

---

# Project Documentation

Version 1.0

May 19, 2014

## Smart-hypermarket

Smart-phone based shopping software

Ahmed Gamal Sdeek 20110012

Eslam Mostafa Abd-Elhameed 20110101

Hesham Hossam Helmy 20110486

Waleed Hesham Mohamed 20110496

Yasser Omar Mohamed 20110499

Submitted in partial fulfillment Of the requirements of CS  
313 Programming -3

---

# Contents

<b>1</b>	<b>Description</b>	<b>4</b>
<b>2</b>	<b>Tasks Distribution</b>	<b>5</b>
<b>3</b>	<b>UML Diagrams</b>	<b>7</b>
<b>4</b>	<b>Open-source Tools Used</b>	<b>10</b>
<b>5</b>	<b>XML Documentation</b>	<b>11</b>

## List of Figures

3.1	Models class diagram . . . . .	7
3.2	Data Entry Manager module class diagram . . . . .	8
3.3	Storage Manager module class diagram . . . . .	9

# 1 Description

Smart Hypermarket converts every single piece in the market into a smart piece that can communicate with shoppers through their smart-phones.

## 2 Tasks Distribution

- Hesham Hossam Helmy
  - Detect offers (Database, Webservice, Bluetooth, Offer classes)
  - Detecting the hypermarket you enter and show announcements notifications related to it (Database, Webservice)
  - Browse Products (Database, Webservice)
  - Adding to cart (Order::addProduct)
  - Browse cart (IShowableInterface)
  - Order submission (Database, Webservice)
  - Order ready notification (Webservice)
  - Enlist a product (webservice, IProduct, Model)
  - View products
  - Edit products (Webservice)
  - Delete Products (Webservice)
  - Create offers (Webservice, Offer class, Input, Response classes)
  - Check orders (OrderListing Page xaml code)
  - Review products (Webservice)
  - Check reviews (Webservice)
- Eslam Mostafa Abd-Elhameed
  - Detect offers (Parse the Json file passed from the service)
  - Detecting the hypermarket you enter and show announcements ”notifications” related to it (Parse the Json file passed from the service)
  - browse Products (Make the view product page in data entry manager)
  - Adding to cart
  - View Product information (Parse the Json file passed from the service).
  - Order ready notification
  - Enlist a product (Parse the Json file passed from the service) and create the design of the xaml code
  - View Products (Pares the json file passed from the service).
  - Delete Product

- Create offers (Create The xaml code for design)
- Browse offers (show the offer in data grid)
- Receive payment and deliver order (Parse the Json file passed from the service)
- Read Product Reviews (Parse the Json file Passed from the service)
- Yasser Omar Mohamed
  - Create Desktop bar-code reader library (Desktop)
  - Create Log in interface (Desktop)
  - Implement Enlist a product functionality (Desktop)
  - Implement Edit product functionality (Desktop)
  - Implement Add product to offer functionality (Desktop)
  - Implement Create offers functionality (Desktop)
  - Implement Submit order as served functionality (Desktop)
  - Implement show order functionality (Desktop)
- Ahmed Gamal Sdeek
  - Detect offers (Android, Blue-tooth)
  - Browse Products (Android)
  - Detect Bar-code of each product (Android, Bar-code Reader)
  - View Product information (Android)
  - Adding to cart (Android)
  - Order submission (Android)
  - View Orders to be served (Android)
  - Read Product Reviews (Android)
  - Write Product Review (Android)
  - Submit order as served to notify the shopper to go and receive it (Android)
  - Receive payment and deliver order (Android)
  - Browse offers (Android)
- Waleed Hesham Mohamed
  - Enlist a product (implementation)
  - Delete Product (implementation)
  - Browse offers (User Interface and function implementation)
  - Edit offers (User Interface and function implementation)
  - Delete Offers (Implementation)
  - View Orders to be served (User Interface and implementation)
  - Check orders (User Interface and implementation)

