

**CCENT Database Project**

For

The Center for Convergence and Emerging Networking Technologies

by

Lu Zou

A report submitted in partial fulfillment of the requirements  
for CCENT

Submitted to

Prof. Carlos Caicedo

On

4/24/2012

## Contents

1. Project Statement.....	3
2. Database model.....	3
3. Database Rules .....	6
4. Physical design.....	8
1) Create database for the CCENT .....	8
2) Create table.....	9
3) Create the user .....	11
4) Create the audit table .....	12
5) Create the device status trigger.....	14
6) Create a function .....	15
5. Interface design .....	16
1) Default colors:.....	16
2) Interface structure .....	16
3) Outer form/report header .....	17
4) Left navigation .....	18
5) Outer form/report detail .....	19
6) Button .....	20
6. Operation Handbook.....	20
1) Overview .....	20
2) SQL Server connection .....	21
7. Questions.....	29

## **1. Project Statement**

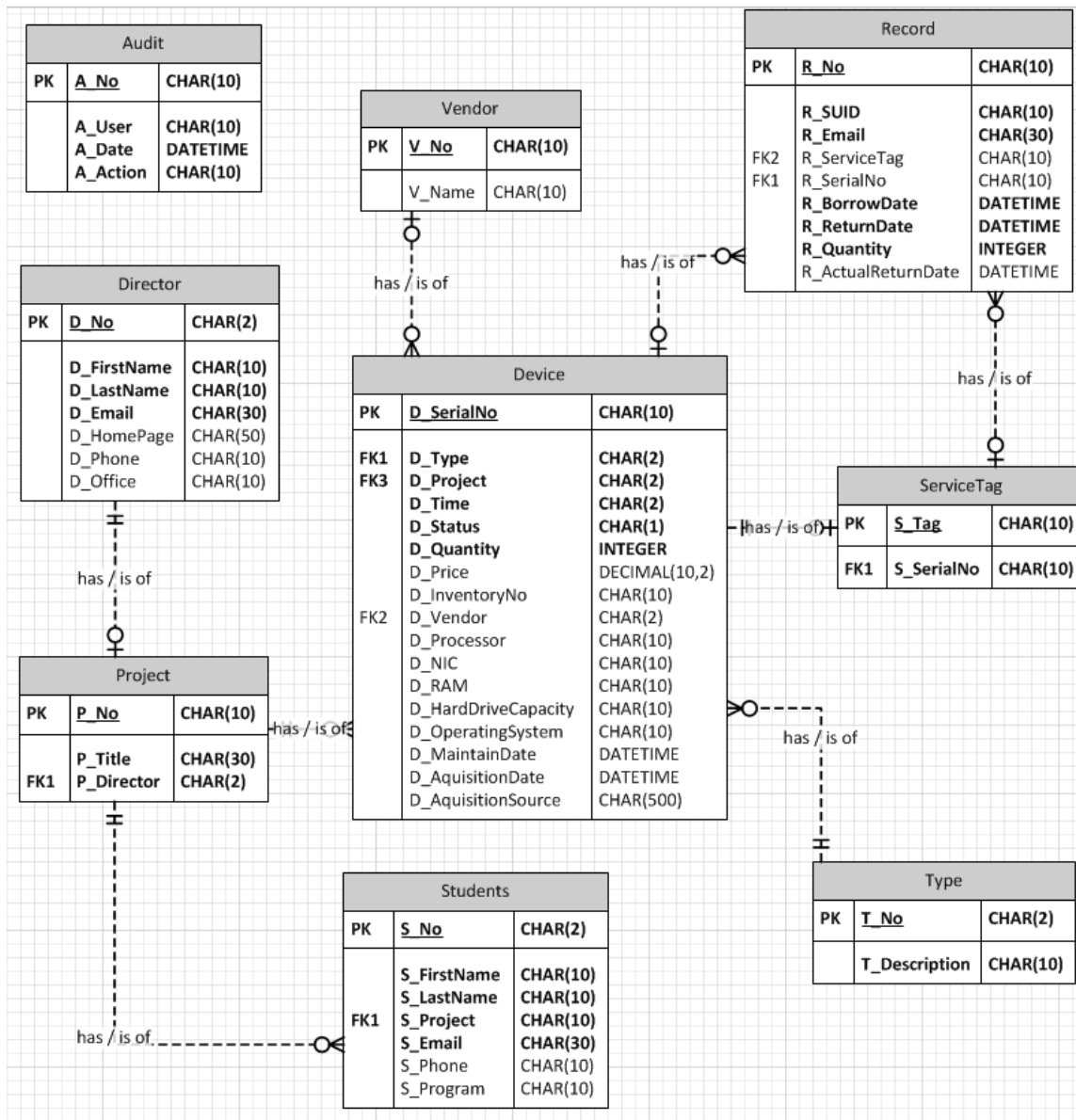
The project is to build a database system for the CCENT to manage the team information as well as the device information.

