COMP 353

Final Project Report

Group Members

Mathieu Dumais-Savard	6095275
Justin Torres	9356592
Ramez Debbas	9103910
Wadih El-Ghoussoubi	9459510

Table Of Contents

Table Of Contents 2
Database design 3
Context and requirements 3
Changes from warm-up and justification 3
Conversion to 3NF 5
Step 1 - From Entity to relations 5
Step 2 - Looking at the various relations between entities 5
Step 3 - Write down all the modified relation and check for 3NF 7
User interface manual 9
Manager View 9
1. Clients Tab 9
Employee View 9
1. Clients Tab 9
Client View 9
1. Accounts Tab 9
How-to 9
Manage employees 9
Manage clients 10
Manage accounts 13
See my schedule and pay 14
List of members contribution 16
Installation Guide 17
How-to access source code 19

Database design

Context and requirements

Your database will be responsible for supporting
The branch (branch_id, location, phone, fax, opening_date, manager_name, etc.)
The employee (employee_id, title, name, address, start_date, salary, etc.)
The client (client_id, name, date of birth, joining_date, address, category, etc.)
The account (account_number, type, option, balance, etc.)
The services (banking, investment, insurance)
The interest rate (kind of service, type of account, percentage, etc.)
The charge plans (option, limit, charge, etc.)

Changes from warm-up and justification

We added the relationship ClientCard, Employee_Payroll, Employee_Schedule, Employee_Login since they were new requirements, ie: not present during the warm-up project.

We added the relationship Reports_Month to ease report generation.

We added views Client_Overview, Password, billableTransactionSoFarThisMonth and profitability_report to support new requirement and ease information display for the front-end.

Conversion to 3NF

Step 1 - From Entity to relations

Branch(branch_id, fax, location, phone, opening_Date)
Employee(employee_id, title, address, start_date, name, salary)
Client(client_id, category, address, name, date_of_birth, joining_date)
Services(banking, investment, insurance)
Employee_Schedule(ScheduleDay, ScheduleCode, ScheduleHours, StartTime)
Transactions(TransactionID, Amount, Data, Type, Description, Fees)
Charge_plans(plan_id, limit, plan_option, charge)
Interest_Rate(rate_id, percentage, type_of_account, kind_of_service)
Account(account_number, account_option, type, balance, creation_date, credit_limit, level)
ClientCard(ClientCardNumber, Password)
Employee_login(loginPassword)
Employee_payroll(PayId, PayDate, PayAmount)

Step 2 - Looking at the various relations between entities

Relation 1 - ManagedBy between Branch and Employee

A branch must have a manager and only one so we add manager_id references Employee(employeeId) to Branch

Branch(branch_id, fax, location, phone, opening_Date, managedBy)

Relation 2 - Employs between Branch and Employee

An employee is employed by a branch and only one, so we decided to add a branchID references Branch(branchId) to Employee

Employee(employee_id, title, address, start_date, name, salary, branchId)

Relation 3 - Provides between Branch and Services

For this one, we added the 3 fields from Services to Branch by using 3 boolean (T/F) fields to Branch. This avoid the need of a Service relation.

Branch(branch_id, fax, location, phone, opening_Date, managedBy, hasBanking, hasInvestment, hasInsurance)
~~Services(banking, investment, insurance)~~

Relation 4 - Has between Branch and Account

Each account belongs to one and only one branch, we added a branchId to Account referencing Branch(branchId).

Account(account_number, account_option, type, balance, creation_date, credit_limit, level, branchId)

Relation 5 - Has between Employees and EmployeeSchedule

Each employee is schedule for several dates, each row in the EmployeeSchedule table represent the working say of a single employee, we added EmployeeID references Employee(EmployeeID) to EmployeeSchedule and added EmployeeID to its primary key.

Employee_Schedule(ScheduleDay, EmployeeID, ScheduleCode, ScheduleHours, StartTime)

Relation 6 - Has between EmployeeLogin and Employee

Each employee has a password but we wanted the passwords to be separate for security reasons, we kept the entity Employee_Login and added EmployeeID references Employee(EmployeeID) and made it its primary key

Employee_login(employeeID, loginPassword)

Relation 7 - Has between Account and ChargePlans

Relation 8 - Has between Account and InterestRate

Each account has a single charge plan and a single interest rate. So we added two fields to account to reference these informations, namely planId REFERENCES Charge_Plan(planID) and RateId REFERENCES Interest_Rate(RateId)

Account(account_number, account_option, type, balance, creation_date, credit_limit, level, branchId, planId, rateId)

Relation 9 - Owns between Client and Account

This is the only many-to-many relationship in our E/R so we had to create a new relation to represent it.

We created the relationship Client_owns_account with two fields. ClientID REFERENCES Client(clientID) and AccountID REFERENCES Account(AccountId), the two fields were primary key of their respectives tables so we took them as the primary key of this new relation.

Client_owns_account(ClientId, AccountId)

Relation 10 - Has between Employee and Employee_Payroll

An employee can have several paycheck but each paycheck belongs to a single employee only. We added employeeId REFERENCES employee(EmployeeID) to Employee_Payroll

Employee_payroll(PayId, EmployeeID, PayDate, PayAmount)

Relation 11 - PerformedOn between Transaction and Account

Each transaction is performed on a single account, so we added an AccountID REFERENCED Account(AccoundID) field to the transaction relation

Transactions(TransactionID, Amount, Data, Type, Description, Fees, AccountId)

