BOM

The specific description of each page is explained in a better way in the source code itself in the form of comments, wherever needed. This description focuses on the overall working of each page. The description will be according to the file structure actually used in the project done in Eclipse. Overview or working of each page, whenever needed, will be given.

# bom

## calendar

## controllers

### addReportSubmit.php

### addressSubmit.php

### bOMSubmit.php

### boxSubmit.php

### brandSubmit.php

### commercialNameSubmit.php

### companySubmit.php

### contactSubmit.php

### datePicker.php

### db.php

### editBrandSubmit.php

### editPhoneSubmit.php

### editRangeSubmit.php

### finishingSubmit.php

### func.php

### functions2.php

### header.php

### headerAdmin.php

### headerAll.php

### headerOthers.php

### itemOrderSubmit.php

### materialColorSubmit.php

### materialPatternSubmit.php

### materialTypeSubmit.php

### meetingSubmit.php

### newContactSubmit.php

### packagingFaceSubmit.php

### phoneSubmit.php

### pOPISubmit.php

### productTermsSubmit.php

### rangeSubmit.php

### realPhoneSubmit.php

### requestSubmit.php

### sKUSubmit.php

### subFamilySubmit.php

### validationPage.php

This is the page where the user authentication is verified. If the user name or password doesn’t match with the entries in database for a user name, the user is notified of it. If the user is authenticated correctly then he/she is rerouted to a dashboard page according to the role assigned to him/her.

## css

This folder contains the .css style sheets needed by the bootstrap and all the pages itself. The following are the 2 main css files used.

### bootstrap.min.css

### bootstrapmain.min.css

## Images

This folder will contain all the images to be used in different pages for the project.

## js

This folder contains the javascript pages needed by the bootstrap.

## views

### admin

### admin.php

This is the dashboard page for the users “admin”. This page has links to reroute to different pages accessible to this user.

### editBrand.php

Using this page the “admin” users can edit an existing brand entity present in the database. Here the user has to specify the brand name to specify which brand he/she wants to edit, after which he/she is given the editable attributes to change, like brand name and other attributes, which after submitting is then sent to ***editBrandSubmit.php*** in /controllers.

### editPhone.php

Using this page the “admin” users can edit an existing phone entity present in the database. Here the user has to specify the phone name to specify which phone he/she wants to edit, after which he/she is given the editable attributes to change, like phone name and other attributes, which after submitting is then sent to ***editPhoneSubmit.php*** in /controllers.

### editRange.php

Using this page the “admin” users can edit an existing range entity present in the database. Here the user has to specify the range name to specify which range he/she wants to edit, after which he/she is given the editable attributes to change, like range name and other attributes, which after submitting is then sent to ***editRangeSubmit.php*** in /controllers.

### requestPage.php

Using this page the “admin” users can create a new request BOM entity for placing an order for the different products provided by the company. Here the user has to specify the range name, the phone brand name, the phone name and other information needed to place a request, which is then sent to ***requestSubmit.php*** in /controllers.

### others

### others.php

This is the dashboard page for the users which are not “admin”. This page has links to reroute to different pages accessible to this user.

### meetingPage1.php

Using this page any of the users can create a new meeting entity for the interactive meetings which take place in the company or are associated with the company. Here the user has to specify the date on which the meeting was held and other information needed, which is then sent to ***meetingSubmit.php*** in /controllers.

### meetingPage2.php

If the ***meetingPage1.php*** is properly submitted and is validated properly, creating the meeting entity, the user is then rerouted to this page. Here user can add attendees and reports to the meeting using the ***contactSubmit.php*** and ***addReportSubmit.php*** pages. Here the user has to specify the contact to be added or the report to be added. If the contact doesn’t exist user can create a new contact and then add it to the meeting as an attendee.

### newAddress.php

Using this page any of the users can create a new address entity for the different companies related to the system. Here the user has to specify the company name and the address for the company along with the district name, which is then sent to ***addressSubmit.php*** in /controllers.

### newBOM.php

### newBox.php

Using this page any of the users can create a new box entity for the different product samples produced and stored by the company. Here the user has to specify the name for the new box and its dimensions, which is then sent to ***boxSubmit.php*** in /controllers.

### newBrand.php

Using this page any of the users can create a new brand entity for the different products produced by the company. Here the user has to specify the name for the new brand, its brand type and other information which is then sent to ***brandSubmit.php*** in /controllers.

