

Governors State University
OPUS Open Portal to University Scholarship

All Capstone Projects

Student Capstone Projects

Spring 7-1-2017

Sales Management Portal

Dileep Ponnekanti
Governors State University

Puvvala Jeevanth Naga Durga
Governors State University

Vidiyala Surendra
Governors State University

Follow this and additional works at: <http://opus.govst.edu/capstones>

 Part of the [Computer Sciences Commons](#)

Recommended Citation

Ponnekanti, Dileep; Naga Durga, Puvvala Jeevanth; and Surendra, Vidiyala, "Sales Management Portal" (2017). *All Capstone Projects*. 317.
<http://opus.govst.edu/capstones/317>

For more information about the academic degree, extended learning, and certificate programs of Governors State University, go to http://www.govst.edu/Academics/Degree_Programs_and_Certifications/

Visit the [Governors State Computer Science Department](#)

This Project Summary is brought to you for free and open access by the Student Capstone Projects at OPUS Open Portal to University Scholarship. It has been accepted for inclusion in All Capstone Projects by an authorized administrator of OPUS Open Portal to University Scholarship. For more information, please contact opus@govst.edu.

ABSTRACT

The main motive behind this project is the combination of Sales Management Portal and Job searching portal. In this project the client, user, job seeker, investors and the companies can communicate each other using this portal. Manager, HR will update the Project information like budget, revenue, type, proposal and how many opportunities are available based on the project to the job seekers. And people also can be the part of the project by investing on the project as a shareholder. Manager and HR have the authorities to add the client details in the portal. The includes many aspects like Company name, Address, division, industry type, website of the company, Contact details of the HR, the main proposal of the project, opportunities, budget, revenue, duration of the project, deadline of the project etc. This user can login to the website and be the part of the Project for a particular company. Job seekers also can login and contact the HR for the job details. They also can submit the employment application form to the companies through online.

Investors will get the updates about their share values which were invested in the company. Many Multi-National Companies can directly interact to their clients through this portal. And there is lot of scope to the companies to get more investors and clients. The job seekers also can easily get the appropriate jobs without wasting the time for job search and they really can achieve the growth in their career. This is the one place where many people can get the opportunities. This is the right place for everyone for their growth in all sectors. Choose the right path for the growth of your career and organization too. The main goal of this project is to make the companies, clients, job seekers and users to reach mutual grow professionally in one place.

Table of Content

1	<i>Project Description</i>	1
1.1	Competitive Information	1
1.2	Relationship to Other Applications	1
1.3	Assumptions and Dependencies	1
1.4	Future Enhancements	1
2	<i>Technical Description</i>	1
2.1	Application Architecture	2
2.2	Application Information flows	2
2.3	Interactions with other Applications	3
2.4	Capabilities	3
2.5	Risk Assessment and Management	3
3	<i>Project Requirements</i>	4
3.1	Operations, Administration, Maintenance and Provisioning (OAM&P)	5
3.2	Security and Fraud Prevention	5
3.3	Release and Transition Plan	5
4	<i>Project Design Description</i>	5
5	<i>Project Internal/external Interface Impacts and Specification</i>	7
6	<i>Project Design Units Impacts</i>	8
6.1	Functional Area/Design Unit A	10
6.1.1	<i>Functional Overview</i>	10
6.1.2	<i>Impacts</i>	11
6.1.3	<i>Requirements</i>	14
6.2	Screenshots	15
7	<i>Acknowledgements</i>	19
8	<i>References</i>	19

1 Project Description

In the process of fulfillment of the graduate capstone project here we propose a project entitle Sales Management Portal. The main goal of this project is to make the companies, clients, users, jobseekers to communicate each other and make them to fulfill their goals. The Companies can get the investments in their projects easily and opportunist is easily available to the job seekers. Manager can monitor the companies' details like proposal, opportunities, budget, revenue etc. Human Resource department is responsible to monitor the Employment application forms and screening the skilled persons for those particular positions.

1.1 Competitive Information

Identify competitor products, applications, or services that this project will directly compete against are really a major challenge. Fortunately, we didn't find any company or website that provides all the services like opportunities, investors, shareholders, clients at one place. And our team has the potential to be the first to market this new project application or capability.

1.2 Relationship to Other Applications

We are developing a unique project which doesn't have any relationships with other Applications or projects.

1.3 Assumptions and Dependencies

- The details of the project proposal can be considered as sample values, those sample values will be stored in the database for each attribute to check the execution of the program dynamically.
- This project needs the people who have extraordinary knowledge on Human Resource department and share market.

1.4 Future Enhancements

This application we are developing is based on the frameworks, web service and bootstrap for responsive design. This application will work on any browsers and there are no barriers for mobiles, tablets and laptops. This application also runs on multiple operating systems like Android, Linux, windows, mac, whether it's a PC or mobile doesn't matter. In future, if needed any enhancements to be more users friendly or any services requested by the user will take into considerations and will be implemented.

2 Technical Description

This project consists of 9 web pages that include Manager Login, User login, General Info, Notes, contacts, opportunities, proposals, projects, employment application, job announcements etc. These web pages are designed and developed to be more responsive for every screen using Bootstrap.

The Database or schema is created containing all the tables which can store the User data, client data, manager data, messages, project details, proposals, opportunities and user type values of

attributes are stored in the database. We have used multiple modules to develop this application like Model-View-Controller, MySQL database connectors to interact the database.

2.1 Application Architecture

In the commercial world, in this project we use Java Enterprise Edition to solve business problems, to develop commercial software, or to provide contract services to other businesses' projects. In our project we are using a multitier architecture; it usually involves managers, architects, designers, programmers, testers, and database experts throughout the development lifecycle. The first components User are of two kinds, they are Manager and normal user. The Manager has more privileges compare to the normal user. Second Component Room will contain the information of the specific room. The attribute in are project are divided into three categories like Range, Interval and Boolean. The Manager allows the client to invest to the companies.



FIGURE 1. Application Architecture

2.2 Application Information flows

The Manager main component of the architecture, the information flow is shown in the below diagram.

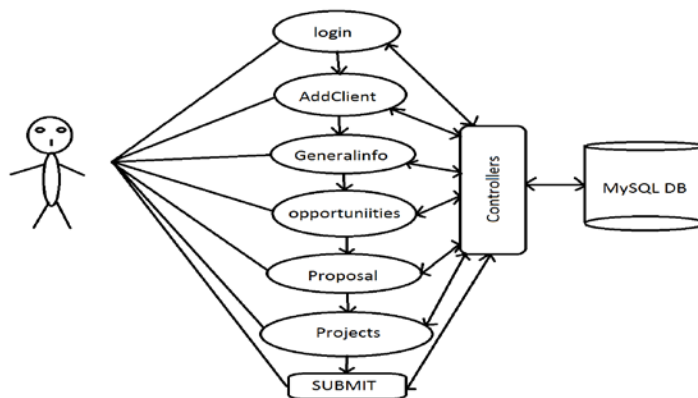


FIGURE 2 Information flow.

2.3 Interactions with other Applications

Since, we have developed a web based application which is indeed an independent application. The plant house can be maintained by comparing the attribute to the sample values stored into the database. Our project does not contain any payment system but it has human resource maintained so the interactions with the other application are null.

2.4 Capabilities

In this application we have two users they are Manager and the normal user. The Manager will have the capability add a client, the proposal and the project details which can viewable by the user. The database which we created has a capability of adding dropping, retrieving the user's data, attribute values through front end. The front end has some capability to provide the validations. Validations are provided in all the web pages created using jquery. Apache tomcat server used as the server.

2.5 Risk Assessment and Management

Risks are common when we start any project. Identification of those risks at the requirement and designing stage will be very easy to handle. Maintaining the session for the use and implementing that through connecting through the database has a few risk. This can be eliminated by normalizing the database. The other risk is to maintain the data for the room. If the data is more regarding the attributes it may result to data complexity this is overcome by getting the backup for a certain interval period. One of the most critical sources of risk to organizations today resides within their Web servers. This is because Web servers and applications open systems and information to be accessed by suppliers, partners, and customers. Performing a security risk assessment and implementing adequate security risk management policies in this area can be critical. Compromised Web servers can damage organizations in many ways, from surrendering customer privacy data and accepting fraudulent transactions to indirectly damaging corporate reputation as the result of a defaced homepage. While it may seem that a myriad of bad things can happen as the result of a million different vulnerabilities, we can succinctly categorize the core 'points of pain' to be addressed in your Web security risk management plan in a few primary areas:

* *Default configuration.* Web servers often are installed with default configurations that may not be secure. These insecurities include unnecessary samples and templates, administrative tools, and predictable locations of utilities used to manage servers. Without appropriate security risk management, this can lead to several types of attacks that allow hackers to gain complete control over the Web server.

