# INTRODUCTION

CRM is a strategy and technology that is used to build stronger relationships between organizations and their clients. An organization will store information that is related to their clients, and employees will analyze it to use it in forecasting and making reports. Some of the strategies of CRM software are planning for targeted marketing campaigns towards specific clients. The strategies used will be relying on the information that is coming from the CRM system. CRM software solution is commonly used by all kind of corporations that focusing on maintaining a strong relationship with their customers.

There are many reasons why CRM software has become the most important software in the last 10 years. The competition has become highly competitive in the global market, and it has become easier for clients to switch companies if they are not happy with the service they receive or want the product with the low price. One of the primary goals of CRM solution is to retain clients and protect them from competitor’s temptations. When CRM is used effectively, an organization will be able to build a relationship with their clients that can last a lifetime. It is so important to realize that CRM system solution is more than just software.

This CRM introduction aims to explaining what the CRM is in a nutshell and making it easier to realize the tremendous benefits of purchasing Customer Relationship Management software that will help drive more customers thus more profit towards you and makes your life easier.

## 1.1. Organization Profile

Forbes Advisor is a global platform dedicated to helping consumers make the best financial choices for their individual lives. We support your pursuit of success by making smart financial decisions simple, to help you get back to doing the things you care about most. We do this by helping turn your aspirations into reality. By arming you with trusted advice and guidance, you can make informed financial decisions you feel confident in and achieve your financial goals. Visit Forbes Advisor for unbiased personal finance advice, news and reviews, plus a comparison marketplace that helps you find the financial products that best fit your life and goals.

**Website**

[**https://www.forbesadvisor.com**](https://www.linkedin.com/redir/redirect?url=https%3A%2F%2Fwww%2Eforbesadvisor%2Ecom&urlhash=pErt&trk=about_website)

**Industries**

Consumer Services

**Company size**

201-500 employees

**Headquarters**

Jersey City, New Jersey

**Type**

Privately Held

**Founded**

1917

**Specialties**

Personal Finance, Best Consumer Financial Products, Credit Cards, Student Loans, Personal Loans, Mortgages, Insurance, Life Insurance, Travel Insurance, Bank Accounts, Travel, Online Brokers, Investing, Retirement, Car Insurance, Checking Accounts, Energy, Cars, Credit Card Rewards, and Comparison Marketplace

## System Specifications

### HARDWARE CONFIGURATION

**Processor** : Pentium -IV

**Speed** : 1 GHz

**Hard Disk Capacity** : 40GB

**RAM Capacity** : 1GB RAM

**CD-ROM Drive** : 52x speed

**Keyboard** : 104 keys

**Mouse** : Logitech

**Printer** : HP3745 series DeskJet printer

### SOFTWARE SPECIFICATION

**Operating System** : Windows 7/8/10

**Front End** : JAVA

**Back End** : SQL

**Feasibility Study**

# SYSTEM STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

The feasibility of a proposed solution is evaluated in teams of its components. These components are:

* + - * Economic feasibility
      * Technical feasibility

## Economic Feasibility

The economic feasibility study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development or the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

## Technical Feasibility

The technical feasibility study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. The will lead to high demands on the available technical resources. This will lead to high demands being places on the client. The developed system must have modest requirements, as only minimal or null changes are required for implementing this system.

## EXISTING SYSTEM

There is not any existing system for client side. SO all the work are handle manually and have to be noted down in some register and also taking care of that documentation. They are arranged meeting by call and if any update occurred then again call the client and update meeting schedule, its wasting time and as well as money also and also the disturb the valuable clients.

### DRAWBACKS

* Complexity: CRM systems can be complex and require significant training and resources to implement and maintain. This can be particularly challenging for smaller businesses with limited staff and resources.
* Data Quality: CRM systems rely heavily on accurate and up-to-date data to be effective. If data is not entered correctly or is outdated, the system may not work as intended, which can lead to customer dissatisfaction.

## PROPOSED SYSTEM

The focus of customer relationship management is on creating value for the customer and the company over the longer term. when the customers value the customer service that they receive from suppliers, they are less likely to look to alternative suppliers for their needs and CRM enables organization to gain competitive advantage over competitors that supply similar products or services.

### FEATURES

* Customer Service and Support:

Many CRM systems include tools for managing customer service and support, such as case management, ticketing, and knowledge management.

* Analytics and Reporting:

A CRM system provides businesses with insights into customer behavior and trends through data analytics and reporting.

# SYSTEM DESIGN AND DEVELOPMENT

Design is concerned with identifying software components specifying relationship Among components. Specifying software structure and providing blue print for the document phase. Modularity is one of the desirable properties of large systems. It implies that the system is divided into several parts. In such a manner, the interaction between parts is Minimal clearly specified. Design will explain software components in details. This will help the implementation of the system. Moreover, this will guide the further changes in the system to satisfy the further requirements.

The design document describes how to transform, the requirement and the functional design into more technical system design specification. This design involves conceiving and planning out in the mind and making a drawing pattern of sketch of. It includes type of activities, External Design, Architectural Design and Detailed Design. The architectural design and detailed design collectively referred to as internal design.

The external design involves specifying the externally observable characteristics of a software product and the internal design involves specifying the internal structure and processing details of the system. The fundamental concept of the design includes abstraction structure, information hiding Modularity, concurrency, verification and design aesthetics.

## FILE DESIGN

In computing, a file design (or file system) is used to control how data is stored and retrieved. Without a file system, information placed in a storage area would be one large body of data with no way to tell where one piece of information stops and the next begins. By separating the data into individual pieces, and giving each piece a name, the information is easily separated and identified. Taking its name from the way paper-based information systems are named, each group of data is called a "file". The structure and logic rules used to manage the groups of information and their names are called a "file system".