Currently, the CCENT has several research projects, and each project has a director and a certain number of student members. Most of the student members are master students who will leave the school when they graduate after two years of study. That means the team members will change almost every two years. So the database is needed to record history team member information and their efforts to the project.

On the other hand, the CCENT is a research lab which gets a lot of devices from different resources. Most of the devices are donated to the CCENT. Currently the servers and routers are managed by one student and the mobile devices are managed by another one. Other devices like cables and computer frames are not recorded. The new database system will record all the devices in a uniform format. It will give great convenience for the managers to manage the devices.

And for the convenience of the students outside the lab, the CCENT decides to lend devices to other students to help them improve the skills especially on IST600, IST648, IST653, IST656, and IST753. The new database system would help the manager to record the rent information, including the person who operates the system, the person who authorizes the action, the person who borrows the device and the device which is borrowed. The system would improve the efficiency of the lab, and ensure the all devices would not be lost.

## **2. Database model**



### Audit Table:

The Audit table is used to record the changes on the database. It will store any actions of any users, as well as the time when the actions happen.

### Device Table:

The Device table is used to store the information of all equipment.

D_SerialNo	The Serial No is the primary identifier of the equipment. It is unique for each equipment
D_Type	It records the type of the equipment, such as router or switch. It is connected with the Type Table to reduce the redundancy.

D_Project	It is the project that the equipment belongs to. It is connected with the Project Table to reduce the redundancy.
D_Time	It is the time that this piece of equipment could be rented.
D_Status	It is the rental status of the equipment. It could have one of the four values: 'A' (available), 'B' (Borrowed), 'D' (in delay), 'N' (not available).
D_Quantity	The amount of the devices (for cables)
D_Price	It is the price of this piece of equipment. It used when the device is broken by students, the students may be charged for a certain percent of the price.
D_InventoryNo	If the device is donated by the iSchool, the device would have a Inventory Number which is assigned by the iSchool.
D_Vendor	It is the manufacturer of the equipment. It is connected with the Manufacturer Table to reduce the redundancy.
D_Processor	The technical details of the devices
D_NIC	
D_RAM	
D_HardDriveCapacity	
D_OperatingSystem	
D_MaintainDate	It is the date when the equipment is maintained last time.
D_AcquisitionDate	The date the device is obtained
D_AcquisitionSource	The source of the device

### **ServiceTag Table:**

The ServiceTag table is used to store the service tag of the devices. The service tag is the code tag which can be scanned by the code reader. But only some of the devices have a service tag, this attribute could not be put in the Device table.

So the solution is: if the device has a service tag, the device could be scanned by the code reader; or else if the device does not have a service tag, the device could only be identified by its serial number which can only be input by manual work.

### **Project Table:**

The Project table stores the project information of the CCENT Lab. It contains the project number, the project description and the project director.

### **Director Table:**

The Director table stores the information of the project directors. It contains the name and the contact information of the directors.

**Students Table:**

The Students table stores the information of the project members. It contains the name and the contact information of the students, as well as the program the students study.

**Type Table:**

The Type table stores the types of the equipment.

**Manufacturer Table:**

The Manufacturer table stores the information of the manufacturers.

**Record Table:**

The Record table records the actions of rent and return.

R_No	The unique identifier of each piece of record.
R_SUID	The SUID of the student who rents the equipment.
R_Email	The contact information of the student who rents the equipment.
R_ServiceTag	If the device has a service tag, the service tag number could be scanned by the code reader into the database. Or, the device will be identified by the serial number.
R_SerialNo	If the device has no service tag, then the serial number will be the identifier. But the serial number could not be scanned, the only solution is to be input by manual work.
R_BorrowDate	The date when the equipment is rented.
R_ReturnDate	The date when the equipment should be returned.
R_ActualReturnDate	The date when the equipment is actually returned.
R_Quantity	The unit of the device (for cables)
R_AuthorizedPerson	The person who authorizes the rent action

### **3. Database Rules**

- 1) Each piece of equipment has a unique identify.

**Implement:** in the Equipment table which is used to store the equipment information, the equipment is identified by the primary key serial number. But the serial number could not be scanned by the code reader, the devices which have the service tag will use the service tag as the primary key. (details can be seen in the ERD diagram)

- 2) Each project is unique.

**Implement:** in the Project table which is used to store the project information, the project is identified by the primary key project number.

- 3) Each record is unique.

**Implement:** in the record table which is used to record the rent actions, the record is identified by the primary key record number.

- 4) Each vendor is unique.

**Implement:** in the vendor table which is used to record the manufacturer of the equipment, the manufacturer is identified by the primary key manufacturer number.

- 5) Each piece of equipment can be rented by one student at one time. And each student can rent many pieces of equipment at one time.

**Implement:** there is a one to zero or many relationship between the Record table and the Equipment table.

- 6) Each project has one director.

**Implement:** there is a one to one relationship between the Project table and the Director table.

- 7) Each piece of the equipment has a unique vendor. And each vendor can provide many pieces of equipment.