### 3 UML Diagrams

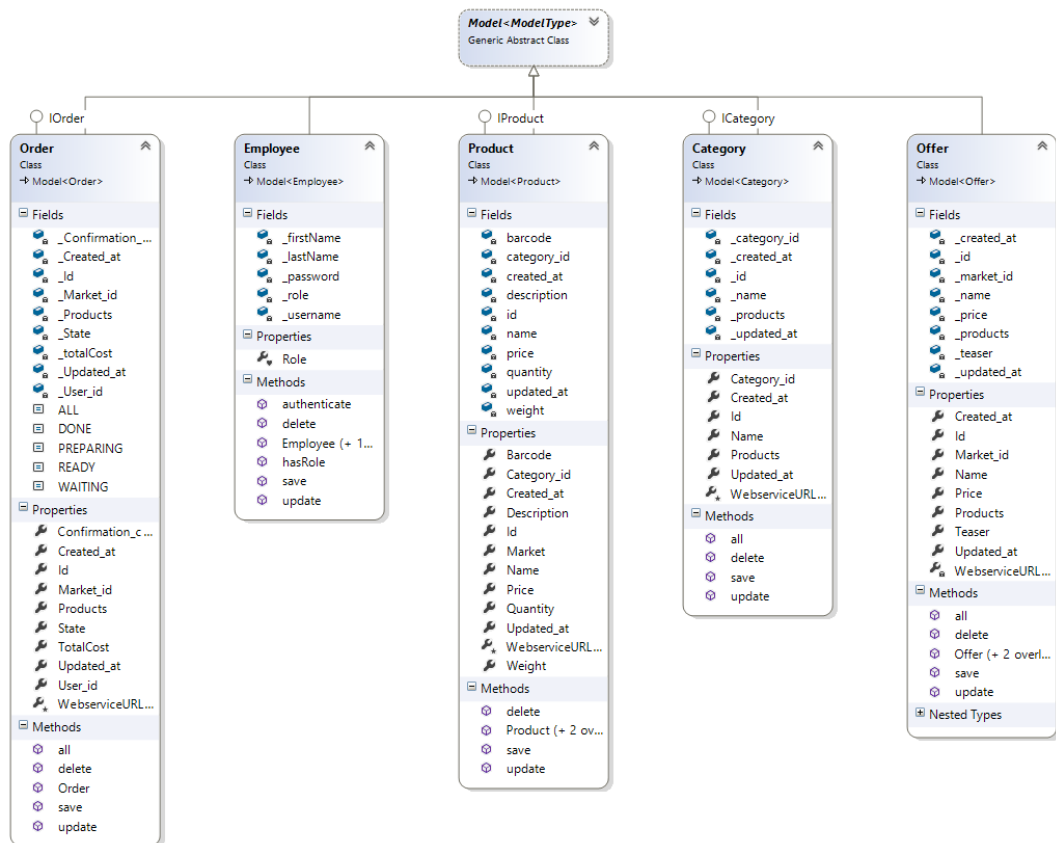


Figure 3.1: Models class diagram

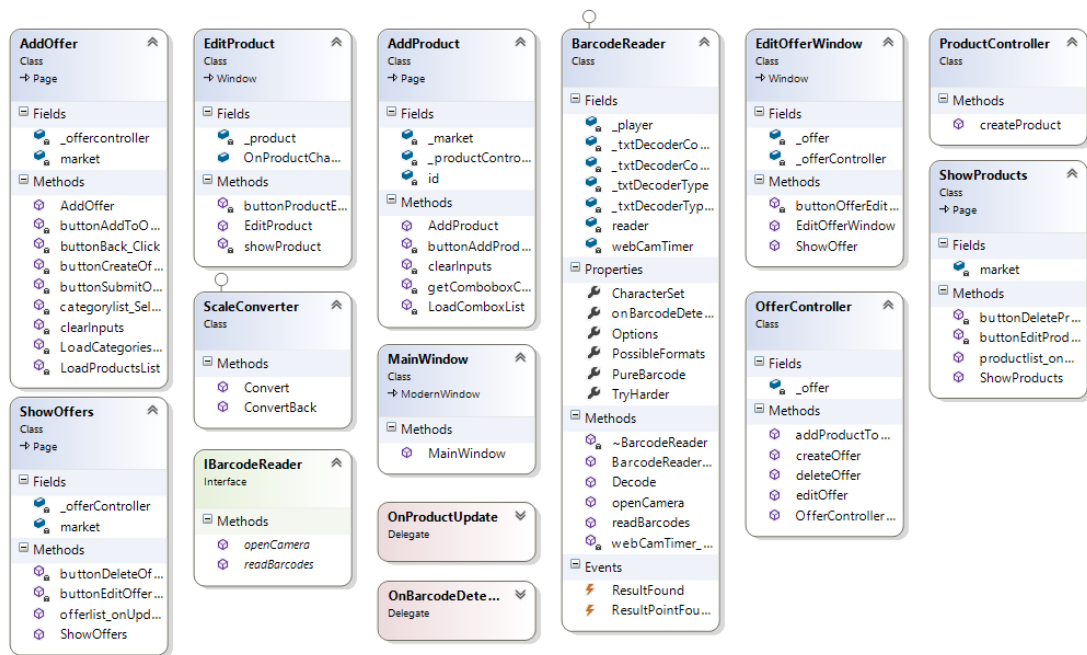


Figure 3.2: Data Entry Manager module class diagram



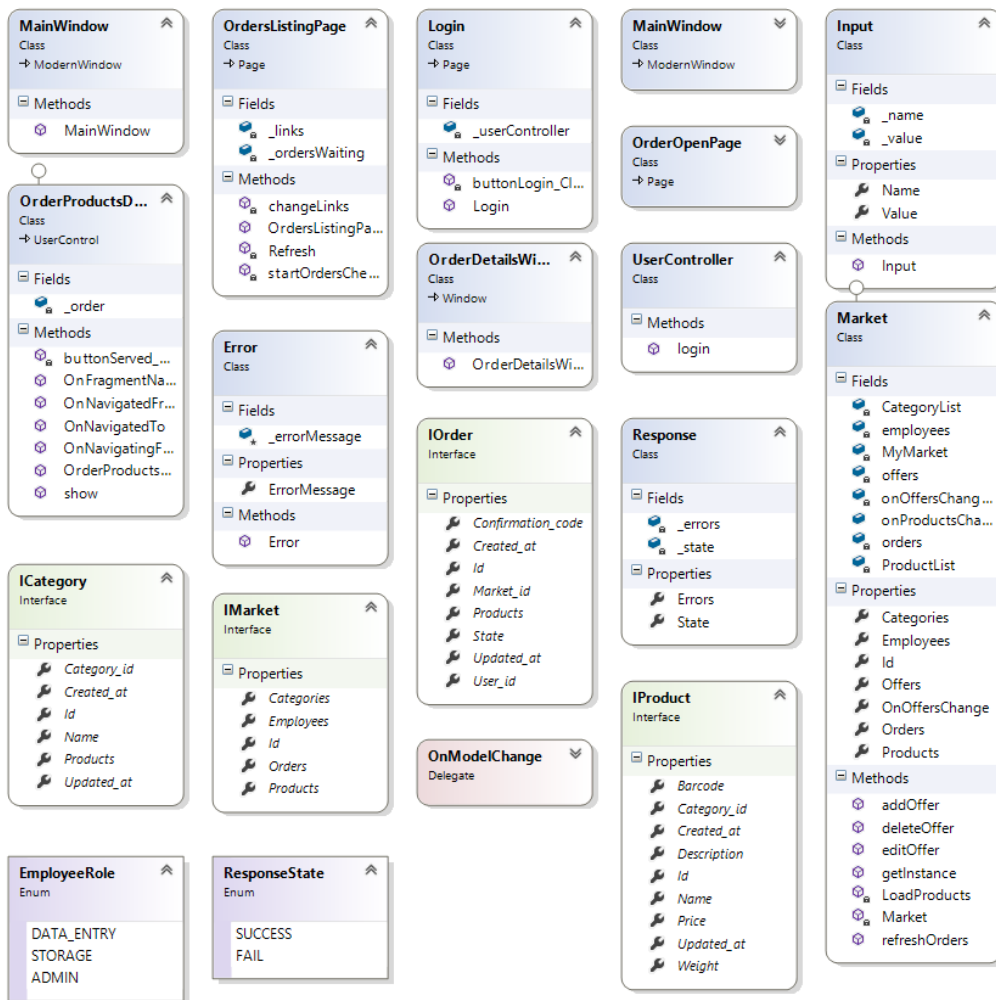


Figure 3.3: Storage Manager module class diagram

## 4 Open-source Tools Used

- ZXing.Net project, [zxingnet.codeplex.com](http://zxingnet.codeplex.com)
- ZXing project, [github.com/zxing/zxing](https://github.com/zxing/zxing)
- Laravel PHP Framework, [github.com/laravel/laravel](https://github.com/laravel/laravel)

## 5 XML Documentation

```
<?xml version="1.0"?>
<doc>
  <assembly>
    <name>SmartHyperMarket</name>
  </assembly>
  <members>
    <member name="T:SmartHyperMarket.App">
      <summary>
        Interaction logic for App.xaml
      </summary>
      <summary>
        App
      </summary>
    </member>
    <member name="M:SmartHyperMarket.App.InitializeComponent">
      <summary>
        InitializeComponent
      </summary>
    </member>
    <member name="M:SmartHyperMarket.App.Main">
      <summary>
        Application Entry Point.
      </summary>
    </member>
    <member name="T:SmartHyperMarket.Common.Controllers.Error">
      <summary>