Some file systems are used on local data storage devices; others provide file access via a network protocol. Some file systems are "virtual", in that the "files" supplied are computed on request or are merely a mapping into a different file system used as a backing store. The file system manages access to both the content of files and the metadata about those files. It is responsible for arranging storage space; reliability, efficiency, and tuning with regard to the physical storage medium are important design considerations.

Following files are available in this application

## INPUT DESIGN

The input design is the process of entering data to the system. The input design goal is to enter to the computer as accurate as possible. Here inputs are designed effectively so that errors made by the operations are minimized.

The inputs to the system have been designed in such a way that manual forms and the inputs are coordinated where the data elements are common to the source document and to the input. The input is acceptable and understandable by the users who are using it.

Input design is the process of converting user-originated inputs to a computer-based format input data are collected and organized into group of similar data. Once identified, appropriate input media are selected for processing.

The input design also determines the user to interact efficiently with the system. Input design is a part of overall system design that requires special attention because it is the common source for data processing error. The goal of designing input data is to make entry easy and free from errors.

Input design is the process of connecting the user-originated inputs into a computer to used format.

The goal of the input design is to make the data entry logical & free from errors.

## OUTPUT DESIGN

Output design is the process of converting computer data into hard copy that is understood by all. The various outputs have been designed in such a way that they represent the same format that the office and management used to.

Computer output is the most important and direct source of information to the user. Efficient, intelligible output design should improve the systems relationships with the user and help in decision making. A major form of output is the hardcopy from the printer.

Output requirements are designed during system analysis. A good starting point for the output design is the Data Flow Diagram (DFD). Human factors educe issues for design involves addressing internal controls to ensure readability.

The output form in the system is either by screen or by hard copies. Output design aims at communicating the results of the processing of the users. The reports are generated to suit the needs of the users. The reports have to be generated with appropriate levels.

All reports are output formats, maintained details can be reported over crystal reports, this project sustain following reports

## DATABASE DESIGN

The most important consideration in designing the database is how information will be used.

The main objectives of designing a database are:

### Data Integration

In a database, information from several files are coordinated, accessed and operated upon as through it is in a single file. Logically, the information are centralized, physically, the data may be located on different devices, connected through data communication facilities.

### Data Integrity

Data integrity means storing all data in one place only and how each application to access it. This approach results in more consistent information, one update being sufficient to achieve a new record status for all applications, which use it. This leads to less data redundancy; data items need not be duplicated; a reduction in the direct access storage requirement.

### Data Independence

Data independence is the insulation of application programs from changing aspects of physical data organization. This objective seeks to allow changes in the content and organization of physical data without reprogramming of applications and to allow modifications to application programs without reorganizing the physical data.

The tables needed for each module were designed and the specification of each and every column was given based on the records and details collected during record specification of the system study.

## SYSTEM DEVELOPMENT

The key to control maintenance costs is to design systems that are easy to change, so the link between development and maintenance is very strong. Many of the analysis and design methodologies, tools, and techniques employed during system development can be applied to system maintenance, but there are significant differences between development and maintenance. Maintainability is the ease with which software can be understood, corrected, adopted and enhanced.

### DESCRIPTION OF MODULES

To develop this project several step should be followed. There are various modules in this proposed system they are listed below.

### Sales Manager:

A Sales manager using this web page to login and do their activity work. In login page the manager select the sales manager radio button and enter the login credential to access this application. Without credential they cannot access this application.

### Sales People:

A Sales people also use the same page to access this login page, but instead of selecting the sales manager option they should choose the sales people option to use the login page. A sales people use the valid credential to cross the login page otherwise it will show an error which does not allow to access this application.

### Create/manage Task:

Both user can create the task but only the sales manager can have an access to manage the task details. When the user wants to remove or update the ticket details they can have an access to these details. Once the task has been created it will be assigned to the sales people.

### Create/manage Leads:

Sales manager can create and manage the lead details, every details will be stored in a database tables. Once the details are stored then can modify the details.

### Task Report:

Finally can view the details of the task report, which could to be generate overall performance and task status whether they have completed or not. Also whenever they want to download or print the report which could be possible.

# TESTING AND IMPLEMENTATION

System testing is the process of exercising software with the intent of finding and ultimately correcting errors. This fundamental philosophy does not change for web applications, because Web-based systems and application reside on a network and interoperate with many different operating system, browsers, hardware platforms, and communication protocols; the search for errors represents a significant challenge for web application.

The distributed nature of client\server environments, the performance issues associated with transaction processing, the potential presence of a number of different hardware platforms, the complexities of network communication, the need to serve multiple clients from a centralized database and the requirements imposed on the server all combine to make testing of client\server architectures.

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer based system. System testing is the state of implementation that is aimed at assuring that the system works accurately and efficiently. Testing is the vital to the success of the system. System testing makes the logical assumption that if all the parts of the system are correct, the goal will be successfully achieved.

### Unit Testing

Unit testing focuses verification efforts on the smallest unit of software design of the module. This is also known as “module testing”. This testing is carried out during programming stage itself. In this testing step, each module is found to be working satisfactorily as regards to the expected output of the modules.

### Integration Testing

Data can be lost across an interface, one module can have adverse effect on another sub function when combined it may not produce the desired major functions. Integration testing is a systematic testing for constructing test to uncover errors associated within an interface.

The objectives taken from unit tested modules and a program structure is built for integrated testing.

All the modules are combined and the test is made.

A correction made in this testing is difficult because the vast expenses of the entire program complicated the isolation of causes. In this integration testing step, all the errors are corrected for next testing process.

### Validation Testing

After the completion of the integrated testing, software is completely assembled as a package; interfacing error has been uncovered and corrected and a final series of software test validation begins.

