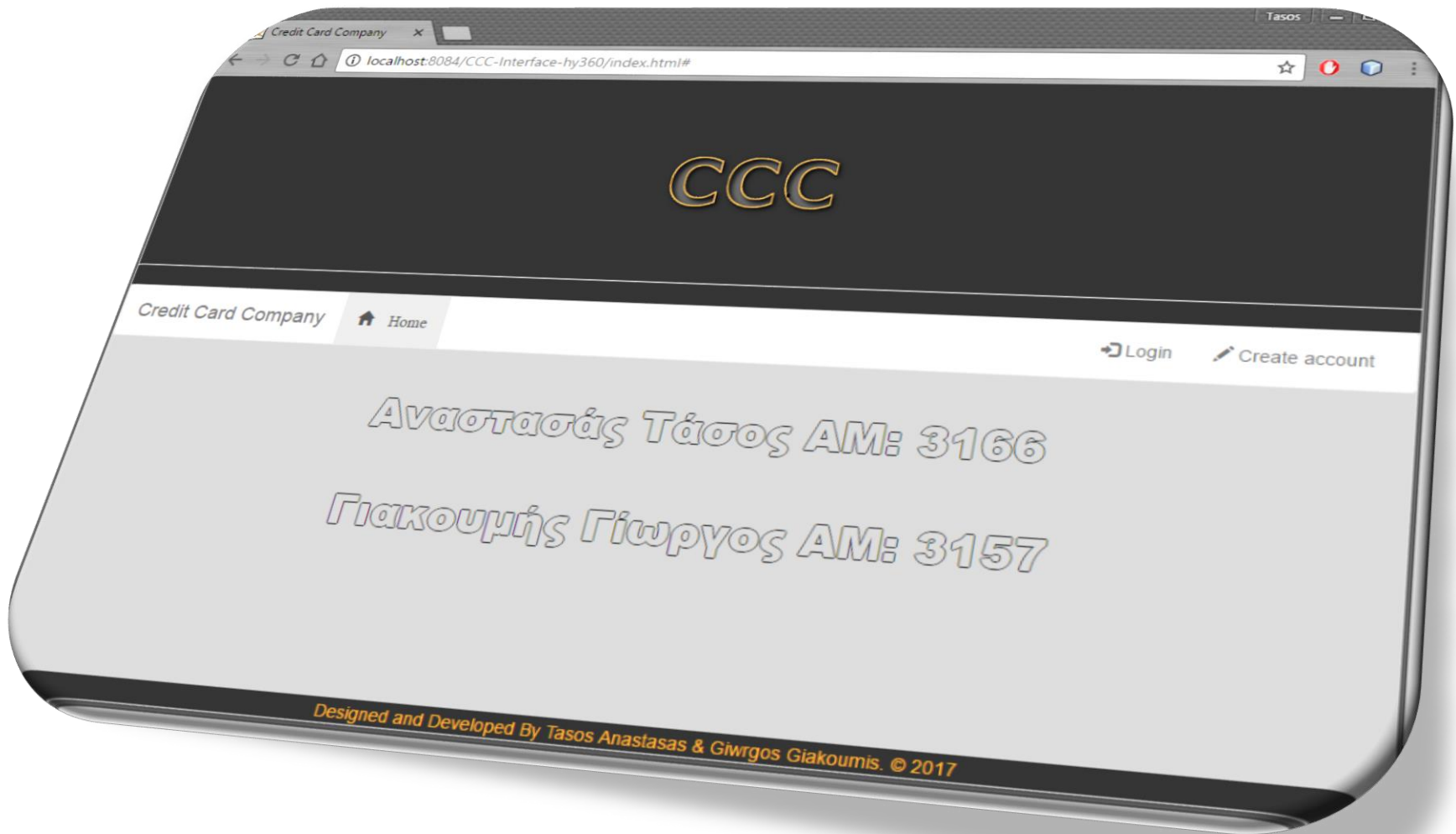
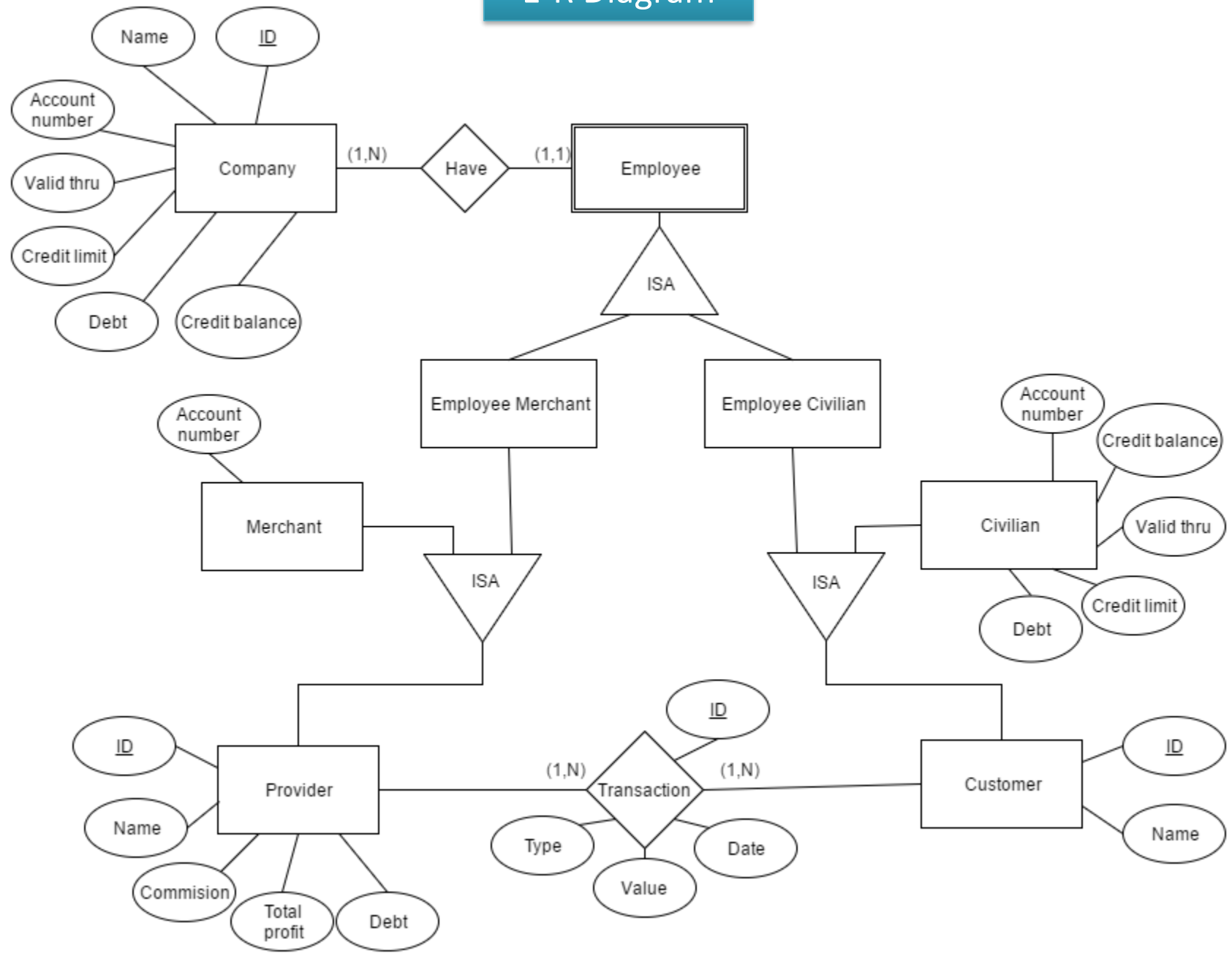


HY-360 Project Report



E-R Diagram



Γνωρίσματα με τους τύπους τους για κάθε οντότητα

Company:	(ID, varchar(30)), (Name, varchar(20)), (Account number, int), (Valid thru, varchar(10)), (Credit limit, real), (Credit balance, real)
Merchant:	(Account number, int),
Civilian:	(Account number, int), (Valid thru, varchar(10)), (Credit limit, real), (Debt, real), (Credit balance, real)
Provider:	(ID, varchar(30)), (Name, varchar(20)), (Total profit, real), (Commission, real), (Debt, real)
Customer :	(ID, varchar(30)), (Name, varchar(20))
Transaction:	((Transaction_ID, int), (Date, date), (Type, enum('charge', 'credit')), (Value, real)

Πρωτεύοντα κλειδιά στο E-R

- Company: primary key (ID)
- Provider: primary key (ID)
- Customer: primary key (ID)
- Transaction: primary key (ID)

Επεξηγήσεις

- Στις οντότητες Customer, Provider και Company προσθέσαμε το γνώρισμα “ID” το οποίο θα χρησιμοποιείτε ως μοναδικό κλειδί για να γίνεται login (το ID δεν είναι ανεξάρτητο στις 3 οντότητες). Σαν ID θα χρησιμοποιήσουμε ένα email που είναι εξ ορισμού μοναδικό.
- Προσθέσαμε ακόμα ένα “ID” στην σχέση Transaction για να μπορούμε να επιτρέπουμε πολλαπλές συναλλαγές μεταξύ του ίδιου Customer & Provider. Σαν ID θα δημιουργείται ένας τυχαίος και μοναδικός αριθμός.
- Θεωρούμε πως οι υπάλληλοι δεν μπορούν να πληρώσουν χρέος, γιατί έχουν το λογαριασμό της εταιρίας οπότε το χρέος θα το πληρώνει η εταιρία.

