|  |  |  |  |
| --- | --- | --- | --- |
|  | |  | |
| Master’s Thesis | Internship Report | | | |
|  |  | |  |
|  |  | | ***Amey BHATKAR***  *Software Engineering*  *EPITA International Masters*  *2012 - 2013* |







**CERTIFICATE**

For

Successful completion of internship at

***ascendeo France, La Courneuve, France***

By

***M. Amey BHATKAR***

A project report submitted to EPITA for grading the third semester of

***Master in Science***

Major:

***SOFTWARE ENGINEERING***

From

***EPITA INTERNATIONAL MASTERS***

***August 2013***

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **M. Olivier JEUSSET** | **M. Christophe ROUVRAIS** | **M. Rabih HADDAD** |
| *(CTO, ascendeo France)* | *(Director,*  *Development & International Office)* | *(Deputy Director,*  *Development & International Office)* |

# Table of Contents

[Table of Contents 2](#_Toc366598693)

[Acknowledgement 5](#_Toc366598694)

[Introduction 6](#_Toc366598696)

[My Roles and Goals 6](#_Toc366598697)

[Evolution in the project 7](#_Toc366598698)

[About ascendeo France 8](#_Toc366598699)

[History 9](#_Toc366598700)

[Missions 10](#_Toc366598701)

[Some Key Figures 10](#_Toc366598702)

[Operations 11](#_Toc366598703)

[Brands Distributed 11](#_Toc366598704)

[Carriers 11](#_Toc366598705)

[Consumer Electronics 11](#_Toc366598706)

[Mass Merchandisers 12](#_Toc366598707)

[E-commerce websites 12](#_Toc366598708)

[Manufacturers’ Brands 12](#_Toc366598709)

[Need for an ERP tool 13](#_Toc366598710)

[Added-value of the internship 14](#_Toc366598711)

[Importance of the internship 15](#_Toc366598712)

[Work environment and specifications 16](#_Toc366598713)

[Organizational aspects 17](#_Toc366598714)

[Work Breakdown Structure 17](#_Toc366598715)

[Deadlines and Criticality 22](#_Toc366598716)

[Meetings and Reviews 22](#_Toc366598717)

[Project Description & Technical Aspects 23](#_Toc366598718)

[Forecast - database 23](#_Toc366598719)

[Forecast - web pages 26](#_Toc366598736)

[Work flow in forecast tool 35](#_Toc366598792)

[For *admin* user 35](#_Toc366598793)

[**Import to database** 35](#_Toc366598794)

[Export tables 35](#_Toc366598795)

[Create Accounts 36](#_Toc366598796)

[Delete Accounts 36](#_Toc366598797)

[View Customer Forecasts 36](#_Toc366598798)

[View Global Forecasts 36](#_Toc366598799)

[Generate Calculated Forecast 36](#_Toc366598800)

[Update Salesout Forecasts 36](#_Toc366598801)

[Generate Customer Forecasts 37](#_Toc366598802)

[Generate Global Forecasts 37](#_Toc366598803)

[For *sales* user 37](#_Toc366598804)

[Go to Mass Prod App 37](#_Toc366598805)

[Go to TTL App 37](#_Toc366598806)

[View Customer Forecasts 37](#_Toc366598807)

[View Global Forecasts 38](#_Toc366598808)

[BOM - database 38](#_Toc366598809)

[BOM – web pages 48](#_Toc366598873)

[Work flow in BOM tool 56](#_Toc366598949)

[New / Create Pages 57](#_Toc366598950)

[Edit Pages 57](#_Toc366598951)

[List pages 57](#_Toc366598952)

[Display pages 58](#_Toc366598953)

[Problem Pages 58](#_Toc366598954)

[Technologies Used 58](#_Toc366598955)

[PHP 58](#_Toc366598956)

[Eclipse Juno 59](#_Toc366598957)

[MySQL 59](#_Toc366598958)

[JavaScript 60](#_Toc366598959)

[jQuery 60](#_Toc366598960)

[AJAX 61](#_Toc366598961)

[HTML & CSS 61](#_Toc366598962)

[ERP 62](#_Toc366598963)

[Security Techniques Implemented 63](#_Toc366598964)

[SQL Injection 63](#_Toc366598965)

[Session Management 63](#_Toc366598966)

[Authorization & Access Control 64](#_Toc366598967)

[Possible Technical Choices 64](#_Toc366598968)

[First Review 65](#_Toc366598969)

[Added - Value to the company 65](#_Toc366598970)

[Added – Value to Myself 66](#_Toc366598971)

[Conclusion 67](#_Toc366598972)

[Bibliography 68](#_Toc366598973)

[Abbreviations 69](#_Toc366598974)

[Notes 70](#_Toc366598975)

# Acknowledgement

I would like to take this opportunity to acknowledge all those who helped me with their expertise to accomplish what I done have so far in my internship in ascendeo France as a “developer”.

First and foremost, I would like to thank M. Philippe LANG and M. Nicolas CECCALDI, the CEOs of ascendeo France, M. Olivier JEUSSET and M. Olivier YANG, who interviewed me and gave me the opportunity to work in this auspicious company. Also I would like to thank M. Neil MINKLEY for referring me to ascendeo France.

I would also like to thank M. Piotr KOROSTELEV and M. Benjamin LEMOINE, for whom I have developed the ERP tools for, for all their guidance and support. I would especially like to thank them and Mme. Yun POTHAIN, for all their help and support while I was working in ascendeo China.

I am also thankful to EPITA and its authorities, M. Christophe ROUVAIS and Mlle. Albane ARIZA, for giving me the opportunity to study in this renowned school.

And finally I would like to thank my family members, without whose support and blessings, I won’t have been able to reach where I am today.

# Introduction

In this section I would like to explain more about the company and my role in the company. In this section I would try to explain in detail about the context and complexity of the work I did in ascendeo France.

## My Roles and Goals

I joined the company as an **Intern Junior Developer** on 2nd of April, 2013. As discussed in the interview with M. Olivier JEUSSET and M. Olivier YANG, I was supposed to create a tool for automatizing the forecast process which was carried out at that time, manually in excel sheets. This excel sheet was handled by M. Benjamin LEMOINE, who was responsible for all the forecasts of stocks for the products in company.

In less than 3 months I was able to create the tool which calculated the forecasts as accurately as in the excel sheets. The formulas for all the calculation was taken from the macros in the excel sheets and some which I couldn’t understand, provided by M. Benjamin. I had a great lot of help in this project from M. Olivier JEUSSET for the access of the company’s database viz. SAGE, which was only accessible by limit number of admin users in the company. He provided me with lots of batch files which I could run to get the data from this database to the tool’s database to populate the database.

After the completion of the first tool, I started working on the second, rather larger, tool viz. BOM. This tool was loosely based on the structure present in the excel sheets, which were used by M. Piotr KOROSTELEV and other employees working on it. M. Piotr provided me with an idea about the database along with a class diagram of all the entities which he had or wanted to implement in this tool. He had a great input in the creation of the database for BOM module, as he has lots of ideas for the database of it.

I was able to create the database in 5 days. All the while I was working on the web pages for the insertion of data in this database. The idea we had was to create a set of 5 different pages for each entity, differentiation between these pages being their role. Within the duration of next 3 months I was able to provide with one type of page for the entities, second type of page to 3 main entities and fifth type of page to one entity.

## Evolution in the project

Only one time in the time period of 6 months, a modification has to be made in the overall structure of the working process.

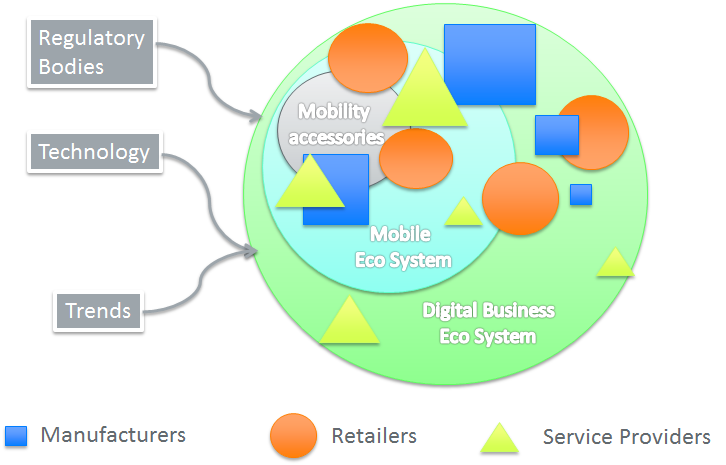
As I was a beginner in PHP with only an experience of 3 – 4 months before joining the company, and that experience being a long time back. I was unaware of frameworks used to build tools in PHP. I started working in Notepad ++ with no proper structure to the code, having both the UI and controller logic in the same page.

After 10 – 15 days of starting my internship, we had a meeting in which M. Olivier YANG helped me to understand few frameworks such as Eclipse PHP, Symphony and others. He also made me aware of the MVC structure which can be used to clearly differentiate the code in 2 parts viz. User interface and controllers behind it.

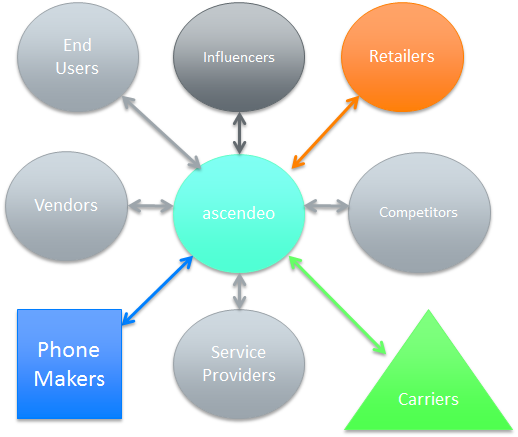
So, I had to modify the code I had made so far, which was consequential, so that it did fit the (M)VC structure. The “model” part in the MVC structure was opted to be neglected since the tool was mainly based on only database interactions and very little controller code. So we decided that we can merge both “controllers” and “model” in one single category as “controllers”.