Relation 12 - ClientOwnsClientCard between Client and Client Card

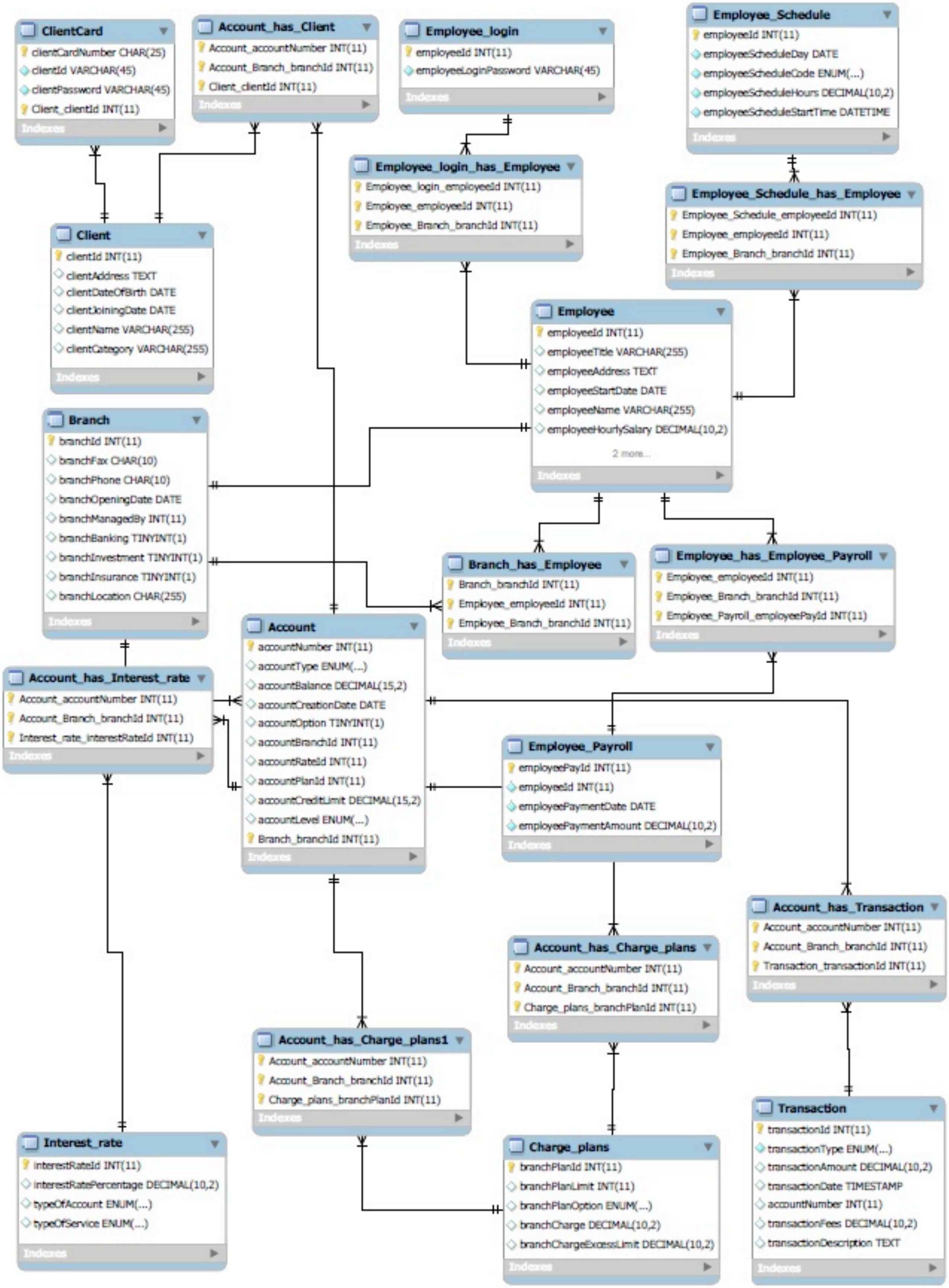
We made the assumption that a card represents a client after reading the project description several time. A client can have several cards, each card have its own password associated with it like Desjardins does. We then added the property ClientID REFERENCED Client(ClientID) to ClientCard

ClientCard(ClientCardNumber, Password, ClientID)

Step 3 - Write down all the modified relation and check for 3NF

ClientCard(ClientCardNumber, Password, ClientID)
Transactions(TransactionID, Amount, Data, Type, Description, Fees, AccountId)
Employee_payroll(PayId, EmployeeID, PayDate, PayAmount)
Client_owns_account(ClientId, AccountId)
Account(account_number, account_option, type, balance, creation_date, credit_limit, level, branchId, planId, rateId)
Branch(branch_id, fax, location, phone, opening_Date, managedBy, hasBanking, hasInvestment, hasInsurance)
Employee(employee_id, title, address, start_date, name, salary, branchId)
Employee_login(employeeID, loginPassword)
Employee_Schedule(ScheduleDay, EmployeeID, ScheduleCode, ScheduleHours, StartTime)
Charge_plans(plan_id, limit, plan_option, charge)
Interest_Rate(rate_id, percentage, type_of_account, kind_of_service)

For each of these relations, it turns out the PrimaryKey determine all other information so they are all BCNF wich includes being all in 3NF.


