



Quick Intro

Mohamed Mansour - 2911834 (section A)

Aha Remember to look to at my custom made blogger that explains briefly about this project!

http://www.m0interactive.com/portfolio/java/j2ee_online_banking_web_application/

Theory

Architecture [^]

J2EE applications are best developed using the Model-View-Controller (MVC) application model, which consists of the following three elements:

Element	Description
Model	Represents the application data and the business rules that manage the data. In J2EE applications, the model is typically represented by EJBs.
View	Renders the content of the model to the user of the application. In J2EE applications, the view is typically provided by JSP pages.
Controller	Defines how the application works. It maps user actions (such as button clicks) to operations performed by the model (such as updating information in a database). The controller mediates between the view and the model. In J2EE applications, the controller is typically a servlet, JavaBean, or session bean.

Using the MVC architecture, we separated the data, display, and flow of an application, allowing for greater flexibility and ease of reuse. MVC is also was a very good way to develop applications that support multiple presentations of the same data. As you can see we change the style of each page with ease.

Website Layout and Design [^]

The overall layout was using the latest technology that includes XHTML, CSS 2.0, using some DOM for the AJAX development. The reason why such technology was used is because web standards in web design are the most important criteria to look at. The stats in table 1 show the different stats among the users who browse online. It is clearly an issue these days of how really a webpage looks like among all these browsers.

The answer really, is that each browser has its own way to render html pages. Therefore don't be sure the webpage that runs in Internet Explorer will look the same as Netscape. Internet Explorer is known to be poor at rendering. So a web designer should be aware to use Web standards to construct pages. Web standards allow much joy in a webpage, your webpage will look at its best in every device (PALM/PDA/BROWSERS/Ect.).

2005	IE 6	IE 5	Firefox	Mozilla	Netscape7	Opera 8	Opera 7
October	67.5%	6.0%	19.6%	2.6%	0.4%	1.2%	0.2%
September	69.8%	5.7%	18.0%	2.5%	0.4%	1.0%	0.2%
August	68.4%	6.3%	18.9%	2.4%	0.4%	0.8%	0.3%
July	67.9%	5.9%	19.8%	2.6%	0.5%	0.8%	0.4%
June	65.0%	6.8%	20.7%	2.9%	0.6%	0.7%	0.5%
May	64.8%	6.8%	21.0%	3.1%	0.7%	0.7%	0.6%
April	63.5%	7.9%	20.9%	3.1%	0.9%	0.4%	1.0%
March	63.6%	8.9%	18.9%	3.3%	1.0%	0.3%	1.6%
February	63.9%	9.5%	17.9%	3.3%	1.0%		1.7%
January	64.8%	9.7%	16.6%	3.4%	1.1%		1.9%

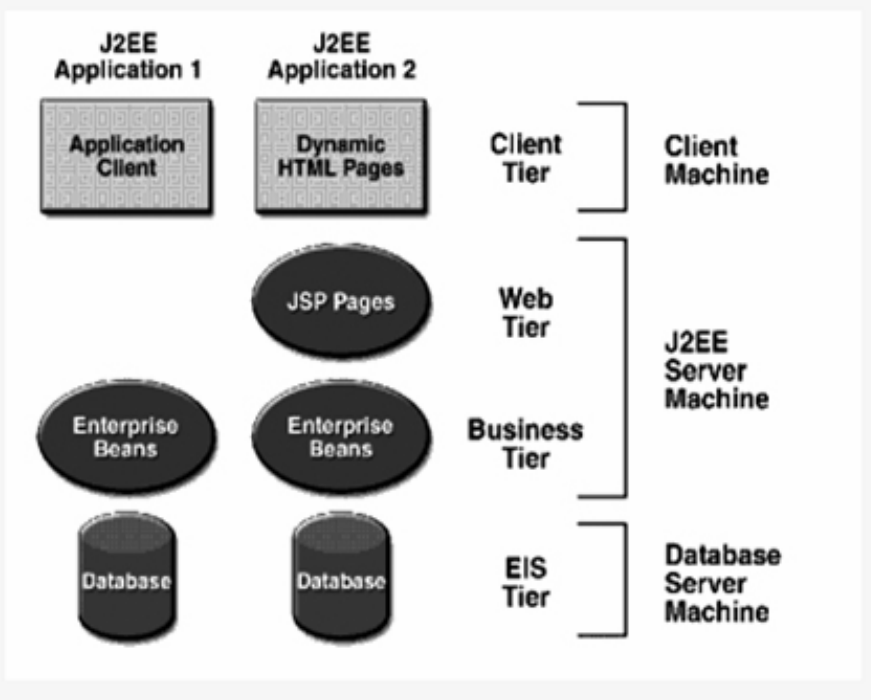
TABLE 1: Source: <http://www.w3schools.com>