* *User input validation.* Web sites and applications need to be interactive in order to be useful. However, Web applications that do not perform sufficient validation of user input screens allow hackers to directly attack the Web server and its sensitive databases. Invalid input leads to many of the most popular attacks. A thorough security risk assessment on your organization's internal and external Web applications can reveal what, if any, actions need to be taken.

* *Encryption.* It is a sad fact that although modern encryption algorithms are virtually unbreakable, they are underutilized. In years past, performance considerations were cited as a factor in limited

usage of encryption. However, today's high-performing CPUs and specialized cryptographic accelerators have broken down the price/performance barriers related to encryption. The issue with limited encryption has more to do with poor application design and a lack of awareness among developers. Nearly all Web traffic passes in the clear, and can be snooped by an alert hacker.

* *Secure data storage.* While it is critical to secure data in transit, it is just as important to implement security risk management policies that ensure that data is being stored securely. This includes encrypting data at rest, but it does not stop there. Many Web applications store sensitive files on publicly accessible servers, rather than on protected servers. Other applications do a poor job of cleaning up temporary files, leaving valuable data accessible to the hacker who knows how to find it.

* *Session management.* Another factor one should consider when developing a security risk management plan is that many Web applications do a poor job of managing unique user sessions. This can include using weak authentication methods, poor cookie management, failure to create session timeouts, and other session weaknesses. This often leads to session hijacking and other compromises of legitimate user identities. A security risk assessment can determine whether this is a potential problem for your organization.

* *Maintenance.* Failure to implement security risk management policies that keep Web servers updated with the latest vendor patches, as well as neglecting to perform continued testing of proprietary Web applications, creates additional risk. All of these major problems usually are the result of a lack of due care within the Web application development and maintenance processes. In organizations where security is not 'baked in' to both the business planning and application development processes, there can be an appalling lack of awareness of the need to incorporate security best practices from day one. This is a dangerous situation, and the results of the general lack of awareness about the risks associated with Web servers and applications are evident from the weekly headlines reporting stolen consumer and corporate information. The best way to avoid such disasters is to establish an on-going security risk management process that begins with quantifying the value of Web applications, as well as the data they manage, through a complete security risk assessment. Organizations then must continuously identify and mitigate the vulnerabilities and risks associated with those systems from the beginning and throughout their lifecycle: from development through production. This approach to security risk management—consistently performing a security risk assessment, then identifying and remedying vulnerabilities by correcting application development errors, applying security patches, and fixing system misconfigurations—will lead organizations to continuous improvement of their business-technology infrastructure and a thorough reduction of risk.

3 Project Requirements

This project has two types of requirements. They are Hardware and Software requirements.

Hardware requirements are

A, i3 processor or i5 or i7

B, 4GB RAM and above.

C, SSD 250GB.

Software requirements are

A, Windows operating system,

B, JDK and JRE 1.8

C, Apache Tomcat Server 8.0.

D, MySQL Database,

E, Eclipse or net beans IDE for development,

F, MySQL workbench

3.1 Operations, Administration, Maintenance and Provisioning (OAM&P)

When the new system is proposed according to the requirements we have to types of users, one is the Admin and the second will be the Normal User. Each has different access codes they are ADM and USR respectively. Maintenance of the database of the project is done by the Admin. The admin will have the privileges to add a normal user, to add a new room and to edit the room details. The normal user has the only right to check and maintain the software. The fraud detection is easily done by maintaining and comparing the details of the customer in the Database. We will implement the recovery plan in further steps.

3.2 Security and Fraud Prevention

The Security of the developed system is maintained by providing a User ID and Password for all the users. The Admin Level User can't have access, change the values of the attributes for a certain room. The admin authorizes the Normal User of the application so that it increases the security and no other than the authorized users can use the application. Using spring framework will also increase the security to the system.

3.3 Release and Transition Plan

The execution of the program is shown on the local host on the personal computer and for the further transition plan the complete project will be deployed in the cloud by using the Amazon Web Services which will provide a free account for the students for nine months.

4 Project Design Description

After the requirement collect phase when we started the designing phase we have divided the project into three parts like developing the Front End pages using Html and CSS. These pages are designed responsive so that the page will display accordingly to the size of the screen. The second part of the development phase is designing the Database of the project. We have Used My SQL Work Bench to develop the database. In the database we can store the user's data, Manager Data, attributes data, access permissions, types of attributes and the sample values for the attributes. Designing the front-end, we have designed six responsive web pages which also have the validations created. These pages are linked with each other, so that the user can be redirected from one page to another by clicking the respective Button. In the database we are writing Stored

Procedures so that the duplication of database code is reduced and increases the performance of the queries. The following is a diagram which shows the design of the application.

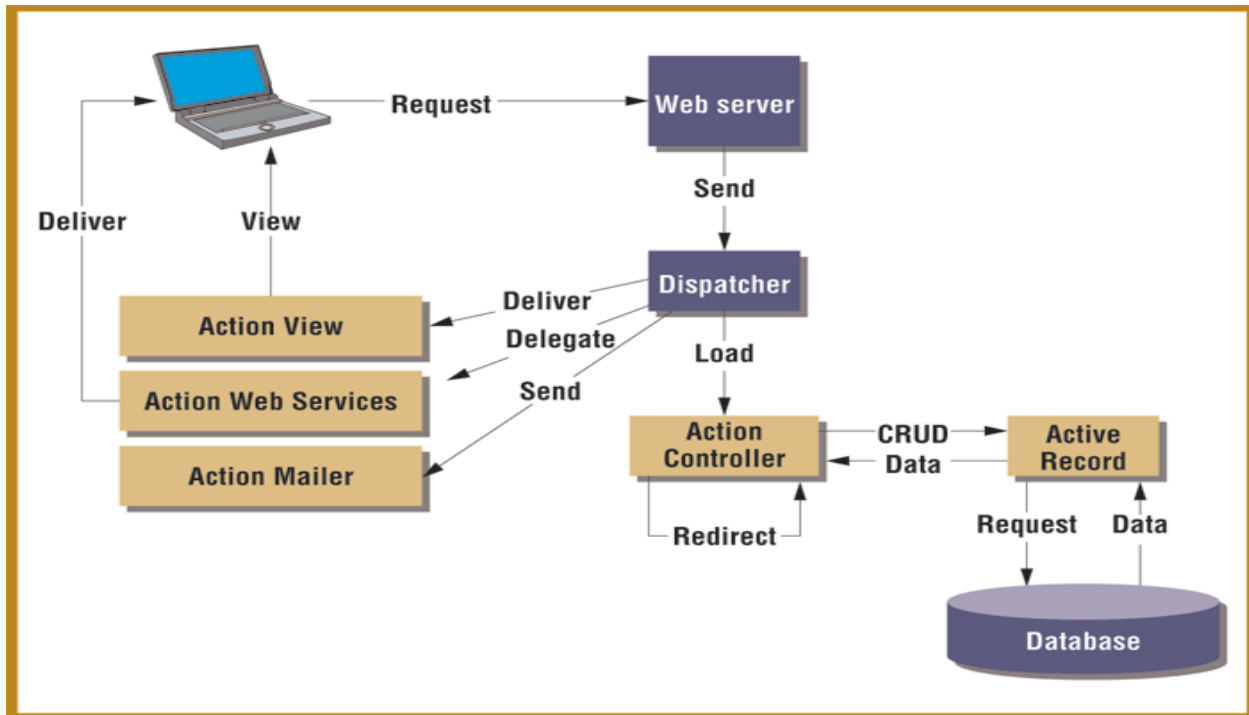


FIGURE 3 BASE DESIGN

After the designing of both front end and database is done both are connected using the spring and hibernate frame works. The spring framework will control the program like where to get redirected and also to provide web services. While the Hibernate helps in converting the database queries to the java objects. Eclipse is the tool used. Here are the following tables which shows what files does the project contain and which part the files belong to. DAO is an abbreviation for Data Access Object, so it should encapsulate the logic for retrieving, saving and updating data in your data storage (a database, a file-system, whatever). The MVC is a wider pattern. Data Access Object Pattern or DAO pattern is used to separate low level data accessing API or operations from high level business services. Following are the participants in Data Access Object Pattern. The DTO/DAO would be your model in the MVC pattern. It tells you how to organize the whole application, not just the part responsible for data retrieval. if you want to follow the MVC pattern it would be better to have a separate controller, which would contain the business logic for your frame in a separate class and dispatch messages to this controller from the event handlers. This is our proposed design for the project.

