrentItem = 1
        For Each Update As OrderUpdatesModel In OrderUpdatesModule.Updates
            UpdatesList.Items.Add(Update.Date.ToString("dd-MM-yy"))
            UpdatesList.Items(CurrentItem).SubItems.Add(Update.Memo)
            CurrentItem += 1
        Next

        If (OrderModule.Order.Delivered = True) Then
            UpdatesList.Items.Add(OrderModule.Order.Delivery.ToString("dd-MM-
yy"))
            UpdatesList.Items(CurrentItem).SubItems.Add("Delivered")
        Else
            UpdatesList.Items.Add(OrderModule.Order.Delivery.ToString("dd-MM-
yy"))
            UpdatesList.Items(CurrentItem).SubItems.Add("Expected delivery")
        End If
    End Sub

```

```

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
        Form1.LoadUserPage()
    End Sub
End Class

```

DbHelper.vb

```
Imports MySql.Data.MySqlClient
```

```
Module DbHelper
```

```

    Sub CreateTables()
        Dim UsersTableSchema = "create table if not exists users (id int not null
auto_increment primary key, name varchar(25) not null, username varchar(25) not
null, password varchar(20) not null, is_admin bool not null )"
        Dim OrdersTableSchema = "create table if not exists orders (id int not
null auto_increment primary key, content varchar(50) not null, weight int not
null, amount int not null, is_fragile boolean, sender_name varchar(25) not null,
sender_phone varchar(12) not null, sender_addr1 varchar(50) not null,
sender_addr2 varchar(50) not null, sender_addr3 varchar(50) not null, sender_pin
varchar(20) not null

```

```
, recipient_name varchar(25) not null, recipient_phone varchar(12) not null,
recipient_addr1 varchar(50) not null, recipient_addr2 varchar(50) not null,
recipient_addr3 varchar(50) not null, recipient_pin varchar(20) not null,
created_at datetime not null, delivery datetime not null, has_delivered boolean
default false)"
```

```
Dim OrderUpdatesTableSchema = "create table if not exists
order_updates(id int not null auto_increment primary key, date datetime not null,
memo varchar(100) not null, rel int)"
```

```
Try
    Form1.DbConnection.Open()
    Dim cmd = New MySqlCommand(UsersTableSchema, Form1.DbConnection)
    Dim Cmd2 = New MySqlCommand(OrdersTableSchema, Form1.DbConnection)
    Dim Cmd3 = New MySqlCommand(OrderUpdatesTableSchema,
Form1.DbConnection)
```

```
    cmd.ExecuteNonQuery()
    Cmd2.ExecuteNonQuery()
    Cmd3.ExecuteNonQuery()
```

```
Catch ex As Exception
    MessageBox.Show(ex.Message)
```

```
Finally
    Form1.DbConnection.Close()
    Form1.DbConnection.Dispose()
```

```
End Try
```

```
End Sub
```

```
End Module
```

AdminModule.vb

```
Imports MySql.Data.MySqlClient
```

```
Module AdminModule
```

```
Sub AddAdmin(Name As String, Username As String, Password As String)
```

```
    Dim Sql = "insert into users (name, username, password, is_admin)
values(@Name, @Username, @Password, true)"
```

```
Try
    Form1.DbConnection.Open()
    Dim Cmd = New MySqlCommand(Sql, Form1.DbConnection)
```

```
    Cmd.Parameters.AddWithValue("@Name", Name)
    Cmd.Parameters.AddWithValue("@Username", Username)
    Cmd.Parameters.AddWithValue("@Password", Password)
```

```
    Dim Res = Cmd.ExecuteScalar()
```

```
    ' Navigate to admin page
    Form1.LoadAdminPage()
```

```
Catch ex As Exception
    MessageBox.Show(ex.Message)
```

```
Finally
    Form1.DbConnection.Close()
```

```
End Try
```

```
End Sub
```

```
Sub LoginAdmin(Username As String, password As String)
```

```
    Dim Sql = "select name, username, password, is_admin from users where
username=@Username"
```

```
Try
    Form1.DbConnection.Open()
    Dim Cmd = New MySqlCommand(Sql, Form1.DbConnection)
```



