eProject

Advertisement Agency application

with

Microsoft SQL 2005

**[Index]**

1. Problem Definition 4

1.1. Acknowledgements 4

1.2. eProject Synopsis 4

1.2.1. Background 4

1.2.2. Non-Financial 4

1.2.3. Financial 5

1.2.4. Functional Requirements: 5

1.3. eProject Analysis 6

1.3.1. Login feature 6

1.3.2. Common features 7

1.3.3. Definition of Administration User 7

1.3.4. Definition of cost parameters 8

1.3.5. Definition of Employee 9

1.3.6. Definition of error sequences 10

1.3.7. Definition of reaming 11

2. Entity Relationship (ER) Diagram 12

2.1. Database Design/Structure 13

3. Architecture and Design in the Project 15

4. User Case 16

4.1. Administrator 16

4.2. Employee 16

5. Data Flow Diagram (DFD) 17

6. Flowchart 19

6.1. Booting up 19

6.2. Login 20

6.3. editCustomerOK 21

7. GUI Standards Documents 22

8. Interface Design Documents 23

9. Customer Requirement Specification (CRS) 24

9.1. Client: 24

9.2. Business/Project Objective: 24

9.3. Input/Output: 24

9.4. Hardware Requirements: 24

9.5. Operating System: 24

9.6. Software Requirement: 24

10. Task 25

# Problem Definition

## Acknowledgements

This document describes the information regarding our product designs, database structures, flow charts, and so on.

## eProject Synopsis

### Background

Our client is the local news paper editor. He ensures distribution of news papers all over various localities. The newspaper covers a variety of news like the weather forecasts, political, cultural, social, and entertainment related issues,

They have also introduced a section called CZ Classifieds , which will be handled and taken care of by the Advertising Agency Department personnel.. In this section they place several advertisements, based on their customer requirements.

### Non-Financial

1. The application should be a server specific application, and it should provide a

multi user access.

2. To easily manage the database and to reduce any discrepancies the client wants the following to be implemented while building the application.

The database will be set up on the server

The People of the agency department should be able to connect to the

database, which is set up on the server from their respective systems.

For example, if there are 4 people in the department, say A , B ,

C , and D , each individual of the department is provided with the

system, like say W , X , Y and Z , so that each will access their

own system for their work. Now all these systems must connect to the

single database that will be set up on the server.

For placing an order for the advertisement, the customer needs to visit

the office and explain the requirements of the advertisement for placing the advertisement.

As soon as a customer places an order, one should be able to record

his details like Name, Payment Done, Preference (like length, size,

and on which page the advertisement is to be displayed, Date

and time of advertisement, etc.)

3. Once the payment is received the details should be recorded in their database

(available on the Server), So that there will be no possibility of Data

Mismatch or Data misplacement, and there can be automation in the process.

4. He should be able to access the database to enable him to view the records, update the records, and delete the records. Also, he l alone will decide the price for(fee/cost of) the advertisement based on the parameters and insert this into the database.

5. Once the requirement of the customer is inputted in the application based on

the parameters, the application will automatically calculate the charges for the

advertisement.

6. The admin will alone have the right to create and delete the users of the

application.

### Financial

The customers will be charged based on the length, size and type of the

advertisement, and also based on the page preferred (e.g. the front page, second

page, etc.). The details of the price will be decided by the Manager (Editor) of the news paper.

Once the details of the parameters chosen by the customer are entered, the

application would automatically display the total charges for placing the

advertisement.

### Functional Requirements:

1. The application should contain the database, which should be server specific.

2. The application should contain two logins -

Admin

Employee

3. Admin will have the following functionality:

**Creating, editing and deleting the user profiles**

The admin can only have the authority of deciding the cost based on

the parameters like length, size, type of the advertisement and based

on the page on which the advertisement is to be displayed.

4. The employee will login in to his account and then he can place the order for

the customer on the application and can finalize the advertisement.

5. The application would calculate the total charges automatically based on the

parameters provided.