**Implement:** there is a one to many relationship between the Vendor table and the Device table.

- 8) Each piece of equipment can be in four statuses: available (can be rented by students), borrowed (has been rented by the students), in delay (has been rented and should be returned but not yet), and not available (has been broken and needs to ben maintain).

**Implement:** in the Equipment table, the D\_status attribute can be in one of four values: 'A' (available), 'B' (borrowed), 'D' (in delay), 'N' (not available).

- 9) Each piece of equipment should be returned within the time the equipment could be rented.

**Implement:** in the Equipment table, the D\_time attribute is used to set the rental time of the equipment and could be set by the project manager. And in the Record table, the  $R\_returnDate = R\_borrowDate + D\_time$ .

- 10) Each piece of equipment should be maintained in every certain duration.

**Implement:** in the Equipment table, the D\_MaintainDate is used to record the date when the equipment is maintained last time. And a trigger will be implemented to alarm before the next maintain date.

- 11) Every action on the database will be audit automatically.

**Implement:** the Audit table will record the actions on the database, including the user, the date, and the action.

## 4. Physical design

The developer account for SQL Server:

Account: Lu  
Password: ccentadmin123

### 1) Create database for the CCENT

A database is a storage location where the actual data underlying database objects can be kept. The system needs a database space to store all the tables and the functions/procedures (in the future) of the CCENT Device. As there is no user in the system, the database is created using the developer account.

```
/*
*****
    DATABASE NAME: CCENT_INVENTORY
    DESCRIPTION: CCENT DATABASE FOR DEVICE MANAGEMENT
    DATE: 2/21/2012
    CREATOR: LU ZOU
*****
*/
```



```
CREATE DATABASE CCENT_INVENTORY;  
USE CCENT_INVENTORY;
```

## 2) Create table to store the data

Tables are the basic unit of data storage in a Database. The CCENT Inventory Database has 8 tables to store the data required to fulfill the functions of the system. The vendor table and type table are used to avoid data repeat.

```
/*  
*****  
    PROCESS: CREATE TABLE  
    DESCRIPTION: CREATE TABLE BASED ON THE ERD  
                  THE TABLES ARE USED TO STORE THE DATA  
    DATE: 2/21/2012  
    CREATOR: LU ZOU  
*****  
*/  
CREATE TABLE DIRECTOR(  
    D_NO CHAR(2) PRIMARY KEY,  
    D_FIRSTNAME CHAR(20) NOT NULL,  
    D_LASTNAME CHAR(20) NOT NULL,  
    D_EMAIL CHAR(30) NOT NULL,  
    D_HOMEPAGE VARCHAR(100),  
    D_PHONE CHAR(15),  
    D_OFFICE CHAR(10)  
);  
  
CREATE TABLE PROJECT(  
    P_NO CHAR(3) PRIMARY KEY,  
    P_TITLE CHAR(50) NOT NULL,  
    P_DIRECTOR CHAR(2) NOT NULL,  
  
    CONSTRAINT FOREIGN KEY PROJECT_FK REFERENCES DIRECTOR(D_NO)  
);
```

```
CREATE TABLE STUDENTS(  
    S_SUID CHAR(10) PRIMARY KEY,  
    S_FIRSTNAME CHAR(20) NOT NULL,  
    S_LASTNAME CHAR(20) NOT NULL,  
    S_PROJECT CHAR(3) NOT NULL,  
    S_EMAIL CHAR(30) NOT NULL,  
    S_PHONE CHAR(15),  
    S_PROGRAM CHAR(5) CHECK (S_PROGRAM IN ('IM', 'TNM', 'LIS')),  
  
    CONSTRAINT FOREIGN KEY STUDENTS_FK REFERENCES PROJECT(P_NO)  
);
```

```
CREATE TABLE VENDOR(  
    V_NO CHAR(10) PRIMARY KEY,  
    V_NAME CHAR(100) NOT NULL  
);
```

```
CREATE TABLE T_TYPE(  
    T_NO CHAR(3) PRIMARY KEY;  
    T_DESCRIPTION CHAR(10) NOT NULL  
);
```

```
CREATE TABLE DEVICE(  
    D_SERIALNO CHAR(30) PRIMARY KEY,  
    D_TYPE CHAR(3) NOT NULL,  
    D_PROJECT CHAR(3) NOT NULL,  
    D_TIME CHAR(2) NOT NULL,  
    D_STATUS CHAR(1) NOT NULL CHECK (D_STATUS IN ('A', 'B', 'D', 'N')),  
    D_QUANTITY INT NOT NULL DEFAULT 1,  
    D_PRICE DECIMAL(10,2),  
    D_INVENTORYNO CHAR(30),  
    D_VENDOR CHAR(10),  
    D_PROCESSOR CHAR(10),  
    D_NIC CHAR(10),  
    D_RAM CHAR(10),  
    D_HARDDRIVECAPACITY CHAR(10),
```

```

D_OPERATINGSYSTEM CHAR(10),
D_MAINTAINANCE DATE,
D_AQUISITIONDATE DATE,
D_AQUISITIONSOURCE VARCHAR(500),

CONSTRAINT FOREIGN KEY DEVICE_FK1 REFERENCES T_TYPE(T_NO),
CONSTRAINT FOREIGN KEY DEVICE_FK2 REFERENCES PROJECT(P_NO),
CONSTRAINT FOREIGN KEY DEVICE_FK3 REFERENCES VENDOR(V_NO)
);