ClientCard
clientCardNumber CHAR(25)
clientId VARCHAR(45)
clientPassword VARCHAR(45)
Client_clientId INT(11)
Indexes
Account_has_Client
Account_accountNumber INT(11)
Account_Branch_branchId INT(11)
Client_clientId INT(11)
Indexes
Employee_login
employeeId INT(11)
employeeLoginPassword VARCHAR(45)
Indexes
Employee_Schedule
employeeId INT(11)
employeeScheduleDay DATE
employeeScheduleCode ENUM(...)
employeeScheduleHours DECIMAL(10,2)
employeeScheduleStartTime DATETIME
Indexes
Employee_login_has_Employee
Employee_login_employeeId INT(11)
Employee_employeeId INT(11)
Employee_Branch_branchId INT(11)
Indexes
Employee_Schedule_has_Employee
Employee_Schedule_employeeId INT(11)
Employee_employeeId INT(11)
Employee_Branch_branchId INT(11)
Indexes
Client
clientId INT(11)
clientAddress TEXT
clientDateOfBirth DATE
clientJoiningDate DATE
clientName VARCHAR(255)
clientCategory VARCHAR(255)
Indexes
Employee
employeeId INT(11)
employeeTitle VARCHAR(255)
employeeAddress TEXT
employeeStartDate DATE
employeeName VARCHAR(255)
employeeHourlySalary DECIMAL(10,2)
2 more...
Indexes
Branch
branchId INT(11)
branchFax CHAR(10)
branchPhone CHAR(10)
branchOpeningDate DATE
branchManagedBy INT(11)
branchBanking TINYINT(1)
branchInvestment TINYINT(1)
branchInsurance TINYINT(1)
branchLocation CHAR(255)
Indexes
Branch_has_Employee
Branch_branchId INT(11)
Employee_employeeId INT(11)
Employee_Branch_branchId INT(11)
Indexes
Employee_has_Employee_Payroll
Employee_employeeId INT(11)
Employee_Branch_branchId INT(11)
Employee_Payroll_employeePayId INT(11)
Indexes
Account
accountNumber INT(11)
accountType ENUM(...)
accountBalance DECIMAL(15,2)
accountCreationDate DATE
accountOption TINYINT(1)
accountBranchId INT(11)
accountRateId INT(11)
accountPlanId INT(11)
accountCreditLimit DECIMAL(15,2)
accountLevel ENUM(...)
Branch_branchId INT(11)
Indexes
Account_has_Interest_rate
Account_accountNumber INT(11)
Account_Branch_branchId INT(11)
Interest_rate_interestRateId INT(11)
Indexes
Employee_Payroll
employeePayId INT(11)
employeeId INT(11)
employeePaymentDate DATE
employeePaymentAmount DECIMAL(10,2)
Indexes
Account_has_Transaction
Account_accountNumber INT(11)
Account_Branch_branchId INT(11)
Transaction_transactionId INT(11)
Indexes
Account_has_Charge_plans
Account_accountNumber INT(11)
Account_Branch_branchId INT(11)
Charge_plans_branchPlanId INT(11)
Indexes
Account_has_Charge_plans1
Account_accountNumber INT(11)
Account_Branch_branchId INT(11)
Charge_plans_branchPlanId INT(11)
Indexes
Interest_rate
interestRateId INT(11)
interestRatePercentage DECIMAL(10,2)
typeOfAccount ENUM(...)
typeOfService ENUM(...)
Indexes
Charge_plans
branchPlanId INT(11)
branchPlanLimit INT(11)
branchPlanOption ENUM(...)
branchCharge DECIMAL(10,2)
branchChargeExcessLimit DECIMAL(10,2)
Indexes
Transaction
transactionId INT(11)
transactionType ENUM(...)
transactionAmount DECIMAL(10,2)
transactionDate TIMESTAMP
accountNumber INT(11)
transactionFees DECIMAL(10,2)
transactionDescription TEXT
Indexes

User interface manual

Manager View

Username: 3
Password: admin

1. Clients Tab
   Managing Clients
2. Accounts Tab
   Managing Accounts
3. Employee Tab
   Managing Employees
4. Manager Information Tab
   Viewing Manager Information / Pay stub / work schedual

Employee View

Username: 1
Password: admin

1. Clients Tab
   Managing Clients
2. Accounts Tab
   Managing Accounts
3. Manager Information Tab
   Viewing Employee Information/ Pay stub / work schedual

Client View

Username: 5181879446019637
Password: user

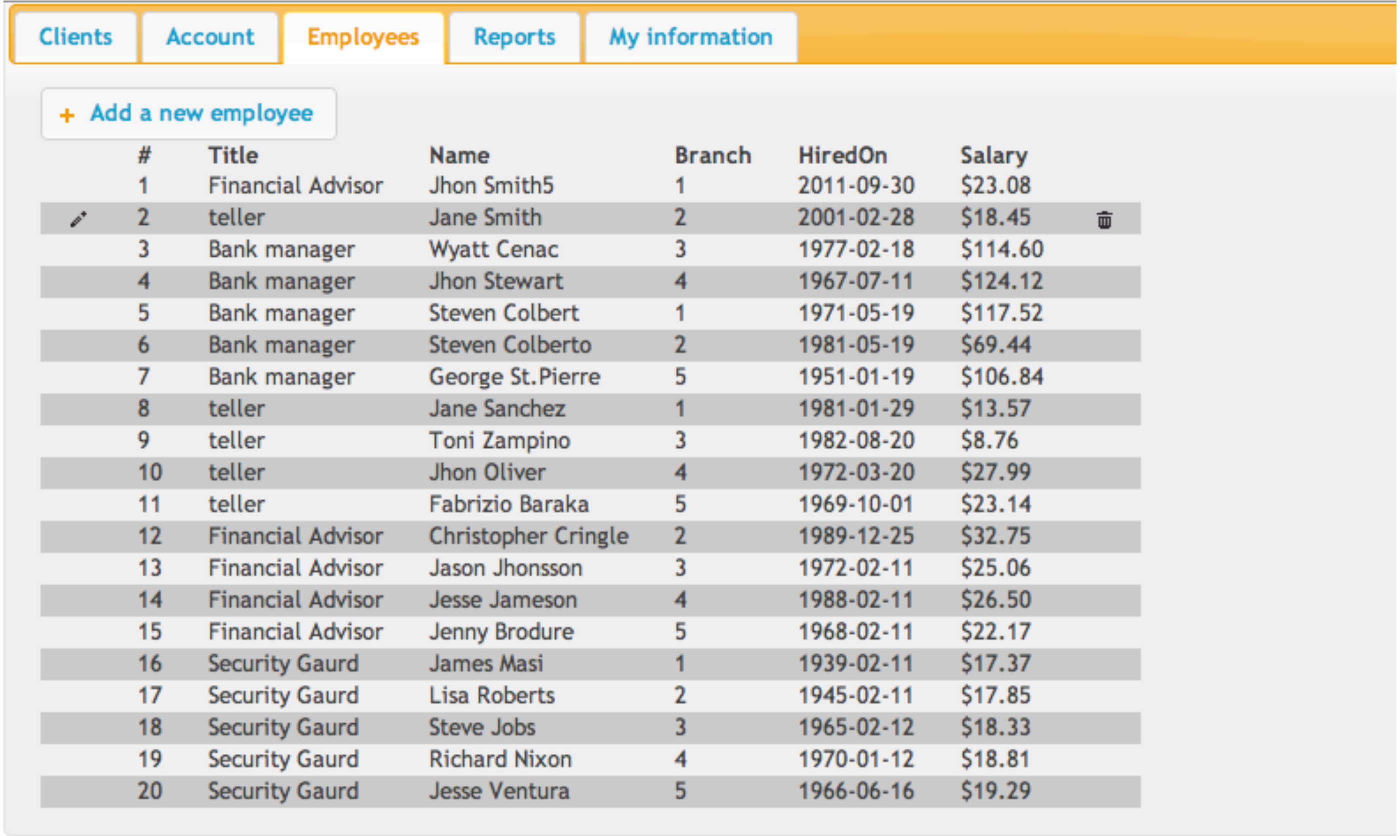
1. Accounts Tab
   Managing Accounts such as transfer money between accounts
2. Transaction Tab
   Viewing Clients Transaction. Also ability to deposit and withdraw money.
3. Client Information Tab
   Viewing Client Information

How-to

Manage employees

Managers of branches can only manage employees. Once you are logged as a manager, you will have an employee tab where you can add, edit and delete each employee.