Σχεσιακό Μοντέλο

Company

Name	<u>ID</u>	Account number	Valid thru	Credit limit	Debt	Credit balance
------	-----------	----------------	------------	--------------	------	----------------

Employee Merchant

Name	<u>ID</u>	<u>Company_ID</u>	Total_profit	Commission	Debt
------	-----------	-------------------	--------------	------------	------

Employee Civilian

Name	<u>ID</u>	Company_ID
------	-----------	------------

Merchant

Name	<u>ID</u>	Account number	Commission	Total profit	Debt
------	-----------	----------------	------------	--------------	------

Civilian

Name	<u>ID</u>	Account number	Valid thru	Credit limit	Debt	Credit balance
------	-----------	----------------	------------	--------------	------	----------------

Merchant – Transaction – Civilian

<u>Merchant_ID</u>	<u>Civilian_ID</u>	Value	Type	Date	<u>Transaction_ID</u>
--------------------	--------------------	-------	------	------	-----------------------

Merchant – Transaction – Employee Civilian

<u>Merchant_ID</u>	<u>Employee_Civilian_ID</u>	<u>Company_ID</u>	Value	Type	Date	<u>Transaction_ID</u>
--------------------	-----------------------------	-------------------	-------	------	------	-----------------------

Employee Merchant – Transaction – Civilian

<u>Employee_Merchant_ID</u>	<u>Company_ID</u>	<u>Civilian_ID</u>	Value	Type	Date	<u>Transaction_ID</u>
-----------------------------	-------------------	--------------------	-------	------	------	-----------------------

Employee Merchant – Transaction – Employee Civilian

<u>Employee_Merchant_ID</u>	<u>Company_ID_Merchant</u>	<u>Employee_Civilian_ID</u>	<u>Company_ID_Civilian</u>	Value	Type	Date	<u>Transaction_ID</u>
-----------------------------	----------------------------	-----------------------------	----------------------------	-------	------	------	-----------------------

Συναρτησιακές Εξαρτήσεις

- Civilian

ID -> {Name, Account_number, Valid_thru, Credit_limit, Debt, Credit_balance}

Key: ID

- Γιατί το ID⁺ περιέχει το σύνολο των γνωρισμάτων της σχέσης Civilian

- Merchant

ID -> {Name, Account_number, Total_profit, Commission, Debt}

Key: ID

- Γιατί το ID⁺ περιέχει το σύνολο των γνωρισμάτων της σχέσης Merchant

- Company

ID -> {Name, Account_number, Valid_thru, Credit_limit, Current_debt, Credit_Balance}

Key: ID

- Γιατί το ID⁺ περιέχει το σύνολο των γνωρισμάτων της σχέσης Company

- Employee_Civilian

Customer_ID -> {Company_ID, Name}

Key: Customer_ID

- Γιατί το Customer_ID + περιέχει το σύνολο των γνωρισμάτων της σχέσης Employee_Civilian

- Employee_Merchant

Provider_ID -> {Company_ID, Name, Total_profit, Commission , Debt}

Key: Provider_ID

- Γιατί το Provider_ID + περιέχει το σύνολο των γνωρισμάτων της σχέσης Employee_Merchant

- Merchant-Transaction-Civilian

Transaction_ID-> {Merchant_ID, Civilian_ID, Type, Amount, Date}

Key: Transaction_ID

- Γιατί το Transaction_ID+ περιέχει το σύνολο των γνωρισμάτων της σχέσης
Merchant-Transaction-Civilian

- Merchant-Transaction-Employee Civilian

Transaction_ID-> {Merchant_ID, Employee_civilian_ID, Civilian_company_ID,
Type, Amount, Date}

Key: Transaction_ID

- Γιατί το Transaction_ID⁺ περιέχει το σύνολο των γνωρισμάτων της σχέσης
Merchant-Transaction-Employee Civilian
-

- Employee Merchant-Transaction-Civilian

Transaction_ID-> {Employee_merchant_ID, Merchant_company_ID, Civilian_ID,
Type, Amount, Date}

Key: Transaction_ID

- Γιατί το Transaction_ID⁺ περιέχει το σύνολο των γνωρισμάτων της σχέσης
Employee Merchant-Transaction-Civilian
-

- Employee Merchant-Transaction-Employee Civilian

Transaction_ID-> {Employee_merchant_ID, Merchant_company_ID, Employee_civilian_ID,
Civilian_company_ID, Type, Amount, Date}

Key: Transaction_ID

- Γιατί το Transaction_ID⁺ περιέχει το σύνολο των γνωρισμάτων της σχέσης
Employee Merchant-Transaction-Employee Civilian

Μετατροπή σε 3NF

Από τις συναρτησιακές εξαρτήσεις παρατηρούμε ότι όλες οι σχέσεις είναι σε BCNF. Αυτό προκύπτει από το γεγονός ότι κάθε αριστερό μέρος των συναρτησιακών εξαρτήσεων αποτελεί κλειδί για την εκάστοτε σχέση.

Επειδή είναι σε BCNF, θα είναι σίγουρα και σε 3NF.

Επομένως primary key στους πίνακες θα είναι αυτό που ορίζεται από τις συναρτησιακές εξαρτήσεις της εκάστοτε σχέσης.

Εντολές γλώσσας ορισμού δεδομένων

DROP TABLE IF EXISTS Civilian;

CREATE TABLE Civilian(

Name	varchar(20)	not null,
Debt	real	DEFAULT 0 not null,
Credit_balance	real	DEFAULT 2000 not null,
Credit_limit	real	DEFAULT 2000 not null,
Account_number	int	not null,
Valid_thru	date	not null,
ID	varchar(30)	not null,
PRIMARY KEY (ID));		

DROP TABLE IF EXISTS Merchant;

CREATE TABLE Merchant(

Name	varchar(20)	not null,
Total_profit	real	DEFAULT 0 not null,
Debt	real	DEFAULT 0 not null,
Commission	real	DEFAULT 3 not null,
Account_number	int	not null,
ID	varchar(30)	not null,
PRIMARY KEY (ID));		

```
DROP TABLE IF EXISTS Company;
```

```
CREATE TABLE Company(
```

```
    Name
```

```
    varchar(20)
```

```
    not null,
```

```
    Debt
```

```
    real
```

```
    DEFAULT 0 not null,
```

```
    Credit_balance
```

```
    real
```

```
    DEFAULT 20000 not null,
```

```
    Credit_limit
```

```
    real
```

```
    DEFAULT 20000 not null,
```

```
    Account_number
```

```
    int
```

```
    not null,
```

```
    Valid_thru
```

```
    date
```

```
    not null,
```

```
    ID
```

```
    varchar(30)
```

```
    not null,
```

```
    PRIMARY KEY ( ID ));
```

DROP TABLE IF EXISTS Employee_Merchant;

CREATE TABLE Employee_Merchant(

Name	varchar(20)	not null,
ID	varchar(30)	not null,
Company_ID	varchar(30)	not null,
Total_profit	real	DEFAULT 0 not null,
Commission	real	DEFAULT 1 not null,
Debt	real	DEFAULT 0 not null,

FOREIGN KEY (Company_ID) references Company(ID) ,
PRIMARY KEY (ID));

DROP TABLE IF EXISTS Employee_Civilian;

CREATE TABLE Employee_Civilian(

Name	varchar(20)	not null,
ID	varchar(30)	not null,
Company_ID	varchar(30)	not null,

FOREIGN KEY (Company_ID) references Company(ID) ,
PRIMARY KEY (ID));

```
DROP TABLE IF EXISTS Merchant_Transaction_Civilian;
CREATE TABLE Merchant_Transaction_Civilian(
  Merchant_ID          varchar(30)          not null,
  Civilian_ID          varchar(30)          not null,
  Transaction_ID       real                 not null,
  Value               real                 not null,
  Type                enum("charge", "credit") not null,
  Date               date                 not null,
  PRIMARY KEY (Transaction_ID));
```

```
DROP TABLE IF EXISTS Merchant_Transaction_eCivilian;
CREATE TABLE Merchant_Transaction_eCivilian (
  Merchant_ID          varchar(30)          not null,
  Employee_civilian_ID varchar(30)          not null,
  Transaction_ID       real                 not null,
  Value               real                 not null,
  Type                enum("charge", "credit") not null,
  Date               date                 not null,
  Civilian_company_ID  varchar(30)          not null,
  PRIMARY KEY (Transaction_ID));
```

```

DROP TABLE IF EXISTS eMerchant_Transaction_Civilian;
CREATE TABLE eMerchant_Transaction_Civilian(
  Employee_merchant_ID      varchar(30)      not null,
  Civilian_ID               varchar(30)      not null,
  Transaction_ID            real              not null,
  Value                     real              not null,
  Type                      enum("charge", "credit") not null,
  Date                      date              not null,
  Merchant_company_ID       varchar(30)      not null,
  PRIMARY KEY (Transaction_ID));

```

```

DROP TABLE IF EXISTS eMerchant_Transaction_eCivilian ;
CREATE TABLE eMerchant_Transaction_eCivilian (
  Employee_merchant_ID      varchar(30)      not null,
  Employee_civilian_ID      varchar(30)      not null,
  Transaction_ID            real              not null,
  Value                     real              not null,
  Type                      enum("charge", "credit") not null,
  Date                      date              not null,
  Civilian_company_ID       varchar(30)      not null,
  Merchant_company_ID       varchar(30)      not null,
  PRIMARY KEY (Transaction_ID));

```


SQL Queries

Περιλαμβάνονται όλα τα query που χρησιμοποιήθηκαν για βασικές και μη λειτουργίες. Όποιο στοιχείο έχει '_param' , δεν είναι συγκεκριμένο οπότε περνιέται σαν παράμετρος στο query.