As this happened in a very early phase of my internship, it didn’t hinder the project in any way. Rather it did helped me give a proper readability to my code, further clearly differentiating the two parts.

## About ascendeo France

ascendeo is positioned at a pivotal point of the Mobile Accessory Eco System, in the middle of all stakeholders, the role being, to analyze both demand and supply, advice both sides and maintain a balanced interest of all eco system members.

Mobility Accessory Eco System

The company has a SEER approach i.e. Social, Environmental and Ethical Responsibility. They have a global thinking with a local approach. It’s a one stop shopping.

Positioning in Eco System

### History

* A pan – European joint venture created in 2008.
* Specialized in mobile accessories (GSM, tablets, MP3s, …)
* Recognized expertize in specialized retail channels.

### Missions

* Offer the broadest possible range of products according to customer needs.
* Help customers anticipate the major market trends related to convergence.
* Propose the widest range of services
* Its on company for 3 business logics viz. Wholesaler (mainly logistics), Added-value distributor, Manufacturer (muvit, white label and licenses)

Which all comes down to this final mission, which is,

**Generate revenue growth**

**+ margin optimization**

**+ brand image**

### Some Key Figures

* Finance
  + 104 M € revenue in 2012
  + 8 M € EBITDA
* Logistics
  + 6400+ SKUs
  + 15, 000+ points of sales in Europe
  + 8 M+ € inventory
  + 1, 000+ shipments per day
* 180 employees in Europe and 160 in China
* Distribution agreements
  + OAP contracts with all mobile phone manufacturers
  + Portfolio of more than 50 brands

### Operations

* 5 European Specialists in the mobile accessory market
  + ascendeo Group (Philippe Lang, FR, CEO)
  + ascendeo France (Nicolas Ceccaldi, FR, CEO)
  + ascendeo UK and Ireland (Ryan Timoney, UK, CEO)
  + Mconomy (Chang NG, NL, CEO)
  + InOffCom (Manuel Hassig, ES, CEO)
  + Celly (Stefano Bonfanti, IT, CEO)
* 3 partners in Asia
  + Ash Cloud in Shenzhen
  + ascendeo HK in Dongguan
  + Tektos in Hong Kong

### Brands Distributed

#### Carriers







#### Consumer Electronics





#### Mass Merchandisers







#### E-commerce websites





#### Manufacturers’ Brands





## Need for an ERP tool

The company has been working for a long time now, with its inventory growing day by day. Also with its relations with so many other companies in different companies a need aroused from day one to maintain records for all the interactions and transactions and all other things.

So, all the data started being recorded in excel sheets by different people, responsible for handling and maintaining many different entities like orders, sales, forecasts, requests and many more.

As time passed these excel sheets became enormous and it became too hard to maintain the stability of these sheets with too many people interacting with it daily. Many issues arose, like data inconsistency, overwrite conflicts and others. To avoid this people were assigned to cross check and maintain them explicitly.

And so finally the need arose to take all this data to a secure place which would be easy to maintain and access. And so they decided to make an ERP tool, which would include several different “databases” to handle each different module of the product development process and other processes, and a “user interface” to interact with this data.

This was a crucial decision taken by the company, the one to secure the data in one location, and have access to it through that location, for all who needed. The idea behind it was, once the tool was complete it would take one big overhead over everyone, one of handling and maintenance of all the data for all different entities.

This is a big project, which can take years, if not many, to develop all the individual entities and later integrate them in one tool, the final ERP tool. This internship was the first step towards the development of this huge project.

## Added-value of the internship

As I was a beginner in PHP + MySQL before I joined this internship, this internship has been a great added – value to my pool of knowledge.

When I started working on this project, the idea of me developing it alone made me nervous at first but at the same time I was excited to develop something big from just an idea. As it is quoted by a famous US motivational writer and author, Earl Nightingale (1921 – 1989) “*Everything begins with an idea*”, I was experiencing a live example of it.

At the start, I had to code as per the excel sheets provided to me as there were no hard copied specifications for the project. Time to time, I used to ask M. Benjamin and M. Piotr about the specificities of the excel sheets, because most of it was completely based on business and statistical aspects.

But as time passed, I was able to create a considerable part of the project. All the while, I kept on learning new things, like Javascript, jQuery and AJAX. I was unaware of many things in the scripting languages before I started using them. I can say, I learned a great deal about the scripting languages, more than I could have learned on my own, not without using them in real time.

Also it was fun playing around with HTML / CSS, for the formatting of the user interface pages. I won’t say I have made beautiful pages, but I can say I made very simple but powerful and most importantly functional pages. I preferred concentrating on functionality of pages rather than “LnF – Look and Feel” of the pages, as that can be done on a later stage in the project.

All in all, I can say, it not only increased my knowledge of web development in the field of PHP + MySQL, but made me confident enough to actually think of it as a future career perspective. And not just PHP + MySQL but also the scripting language which mostly goes hand in hand with web development, mostly Javascript, jQuery and AJAX. Also I learnt few new things in formatting languages viz. HTML / CSS. It also made me very comfortable with the IDE: Eclipse. I also used version control tools like Tortoise SVN to manage all my files and directories, over time. It helped me have a record of my working in this project.

## Importance of the internship

As discussed earlier, there was an idea to have a system which could handle all the different entities in this product development process and other process, and store all the data in a secure place, more secure than excel sheets, one which is easy to handle and maintain.

As this is more business oriented company, its IT team consisted of only 4 members in France. And this team has 1 developer, who was handling the website and all other things related to it. He was an important enough resource to be shifted on a different project, surely not on one which was going to take a long time.

So they decided to hire an intern to start working on this project. When I was interviewed for this position, I was both nervous and excited to work on such an enormous project. I was very glad I got selected to work on this project. I knew it would be a very educative experience for me and productive for the company, likewise.

I was driven by curiosity, to make something from nothing but an idea. With very few specifications, but good visualization and creativity of all my supervisors, I was able to create a good deal of the project. I hope it is helpful for the company, what I have created so far.

## Work environment and specifications

ascendeo France being mainly a business oriented company, it doesn’t have an extensive IT department. But it is a great environment to work in nevertheless, whatever the position is. I felt very welcomed from the first day itself.

It has a very friendly work environment, where everyone is friendly enough, even after a clearly defined hierarchy. We actually had ping – pong tables, a gym, cardio and many other things to chill out when it felt too stressful.

Even after being a company not mainly concentrated on IT, I was provided with the latest technologies to use. I had my own desk with 2 computers, one a server to implement the project on, and other to do the actual development on. Since I was going to go to China, as a part of my internship, I was given a powerful laptop, to carry my work with me wherever I went.

As mentioned earlier, this project was more an innovative idea, but a desperate one at that, because of the need to shift to a database sooner than later with a proper user interface. Thus, there was not any pre – defined set of documentation for the project. The only documents provided to me were the excel sheets, based on which I had to make the web pages and the database.

M. Olivier Yang, the other developer in the company, helped me whenever I asked and whenever he could. As he was mostly busy with his own work, I asked him only in dire conditions. For most part, I used to check for answers on internet for my questions. I only use to disturb Olivier when I couldn’t find an answer over the internet, as I knew he would have the answer most of the times, to any question. He is one of the talented developers I have met, and I am glad he was there to help me whenever needed.

Overall, all the aspects helped me develop the project as efficiently as I could. If I had access to complete drafted documentations, I could have tried to develop more aspects of the project, efficiently and faster than I already have.

# Organizational aspects

In this chapter I will try to explain as briefly as I can, the flow of the project I was in, all the benchmarks reached and all the milestones achieved in the project.

It was a little difficult in the start, to work on this project as we didn’t have any drafted documentations to refer or a constricted timeline to follow. Since it was just an idea, no one really knew how much time it would take to completely finish the project, set aside the two main modules they wanted me to develop as a part of my internship.

After analyzing my skill set and the complexity of the project, a decision was reached, that I will have to make my own schedules and milestones and try to complete as much of the two modules viz. forecast and BOM, major parts of the project, as efficiently as possible.

We had to analyze a lot of things, me, M. Olivier JEUSSET and M. Olivier YANG, over what tools to use, which technology to implement, a proper strategy to finalize, make a presentation of these things and many other. I would try to explain, as best as I can, these things in detailed manner, in this chapter.

## Work Breakdown Structure

In this part I would like to describe the detailed breakdown of the work I did in my internship. For keeping a log of all the work I did, with all the development, modifications and revisions, I used the version control tool, Tortoise SVN, which proved very helpful. For further simplification, better interpretation of the work breakdown I will make use of Gantt chart which can be seen as a part of the explicit documentation provided along with the report. It’s attached as a separate documentation, so as not to increase the size of report unnecessarily.

I was interviewed by M. Olivier JEUSSET and M. Olivier YANG, in which I was explained the kind of business the company is into, and the ERP tool they wanted to develop. As they told me it was going to be an online tool, I suggested (which was also their initial thought), that this tool can be developed with one of the best web development technologies viz. PHP with the database being SQL. We chose SQL since the data, the company has, was already extensive enough, which couldn’t be handled well in excel sheets.

After all the formalities were done, required for me to start my internship, such as preparing the contract, signing of the contract and other, I started my internship in ascendeo France, as a Junior Software Developer, from April 2nd, 2013.

The first 2 weeks were spent on the analysis of the project, the technologies to be used, the hardware / software required for the project, in general the overall analysis of the project, and my part in that. In these two weeks I also tried to learn few more things in PHP + MySQL, as it was my first time on working on such a big project with PHP.

After spending 2 weeks on analysis of the project, I started to develop on April 17th, 2013. It was decided in the analysis phase, that I will develop the pages and the database required in both of modules based on the excel sheets provided to me, like the database tables will mostly be the individual worksheets in the excel workbook given to me. Also the pages which were to be developed for user interaction would roughly resemble the ones present in the sheets, mostly in the working of it.

I started with the forecast tool, as it was on priority. As for the user interaction goes in this tool, so far in the excel sheet, everything was accessible for changes if you had user rights. But as it to be implemented in web pages, they wanted to have right management for the accessibility of different aspects of forecast. So I decided to divide the users in 2 different groups, one as “*admin*” who had all the rights and “*sales*” who had limited rights. The technicality of these roles and pages assigned to each role will be explained in the latter chapter.