```

```

CREATE TABLE SERVICE_TAG(
    S_TAG CHAR(30) PRIMARY KEY,
    S_SERIALNO CHAR (30) NOT NULL,

    CONSTRAINT FOREIGN KEY SERVICE_TAG_FK REFERENCES DEVICE(D_SERIALNO)
);

```

```

CREATE TABLE RENT_RECORD(
    R_NO INT PRIMARY KEY,
    R_SUID CHAR(10) NOT NULL,
    R_EMAIL CHAR(30) NOT NULL,
    R_SERVICETAG CHAR(30),
    R_SERIALNO CHAR(30),
    R_BORROWDATE DATE NOT NULL DEFAULT SYSDATE,
    R_RETURNDATE DATE NOT NULL,
    R_QUANTITY INT DEFAULT 1,
    R_ACTUALRETURNDATE DATE
);

```

### 3) Create the user who will manager the database

The only stakeholder of the system who will directly use the system is the manager. So only one user account is needed for the manager to use the database to manage the devices.

```

/*
*****

```

PROCESS: CREATE LOGIN, AND USER

DESCRIPTION: CREATE USERS, AND ASSGIN APPROPRIATE PRIVILEGES TO THE USERS

DATE: 2/21/2012

CREATOR: LU ZOU

\*\*\*\*\*

\*/

CREATE LOGIN LOG\_CCENT

WITH PASSWORD = ccent123;

USE LOG\_CCENT;

CREATE USER tester FOR LOGIN LOG\_CCENT;

GO;

#### 4) Create the audit table which will audit the performance of the database

The first trigger is going to record the daily operation of the database. it means that is the user made some changes to the data, the trigger will record the user's action and put into the audit table. The audit table is important for the DBA to guard the database is operated well. And once some exceptions happen, the audit table can be the evidence in the law suit.

/\*

\*\*\*\*\*

PROCESS: AUDIT TABLE, AUDIT SEQUENCE, AUDIT TRIGGER

DESCRIPTION: AUDIT TABLE IS USED TO AUDIT THE PERFORMANCE OF THE DATABASE

DATE: 2/21/2012

CREATOR: LU ZOU

\*\*\*\*\*

\*/

CREATE TABLE T\_AUDIT(

A\_NO BIGINT IDENTITY(1,1)

CHECK (A\_NO BETWEEN 1 AND 9999999999999999) PRIMARY KEY,

A\_USER VARCHAR(100) NOT NULL,

A\_EVENT VARCHAR(2000) NOT NULL,

A\_TIME DATETIME NOT NULL

);

```

CREATE TRIGGER TRI_DB
ON DATABASE
FOR CREATE_TABLE, DROP_TABLE, ALTER_TABLE, CREATE_VIEW,
    DROP_VIEW, CREATE_PROCEDURE, ALTER_PROCEDURE, DROP_PROCEDURE,
    CREATE_FUNCTION, ALTER_FUNCTION, DROP_FUNCTION, CREATE_TRIGGER,
DROP_TRIGGER
AS
    INSERT INTO T_AUDIT (A_USER, A_EVENT, A_TIME)
    VALUES
        (SYSTEM_USER,
EVENTDATA().value('(/EVENT_INSTANCE/TSQLCommand/CommandText)[1]', 'nvarchar(max)'),
convert(varchar, getdate(), 126));
GO

/* *****
PROCESS: RENT_RECORD TABLE TRIGGER
DESCRIPTION: THE RENT_RECORD TABLE IS ONE OF THE MOST IMOIRTANT TABLE IN THE
DATABASE, SO THE ACTIONS ON THIS TABLE SHOULD BE RECORDED
DATE: 2/28/2012
CREATOR: LU ZOU
***** */

--DML TRIGGER TO AUDIT THE SPECIFIC TABLE RENT_RECORD
CREATE TRIGGER TRI_INSERT
ON rent_record
AFTER INSERT
AS
    INSERT INTO T_AUDIT (A_USER, A_EVENT, A_TIME)
    VALUES (SYSTEM_USER, 'Inserted on rent_record' , convert(varchar, getdate(), 126));
GO

CREATE TRIGGER TRI_UPDATE
ON rent_record
AFTER UPDATE
AS
    INSERT INTO T_AUDIT (A_USER, A_EVENT, A_TIME)
    VALUES (SYSTEM_USER, 'Updated on rent_record' , convert(varchar, getdate(), 126));
GO

CREATE TRIGGER TRI_DELETE
ON rent_record
AFTER delete
AS
    INSERT INTO T_AUDIT (A_USER, A_EVENT, A_TIME)
    VALUES (SYSTEM_USER, 'Deleted on rent_record' , convert(varchar, getdate(), 126));
GO