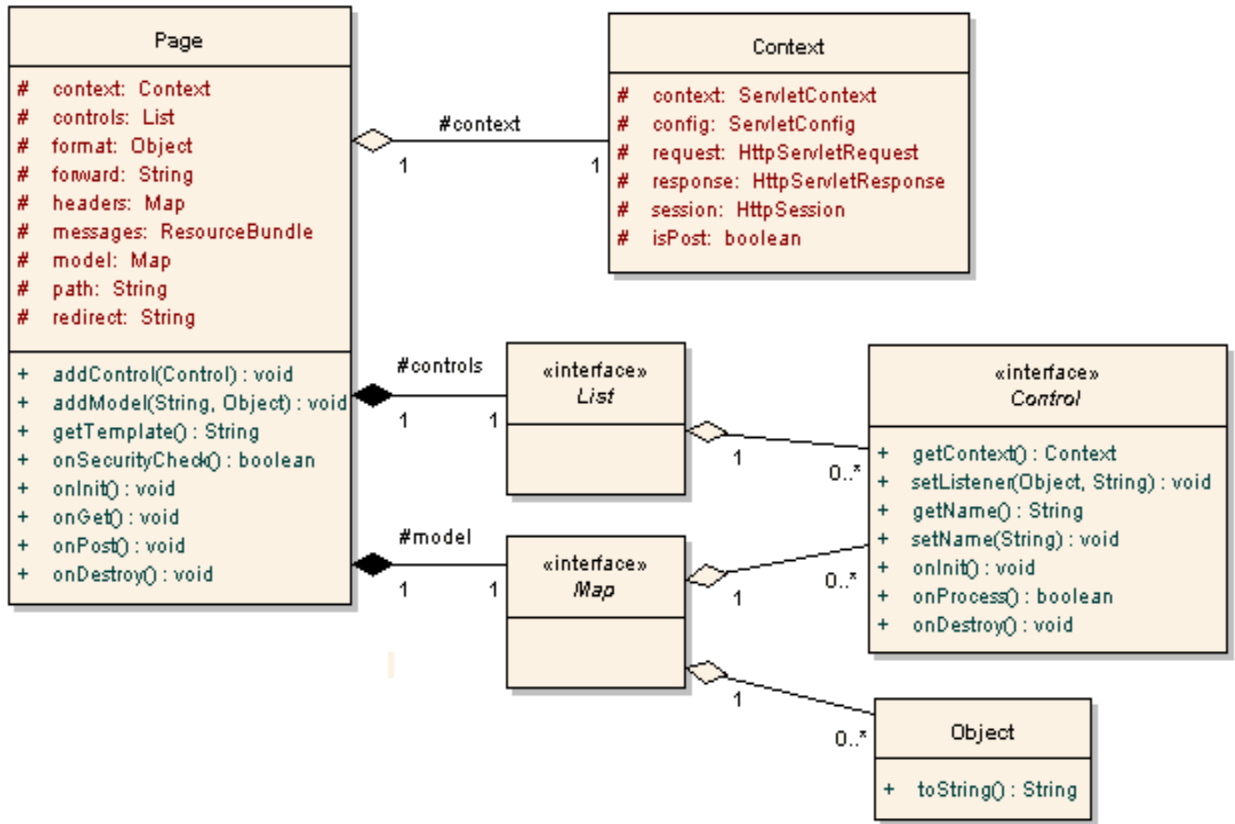


FIGURE 4 FLOW OF CONTROL STRUCTURES

5 Project Internal/external Interface Impacts and Specification

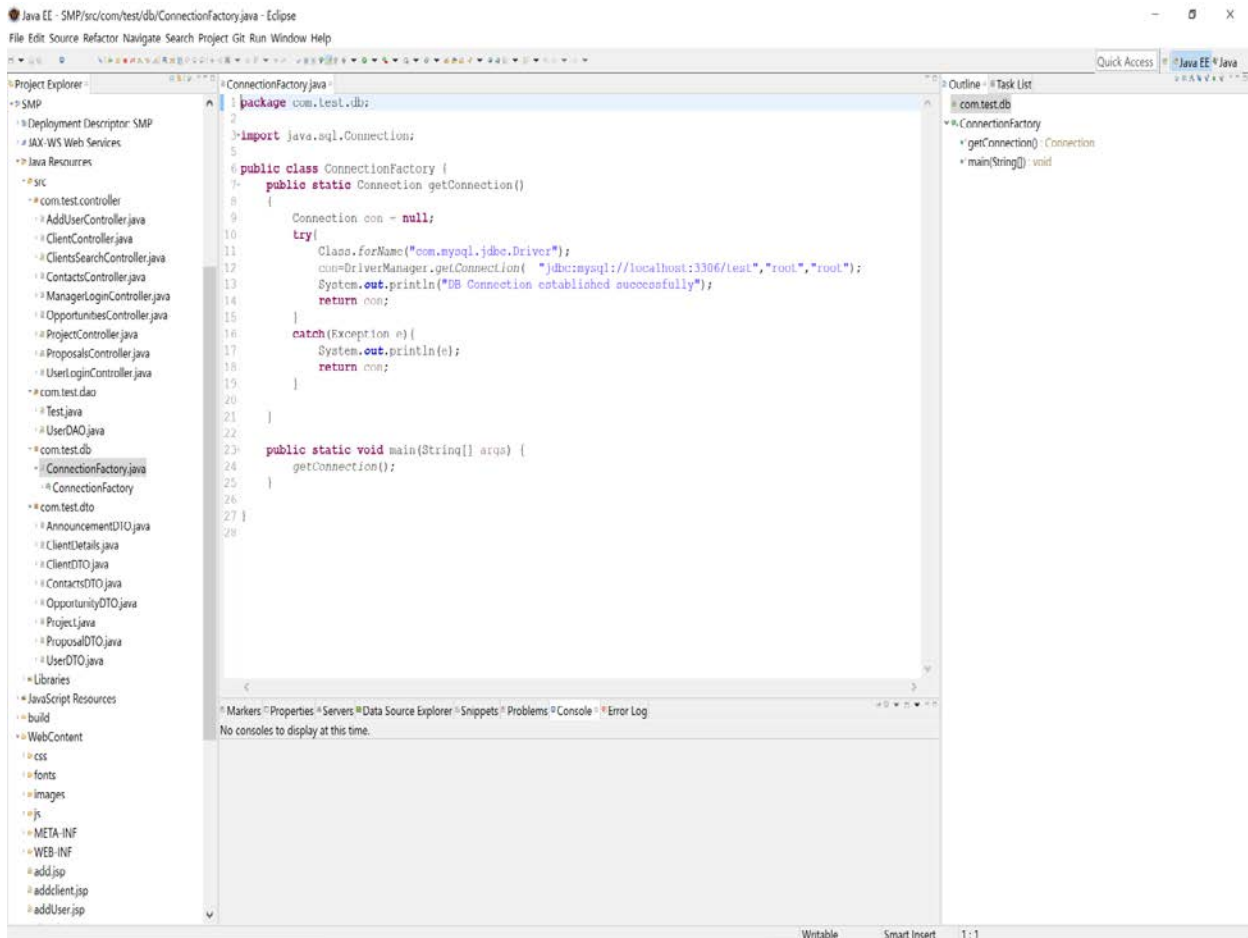
“An interface is a boundary where, or across which, two or more parts interact.” Another definition is: “An interface is that design feature of a piece of equipment that affects or is affected by a design feature of another system.”

The key words here are “interact” and “affects or is affected by another system”. From a requirements standpoint, any time the wording of a requirement indicates or implies one of these conditions, there is an interface involved. If there is an interface involved, then the requirement dealing with this interface is classified as an interface requirement.

Using SDLC, the roles are clearly assigned to the participants of the project which saves the time and increases the efficiency of the system. The system can be maintained well and modularity is achieved. For complex projects, SDLC helps to reach the goals in specified time. This layer comprises of the Database Components such as DB Files, Tables, Views, etc. The Actual database in our project is created using MySQL Server. In an n-tier application, the entire application is implemented in such a way that it is independent of the actual Database. For instance, we change the Database Location with minimal changes to Data Access Layer. The rest of the Application should remain unaffected.