I started with the development of one of the main pages for the forecast tool. I was using Notepad++ to write the code in, including both, the controller logic and the user interactive forms, on the same .php file. As mentioned earlier in the report, I never had a prior experience to code in PHP on such a big project. It was too difficult to maintain these files, I understood that from the start of the project only and so I was searching for an alternative, all the while developing the tool.

On April 26th, 2013 I was able to present my supervisor, M. Olivier JEUSSET, the tool I have developed so far. It was functioning as per it was developed to. After M. Olivier JEUSSET and M. Olivier YANG had a look at the code I have coded so far, I came to realize that they were not satisfied with that.

On May 1st, 2013 we had a meeting to discuss the IDEs to use and the structure to implement for the source code, so as to make it easy to understand, implement and maintain. After a long discussion, it was decided to use MVC – Model View Controller structure for the development of the code, except for the merging of Model and Controller parts, and the use of Eclipse Juno with PHP, as a framework to write the code in. There were other options too, but I personally chose Eclipse as I had worked on it before and I was quite comfortable with using it.

So, I modified the source code I and made it fit the View - Controller structure. Also I added few parts and it was the first benchmark in the project as we had one functional part of forecast tool viz. Mass - Prod Application. This was finalized on May 3rd, 2013.

I started working on the second aspect of the forecast tool, which was the TTL Application. The skeleton was ready by May 7th, 2013.

By May 13th, 2013 the TTL Application was ready and functional.

On May 14th, 2013 the merging of both the application under the forecast tool was done. Some modifications were told to be made in the tool, and a directory was added to the project to hold all the excel sheets needed by the forecast tool. The code was modified and all the changes were made on the same day.

All the export handlers were added to the tool for getting the data present in selected tables, such as global forecast and decided forecast, into .csv files. This was implemented on May 17th, 2013. The actual calculations for the different forecasts, except for global forecast, were made and committed on the same day.

Modifications were told to be made in the code, and were changed and committed by May 21st, 2013.

A major modification in the calculated forecast was made, along with the finalized version for global forecast, on May 22nd, 2013.

By May 24th, 2013 all the functionalities were ready and was awaiting for further instructions from M. Benjamin.

Some changes were asked and were made and finalized by May 27th, 2013 and the tool was ready for its code and design check.

Some modifications were asked in the designing of the tool, which commenced on May 28th and was ready for a confirmation by May 30th, 2013.

After some minor modifications, and with a not so bad design, the tool was ready for testing the "*sales*" functionality, by May 31st, 2013.

On June 3rd, some extra functionalities were added for admin user, like create a new account, delete an existing account and others.

Some more functionalities were added for admin user on June 4th, 2013.

With some more functionalities, and minor modifications, the forecast tool was ready for its Beta test on June 12th, 2013.

On June 13th, 2013 I started the analysis of the BOM tool, whose working, for a small part of it, was explained to me, when M. Piotr came to France. After the analysis of the small part of BOM process, I started the development of few pages for the tool. They were actually supposed to be made, to let M. Piotr know the limitations and functionality, to which we can extend and develop the tool.

By June 25th, 2013 the small part was developed. M. Piotr was analyzing the functionalities of the pages and we were discussing what more can be included.

On June 27th, 2013 I departed for continuing my internship in China, as per discussed in the contract, which was signed in the beginning.

After getting to China, the actual working on the BOM tool was started. We had few meetings to discuss the final tool which was to be developed. M. Piotr had lots of ideas on how to make the tool more efficient. After long discussions about the database, the initial database was made.

By July 10th, 2013 the database along with the first initial pages with the new functionalities were made. M. Piotr reviewed them for further evaluation of the pages. We had to review the pages to see if they were meeting the requirements we needed them to have.

After further review and its extension, the first commit for BOM tool was made on July 17th, 2013.

By July 26th, 2013 we had thoroughly discussed the structure of the database and the structure we should have for the BOM tool. The final database, which included 204 tables, was created on July 24th, 2013. On July 26, I developed one of the first actually complex pages, which was for the Meeting entity. This page was supposed to create a meeting entry in the database along with the people who attended it and the documents presented (if any), in the meeting.

It was reviewed and after some modifications, was finalized on July 30th, 2013. After M. Piotr saw the functionalities to which we can extend our tool to, it was decided that I will develop the pages to create a new entry for each real time entity we had in the database, myself, taking the input from the users for the attributes of the corresponding table. After that I started making the pages for each entity according to the structure of the table.

By August 2nd, 2013 I made 11 pages to create 11 of all the main entities present in the database.

By August 6th, 2013 7 more pages for another 7 of the entities present, were made.

By August 20th, 2013 17 more pages for 17 of the entities present, were made.

On August 21st, 2013 12 more pages for 12 of the entities present, were made. After the completion of major part of the tool, I asked M. Piotr to review my work and let me know if he wants to add some functionalities or to remove some existing ones.

After many major modifications, the functioning module for the BOM tool was ready on September 6th, 2013 which included 44 web pages for the insertion of an entry in the database for the 44 main tables present in the database, 3 pages to edit / update the entries already present in the database for 3 tables and 1 display page to display a real time entity, with its relations with all other entities in real world.

## Deadlines and Criticality

As the project was not too much constricted with time, there was a bit of flexibility with time restrictions. But noting that, I tried to complete each of the benchmarks as fast as I could. In fact, now at the end of my internship it seems like we were able to achieve a lot in so less time.

But as the main focus was on the quantity of the project to be covered, the quality of the design as well as the code was not too good. It could have been improved, if only more focus was on the quality. Being said that, the authorities were very content on the percentage of completion of the project in such a small amount of time, and all this without any drafted documentations.

The overall performance could have been improved by giving a predefined set of drafted documentations rather than having the project self-evolve in the development phase itself. There was a big scope of improvement in the quality of code, reusability of code and most importantly the formatting of web pages, even if it was a designing concern.

But the results provided at each benchmark were more than satisfactory for a beginner in this particular field. And most importantly the results were confirmed for their working and design at each benchmark. A self-evolution would have been good here too, but it was avoided to avoid rewriting of the code.

## Meetings and Reviews

As mentioned above we had reviews on regular intervals and the modifications were made accordingly, as soon as possible.

As for the meetings, it was not possible to have face to face meetings with M. Piotr and M. Benjamin regularly as they were in China. But we had few online meetings for explaining me the working of both the tools viz. forecast and BOM. But, I use to have regular meetings with M. Olivier JEUSSET, as he was my supervisor in France. These were of importance as his guidance and inputs helped me in my design of the tools very much.

The regular reviews were of critical importance as it was necessary to have the pages and database resemble the requirements they had.

# Project Description & Technical Aspects

There were 2 main tasks in my internship, which was to make the 2 tools, viz. forecast and BOM. I will try to explain each of the tasks, with a brief description for their databases and the source code.

## Forecast - database

The structure for all the tables can be seen in the data dictionary, because including it here would make the description unnecessarily large and confusing. Here only the important attributes would be mentioned and the tables will be described in a generalized way.

### calculated forecast

This table holds the forecast generated by calculations for the current week and for 25 weeks after, which is calculated based on the history of sales for a particular product by a particular customer for the last 4 months. Using the history for last 4 months the forecast for current month is generated. And then the forecast for the current week along with the forecast for next 25 weeks is generated based on this forecast value. The calculations for this forecast is generated in .php pages and then stored in this table.

### customer forecast

This table holds the actual forecast values submitted by customers via the sales managers in the ***forecast\_decision\_table***. Like ***calculated\_forecast*** table, this table too holds forecast for the current week and 25 weeks after that. But unlike calculated forecasts, the customer forecast is given by the customers based on their own evaluations. A row in this table signifies the forecast for a particular product for a particular customer. And if for a product – customer couple, a forecast is not given by the customer itself, the values for the forecast are taken from ***calculated\_forecast*** for that particular product – customer couple. The calculations for this forecast is generated in .php pages and then stored in this table.

### customer prod relation

This table holds the relation between the product and a customer, based on the minimum and maximum stock values and comments if any. This table is used to differentiate between the forecasts and the relation of a product to a customer.

### customer spec

This table holds the specifics about the customer needed by the system to describe a customer entity.

### cust prod current

This is one of the table that is used by the system in a unique way. It is a single row table, which is modified every time a selection for a product and customer couple is made in the .php pages. It helps us to know the current product and customer couple, which are needed to generate the different values needed in the pages.

### dev

This table holds all the specifics about the development products needed by the system to describe an entity of product under development.

### forecast decision table

This table holds the actual forecast values submitted by customers via the sales managers. Like ***calculated\_forecast*** table, this table too holds forecast for the current week and 25 weeks after that. But unlike calculated forecasts, the customer forecast is given by the customers based on their own evaluations. A row in this table signifies the forecast for a particular product for a particular customer. The values for this forecast is submitted through a form in .php pages and then stored in this table.

### global forecast

This table holds the forecast values for a particular product from all its customers. Like the other forecast tables, this table also has forecasts for the current week and 25 weeks after that. It also has 2 other attributes viz. MM and Decided Volume, which are needed by the system for further process in the product development cycle.

### history

This table holds all the information needed by the system to indicate the history of a product with a particular customer till date.

### liste display

This table is used for a specific purpose in the .php pages. It holds the list of a product – customer couple along with its monthly orders. The list of products for a particular customer are divided in three parts, first being the recent products it interacted with, second being the list of products with history of monthly orders sorted in an descending way and third is the list of products which are EOL for that particular customer. Using this table the required list can be generated quite easily.

### lt (lead time)

This table holds the lead time specified by a customer for a particular product. It is useful for the forecast, as the forecasts are to be given or generated based on this lead time.

### main table entries

This is second of the unique tables mentioned earlier. This too is a single row table. This table holds all the information related to the current customer – product couple.

### orders

This table holds the history of orders by a customer for a particular product. This table is used in forecast generation/ evaluation as well.

### product list

This table holds all the information needed by the system to describe a product entity.

### product spec

This table holds most of the information needed by the system to describe a product – customer couple.