- Διαγράφει έναν χρήστη

```
DELETE FROM table_param WHERE ID = 'id_param'
```

- Βρίσκει την εταιρία που έχει
έναν συγκεκριμένο υπάλληλο

```
SELECT c.id FROM company c, table_param e  
WHERE e.Company_ID = c.ID and e.ID = 'id_param'  
and e.Company_ID = 'companyID_param'
```

- Βρίσκει τον χρήστη με το συγκεκριμένο id

```
SELECT ID FROM table_param WHERE ID = 'id_param'
```

- Βρίσκει αν υπάρχει ένα συγκεκριμένο ID.
Χρησιμοποιείται στο Register για τον έλεγχο του email

```
SELECT ID FROM civilian WHERE ID = 'id_param'  
UNION  
SELECT ID FROM merchant WHERE ID = 'id_param'  
UNION  
SELECT ID FROM company WHERE ID = 'id_param'  
UNION  
SELECT ID FROM employee_civilian WHERE ID = 'id_param'  
UNION  
SELECT ID FROM employee_merchant WHERE ID = 'id_param'
```

- Βρίσκει όλους τους εμπόρους ή τους ιδιώτες, ανάλογα με το type

```
SELECT ID FROM 'type_param' UNION SELECT ID FROM employee_'type_param'
```

- Βρίσκει όλες τις εταιρίες

```
SELECT ID FROM company
```

- Βρίσκει μια συγκεκριμένη εταιρία

```
SELECT company_id FROM table_param WHERE ID = 'id_param'
```

- Βρίσκει τους εμπόρους με τους οποίους ένας συγκεκριμένος ιδιώτης έχει κάνει συναλλαγές, αντίστοιχα αν είναι ιδιώτης, υπάλληλος

```
SELECT employee_merchant_id AS ID FROM emerchant_transaction_civilian
WHERE Civilian_id = 'civilianID_param'
UNION
SELECT merchant_id AS ID FROM merchant_transaction_civilian
WHERE Civilian_id = 'civilianID_param'
```

- Βρίσκει τους χρήστες που έχουν χρέος προς τη CCC

```
SELECT ID,Debt FROM civilian WHERE Debt <> 0
UNION
SELECT ID,Debt FROM merchant WHERE Debt <> 0
UNION
SELECT ID,Debt FROM company WHERE Debt <> 0
ORDER BY debt
```

- Βρίσκει τους χρήστες που δεν έχουν χρέος προς τη CCC

```
SELECT ID FROM civilian WHERE Debt = 0
UNION
SELECT ID FROM merchant WHERE Debt = 0
UNION
SELECT ID FROM company WHERE Debt = 0
```

- Βρίσκει τα δεδομένα ενός χρήστη που δεν είναι υπάλληλος

```
SELECT * FROM table_param WHERE ID = 'id_param'
```

- Βρίσκει τα δεδομένα ενός χρήστη που είναι υπάλληλος

```
SELECT * FROM employee_table_param e ,company c
WHERE e.id = 'id_param' && c.ID = e.Company_ID
```

- Εισάγει έναν ιδιώτη (οι υπόλοιπες τιμές είναι default)

```
INSERT INTO table_param  
(NAME, ACCOUNT_NUMBER, VALID_THRU, ID)  
VALUES  
('name_param','account_param','date_param','id_param')
```

- Εισάγει έναν ιδιώτη, υπάλληλο

```
INSERT INTO table_param  
(NAME, ID, COMPANY_ID)  
VALUES  
('name_param','id_param','companyID_param')
```

- Εισάγει μια εταιρία

```
INSERT INTO table_param  
(NAME, ACCOUNT_NUMBER, VALID_THRU, ID)  
VALUES  
('name_param','account_param','date_param','companyID_param')
```

- Εισάγει έναν έμπορο

```
INSERT INTO table_param  
(NAME, ACCOUNT_NUMBER, ID)  
VALUES ('name_param', 'account_param','id_param')
```

- Εισάγει έναν έμπορο, υπάλληλο

```
INSERT INTO table_param  
(NAME, ID, COMPANY_ID)  
VALUES ('name_param','id_param','companyID_param')
```

- Κάνει μια δοσοληψία ανάμεσα σε έναν
έμπορο υπάλληλο και έναν ιδιώτη

```
INSERT INTO transTable_param  
(EMPLOYEE_MERCHANT_ID, CIVILIAN_ID, TRANSACTION_ID, VALUE,  
TYPE, DATE, MERCHANT_COMPANY_ID)  
VALUES  
('merchantID_param','civilianID_param','transactionID_param','value_param',  
'type_param','date_param','merchantCompanyID_param')
```

- Κάνει μια δοσοληψία ανάμεσα σε έναν έμπορο υπάλληλο και έναν ιδιώτη, υπάλληλο

```
INSERT INTO transTable_param  
(EMPLOYEE_MERCHANT_ID, EMPLOYEE_CIVILIAN_ID, TRANSACTION_ID, VALUE,  
TYPE, DATE, CIVILIAN_COMPANY_ID, MERCHANT_COMPANY_ID)  
VALUES  
( 'merchantID_param', 'civilianID_param', 'transactionID_param', 'value_param',  
'type_param', 'date_param', 'civilianCompanyID_param', 'merchantCompanyID_param')
```

-
- Κάνει μια δοσοληψία ανάμεσα σε έναν έμπορο και έναν ιδιώτη

```
INSERT INTO transTable_param  
(MERCHANT_ID, CIVILIAN_ID, TRANSACTION_ID, VALUE, TYPE, DATE)  
VALUES  
( 'merchantID_param', 'civilianID_param', 'transactionID_param',  
'value_param', 'type_param', 'date_param')
```

- Κάνει μια δοσοληψία ανάμεσα σε έναν έμπορο και έναν ιδιώτη, υπάλληλο

```
INSERT INTO transTable_param  
(MERCHANT_ID, EMPLOYEE_CIVILIAN_ID, TRANSACTION_ID, VALUE,  
TYPE, DATE, CIVILIAN_COMPANY_ID)  
VALUES  
('merchantID_param','civilianID_param','transactionID_param','value_param',  
'type_param','date_param','civilianCompanyID_param')
```

-
- Βρίσκει όλους τους υπαλλήλους μιας εταιρίας

```
SELECT id FROM employee_civilian  
WHERE company_id = 'companyID_param'  
UNION  
SELECT id FROM employee_merchant  
WHERE company_id = 'companyID_param'
```

- Βρίσκει και ταξινομεί, με φθίνουσα σειρά, τους εμπόρους με το μεγαλύτερο κέρδος ανάμεσα σε 2 ημερομηνίες που εμείς ορίζουμε.

```
SELECT allTransactions.merchant AS ID, SUM(allTransactions.valueSum) AS profit FROM
  (SELECT t.Merchant_ID AS merchant, SUM(t.value) as valueSum FROM merchant_transaction_civilian t
    WHERE t.TYPE = 'charge' AND t.DATE BETWEEN 'minDate_param' AND 'maxDate_param' GROUP BY t.merchant_id
  UNION
  SELECT t.Merchant_ID AS merchant, SUM(-t.value) as valueSum FROM merchant_transaction_civilian t
    WHERE t.TYPE = 'credit' AND t.DATE BETWEEN 'minDate_param' AND 'maxDate_param' GROUP BY t.merchant_id
  UNION
  SELECT t.Merchant_ID AS merchant, SUM(t.value) as valueSum FROM merchant_transaction_ecivilian t
    WHERE t.TYPE = 'charge' AND t.DATE BETWEEN 'minDate_param' AND 'maxDate_param' GROUP BY t.merchant_id
  UNION
  SELECT t.Merchant_ID AS merchant, SUM(-t.value) as valueSum FROM merchant_transaction_ecivilian t
    WHERE t.TYPE = 'credit' AND t.DATE BETWEEN 'minDate_param' AND 'maxDate_param' GROUP BY t.merchant_id
  UNION
  SELECT t.Employee_Merchant_ID AS merchant, SUM(t.value) as valueSum FROM emerchant_transaction_civilian t
    WHERE t.TYPE = 'charge' AND t.DATE BETWEEN 'minDate_param' AND 'maxDate_param'
    GROUP BY t.Employee_Merchant_ID
  UNION
  SELECT t.Employee_Merchant_ID AS merchant, SUM(-t.value) as valueSum FROM emerchant_transaction_civilian t
    WHERE t.TYPE = 'credit' AND t.DATE BETWEEN 'minDate_param' AND 'maxDate_param'
    GROUP BY t.Employee_Merchant_ID
  UNION
  SELECT t.Employee_Merchant_ID AS merchant, SUM(t.value) as valueSum FROM emerchant_transaction_ecivilian t
    WHERE t.TYPE = 'charge' AND t.DATE BETWEEN 'minDate_param' AND 'maxDate_param'
    GROUP BY t.Employee_Merchant_ID
  UNION
  SELECT t.Employee_Merchant_ID AS merchant, SUM(-t.value) as valueSum FROM emerchant_transaction_ecivilian t
    WHERE t.TYPE = 'credit' AND t.DATE BETWEEN 'minDate_param' AND 'maxDate_param'
    GROUP BY t.Employee_Merchant_ID)
AS allTransactions
GROUP BY allTransactions.merchant
ORDER BY profit DESC
```