        Handel errors of the application occured during runtime
      </summary>
    </member>
    <member name="T:SmartHyperMarket.Common.Controllers.Input">
      <summary>
        Work as a container for diffirent inputss
      </summary>
    </member>
    <member name="T:SmartHyperMarket.Common.Controllers.Response">
      <summary>
        A type which handels the end result of the function and keep track of errors o
```

```

        </summary>
    </member>
    <member name="M:SmartHyperMarket.Common.Models.Market.GetInstance">
        <summary>
            get instance of the static object market
        </summary>
        <returns>market object</returns>

    </member>
    <member name="T:SmartHyperMarket.Controllers.UserController">
        <summary>
            Control user behavior and functionality
        </summary>
    </member>
    <member name="M:SmartHyperMarket.Controllers.UserController.login(SmartHyperMarket.Common.Models.Market, SmartHyperMarket.Common.Models.User)">
        <summary>
            Login user into server using his information
        </summary>
        <param name="inputs">List of inputs : username, password</param>
        <returns>Result of runing the function</returns>
    </member>
    <member name="M:SmartHyperMarket.DataEntryManager.BarcodeReading.IBarcodeReader.OpenWebcam">
        <summary>
            Opens the webcam of computer
        </summary>
        <returns>Boolean true if camera is open successfully, false if some error happens</returns>
    </member>
    <member name="M:SmartHyperMarket.DataEntryManager.BarcodeReading.IBarcodeReader.ReadBarcode">
        <summary>
            Take a frame from camera and try to convert it into a barcode if succeeded the
        </summary>
    </member>
    <member name="M:SmartHyperMarket.DataEntryManager.BarcodeReading.BarcodeReader.#ctor">
        <summary>
            Constructor of BarcodeReader class
        </summary>
        <param name="txtDecoderType">Reference to a label which will hold the barcode type</param>
        <param name="txtDecoderContent">Reference to a label which will hold the barcode content</param>
        <param name="pictureBox">Reference to a PictureBox which will hold the Web Camera</param>
    </member>
    <member name="M:SmartHyperMarket.DataEntryManager.BarcodeReading.BarcodeReader.WebCamTick">
        <summary>
            Fired each tick to capture a frame and decode it
        </summary>
    </member>