### salesout

This table holds the information needed by the system to describe the sales out stock of a product by a particular customer. This table holds the warehouse and agencies stock for a particular product for a customer. It also holds the actual stock of a product for a customer for last 6 weeks and also the stock for the current week. Based on this stock for the next 25 weeks is calculated using .php pages.

## Forecast - web pages

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.

### controllers

#### class

This folder contains all the files needed to generate the graph files, store them in indicated location and display them wherever needed. It contains the three important class.php files viz. ***pDraw.class.php***, ***pImage.class.php*** and ***pData.class.php***. They are included in all the pages which need graph to be generated and displayed, currently in ***getCurrentValues.php***.

#### delayedLoader

This folder contains a gif file to indicate the delay before generating and loading the graph on php pages. It also contains some demo files to demonstrate the graph generation and display them.

#### exportHandler

##### CustProdRelnExport.php

This page is used to get the values from ***customer\_prod\_relation*** table and store them in a csv format, which can then be exported as .csv file.

##### ForecastDecisionTableExport.php

This page is used to get the values from ***forecast\_decision\_table*** table and store them in a csv format, which can then be exported as .csv file.

##### GlobalForecastExport.php

This page is used to get the values from ***global\_forecast*** table and store them in a csv format, which can then be exported as .csv file.

#### fonts

This folder contains the different fonts needed by the graph library used in the project for generating the graphs for forecast values.

#### imageMaps

This folder contains the image maps needed by the graph library used in the project for generating the graphs for the forecast values.

#### pictures

This folder contains all the images and graphs, needed or generated in the project.

#### calculatedForecast.php

This page is used to calculate the calculated forecast values based on the history of sales for the last 4 months which we get from ***history*** table, for all the different customer – product couple. After calculation the values are stored in ***calculated\_forecast*** table in the database.

#### createAccount.php

This page takes the name, password and the role of the user to be created from ***createAccounts.php*** in /adminViews. The name is then searched in the database and if a user is found with same name the administrator is notified. If no matching entry is found this new user is created and given the corresponding role.

#### customerForecast.php

This page is used to calculate the customer forecast values based on the calculated forecast list and the decided forecast list which we get from ***calculated\_forecast*** and ***forecast\_decision\_table*** tables, for all the different customer – product couple. After calculation the values are stored in ***customer\_forecast*** table in the database.

#### db.php

This page holds all the information needed by the system to connect to the database. It is included in all the /controller pages since almost all need to connect to the database in one or another way.

#### deleteAccount.php

This page takes the name of the user to be deleted from ***deleteAccounts.php*** in /adminViews. The name is then searched in the database and the related entry is deleted along with the user itself.

#### displayCustomerForecasts.php

This page is used to display all the values of customer forecasts for all the customer – product couple who has forecasts and have values stored in ***customer\_forecast*** table in database accordingly. It is referenced from both admin and sales pages for displaying the list. The search is a complete search or based on a keyword which is matched with the product name or customer name or both.

#### displayGlobalForecasts.php

This page is used to display all the values of global forecasts for all the products who has forecasts and have values stored in ***global\_forecast*** table in database accordingly. It is referenced from both admin and sales pages for displaying the list. The search is a complete search or based on a keyword which is matched with the product name.

#### func.php

This page is basically used to store all the functions used in the different pages via direct calls or calls through javascript. It is included in most of the pages for direct calling and referenced by many for calling through javascripts.

#### getCurrentValues.php

This page is responsible for getting all the different values needed by the *mainPage.php* and *mainTTLPage.php* pages for displaying. It takes the customer – product couple from cust\_prod\_current table and gets their corresponding values to be displayed from various tables.

#### globalForecast.php

This page is used to calculate the global forecast values based on the customer forecast list for all the different products. After calculation the values are stored in ***global\_forecast*** table in the database.

#### header.php

This is the header for the ***index.php*** page.

#### headerAdmin.php

This is the header file for all the admin pages like ***admin.php***, ***calculateForecasts.php*** and others.

#### headerAll.php

This is the header file for ***mainPage.php*** and ***mainTTLPage.php***. Two different headers for sales pages are created because the display size of both the pages varies.

#### headerSales.php

This is the header file for all the sales pages. It is included in the headers of all sales pages like ***displayCustomerForecast.php***, ***sales.php*** and others, except for mainPage.php and mainTTLPage,php.

#### logout.php

When the user presses on the sign out on any of the pages, he/she is rerouted to this page, where the session is destroyed and the user is then rerouted to the main ***index.php*** login page.

#### salesout.php

This page is used to calculate the salesout values based on the history of stock for the last 4 months. After calculation these values are stored in the ***salesout*** table in the database.

#### updateDecisionTable.php

This page contains the code for getting the forecast value, comments and minimum stock values from the user through a form and store them ***forecast\_decision\_table*** and ***customer\_prod\_relation*** tables.

#### updateNames.php

This page contains the code for setting the current customer – product couple in the ***cust\_prod\_current*** table. It also calculates the values which are needed for displaying and calculating purposes in the forecast of a product for a customer. It then stores these values in the ***main\_table\_entries*** table.

#### validationPage.php

This page is where the validity of login will be checked and if the user is a valid user then he/she will be rerouted to ***admin.php*** or ***sales.php*** according to his/her role.

### css

This folder contains the .css’ needed by the bootstrap as well as by the pages in the project.

#### bootstrap.min.css

#### bootstrapmain.min.css

### excelsheets

This folder should contain the excel sheets/ .csv files after exporting the needed tables in the project.

### img

This folder will contain all the images needed by the bootstrap.

### js

This folder contains the javascript pages required by the bootstrap.

### views

#### adminViews

##### adminPage.php

This will be the main dashboard page for the users with role defined as “admin”. It has links to other pages which are made accessible to these users.

##### calculateForecasts.php

This page asks for a confirmation from the user, whether he/she actually wants to calculate the calculated forecasts. It is implemented because the calculation of forecasts takes a long time and sometimes user may not want to spend that time right away. If accepted the user is rerouted to ***calculateForecast.php*** in /controllers.

##### createAccounts.php

This page is basically used by this user to create login accounts for new users. Their name, password and roles are taken through a form and sent to ***createAccount.php*** in **/controllers** for validation, processing and insertion, after submission.

##### customerForecasts.php

This page asks for a confirmation from the user, whether he/she actually wants to calculate the customer forecasts. It is implemented because the calculation of forecasts takes a long time and sometimes user may not want to spend that time right away. If accepted the user is rerouted to ***customerForecast.php*** in **/controllers**.

##### deleteAccounts.php

This page is basically used by this user to delete login accounts for old users. Their name is taken through a form and sent to ***deleteAccount.php*** in **/controllers** for validation, processing and deletion, after submission.

##### displayCustomerForecasts.php

This page is used only for displaying all the forecasts for all the products for their corresponding customers, we currently have in the system. The list either can be complete or shortened by submitting a keyword which matches itself to product names and customer names and displays the results accordingly.

##### displayGlobalForecasts.php

This page is used only for displaying all the forecasts for all the products we currently have in the system. The list either can be complete or shortened by submitting a keyword which matches itself to product names and displays the results accordingly.

##### exportHandler.php

This is an index page, with links to ***CustProdRelnExport.php***, ***ForecastDecisionTableExport.php*** and ***GlobalForecastExport.php*** in **/controllers.** It is used to export the corresponding tables in the form of .csv files.

##### fillSalesout.php

This page asks for a confirmation from the user, whether he/she actually wants to calculate the calculations needed to generate the ***salesout*** table in database. It is implemented because the calculation of such things take a long time and sometimes user may not want to spend that time right away. If accepted the user is rerouted to ***salesout.php*** in **/controllers**.

##### globalForecasts.php

This page asks for a confirmation from the user, whether he/she actually wants to calculate the global forecasts. It is implemented because the calculation of forecasts takes a long time and sometimes user may not want to spend that time right away. If accepted the user is rerouted to ***globalForecast.php*** in **/controllers**.

##### importHandler.php

This page currently is empty, but it is created with an idea to let admin user import data to the database and server as needed directly from either the terminal he/she working on, or from other source. The working for this page is still to be thought of. Currently the data is imported into the database through an already implemented database by the company in SAGE, using .dat files, which is done by M. Olivier JEUSSET.

#### salesViews

##### displayCustomerForecast.php

This page is used only for displaying all the forecasts for all the products for their corresponding customers, we currently have in the system. The list either can be complete or shortened by submitting a keyword which matches itself to product names and customer names and displays the results accordingly.

##### displayGlobalForecast.php

This page is used only for displaying all the forecasts for all the products we currently have in the system. The list either can be complete or shortened by submitting a keyword which matches itself to product names and displays the results accordingly.

##### mainPage.php

This will be the main page for the Mass Production application, a page which takes forecasts sales values, and other values and sends them to ***updateDecisionTable.php*** for validation, processing and insertion in the database, for a product related to a particular customer, for all the products which are in mass production only. Only after selection of customer and product and submitting the selection to ***updateNames.php***, the form for that corresponding couple is shown.

##### mainTTLPage.php

This will be the main page for the TTL application, a page which takes forecasts sales values, and other values and sends them to ***updateDecisionTable.php*** for validation, processing and insertion in the database, for a product related to a particular customer, for all the products which are in under development only. Only after selection of customer and product and submitting the selection to ***updateNames.php***, the form for that corresponding couple is shown.

##### salesPage.php

This will be the main dashboard page for the users with role defined as “sales”. It has links to other pages which are made accessible to these users.

### index.php

This will be the main index page, where user will be asked for his/her login and password. After the credentials are entered it will be sent to the validation page in **/controllers** for verification. This page is placed in all folders for avoiding direct entry to any folder, a small step towards securing unwanted entry. Also each page is given a session, so that when it expires, which it does after signing out or no interaction with the page for a certain amount of time, the user is indicated the session ending message and given a redirection to this page for restarting the session.

## Work flow in forecast tool

There are two different users for this tool viz. *admin* and *sales.* The work flow for each of the user type is different than that of the other. In this tool, mostly all pages are different than the others, except for certain base similarities, so the work flow for each page is different.

