

**Department of Computer Science  
University of Leicester**

**CO3014: Computer Science Project**

**Project Plan – February 2012**

**Simplified Enterprise Resource Planning System**

**Miguel Cruz Zúnica  
(Student ID: 119038672)**

# Table of contents

<b>1. Introduction</b>	
1.1 Research Context	pg. 1
1.2 Aims	pg. 1
1.3 Objectives	pg. 1
<b>2. Requirements and specifications</b>	
2.1 Data	pg. 1
2.2 Settings	pg. 2
2.3 Sales	pg. 2
2.4 Purchases	pg. 3
<b>3. Design</b>	
3.1 System design	pg. 3
3.2 GUI design	pg. 3
<b>4. Implementation Details</b>	
4.1 Programming language	pg. 4
4.2 Development methodology	pg. 4
4.3 Limitations	pg. 4
4.4 Performance	pg. 4
4.5 Installation requirements	pg. 5
<b>5. Software Testing</b>	pg. 5
<b>6. Planning and timescales</b>	pg. 5
<b>7. References</b>	pg. 6

# 1. Introduction

## 1.1 Research Context

An enterprise resource planning [1] (ERP from now on) is a system designed to manage information from a business or an entire organization. It integrates both external and internal information across departments. From sales, accounting and manufacturing to customer relationship, its goal is to automate and enhance control over business' information. This is a key factor for success in today's high-tech world where competition between companies is fierce. The problem is that often this kind of software is complex and expensive, specially for smaller businesses.

## 1.2 Aims

The aim of the project is to build a simplified ERP (SERP from now on) that will cover the basic features required by businesses in a simple and generic way that will allow even small companies to use it out of the box without big investments.

## 1.3 Objectives

The objective of the project is to create a modular system with starting features including: sales and purchasing, user management, client profiling, product specification and accounting. Due to its modular nature it will be possible to extend it with additional functionalities in the future. For the same reason the look and feel must be consistent among all modules. To accomplish this, both a client and a server will be developed. The server will authenticate users login credentials and provide information from database when requested from client. The client will prompt a login screen, after which it will present the functionalities available to the user depending on his access group.

# 2. Requirements and specifications

The system functionalities will be organized in four modules. All of them, except configuration one, will have in common that you will be able to search through them, filtering search results by several parameters and loading desired result to be displayed.

## 2.1 Data

This module will be dedicated to management of data objects that will be used in other modules, for example:

- Clients  
Here you will be able to view, create, delete or modify instances of clients. In addition you will be able to fill information about them like their name, address, postal code, country, telephone number, e-mail address, company name, id code, website and billing information.

- Products  
Here you will be able to view, create, delete or modify instances of your products. In addition, you will be able to fill information about them like their name, family, model, base price, selling price and the amount you have on stock.
- Employees  
Here you will be able to view, create, delete or modify instances of your employees. In addition, you will be able to fill information about them like their name, surname, address, telephone number, serp username, e-mail address, billing information, and cost per hour.

## 2.2 Settings

This module will be dedicated to serp settings:

- Configuration  
Here you will be able to fill various parameters required by the system like your company information, tax rates and others.
- Users  
Here you will be able to view, create and modify serp users. In addition, users will have an access group that will restrict the functionalities they can access.

## 2.3 Sales

This module will be dedicated to sales:

- Budgets  
Here you will be able to view, create or modify budgets. Furthermore, all the information related with the client and the products will be obtained from their respective profiles. Also, there will be a button to create a sale out of current budget and you won't be able to delete or modify a budget if there is a sale made out of it.
- Sales  
Here you will be able to view, create or modify sales. Furthermore, all the information related with the client and the products will be obtained from their respective profiles or budgets. Also, there will be a button to create a recovery for the current sale and the stock of the products being sold will be updated on creation of the sale. In addition, you won't be able to delete a sale if there is a recovery related with it and there will be an indicator of the amount that has already been paid.
- Recoveries  
Here you will be able to view, create or modify recoveries. Furthermore, all the information related with the client and the sales will be obtained from their respective profiles or sales. In addition, you will only be able to create a recovery for a sale if it isn't fully paid already.

