Contacts Central®

Requirements Specification

*e*Rational Solutions Inc.
1145 Harold Rd.
North Vancouver, B.C.
V7K 1G3

Revision 12
April 18, 2006

Prepared by:

Mehdi Abdollahi ……………………..
Project Manager

Accepted by:

Craig Scratchley ……………………..
Project Supervisor

Revision history:

Revision 0	Dec 20, 2005	Original template generated by Mehdi Abdollahi.
Revision 1	Dec 23, 2005	Modified by: Mehdi Abdollahi Executive Summary and Introduction was added to the document.
Revision 2	Dec 24, 2005	Modified by: Mehdi Abdollahi First draft of the Object Relation Diagram (ORD) was added.
Revision 3	Jan 3, 2006	Modified by: Mehdi Abdollahi Completed the Environment section (2.0).
Revision 5	Jan 10, 2006	Modified by: Mehdi Abdollahi Finished ERD and ERD discussion.
Revision 6	Jan 12, 2006	Modified by: Mehdi Abdollahi Completed Functional description (section 4).
Revision 7	Jan 15, 2006	Modified by: Mehdi Abdollahi Complete Use Case Scenarios (section 5)
Revision 8	Jan 20, 2006	Modified by: Mehdi Abdollahi Completed Sections 6,7,8 and Appendix A
Revision 9	Jan 26, 2006	Modified by: Mehdi Abdollahi Corrected page number formatting
Revision 10	Jan 29, 006	Modified by: Mehdi Abdollahi Updated ERD
Revision 11	April 9, 2006	Modified by: Mehdi Abdollahi Updated ERD
Revision 12	April 17, 2006	Modified by: Mehdi Abdollahi Updated Client computer requirements.

Executive Summary

Contacts Central® is a web-based customized Customer Relationship Management (CRM) tool for Pacific Construction Inc. designed and being developed by *e*Rational Solutions Inc. Due to the volume of work, the current paper-based system has proven to be insufficient. A computerized system will improve the reliability and efficiency.

The development of the system will start immediately due to the early-anticipated release date. It is expected that the software development team will complete the product by April 10th 2006. The initial estimation of the software price is $12,000. In addition to this, Pacific Construction will provide *e*Rational Solutions Inc. with all the information necessary for the development of the system. It is anticipated that the product will eventually be introduced to the general market perhaps with enhanced features.

Conventions Used In This Document

- All objects in this document are capitalized.
- All attributes are italicized.

Table of Contents

1.0 Introduction

1.1 System Overview

The system is a Computer Customer Relationship Management (CRM) tool used by construction companies to keep track of customers and employees. The system will store information about the customer or the employee such as personal information (i.e. name and address) as well as custom fields created by each user that matches his or her business model. These custom fields can be number such as salary or date such as first contact date or just a basic string. Users of the system will be able to display and modify all the information that is available for a particular contact. He or she can add or delete contact through user friendly interfaces. Each time a contact profile is updated, the date will be saved for future references. User can easily search the list of contacts by on any of the available fields. The result of the search can be either displayed on the monitor or be sent to the printer.

The system is necessary as it can be very complicated to keep track of each contact. The customer is currently using a paper-based system to track their contact, and as such they are unable to keep up with the volume inquiries. Contacts information is frequently lost, and is generally in a state of disarray. Users are unable to generate reports, they cannot search the records, and they complain that more time is spent organizing customers database than is spent dealing with the actual information.

1.2 Business Objectives

The product is a custom job for Pacific Construction Inc., a company of about 15 people. Pacific Construction specializes in residential and commercial construction. Pacific Construction has been enjoying a 200% growth in past three years, and the management has big plans for the coming years.

*e*Rational Solutions plans to build the product to Pacific Construction’s specifications. Pacific Construction has indicated that the product will be English language only. After this is completed, *e*Rational Solutions will start another company under a different name (to be determined). This new company will take the existing product, extract a more general version (and possibly add multi-language support), and sell it to the general software market. If this proves to be unfeasible, another option is to sell the new product to other companies in the construction business. This is all pending advice from our legal department.

A preliminary version of the product is set to be demonstrated to the management of Pacific Construction on March 1st 2006. The beta and final demonstrations have been set to take place on March 20th and April 10th of 2006 respectively.

1.3 Software Project Outline

The product will be started immediately due to the early targeted release date. The requirements specification will be completed by January 20th 2006. It will require approximately 200 man-hours to complete the project by April 10th 2006. It has been decided that an Object Oriented methodology implemented in C# would be suitable for this type of project.