### newCommercialName.php

Using this page any of the users can create a new commercial name entity for the different products produced by the company. Here the user has to specify the name for the new commercial name, which is then sent to ***commercialNameSubmit.php*** in /controllers.

### newCompany.php

Using this page any of the users can create a new company entity for the different companies associated with the company. Here the user has to specify the name for the new company, its activity type, and other information, which is then sent to ***companySubmit.php*** in /controllers.

### newContact.php

Using this page any of the users can create a new contact entity for the different roles represented in the company. Here the user has to specify the name for the new contact, its company name, and other information, which is then sent to ***newContactSubmit.php*** in /controllers. There are 2 entry points for this page. One is the direct way which is present as a button on the dashboard of users. Second is when a meeting is created, and its attendees are to be listed. If the contact is not already present it can created and added directly to the meeting via a link in the ***meetingPage2.php*** page to add a new contact.

### newFinishing.php

Using this page any of the users can create a new finishing entity for the different products produced by the company. Here the user has to specify the name for the new finishing, its pattern type, and its color type, which is then sent to ***finishingSubmit.php*** in /controllers.

### newItemOrder.php

If all the values which ***pOPISubmit.php*** gets are proper and the PO/PI is created then the user is rerouted to this page. On this page user can place an item order, where he/she has to specify the BOM associated to the PO/PI, SKU associated to it and other information needed to place an item order. After the form is filled the inputs are sent to ***itemOrderSubmit.php*** for verification and if correct insertion, in /controllers.

### newMaterialColor.php

Using this page any of the users can create a new material color entity for the different products produced by the company. Here the user has to specify the name for the new material color, which is then sent to ***materialColorSubmit.php*** in /controllers.

### newMaterialPattern.php

Using this page any of the users can create a new material pattern entity for the different products produced by the company. Here the user has to specify the name for the new material pattern, which is then sent to ***materialPatternSubmit.php*** in /controllers.

### newMaterialType.php

Using this page any of the users can create a new material type entity for the different products produced by the company. Here the user has to specify the name for the new material type, which is then sent to ***materialTypeSubmit.php*** in /controllers.

### newPackagingFace.php

Using this page any of the users can create a new packaging face entity for the packaging of different products produced by the company. Here the user has to specify the name for the new packaging face and the dimensions for it, which is then sent to ***packagingFaceSubmit.php*** in /controllers.

### newPhone.php

Using this page any of the users can create a new phone entity for the different products produced by the company. Here the user has to specify the name for the new phone, which brand it is from, and other details associated with it, which is then sent to ***phoneSubmit.php*** in /controllers.

### newPOPI.php

Using this page any of the users can create a new product order/ product invoice entity for the different products produced by the company. First user has to specify which master P O the new PO/PI will be associated to. After the master P O is selected, the user has to specify what he/she wants to create viz. a PO or a PI. After both of things are selected a form for the corresponding is loaded. If no master P O is selected the new form is supposed to be a master P O and is stored in the ***master\_p\_o*** table. After the form for PO/PI is filled properly, it is submitted to ***pOPISubmit.php***. If all the entered values are correct, the PO/PI is stored in the ***P\_O\_P\_I*** table.

### newProductTerms.php

### newRange.php

Using this page any of the users can create a new range entity for the different products produced by the company. Here the user has to specify the name for the new range, which sub family it is from, and the commercial name associated with it, which is then sent to ***rangeSubmit.php*** in /controllers.

### newRealPhone.php

### newSKU.php

Using this page any of the users can create a new sku entity for the different products produced by the company. Here the user has to specify the name for the new sku, its ean number, and the bom id associated with it, which is then sent to ***sKUSubmit.php*** in /controllers.

### newSubFamily.php

Using this page any of the users can create a new sub family entity for the different products produced by the company. Here the user has to specify the name for the new sub family and its family type, which is then sent to ***subFamilySubmit.php*** in /controllers.

## index.php

This is the login page for the project. Here the user is asked for his/her user id and password, which then is sent to ***validationPage.php*** in /controllers. There the authentication of the user is checked and if found a valid user, is then rerouted to different pages according to the role assigned to the user. It is included in all the folders in the project to avoid direct entry to any folder, a little, but effective step towards security.