- Βρίσκει με ποιούς έχει κάνει έστω και μια συναλλαγή
ένας έμπορος υπάλληλος

```
SELECT employee_civilian_id AS ID FROM emerchant_transaction_ecivilian  
WHERE employee_merchant_id = 'id_param'  
UNION  
SELECT civilian_id FROM emerchant_transaction_civilian  
WHERE employee_merchant_id = 'id_param'
```

- Βρίσκει με ποιούς έχει κάνει έστω και μια συναλλαγή
ένας έμπορος (όχι υπάλληλος)

```
SELECT employee_civilian_id AS ID from merchant_transaction_ecivilian  
WHERE merchant_id = 'id_param'  
UNION  
SELECT civilian_id FROM merchant_transaction_civilian  
WHERE merchant_id = 'id_param'
```

- Μειώνει το χρέος ενός εμπόρου κατα 5%

```
UPDATE merchant_table_param  
SET debt = debt - debt*5/100 WHERE ID = 'merchantID_param'
```

- Βρίσκει τους υπαλλήλους μια εταιρίας

```
SELECT id FROM employee_civilian WHERE company_id = 'companyID_param'  
UNION  
SELECT id FROM employee_merchant WHERE company_id = 'companyID_param'
```

- Βρίσκει τους ιδιώτες με τους οποίους έχει κάνει έστω και μια συναλλαγή ένας συγκεκριμένος έμπορος, υπάλληλος

```
SELECT employee_civilian_id as ID from emerchant_transaction_ecivilian  
WHERE employee_merchant_id = 'merchant_id_param'  
UNION  
SELECT civilian_id from emerchant_transaction_civilian  
WHERE employee_merchant_id = 'merchant_id_param'
```

- Βρίσκει τους ιδιώτες με τους οποίους έχει κάνει έστω και μια συναλλαγή ένας συγκεκριμένος έμπορος

```
SELECT employee_civilian_id as ID from merchant_transaction_ecivilian  
WHERE merchant_id = 'merchant_id_param'  
UNION SELECT civilian_id from merchant_transaction_civilian  
WHERE merchant_id = 'merchant_id_param'
```

- Βρίσκει όλους τους πελάτες ενός συγκεκριμένου τύπου (εμπόρους, ιδιώτες)

```
SELECT id FROM type_param UNION SELECT id FROM employee_type_param
```

- Βρίσκει όλους τους υπαλλήλους ενός συγκεκριμένου τύπου (εμπόρους, ιδιώτες), για μια συγκεκριμένη εταιρία

```
SELECT id FROM employee_table_param WHERE company_id = 'companyID_param'
```

- Ένας ιδιώτης πληρώνει κάποιο απο το ποσό οφειλής του

```
UPDATE table_param  
SET Credit_balance = Credit_balance + value_param , Debt = Debt - value_param  
WHERE ID = 'civilianID_param'  
AND Credit_limit >= Credit_balance + value_param AND Debt >= value_param
```

- Ένας έμπορος πληρώνει κάποιο απο το ποσό οφειλής του

```
UPDATE table_param  
SET Total_profit = Total_profit - value_param , Debt = Debt - value_param  
WHERE ID = 'merchantID_param' AND Debt > value_param AND Total_profit > value_param
```

- Ενας ιδιώτης αγοράζει προϊόντα

```
UPDATE table_param  
SET Credit_balance = Credit_balance - value_param, Debt = Debt + value_param  
WHERE ID = 'civilianID_param' AND Credit_balance >= value_param  
AND Valid_thru >= 'date_param'
```

- Ενας ιδιώτης επιστρέφει κάποιο προϊόν

```
UPDATE table_param  
SET Credit_balance = Credit_balance + value_param, Debt = Debt - value_param  
WHERE ID = 'civilianID_param' AND Debt >= value_param  
AND Valid_thru >= 'date_param'
```

- Αγοράζονται κάποια προϊόντα από έναν έμπορο

```
UPDATE merchant  
SET Total_profit = Total_profit + value_param , Debt = Debt + value_param *Commission/100  
WHERE ID = 'merchantID_param'
```

- Επιστρέφονται κάποια προϊόντα από σε έναν έμπορο

```
UPDATE table_param  
SET Total_profit = Total_profit - value_param + value_param *Commission/100  
WHERE ID = 'merchantID_param' AND Total_profit >= value_param
```

Instructions to run project (servlets)

- First start Apache and MySQL servers from XAMPP
(DB name **must** be 'hy360' with username 'root' and no password)
- Then Download the last version of **NetBeans IDE**
 - <http://netbeans.org/downloads/>
 - Choose a version that contains the Apache Tomcat server and Java EE

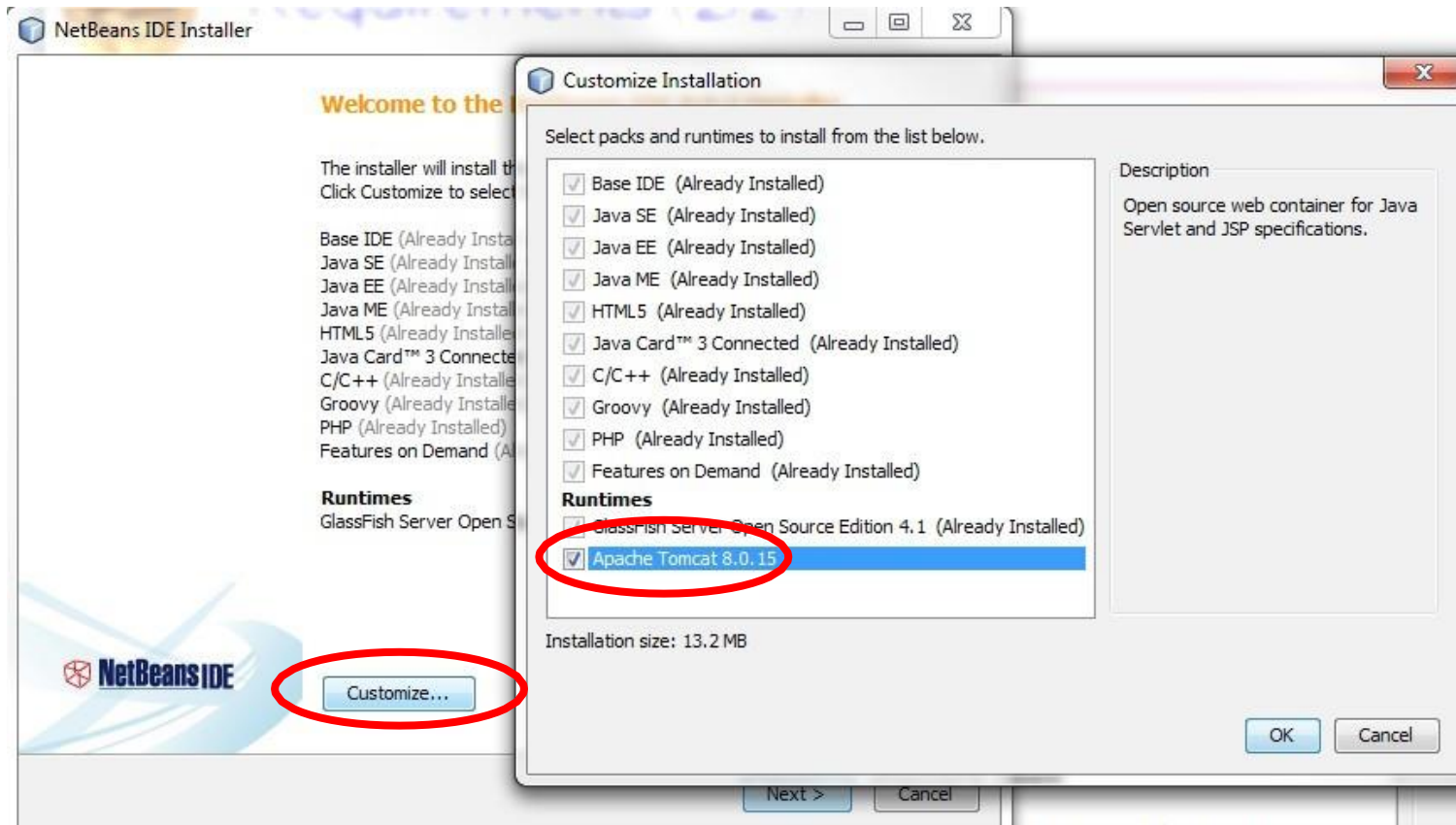
NetBeans IDE Download Bundles

Supported technologies *	Java SE	Java EE	HTML5/JavaScript	PHP	C/C++	All
NetBeans Platform SDK	•	•				•
Java SE	•	•				•
Java FX	•	•				•
Java EE		•				•
Java ME						•
HTML5/JavaScript		•	•	•		•
PHP			•	•		•
C/C++					•	•
Groovy						•
Java Card™ 3 Connected						•
Bundled servers						
GlassFish Server Open Source Edition 4.1.1		•				•
Apache Tomcat 8.0.27		•				•

