**Cell Men**

**Nathan Silvester and Richard Olivera**

**Application Summary:**

For this project our team decided to make a website for Cell Phones which we are very passionate about. Although the company does not exist we set out to build something that would work, and look great if it were ever to be posted for the general public.

The idea behind the website was to create a “One Stop Cell Shop” where people could buy cell phones and tablets, sell cell phones and tablets, and purchase repair parts for their cell phones and tablets. Our main target audience are those people that want to upgrade their phones but are stuck in contracts or people who simply want to repair their broken phone. We envision people going to our website and receiving a reasonable offer for their current device and using that money to purchase another device from our inventory.

As for the functionality, it is safe to say that we wanted to go with a simplistic design that is easy to use. That is precisely what we got, our navigation is fairly intuitive and people can easily find what they are looking for. Apart from those users looking to sell, buy, or purchase repair parts, our intended users include admins. An admin can have the option to review the devices that are in inventory simply by using their login credentials.

**Application Guide:**

1. Master pages
2. .aspx pages
3. User controls
4. Databases
   1. CellDatabase.mdf
   2. ASPNETDB.MDF
5. Other supporting files (Themes & Skins, XML files, sitemap files, etc.)

**Role Descriptions:**

**Project Requirements:**

|  |  |
| --- | --- |
| **MIS 5050 Project Requirements** | **Project Component(s)** |
| 5 interlinked Web forms (.aspx pages) that display dynamically-generated content **per team member** | 1. Default.aspx 2. Type.aspx 3. Manufacturer.aspx 4. Model.aspx 5. Inventory.aspx 6. Phone.aspx 7. Type - Repair.aspx 8. Manufacturer - Repair.aspx 9. Model - Repair.aspx 10. Phone - Repair.aspx 11. Type - Sell.aspx 12. Manufacturer - Sell.aspx 13. Model - Sell.aspx 14. Phone - Sell.aspx |
| At **least** 5 database transactions (a transaction includes selecting, inserting, updating, or deleting data from the database) **per team member** | 1. Default.aspx 2. Type.aspx 3. Manufacturer.aspx 4. Model.aspx 5. Inventory.aspx 6. Phone.aspx 7. Type - Repair.aspx 8. Manufacturer - Repair.aspx 9. Model - Repair.aspx 10. Phone - Repair.aspx 11. Type - Sell.aspx 12. Manufacturer - Sell.aspx 13. Model - Sell.aspx 14. Phone - Sell.aspx |
| User/Role management for at least 2 roles | Roles Supported:   1. Admin 2. Users |
| Preservation of state (may include cookies, session state, application state, cross-page posting, or querystring) | When an employee logs on to the system, her employeeID is stored in a session variable. This variable is accessed on all other employee pages in the system. |
| 10 different ‘basic’ ASP.NET Web controls (those that map directly to an html tag) | Admin.aspx:   1. TextBox1 (TextBox) 2. lblEmployeeID (Label) 3. ddlEmployees (DropDownList) 4. … |
| 5 different ‘advanced’ ASP.NET Web controls (those that do not map  directly to an html tag) | Default.aspx:   1. SqlDataSource1 (SqlDataSource) 2. DataList1 (DataList)   Login.aspx:   1. Login1 (Login)   SellItem.ascx:   1. ConfirmButtonExtender1 (Ajax Toolkit ConfirmButtonExtender)   ForgotPassword.aspx:   1. PasswordRecovery1.aspx (PasswordRecovery) |
| Input validation for **all** free-form user entries | SellItem.ascx:   1. RequiredFieldValidator1 – Make sure they select a condition. 2. RequiredFieldValidator2 – Make sure they select an image to upload. |
| 1 user control | 1. SellItem.ascx: A user control that allows the user to select Device Condition and upload an image and insert in to the database. |
| 1 navigation control | 2. SiteMapPath1: A SiteMapPath control included on SiteMaster.Master |