**CPSC 471 Final Report**

**Abstract**

300 word summary

whole project is basically:

-Real world scenario as though a use case analysis was done

-Complete description of the database and its interfaces

\*\*\*Summary to see if worth reading details? Also used for searches?

**yes**

**Introduction**

-Brief

-Describe the problem/task the database is designed to solve

-Describe the database solution to the above problem

\*\*\*Leads into the topic? Include use case analysis here?

**motivation, overview, more dense abstract, don’t NEED use case analysis here but it is nice to have one somewhere**

**Project Design**

-Detailed

-Describe users of system

-Describe complete transaction collection (every effing transaction)

-Complete picture of system functionality

-Thorough ER diagram (identify changes compared to previous version)

\*\*\*So basically our presentation? Describe the entities too? transaction collection?

**yes. up to you. its basically all your relationships**

**Implementation**

-Complete relational schema diagram **(must begin section with this)**

-Indicate that the algorithm for converting ER diagrams to relational schema diagrams was followed

-Discuss significant decisions

\*\*\*How do you want us to indicate the algorithm was used? Detailed? Example of significant decision?

**very much up to you how much to include**

-Describe DBMS selected: **phpadmin, mysql workbench6.3, dbforge**

-Describe each SQL statement for every transaction

- intro talking about Brian’s generic methods

<?php

CRUD:

**"INSERT INTO transaction (transactionid, date, memberid, inventorynum, amount, totals, cashierid) VALUES ('45', '2015-12-05', '0', '0', '5', '25', '1234')"**

function RowCreate($database, $table, $value\_array)

{

// build query

$query = "INSERT INTO " . $table . " ";

$columns = "(";

$values = "(";

list($key, $val) = each($value\_array);

do

{

$columns .= $key;

$values .= "'$val'";

// check next

unset($key);

unset($val);

list($key, $val) = each($value\_array);

if (isset($key) && isset($val))

{

$columns .= ", ";

$values .= ", ";

}

} while (isset($key) && isset($val));

$columns .= ")";

$values .= ")";

$query .= $columns . " VALUES " . $values;

$result = mysqli\_query($database, $query) or die(mysqli\_error($database));

return $result;

}

function RowDelete($database, $table, $key\_array)

{

// build query

$query = "DELETE FROM " . $table . " WHERE ";

// key values

list($key, $val) = each($key\_array);

do

{

$query .= $key . " = '$val'";

// check next

list($key, $val) = each($key\_array);

if ($key && $val)

{

$query .= " AND ";

}

} while ($key && $val);

$result = mysqli\_query($database, $query) or die(mysqli\_error($database));

return $result;

}

**"UPDATE member SET memberid = '1234', firstname = 'Mike', middlename = ‘S’ '', lastname = ‘Smith’ '', phonenumber = ‘403-123-1234’ '', streetnum = ‘123’'', streetname = ‘my street’ '', city = ‘Calgary’ '', postalcode = ‘B1B-1B1 '', rating =’5.0’ '', renewaldate =’2015-01-14 '', datejoined =’2014-01-01’ '', password = ‘password'', active = '' WHERE memberid = '1234'"**

function RowUpdate($database, $table, $key\_array, $update\_array)

{

// build query

$query = "UPDATE " . $table . " SET ";

// update values

list($key, $val) = each($update\_array);

do

{

$query .= $key . " = '$val'";

// check next

unset($key);

unset($val);

list($key, $val) = each($update\_array);

if (isset($key) && isset($val))

{

$query .= ", ";

}

} while (isset($key) && isset($val));

$query .= " WHERE ";

// key values

list($key, $val) = each($key\_array);

do

{

$query .= $key . " = '$val'";

// check next

list($key, $val) = each($key\_array);

if ($key && $val)

{

$query .= " AND ";

}

} while ($key && $val);

//echo $query;

$result = mysqli\_query($database, $query) or die(mysqli\_error($database));

return $result;

}

**"SELECT \* FROM member”**

function TableRetrieve($database, $table)

{

$query = "SELECT \* FROM " . $table;

$result = mysqli\_query($database,$query) or die(mysqli\_error($database));

return $result;

}

function getNextID($database, $table, $column)

{

$query = "SELECT max(".$column.") as '$column' FROM " . $table;

$result = mysqli\_query($database,$query) or die(mysqli\_error($database));

while ($tableRow = mysqli\_fetch\_assoc($result))

{

foreach ($tableRow as $key => $value)