Each row of employee has an edit and delete buttons.



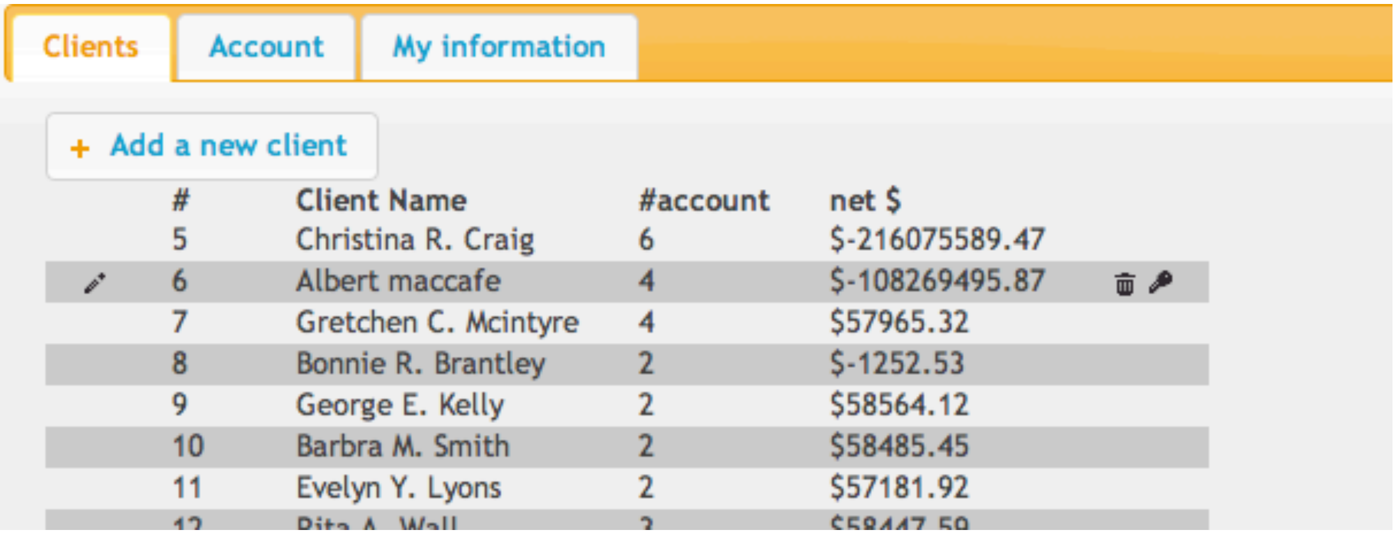
	#	Title	Name	Branch	HiredOn	Salary	
	1	Financial Advisor	Jhon Smith5	1	2011-09-30	$23.08	
✎	2	teller	Jane Smith	2	2001-02-28	$18.45	🗑
	3	Bank manager	Wyatt Cenac	3	1977-02-18	$114.60	
	4	Bank manager	Jhon Stewart	4	1967-07-11	$124.12	
	5	Bank manager	Steven Colbert	1	1971-05-19	$117.52	
	6	Bank manager	Steven Colberto	2	1981-05-19	$69.44	
	7	Bank manager	George St.Pierre	5	1951-01-19	$106.84	
	8	teller	Jane Sanchez	1	1981-01-29	$13.57	
	9	teller	Toni Zampino	3	1982-08-20	$8.76	
	10	teller	Jhon Oliver	4	1972-03-20	$27.99	
	11	teller	Fabrizio Baraka	5	1969-10-01	$23.14	
	12	Financial Advisor	Christopher Cringle	2	1989-12-25	$32.75	
	13	Financial Advisor	Jason Jhonsson	3	1972-02-11	$25.06	
	14	Financial Advisor	Jesse Jameson	4	1988-02-11	$26.50	
	15	Financial Advisor	Jenny Brodure	5	1968-02-11	$22.17	
	16	Security Gaurd	James Masi	1	1939-02-11	$17.37	
	17	Security Gaurd	Lisa Roberts	2	1945-02-11	$17.85	
	18	Security Gaurd	Steve Jobs	3	1965-02-12	$18.33	
	19	Security Gaurd	Richard Nixon	4	1970-01-12	$18.81	
	20	Security Gaurd	Jesse Ventura	5	1966-06-16	$19.29	

When adding, editing an employee a pop up page will show where you can select an options of that employee.
When deleting an employee, a alert message will show asking you if you want to delete the employee.