Validation testing can be defined in many ways but a simple definition is that validation succeeds when the software function in a manner that can be reasonably expected by the customer. After validation test has been conducted, one of two possible conditions exists:

### Output Testing

The next process of validation testing, is output testing of the proposed system, since no system could be successful if it does not produce the required output in the specified format. Asking the user about the format required, list the output to be generated or displayed by the system under considerations.

Output testing is a different test whose primary purpose is to fully exercise the computer based system although each test has a different purpose all the work should verify that all system elements have been properly integrated and perform allocated functions.

The output format on the screen is found to be corrected as the format was designed in the system design phase according to the user needs for the hard copy also; the output testing has not resulted in any correction in the system.

**IMPLEMENTATION**

System implementation is the stage of the project that the theoretical design is turned into a working system. If the implementation stage is not properly planned and controlled, it can cause error. Thus it can be considered to be the most crucial stage in achieving a successful new system and in giving the user confidence that the new system will work and be effective.

Normally this stage involves setting up a coordinating committee, which will act as a sounding board for ideas; complaints and problem. The first task is implementation planning; i.e., deciding on the methods and time scale to be adopted. Apart from planning two major task of preparing for implementation are, education takes place much earlier in the project; at the implementation stage the emphasis must be on training in new skills to give staff confidence they can use the system. Once staff has been trained, the system can be tested.

After the implementation phase is completed and the user staff is adjusted to the changes created by the candidate system, evaluation and maintenance is to bring the new system to standards.

# CONCLUSION

In conclusion, customer relationship management (CRM) is a strategy that helps businesses manage their interactions with customers to improve relationships, increase customer loyalty, and drive revenue growth. A CRM system provides businesses with tools to manage customer data, automate sales and marketing processes, and track customer interactions across multiple channels.

While implementing a CRM system can have significant benefits, it is important to carefully consider the potential drawbacks and ensure that the system is aligned with the needs of the business and its customers. The success of a CRM system depends on user adoption, data quality, and integration with other business applications.

Overall, a well-implemented CRM system can help businesses better understand their customers and provide personalized experiences that drive customer loyalty and advocacy. By improving customer relationships, businesses can improve their bottom line and achieve long-term success..

## BIBLIOGRAPY

**Books Referred:**

* Bloch, Joshua. Effective Java: Programming Language Guide. Addison-Wesley, 2017.
* Peter Foster & Dr. Thomas B. Schalk , “Speech Recognition the complete Practical Reference Guide”, First Edition, August 1993.
* Horstmann, Cay S. Core Java Volume I--Fundamentals. Prentice Hall, 2018.
* Lea, Doug. Concurrent Programming in Java: Design Principles and Patterns. Addison-Wesley, 2000.
* Naftalin, Maurice, and Philip Wadler. Java Generics and Collections. O'Reilly Media, 2006.
* Elias M. Award, “System Analysis and design", Tata McGraw Hill , 3rd edition.
* Jeffrey A. Hoffer, Joey F. George and Joseph S. Valacich (2005), “Modern systems Analysis & Design”, pearson Education, 3rd Edition.

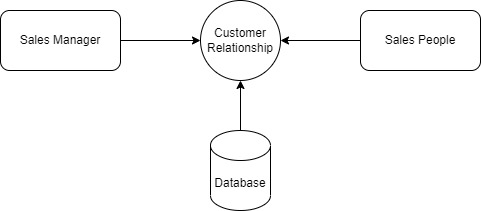
## Websites:

* GeeksforGeeks. "Java Programming Language." GeeksforGeeks, 2023, <https://www.geeksforgeeks.org/java/>.
* Tutorials Point. "Java Tutorial." Tutorials Point, 2023, https://www.tutorialspoint.com/java/index.htm.
* Vogella. "Java Tutorials." Vogella, 2023, <https://www.vogella.com/tutorials/java.html>.

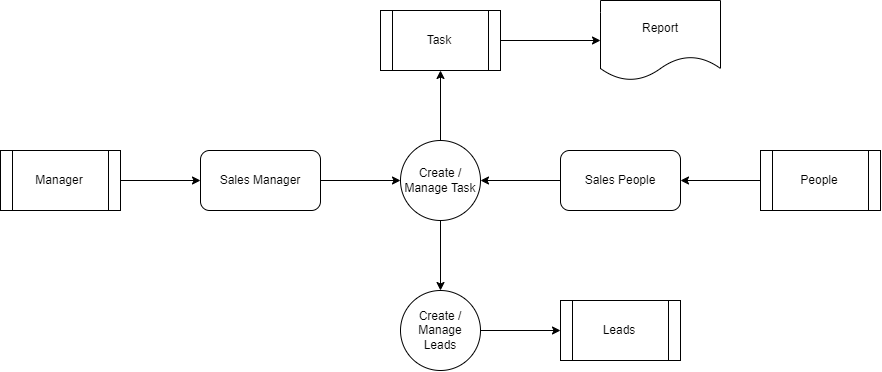
# APPENDICES

## Data Flow Diagram

**Level 0**



**Level 1**



## TABLE STRUCTURE

**Table Name :** User

**Primary Key :** User\_id

**Table Description :** This table is used to maintain the details about User

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **SIZE** | **CONSTRAINT** |
| User\_id | Int | 8 | Primary Key |
| Name | Varchar | 25 | Not null |
| Email | Varchar | 20 | Not null |
| Password | Varchar | 20 | Not null |
| Role | Varchar | 10 | Not null |
| Mobile | Int | 10 | Not null |