2.0 Environment

2.1 Interface overview

2.1.1 Context Diagram Discussion

The context diagram depicts any external systems or entities with which Contacts Central® must interface, and the flow of data between them. The above diagram illustrates that the keyboard will be providing information to the Contacts Central®, and the system will supply data to the Terminal and Printer. The large circle represents the boundary between entities contained by the system and those external to the system.

- **Keyboard**: The keyboard will be the main data source for the system. Any information about contacts will be entered through the keyboard.
- **Terminal:** The terminal will be a data sink for queries and reports.
- **Printer:** The printer will also be a data sink for queries and reports.

2.2 Hardware

2.2.1 Web Server

- Pentium 4 2.0 GHz Recommended
- 512 MB RAM
- Minimum 10 GB of free hard disk space
- 10/100/1000 MBit Network card
- Wideband internet access with an at least one static IP address
- 3 ½” Floppy disk drive
- CD-ROM
- It is required to have Microsoft .NET® platform to be installed on the web server

2.2.2 Database Server

- Pentium 4 2.0GHz
- 1 GB RAM
- Minimum 30 GB disk space
- The server must have SCSI disk drives in a RAID configuration
- 10/100/1000 MBit Network card
- Wideband internet access with an at least one static IP address
- Backup system(tape or optical)
- 3 ½” Floppy disk drive
- CD-ROM

2.2.3 Client

- Internet Explorer 5.5 and above are fully supported. Other browsers may run into some problems.

2.3 Operating System

The web server and the database server are required to run one of the following:

- Microsoft Windows® 2003 Server
- Microsoft Windows® 2000 Server
- Windows 2000 Advanced Server
- Microsoft Windows NT® Server version 4.0 Service Pack 5 (SP5) or later,
- Windows NT Server 4.0 Enterprise Edition with SP5 or later

2.4 Database

For this project, we will be using SQL Server 2000 Standard Edition.

2.5 Development Environment

The software will be developed using Microsoft Visual Studio 2002 as *e*Rational Solutions already owns this product. We will be using C# as the main language to develop this product.

2.6 Maintenance Environment

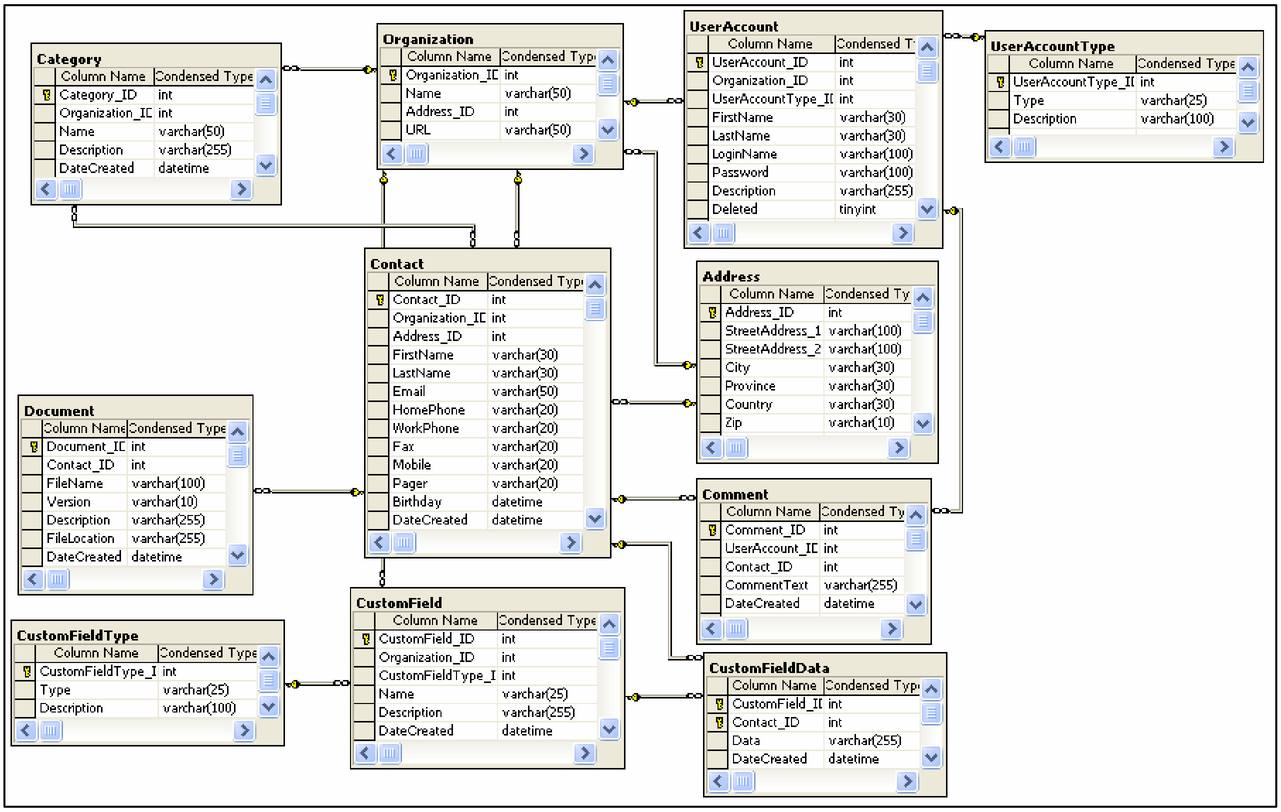
- The maintenance of the product will be the responsibility of the Maintenance Division of eRational Solutions Inc. as the customer will not have access to the source code.
- The system will be available for operations 24 hours of the day.
- Any maintenance or upgrades of the system is done while a backup system comes online in order to avoid any disruption. In a case that a system shutdown is needed, this operation will be done when the number of the users is the lowest. We will also make sure to notify all the users of the shutdown.
- The new version will be backwards compatible so users won't be loosing their clients' information.
- Customers will be notified by e-mail when the new releases or updates have been applied.