### For *admin* user

This user type has total 10 different options to select from. Some of them are interdependent, and some are independent.

#### **Import to database**

When **Import to database** button is clicked a batch file is ran which updates all the tables in the database. This was made since most of the tables in database are based on values for a week, or a month. So at the start or end of each week or start or end of each month this set of batch files are ran. Then these batch files access the SAGE database we have and get all the values needed to populate the database for this tool. These batch files were provided by M. Olivier JEUSSET.

#### Export tables

As mentioned earlier the system is made in such a way that it calculates the values for different tables based on values in certain tables for the current week and last 4 months. And since this being a forecast tool, it holds information for next 25 weeks. So as to have a log of all these things for future use we needed to have the values stored somewhere other than in database itself. Also saving these values in database was of not much use, and eventually it would have resulted in the overflow of database.

So whenever **Export tables to excel sheets** is clicked the 3 main tales viz. global forecast, decided forecast and customer product relation tables are exported to .csv files which can later be converted to excel sheets.

#### Create Accounts

When **Create Accounts** is clicked, a form is retrieved and presented to the user to create a new user with a password and a role for tool accessibility. Here the *admin* can also see the existing users, so as to avoid getting a ‘*User Exists*’ message.

#### Delete Accounts

When **Delete Accounts** is clicked, the *admin* can select from different users, which already have access to the tool, and delete their account, so as to block them from having further access to the tool.

#### View Customer Forecasts

When **View Customer Forecasts** is clicked the user can view all the different forecast values for all the customer – product couples.

#### View Global Forecasts

When **View Global** **Forecasts** is clicked the user can view all the different forecast values for all the products.

#### Generate Calculated Forecast

When **Generate Calculated Forecast** is clicked, a big and complex script is ran to calculate the calculated forecasts, which is used by many other forecasts and eventually by global forecast. The formula for these calculations, and many others, were defined by M. Benjamin LEMOINE, in the excel sheets using macro. These were of critical importance as they were specifically designed for the forecast purpose of the company.

#### Update Salesout Forecasts

When **Update Salesout Forecasts** is clicked, the columns whose values are dependent on the orders of that particular product for the particular customer for last 4 months, which is present in orders and history table, are calculated and then stored in the salesout table.

#### Generate Customer Forecasts

When **Generate Customer Forecast** is clicked, a big and complex script is ran to calculate the customer forecasts, which is used by many other forecasts and eventually by global forecast.

#### Generate Global Forecasts

When **Generate Global Forecast** is clicked, a big and complex script is ran to calculate the global forecasts, which is mainly dependent on decided forecasts, calculated forecasts and customer forecasts. The output of this process is a global forecast for the all the products manufactured and sold by the company. This forecast plays a pivotal role in the production of the products for the company.

### For *sales* user

#### Go to Mass Prod App

When **Go to Mass Prod App** is clicked, the user is forwarded to the main page for all the products which are in mass production already. After the relevant customer and product is selected, the user gets a form in which he / she can see all the different information related to this couple, enter certain important values like the decided forecast, minimum order value and some comments. A graph is also generated depicting the forecast over 26 weeks. This graph changes each time any value is changed on the form.

#### Go to TTL App

When **Go to TTL App** is clicked, the user is forwarded to the main page for all the products which are in development or confirmation process. Other than that the functionality of this page is almost similar to the Mass Prod App, except for certain specific differences.

#### View Customer Forecasts

When **View Customer Forecasts** is clicked the user can view all the different forecast values for all the customer – product couples.

#### View Global Forecasts

When **View Global** **Forecasts** is clicked the user can view all the different forecast values for all the products.

## BOM - database

The structure for all the tables can be seen in the data dictionary, because including it here would make the description unnecessarily large and confusing. Here only the important attributes would be mentioned and the tables will be described in a generalized way.

### address

This table holds the address for the companies related to the company in any way possible. A dedicated table for address is used since one company can have workstations at more than one place. Its primary key is its UID which would be used in ***address\_company,*** which helps us understand that one company can have more than one address associated to it, as a foreign key. It has a *disctrict\_ID* associated with it to indicate the district it is in.

### ai\_text

### benefit

This table holds the name of the benefits there are and its corresponding *UID,* a primary key for the table, which becomes the foreign key for certain tables.

### bom

This table holds all the essential information to place an order for a product to a company like the range, phone, brand, the *request\_BOM\_ID* (a foreign key from ***request\_BOM*** table, indicating the request\_BOM number), bulk and full prices along with the minimum order quantity and other things. Its primary key, its UID, which becomes foreign key for certain tables.

### box

This table holds all the information related to a box, that is, its name and dimensions.

### brand

This table holds all the information related to a brand of whose product we have to create along with its owner and other details related to that particular brand.

### brand\_category

Brand is categorized mainly into 3 categories viz. Distribution, marque constructeur and Ascendeo. This table holds these names along with their IDs which are used in ***brand*** table as *brand\_category\_type*.

### commercial\_name

This table holds the list of commercial names currently in use for all the ranges.

### companies

This table holds all the information needed by the system for a particular company like the date when it was added to the list, name, website, address and other info.

### company\_activity

This table hold the information indicating all the types of activities a company has.

### component type

This table holds all the different packaging components specific to phone models and the names of criteria’s for each of them. Its UID is sent to ***packaging\_component*** table, to indicate the the type of packaging component that particular entry is related to, and the names of the criteria’s whose values are stored in packaging component table for that particular entry.

### contact

This table holds all the information needed by the system for a particular contact like name, company from which that person is from, language that person speaks, mail and other info.

### defect

This table holds all the information related to the various defects related to many other entities in product development process, like its name, importance and description of the defect.

### district

This table holds the different districts along with their town\_ID, indicating from which town the district is. Its UID is sent to ***address*** for indicating the district the address is in.

### document

This table holds all the information needed by the system for a document, like the name, path where it’s stored and any comments related to the document. It is used in many different many other tables as reference.

### document\_ext

This table holds all the different types of document extensions used by the system. Its UID is sent to ***document*** table for indicating the extension type.

### document\_type

This table indicate the type of document like meeting report document and other info. Its UID is sent to ***document*** table for indicating the type of document it is.

### event\_type

This table is used in a very special case. The system holds a lot of dates and it’s needed to be related to most of the status tables. This table indicates the event at which an entry is to be entered in ***xxx – status timing*** table, the events being request, promised or real.

### expedition

This table holds all the information related to the shipment of products, like tracking\_number, sender\_ID, receiver\_ID and other info. Its UID is sent to ***sample*** and ***device*** to indicate the expedition of the sample or device respectively.

### exp\_companies

This table holds the different companies which provide shipment for our products, viz. TNT, UPS and others.

### family

This table holds all the families of products related to the company. Its UID is sent to the ***sub – family*** table as a family reference.

### finishing

This table holds all the different type of finishing we have for all the different products. Its UID is sent to other tables as a reference to the finishing used.

### interest\_in

This table holds the range of interest (from 1 to 5) for the selection or approval of a particular entity in the system.

### item\_order

This table holds all the information needed by the system, related to the ordering of an item, like BOM\_ID, SKU\_ID, PO\_ID, quantity and other info.

### master\_p\_o

The system is built in such a way that, when a product order is placed, it can be verified and re-verified again, using the same PO\_ID. So it is necessary to keep a track of the original PO\_ID. This table holds the PO\_ID using which the order was placed for a particular product. Its UID is sent to ***P\_O\_P\_I*** table as a reference to the master\_P\_O\_ID.

### material

This table holds all the information needed by the system, related to material, like material type id, material color id, material pattern id and other info. Its UID is sent to ***sets*** table to indicate the material used for the different products.

### material color

This table holds all the colors available for different materials used for the products. Its UID is sent to ***material*** table to indicate the material color that is used.

### material pattern

This table holds all the different patterns available for materials used for the products. Its UID is sent to ***material*** to indicate the material pattern that is used,

### material type

This table holds all the different types of materials used for the products. Its UID is sent to ***material*** table to indicate the material type that is used.

### meeting

This table holds all the information needed by the system, needed for a particular meeting, like meeting date, meeting place, summary, and other info. Its UID is sent to ***attendees*** and ***report*** tables to indicate the contacts attending the meeting and documents used in meeting, respectively.

### mold terms

This table holds the primary keys of ***phone\_size\_type*** and ***range\_supplier*** tables, along with other info like cost, MOQ, and other info, which indicates the different companies supplying a particular range for the particular phone size type.

### numbered status tables

As described earlier, this system works in a unique way. A row for certain entities like phone, product order, can be associated to lots and lots of dates. So instead of including them in those tables itself, we used 2 extra tables. The first of them is this, which holds the list of all the status tables for which dates are to be maintained. Its UID is sent to ***xxx – status timing*** table to indicate which table’s entry’s dates are it (for a particular row).

### packaging component

This table holds the primary key of ***component\_type*** table and the criteria values for that particular type.

### packaging face

In the system, it was needed to clearly distinguish between the front face of packaging and side face of packaging, since one type of packaging face can be used with more than one packaging of different widths. This table holds all the information needed to describe a packaging face, for a particular packaging, like its name, height and length. Its UID is sent to ***packaging\_size*** table to indicate the face used for that particular entry.

### packaging size

This table holds the primary key of ***packaging\_face*** table along with other information like its name and width, which indicates the dimensions of the actual packaging.

### packaging type

This table holds all the information needed by the system, to describe the packaging types used in the system. Its UID is sent to certain other tables to indicate the packaging type used in their entries correspondingly.

### phone characteristics

This table holds the names of the different characteristics that can be related to a phone. Its UID is used in ***phone\_characteristics\_defects*** table to indicate the defect related to that particular characteristic.

### phone model

This table holds the primary keys of ***brand***, ***size\_kind*** and ***size\_type*** tables, along with other information, like screen size, popularity, release date and other info, needed by the system to describe an actual phone model. Its UID is used in certain tables to indicate the phone model related to that particular entry in that table.

### pms

This table holds all the information needed by the system to describe a particular project manager. Its UID is used in certain tables to indicate the project manager responsible or related to that particular entry in that particular table.