6 Project Design Units Impacts

The project we develop will have only one design unit which consists of the three sub-unites. The standard architecture for project design is MVC architecture. MVC stands for Model View Controller. MVC is a standard architectural design pattern for developing World Wide Web apps. The central idea behind MVC is code reusability and separation of roles of code development. First DB design is followed by Model components, next View components to collect input and to display response and finally Controller components to integrate View and Model components.



```
package com.test.db;

import java.sql.Connection;

public class ConnectionFactory {
    public static Connection getConnection()
    {
        Connection con = null;
        try{
            Class.forName("com.mysql.jdbc.Driver");
            con=DriverManager.getConnection( "jdbc:mysql://localhost:3306/test","root","root");
            System.out.println("DB Connection established successfully");
            return con;
        }
        catch(Exception e){
            System.out.println(e);
            return con;
        }
    }

    public static void main(String[] args) {
        getConnection();
    }
}
```

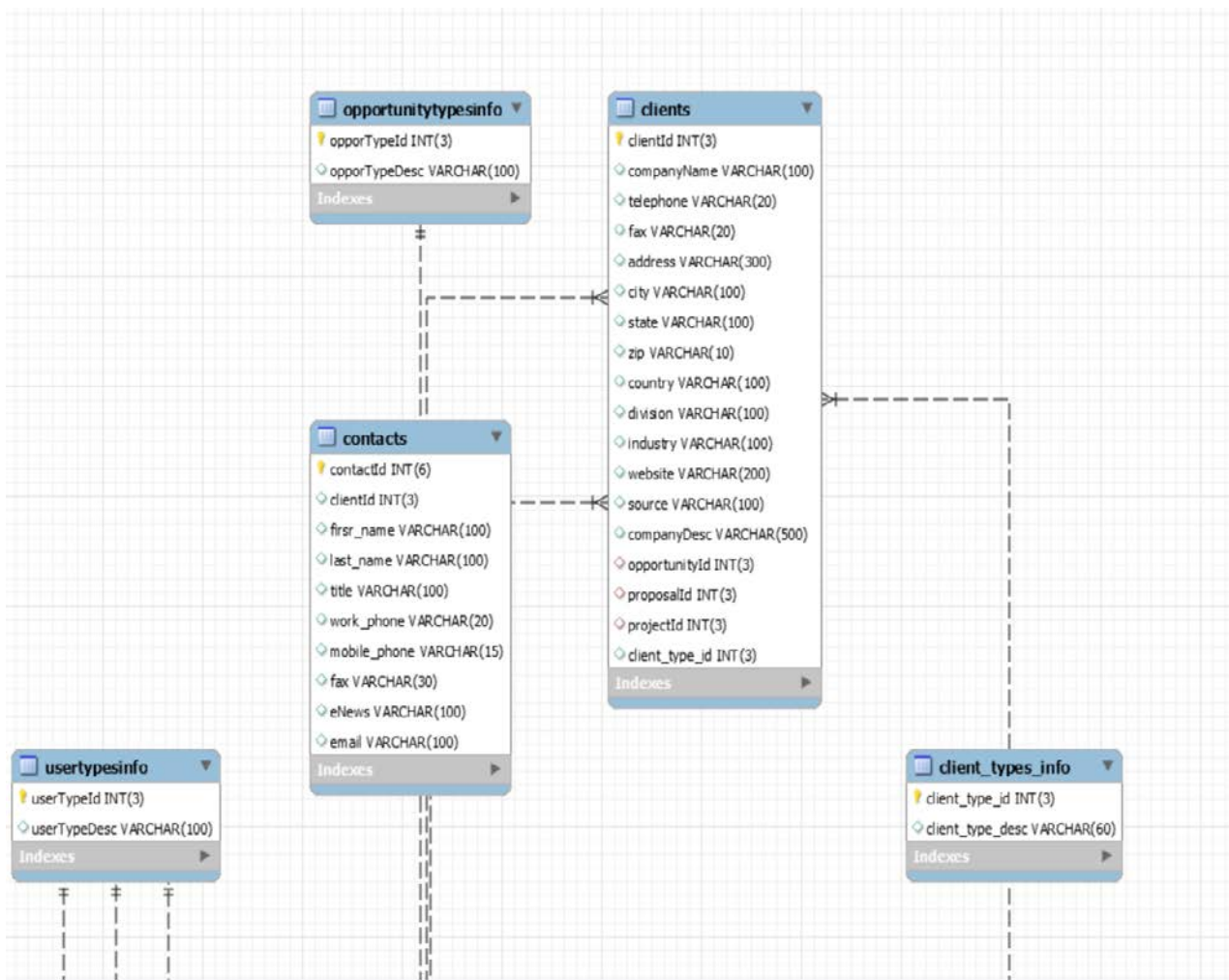
FIGURE 5 CONNECTION FACTORY CODE FIGURE.

DTO is an abbreviation for Data Transfer Object, so it is used to transfer the data between classes and modules of your application. DTO should only contain private fields for your data, getters, setters and constructors. It is not recommended to add business logic methods to such classes, but it is OK to add some java utility methods.

There are tools which can generate the DAO, DTO and Factory class code for you. These tools are helpful because they automate the generation of DAO classes. The importance of these tools even becomes great in the initial phase of application development because the database schema keeps

of changes frequently and hence manually making the changes in database tables and the DAO code.

- The DAO has been declared dead because you might just as well invoke the Entity Manager directly. It provides a nice enough interface and switching from JPA to a different persistence implementation is not something the DAO abstraction would make much easier.
- DTO's have been deemed superfluous because you can also use your domain objects directly in the presentation layer. This is made possible by a combination of the *open Entity Manager in view* pattern, tag libraries to display your domain objects in JSP's and data-binding utilities to map HTTP request parameters back to domain objects.
- And finally Service Facades also seem to have gone out of fashion. Instead you can have the controller directly invoke the services it needs or, even simpler, directly contain the business logic.



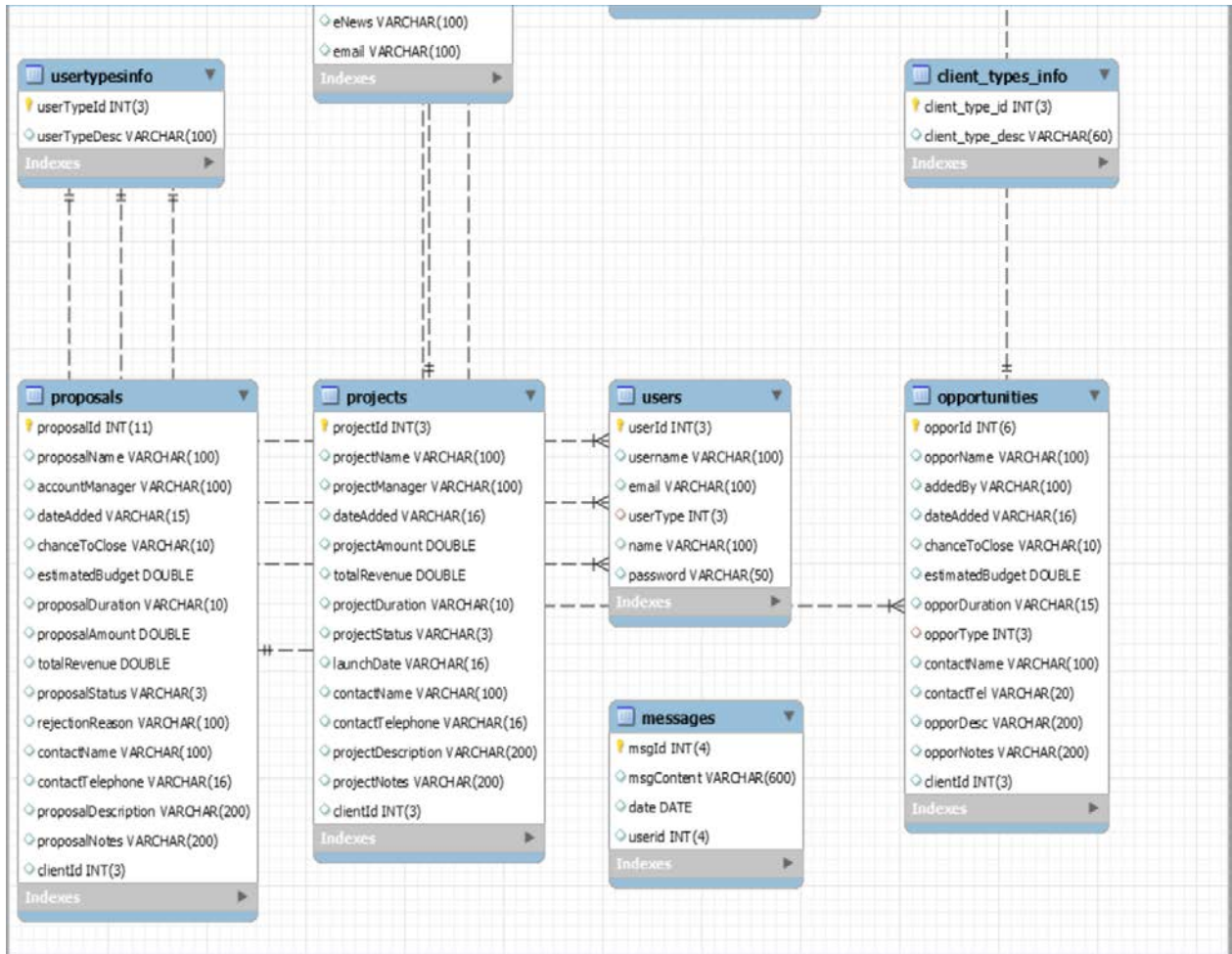


FIGURE 6 DATA MODEL DIAGRAM.