2.7 Customer Support Services

*e*Rational Solutions will provide new customers with 3 months of unlimited support. After this period is exhausted, there will be different packages available that can be purchase for by customers who would like to take advantage of the full scope of the software.

3.0 Retained Data Model

3.1 Entity Relationship Diagram (ERD)



Category

	Column Name	Condensed Type
🔑	Category_ID	int
	Organization_I	int
	Name	varchar(50)
	Description	varchar(255)
	DateCreated	datetime

Organization

	Column Name	Condensed Typ
🔑	Organization_I	int
	Name	varchar(50)
	Address_ID	int
	URL	varchar(50)

UserAccount

	Column Name	Condensed T
🔑	UserAccount_ID	int
	Organization_ID	int
	UserAccountType_I	int
	FirstName	varchar(30)
	LastName	varchar(30)
	LoginName	varchar(100)
	Password	varchar(100)
	Description	varchar(255)
	Deleted	tinyint

UserAccountType

	Column Name	Condensed Type
🔑	UserAccountType_I	int
	Type	varchar(25)
	Description	varchar(100)

Contact

	Column Name	Condensed Typ
🔑	Contact_ID	int
	Organization_I	int
	Address_ID	int
	FirstName	varchar(30)
	LastName	varchar(30)
	Email	varchar(50)
	HomePhone	varchar(20)
	WorkPhone	varchar(20)
	Fax	varchar(20)
	Mobile	varchar(20)
	Pager	varchar(20)
	Birthday	datetime
	DateCreated	datetime

Address

	Column Name	Condensed Ty
🔑	Address_ID	int
	StreetAddress_1	varchar(100)
	StreetAddress_2	varchar(100)
	City	varchar(30)
	Province	varchar(30)
	Country	varchar(30)
	Zip	varchar(10)

Document

	Column Name	Condensed Type
🔑	Document_I	int
	Contact_ID	int
	FileName	varchar(100)
	Version	varchar(10)
	Description	varchar(255)
	FileLocation	varchar(255)
	DateCreated	datetime

Comment

	Column Name	Condensed Ty
🔑	Comment_ID	int
	UserAccount_I	int
	Contact_ID	int
	CommentText	varchar(255)
	DateCreated	datetime

CustomFieldType

	Column Name	Condensed Type
🔑	CustomFieldType_I	int
	Type	varchar(25)
	Description	varchar(100)

CustomField

	Column Name	Condensed Type
🔑	CustomField_ID	int
	Organization_ID	int
	CustomFieldType_I	int
	Name	varchar(25)
	Description	varchar(255)
	DateCreated	datetime

CustomFieldData

	Column Name	Condensed Typ
🔑	CustomField_I	int
🔑	Contact_ID	int
	Data	varchar(255)
	DateCreated	datetime

Figure 1 – Entity Relationship Diagram (ERD)

3.2 Entity Relationship Diagram (ORD) Discussion

For specifics on entity attributes, please refer to the Data Dictionary (Section 6.0). The following discusses the keys of and relationships between the various entities.