Download	Download	Download x86	Download x86	Download x86	Download
		Download x64	Download x64	Download x64	
Free, 95 MB	Free, 197 MB	Free, 108 - 112 MB	Free, 108 - 112 MB	Free, 107 - 110 MB	Free, 221 MB

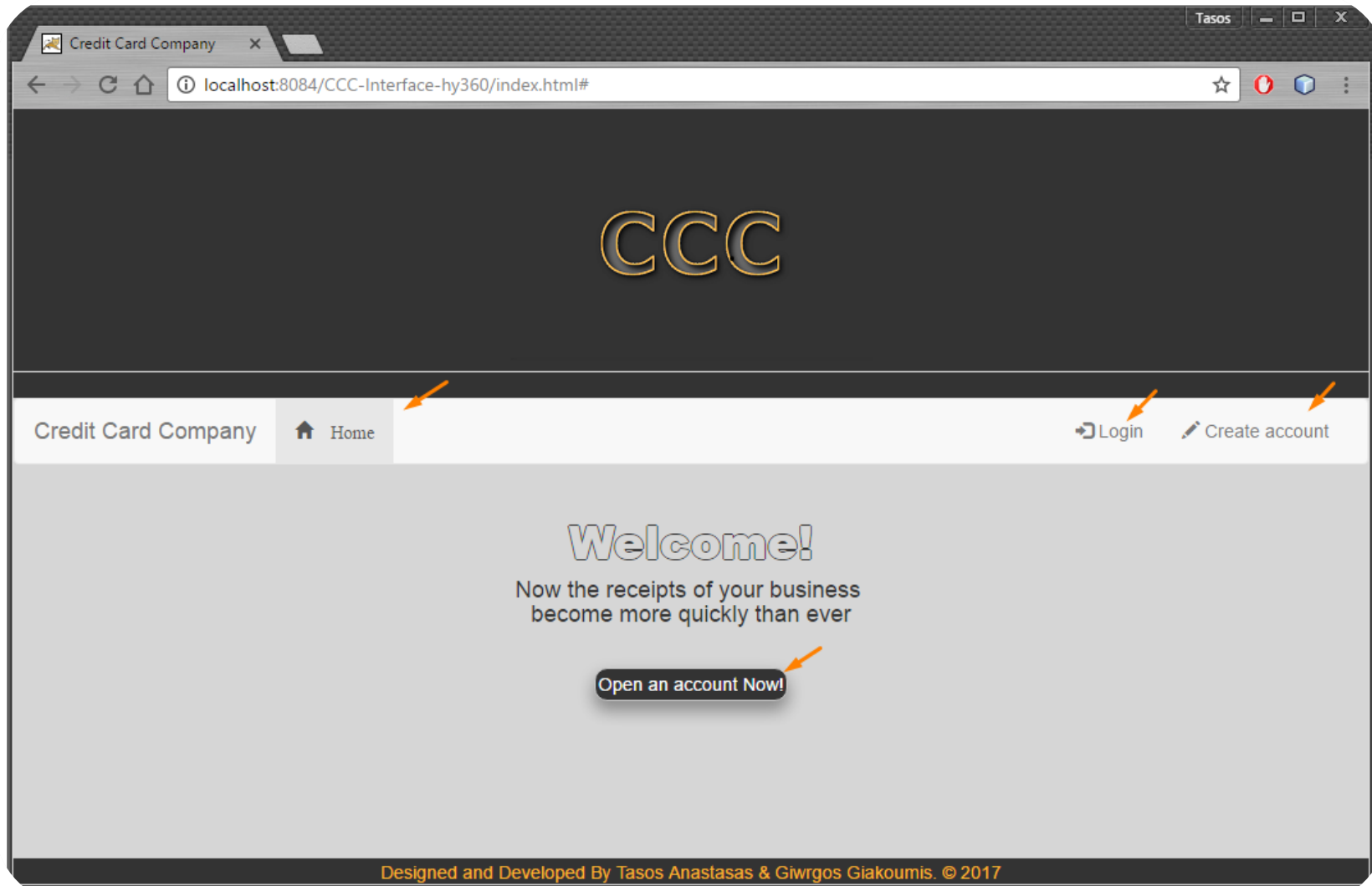


- From the installer, click **Customize** and select the **Apache Tomcat Runtime**
- Continue the installation



- Open **Services** tab
 - Windows → Services
- Expand **Servers**
- If there is not Apache Tomcat in the servers:
 - Right click → Add Server → Apache Tomcat
 - At Server Location, browse the installation folder
(e.g. C:\Program Files\Apache Software Foundation\Apache Tomcat 8.0.15)
 - Check the option “Use a Private Configuration Folder (Catalina base)” and choose a folder (you can choose any folder in your computer)
 - Give a username and password (e.g. username=admin, password=admin)
 - Check the option “Create user if does not exist”
 - Click Finish
- Go (File → Open Project) and open the submitted ‘CCC-Interface-hy360’
- Run project (Tomcat should start automatically)
- We have provided a sample DB with some users and transactions in folder easeOfUse

Quick Guide of Use



- Άνοιγμα ενός λογαριασμού

The screenshot shows a web browser window with the URL `localhost:8084/CCC-Interface-hy360/#`. The page features the 'Credit Card Company' logo and navigation links for 'Home', 'Login', and 'Create account'. The main content is a 'Registration form' with the following fields and annotations:

- Name:** A text input field containing 'Company1'. It is labeled 'STEP 1:' and has a green checkmark icon next to it, indicating validity.
- Email:** A text input field containing 'company@ccc.gr'. It is labeled 'STEP 2:' and has a red asterisk icon next to it, indicating an error.
- Role:** A dropdown menu with 'Company' selected. It is labeled 'STEP 3' and has a red asterisk icon next to it, indicating an error.
- Open Account:** A button labeled 'Open Account'. It is labeled 'STEP 4:' and has an orange arrow pointing to it.

Additional annotations include a 'Validity' label with a red asterisk icon pointing to the email field, and a red asterisk icon pointing to the role dropdown menu.

Designed and Developed By Tasos Anastasas & Giwrgos Giakoumis. © 2017

- Κατά την είσοδο επιλέγουμε τον βασικό τύπο λογαριασμού μας και στην συνέχεια αν είμαστε υπάλληλοι μιας εταιρίας.

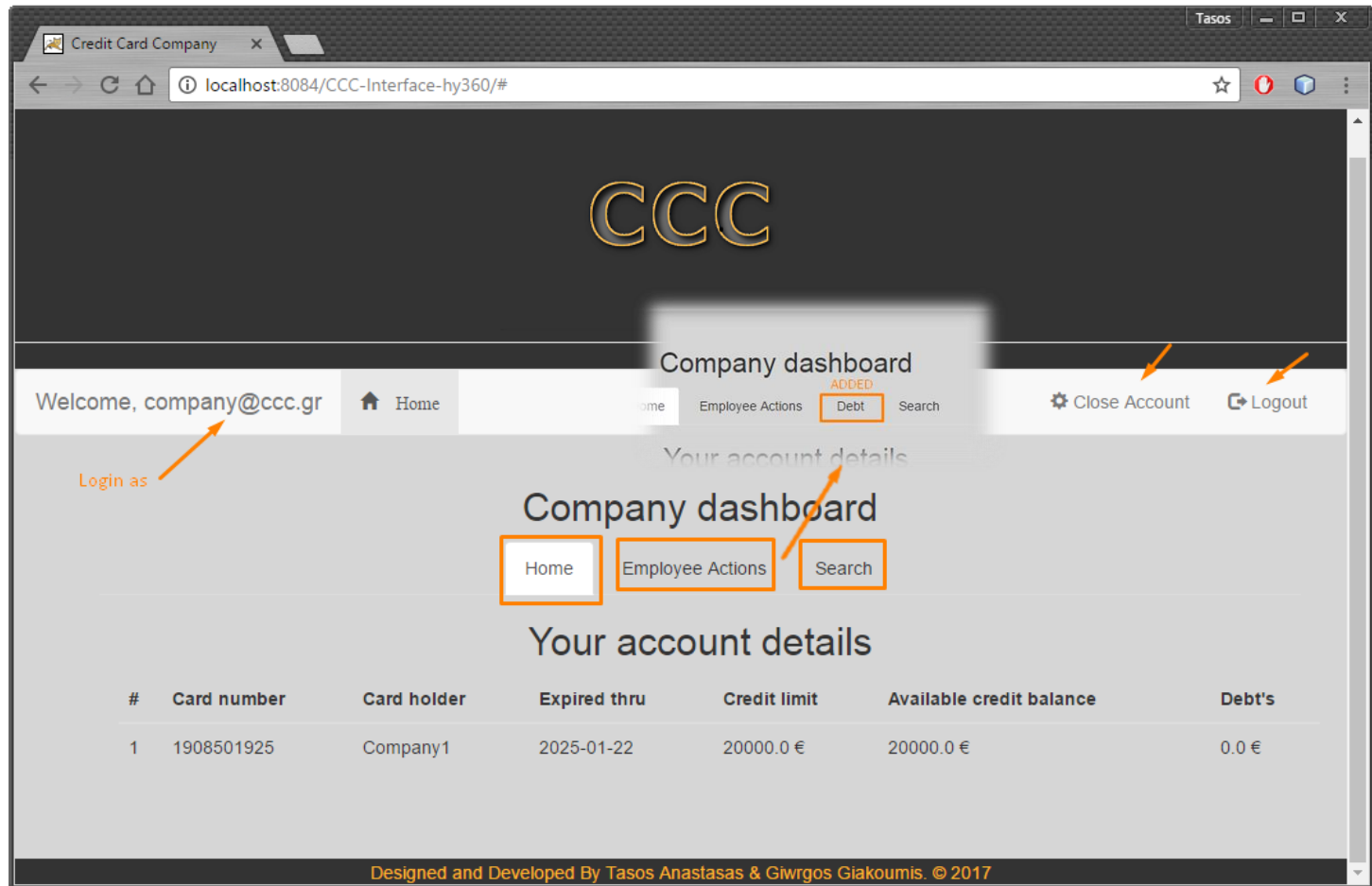
The screenshot shows a web browser window titled "Credit Card Company" with the URL `localhost:8084/CCC-Interface-hy360/index.html#`. The page features a large "CCC" logo at the top. Below the logo, there is a navigation bar with "Credit Card Company", a "Home" link, and "Login" and "Create account" buttons. The main content area is titled "Login" and contains the following form elements:

- Email:** `em1@ccc.gr` (Annotated with **STEP 1:**)
- Merchant** dropdown menu (Annotated with **STEP 2:**)
 - Working on Company
 - ☐ Yes
 - ☒ No (Annotated with **STEP 3:**)
 - Valid email (Annotated with **STEP 4:**)
- Login** button

An orange arrow points from the "Merchant" dropdown to a separate dropdown menu on the left, which lists "Company", "Company", "Customer", and "Merchant". The "Company" option is highlighted.

At the bottom of the page, it says: "Designed and Developed By Tasos Anastasas & Giwrgos Giakoumis. © 2017"

- Μετά την είσοδο στην σελίδα εμφανίζεται ένα dashboard για κάθε χρήστη με τις λειτουργίες που μπορεί να επιτελέσει. Επίσης φαίνονται και τα στοιχεία του λογαριασμού του. Στην περίπτωση που είναι employee εμφανίζονται και τα στοιχεία της εταιρίας



- Στο τέλος της σελίδας σε κάθε χρήστη εμφανίζεται 3 λίστες με τους καλούς, κακούς πελάτες της CCC καθώς και ο/οι καλύτερος/οι έμπορος/οι του προηγούμενου μήνα.

The screenshot shows a web browser window with the URL `localhost:8084/CCC-Interface-hy360/index.html`. The page is titled "Company dashboard" and includes a navigation bar with "Home", "Employee Actions", and "Search". Below the navigation bar, there is a section titled "Your account details" which contains a table with the following data:

#	Card number	Card holder	Expired thru	Credit limit	Available credit balance	Debt's
1	1908501925	Company1	2025-01-22	20000.0 €	20000.0 €	0.0 €

Below the account details, there is a section titled "CCC Info about customers" which is highlighted with an orange border. This section contains three sub-sections:

- The Good CCC Customers**: A table with one row showing ID 1 for company@ccc.gr.
- The Bad CCC Customers**: A table with one row showing ID 1 for Nobody with a debt of 0.
- Best Merchant of the previous month**: A table with one row showing ID 1 for Nobody.

The footer of the page states: "Designed and Developed By Tasos Anastasas & Giwrgos Giakoumis. © 2017".

- Κατά το submit εμφανίζεται κατάλληλο alert box για το εάν έγινε ή υπήρξε κάποιο πρόβλημα με την εγγραφή του υπαλλήλου

The screenshot shows a web browser window with the URL `localhost:8084/CCC-Interface-hy360/#`. The page is titled "Company dashboard" and includes a navigation bar with "Home", "Employee Actions", and "Search". The main content area contains a form for adding or removing an employee. The form fields are as follows:

- Account number: `1908501925` (Status: `Can't modified`)
- Employee id: `em1@ccc.gr` (STEP 1: ✓)
- Employee name: `Employee1` (STEP 2: ✓)
- Account type: `Customer` (STEP 3: ✓)
- Action: `Add` (STEP 4: ✓)
- Submit button (STEP 5: ✓)

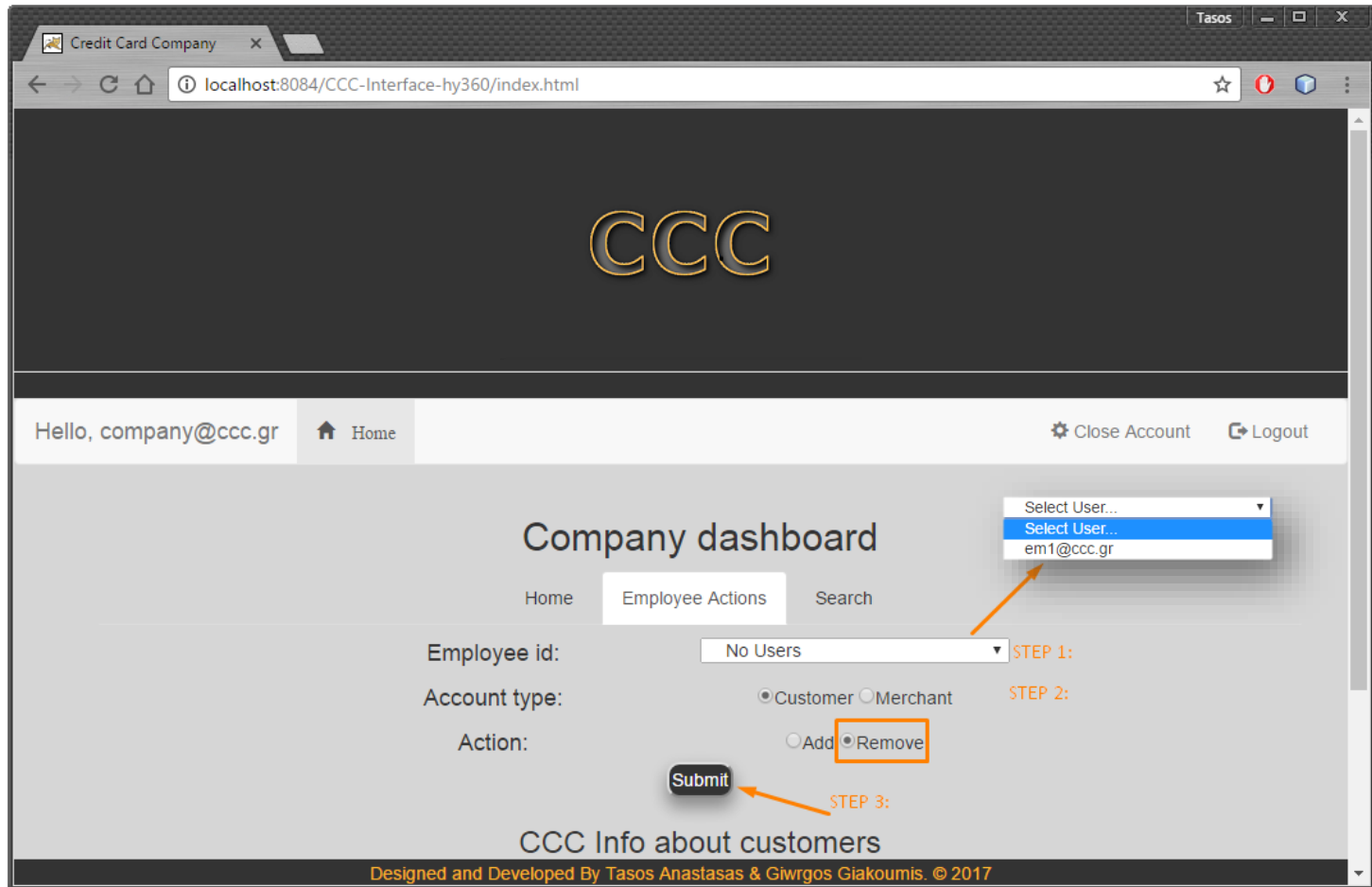
Annotations on the right side of the form indicate the status of each step:

- STEP 1: ✓ Checks for validity
- STEP 2: ✓ and ID exists
- STEP 3: ✓
- STEP 4: ✓
- STEP 5: ✓

An alert box is visible in the top right corner with the message "Email Already Exist" and a green checkmark.