## 2.4 Purchases

This module will be dedicated to purchases:

- Purchases  
Here you will be able to view, create or modify purchases. Furthermore, all the information related with the client and the products will be obtained from their respective profiles. Also, there will be a button to create payments for the current purchase and there will be an indicator of the amount that has already been paid. In addition, the stock of the products purchased will be updated when the purchase is created and you won't be able to delete a purchase if there is a payment linked to it.
- Payments  
Here you will be able to view, create or modify purchases. Furthermore, all the information related with the client and the products will be obtained from their respective profiles or purchases. In addition, you will only be able to create a payment if there is money left to be paid in the related purchase.

## 3. Design

### 3.1 System design

To design the system UML[2] diagrams and functionality specifications will be used, as a result, implementation will be faster and easier. In addition, there will be less chances of committing mistakes and having to rewrite code once in implementation phrase.

In addition, the system will be divided in three layers: presentation, logic and storage. The presentation layer in the server will embrace minimum aspects like turning on and off the server. In case of the client it will embrace the full GUI with all the features available. The logic of the server will be related with the database functionalities, while in the client it will cover all the functionalities offered by the system. The storage layer will be implemented in the form of a database on the server and a configuration file on the client. Client's configuration file will be used to save previous login credentials or server IP to avoid having to repeat it each time.

Regarding database, the first choice will be Java Derby[3] (javadb) because its integrated with java platform and that minimizes the requirements of the system. Otherwise free databases as mysql, sqlite or postgresql would be considered.

### 3.2 GUI design

Graphical design of the system will play an important role in the development process. Since the system is going to be used by non technical users it is important that graphical user interfaces are as simple and minimalistic as possible. Furthermore, as the system is going to be used in a business environment, the main theme should be based on plain and soft colors and the font should be a formal one. In addition, any eventuality should be informed in a clear way to the user by the means of pop-up messages that unambiguously inform of what happened in the system.

In the technical side Java offers two official components in order to develop graphic interfaces: Abstract Window Toolkit (AWT) and Swing[4]. In addition to these there is an alternative called Standard Widget Toolkit[5] (SWT) developed by IBM and maintained by the Eclipse Foundation[6]. Serp will use Swing for a number of reasons:

- In contrast with SWT, Swing is part of java library, so it doesn't need additional native libraries.
- Works in all major platforms.
- Has an integrated GUI editor both in Eclipse and Netbeans[7].
- Better support and documentation from sun than SWT.
- Includes all AWT features as Swing is a newer and more sophisticated version of it.

## **4. Implementation Details**

### **4.1 Programming language**

The programming language chosen for this project is Java[8]. This is because it allows object- oriented programming, its robust and secure, provides high performance and its portable across different architectures. This last point is specially important because the current trend is towards diversity of operating systems, both on portable devices and pc. Hence building the system with java allows it to run on almost all platforms available, which can target niches of potential customers that use less common operating systems. Furthermore, Eclipse will be used as a integrated development environment for its simplicity, while Netbeans would be the second choice if necessary.

### **4.2 Development methodology**

The development methodology chosen for the project will be a custom one based on the principle of agile development[9], and more specifically on iterative development. Therefore, on every iteration a small functionality will be implemented and tested before to move on to the next one. This will hopefully prevent code rewriting and will help building a more robust system while minimizing the time spent on debugging.

### **4.3 Limitations**

Although the first version of the system won't have any limitation on the number of concurrent users connected to the system, a future market version would implement a limit and charge for any extra user slot.

### **4.4 Performance**

The system should consume a reasonable amount of memory under 1 GB, it should also leave at least 50% of cpu processing available and transitions should run smoothly without any sensible wait time. Processes, specially searches are an exception and should be used with caution by the user.

## 4.5 Installation requirements

The system will require java runtime environment to be installed on the system to work. Also, if the machine running the server is required to be accessible from the internet, it may need specific configuration to route extern ports used by the server to the specific machine where the server is located.