3.2.1 Entity: Organization

Primary Key

Organization is uniquely identified by Organization_ID.

Relationships

Client has no foreign keys.

3.2.2 Entity: Contact

Primary Key

Contact is uniquely identified by Contact_ID.

Relationships

Each Contact has an Organization to which he or she belongs. Currently, each individual is only identified as being associated with one Organization, so the relationship has a multiplicity of one. The optionality of this relationship is also one, as each Contact must belong to an Organization.

Each Organization has possibly zero or many Contacts associated with it. If the multiplicity of this relationship were one, it would imply that each Organization could only have one Contact, which is not acceptable. The optionality of this relationship is zero, as it is possible that a given Organization has not had a Contact associated with it.

3.2.3 Entity: User

Primary Key

Each User is uniquely identified by User_ID. The Email is also used as the user name when logging on. Therefore it is apparent that we have to make sure of the uniqueness of every email address.

Relationships

Each Organization has one or many Users and each User belongs to one and only one Organization. This means *Organization_ID* is a foreign key. Therefore the relationship that exists between Organization and User is one to many. This means Organization_ID is a foreign key in the User entity.

3.2.4 Entity: Category

Primary Key

Each Category is uniquely identified by Category_ID.

Relationships

Each Category has an Organization to which it belongs thus *Organization_ID* is a foreign key. Currently, each Category is only identified as being associated with one Organization, so the relationship has a multiplicity of one. The optionality of this relationship is also one, as each Category must belong to an Organization.

Each Organization has possibly zero or many Categories associated with it thus multiplicity is one. The optionality of this relationship is zero, as it is possible that a given Organization has not had a Category associated with it.

3.2.5 Entity: Address

Primary Keys

Each Address is uniquely identified by its *Address_ID*. These *Address_ID*s are assigned by the system such that each *Address_ID* is unique to that particular change.

Relationships

The Address entity doe not have a foreign key associated with it.

3.2.6 Entity: Document

Primary Keys

Each Document is uniquely identified by a *Document_ID*. This ID is system generated and is unique.

Relationships

Each Document has one foreign key, *Contact_ID*. *Contact_ID* has a multiplicity of one, because a multiplicity of many would imply that the given Document belongs to many different Contacts. This could be used to handle public Documents, but is not supported by the current system. The optionality of this relationship is one, as a Document must be a associated with a specific Contact.

Each Contact can be associated with many Documents, as a Contact will typically have several Documents associated with it. The optionality of this relationship must be zero, as a Contact could have no Documents associated with it.

3.2.7 Entity: CustomFieldType

Primary Keys

The primary key of the CustomFieldType class is *CustomFieldType_ID*. This must be unique, as the software will not allow the entry of products with duplicate names.

Relationships

As of this release, Product has no foreign keys.

3.2.8 Entity: Comment

Primary Key

Each Comment is uniquely identified by Comment_ID.

Relationships

Each Comment has a Contact to which it belongs thus *Contact_ID* is a foreign key. Currently, each Comment is only identified as being associated with one Contact, so the relationship has a multiplicity of one. The optionality of this relationship is also one, as each Comment must belong to a Contact.

Each Contact has possibly zero or many Comment associated with it thus multiplicity is one. The optionality of this relationship is zero, as it is possible that a given Contact has not had a Category associated with it.

Another foreign key is *User_ID*. This means Comments are written by a User. Currently, each Comment is only created by one Contact, so the relationship has a multiplicity of one. The optionality of this relationship is also one, as each Comment must belong to a User.
Each User has written zero or many Comments. The optionality of this relationship is zero, as it is possible that a given User has not written a Comment at all.

3.2.9 Entity: CustomField

Primary Key

Each CustomField is uniquely identified by CustomField_ID.

Relationships

Each CustomField has an Organization to which it belongs thus *Organization_ID* is a foreign key. Each CustomField is only identified as being associated with one Organization, so the relationship has a multiplicity of one. The optionality of this relationship is also one, as each CustomField must belong to an Organization.

Each Organization has possibly zero or many CustomField associated with it thus multiplicity is one. The optionality of this relationship is zero, as it is possible that a given Organization has not created any CustomFields.