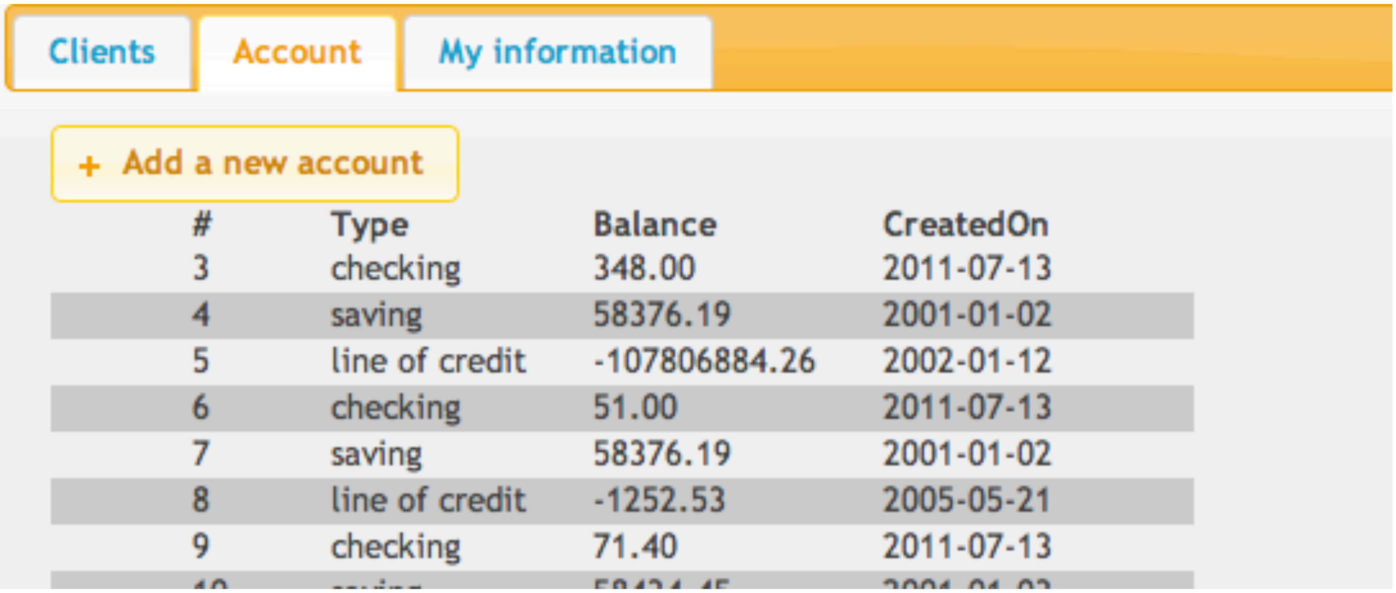
Manage clients

- **Managers and Employees logged in**

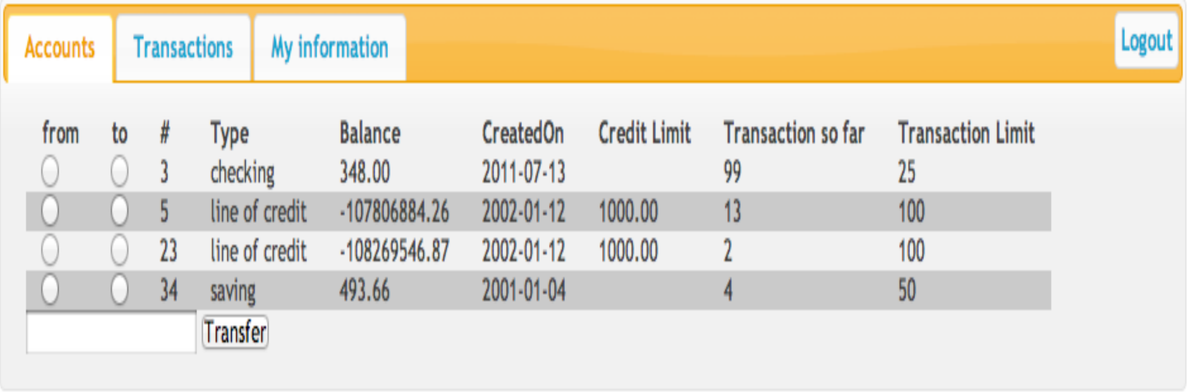
Managers and employees of a branch have access to an clients tab. Where they can add, edit and delete clients. Also, on each row there is an extra button where you can associate each client to a specific account that was created before.



	#	Client Name	#account	net $
	5	Christina R. Craig	6	$-216075589.47
	6	Albert maccafe	4	$-108269495.87
	7	Gretchen C. Mcintyre	4	$57965.32
	8	Bonnie R. Brantley	2	$-1252.53
	9	George E. Kelly	2	$58564.12
	10	Barbra M. Smith	2	$58485.45
	11	Evelyn Y. Lyons	2	$57181.92
	12	Rita A. Wall	3	$58447.59



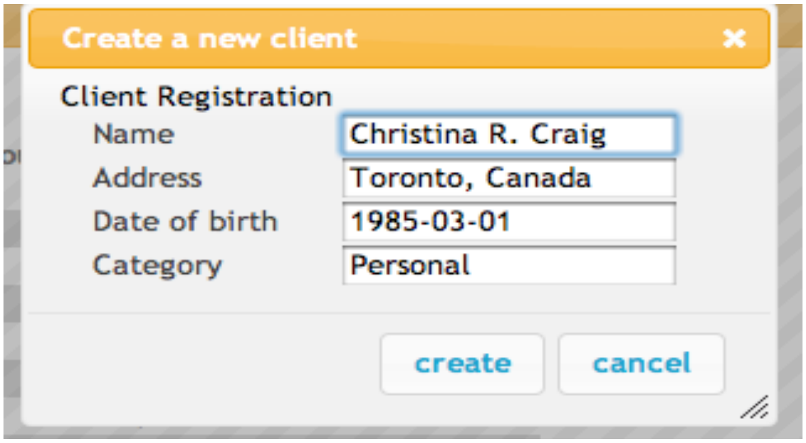
#	Type	Balance	CreatedOn
3	checking	348.00	2011-07-13
4	saving	58376.19	2001-01-02
5	line of credit	-107806884.26	2002-01-12
6	checking	51.00	2011-07-13
7	saving	58376.19	2001-01-02
8	line of credit	-1252.53	2005-05-21
9	checking	71.40	2011-07-13
10	saving	58424.45	2001-01-02



from	to	#	Type	Balance	CreatedOn	Credit Limit	Transaction so far	Transaction Limit
○	○	3	checking	348.00	2011-07-13		99	25
○	○	5	line of credit	-107806884.26	2002-01-12	1000.00	13	100
○	○	23	line of credit	-108269546.87	2002-01-12	1000.00	2	100
○	○	34	saving	493.66	2001-01-04		4	50

When adding, editing clients a pop up page will show where you can select an options of that client.
When deleting an client, a alert message will show asking you if you want to delete the client.

When associating an account to a client, a prompt pop up will show asking you to enter account id to be associated.