**Table Name :** Task

**Primary Key :** Task\_id

**Table Description :** This table is used to maintain the details about Task

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **SIZE** | **CONSTRAINT** |
| Task\_id | Int | 8 | Primary Key |
| Task type | Varchar | 10 | Not null |
| Task Description | Varchar | 50 | Not null |
| Task due date | Date | 10 | Not null |
| Task status | Varchar | 10 | Not null |
| Contact | Int | 10 | Not null |
| Sales rep Id | Int | 10 | Foreign key |

**Table Name :** Sales Rep

**Primary Key :** rep\_id

**Table Description :** This table is used to maintain the details about Task

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **SIZE** | **CONSTRAINT** |
| Rep\_id | Int | 8 | Primary Key |
| Rep Title | Varchar | 10 | Not null |
| First name | Varchar | 15 | Not null |
| Last name | Varchar | 15 | Not null |
| Middle name | Varchar | 15 | Not null |
| Company | Varchar | 15 | Not null |
| Industry | Varchar | 15 | Not null |
| Address | Varchar | 50 | Not null |
| City | Varchar | 15 | Not null |
| State | Varchar | 15 | Not null |
| Country | Varchar | 15 | Not null |
| Zip | Int | 10 | Not null |
| Phone | Int | 10 | Not null |
| Email | Varchar | 25 | Not null |
| Website | Varchar | 30 | Not null |

## B. Sample Coding

<?java

include\_once 'config/Database.java';

include\_once 'class/User.java';

$database = new Database();

$db = $database->getConnection();

$user = new User($db);

if(!$user->loggedIn()) {

header("Location: index.java");

}

include('inc/header.java');

?>

<title>webdamn.com : Demo Customer Relationship Management (CRM) System</title>

<script src="js/jquery.dataTables.min.js"></script>

<script src="js/dataTables.bootstrap.min.js"></script>

<link rel="stylesheet" href="css/dataTables.bootstrap.min.css" />

<script src="js/contact.js"></script>

<script src="js/general.js"></script>

<?java include('inc/container.java');?>

<div class="container" style="background-color:#f4f3ef;">

<h2>Customer Relationship Management (CRM) System</h2>

<?java include('top\_menus.java'); ?>

<br>

<h4>Contact</h4>

<div>

<div class="panel-heading">

<div class="row">

<div class="col-md-10">

<h3 class="panel-title"></h3>

</div>

<div class="col-md-2" align="right">

<button type="button" id="addContact" class="btn btn-info" title="Add Contact"><span class="glyphicon glyphicon-plus"></span></button>

</div>

</div>

</div>

<table id="contactListing" class="table table-bordered table-striped">

<thead>

<tr>

<th>Id</th>

<th>Name</th>

<th>Company</th>

<th>Industry</th>

<th>Budget</th>

<th>Sales Rep</th>

<th></th>

<th></th>

<th></th>

</tr>

</thead>

</table>

</div>

<div id="tasksDetails" class="modal fade">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal">&times;</button>

<h4 class="modal-title"><i class="fa fa-plus"></i> Tasks Details</h4>

</div>

<div class="modal-body">

<table id="" class="table table-bordered table-striped">

<thead>

<tr>

<th>Id</th>

<th>Created</th>

<th>Task Type</th>

<th>Description</th>

<th>Due Date</th>

<th>Status</th>

<th>Contact</th>

<th>Sales Rep</th>

</tr>

</thead>

<tbody id="tasksList">

</tbody>

</table>

</div>

</div>

</div>

</div>

<div id="contactModal" class="modal fade">

<div class="modal-dialog">

<form method="post" id="contactForm">

<div class="modal-content">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal">&times;</button>

<h4 class="modal-title"><i class="fa fa-plus"></i> Edit Sales Representative</h4>

</div>

<div class="modal-body">

<div class="form-group"

<label for="project" class="control-label">Contact First</label>

<input type="text" class="form-control" id="contact\_first" name="contact\_first" placeholder="contact name" required>

</div>

<div class="form-group"

<label for="project" class="control-label">Contact Last</label>

<input type="text" class="form-control" id="contact\_last" name="contact\_last" placeholder="contact last" >

</div>

<div class="form-group">

<label for="address" class="control-label">Comapny</label>

<input type="text" class="form-control" id="contact\_company" name="contact\_company" placeholder="company" required>

</div>

<div class="form-group">

<label for="address" class="control-label">Industry</label>

<input type="text" class="form-control" id="contact\_industry" name="contact\_industry" placeholder="industry" required>

</div>

<div class="form-group">

<label for="address" class="control-label">Budget</label>

<input type="text" class="form-control" id="contact\_budget" name="contact\_budget" placeholder="budget" required>

</div>

<div class="form-group">

<label for="country" class="control-label">Sales Rep</label>

<select class="form-control" id="contact\_sales\_rep" name="contact\_sales\_rep"/>

<?java

$salesRepResult = $user->salesRepList();

while ($salesRep = $salesRepResult->fetch\_assoc()) {

?>

<option value="<?java echo $salesRep['id']; ?>"><?java echo $salesRep['name']; ?></option>

<?java } ?>

</select>

</div>

</div>

<div class="modal-footer">

<input type="hidden" name="id" id="id" />

<input type="hidden" name="action" id="action" value="" />

<input type="submit" name="save" id="save" class="btn btn-info" value="Save" />

<button type="button" class="btn btn-default" data-dismiss="modal">Close</button>

</div>

</div>

</form>

</div>

</div>

</div>

<?java include('inc/footer.java');?>

<?java

include\_once 'config/Database.java';

include\_once 'class/Contact.java';

$database = new Database();

$db = $database->getConnection();

$contact = new Contact($db);

if(!empty($\_POST['action']) && $\_POST['action'] == 'listContact') {

$contact->listContact();

}

if(!empty($\_POST['action']) && $\_POST['action'] == 'addContact') {

$contact->contact\_first = $\_POST["contact\_first"];