Another foreign key is *CustomFieldType_ID*. Each CustomField must have a type; therefore the relationship between CustomField and CustomFieldType is one to one.

3.2.10 Entity: CustomFieldData

Primary Key

Each CustomFieldData is uniquely identified by *CustomField_ID* and *Contact_ID*.

Relationships

Each CustomFieldData has Contact to which it belongs thus *Contact_ID* is a foreign key. Each CustomFieldData is only identified as being associated with one Contact, so the relationship has a multiplicity of one. The optionality of this relationship is also one, as each CustomFieldData must belong to a Contact.

Each Contact has possibly zero or many CustomFieldData associated with it thus multiplicity is one. The optionality of this relationship is zero, as it is possible that a given Contact has no CustomFieldData associated with it.

Another foreign key is *CustomField_ID.* Each CustomFieldData must be associated with a CustomField; in the other hand, each CustomField can be associated zero or many CustomFieldData.

3.2.11 Data Exception Handling

To ensure the integrity of the database, the following data exceptions are handled as described:

- Each entity referred to in a foreign key will be checked for validity before the operation is complete. This will prevent problems such as Contact making reference to non-existent Organization and so forth.
- Each entry will be checked for completeness – all relationships with optionality one will be checked to see if they have been completed.
- All entries will be checked for the appropriate format (i.e. date format, phone number format, etc.)
- The current system does not allow for multiple Users with the same *Email* since this property will be used to login. Future releases could include more sophisticated handling of this problem such as taking into account the Organization to which a User belong.