```

## 5) Create the device status trigger which will automatically update the device status

The seconde trigger is used to automatically update the device status. It means that:

- 1) If the manager adds new rent record to the record table, the trigger will update the device status in the device table to 'borrowed'.
- 2) If the rented device is not returned on time, the trigger will update the device status to 'delay'.
- 3) If the device is returned, the trigger will update the device status to 'available'.

```
/*
*****

PROCESS: DEVICE STATUS TRIGGER
DESCRIPTION: CHANGE THE DEVICE STATUS AFTER THE DEVICE IS BORROWED
DATE: 2/21/2012
CREATOR: LU ZOU
*****

*/

CREATE TRIGGER TRI_STATUS_INSERTED
ON RENT_RECORD
AFTER INSERT
AS
    DECLARE @SERVICETAG CHAR, @SERIALNO CHAR
    SET NOCOUNT ON;
    SELECT @SERVICETAG=R_SERVICETAG,
           @SERIALNO=R_SERIALNO
           FROM INSERTED
    IF @SERIALNO IS NULL
    BEGIN
        UPDATE DEVICE SET D_STATUS='B'
        WHERE D_SERIALNO = (SELECT S_SERIALNO FROM SERVICE_TAG
                           WHERE S_TAG = @SERVICETAG)
    END

    IF @SERVICETAG IS NULL
    BEGIN
        UPDATE DEVICE SET D_STATUS='B'
        WHERE D_SERIALNO = @SERIALNO
    END
GO
```

```

CREATE TRIGGER TRI_STATUS_UPDATED
ON RENT_RECORD
AFTER UPDATE
AS
    DECLARE @SERVICETAG CHAR, @SERIALNO CHAR
    SELECT @SERVICETAG=R_SERVICETAG,
           @SERIALNO=R_SERIALNO
    FROM inserted
    IF UPDATE(R_ACTUALRETURNDATE)
    BEGIN
        IF @SERIALNO IS NULL
        BEGIN
            UPDATE DEVICE SET D_STATUS='A'
            WHERE D_SERIALNO = (SELECT S_SERIALNO FROM SERVICE_TAG
                                WHERE S_TAG = @SERVICETAG)
        END
        IF @SERVICETAG IS NULL
        BEGIN
            UPDATE DEVICE SET D_STATUS='A'
            WHERE D_SERIALNO =@SERIALNO
        END
    END
GO

```

## 6) Create a function to calculate the return date for each device

The function is to calculate return date of each device when the device is borrowed. Each device has the 'LoanPeriod' attribute which identifies the loan period of the device. When students borrow the devices, the return date of the device is the (current date + loan period). The return date will be used to check the status of the device, e.g. if the current date is larger than the return date, the status of the device is 'delayed'.

```

/* *****
PROCESS: FUNCTION
DESCRIPTION: CALCULATE THE RETURN DATE
DATE: 3/6/2012
CREATOR: LU ZOU
***** */

CREATE FUNCTION RETURNDATE_CAL (@BORROWDATE DATE, @SERIALNO CHAR(30))
RETURNS DATE
WITH EXECUTE AS CALLER
AS
    BEGIN
        DECLARE @LOANPERIOD INT, @RETURNDATE DATE;

```

```

SELECT @LOANPERIOD=D_LOANPERIOD FROM DEVICE WHERE D_SERIALNO = @SERIALNO;
SET @RETURNDATE = DATEADD(DAY, @LOANPERIOD ,@BORROWDATE);
RETURN(@RETURNDATE);
END;
GO;

ALTER TRIGGER RETURNDATE_UPDATE
ON RENT_RECORD
AFTER INSERT
AS
    DECLARE @SERVICETAG CHAR(30), @SERIALNO CHAR(30), @BORROWDATE DATE,
    @RETURNDATE DATE, @RENTNO INT
    SELECT @SERVICETAG=R_SERVICETAG,
           @SERIALNO=R_SERIALNO,
           @SERIALNO=R_BORROWDATE,
           @RENTNO=R_NO
           FROM INSERTED
    IF @SERIALNO IS NULL
    BEGIN
        SELECT @SERIALNO = S_SERIALNO
        FROM SERVICE_TAG
        WHERE S_TAG = @SERVICETAG;
    END

    SET @RETURNDATE = DBO.RETURNDATE_CAL(@BORROWDATE, @SERIALNO)
    UPDATE RENT_RECORD SET R_RETURNDATE = GETDATE() WHERE R_NO = @RENTNO;
--UPDATE RENT_RECORD SET R_RETURNDATE = @RETURNDATE;
GO