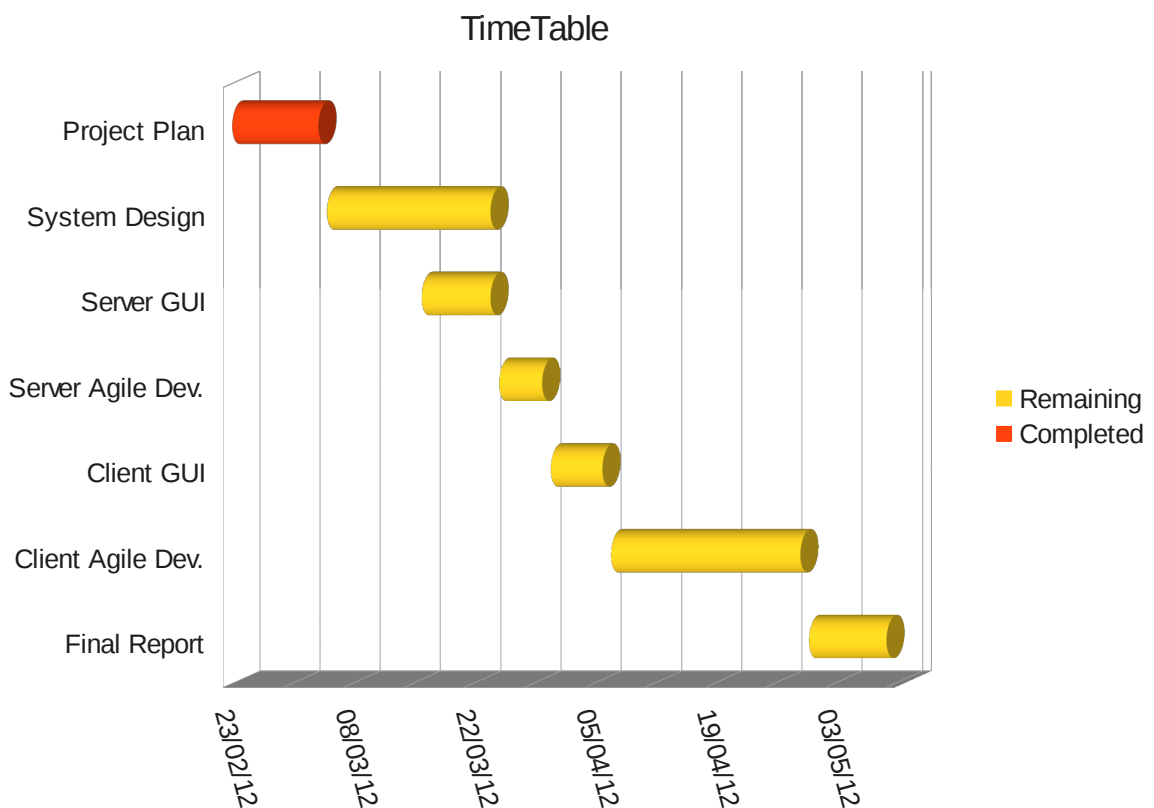
## 5. Software Testing

Software tests will be performed with the Junit[10] platform after every development iteration to test the specific feature implemented in the iteration. In addition, complete system tests will be performed periodically and after certain milestones to ensure the whole system remains robust and reliable.

## 6. Planning and timescales

The implementation phrase, which will be composed of iterations of designing, coding and testing sessions is expected to require the most part of the remaining time. It is also difficult to predict the exact amount of time as the very nature of this phrase makes it unpredictable. In the rare case that everything went abnormally smooth, additional features could be implemented as the modular design of the system permits to extend it with ease.

The last weeks of the remaining time will be spent on finishing the final report.



## 7. References

- [1]. Shields, Murrell G., E-Business and ERP: Rapid Implementation and Project Planning. (2001)
- [2]. Fowler, Martin. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. (2003)
- [3]. <http://docs.oracle.com/javadb/>
- [4]. James Elliott, Robert Eckstein, Marc Loy, David Wood, Brian Cole: *Java Swing*, O'Reilly. (2002).
- [5]. Northover, Steve. *SWT: Implementation Strategy for Java Natives*. (2001)
- [6]. [www.eclipse.org](http://www.eclipse.org)
- [7]. [www.netbeans.org](http://www.netbeans.org)
- [8]. Schildt, Herbert. *Java The Complete Reference*, 8th Edition (2011)
- [9]. Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. *Agile Software Development Methods: Review and Analysis*. (2002)
- [10]. J. B. Rainsberger, Scott Stirling, *JUnit Recipes: Practical Methods for Programmer Testing*. (2004)