This site is designed using the web standard approach. It validates the W3C test and is Bobby A approved. Tools were used to prove this outcome! The HTML structure was using XHTML Strict 1.0 template, and by making a structural XHTML design, it allows the flexibility in our overall design. By changing 1 file, the whole website look will change completely!

We have included a custom built JavaScript style chooser which demonstrates this point. By selecting your style of your choice, the whole website template changes!

Programming Design and Structure [^]

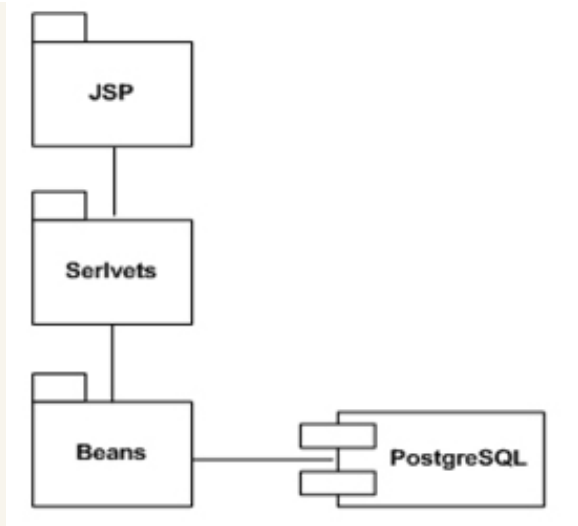
The overall programming was developed using 3-tier architecture. We have a Model View Controller (MVC) ideology in using Enterprise Java Beans as our development tool. Ideally speaking this is the best approach into building webpage's.



The reason being, we have separated the data, design, and logic into different machines. If a person wants to change his database from using PostgreSQL to MySQL, only one variable needs to be changed and the whole website will still have the same functionality.

We have used a package modular approach for the decouplization of components. That allowed us for reusability and flexibility in the future.

The following is the package structure of the Main Savings Inc.



We have used Java beans and servlets and JSP for our core implementations. The servlets follow the action pattern and the database follows the data access object pattern. We have used templating and other sort of usefull techniques.

Database Structure [^](#)

The current database was originally 18 tables, we added 3 extra tables:

Payee - Was added to let the bank be control of the payees relevant to the bill payment. So the employees of the bank are allowed to add differnt payee information so that the user can add them to his account for future bill payments.

Customer Payee - The customer payee table is just a relationship. Whenever a customer adds a payee to his account, this table maps the relationship between the customer and the payee, making it a many to many relationship.

Holds - We included this to tell which account is which, withought that, there are no representation to tell which account type is for which user.

We added 3 Procedures into the database which were called:

payBill(AMOUNT,ACCOUNT,PAYEE,CUSTOMER,DATE);