4.0 Functional Description

4.1 First Level Data Flow Diagram(DFD)

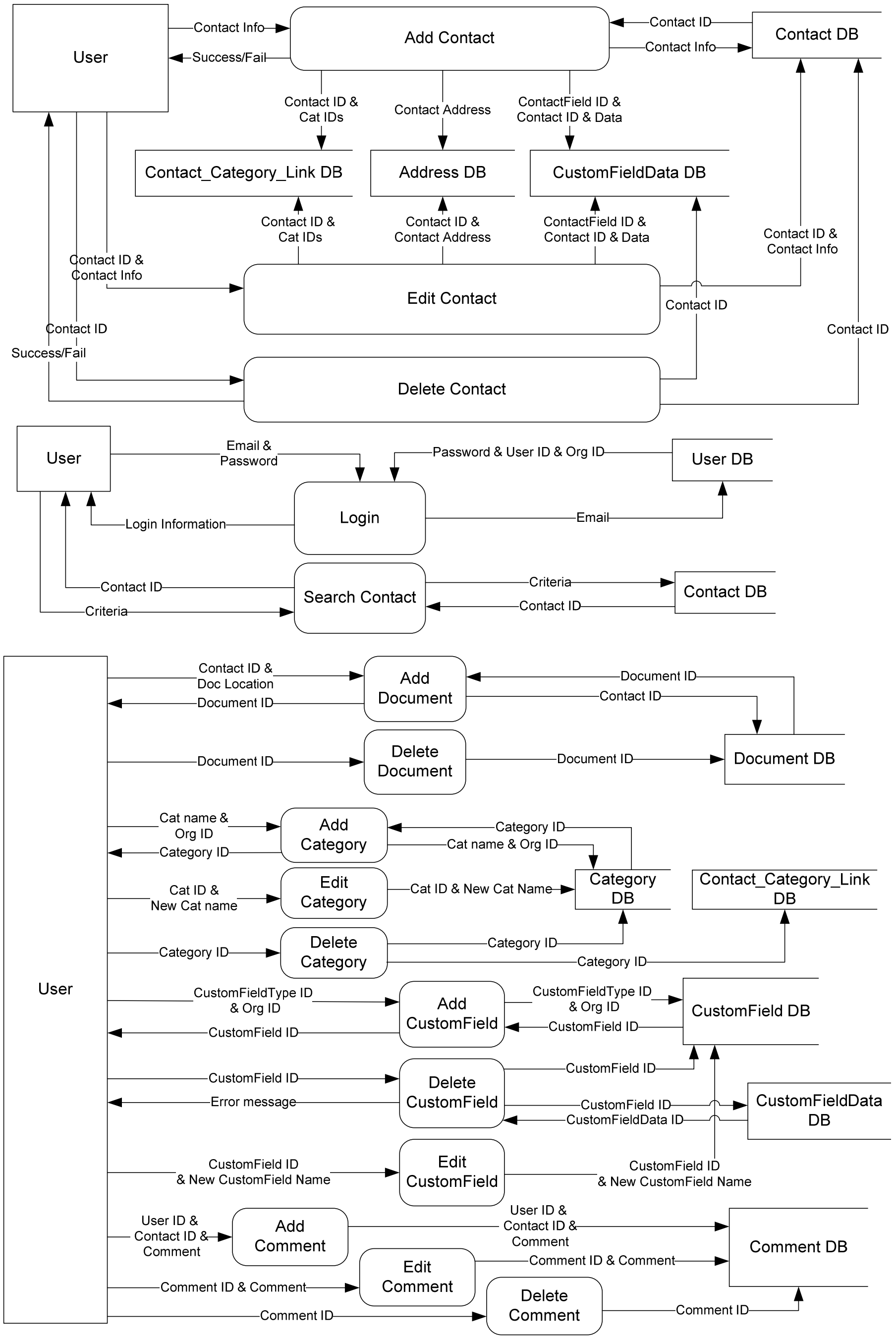


Figure 2 – Level 1 Data Flow Diagram (DFD)

4.2 Process Description

The main processes in Contacts Central® are the Add Contact, Edit Contact, Delete Contact, Add CustomField, Edit CustomField, Delete CustomField, Add Category, Edit Category, Delete Category, Add Document, Delete Document, Add Comment, Edit Comment and Login.

4.2.1 Add Contact:

This process facilitates the addition of new contacts. Information such as personal information, address and category to which the contact belongs to is taken from the user and Contact, Address, Contact_Category_Link and CustomFieldData tables are update.

4.2.2 Delete Contact:

This process removes a contact from user's contact list. User indicates the contact(s) he or she wishes to delete by selecting the contact from a list displayed in a grid format.

4.2.3 Edit Contact:

This process edits a currently existing contact. It can also update the values

4.2.4 Add CustomField:

This process adds a CustomField. It modifies CustomField table.

4.2.5 Edit CustomField:

The only property of a CustomField that can be modified is its name. This process modifies a CustomField's name by modifying CustomField table.

4.2.6 Delete CustomField:

This process deletes a CustomField. Since deleting a CustomField might cause loss data, this process checks CustomFieldData table and warns the user that in case of deleting the CustomField, all data associated with the CustomField will be deleted too.

4.2.7 Add Category:

This process adds categories which can be used to organize the contact list. Each contact can me a member of one or more categories.

4.2.8 Edit Category:

The only property of a Category that can be modified is its name. This process modifies a Category's name by modifying Category table.

4.2.9 Delete Category:

Process deletes a category from Category table. It also deletes records that link a contact to the category by modifying Contact_Category_Link.

4.2.10 Add Document:

This process uploads a document that is associated with a Contact to the server and updates the Document table.

4.2.11 Delete Document:

This process deletes a document from the server and update Document table.

4.2.12 Add Comment:

This process is used to add comments under a Contact profile. It modifies Comment table.

4.2.13 Edit Comment:

The text for a comment can be modified by the user. This task is done through Edit Comment process. Edit Comment process modifies Comment table by updating the text associated with a particular Comment.

4.2.14 Delete Comment:

This process is used to delete Comments associated with Contact. It modifies Comment table.