## eProject Analysis

### Login feature

Firstly, we have focused on the multi-user feature which is based on two types of users: One is Admin and another is Employee. The application which is called “adagent” has a login feature: In the login feature, users can select a login dialog box for inputting user name and password from the menu bar. In this case, the menu bas has at least two menu items.

|  |
| --- |
| File |
| Login |
| Logout |

**Figure.1 Menu Bar prototype**

And we have decided a design for the login dialog box.

Login

Password

Textbox1

Textbox2

OK

**Figure.2 Login dialog**

We have decided to use Microsoft SQL 2005 and type 4 JDBC driver of SQL 2005 to access database servers.

Note: In our understanding, there are no requirements for application settings corresponding to database servers in the specification; we have decided to write how to configure applications settings for database server in an install manual. In these cases, though we should create the application installers in the real world. We have decided to skip the step at this time, because it is not the main requirement.

### Common features

We have decided to equip a “refresh” button, a “save” button and an accessing progress bar in the status bar.

Save

Refresh

**Figure.3 Status panel**

1. To save the finalized data in the database server, click on the “save” button.
2. To revert to the original data, click on the “refresh” button.
3. While reverting to data, the progress bar indicates progress.

Kawai has found a custom database desktop application template in NetBeans IDE. We have decided to use the template, because of which the application would involve a mechanism to take and obtain data from database servers, as also to bind swing components with the data obtained.

### Definition of Administration User

1. Administration user name is fixed as “admin”, password as “123”.
2. Employee user management (Admin shall change an employee user table to add, delete, and, rename the users.)
   1. To elaborate:
   2. **To add**: Click on “new” button. The new name will be displayed and the original will be hidden.
   3. **To revert to the original data**, click on “refresh” button.
   4. **To save the finalized date**, click on “save” button.
3. Administration Order Table
   1. Provisions made:
   2. The admin can access the order table by clicking the user name and the password.
   3. **Only the admin has the privilege** of editing, deleting or modifying any information contained in the order table. Here again, clicking on the “refresh” button will display the original information and clicking on “save” button will store the finalized information.
   4. Clicking on any selected client order will display an “Edit Order” dialog box which will show the customer requirements noted by the employee.

Employee User Table

|  |  |
| --- | --- |
| User Name | Password |
|  |  |

Order Table

|  |  |  |  |
| --- | --- | --- | --- |
| Customer Name | Cost Parameter | Posting  Date | Bill |
|  |  |  |  |

Delete

New

Delete

Edit

**Figure.4 User settings panel**

We have decided to fix the Administration user name as “admin”, password as “123”. The custom database desktop application has a “new” button and a “delete” button as default for each table. We have decided to use the “new” button and the “delete” button for Employee User Table. In the Admin order table, we have decided to change from the “new” button to an edit” button, because employee users can add new orders received from customers, whereas the admin user only shall modify or delete the finalized data for avoiding inconsistency, as and when required..

### Definition of cost parameters

We had to decide the detail parameters for Advertisement. We have decided the four parameters mentioned below: content type, page type, size type, and posting date from number 2 in chapter 1.2.2

Content Type

|  |  |
| --- | --- |
| Type name | Description |
| Text | Text advertisement |
| Graphics | Graphical advertisement |

Page Type

|  |  |
| --- | --- |
| Type name | Description |
| Front | Front Page |
| Second | Second Page |
| Middle | Middle Page |
| Last | Last Page |

Size Type

|  |  |
| --- | --- |
| Type name | Description |
| Large | Maximum size |
| Medium | Medium size |
| Small | Mini size |

The above principle is applicable even to the unit fee columns. To elaborate:

any change in the unit fee will hide the original unit fee. Again on clicking the “refresh” button, the original hidden information will be displayed.

In addition we have decided a calculation formula for calculating bills:

**Bill = (Content type fee + Page type fee) \* Size type multiplex**

This formula is fixed by our application. Admin shall change the advertisement costs based on cost items. The panel shown below is a cost items panel.