Designed and Developed By Tasos Anastasas & Giwrgos Giakoumis. © 2017

- Στο Βήμα 2 όταν επιλέγουμε/αλλάζουμε τύπο φορτώνονται και τα κατάλληλα IDs (Όπως φαίνεται και στην εικόνα πανω από το βήμα 1)



* At Account type, 'Customer' has renamed to 'Civilian'

- Τα βήματα είναι όλα προαιρετικά. Όταν δεν επιλέξουμε κάποιο συγκεκριμένο user (βημα 1) τότε επιλέγονται όλοι οι πιθανοί χρήστες.

The screenshot displays the 'Credit Card Company' web application. The browser address bar shows 'localhost:8084/CCC-Interface-hy360/index.html'. The user is logged in as 'company@ccc.gr'. The main content area is titled 'Company dashboard' and includes a 'Transaction search' section. This section contains several search filters, each with a numbered step label:

- STEP 1: Transactions between: Select User... (dropdown)
- STEP 2: with (dropdown)
- STEP 3: Transaction type: Both (dropdown)
- STEP 4: Value: None (dropdown)
- STEP 5: (text input)
- STEP 6: Value between: min (text input)
- STEP 7: max (text input)
- STEP 8: Date between: mm/dd/yyyy (text input)
- STEP 9: mm/dd/yyyy (text input)
- STEP 10: Search now (button)

Below the search filters, a 'Search Results' table is visible, showing a list of transactions with columns for Merchant ID, Civilian ID, Transaction Value, Transaction Type, and Transaction Date.

#	Merchant ID	Civilian ID	Transaction Value	Transaction Type	Transaction Date
1	em3@ccc.gr	c2@ccc.gr	400	credit	2017-01-22
2	em3@ccc.gr	c1@ccc.gr	150	charge	2017-01-22
3	em1@ccc.gr	c1@ccc.gr	200	charge	2017-01-22
4	em3@ccc.gr	c2@ccc.gr	500	charge	2017-01-22
5	em3@ccc.gr	ec1@ccc.gr	100	credit	2017-01-22
6	em3@ccc.gr	ec1@ccc.gr	200	charge	2017-01-22

At the bottom of the page, a footer reads: 'Designed and Developed By Tasos Anastasas & Giwrgos Giakoumis. © 2017'.

-Στο βήμα 1 εμφανίζονται όλοι οι έμποροι από τους οποίους μπορεί να αγοράσει αγαθά ο ιδιώτης.

Credit Card Company

localhost:8084/CCC-Interface-hy360/index.html#

Hello, c1@ccc.gr Home Close Account Logout

Customer dashboard

Home Buy Debt Refund Search

Buy goods

Merchant: m2@ccc.gr Select User... m1@ccc.gr m2@ccc.gr m3@ccc.gr em1@ccc.gr em2@ccc.gr

Merchant: m2@ccc.gr STEP 1:

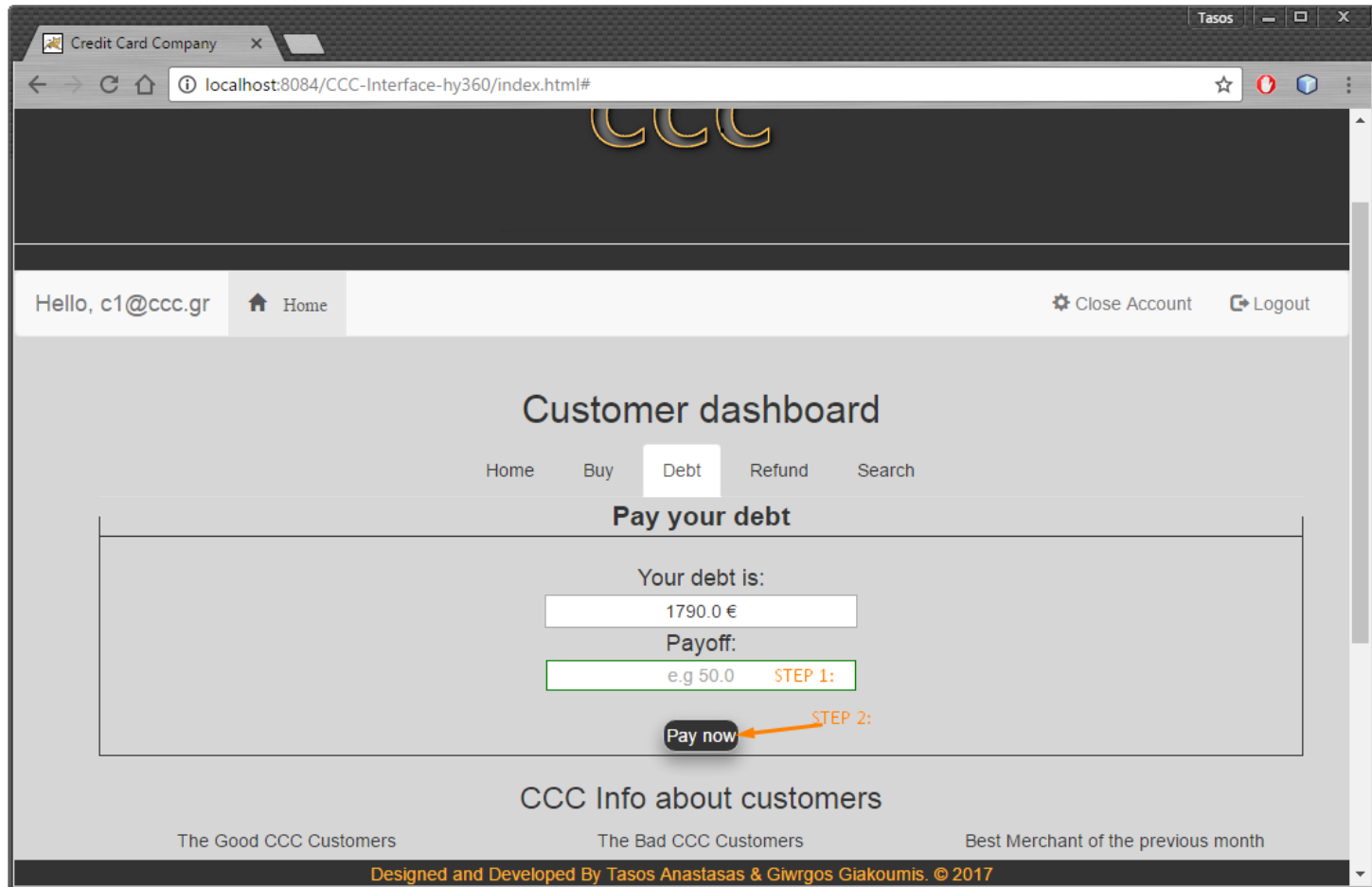
Items Cost: 10 STEP 2:

Order now STEP 3:

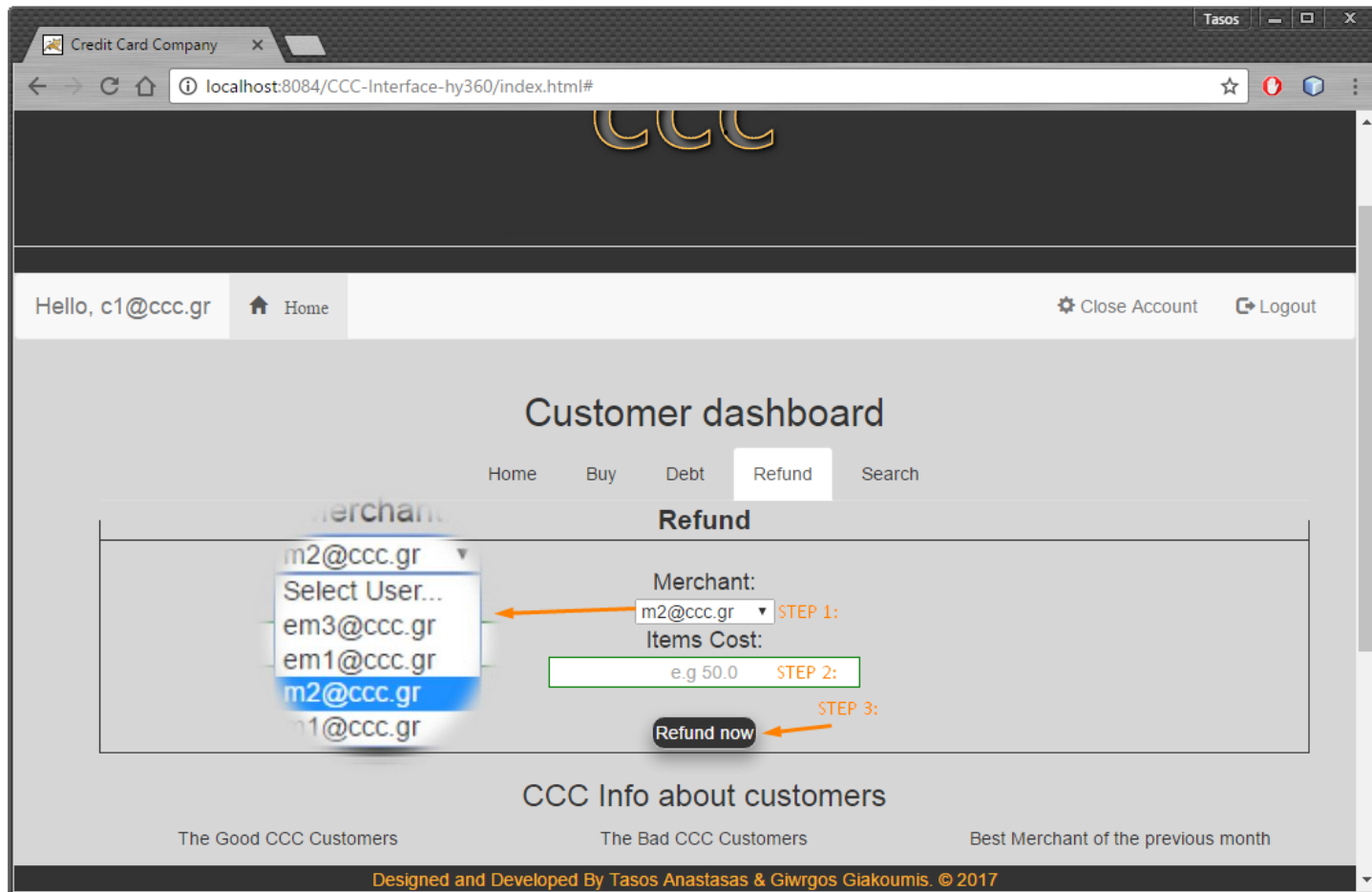
CCC Info about customers

The Good CCC Customers The Bad CCC Customers Best Merchant of the previous month

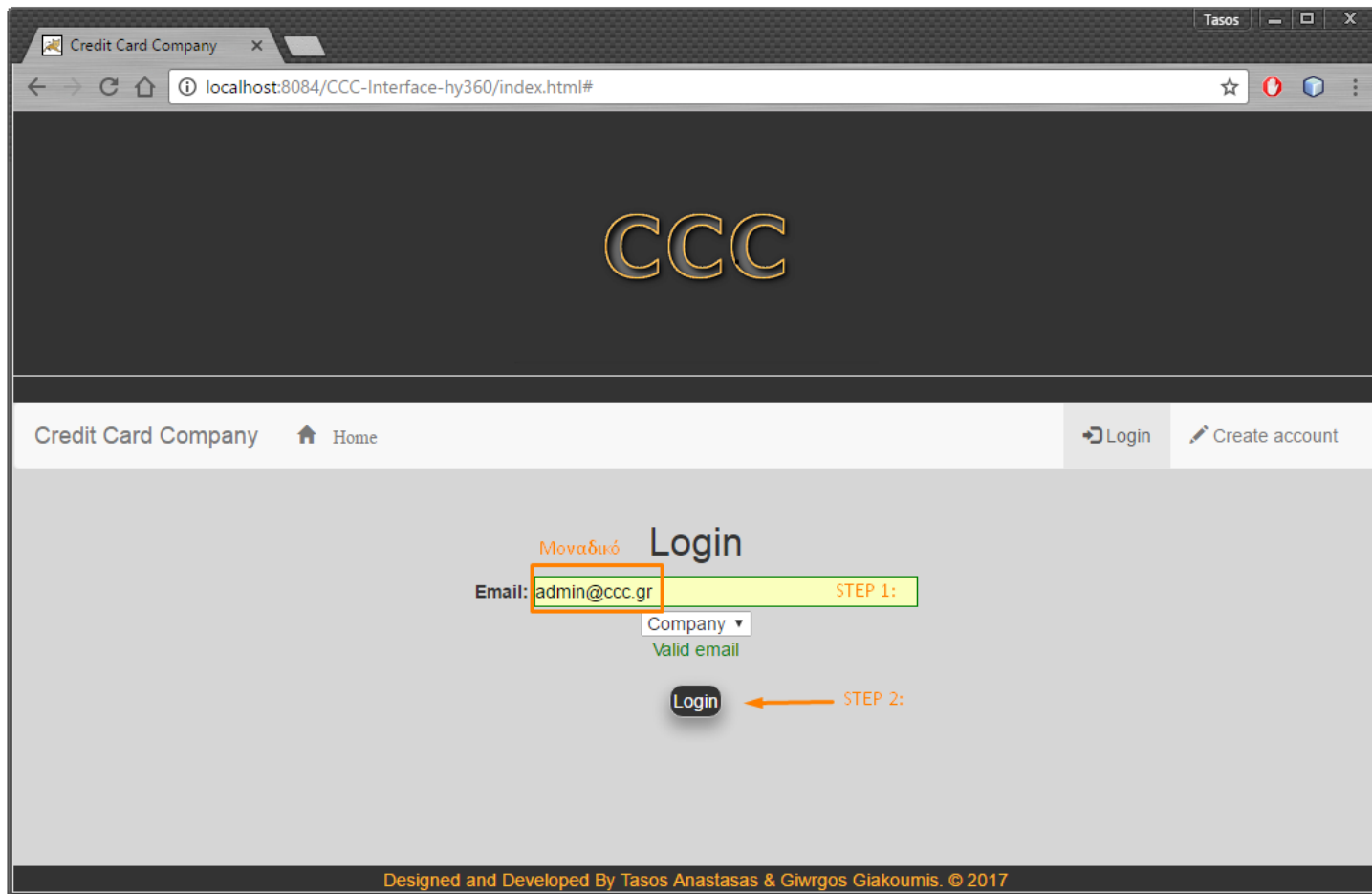
Designed and Developed By Tasos Anastasas & Giwrgos Giakoumis. © 2017



- Στους merchant εμφανίζονται τα ID των εμπόρων με τους οποίους έχει κάνει καποια συναλλαγή στο παρελθόν



- Admin id



- Έμποροι του μήνα μπορούν να είναι παραπάνω από ένας. Τους κάνουμε μείωση οφειλής

The screenshot shows a web browser window with the address bar displaying 'localhost:8084/CCC-Interface-hy360/index.html#'. The page title is 'Credit Card Company'. The user is logged in as 'admin@ccc.gr'. The dashboard is titled 'Admin dashboard' and has a navigation bar with 'Home', 'Merchant of the month', and 'Search'. A button labeled 'Make discount on best Merchant' is highlighted with an orange arrow pointing to the 'Best Merchant of the previous month' table. The dashboard also features three tables: 'The Good CCC Customers', 'The Bad CCC Customers', and 'Best Merchant of the previous month'. The footer states 'Designed and Developed By Tasos Anastasas & Giwrgos Giakoumis. © 2017'.

#	ID
1	c3@ccc.gr
2	m3@ccc.gr

#	ID	Debt
1	m1@ccc.gr	0.5
2	m2@ccc.gr	51.3
3	company@ccc.gr	100

#	ID
1	m2@ccc.gr

- Τα βήματα 1,2 είναι υποχρεωτικά ώστε να ενεργοποιηθεί η αναζήτηση

The screenshot shows the 'Credit Card Company' Admin dashboard. The browser address bar indicates the URL is `localhost:8084/CCC-Interface-hy360/index.html#`. The user is logged in as `admin@ccc.gr`. The dashboard features a 'Transaction search' section with a 'Choose Customer Type' dropdown set to 'Company'. Below this, there are fields for 'Transactions between', 'Transaction type' (set to 'Both'), 'Value' (set to 'None'), 'Date between', and a 'Search now' button. A callout box on the left provides a detailed view of these fields, showing the 'company@ccc.gr' email address in the 'Transactions between' field and the 'Search now' button. The main dashboard also includes a 'Transaction search' section with a 'Choose Customer Type' dropdown set to 'Company'. Below this, there are fields for 'Transactions between', 'Transaction type' (set to 'Both'), 'Value' (set to 'None'), 'Date between', and a 'Search now' button. The dashboard also includes a 'CCC Info about customers' section with three sub-sections: 'The Good CCC Customers', 'The Bad CCC Customers', and 'Best Merchant of the previous month'. The footer of the dashboard states 'Designed and Developed By Tasos Anastasas & Giwrgos Giakoumis. © 2017'.

Admin dashboard

Transaction search

Choose Customer Type: Company

Transactions between: company@ccc.gr with Select User... with

Transaction type: Both

Value: None

Value between: min - max

Date between: mm/dd/yyyy - mm/dd/yyyy

Search now

Transaction search

Choose Customer Type: Company

Transactions between: Select User... with with

Transaction type: Both

Value: None

Value between: min - max

Date between: mm/dd/yyyy - mm/dd/yyyy

Search now

CCC Info about customers

The Good CCC Customers

The Bad CCC Customers

Best Merchant of the previous month

ID # ID Debt # ID

Designed and Developed By Tasos Anastasas & Giwrgos Giakoumis. © 2017

Δυνατότητες Βελτιωσης

- Κάποιες φορές στα search δεν βγάζει τα dropdown. Αν ξαναγίνει κλίκ το search φτιάχνει.

Submitted files

- Forlder 'CCC-Interface-hy360' is a web application project for Netbeans editor.
- Forlder 'easeOfUse' contain files that can help to test Data Base.
 - Test's and print screens are made according to 'sampleDB.sql'.
 - sqlTABLES_CMD.txt contain the queries to create all tables
- Για να κάνετε import το sampleDB.sql, φτιάχνεται μια καινούρια βάση (το όνομα **πρέπει** να είναι 'hy360'). Πηγαίνετε στην καρτέλα 'Εισαγωγή', στο Choose file διαλέγεται το sampleDB.sql και πατάτε εκτέλεση.