### popularity

This table holds the information needed by the system to describe popularity of a certain entity, Its UID is sent to ***phone\_model*** table to indicate the popularity of that particular phone model.

### product component

This table holds the primary key of ***component\_type*** table and the criteria values for that particular type.

### product terms

This table holds the primary keys of ***phone\_size\_type*** and ***range\_supplier*** tables, along with other info like cost, MOQ, and other info, which indicates the different companies supplying a particular range for the particular phone size type.

### p o p i

This table is unique for system. Instead of making 2 different tables each for product order and product invoice, they were merged in one single table, generalizing the process. This table holds the primary key of ***master\_p\_o*** table, to indicate the original product order which is related to an entry. Product order and product invoice is differentiated using a column name *PO\_Type* which indicates what type of entry it is. It also has other information like customer company id, supplier company id, packaging supplier id, and other info, sufficient enough for the system to describe the product order/ invoice. Its UID is used in ***master\_p\_o*** table to indicate the master product order, which then is used in different po’s or pi’s.

### quality standard

This table holds all the names of the quality standards needed by the system. Its UID is sent to ***qs\_defect*** to indicate the defects related to that particular quality standard.

### range

This table holds the primary keys of ***sub\_family*** and ***commercial\_name*** tables, along with other information needed by the system to describe a range. Its UID is sent to certain tables to indicate the range used in that particular entry for that particular table.

### real mold

This table holds the primary key of ***mold\_supplier*** table along with other information like date when it was created, its quality and other info, which is enough for the system to describe an actual mold entity.

### real packaging

This table holds the primary keys of ***brand***, ***packaging\_size***, ***packaging\_type***, ***packaging\_size\_brand***, ***packaging\_type\_brand***, ***packaging\_type\_size***, ***phone\_model***, ***range*** and ***packaging\_size\_type\_brand***, which is enough for the system to describe an actual packaging entity.

### real phone

This table holds primary keys of ***phone\_model*** table, which is enough for the system to describe an actual phone entity.

### request bom

This table holds the primary keys of range, brand, phone\_model and finishing, along with other information like maximum date of delivery, maximum price and other info, needed to place request for a particular order.

### sample

This table holds primary keys of ***box***, ***companies***, ***brand***, ***set***, ***finishing***, ***packaging***, ***usage*** and ***bom*** tables, along with other information like maximum order quantity, received date and other info, needed by the system to describe a sample for a particular product.

### sets

### size type

This table indicates what type it is, viz. tablet, phablet, smartphone and others.

### sku

This table holds all the information needed by the system, like SKU number, EAN number and other info, to describe an SKU for a product.

### status

This table holds all the different kinds of status related to different development processes throughout the project. Its UID is used in many tables to indicate the status for that corresponding entry.

### sub family

This table holds the primary key of ***family*** table and other info like its name, needed by the system to describe a sub family for the product. Its UID is sent to many tables to indicate the sub family of the product.

### technology

This table holds the different types of technologies used in different process throughout the product development process. Its UID is sent to ***technology\_company*** and ***technology\_material*** to indicate the type of technology provided by the company, and material used in that technology, respectively.

### test

This table holds primary key of ***test\_type*** table and other information like the criteria values for corresponding test type, its name and other info needed by the system to describe different kind of tests used in the system.

### testing tool

This table holds names of the different type of testing tools used in the system to perform testing. Its UID is send to ***test\_testing\_tool*** table to indicate the testing tool used in that particular test.

### test type

This table holds all the criteria’s needed for testing. Its UID is sent to **test** table to indicate all the criteria’s whose value is in test table.

### town

This table holds names of all the towns, currently related to different entity addresses in the system like contact, companies and others.

### usage

This table holds all the different usage types used in the system. Its UID is sent to ***sample*** table, which indicates where the sample is or rather what the usage of that corresponding sample is.

### use of document

This table holds primary keys of *document* and *document\_type* table, which indicates the type of the corresponding document or rather what is the use of that corresponging document.

### xxx – status timing

This is the second table which is used by the system in a unique way. It holds primary keys ***xxx\_status*** and ***event\_type*** tables, where xxx\_status is the corresponding status table inserting a row in this table and event type is the event in which it is inserting.

## BOM – web pages

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.

All the pages are not explained here, but rather a structure of the project is given, till the date of September 1st, 2013 to exhibit the expanse and complexity of the project. Since this tool is not yet complete, as it is a big part of the final ERP tool, we were not able to include all the pages which were developed. Even while making the report, the pages were being developed and the project was not hindered in any way.

### calendar

### controllers

All the controller pages have a similar function of getting the information from the relevant view page and validating it. If found valid enough, insert it into its corresponding table.

func.php and functions2.php are specific files whose main function is to hold all the different methods which are used in JavaScript or direct call. They are the largest of files since they hold a lot of methods.

All file with header in them are related to the header of the web pages. They are still to be developed specifically for the project.

db.php holds the information needed to connect to the corresponding database.

validationPage.php is the page where the credential are validated, and if found that users are authorized to enter, they are rerouted to the dashboard pages assigned to their roles.

#### 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.

## Work flow in BOM tool

There are two different users for this tool viz. *admin* and *others.* The work flow for both the users differ only in one aspect, *admin* has certain extra rights than that of normal *other* user. In this tool, there are 5 different type of pages, which comprise to make around 50 different interactive web pages. The work flow is similar for all the pages in a particular type. Each main entity in the BOM process will have these 5 set of pages, and depending on the role of the user they will be accessible. But for now we haven’t developed all the pages, but a considerable amount of pages so as to make the tool accessible to users.

### New / Create Pages

For each main entity in the BOM process we have developed, a **New / Create** which enables the user to insert a new row entry for an entity, so there are total of 44 pages of this type. All the users who are validated have access to these pages. In the corresponding new / create pages, the user has to fill the form with all the essential information needed. On Submit of that form, the information is sent to the corresponding controller page, where the information is validated based on many criteria, and if the information is in proper format, it is inserted in the corresponding table. For many pages, there is a second page after this first insertion, so as to facilitate the insertion in the level 2 and more tables, like after the meeting is created the user can select the attendees and the reports presented in the meeting.

### Edit Pages

For now, only admin users have the access right to edit the information present in the database. There are total of 3 edit pages so far in the system and others are still to be developed. When the *admin* user clicks on **Edit** the user is forwarded to the corresponding page, where after selection of the entry to be edited, the user can see the existing values for the entry and can choose to enter a new value, which are included in a form. After Submit the information is sent to the corresponding controller pages, where the information is validated based on many different criteria, and if the information is in proper format, it is inserted in the corresponding table.

### List pages

All the users should have access to these pages. These pages list all the values, existing for that corresponding entity in the database. After clicking **List** for the corresponding entity, all the values present in the database are displayed on the web page. There are different interactive options to sort the table generated based on the different columns.

### Display pages

Only certain type of users should have access to these pages, the roles are still to be decided. When a user clicks on **Display** the user is forwarded to a page where all the relations of the other entities with this particular entity is displayed, along with all the other information this entity has. This is a very useful tool to have all the information related to a particular entity at a single glance. This page doesn’t have any user interaction, its sole purpose being just display.

### Problem Pages

Only certain type of users should have access to these pages, the roles are still to be decided. When a user clicks on **Problem** the user is forwarded to a page, where he / she can see directly, which all databases have problems with values, which all data is still to be validated, which of the data is to be confirmed and many other things. This page is supposed to user dependent. No 2 users should have same problem pages, unless specified so or they have same specific roles.

## Technologies Used

### PHP

**PHP** is a server – side scripting language designed for web development but also used as a general – purpose programming language. Originally created by Rasmus LERDORF in 1995, the reference implementation of PHP is now produced by The PHP Group. While PHP originally stood for *Personal Home Page*, it now stands for *PHP: Hypertext Preprocessor*, a recursive acronym.

PHP code is [interpreted](http://en.wikipedia.org/wiki/Interpreter_(computing)) by a web server with a PHP processor module, which generates the resulting web page: PHP commands can be embedded directly into an [HTML](http://en.wikipedia.org/wiki/HTML) source document rather than calling an external file to process data. It has also evolved to include a command – line interface capability and can be used in standalone graphical applications.

PHP is free software released under the PHP license, which is incompatible with the GNU General Public License (GPL) due to restrictions on the usage of the term *PHP*. PHP can be deployed on most web servers and also as a standalone shell on almost every operating system and platform, free of charge.

We chose PHP as the development language as it was meeting our requirements. We wanted to have a language which is flexible enough to evolve or even can be rewritten if need be. We didn’t choose JAVA or any other languages as they are too structured to suit our needs. And as PHP is now one of the best web development languages, we chose it as later in the future, the company may decide to actually make this tool accessible to any user who needs it.

### Eclipse Juno

In computer programming, **Eclipse** is a multi-language Integrated Development Environment (IDE) comprising a base workspace and an extensible plug – in system for customizing the environment. It is written mostly in JAVA. It can be used to develop applications in Java and, by means of various plug-ins, other programming languages including ADA, C, C++, COBOL, Fortan, Haskell, JavaScript, Perl, PHP, Python, R, Ruby (including Ruby on Rails framework), Scala, Clojure, Groovy, Scheme, and Erlang. It can also be used to develop packages for the software Mathematica. Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++ and Eclipse PDT for PHP, among others.

The initial codebase originated from IBM VisualAge. The Eclipse software development kit (SDK), which includes the Java development tools, is meant for Java developers. Users can extend its abilities by installing plug-ins written for the Eclipse Platform, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules.

### MySQL

**MySQL** ( "My S-Q-L",officially, but also called  "My Sequel") is (as of July 2013) the world's most widely used open-source relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases, though SQLite probably has more total embedded deployments. It is named after co-founder Michael Widenius’s daughter, My. The SQL phrase stands for Structured Query Language.

MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack (and other ‘AMP’ stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Free – software – open source projects that require a full-featured database management system often use MySQL.

### JavaScript

**JavaScript** (**JS**) is an interpreted computer programming language. As part of web browsers, implementations allow client side scripts to [interact with the user](http://en.wikipedia.org/wiki/User_interface), control the browser, communicate asynchronously, and alter the [document content](http://en.wikipedia.org/wiki/Document_Object_Model) that is displayed. It has also become common in server-side programming, game development and the creation of desktop applications.