Content type fee

|  |  |
| --- | --- |
| Text | 100.00 |
| Graphics | 200.00 |

Page type fee

|  |  |
| --- | --- |
| Front | 500.00 |
| Second | 200.00 |
| Middle | 100.00 |
| Last | 300.00 |

Size type multiplex

|  |  |
| --- | --- |
| Large | 10.0 |
| Medium | 5.0 |
| Small | 1.0 |

**Figure.5 Cost items panel**

Admin can only access the above “User settings panel” and “Cost items panel”

### Definition of Employee

Finally, we have decided the features of employee user. Employee users can add a new advertisement order to the order table.

Order Table

|  |  |  |  |
| --- | --- | --- | --- |
| Customer Name | Cost Parameter | Posting  Date | Bill |
|  |  |  |  |

New

**Figure.6 Employee order panel**

1. Employee Order Table   
   Provisions made:
   1. The employee has access to the order table to which he can add requisite information regarding new customers.
   2. The employee needs to click the “refresh” button to revert to the original data and click the “save” button to store the finalized data.
   3. If the employee enters the relevant data regarding size, content type, page type etc., the respective/corresponding unit fee will be automatically calculated and inserted in the order table.

Employee should save the finalized data after payment has been received. To add customer order details correctly, we have decided a new order dialog box, shown below..

Posting date

Textbox2

OK

Size

Combo3 box

Combo2 box

Combo1 box

Textbox1

Content

Page

Customer name

**Figure.7 Order dialog box**

The above dialog box would be used by admin to edit the finalized data.

### Definition of error sequences

We recognized the error sequences after reviewing the information mentioned above.

1. If two admin users login from different places at the same time, persistent date will become inconsistent.
2. There are possibilities for losing the connection of database server, during operations time.
3. There are possibilities of failure to login.

In this case, we have decided not to implement countermeasure mechanisms for 1) and 2), because implementing the countermeasure of 1) will become complicated in the login process. Indeed, we should exclude the other admin login, and we should separate the user in database software for admin users with normal users for employee users. We have decided to assume that only one authorized person knows admin user name and password. Furthermore, if the application is used by intranet, case umber 2) would be rare. Probably, we do not need to consider it. To be honest, these specifications in error sequences are very important in the real world. In case of real products, we should carefully check these detail specifications with clients. Kawai believes that we cannot satisfy client requirements, if designers neglect these aspects. At this juncture, we have decided to consider only the countermeasure No 3.

Login failure

OK

Invalid user name and password

**Figure.8 login failure alert box**

### Definition of reaming

We have decided to use the tab bar for switching the panel, because it is easy to add and remove the panels on login/logout. We have decided that the menu bar has “help” and “exit” to exit application and display copyright.

|  |  |
| --- | --- |
| File | Help |
| Login | About |
| Logout |  |
| Exit |  |

**Figure.9 Definition of menu bar**

**I would like to know your satisfaction level for these specifications.**

# Entity Relationship (ER) Diagram

We decided the ER Diagram from chapter 1.3. Each table has a primary key to avoid inconsistency.

|  |  |
| --- | --- |
| **Table : tContType** | |
| ***Column Name*** | ***Key*** |
| cID | PK |
| ContentName | - |
| ContentCost | - |

|  |  |
| --- | --- |
| **Table : tEmployee** | |
| ***Column Name*** | ***Key*** |
| eID | PK |
| eName | - |
| ePassword | - |

|  |  |
| --- | --- |
| **Table : tPageType** | |
| ***Column Name*** | ***Key*** |
| pID | PK |
| PageName | - |
| PageCost | - |

|  |  |
| --- | --- |
| **Table :tOrder** | |
| oId | PK |
| eID | FK |
| cName | - |
| aCont | FK |
| aSize | FK |
| aPage | FK |
| aBill | - |
| aPostingDate | - |

|  |  |
| --- | --- |
| **Table : tSizeType** | |
| ***Column Name*** | ***Key*** |
| sID | PK |
| SizeName | - |
| SizeWeight | - |

**Figure 2.1 ER Diagram**

## Database Design/Structure

The table structure in the database is given.