4.2.15 Search Contact:

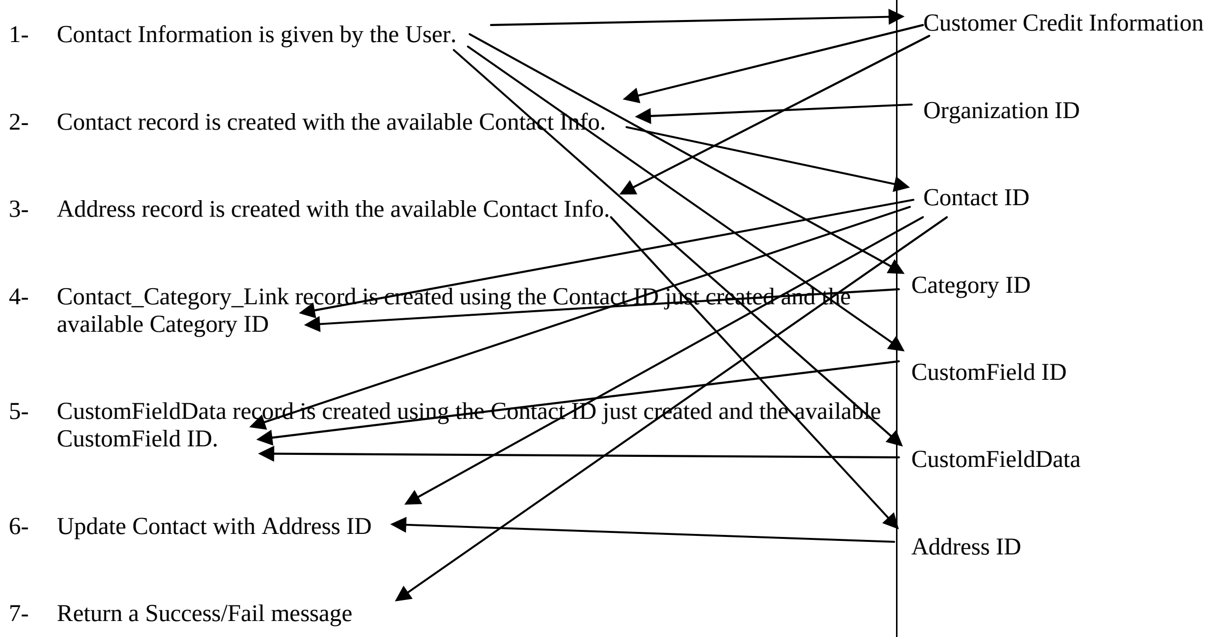
This goal of this process is to search Contact, Contact_Category_Link and CustomFieldData tables with the criteria indicated by the user. These criteria can be name, category, email address or even values under custom fields.

4.2.16 Login:

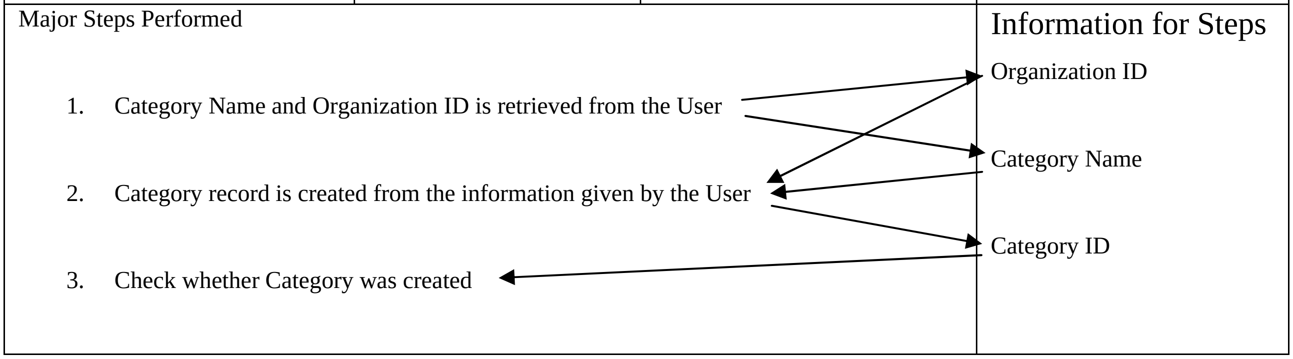
This process takes the Users email and passwords, checks the database and if they match, it will login the User.

5.0 Use Case Scenarios

5.1 Use Case Scenario # 1

Use Case Name: **Add Contact**			ID Number: **1**
Use Case Description: **This describes how to add a contact**			
Trigger: **User makes a request to add a contact** Type: **External** / Temporal			
Major Inputs:		Major Outputs:	
Description	**Source**	**Description**	**Destination**
Add request Organization ID Contact Personal Info Contact Address Category ID(s) CustomField ID CustomFieldData	User User User User User User User	Contact ID Success/Fail message	Add Contact Process User
Major Steps Performed  1- Contact Information is given by the User. 2- Contact record is created with the available Contact Info. 3- Address record is created with the available Contact Info. 4- Contact_Category_Link record is created using the Contact ID just created and the available Category ID 5- CustomFieldData record is created using the Contact ID just created and the available CustomField ID. 6- Update Contact with Address ID 7- Return a Success/Fail message			Information for Steps Customer Credit Information Organization ID Contact ID Category ID CustomField ID CustomFieldData Address ID