$contact->contact\_last = $\_POST["contact\_last"];

$contact->contact\_company = $\_POST["contact\_company"];

$contact->contact\_industry = $\_POST["contact\_industry"];

$contact->contact\_budget = $\_POST["contact\_budget"];

$contact->contact\_sales\_rep = $\_POST["contact\_sales\_rep"];

$contact->insert();

}

if(!empty($\_POST['action']) && $\_POST['action'] == 'getContact') {

$contact->contact\_id = $\_POST["id"];

$contact->getContact();

}

if(!empty($\_POST['action']) && $\_POST['action'] == 'updateContact') {

$contact->contact\_id = $\_POST["id"];

$contact->contact\_first = $\_POST["contact\_first"];

$contact->contact\_last = $\_POST["contact\_last"];

$contact->contact\_company = $\_POST["contact\_company"];

$contact->contact\_industry = $\_POST["contact\_industry"];

$contact->contact\_budget = $\_POST["contact\_budget"];

$contact->contact\_sales\_rep = $\_POST["contact\_sales\_rep"];

$contact->update();

}

if(!empty($\_POST['action']) && $\_POST['action'] == 'deleteContact') {

$contact->contact\_id = $\_POST["id"];

$contact->delete();

}

if(!empty($\_POST['action']) && $\_POST['action'] == 'getTasks') {

$contact->contact\_id = $\_POST["id"];

$contact->getTasks();

}

?><?java

include\_once 'config/Database.java';

include\_once 'class/User.java';

$database = new Database();

$db = $database->getConnection();

$user = new User($db);

if(!$user->loggedIn()) {

header("Location: index.java");

}

include('inc/header.java');

?>

<title>webdamn.com : Demo Customer Relationship Management (CRM) System</title>

<script src="js/jquery.dataTables.min.js"></script>

<script src="js/dataTables.bootstrap.min.js"></script>

<link rel="stylesheet" href="css/dataTables.bootstrap.min.css" />

<script src="js/customer\_win.js"></script>

<script src="js/general.js"></script>

<?java include('inc/container.java');?>

<div class="container" style="background-color:#f4f3ef;">

<h2>Customer Relationship Management (CRM) System</h2>

<?java include('top\_menus.java'); ?>

<br>

<h4>Customer / Won</h4>

<div>

<div class="panel-heading">

<div class="row">

<div class="col-md-10">

<h3 class="panel-title"></h3>

</div>

<div class="col-md-2" align="right">

<button type="button" id="addOpportunity" class="btn btn-info" title="Add Opportunity"><span class="glyphicon glyphicon-plus"></span></button>

</div>

</div>

</div>

<table id="leadsListing" class="table table-bordered table-striped">

<thead>

<tr>

<th>Id</th>

<th>Name</th>

<th>Company</th>

<th>Industry</th>

<th>Phone</th>

<th>Email</th>

<th>Website</th>

<th></th>

<th></th>

<th></th>

</tr>

</thead>

</table>

</div>

<div id="leadsDetails" class="modal fade">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal">&times;</button>

<h4 class="modal-title"><i class="fa fa-plus"></i> Tasks Details</h4>

</div>

<div class="modal-body">

<table id="" class="table table-bordered table-striped">

<thead>

<tr>

<th>Id</th>

<th>Created</th>

<th>Task Type</th>

<th>Description</th>

<th>Due Date</th>

<th>Status</th>

<th>Contact</th>

<th>Sales Rep</th>

</tr>

</thead>

<tbody id="leadsList">

</tbody>

</table>

</div>

</div>

</div>

</div>

<div id="leadsModal" class="modal fade">

<div class="modal-dialog">

<form method="post" id="leadsForm">

<div class="modal-content">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal">&times;</button>

<h4 class="modal-title"><i class="fa fa-plus"></i> Edit Leads</h4>

</div>

<div class="modal-body">

<div class="form-group"

<label for="project" class="control-label">Contact First</label>

<input type="text" class="form-control" id="lead\_first" name="lead\_first" placeholder="contact name" required>

</div>

<div class="form-group"

<label for="project" class="control-label">Contact Last</label>

<input type="text" class="form-control" id="lead\_last" name="lead\_last" placeholder="contact last" >

</div>

<div class="form-group">

<label for="address" class="control-label">Comapny</label>

<input type="text" class="form-control" id="lead\_company" name="lead\_company" placeholder="company" required>

</div>

<div class="form-group">

<label for="address" class="control-label">Industry</label>

<input type="text" class="form-control" id="lead\_industry" name="lead\_industry" placeholder="industry" required>

</div>

<div class="form-group">

<label for="address" class="control-label">Budget</label>

<input type="text" class="form-control" id="lead\_budget" name="lead\_budget" placeholder="budget" required>

</div>

<div class="form-group">

<label for="country" class="control-label">Status</label>

<select class="form-control" id="lead\_status" name="lead\_status"/>

<option value="Lead">Lead</option>

<option value="Proposal">Proposal</option>

<option value="Customer / won">Customer / won</option>

<option value="Archive">Archive</option>

</select>

</div>

<div class="form-group">

<label for="address" class="control-label">Email</label>

<input type="text" class="form-control" id="lead\_email" name="lead\_email" placeholder="Email" required>

</div>

<div class="form-group">

<label for="address" class="control-label">Phone</label>

<input type="text" class="form-control" id="lead\_phone" name="lead\_phone" placeholder="Phone" required>

</div>

<div class="form-group">

<label for="address" class="control-label">Website</label>

<input type="text" class="form-control" id="lead\_website" name="lead\_website" placeholder="Website" required>