tOrder table containss all order data from customers

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table : tOrder** | | | | | |
| ***Column***  ***Name*** | ***Data Type*** | ***Max Digits*** | ***Key*** | ***Null*** | ***Description*** |
| oID | integer | - | PK | NO | Identity(1,1) |
| eID | integer | - | FK | NO | tEmployee |
| aCont | smallint | - | FK | NO | tContType |
| aPage | smallint | - | FK | NO | tPageType |
| aSize | smallint | - | FK | NO | tSizeType |
| aBill | money | - | - | NO |  |
| aPostingdate | datetime | - | - | NO | - |

**Figure 2.2 tOrder table**

tEmployee table contains all employee users information.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table : tEmployee** | | | | | |
| ***Column***  ***Name*** | ***Data Type*** | ***Max Digits*** | ***Key*** | ***Null*** | ***Description*** |
| eID | integer | - | PK | NO | Identity(1,1) |
| eName | char | 64 | - | NO | - |
| ePassword | char | 32 | - | NO | - |

**Figure 2.3 tEmployee table**

ContType table contains content type names and fees

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table : tContType** | | | | | |
| ***Column***  ***Name*** | ***Data Type*** | ***Max Digits*** | ***Key*** | ***Null*** | ***Description*** |
| cID | integer | - | PK | NO | Identity(1,1) |
| ContentName | char | 16 | - | NO | - |
| Cost | money | - | - | NO | - |

We have already decided default values

|  |  |  |
| --- | --- | --- |
| cID | Content Name | Cost |
| 1 | Text | 200.00 |
| 2 | Graphics | 500.00 |

**Figure 2.3 tContType table**

PageType table contains page type names and fees

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table : tPageType** | | | | | |
| ***Column***  ***Name*** | ***Data Type*** | ***Max Digits*** | ***Key*** | ***Null*** | ***Description*** |
| pID | Integer | - | PK | NO | Identity(1,1) |
| Page Name | Char | 16 | - | NO | - |
| Cost | Money | - | - | NO | - |

We have already decided default values

|  |  |  |
| --- | --- | --- |
| pID | Page Name | Cost |
| 1 | Front | 1000.00 |
| 2 | Second | 700.00 |
| 3 | Middle | 500.00 |
| 4 | Last | 800.00 |

**Figure 2.3 tPageType table**

SizeType table holds size type names and mulitiplex

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table : tSizeType** | | | | | |
| ***Column***  ***Name*** | ***Data Type*** | ***Max Digits*** | ***Key*** | ***Null*** | ***Description*** |
| sID | Integer | - | PK | NO | Identity(1,1) |
| Size Name | Char | 16 | - | NO | - |
| Size Weight | float | - | - | NO | - |

We have already decided default values

|  |  |  |
| --- | --- | --- |
| sID | Page Name | Cost |
| 1 | Large | 10.0 |
| 2 | Medium | 3.0 |
| 3 | Small | 1.0 |

**Figure 2.3 tSizeType table**

# Architecture and Design in the Project

Figure 3.1-3.3 shows the architecture of the project.

Tier1

ADagent desktop application

JDBC Type4 driver

Tier2

Database ADagent (MS SQL2005)

**Figure 3.3: Architecture of the project**

# User Case

## Administrator

Administrator

Log in

Add/Delete employees’ account

Edit employees’ account

Log off

Modify order data in each employee

Delete order data

Modify cost fee parameters

**Figure 4.1: Administrator use case**

## Employee

Employees

Log in

Create a new order

Log off

**Figure 4.2: Employees use case**

# Data Flow Diagram (DFD)

We describe entity level in our application.

Customer

(Employee)

Placing a new order

checking details of advertisement fee.