By entering the given parameters the funtion checks many stuff if money is available, user is available else throws an exception. This function calls another function which is updateAccount

```

CREATE FUNCTION payBill(NUMERIC,INTEGER,INTEGER,INTEGER,DATE) RETURNS INTEGER AS '
    DECLARE v_desc VARCHAR;
    DECLARE v_payee VARCHAR;
    DECLARE v_remaining NUMERIC;
    DECLARE v_amount NUMERIC;
    BEGIN
        IF ($1 < 0 ) THEN
            RAISE EXCEPTION '% cannot have negative amount', $1;
        END IF;

        v_amount := -1 * $1;
        IF v_overdraft IS NULL THEN
            v_overdraft := 0;
        END IF;

        SELECT (amount-abs($1)) INTO v_remaining FROM account WHERE account=$2;
        SELECT payee INTO v_payee FROM payee WHERE pid=$3;

        v_desc := 'Bill payment - ' || v_payee;

        IF (v_remaining < 0) THEN
            RAISE EXCEPTION 'Insuffient funds, % does not exist in your account', $1;
        END IF;

        INSERT INTO bill(pid,date,account,cid,amount)
            VALUES($3,current_date,$2,$4,$1);
  
```

```

        INSERT INTO activity(descr,account,cid,date,rem_bal,amount)
            VALUES(v_desc,$2,$4,$5,v_remaining,v_amount);

        PERFORM updateaccount($2,v_amount);
    RETURN 0;
END;' LANGUAGE plpgsql;

```

updateAccount(ACCOUNT,AMOUNT)

This function just updates the amount field in the Account table, used from the other two functions.

```

CREATE FUNCTION updateAccount(INTEGER,NUMERIC) RETURNS INTEGER AS '
    BEGIN
        IF $2 IS NULL THEN
            RAISE EXCEPTION '% cannot have null amount', $2;
        END IF;
        IF ($2 < 0) THEN
            IF (SELECT true
                FROM account
                WHERE account=$2
                AND amount < abs($2))
            THEN
                RAISE EXCEPTION '% not enough money', $2;
            END IF;
        END IF;
        UPDATE account SET amount=amount+$2 WHERE account=$1;
    RETURN 0;
END;' LANGUAGE plpgsql;

```

transferMoney(AMOUNT,ACCOUNTFROM,ACCOUNTTO,CUSTOMER);

Transfers the money from one account to an other calls the updateAccount function to update the amount, extensive error checking consists as well

```

CREATE FUNCTION transferMoney(NUMERIC,INTEGER,INTEGER,INTEGER) RETURNS INTEGER AS '
    DECLARE v_desc VARCHAR;
    DECLARE v_remaining NUMERIC;
    DECLARE v_amount NUMERIC;
    BEGIN
        IF ($1 < 0 ) THEN
            RAISE EXCEPTION '% cannot have negative amount', $1;
        END IF;

        -- INSERT MONEY TO THE TO
        v_amount := -1 * $1;
        SELECT (amount-abs($1)) INTO v_remaining FROM account WHERE account=$2;
        v_desc := 'Transfer to - ' || to_char($3,'000000000000');

        IF (v_remaining < 0) THEN
            RAISE EXCEPTION 'Insufficient funds, % does not exist in your account', $1;
        END IF;

        INSERT INTO activity(descr,account,cid,date,rem_bal,amount)
            VALUES(v_desc,$2,$4,current_date,v_remaining,v_amount);

        PERFORM updateaccount($2,v_amount);

        -- INSERT MONEY TO THE FROM
        SELECT (amount+abs($1)) INTO v_remaining FROM account WHERE account=$3;

        IF (v_remaining < 0) THEN
            RAISE EXCEPTION 'Insufficient funds, % does not exist in your account', $1;
        END IF;

        v_desc := 'Transfer from - ' || to_char($2,'000000000000');

        INSERT INTO activity(descr,account,cid,date,rem_bal,amount)

```

```
VALUES(v_desc,$3,$4,current_date,v_remaining,$1);
```

```
PERFORM updateaccount($3,$1);
```

```
RETURN 0;
```

```
END;' LANGUAGE plpgsql;
```

Installation [^]

Previous knowledge of J2EE programming is needed.