```

```

        <param name="sender"></param>
        <param name="e"></param>
    </member>
    <member name="T:SmartHyperMarket.DataEntryManager.Controllers.OfferController">
        <summary>
            Handel the logic operations dealing with offers and it's related functionality
        </summary>
    </member>
    <member name="M:SmartHyperMarket.DataEntryManager.Controllers.OfferController.add">
        <summary>
            Add new product with it's quantity to controller of the offer
        </summary>
        <param name="inputs">List of inputs : productID and productQuantity</param>
        <returns>Result of runing the function</returns>
    </member>
    <member name="M:SmartHyperMarket.DataEntryManager.Controllers.OfferController.cre">
        <summary>
            Create the controller offer constructed in the system
        </summary>
        <param name="inputs">List of inputs : name, price and teaser</param>
        <returns>Result of runing the function</returns>
    </member>
    <member name="M:SmartHyperMarket.DataEntryManager.Controllers.OfferController.edi">
        <summary>
            Edit the offer of offer controller instance
        </summary>
        <param name="inputs">List of edited inputs : name, price and teaser</param>
        <returns>Result of runing the function</returns>
    </member>
    <member name="M:SmartHyperMarket.DataEntryManager.Controllers.OfferController.del">
        <summary>
            Delete the offer currently held by the offer controller
        </summary>
        <returns>Result of runing the function</returns>
    </member>
    <member name="T:SmartHyperMarket.DataEntryManager.Controllers.ProductController">
        <summary>
            Handel of the logic operations related with product
        </summary>
    </member>
    <member name="M:SmartHyperMarket.DataEntryManager.Controllers.ProductController.c">
        <summary>
            Create new product in the system
        </summary>

```

```

        <param name="inputs">List of inputs : name, barcode, price, categoryId, weight
        <returns>Result of runing the function</returns>
    </member>
    <member name="T:SmartHyperMarket.DataEntryManager.Views.AddOffer">
        <summary>
            Interaction logic for AddOffer.xaml
        </summary>
        <summary>
            AddOffer
        </summary>
    </member>
    <member name="M:SmartHyperMarket.DataEntryManager.Views.AddOffer.InitializeCompon
        <summary>
            InitializeComponent
        </summary>
    </member>
    <member name="T:SmartHyperMarket.DataEntryManager.Views.AddProduct">
        <summary>
            Interaction logic for AddProduct.xaml
        </summary>
        <summary>
            AddProduct
        </summary>
    </member>
    <member name="M:SmartHyperMarket.DataEntryManager.Views.AddProduct.InitializeComp
        <summary>
            InitializeComponent
        </summary>
    </member>
    <member name="T:SmartHyperMarket.DataEntryManager.Views.EditOfferWindow">
        <summary>
            Interaction logic for EditOfferWindow.xaml
        </summary>
        <summary>
            EditOfferWindow
        </summary>
    </member>
    <member name="M:SmartHyperMarket.DataEntryManager.Views.EditOfferWindow.Initializ
        <summary>
            InitializeComponent
        </summary>
    </member>
    <member name="T:SmartHyperMarket.DataEntryManager.Views.EditProduct">
        <summary>

```

```

        Interaction logic for EditProduct.xaml
    </summary>
    <summary>
        EditProduct
    </summary>
</member>
<member name="M:SmartHyperMarket.DataEntryManager.Views.EditProduct.#ctor(SmartHyperMarket.DataEntryManager.Views.EditProduct)">
    <summary>
        Constructor of edit product page
    </summary>
    <param name="product">the product to edit</param>
</member>
<member name="M:SmartHyperMarket.DataEntryManager.Views.EditProduct.showProduct(SmartHyperMarket.DataEntryManager.Views.EditProduct)">
    <summary>
        Show Product member fields in the textboxes and other input fields
    </summary>
    <param name="product">product to show</param>
</member>
<member name="M:SmartHyperMarket.DataEntryManager.Views.EditProduct.InitializeComponent()">
    <summary>
        InitializeComponent
    </summary>
</member>
<member name="T:SmartHyperMarket.DataEntryManager.Views.MainWindow">
    <summary>
        Interaction logic for MainWindow.xaml
    </summary>
    <summary>
        MainWindow
    </summary>
</member>
<member name="M:SmartHyperMarket.DataEntryManager.Views.MainWindow.InitializeComponent()">
    <summary>
        InitializeComponent
    </summary>
</member>
<member name="T:SmartHyperMarket.DataEntryManager.Views.ShowOffers">
    <summary>
        Interaction logic for ShowOffers.xaml
    </summary>
    <summary>
        ShowOffers
    </summary>
</member>