JavaScript is a prototype – based scripting language with dynamic typing and has first – class functions. Its syntax was influenced by C. JavaScript copies many names and naming conventions from JAVA, but the two languages are otherwise unrelated and have very different semantics. The key design principles within JavaScript are taken from the Self and Scheme Learning Language. It is a multi – paradigm language, supporting object – oriented, imperative, and functional programming styles.

The application of JavaScript to uses outside of web pages—for example, in [PDF](http://en.wikipedia.org/wiki/Portable_Document_Format) documents, site – specific browsers, and desktop widgets — is also significant. Newer and faster JavaScript VMs and frameworks built upon them (notably Node,js) have also increased the popularity of JavaScript for server-side web applications.

### jQuery

**jQuery** is a multi-browser (cf. cross - browser) JavaScript library designed to simplify the client – side scripting of HTML. It was released in January 2006 at BarCamp NYC by John Resig. It is currently developed by a team of developers led by Dave Methvin. Used by over 65% of the 10,000 most visited websites, jQuery is the most popular JavaScript library in use today.

jQuery is [free,](http://en.wikipedia.org/wiki/Free_and_open_source_software) open source software, licensed under the MIT license. jQuery's syntax is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop AJAX applications. jQuery also provides capabilities for developers to create plug - ins on top of the JavaScript library. This enables developers to create [abstractions](http://en.wikipedia.org/wiki/Abstraction_(computer_science)) for low-level interaction and animation, advanced effects and high-level, theme-able widgets. The modular approach to the jQuery library allows the creation of powerful dynamic web pages and web applications.

The set of [jQuery](http://en.wikipedia.org/wiki/JQuery" \l "Features) core features — DOM element selections, traversal and manipulation —, enabled by its *selector engine* (named "Sizzle" from v1.3), created a new "programming style", fusing algorithms and DOM – data - structures; and influenced the architecture of other JavaScript frameworks like YUI v3 and Dojo.

### AJAX

**Ajax** (also AJAX; an acronym for asynchronous JavaScript and XML) is a group of interrelated web development techniques used on the client – side to create asynchronous web - applications. With Ajax, web applications can send data to, and retrieve data from, a server asynchronously (in the background) without interfering with the display and behavior of the existing page. Data can be retrieved using the XMLHttpRequest Object. Despite the name, the use of XML is not required (JSON is often used instead), and the requests do not need to be asynchronous.

Ajax is not a single technology, but a group of technologies. HTML and CSS can be used in combination to mark up and style information. The DOM is accessed with JavaScript to dynamically display, and allow the user to interact with, the information presented. JavaScript and the XMLHttpRequest object provide a method for exchanging data asynchronously between browser and server to avoid full page reloads.

### HTML & CSS

**HyperText Markup Language** (**HTML**) is the main markup language for creating web pages and other information that can be displayed in a web browser.

HTML is written in the form of HTML elements consisting of *tags* enclosed in [angle brackets](http://en.wikipedia.org/wiki/Angle_brackets) (like <html>), within the web page content. HTML tags most commonly come in pairs like <h1> and </h1>, although some tags represent *empty elements* and so are unpaired, for example <img>. The first tag in a pair is the *start tag*, and the second tag is the *end tag* (they are also called *opening tags* and *closing tags*). In between these tags web designers can add text, further tags, [comments](http://en.wikipedia.org/wiki/Comment_(computer_programming)) and other types of text-based content.

The purpose of a [web browser](http://en.wikipedia.org/wiki/Web_browser) is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page.

HTML elements form the building blocks of all [websites](http://en.wikipedia.org/wiki/Website). HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, [links](http://en.wikipedia.org/wiki/Hyperlink), quotes and other items. It can embed scripts written in languages such as JavaScript which affect the behavior of HTML web pages.

Web browsers can also refer to Cascading Style Sheets (CSS) to define the appearance and layout of text and other material. The [W3C](http://en.wikipedia.org/wiki/W3C), maintainer of both the HTML and the CSS standards, encourages the use of CSS over explicit presentational HTML markup.

### ERP

**Enterprise resource planning** (**ERP**) is a cross-functional enterprise system driven by an integrated suite of software modules that supports the basic internal business processes of a company. ERP gives a company an integrated real-time view of its core business processes such as production, order processing, and inventory management, tied together by ERP applications software and a common database maintained by a database management systems. ERP systems track business resources (such as cash, raw materials, and production capacity) and the status of commitments made by the business (such as customer orders, purchase orders, and employee payroll), no matter which department (manufacturing, purchasing, sales, accounting, and so on) has entered the data into the system. ERP facilitates information flow between all business functions inside the organization, and manages connections to outside stakeholders.

Enterprise system software is a multi-billion dollar industry that produces components that support a variety of business functions. IT investments have become the largest category of capital expenditure in United States-based businesses over the past decade. Enterprise systems are complex software packages that offer the potential of integrating data and processes across functions in an enterprise. Although the initial ERP systems focused on large enterprises, there has been a shift towards smaller enterprises also using ERP systems.

Organizations consider the ERP system a vital organizational tool because it integrates varied organizational systems and enables flawless transactions and production. However, an ERP system is radically different from traditional systems development. ERP systems can run on a variety of computer hardware and network configurations, typically employing a database as a repository for information.

## Security Techniques Implemented

### SQL Injection

**SQL injection** is a code injection technique, used to [attack](http://en.wikipedia.org/wiki/Attack_(computing)) data driven applications, in which malicious SQL statements are inserted into an entry field for execution (e.g. to dump the database contents to the attacker). SQL injection must exploit a [security vulnerability](http://en.wikipedia.org/wiki/Security_vulnerability) in an application's software, for example, when user input is either incorrectly filtered for string literal [escape](http://en.wikipedia.org/wiki/Escape_sequence) characters embedded in SQL statements or user input is not strongly typed and unexpectedly executed. SQL injection is mostly known as an attack vector for websites but can be used to attack any type of SQL database.

In a 2012 study, security company Imperva observed that the average web application received 4 attack campaigns per month, and retailers received 2 times as many attacks as other industries.

### Session Management

In computer science, in particular networking, a **session** is a semi-permanent interactive information interchange, also known as a dialogue, a conversation or a meeting, between two or more communicating devices, or between a computer and user. A session is set up or established at a certain point in time, and torn down at a later point in time. An established communication session may involve more than one message in each direction. A session is typically, but not always, stateful, meaning that at least one of the communicating parts needs to save information about the session history in order to be able to communicate, as opposed to stateless communication, where the communication consists of independent requests with responses.

An established session is the basic requirement to perform a connection – oriented communication. A session also is the basic step to transmit in connectionless communication modes. However any unidirectional transmission does not define a session.

### Authorization & Access Control

In the fields of physical security and information security, **access control** is the selective restriction of access to a place or other resource. The act of *accessing* may mean consuming, entering, or using. Permission to access a resource is called **authorization**.

Locks and login credentials are two analogous mechanisms of access control.

## Possible Technical Choices

Other frameworks like Symfony or Zend could have been used for the development of the project. But I personally suggested that I would like to use Eclipse since I was used to using it before for other development languages. The company didn’t have any problem with that.

As for the language, we chose PHP for its flexibility over JAVA and other object oriented languages, as it was the requirement of the project. Also one of the reasons were that the tool was going to be a web application and what better than using one of the best languages for web development.

# First Review

This chapter will focus mainly on how the tool which is developed will help the growth of company. How well the requirements of the company are met and what was my added to the company and myself as well.

As told earlier, in the start the company had this great idea, to develop an ERP tool to manage all the different aspects of the product development process which the company has. They were working on excel sheets before this. They came to realize that excel sheets were not a good option, because of the restricted functionality of it, even after having macros and many other advancements in it over time. The main restriction was the size of the excel sheets, which was getting less and less as time passed.

The motive of this internship was to give a start to this “*idea*”. They wanted to have the main 2 modules of this process viz. forecast and BOM, to be developed as a part of the mission for this internship. With proper guidance from my tutors in the company, M. Olivier JEUSSET, M. Benjamin LEMOINE and M. Piotr KOROSTELEV, we were able to develop such a tool.

The output what we had was a glimpse of the ERP tool that they thought of at the start. It still needs a lot of things to be included in the tool, but what we have developed so far can surely provide a good base for further advancements of the tool.

Many more things can be included to this tool, as the product development process is very extensive. The “Look and Feel” of the tool can be improved a great deal. Even the source code for the web pages can be optimized for maintenance. There is always scope for improvement, even if the project was complete and we had an ERP tool.

## Added - Value to the company

This internship hold a critical value towards the completion of this tool they had in their minds. And eventually this gives a great added – value towards the company’s overall growth.

With the development of this tool they will be able to handle the different prospects of the product development phase at ease. With everything in a database they won’t have to worry about the maintenance of the data they had to handle and maintain manually earlier. And with the help of web pages it would be very easy and efficient to enter the data in this database which was rather complicated earlier when they were working in an excel sheet.

The greatest advantage I would say is the availability of the tool. Since it’s an online tool, it would be accessible by anyone anywhere, unlike prior to using this tool. Earlier they use to share the related excel sheet with all the people working on it, and use to merge it at the end of the day. This was quite cumbersome and difficult to maintain since they had to look out for many things.

The tool which we have developed so far will be useful in short term as well as in the long run. Before the end of the internship the idea is to implement what is developed and let people use it. If it does increase the efficiency of the work, the further development of the work would be pursued. It is likely that once these tools are ready for use they would increase the efficiency, and would result in the development of the final product. And once the final product is ready to use, it will surely increase the overall growth of the company.

## Added – Value to Myself

Personally, I got to learn a lot of new things from this internship. As this was my first time to work in a French company as well as a Chinese environment in which I worked for 2 months. It was very good to know these cultures and work environments. Especially it was a new experience for me to work in the French environment, as it is very different than what I experienced so far. It is too difficult to just express in words.