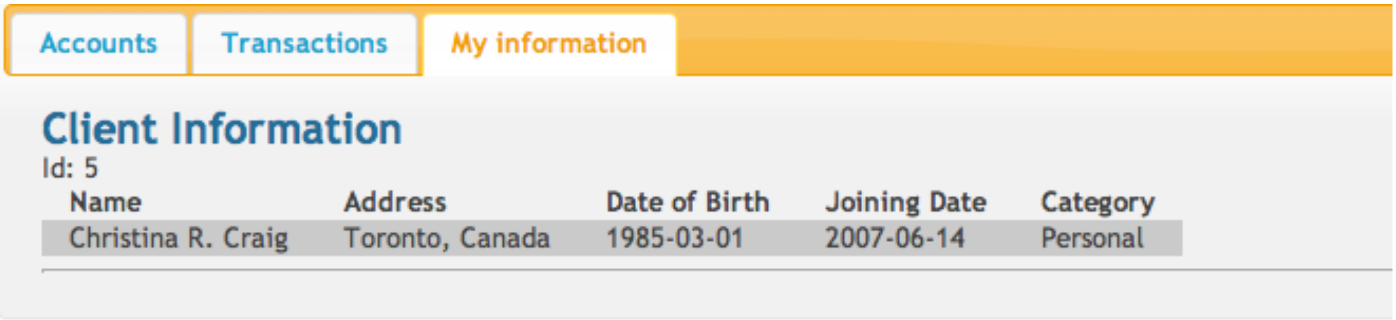
Client Registration

Name: Christina R. Craig
Address: Toronto, Canada
Date of birth: 1985-03-01
Category: Personal

Note: you can associate multiple accounts for every employee . At the same time same account can be associated to many clients as its a joint account.

- **Clients logged in**

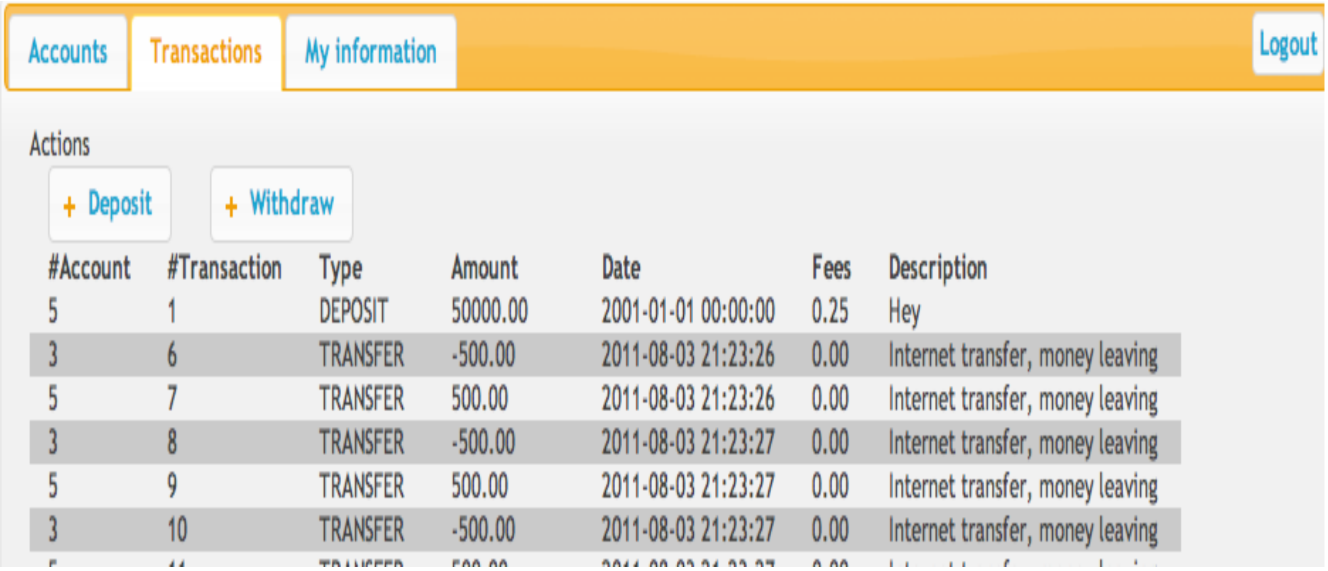
When clients login, they can view their account information under my information tab. Also, under accounts they can view all their account's information's such as account type , balance,and transferring money between them. Transaction tab to view all their past transaction including fees paid if over passed the limit. Also, you can withdraw and deposit money under transaction Tab.



Client Information

Id: 5

Name	Address	Date of Birth	Joining Date	Category
Christina R. Craig	Toronto, Canada	1985-03-01	2007-06-14	Personal

