

# ENSC 494 Special Project Laboratory Proposal

Supervisor: Dr. Craig Scratchley

Mehdi Abdollahi

Revision 7

January 2006

## **Acknowledgments:**

Contribution of Dr. Craig Scratchley in preparing this document is gratefully acknowledged and appreciated.

## Revision History:

Version	Date	Name	Description
1.0	September 14, 2005	Mehdi A.	Template document was created
2.0	September 21, 2005	Mehdi A.	First draft was done. Added Introduction.
3.0	October 22, 2005	Mehdi A.	Modified the topic of the proposal
4.0	November 10, 2005	Mehdi A.	Added Project milestones
5.0	November 18, 2005	Mehdi A.	Added more description in the introduction
6.0	November 22, 2005	Mehdi A.	Added Preliminary Issues, Development Environment Overview, Software Overview
7.0	November 24, 2005	Mehdi A.	Updated definitions for Requirements Specification Document and Architecture Design Document.

## Table of Contents:

Acknowledgments: .....	- 2 -
Revision History: .....	- 3 -
Table of Contents: .....	- 4 -
Acronyms .....	- 5 -
Introduction: .....	- 6 -
Objectives: .....	- 7 -
Future Plans: .....	- 8 -
Project Milestones: .....	- 8 -
Requirement Specification Document .....	- 8 -
Architectural Design Document .....	- 9 -
Project Deliverables: .....	- 9 -

## **Acronyms**

### **GUI**

Graphical User Interface

### **HTML**

Hypertext Markup Language

### **WWW**

World Wide Web

### **CRM**

Customer Relationship Management

### **Three-tier**

In computing, Three-tier is a client-server architecture in which the user interface, functional process logic ("business rules"), data storage and data access are developed and maintained as independent modules, most often on separate platforms.<sup>1</sup>

### **SQL**

Structure Query Language

---

<sup>1</sup> Wikipedia Web Site. [http://en.wikipedia.org/wiki/Client-server\\_architecture](http://en.wikipedia.org/wiki/Client-server_architecture)

## Introduction:

Since 2001, the construction industry has shown an amazing growth. Many companies in this sector of the business have had the resources to grow and many new companies have formed as a result of this lucrative market. With interest rates predicted to stay low and the prospect of the 2010 Whistler Olympics, it is very plausible to predict that the current climate in the construction industry will resume for a while. For companies in this market, growth will require better customer relationship management as well as efficient project management techniques and tools.

We, at eRational Solutions, have realized a market for a good software tool that can help a small or medium size construction company with its customer relationship management as well as the management of each project. This software can have the potential to include many other types of modules such as inventory and accounting tools which can be essential in any business. At this stage Cool Project is the name that we have selected for this software.

We think what will distinguish this software from other software is the fact that it is customized for construction industry. Another attractive aspect of this software is its accessibility from any location. Since it is going to be web-based application with an html interface, users can access their account from job location via a mobile phone or a wireless internet access. Figure 1 shows the multiple layers of the design of the software.

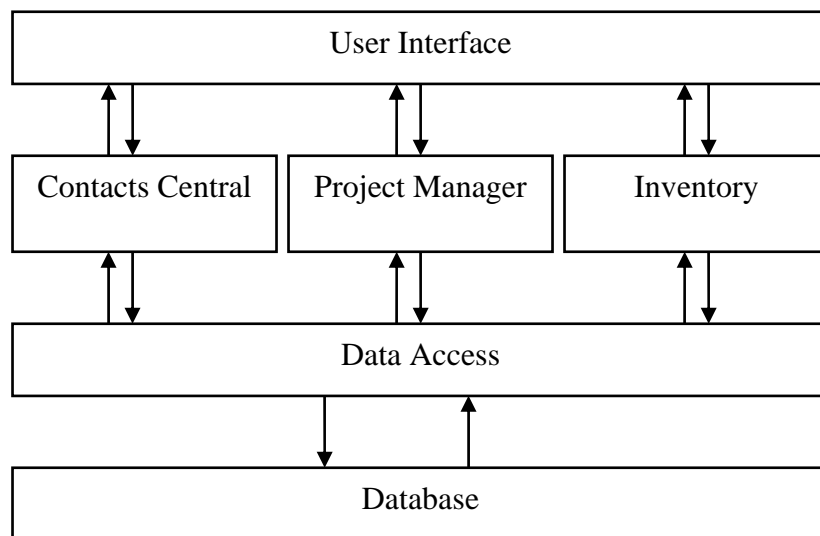


Figure 1 - Layers of the overall software

## **Objectives:**

This proposal is for ENSC 494, Special Project Laboratory, for the Spring of 2006. I would like to use this course as an Engineering elective towards my graduation in April of 2006. The goal of the project is to design and implement Contacts Central, a web-based Customer Relationship Management (CRM) tool which is one of the modules in CoolProject.