They are View, Controller and model. This design totally covers all the requirements and fulfills the project. The View will have all the front end files which will be developed by using the HTML and CSS. We are doing the responsive page designing using Bootstrap. The controller will contain the connection files. The connection from the front end and database is done by using the spring and Hibernate framework. Model will contain the database developed by using My SQL workbench tool. The database is totally normalized. The following are the Database screen shots.

6.1 Functional Area A/Design Unit A

6.1.1 Functional Overview

The three main parts for the development phase are present in this design unit. These impacts are considered essential and the requirements are written accordingly. What is a DAO and how to create a DAO? There are tools which can generate DAO code for an application. But one can also write DAO classes manually. Also one more design pattern very closely related to DAO is the DTO (Data Access Object). A DAO design pattern helps an application to perform various CRUD operations on the database. The DAO classes provide methods for insertion, deletion, updating and finder methods. The basic purpose of creating DAO is the loose coupling and non-repeating of code.

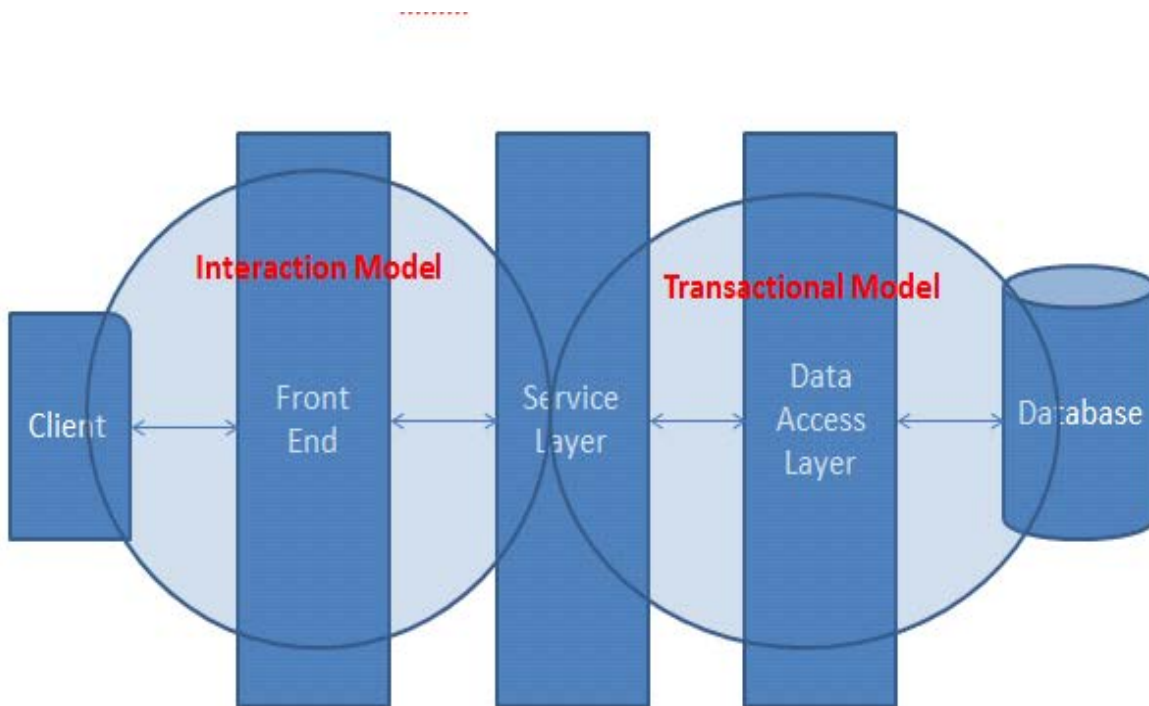
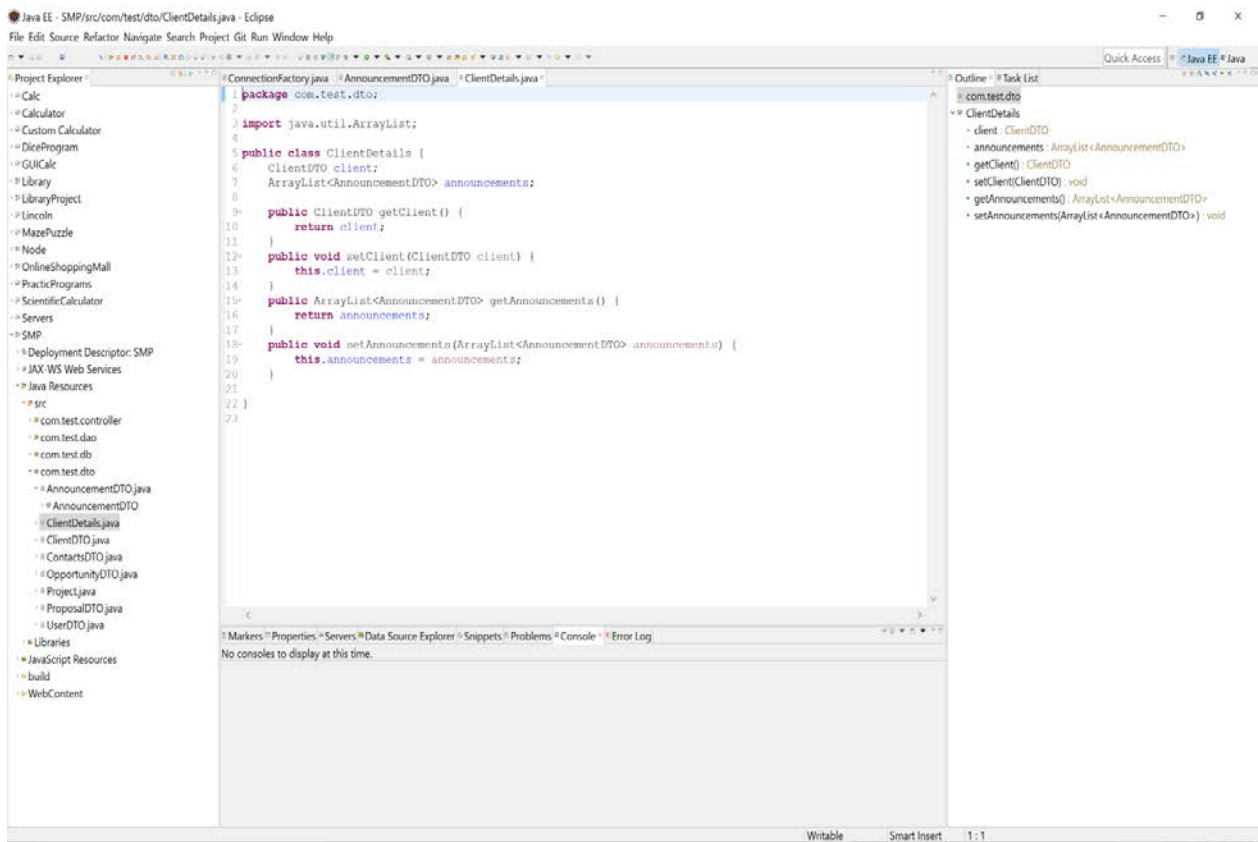
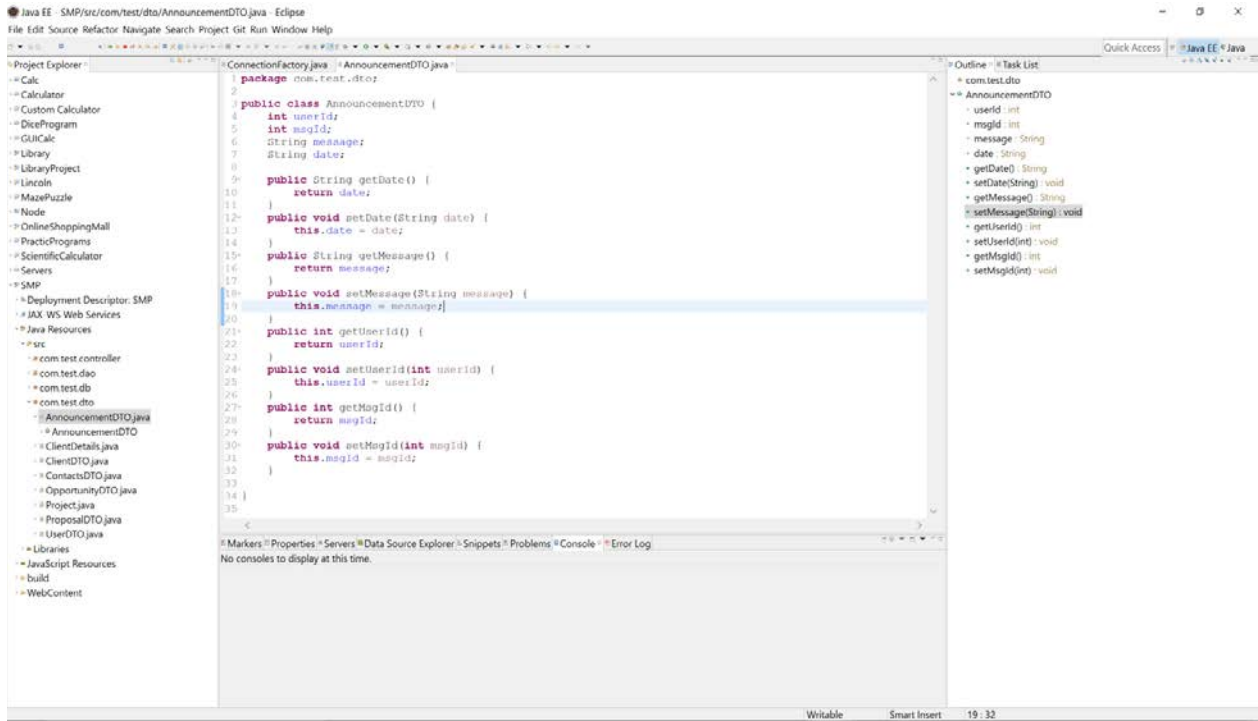