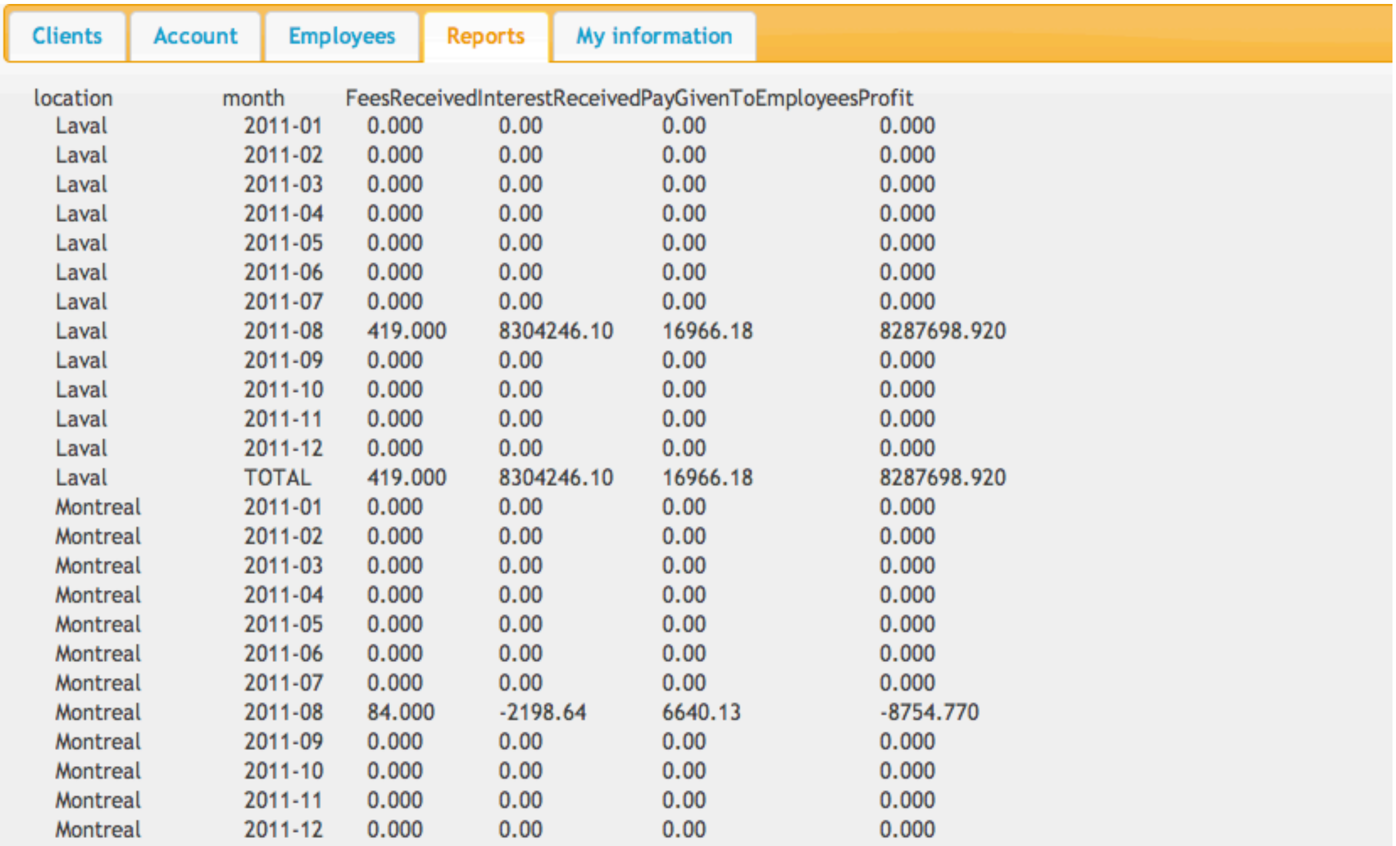


#Account	#Transaction	Type	Amount	Date	Fees	Description
5	1	DEPOSIT	50000.00	2001-01-01 00:00:00	0.25	Hey
3	6	TRANSFER	-500.00	2011-08-03 21:23:26	0.00	Internet transfer, money leaving
5	7	TRANSFER	500.00	2011-08-03 21:23:26	0.00	Internet transfer, money leaving
3	8	TRANSFER	-500.00	2011-08-03 21:23:27	0.00	Internet transfer, money leaving
5	9	TRANSFER	500.00	2011-08-03 21:23:27	0.00	Internet transfer, money leaving
3	10	TRANSFER	-500.00	2011-08-03 21:23:27	0.00	Internet transfer, money leaving

Manage accounts

Managers and employees of a branch have access to an accounts tab. You can add, edit, delete each account.

When adding, editing an account a pop up page will show where you can select an options of that account.
When deleting an account, a alert message will show asking you if you want to delete the account.