Run the sql scripts [bank.sql](#) and the [bank-populate.sql](#)

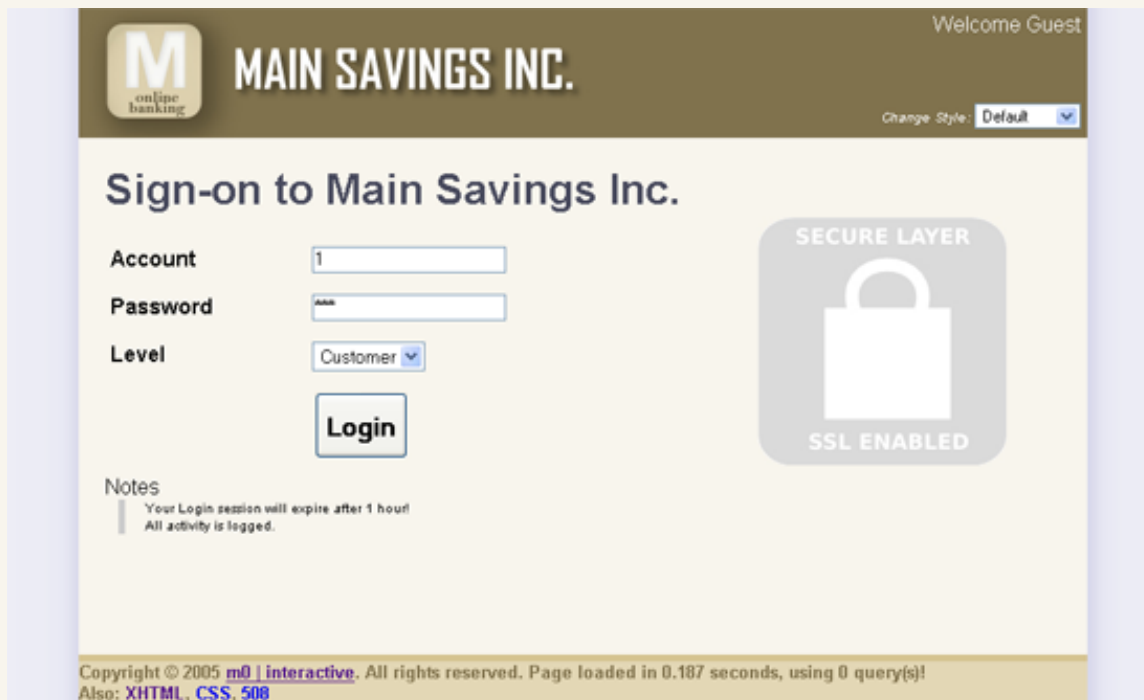
Package the Web Application

Deploy the application

Run by visiting (<http://localhost:8080/Bank/>) Assuming that is your host and port. Account: 1 Password: 111

Screenshots [^]

Login:



View Accounts:


MAIN SAVINGS INC.
Welcome Mohamed Mansour

Change Style: **Default**

[HOME](#)
[ACCOUNTS](#)
[PAYMENTS](#)
[TRANSFERS](#)
[PROFILE](#)
[LOGOUT](#)

Account

My Account Summary


Account Type	Account Number	Balance CAD
Powerchecking	000000000008	2.50
Powerchecking	000000000009	128.50
Powersaving	000000000001	5.50
Powersaving	000000000002	204.50
Powersaving	000000000003	226.50
Total Day-to-Day Banking:		567.50

Notes
 Important information about how your accounts are displayed
 Click account number to view account information.
 Click on the balance to view the account details (where applicable).



Copyright © 2005 m0 | interactive. All rights reserved. Page loaded in 1.109 seconds, using 1 query(s)!
 Also: [XHTML](#), [CSS](#), [508](#)

View Accounts Different Style (by changing the style dropdown simple :), full screen):


Main Savings Inc. ©
Welcome Mohamed Mansour

Change Style: **Simple**

[Home](#)
[Accounts](#)
[Payments](#)
[Transfers](#)
[Profile](#)
[Logout](#)

Account

My Account Summary

Account Type	Account Number	Balance CAD
Powerchecking	000000000008	2.50
Powerchecking	000000000009	100.50
Powersaving	000000000001	5.50
Powersaving	000000000002	204.50
Powersaving	000000000003	214.50
Total Day-to-Day Banking:		527.50

Notes
 Important information about how your accounts are displayed
 Click account number to view account information.
 Click on the balance to view the account details (where applicable).