</div>

</div>

<div class="modal-footer">

<input type="hidden" name="id" id="id" />

<input type="hidden" name="action" id="action" value="" />

<input type="submit" name="save" id="save" class="btn btn-info" value="Save" />

<button type="button" class="btn btn-default" data-dismiss="modal">Close</button>

</div>

</div>

</form>

</div>

</div>

</div>

<?java include('inc/footer.java');?>

<?java

include\_once 'config/Database.java';

include\_once 'class/Customer.java';

$database = new Database();

$db = $database->getConnection();

$customer = new Customer($db);

if(!empty($\_POST['action']) && $\_POST['action'] == 'listOpportunity') {

$customer->listOpportunity();

}

if(!empty($\_POST['action']) && $\_POST['action'] == 'addOpportunity') {

$customer->lead\_first = $\_POST["lead\_first"];

$customer->lead\_last = $\_POST["lead\_last"];

$customer->lead\_company = $\_POST["lead\_company"];

$customer->lead\_industry = $\_POST["lead\_industry"];

$customer->lead\_budget = $\_POST["lead\_budget"];

$customer->lead\_status = $\_POST["lead\_status"];

$customer->lead\_email = $\_POST["lead\_email"];

$customer->lead\_phone = $\_POST["lead\_phone"];

$customer->lead\_website = $\_POST["lead\_website"];

$customer->insert();

}

if(!empty($\_POST['action']) && $\_POST['action'] == 'getLead') {

$customer->id = $\_POST["id"];

$customer->getLead();

}

if(!empty($\_POST['action']) && $\_POST['action'] == 'updateOpportunity') {

$customer->id = $\_POST["id"];

$customer->lead\_first = $\_POST["lead\_first"];

$customer->lead\_last = $\_POST["lead\_last"];

$customer->lead\_company = $\_POST["lead\_company"];

$customer->lead\_industry = $\_POST["lead\_industry"];

$customer->lead\_budget = $\_POST["lead\_budget"];

$customer->lead\_status = $\_POST["lead\_status"];

$customer->lead\_email = $\_POST["lead\_email"];

$customer->lead\_phone = $\_POST["lead\_phone"];

$customer->lead\_website = $\_POST["lead\_website"];

$customer->update();

}

if(!empty($\_POST['action']) && $\_POST['action'] == 'deleteLead') {

$customer->id = $\_POST["id"];

$customer->delete();

}

if(!empty($\_POST['action']) && $\_POST['action'] == 'getOpportunityDetails') {

$customer->id = $\_POST["id"];

$customer->getOpportunityDetails();

}

?><?java

include\_once 'config/Database.java';

include\_once 'class/User.java';

$database = new Database();

$db = $database->getConnection();

$user = new User($db);

if(!$user->loggedIn()) {

header("Location: index.java");

}

include('inc/header.java');