{

//echo "MAX: ".$tableRow[$key];

return $tableRow[$key];

}

}

return -1;

}

event\_schedule:

**$query = "SELECT e.id, e.name, e.date FROM event e WHERE EXISTS (SELECT \* FROM event, volunteers\_for WHERE memberid = '$memberid' and e.id = eventid)";**

**"SELECT e.id, e.name, e.date FROM event e WHERE EXISTS (SELECT \* FROM event, participates\_in WHERE memberid = '$memberid' and e.id = eventid)";**

**"SELECT e.id, e.name, e.date FROM event e WHERE NOT EXISTS (SELECT \* FROM event, volunteers\_for v, participates\_in p WHERE "**

**. "(p.memberid = '$memberid' and e.id = p.eventid) or "**

**. "(v.memberid = '$memberid' and e.id = v.eventid))";**

Volounteer:

* **"SELECT \* FROM volunteers\_for WHERE memberid='$memberid' and eventid='$eid'";**
* **"SELECT \* FROM event WHERE id='$eid'";**
* **"INSERT INTO `mydb`.`volunteers\_for` (`memberid`, `eventid`) VALUES ('$memberid', '$eid')";**
* **"DELETE FROM volunteers\_for WHERE memberid='$memberid' and eventid='$eid'";**

updateinventory

* **RowUpdate()**

updatemember

* **RowUpdate()**

registerevent

* **"SELECT \* FROM participates\_in WHERE memberid='$memberid' and eventname='$ename' and eventdate='$edate'";**
* **"SELECT \* FROM event WHERE name='$ename' and date='$edate'";**
* **"INSERT INTO `mydb`.`participates\_in` (`memberid`, `eventname`, `eventdate`) VALUES ('$memberid', '$ename', '$edate')";**
* **"SELECT \* FROM participates\_in WHERE memberid='$memberid' and eventname='$ename' and eventdate='$edate'";**
* **"DELETE FROM participates\_in WHERE memberid='$memberid' and eventname='$ename' and eventdate='$edate'";**

participate

* **"SELECT \* FROM participates\_in WHERE memberid='$memberid' and eventid='$eid'";**
* **"SELECT \* FROM event WHERE id='$eid'";**
* **"INSERT INTO `mydb`.`participates\_in` (`memberid`, `eventid`) VALUES ('$memberid', '$eid')";**
* **"DELETE FROM participates\_in WHERE memberid='$memberid' and eventid='$eid'";**

listmembers

* **RowDelete($database, 'member', $key\_array);**
* **TableRetrieve($database, "member");**

inventory

* **RowDelete($database, 'inventory\_items', $key\_array);**
* **RowCreate($database, 'inventory\_items', $\_POST);**
* **TableRetrieve($database, "inventory\_items");**

event\_schedule

* **"SELECT e.id, e.name, e.date FROM event e WHERE EXISTS (SELECT \* FROM event, volunteers\_for WHERE memberid = '$memberid' and e.id = eventid)";**
* **"SELECT e.id, e.name, e.date FROM event e WHERE EXISTS (SELECT \* FROM event, participates\_in WHERE memberid = '$memberid' and e.id = eventid)";**
* **"SELECT e.id, e.name, e.date FROM event e WHERE NOT EXISTS (SELECT \* FROM event, volunteers\_for v, participates\_in p WHERE ". "(p.memberid = '$memberid' and e.id = p.eventid) or . "(v.memberid = '$memberid' and e.id = v.eventid))";**

delete\_member

* **RowDelete($database, 'member', $key\_array);**

create\_member

* **RowCreate($database, 'member', $\_POST);**

transaction\_test.php

**"SELECT memberid FROM member WHERE firstname = '".$firstName. "' AND lastname = '".$lastName."'";**

transaction\_Lookup.php

**"SELECT m.firstname AS 'testName', i.name, t.memberid,\n"**

**. " a.firstname, a.lastname, t.transactionid, t.amount, t.totals FROM member AS m, inventory\_items AS i,\n"**

**. " transaction AS t, member AS a WHERE m.memberid = t.memberid \n"**

**. " AND i.inventorynum = t.inventorynum AND a.memberid = t.cashierid ORDER BY t.transactionid\n"**

**. " ";**

\*\*\*Do we need justification?

**yes**

**User Interface**

-Brief description interface design

-Screenshots

\*\*\*Explain choices?

**Entity Framework vs PHPmyadmin**

**HTML/Bootstrap**

**yeah**