FIGURE 7 INTERACTION MODEL & DAO MODEL

In any application which is going to interact with the database, there is need of performing CRUD operations on the database tables and since the table operations may be done by different classes and hence it becomes cumbersome to repeat the same code in various classes. Moreover, even after repeating the code, it becomes difficult to maintain the database interaction code whenever changes are required in the way database interaction is being done. To overcome the above mentioned problems, the DAO design pattern was invented. The DAO pattern is supposed to have a DAO interface, class and factory corresponding to each table in the table.

6.1.2 Impacts

The complete project depends on this part because we have only one design unit. This design unit consists of the parts where we create a database and a front end application and connect both of them by using the controllers. Connecting the front end and the database will be a challenging task in this design. We have created lot of POJO classes for Data Transaction. Some of them are in the below Screenshots.



Java EE - SMP/src/com/test/dto/UserDTO.java - Eclipse

File Edit Source Refactor Navigate Search Project Git Run Window Help

```

package com.test.dto;

public class UserDTO {
    int userId;
    String username;
    String email;
    int userType;
    @Override
    public String toString() {
        return "UserDTO [userId=" + userId + ", username=" + username
            + ", email=" + email + ", userType=" + userType + ", name="
            + name + ", password=" + password + "];"
    }
    String name;
    String password;

    public int getUserId() {
        return userId;
    }
    public void setUserId(int userId) {
        this.userId = userId;
    }
    public String getUsername() {
        return username;
    }
    public void setUsername(String username) {
        this.username = username;
    }
    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }
    public int getUserType() {

```

Outline: Task List

- com.test.dto
- UserDTO
 - userId: int
 - username: String
 - email: String
 - userType: int
 - toString(): String
 - name: String
 - password: String
 - getUserId(): int
 - setUserId(int): void
 - getUsername(): String
 - setUsername(String): void
 - getEmail(): String
 - setEmail(String): void
 - getUserType(): int
 - setUserType(int): void
 - getName(): String
 - setName(String): void
 - getPassword(): String
 - setPassword(String): void

Markers: Properties Servers Data Source Explorer Snippets Problems Console Error Log

No consoles to display at this time.

Writable Smart Insert 1:1

Java EE - SMP/src/com/test/dto/Project.java - Eclipse

File Edit Source Refactor Navigate Search Project Git Run Window Help

```

package com.test.dto;

public class Project {
    int projectId;
    int clientId;
    String projectName;
    String projectMgr;
    String addedDate;
    double projectAmount;
    double totalRevenue;
    String projectDuration;
    String projectStatus;
    String launchDate;
    String contactName;
    String contactTel;
    String projectDesc;
    String projectNotes;

    public int getClientId() {
        return clientId;
    }
    public void setClientId(int clientId) {
        this.clientId = clientId;
    }
    public int getProjectId() {
        return projectId;
    }
    public void setProjectId(int projectId) {
        this.projectId = projectId;
    }
    public String getProjectName() {
        return projectName;
    }
    public void setProjectName(String projectName) {
        this.projectName = projectName;
    }

```

Outline: Task List

- com.test.dto
- Project
 - projectId: int
 - clientId: int
 - projectName: String
 - projectMgr: String
 - addedDate: String
 - projectAmount: double
 - totalRevenue: double
 - projectDuration: String
 - projectStatus: String
 - launchDate: String
 - contactName: String
 - contactTel: String
 - projectDesc: String
 - projectNotes: String
 - getClientId(): int
 - setClientId(int): void
 - getProjectId(): int
 - setProjectId(int): void
 - getProjectName(): String
 - setProjectName(String): void
 - getProjectMgr(): String
 - setProjectMgr(String): void
 - getAddedDate(): String
 - setAddedDate(String): void
 - getProjectAmount(): double
 - setProjectAmount(double): void
 - getTotalRevenue(): double
 - setTotalRevenue(double): void
 - getProjectDuration(): String
 - setProjectDuration(String): void
 - getProjectStatus(): String
 - setProjectStatus(String): void
 - getLaunchDate(): String
 - setLaunchDate(String): void
 - getContactName(): String
 - setContactName(String): void
 - getContactTel(): String
 - setContactTel(String): void
 - getProjectDesc(): String
 - setProjectDesc(String): void
 - getProjectNotes(): String

Markers: Properties Servers Data Source Explorer Snippets Problems Console Error Log

No consoles to display at this time.

Writable Smart Insert 1:1

6.1.3 Requirements

During testing, the implementation is tested against the requirements to make sure that the product is actually solving the needs addressed and gathered during the requirements phase. Unit tests and system/acceptance tests are done during this phase. Unit tests act on a specific component of the system, while system tests act on the system as a whole.

Each stage produces a set of deliverables needed in the next stage. Requirements are analysed, and the functional requirements are determined, based on the functional requirements needed, the system is designed and code is produced during implementation phase. Testing phase verifies the deliverables from the implementation phase meet the functional requirements or not.

DAO4j tools do is import the tables from the database by asking the connection details and then generate the DAO/DTO classes based on the tables being imported. One more important point that is useful when using these tools is that they keep a check on the primary and foreign keys of the table structure. Moreover, the database data types are converted into corresponding data types in Java in the DAO/DTO classes. After the code has been generated by these tools, one can always modify the code to optimize the DAO code.