?>

<title>webdamn.com : Demo Customer Relationship Management (CRM) System</title>

<link href="assets/css/paper-dashboard.css" rel="stylesheet"/>

<link href="assets/css/themify-icons.css" rel="stylesheet">

<script src="js/general.js"></script>

<?java include('inc/container.java');?>

<div class="container" style="background-color:#f4f3ef;">

## D. Sample Input

## Login page for Sales Manager and Sales person

## 

## Input of Add sales representative

## 

## ­

## Input of Task details

## 

## Input of Contact details

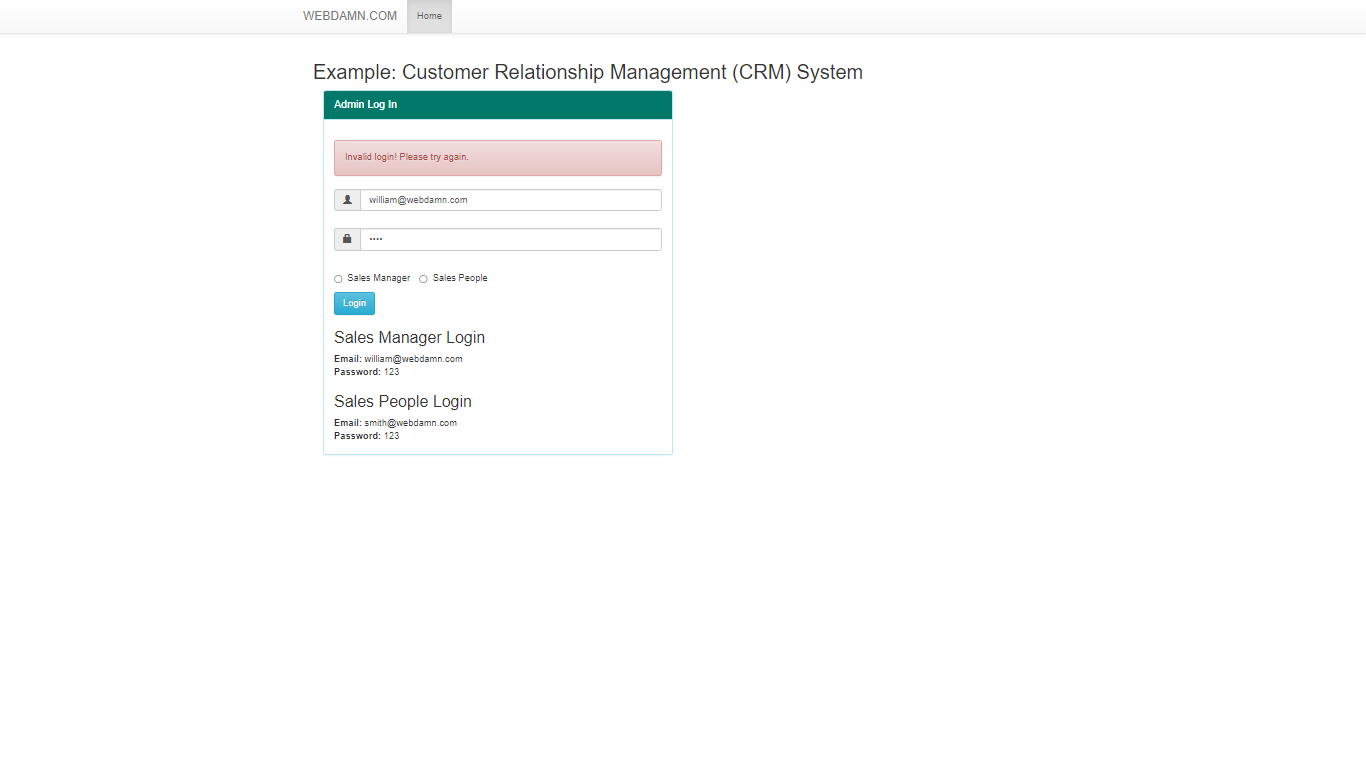
## 

## Input of edit task

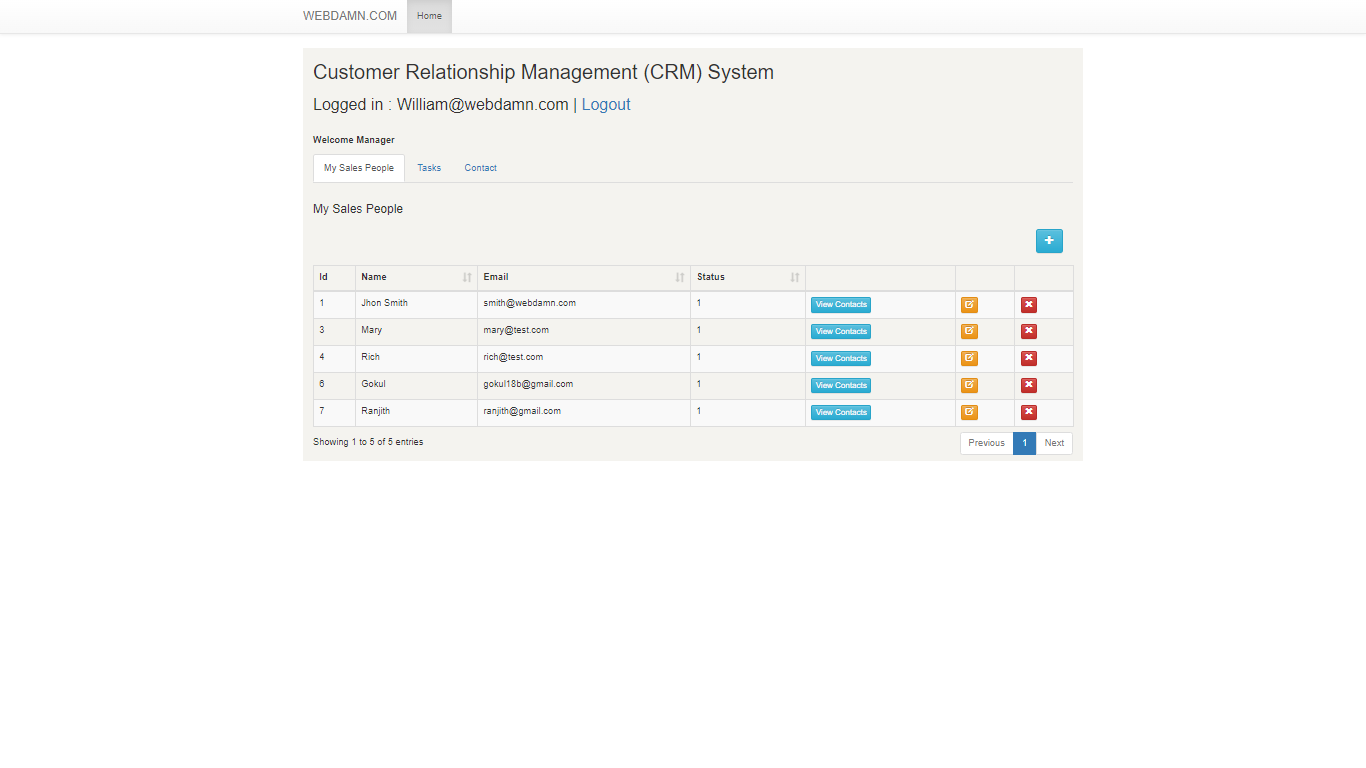
## 

## E. Sample Output

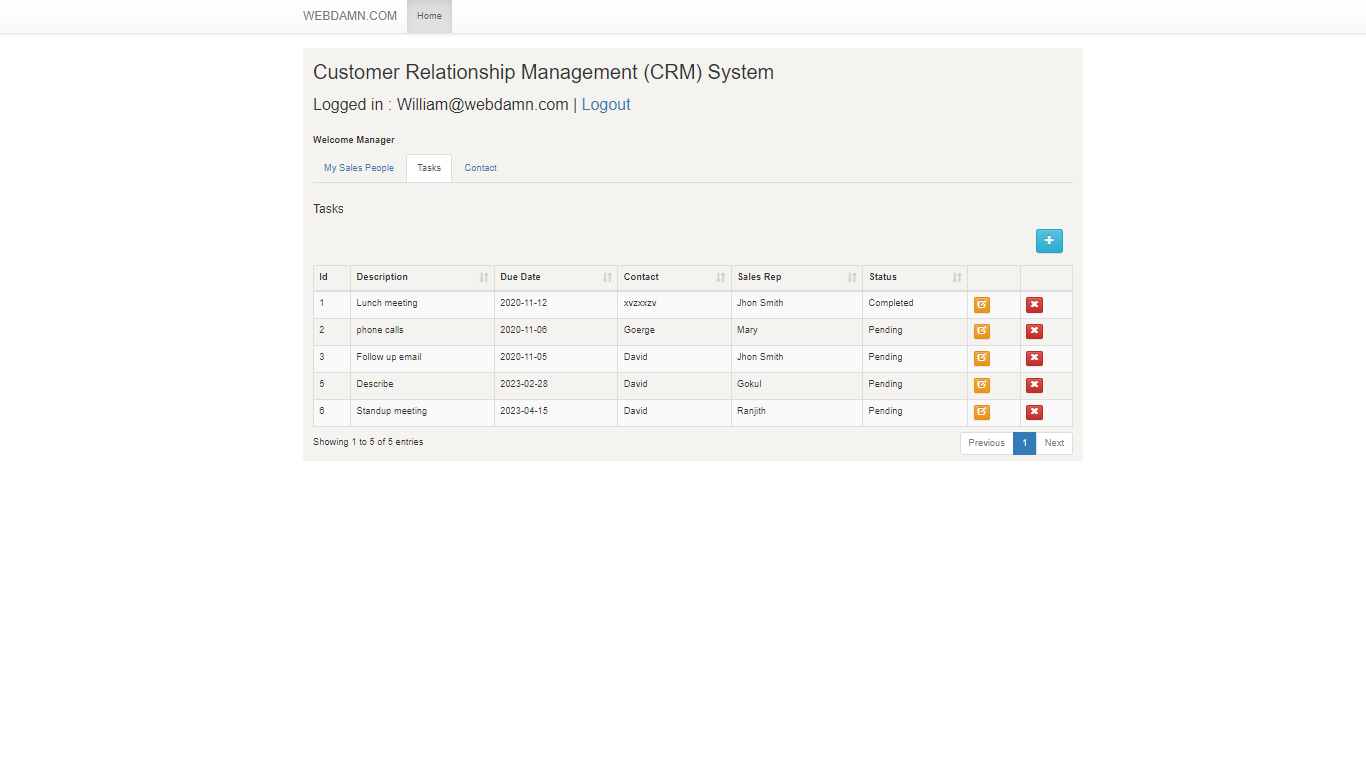
**Output of Invalid credential**



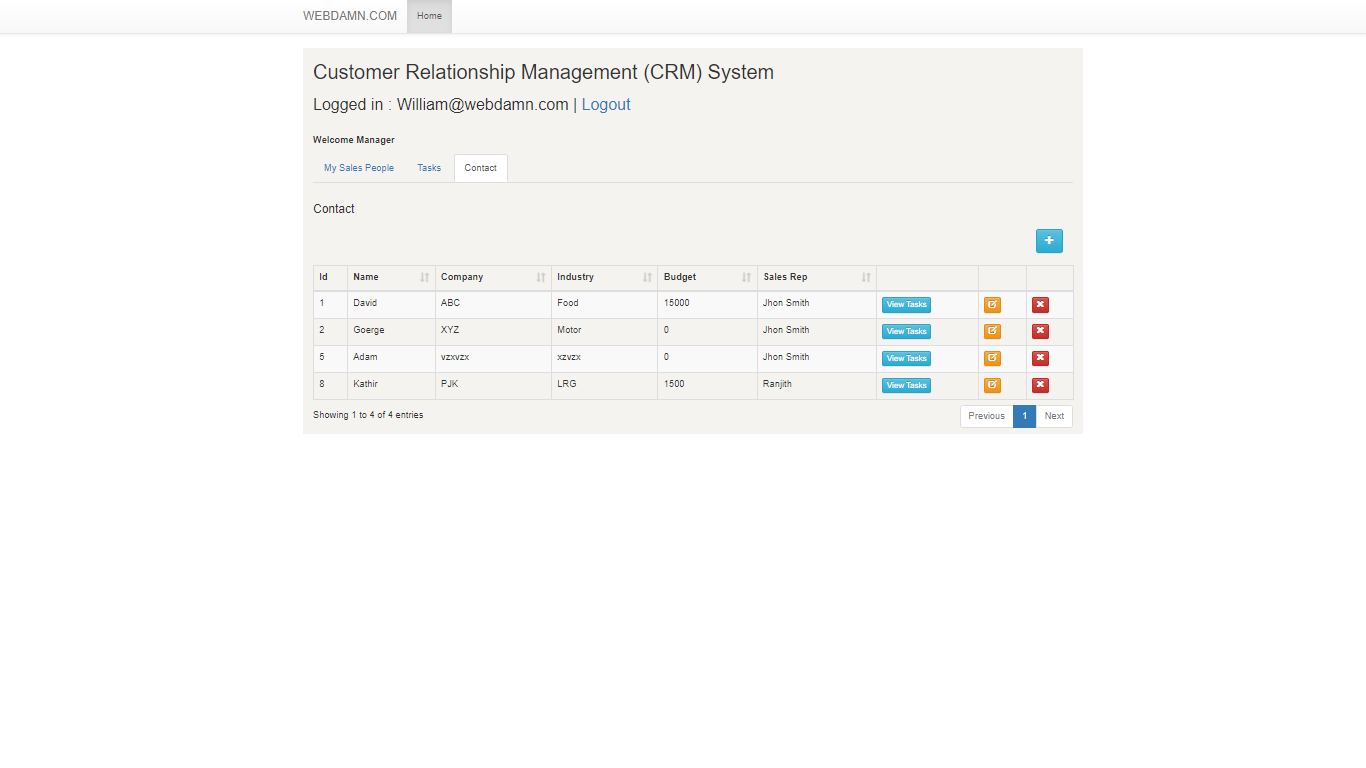
Output of Sales representative page



Output of task details



Output of contact details



Output of Task details