```

## 5. Interface design

### 1) Default colors:

Table 1

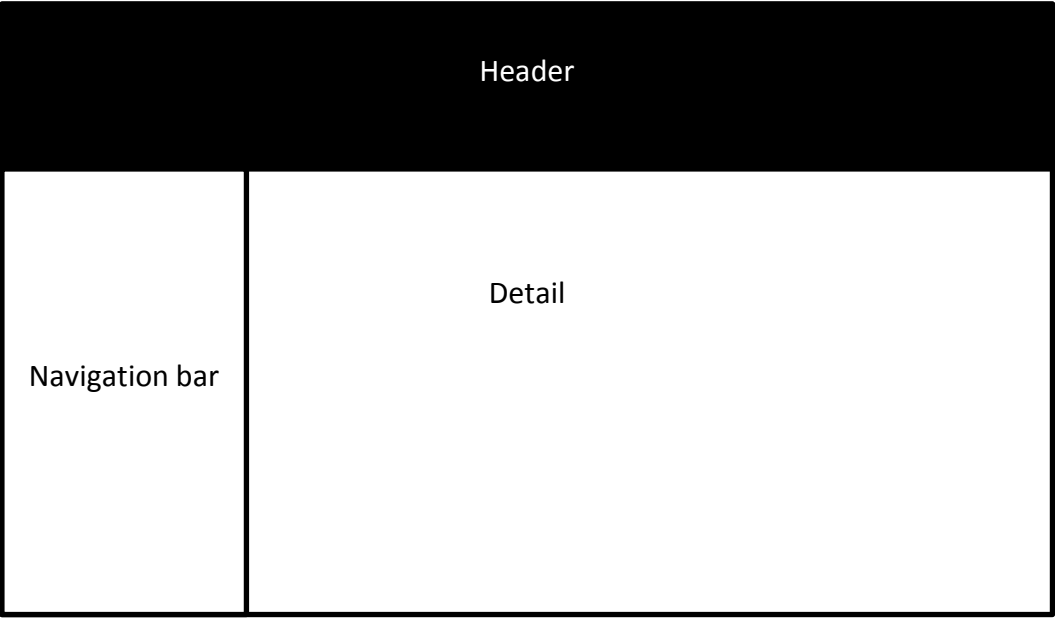
Use	Background	Button	Border	Navigation
Hex Number	333333	FF5A00	8A2003	000000

### 2) Interface structure

The framework of the interface is a kind of embedded structure. For the outer frame, the Header and navigation bar remains unchanged, and the detail changes for every navigation button.

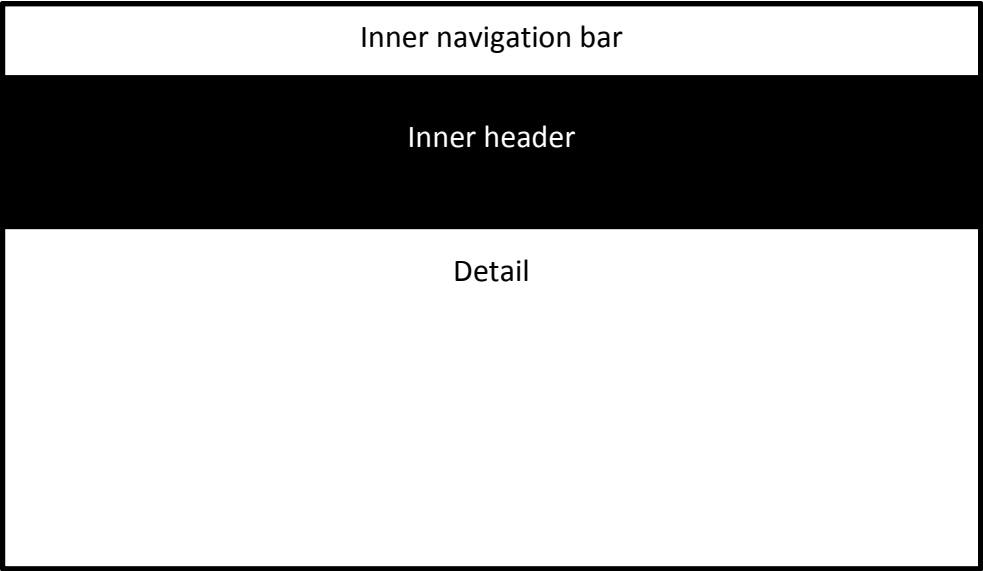
Outer frame





For the content part contains an inner frame, the inner navigation bar controls the inner content part.

Inner frame



**3) Outer form/report header**

Table 2

Title	Details
Height	0.8903”

<b>Color</b>	#000000
<b>Logo</b>	Width: 0.9583"
	Height: 0.8542"
	Top: 0.0208"
	Left: 0.2083"
<b>Page title</b>	Color: #FFFFFF
	Size: 24
	Weight:
	Top: 0.125"
	Left: 1.4167"
<b>Page subtitle</b>	Color: #FFFFFF
	Size: 18
	Weight:
	Top:
	Left: 1.4167"

#### 4) Left navigation

Table 3

Title	Detail
<b>Background</b>	#333333
<b>Border</b>	Invisible
<b>Menu</b>	Back color: #FF5A00
	Hover color: #FF5A00
	Pressed color: #333333
	Font color:
	Hover fore color: #000000
	Pressed fore color: #FF5A00
	Font size: 11
	Weight: bold