Copyright © 2005 m0 | interactive. All rights reserved. Page loaded in 0.218 seconds, using 1 query(s)!
 Also: [XHTML](#), [CSS](#), [508](#)

View Details of a specific Account

Welcome Mohamed Mansour

M online banking **MAIN SAVINGS INC.** Change Style: **Default**

[HOME](#) [ACCOUNTS](#) [PAYMENTS](#) [TRANSFERS](#) [PROFILE](#) [LOGOUT](#)

Account Details

Powersaving - 000000000003

Account

Account Type	Account Number	Balance CAD
Powersaving	000000000003	214.50

Transaction Details

Activity ID	Date	Description	Withdrawals	Deposits	Balance
00006	2005-12-12	Transfer from - 0000000000008	-	50.00	226.50
00010	2005-12-12	Bill payment - Fido	12.00	-	214.50

Notes
 All currency is in Canadian dollars!
 Navigate to your other accounts within the drop down box above.
 The date shown will be the date when the money is being sent!
 Overdraft money is not shown!



Copyright © 2005 [m0](#) | [interactive](#). All rights reserved. Page loaded in 0.281 seconds, using 3 query(s)!
 Also: [XHTML](#), [CSS](#), [508](#)

Add a Payee

Welcome Mohamed Mansour

M online banking **MAIN SAVINGS INC.** Change Style: **Default**

[HOME](#) [ACCOUNTS](#) [PAYMENTS](#) [TRANSFERS](#) [PROFILE](#) [LOGOUT](#)

Bill Payment


Add a Payee

[Add a Payee](#) [View your Payee's](#) [Pay a Bill](#)

Payee List:

Add Payee

Notes
 If you wish to add a payee to the list, please contact the branch for approval. (security purposes)



Copyright © 2005 [m0](#) | [interactive](#). All rights reserved. Page loaded in 0.328 seconds, using 1 query(s)!
 Also: [XHTML](#), [CSS](#), [508](#)

Pay a Bill, successful!

Welcome Mohamed Mansour

M online banking **MAIN SAVINGS INC.** Change Style: **Default**

HOME ACCOUNTS PAYMENTS TRANSFERS PROFILE LOGOUT

Bill Payment

Pay a Bill

[Add a Payee](#) [View your Payee's](#) [Pay a Bill](#)

✔ Bill Payment was Successful!

To:

From:


Date:

Amount \$:

Submit

Notes
Date must be today or later!
The balance on your account doesn't have overdraft protection.
Use the navigation above to add/view a payee or pay a bill.

Copyright © 2005 [m0 | interactive](#). All rights reserved. Page loaded in 0.297 seconds, using 2 query(s)!
Also: [XHTML](#), [CSS](#), [508](#)



Transfer Money to an account, transfer error

Welcome Mohamed Mansour

M online banking **MAIN SAVINGS INC.** Change Style: **Default**

HOME ACCOUNTS PAYMENTS TRANSFERS PROFILE LOGOUT

Transfer

Transfer an Account

⚠ Transfer Error!

Account:

Transfer To:

Amount \$:

Submit

Notes
Transfer exist instantly!
Must have money in the account in order to transfer!

Copyright © 2005 [m0 | interactive](#). All rights reserved. Page loaded in 0.297 seconds, using 2 query(s)!
Also: [XHTML](#), [CSS](#), [508](#)



Alter Profile, fields error

M online banking **MAIN SAVINGS INC.** Welcome Mohamed Mansour

Change Style: **Default**

HOME **ACCOUNTS** **PAYMENTS** **TRANSFERS** **PROFILE** **LOGOUT**

Profile

Update Profile information

Empty: address
Empty: city
Empty: phone

*E-Mail

*Password

Confirm Password

*Address

*City

*Phone

Update



Notes
Those payees are registered under your account.
When you delete a payee it is still recorded in your Activities.

Copyright © 2005 [m0 | interactive](#). All rights reserved. Page loaded in 0.204 seconds, using 0 query(s)!
Also: [XHTML](#), [CSS](#), [508](#)

Notes

- This is a readme file.
- All activity is NOT logged.