5.2 Use Case Scenario # 2

Use Case Name: **Add Category**			ID Number: **2**
Use Case Description: **This describes how to add a Category**			
Trigger: **User makes a request to add a category** Type: **External** / Temporal			
Major Inputs:		Major Outputs:	
Description	**Source**	**Description**	**Destination**
Add request Category Name Organization ID	User User User	Category ID	Add Category Process
Major Steps Performed 1. Category Name and Organization ID is retrieved from the User 2. Category record is created from the information given by the User 3. Check whether Category was created			Information for Steps Organization ID Category Name Category ID

5.3 Use Case Scenario # 3

Use Case Name: **Delete Document**			ID Number: **3**
Use Case Description: **This describes how to delete a Document**			
Trigger: **User makes a request to delete a Document** Type: **External** / Temporal			
Major Inputs:		Major Outputs:	
Description	**Source**	**Description**	**Destination**
Delete request Contact ID Location	User User Document DB	Success/Fail message	User
Major Steps Performed 1. Request is submitted 2. Location of the document retrieved from Document DB 3. Document is deleted from the server and database			Information for Steps Contact ID Document ID Location

6.0 Future Releases

6.1 Future Features

For the future releases a number of features are planned to be implemented. Here are some of them:

- The system should allow the user to be able to send email, snail mail, fax or sms message to contacts.
- Communication mentioned above should be automated. This means user should be able to set a criteria and when these criteria is met then an email (or fax, etc) is sent to a pre-selected group of contact.
- Consider moving the software with open licenses in order to reduce costs.
- The system should implement multiple users that have the same user name (i.e. Email). Organization name can be used in conjunction with the email to ensure uniqueness.
- There should be permission settings for each user so user can be limited to what changes they can make.

7.0 Acceptance Criterion

7.1 Hard Requirement

- Designer/consultants and Customer may agree to allow certain parts of the software to exceed standardized performance criteria. In this event, such circumstances shall be explicitly identified and excluded from requirements to pass standardized tests.
- Acceptance shall be subject to completion of all work, successful post-installation testing which yields 100% PASS rating, and receipt of full documentation.

7.2 Goals

- Be absolutely reliable and consistent.
- Be easy to operate (The assumption is that the system operator has a limited knowledge of computers).
- Be bug free (Any corruption in the data will be costly).
- Be cost-effective (updating and maintenance must be reasonably priced).

7.3 Acceptance Test Requirements

- The main testing will be done by the Quality Assurance division of the *e*Rational Solutions Inc.
- Customer reserves the right to conduct, using Developer's equipment and labor, a random re-test of up to five (5) percent of the system to confirm the documented results. Random re-testing, if performed, shall be at the expense of the customer, using standard labor rates. Any failing part of the software shall be re-tested and restored to a passing condition. In the event more than two (2) percent of the system fails during re-test, the entire system shall be re-tested and restored to a passing condition at no additional cost to the customer.

8.0 Others

8.1 Installation

The installation of the system is done through our IT department. Account created and the customer will be notified with the account information and the URL.

8.2 Backup

A backup system will not be a part of the software; therefore the customer is advised to use backup facilities available in the market to make sure that important data is stored properly.

8.3 Online help

There will be online help available. Users can download these files in Compiled HTML Help (CHM) that can be used locally or they can use the HTML format help files that will be available online.

Appendix A: *Glossary*

Bug **–** an unexpected error in a program caused by faulty logic.

Bug fix **–** a repair of a bug in software.

CASE tool **–** Computer Aided Software Engineering tool.

Computer Aided Software Engineering tool **–** a tool used to enhance a process using computer software.

Data exception handling **–** the handling of invalid data (i.e. “1234” for an email address).

DFD **–** abbreviation for Data Flow Diagram. Shows the flow of information through a system.

Feature **–** a property of software outside of its main task.

Feature request – a request for a feature in software.

GUI **–** Graphical User Interface. An interface in software typically featuring menus and scrollbars, and utilizing a mouse

Object Oriented – a programming paradigm structured around the data a system processes, as opposed to the processes itself.

Operating System – the software which interacts with the computer’s hardware and provides a layer of abstraction between the hardware and a running application.

Prototype **–** a small test version of software intended to give the user a feel for what the completed product will look like.

RAM – Random Access Memory. RAM is short-lived memory that can be accessed at any address in a constant amount of time.

Remote diagnoses **–** a diagnosis of software performed from a remote location.

Response time **–** time the system takes to perform an operation for a given input.

System **–** the combination of specific software running on specific hardware.

Throughput **–** a rate of events, data, etc.

User **–** in this document user refers to the user of the software.