As for my technical improvements, it gave a great added – value to my technical skill set. Before working as an intern in this company, I had an experience of developing small pieces of codes for the e-Learning company I was working for. But there I could only develop my logical skills as the tool we were using was an integrated tool with few different languages. But working here, I learned a lot of things about the technologies I was using viz. PHP + MySQL, Javascript, jQuery and AJAX.

Before working here, I didn’t have any professional experience in any of the languages I worked on other than SQL. But after working here I can say I have learned at least enough that I can confidently add it to my skill set and develop in these technologies if given an opportunity in the future.

Also working in an international company gave me a great advantage and exposure for my future prospect for having a good job in a good company. Also it will count as an added – value to the experience I would need to pursue my PhD in near future.

## Conclusion

For concluding the report I would, first of all, like to thank you for giving your time and patience to read this document. I hope it was clear enough to explain the context of my internship and the project I was working on.

I would have very much appreciated if I was provided with a set of drafted documentations for the development of the tools. But it was quite understandable, since it was an idea which they wanted to test before going forward with it. This internship was the first step towards it. If they feel it gave the output what they expected I would be glad that I was able to give such an added – value to the company.

I would be completing my internship on October 2nd, 2013 and I would like to say that I felt very good working in such a flamboyant and professional environment. I would have liked to work further in the company, if given an opportunity.

Being said everything I would like to conclude my document on this note. I hope it was clear enough and if you have any doubts or questions you can get back to me anytime. I would be more than glad to clarify them for you.

# Bibliography

For the documentation, mainly for the basic information on technologies and techniques used, was taken from Wikipedia, the free encyclopedia.

[*http://en.wikipedia.org/wiki*](http://en.wikipedia.org/wiki/Eclipse_(software))

For the development of the source code,

* the PHP manual: [*http://www.php.net/manual/en/manual.php*](http://www.php.net/manual/en/manual.php)
* the Stack Overflow forum: [*http://stackoverflow.com/*](http://stackoverflow.com/)

For company information, documentations like presentations and reports, present in the company were used along with the reference of the company’s website,

[*http://www.ascendeo.fr/open/store/control/infoSociete*](http://www.ascendeo.fr/open/store/control/infoSociete)

# Abbreviations

**AJAX:** *Asynchronous Javascript and XML*

**BOM:** *Bill of Material(s)*

**CSS:** *Cascading style sheets*

**ERP:** *Enterprise Resource Planning*

**HTML:** *Hypertext Markup Language*

**IDE:** *Integrated Development Environment*

**MVC:** *Model – View – Controller*

**PHP:** *Hypertext Preprocessor*

**P O P I:** *Product Order | Product Invoice*

**SQL:** *Standard Query Language*

**TTM / TTL:** *Time To Market / Time To Launch*

# Notes

## Gantt Chart

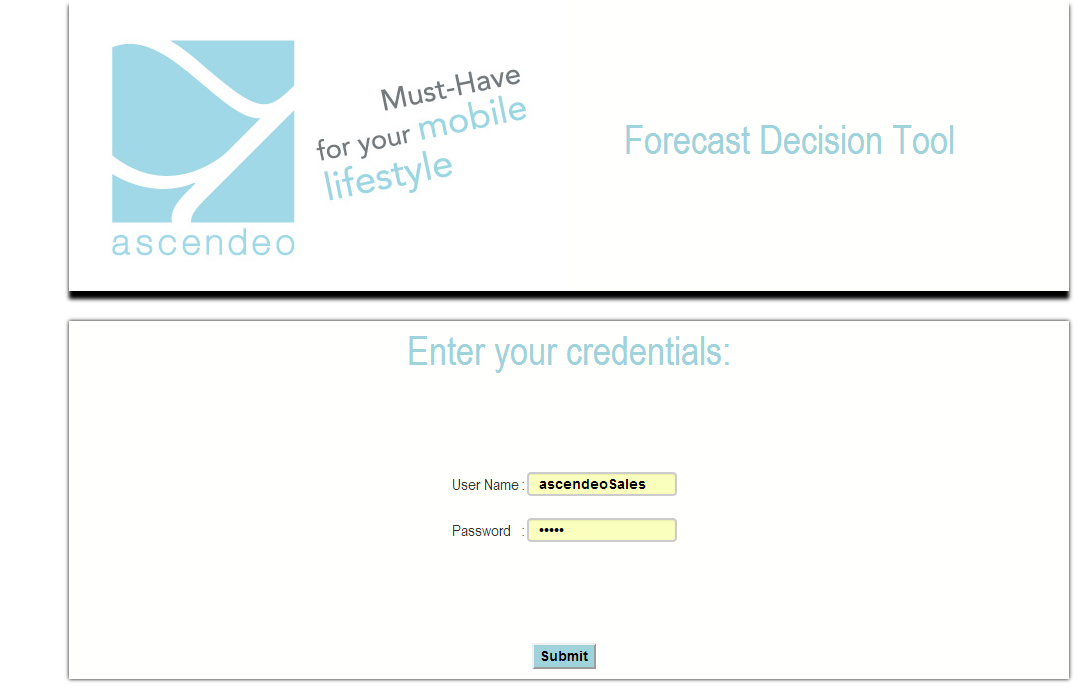
The link to the Gantt chart for the complete internship is at

<https://app.smartsheet.com/b/home>.

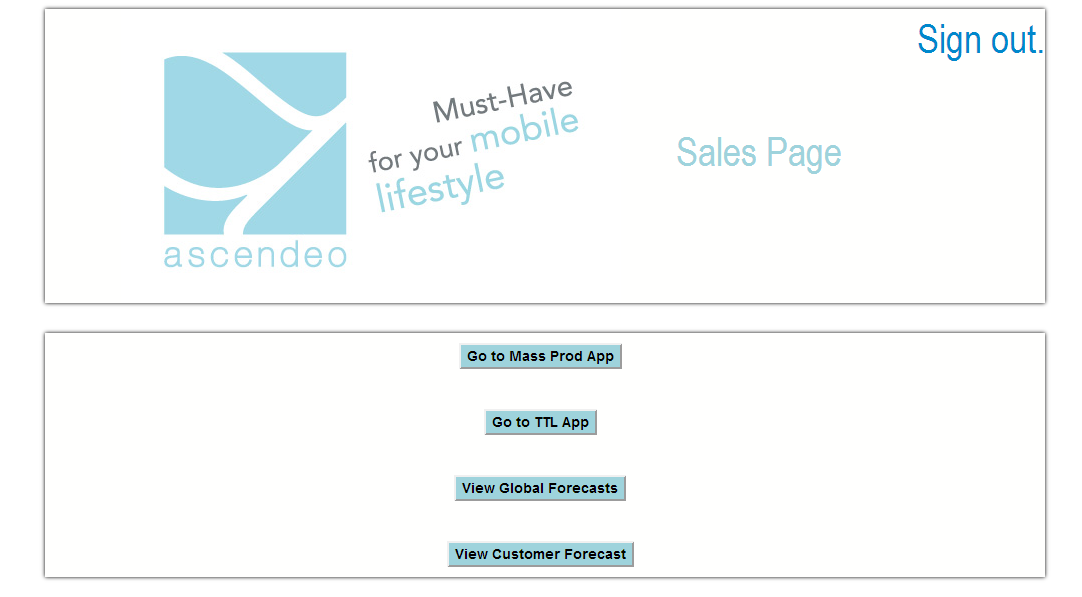
A link is given as it is too big to be displayed.

## Screen Shots

### Forecast Login Page



### Forecast Sales Page

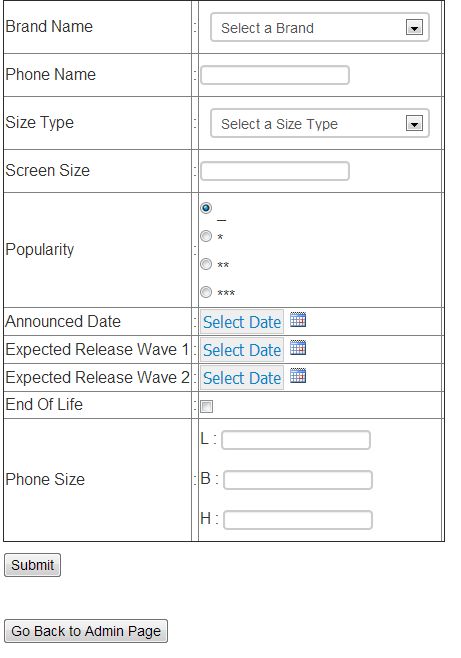


### Mass Prod App Page

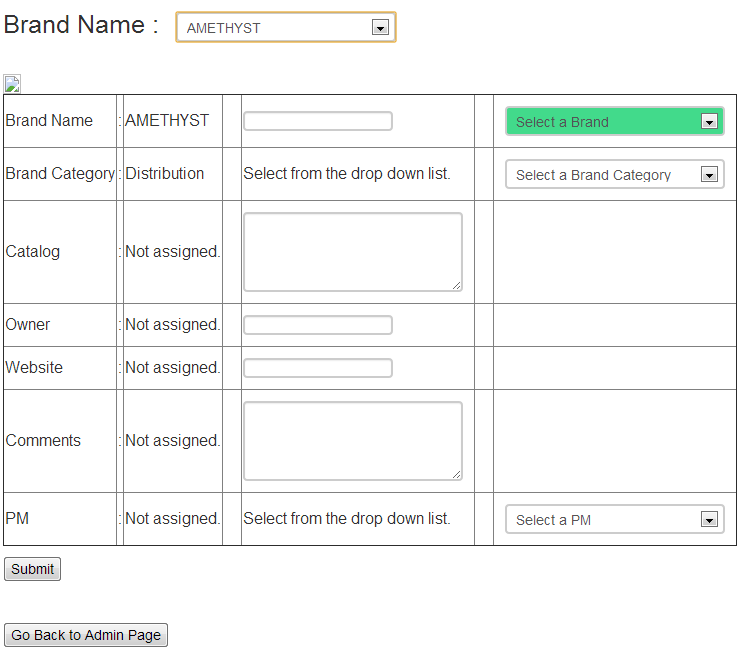
### Forecast Admin Page

### BOM Admin Page

### New Phone Page



### Edit Brand Page



### Display Range Page