```

        Cmd.Parameters.AddWithValue("@Username", Username)

        Dim Reader = Cmd.ExecuteReader

        If Reader.Read() = True Then
            If Reader("is_admin") = True Then
                If Reader("password") = password Then
                    Dim Name = Reader("name")
                    Form1.LoggedInusername = Name

                    Form1.DbConnection.Close()
                    Form1.LoadAdminPage()
                Else
                    MessageBox.Show("Invalid password")
                End If
            Else
                MessageBox.Show("Invalid username")
            End If
        Else
            MessageBox.Show("Invalid username")
        End If
    Catch ex As Exception
        MessageBox.Show(ex.Message)
    Finally
        Form1.DbConnection.Close()
    End Try
End Sub
End Module

```

UserModule.vb

```

Imports MySql.Data.MySqlClient

Module UserModule
    Public UserExists = False
    Public Users As New List(Of UserModel)

    Sub IsUsersExists()
        Dim Cmd = New MySqlCommand("select * from users", Form1.DbConnection)

        Try
            Form1.DbConnection.Open()
            Dim Reader = Cmd.ExecuteReader()

            If Reader.HasRows() = True Then
                UserExists = True
            End If
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        Finally
            Form1.DbConnection.Close()
        End Try
    End Sub

    Sub LoadUserData()
        Users.Clear()
        Dim Cmd = New MySqlCommand("select * from users", Form1.DbConnection)

        Try
            Form1.DbConnection.Open()
            Dim Reader = Cmd.ExecuteReader()

```

```

        If Reader.HasRows Then
            While Reader.Read()
                Users.Add(New UserModel With {
                    .Id = Reader("id"),
                    .Name = Reader("name"),
                    .Username = Reader("username"),
                    .IsAdmin = Reader("is_admin")
                })
            End While
        End If
    Catch ex As Exception
        MessageBox.Show(ex.Message)

    Finally
        Form1.DbConnection.Close()
    End Try
End Sub

Sub AddUser(Name As String, Username As String, Password As String, IsAdmin
As Boolean)
    Dim Sql1 = "select name, username, password, is_admin from users where
username=@Username"
    Dim Sql2 = "insert into users (name, username, password, is_admin) values
(@Name, @Username, @Password, @IsAdmin)"

    Try
        Form1.DbConnection.Open()

        ' Check if the user already exists
        Dim CheckCmd = New MySqlCommand(Sql1, Form1.DbConnection)
        CheckCmd.Parameters.AddWithValue("@Username", Username)
        Dim Reader = CheckCmd.ExecuteReader

        If Reader.Read() = True Then
            MessageBox.Show("User laready exists")
            Return
        End If
        Form1.DbConnection.Close()

        ' Adding new user
        Form1.DbConnection.Open()
        Dim Cmd = New MySqlCommand(Sql2, Form1.DbConnection)
        Cmd.Parameters.AddWithValue("@Name", Name)
        Cmd.Parameters.AddWithValue("@Username", Username)
        Cmd.Parameters.AddWithValue("@Password", Password)
        Cmd.Parameters.AddWithValue("@IsAdmin", IsAdmin)
        Cmd.ExecuteScalar()

        Form1.DbConnection.Close()

        ' Navigate to user management page
        Form1.AdminPage.LoadUserManagementPage()
    Catch ex As Exception
        MessageBox.Show(ex.Message)

    Finally
        Form1.DbConnection.Dispose()
    End Try
End Sub

```

```
End Module
```

```
Public Class UserModel
    Public Id As Integer
    Public Name As String
    Public Username As String
    Public IsAdmin As Boolean
End Class
```

StaffModule.vb

```
Imports MySql.Data.MySqlClient
```

```
Module StaffModule
```

```
    Public Sub StaffLogin(Username As String, Password As String)
        Dim Sql = "select name, username, password, is_admin from users where  
username=@Username and is_admin=false"
        Try
            Form1.DbConnection.Open()
            Dim Cmd = New MySqlCommand(Sql, Form1.DbConnection)
            Cmd.Parameters.AddWithValue("@Username", Username)