## 5) Outer form/report detail

- Inner form/report header

Table 4

Title	Detail
Height	1''
Back color	#333333
Inner form/report title	Color: #FFFFFF
	Size: 20
	Weight: normal
	Top:
	Left: 1.4167''
Inner form/report subtitle	Color: #FFFFFF
	Size: 11
	Weight: normal
	Top:
	Left: 1.4167''

- Inner form/report detail

Table 5

Title	Detail
Height	5''
Label	Color: #333333
	Size:
	Top:
	Left:
	Space:
Data	Color: #333333

	Size:
	Top:
	Left:
	Space:

## 6) Button

Table 6

Title	Detail
<b>Minimal Size</b>	Width: 0.8"
	Height: 0.3"
<b>Color</b>	Back: #FF5A00
	Hover: #FF5A00
	Pressed: #333333
<b>Border</b>	Color: #8A2003
	Weight:
<b>Content</b>	Color:
	Hover: #333333
	Pressed: #FF5A00
	Size:
	Weight:
<b>Position</b>	Top: 0.1667"
	Left: 6.2917"

## 6. Operation Handbook

### 1) Overview

The CCENT Database System is a system which allows user to manage the people, projects, and devices of CCENT. The user groups are divided into two categories: the manager, and the member. The manager has the highest privilege and can execute add/update/delete actions on

the database. The member has the privileges which only allow them to view the database reports. They cannot add/update/delete the objects on the database.

However, currently the database system is only allowed to be accessed through a particular workstation in the CCENT lab. The account for this workstation is:

Account: WebDev  
Password: CCENTAdmin

## 2) SQL Server connection

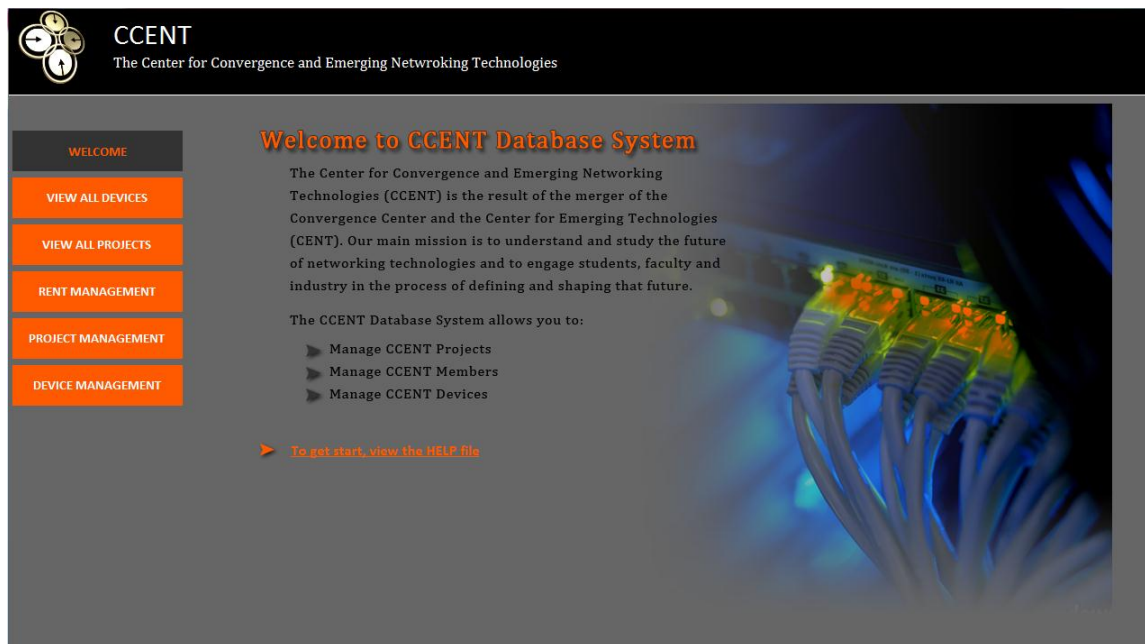
The CCENT Database System is based on MS SQL Server. Currently, the MS SQL Server can only be accessed within the CCENT domain. The user could use the CCENT DB only within the CCENT domain.

Before getting started, the user should have a SQL Server account. The account is used to connect with the SQL Server and controlled by the administrator for privilege assignment.

For testing the system, a test account is created within the highest privilege.


Account: tester  
Password: ccent123

### • Welcome



In the welcome page, a link is created to this document to guide the user.

- **View all devices**


**CCENT**  
 The Center for Convergence and Emerging Networking Technologies

WELCOME  
 VIEW ALL DEVICES  
 VIEW ALL PROJECTS  
 RENT MANAGEMENT  
 PROJECT MANAGEMENT  
 DEVICE MANAGEMENT

### VIEW ALL DEVICE INFORMATION

TO VIEW THE INFORMATION OF ALL DEVICES

Serial No			Service Tag		Device Type	Belongs to
FAVCDLOW03D					Router	IPv6
Rent Time	Status	Quantity	Original Price		Inventory Number	
	30/A	1			200	
Vendor						
Dell						
Next Maintain Date / Aquisition Date						
2012-12-25						
Aquisition Source						
Processor						
NIC		RAM		Hard Drive Capacity		
Operating System						

Serial No			Service Tag		Device Type	Belongs to
FAVCDLOW5D					Router	IPv6
Rent Time	Status	Quantity	Original Price		Inventory Number	
	30/A	1			20	
Vendor						
Dell						
Next Maintain Date / Aquisition Date						
2012-12-05						
Aquisition Source						

The second button on the left navigation bar is 'VIEW ALL DEVICES'. This page displays the parameters and technical details about the devices.