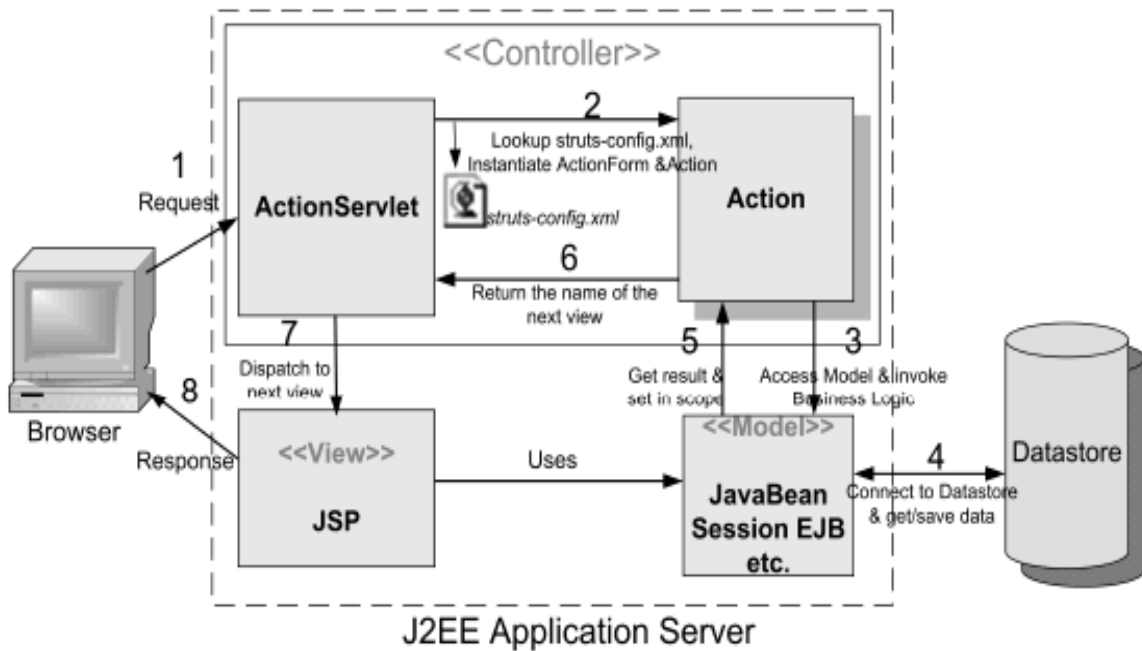
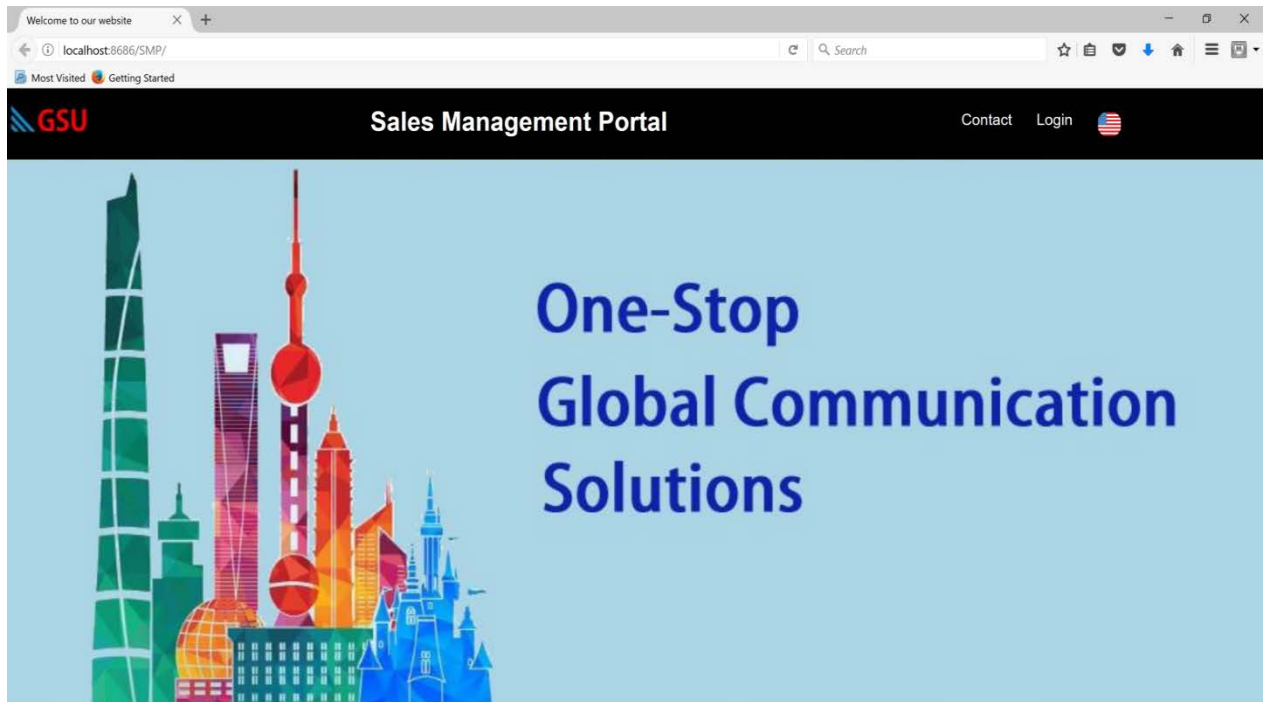


FIGURE 8 APPLICATION SERVER FLOW CONTROL

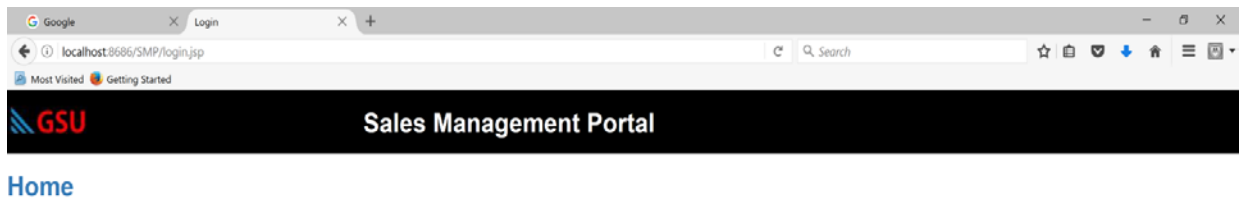
Always remember that the DAO, DTO, Factory, Loose Coupling (A SOLID principle), Factory design patterns all go along. One can also create a connection pool while using the DAO design pattern. The important point is that the code becomes clean and easy to understand when using the DAO, DTO design patterns.

6.2 Screenshots

Home Page:



Manager Login:



Company Details:

Client Information - Indian Arts & Culture

General Info | Contacts | Notes | Opportunities | Proposal | Projects

Company Name: Bank Of America

Telephone: 708-876-5432

Fax: 708-961-7654

Address: PualoAlto

City: PauloAlto

State: California

Zip Code: 95131

Country: USA

Divison: BigData

Type: Client

Industry: Software

Web Site: bofa.com

Source:

Company Description: Banking Company

Next

Contacts:

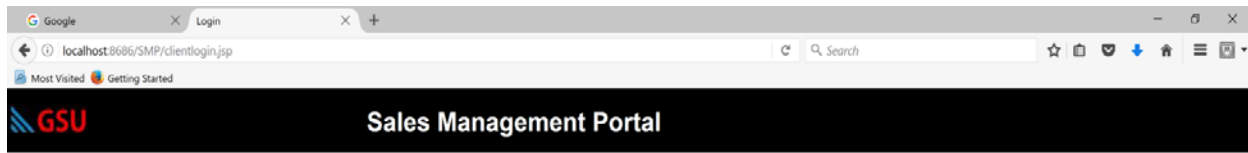
Client Information - Indian Arts & Culture

General Info | **Contacts** | Notes | Opportunities | Proposal | Projects

First Name	Last Name	Title	Work Phone	Mobile Phone
Nelson	Mandela	Manager	708-876-5432	708-898-5463

Next

User Login:



Home

User Login

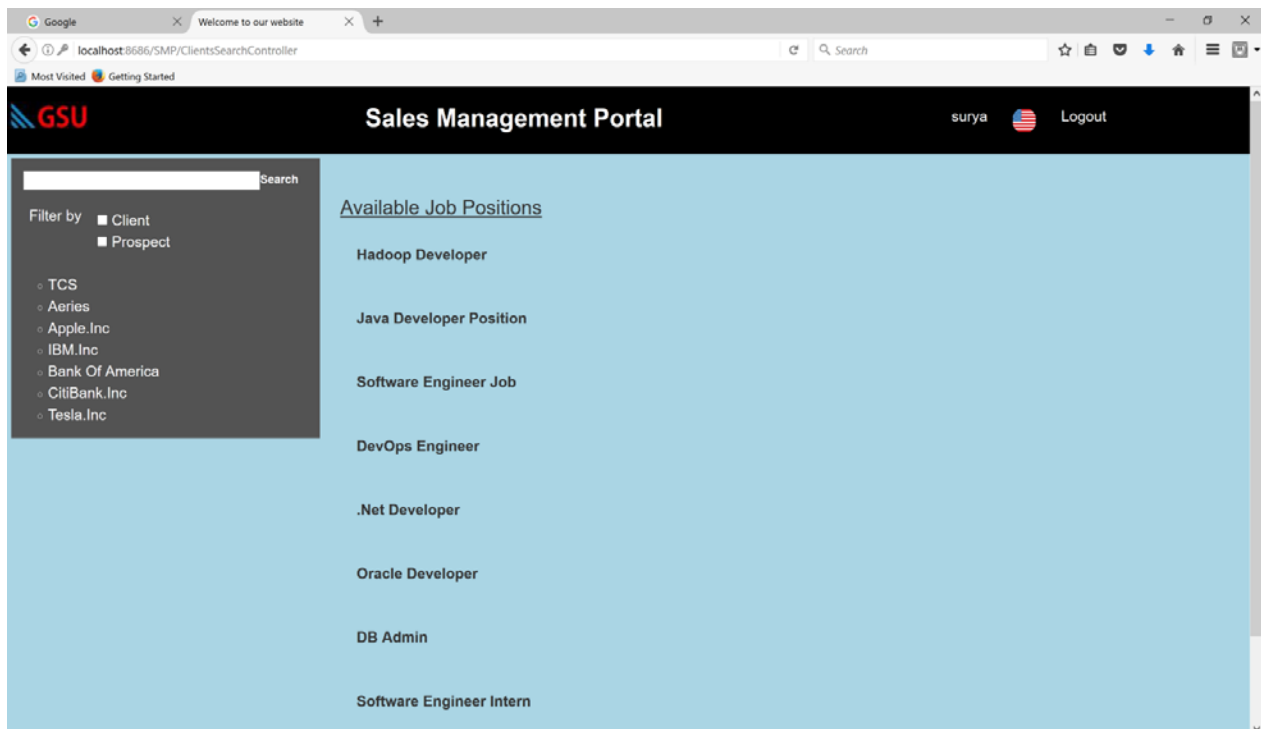
Username

Password

[Click here for Manager Login](#)

localhost:8686/SMP/login.jsp

User Account:



Contact HR Page:

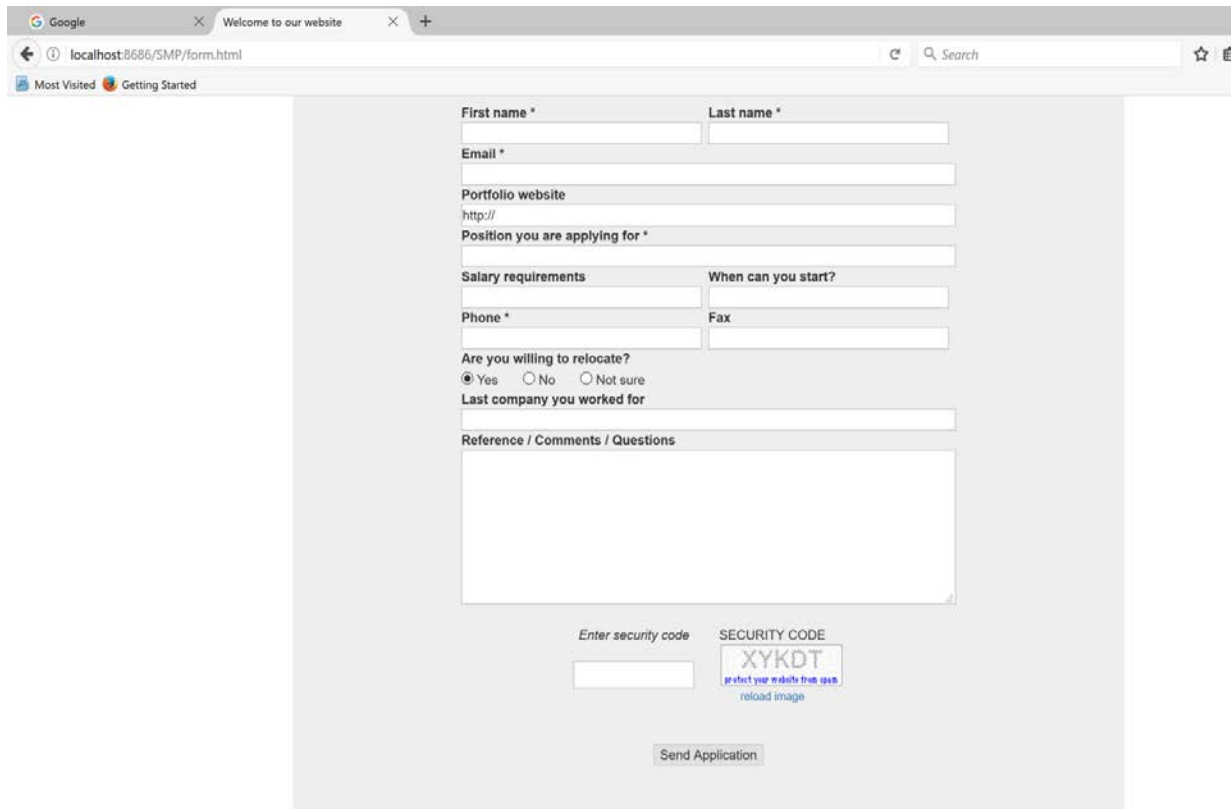
The screenshot shows a web browser window with the URL `localhost:8686/SMP/contacts.html`. The page title is "Sales Management Portal" and it includes a "Hi" greeting with a US flag icon. The main content area is titled "Client Information - Indian Arts & Culture" and features a tabbed interface with "Contacts" selected. A table lists contact details for three individuals: Joe Smith (VP), Surya Dev (HR), and Steve Jobs (CEO). A sidebar on the left contains a search bar and a list of client names under the heading "Add Client".

First Name	Last Name	Title	Work Phone	Mobile Phone	Fax	eNews	Email
Joe	Smith	VP	7091234567	6307654321	7087654321	Yes	jsmith@indianartsculture.org
Surya	Dev	HR	7083457896	6308989891	7087654321	Yes	surya@hr.apple.com
Steve	Jobs	CEO	7091888567	7089898989	7087654321	Yes	steve@ceo.apple.com

Employment form:

The screenshot shows the "Employment Application" form within the Sales Management Portal. The form includes fields for personal information, contact details, and application preferences. The "Are you willing to relocate?" section has radio buttons for "Yes", "No", and "Not sure", with "Yes" selected. A large text area at the bottom is labeled "Reference / Comments / Questions".

Capcha for Security:



The image shows a web browser window with a single tab titled "Welcome to our website". The address bar displays "localhost:8686/SMP/form.html". The page content is a registration form with the following fields and options:

- First name * (text input)
- Last name * (text input)
- Email * (text input)
- Portfolio website (text input, pre-filled with "http://")
- Position you are applying for * (text input)
- Salary requirements (text input)
- When can you start? (text input)
- Phone * (text input)
- Fax (text input)
- Are you willing to relocate? (radio buttons for Yes, No, Not sure; Yes is selected)
- Last company you worked for (text input)
- Reference / Comments / Questions (large text area)
- Enter security code (text input)
- SECURITY CODE (image showing "XYKDT" with "protect your mobile from spam" and "reload image" link)
- Send Application (button)

7 Acknowledgements

We agree this opportunity and thanking each and every person who has involved in this project and supported us in the way a completing this project successfully. We thank Dr. Soon Ok Park for guiding us throughout the project work. We have received an immense support from Dr. Soon Ok Park through her valuable advices, suggestions and correction in the process of completing our application.

We are also proud to our College of Arts and Science, Computer Science Department, how they are encouraging us in finding the new solutions for the existing system. Our team's courtesy to Governors State University, for accepting our project and providing the resources to complete it successfully. We thank our self as a team and we have achieved this success through complete team work and dedication toward our field.

8 References

All references should include, author, title of document, doc ID# and issue date.

[1]. Reviewed many Sales Management Portals:

<http://www.verifytechnologies.com/sales-management-portal/>

<http://www.beyondbonus.com.my/smp/>

[2]. Dalrymple, D.J., Cron, W.L. and DeCarlo, T.E. (2001) Sales Management, Tenth Edition, John Wiley & Sons, Inc.

[3]. McGoldrick, P.J. *Retail Marketing*. McGraw-Hill, 1998.

<https://www.perfectportal.com.au/why-perfect-portal/>

[4]. For jsp pages development referred from Javatpoint.

<http://www.javatpoint.com/jsp-tutorial>

[5]. Spring Framework – Overview, Jan 2016, Tutorials point, Retrieved from:

http://www.tutorialspoint.com/spring/spring_overview.htm

[6]. Hibernate ORM, Idiomatic persistence for Java and relational databases. LGPL V2.1, Retrieved from:

<http://hibernate.org/orm/>

[7]. Postman, Improve API workflow, blog, Post dot Technologies, retrieved from:

<http://blog.getpostman.com/>

[8]. Maven, Apache Maven Project, The Apache Software Foundation Retrieved from:

<https://maven.apache.org/what-is-maven.html>

[9]. Connection factory for JDBC connection and getting the jar files Retrieved from:

<https://dev.mysql.com/downloads/connector/j/>