```

```

<member name="M:SmartHyperMarket.DataEntryManager.Views.ShowOffers.InitializeComp
    <summary>
        InitializeComponent
    </summary>
</member>
<member name="T:SmartHyperMarket.DataEntryManager.Views.ShowProducts">
    <summary>
        Interaction logic for Page1.xaml
    </summary>
    <summary>
        ShowProducts
    </summary>
</member>
<member name="M:SmartHyperMarket.DataEntryManager.Views.ShowProducts.InitializeCo
    <summary>
        InitializeComponent
    </summary>
</member>
<member name="T:SmartHyperMarket.StorageManager.Views.MainWindow">
    <summary>
        Interaction logic for MainWindow.xaml
    </summary>
    <summary>
        MainWindow
    </summary>
</member>
<member name="M:SmartHyperMarket.StorageManager.Views.MainWindow.InitializeCompon
    <summary>
        InitializeComponent
    </summary>
</member>
<member name="T:SmartHyperMarket.StorageManager.Views.OrderDetailsWindow">
    <summary>
        Interaction logic for OrderDetailsWindow.xaml
    </summary>
    <summary>
        OrderDetailsWindow
    </summary>
</member>
<member name="M:SmartHyperMarket.StorageManager.Views.OrderDetailsWindow.Initiali
    <summary>
        InitializeComponent
    </summary>
</member>

```



```

<member name="T:SmartHyperMarket.StorageManager.Views.OrderProductsDetailsControl
    <summary>
        Interaction logic for OrderProductsDetailsControl.xaml
    </summary>
    <summary>
        OrderProductsDetailsControl
    </summary>
</member>
<member name="M:SmartHyperMarket.StorageManager.Views.OrderProductsDetailsControl
    <summary>
        InitializeComponent
    </summary>
</member>
<member name="T:SmartHyperMarket.StorageManager.Views.OrderOpenPage">
    <summary>
        Interaction logic for OrderOpenPage.xaml
    </summary>
    <summary>
        OrderOpenPage
    </summary>
</member>
<member name="M:SmartHyperMarket.StorageManager.Views.OrderOpenPage.InitializeCom
    <summary>
        InitializeComponent
    </summary>
</member>
<member name="T:SmartHyperMarket.StorageManager.Views.OrdersListingPage">
    <summary>
        Interaction logic for OrdersListingPage.xaml
    </summary>
    <summary>
        OrdersListingPage
    </summary>
</member>
<member name="M:SmartHyperMarket.StorageManager.Views.OrdersListingPage.Initializ
    <summary>
        InitializeComponent
    </summary>
</member>
<member name="T:SmartHyperMarket.Views.Login">
    <summary>
        Interaction logic for Login.xaml
    </summary>
    <summary>

```

```

        Login
    </summary>
</member>
<member name="M:SmartHyperMarket.Views.Login.InitializeComponent">
    <summary>
        InitializeComponent
    </summary>
</member>
<member name="T:SmartHyperMarket.Views.MainWindow">
    <summary>
        Interaction logic for MainWindow.xaml
    </summary>
    <summary>
        MainWindow
    </summary>
</member>
<member name="M:SmartHyperMarket.Views.MainWindow.InitializeComponent">
    <summary>
        InitializeComponent
    </summary>
</member>
<member name="T:SmartHyperMarket.Properties.Resources">
    <summary>
        A strongly-typed resource class, for looking up localized strings, etc.
    </summary>
</member>
<member name="P:SmartHyperMarket.Properties.Resources.ResourceManager">
    <summary>
        Returns the cached ResourceManager instance used by this class.
    </summary>
</member>
<member name="P:SmartHyperMarket.Properties.Resources.Culture">
    <summary>
        Overrides the current thread's CurrentUICulture property for all
        resource lookups using this strongly typed resource class.
    </summary>
</member>
</members>
</doc>

```