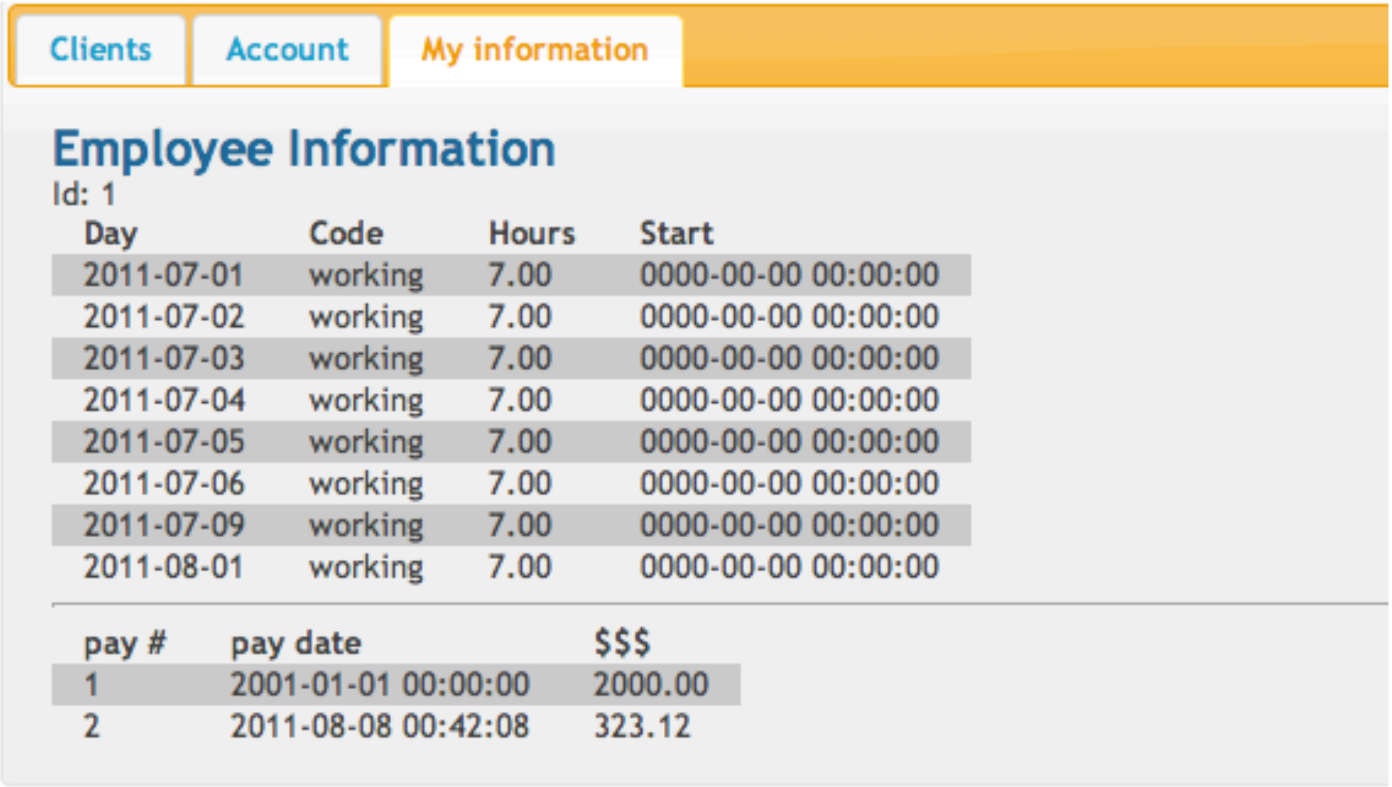


location	month	FeesReceived	InterestReceived	PayGivenToEmployees	Profit
Laval	2011-01	0.000	0.00	0.00	0.000
Laval	2011-02	0.000	0.00	0.00	0.000
Laval	2011-03	0.000	0.00	0.00	0.000
Laval	2011-04	0.000	0.00	0.00	0.000
Laval	2011-05	0.000	0.00	0.00	0.000
Laval	2011-06	0.000	0.00	0.00	0.000
Laval	2011-07	0.000	0.00	0.00	0.000
Laval	2011-08	419.000	8304246.10	16966.18	8287698.920
Laval	2011-09	0.000	0.00	0.00	0.000
Laval	2011-10	0.000	0.00	0.00	0.000
Laval	2011-11	0.000	0.00	0.00	0.000
Laval	2011-12	0.000	0.00	0.00	0.000
Laval	TOTAL	419.000	8304246.10	16966.18	8287698.920
Montreal	2011-01	0.000	0.00	0.00	0.000
Montreal	2011-02	0.000	0.00	0.00	0.000
Montreal	2011-03	0.000	0.00	0.00	0.000
Montreal	2011-04	0.000	0.00	0.00	0.000
Montreal	2011-05	0.000	0.00	0.00	0.000
Montreal	2011-06	0.000	0.00	0.00	0.000
Montreal	2011-07	0.000	0.00	0.00	0.000
Montreal	2011-08	84.000	-2198.64	6640.13	-8754.770
Montreal	2011-09	0.000	0.00	0.00	0.000
Montreal	2011-10	0.000	0.00	0.00	0.000
Montreal	2011-11	0.000	0.00	0.00	0.000
Montreal	2011-12	0.000	0.00	0.00	0.000

View of banking reports 1

See my schedule and pay

when you login as a employee or manager, under my information tab you will find, schedule and pay stub information.



Employee Information

Id: 1

Day	Code	Hours	Start
2011-07-01	working	7.00	0000-00-00 00:00:00
2011-07-02	working	7.00	0000-00-00 00:00:00
2011-07-03	working	7.00	0000-00-00 00:00:00
2011-07-04	working	7.00	0000-00-00 00:00:00
2011-07-05	working	7.00	0000-00-00 00:00:00
2011-07-06	working	7.00	0000-00-00 00:00:00
2011-07-09	working	7.00	0000-00-00 00:00:00
2011-08-01	working	7.00	0000-00-00 00:00:00

pay #	pay date	$$$
1	2001-01-01 00:00:00	2000.00
2	2011-08-08 00:42:08	323.12

List of members contribution

What	Who
Change model to 3NF	justin
Account ADD FE	mathieu
Account MOD FE	mathieu
Account DEL FE	mathieu
Account ADD BE	mathieu
Account MOD BE	mathieu
Employee DEL BE	ramez
Employee ADD FE	ramez
Employee MOD FE	ramez
Employee DEL FE	ramez
Employee ADD BE	ramez
Employee MOD BE	ramez
Employee DEL BE	ramez
Client information FE	wadih
Client information BE	wadih
employee information FE	wadih
employee informaiton BE	wadih
employee schedule maker FE	mathieu
employee schedule maker BE	mathieu
employee schedule viewer BE	mathieu
employee schedule viewer FE	mathieu
monthly fees BE	mathieu

monthly interest BE	mathieu
calculate profit per branch BE+FE	mathieu
client add fe	mathieu, ramez
Update ER accordingly	justin
Creating Client, Employee HTML Forms	justin
client mod fe	mathieu, ramez
client delete fe	mathieu
client add be	mathieu, ramez
client mod be	mathieu
client delete be	mathieu
Client View Accounts	Ramez
Clients Accounts Transfer	mathieu
Clients Transactions	Ramez, mathieu
Client Withdraw, Deposit	Ramez
Link client with account, BE+FE	Mathieu
Server set-up	Mathieu

*fe => Front End
*be => Back End

Installation Guide

1. Obtain the source code by using the instruction given in the section “How-to access source code”.
2. Browse to the database folder and locate the most recent dump.
3. Execute the following command:
   ```
   mysql5 -h clipper.encs.concordia.ca -u dmc353_1 -p dmc353_1 < final.sql
   ```

   At the prompt, enter the password L0ckL0ck

4. Execute the following command:
   ```
   mysql5 -h clipper.encs.concordia.ca -u dmc353_1 -p dmc353_1 < views.sql
   ```

   At the prompt, enter the password L0ckL0ck

5. Move the files from the code folder to the base directory for the website.
   ```
   mv code/* /www/groups/d/dm_comp353_1/
   ```

```
mv code/.htaccess /www/groups/d/dm_comp353_1/
```

6. Edit application/config/database.php to make sure the database credentials are set-up properly

```
$db['default']['hostname'] = 'clipper.encs.concordia.ca:3306';
$db['default']['username'] = 'dmc353_1';
$db['default']['password'] = 'L0ckL0ck';
```

7. Access https://clipper.encs.concordia.ca/cgi-bin/cgiwrap/~dmc353_1/index.php in your browser. username/password are dmc353_1:L0ckL0ck. If everything goes well, you should see the login screen.

8. As a bank we like to do money and our employee require to get paid once per month as specified in their contract. We need to set-up a job for this.

9. type `crontab -e`

   add the following line

```
0 0 0 * * mysql5 -h clipper.encs.concordia.ca -udmc353_1 -pL0ckL0ck < monthlybookkeeping.sql
```

   This will run the bank book-keeping program every first second of every month. This has for effect of paying the employees, charging for the option plans and charging/paying interest.

How-to access source code

Start a terminal session on a unix-compatible machine with access to the Internet
type the following command

```
wget https://github.com/matdumsa/comp353/tarball/master comp353.tar.gz --no-check-certificate
tar -xvfz master
```

This will download the latest version of our project on your computer and extract its content in a folder named matdumsa-comp353-#build number. Enter that directory with cd

The folder is organised as follow:

1. code (the main php application, should be put in the root public folder hosting the website).
   Note: *this site uses code igniter, wich comes with a "way of organising" an application aka MVC. Our code is mostly located in application/controller, application/model, application/view, js/ and css/.*
2. database (various dumps of the database, the views contained in it and the bank bookkeeping program)
3. documentation (documentation for the application, most recent version of this file)