The main purpose for doing this project is to design an enterprise scale database as well as to gain experience in software usability and graphical user interface (GUI) design. This project can be divided into three phases. The first part will be the research needed in order to find a set of guidelines on GUI design as well as designing the database that will be the backbone of this software. Phase one of the project also includes completing the Requirement Specifications Document and Architectural Design Document. The second phase of this project consists of the development of the database, data access layer and the business rules and completion of GUI. The final phase of the project includes testing, debugging and demonstration of the project.

## **Preliminary Issues:**

One of the main issues with any CRM software is accessibility. This means a user's ability to access the software even if they are out of office. This especially is very important for a user in the construction industry since many of the projects are outdoors. This presents a good opportunity for software that is accessible through the World Wide Web (WWW) interface.

Another desirable feature in a CRM is the ability to integrate with other applications that a business uses to run the operations such as accounting and inventory. Imagine CRM software that can talk to the software that is supposed to help a user with the payroll at the end of the month by sending necessary information to do the payroll; or a project management tool that updates the inventory as each project goes along.

## **Development Environment Overview:**

This software will be developed using the Microsoft Visual Studio<sup>®</sup> environment and will be based on the .NET<sup>®</sup> platform. The database server will be Microsoft SQL Server<sup>®</sup>.

Microsoft Internet Information Services<sup>®</sup> (IIS) with .NET<sup>®</sup> 1.1 will be the web server used for this project. The development of this software will closely follow the Three-tier architecture design.

## **Software Overview:**

Contacts Central will be a great asset to businesses that want to keep track of their customers and potential clients as well as their employees in a centralized format which can be accessed freely from different location. A Contacts Central user will be able to

add, edit and delete contacts. Each contact has many pre-defined fields such as “name” and “address”. As well as these pre-defined fields, the user can create custom fields that fit their business best such as “salary” or “first date contacted”. User can create categories which can be applied to each contact when they are added. To give an example, a user can have a category named “Globe and Mail leads” which can be applied to contacts who became clients through an advertisement that was put in the Globe and Mail newspaper. This feature gives the user an extremely useful tool to organize their contact list.

A very convenient tool that this software offers is a powerful search engine. It lets the user search the list of contacts for values in any of the pre-defined or custom fields. The user can also search the list of contacts based on the category that each contact belongs to.

### **Future Plans:**

They capabilities of this software can be used in many ways. For example, users in the future should be able to send emails or direct mail through this software. This can make sending a “Thank You” letter to customers who have been with the firm for more than a year as simple as running a search based on a custom field such as “first date contacted” and then click a few buttons, type the letter and hit send.

### **Project Milestones:**

Following table lists the possible milestones that have been designated for this project.

November 25, 2005	Final version of the project proposal due date
January 20, 2006	Completion of the Requirement Specification Document
February 15, 2006	Completion of Architectural Design Document
March 1, 2006	Alpha version presentation
March 20, 2006	Beta version presentation
April 10, 2006	Final version presentation

### **Requirement Specification Document**

Requirement Specification Document simply lists the functional and nonfunctional requirements. The most obvious purpose of this document is to provide the information needed in order to define the scope of the system. This document describes to the analysts exactly what the system needs to end up doing, and when discrepancies arise; the document serves as the place to go for clarification. The following lists a few major components in a typical requirement specification:

- Data Flow Diagram
- Use Case Scenarios
- Entity Relationship Diagram (ERD)



## Architectural Design Document

The main goal of doing an architecture design is to determine what parts of the application software will be assigned to what hardware. All software systems can be divided into four basic functions. The first is data storage. Most information systems require data to be stored and retrieved, whether it is a small file, such as a memo produced by a word processor, or a large database, such as one that stored an organization's accounting records. These are the data entities documented in ERDs. The second function is the data access logic, the processing required to access data, which often means database queries in Structured Query Language (SQL). The third function is the application logic, which is the logic that is documented in the DFDs, use cases, and functional requirements. The fourth function is the presentation logic, the display of information to the user and the acceptance of the user's command (the user interface). The following are part of an architectural design document.

- Object Communication Diagram (OCD)
- Detailed Interface Design

## Textbook:

The following textbook will be used as a guide for the design and development of this project.

- Dennis, A. and Wixom, B. H. 2003. System Analysis Design. 2<sup>nd</sup> Ed. John Wiley & Sons Inc.

## Project Deliverables:

The breakdown of marks for this special project course will be as follows:

- |   |             |
|---|-------------|
| • Requirement Specifications Document         | <b>15 %</b> |
| • Architectural Design Document               | <b>15 %</b> |
| • Help Documentation                          | <b>10 %</b> |
| • Demonstration of software                   | <b>40 %</b> |
| ○ Alpha (5%)                                  |             |
| ○ Beta (10%)                                  |             |
| ○ Final (25%)                                 |             |
| • General good software engineering practices | <b>20 %</b> |
| ○ Use of source control                       |             |
| ○ Attention to coding details                 |             |
| ○ Meeting participation and punctuality       |             |