            Dim Reader = Cmd.ExecuteReader

            If Reader.Read() = True Then
                If Reader("password") = Password Then
                    Form1.LoggedInusername = Reader("name")
                    Form1.DbConnection.Close()
                    Form1.LoadStaffPage()
                Else
                    MessageBox.Show("Invalid password")
                End If
            Else
                MessageBox.Show("Invalid username")
            End If
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        Finally
            Form1.DbConnection.Close()
        End Try
    End Sub
End Module
```

UI (Screenshots)

Add Admin Page

Swift Courier

Welcome to Swift Courier

Please add the admin details

Name


Username

Password

Create Admin

Admin Login Page

Swift Courier

An illustration of a courier in a blue uniform and cap running while carrying a brown box. A large clock face is positioned behind the courier, and the scene is set against a light blue background with green bushes.

Please login.

Username

Password

Login

Go To Staff Login

Admin Orders Page

Swift Courier

Admin Panel

Orders

Users

Id	Sender Name	Receipient Name	R. Phone	R. Pin	Weight	Delivery
1	Rahul	Akhil	9867678746	600004	100	14-03-22
2	James	Anna	9423456453	678654	100	13-03-22
3	Ravi	Rahul	9098765456	766788	760	20-03-22

Logout

Admin Users Page

Swift Courier

Admin Panel

Orders

Users

Add User

Id	Name	Username	User Type
1	James	jamesbond	Admin
2	Adam	a	Admin
3	Ben	b	Staff

Logout

Admin Add User Page

Swift Courier

Admin Panel

Orders

Users

Add User

Go Back

Name

Username

Password

User Type

Add User

Logout

Staff Login Page

Swift Courier

Staff Login


Username

Pssword

Login

Go To Admin Login

Go To User Page



Staff Orders Page

Swift Courier

Swift Courier

Orders

New Order

Id	Sender Name	Receipient Name	R. Phone	R. Pin	Weight	Delivery
1	Rahul	Akhil	9867678746	600004	100	14-03-22
2	James	Anna	9423456453	678654	100	13-03-22
3	Ravi	Rahul	9098765456	766788	760	20-03-22

Logout

Staff New Order Page

Swift Courier

Swift Courier

Orders

New Order

Go Back

Content

Weight (g)

0

Fragile

Sender Name

Recipient Name

Sender Phone

Recipient Phone

Sender Addr 1

Recipient Addr 1

Sender Addr 2

Recipient Addr2

Sender Addr 3

Recipient Addr 3

Sender PIN

Recipient PIN

Estimated Time (In days)

1

Price (Rs.):

0

Check Price

Create Order

Staff Edit Order Page

Swift Courier

Swift Courier

Orders

Staff

Logout

Update Order

Id : 4

Date	Memo
07-03-22	Dispatched
11-03-22	Expected delivery

Add Point

Add Delay

Mark as Delivered

GoBack

Staff Add Point Page

Swift Courier

Swift Courier

Orders

Staff

Logout

Add Point

Memo

Add Point

GoBack

Staff Add Delay Page

Swift Courier

Swift Courier

Orders

Logout

Add Delay

GoBack

Place

Delay (Days)

0

Add Delay

User page

Swift Courier

Swift Courier

Enter your order id below

Check Status

Go To Staff Login

User Order Details Page

Swift Courier

Order Details

Go Back

Id: 2

Date	Memo
10-03-22	Dispatched
10-03-22	Reached Mysoor
10-03-22	Reached Vellur
10-03-22	3 days delay at Hossur
16-03-22	Expected delivery

REPORTS

Users Report

Add User				
Id	Name	Username	User Type	
1	James	jamesluca	Admin	
2	Ram Krishna	ramkrishna	Staff	
3	Rudy Falcao	rudyfalcao	Admin	
4	Rabecca	rabecca	Admin	

Orders Report

Id	Sender Name	Receipient Name	R. Phone	R. Pin	Weight	Delivery
1	Rahul	Akhil	9867678746	600004	100	16-03-22
2	James	Anna	9423456453	678654	100	16-03-22
3	Ravi	Rahul	9098765456	766788	760	16-03-22