Administrator

1

ADagent desktop application

Managing employee information and cost fees

checking details of ordering information in each employee

1

ADagent desktop application

**Figure 5.1: DFD LEVEL 0 – ADagent**

Submit

Retrieve Details of Orders

Display list of

Advertise Orders

6

Log on

8

Display Employee list

4

Display Orders Details

Admin

User ID &

Password

Administrator

Admin ID &

Password

Log on

Order

Details

Retrieve list of

Available Orders

Display

Orders Details

3

Display Order List

5

Placing a new Order

Employee Details

2

Log on

Employee

7

Display cost fees

Cost fee tables

**Figure 5.2: DFD LEVEL 1 – ADagent desktop application**

# Flowchart

## Booting up

### 

Create Entity Manager

Get Employee user list

Create GUI swing parts

Disable Tab bar panel.

**Figure 6.1 shows the flowchart for the application**.

## Login

### 

Show login dialog

Wait for clicking OK.

Check the input text

other

User = “admin”

Password = “123”

User = list.user

Password = list.passwd

dologinAdmin

dologinEmployee

Show login failure dialog

dologinSuccess

**Figure 6.2: Flow Chart for Comparison**

## editCustomerOK

Get Content Type fee from Table data

Get Page Type fee from Table data

Get Size Type multiply from Table data

Calculate bill

= (Content type fee + Page type fee) \* Size type multiplex

Set bill to Order Table to display users

**Figure 6.3: Flow Chart for editCustomerOk**

# GUI Standards Documents

ADagentView.java

|  |  |
| --- | --- |
| ***Used Swing Parts*** | ***Value name*** |
| JFrame | FrameView |
| JTabbedPane | frameTab |
| JPanel | empPane |
| JPanel | subPane |
| JPanel | mainPane |
| JPanel | statusPanel |
| JDialog | loginDialog |
| JDialog | alertDialog |

CustomerOrder.java

|  |  |
| --- | --- |
| ***Used Swing Parts*** | ***Value name*** |
| JDialog | - |
| JLabel | jLabel4 |
| JTextField | jTextField1 |
| JLabel | jLabel5 |
| JComboBox | contComboBox |
| JLabel | jLabel6 |
| JComboBox | pageComboBox |
| JLabel | jLabel7 |
| JComboBox | jComboBox3 |
| JLabel | jLabel8 |
| JFormattedTextField | jFormattedTextField1 |
| JLabel | jLabel9 |
| JButton | jButton1 |

# Interface Design Documents

|  |  |  |
| --- | --- | --- |
| ***Class Name*** | ***Description*** | ***Controls on the Document*** |
| ADagentView | Main Frame Window | ADagentView.java |
| empPane | Employee User Order Panel | ADagentView.java |
| subPane | Admin User Cost Fee Panel | ADagentView.java |
| mainPane | Admin User Manage Employee user | ADagentView.java |
| loginDialog | Login Dialog | ADagentView.java |
| alertDialog | Show to fail to login | ADagentView.java |
| CustomerOrder | New/Modify customer advertisement orders | CustomerOrder.java |

# Customer Requirement Specification (CRS)

## Client:

ADagent (advertisement order managed software)

## Business/Project Objective:

To create an application named ADagent using NetBeans IDE 6.8/6.9

## Input/Output:

* + Input/output for desktop application

## Hardware Requirements:

* + A minimum computer system that will help you access all the tools in the courses in a Pentium II 266 or better

## Operating System:

* + Windows XP with JRE5 or above.

## Software Requirement:

* + SQL Server 2005
  + Java runtime environment 5 or above

# Task

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project Ref. No.:** | | **Activity Plan**  **Prepared By:** | **Date of Preparation of Activity Plan:** | | | |
| **Sr.**  **No.** | **Task** | **Actual Start**  **Date** | **Actual**  **Days** | **Team Member Names** | **Status** |
| 1 | Definition of detailed specification |  | 23th Aug, 2010 | 2 |  | Completed |
| 2 | Creation of prototype application |  | 25th Aug, 2010 | 1 |  | Completed |
| 3 | Programming |  | 26th Aug, 2010 | 5 |  | Completed |
| 4 | Test |  | 30th Aug 2010 | 2 |  | Completed |
| 5 | Create Documentations |  | 28th Aug, 2010 | 4 |  | Completed |