All users have the privilege to view the device information. This is a report based page, and users cannot modify any data in this page

- **View all projects**

The screenshot shows the CCENT web application. The header includes the CCENT logo and the text 'The Center for Convergence and Emerging Networking Technologies'. The left navigation bar has six buttons: 'WELCOME', 'VIEW ALL DEVICES', 'VIEW ALL PROJECTS' (highlighted), 'RENT MANAGEMENT', 'PROJECT MANAGEMENT', and 'DEVICE MANAGEMENT'. The main content area is titled 'VIEW PROJECT INFORMATION' and contains the following information:

TO VIEW THE INFORMATION OF ALL PROJECTS

Project NO: 001  
 PROJECT TITLE: IPv6  
 DIRECTOR: 03  
 DESCRIPTION: Researchers in this area investigate the inner-workings of the IPv6 protocol with a focus on its security vulnerabilities. IPv6 is the updated version of one of the main protocols that makes the internet work. They have configured a small IPv6 network in which they perform several security attacks in order to characterize their impact on a network and their possible countermeasures. The study of similar issues on Mobile IPv6 networks is under way this semester. This work will also result in the creation of demos and lab exercises about the IPv6 protocol.

SUID	FIRST NAME	LAST NAME	EMAIL	PHONE	PROGRAM
609523845	Lu	Zou	luzou@syr.edu		TNM

Tuesday, April 17, 2012 Page 1 of 1

The third button on the left navigation bar is 'VIEW ALL PROJECTS'. This page displays the information about the project and its members. The user could use the 'Next' and 'Previous' button to navigate the project.

All users have the privilege to view the project information. This is a report based page, and users cannot modify any data in this page.

- **Rent Management**

The fourth button on the left navigation bar is 'RENT MANAGEMENT'. This page has four sub pages: 'ADD NEW RECORD', 'HISTORY RECORD', 'BORROWED DEVICES', and 'DEVICE IN DELAY'.

Only the manager has the privilege to view the 'ADD NEW RECORD' page, and all the users can view the other three pages.

### ***Add new record***

The screenshot displays the CCENT web application interface. At the top, the CCENT logo and name are visible, along with the tagline 'The Center for Convergence and Emerging Networking Technologies'. Below this, a navigation bar contains four tabs: 'ADD NEW RECORD', 'HISTORY RECORDS', 'BORROWED DEVICE', and 'DEVICE IN DELAY'. The 'ADD NEW RECORD' tab is currently selected. On the left side, there is a vertical sidebar with several menu items: 'WELCOME', 'VIEW ALL DEVICES', 'VIEW ALL PROJECTS', 'RENT MANAGEMENT', 'PROJECT MANAGEMENT', and 'DEVICE MANAGEMENT'. The 'RENT MANAGEMENT' item is highlighted. The main content area is titled 'ADD NEW RENT RECORD' and includes a subtitle: 'TO ADD THE SUID, RENT DATE, NAME, CONTACT INFORMATION, DEVICE INFORMATION, AND QUANTITY'. The form contains the following fields: 'RENT RECORD NO' (value: 111), 'SUID' (value: 609523541), 'FIRST NAME' (value: ANN), 'LAST NAME' (value: DXCV), 'EMAIL' (value: ANN@SYR.EDU), 'SERVICE TAG' (empty), 'SERIAL NO' (value: FHK0609N03D, with a dropdown arrow), 'BORROW DATE' (value: 2012-02-29), 'RETURN DATE' (value: 2012-03-06), 'QUANTITY' (value: 1), and 'ACTUALR ETURN DATE' (value: 2012-03-06). At the bottom of the form, there are five buttons: 'Find SUID', 'Add New', 'Previous', 'Next', and 'Save'.

When students borrow and return devices to the CCENT, the manager should use the 'ADD NEW RECORD' page to make a record.

When students borrow a piece of device, the manager should use the 'Add New' button on the bottom to create a new record. The information which should be input includes: rent record no, SUID, first and last name, email, either service tag or serial no, borrow date and quantity. And then save the record.

\* The return date attribute will be automatically generated by the database. There is no need to input one.

\* Leave the actual return date attribute blank when the students borrow devices.

When the students return devices, the manager could use 'Find SUID' button on the bottom to find the record. Once the record is found out, the manager could input the date into the actual return date attribute. And then save the record.

- ***History Record/Borrowed Device/Device in Delay***



The screenshot displays the CCENT web application interface. At the top, the CCENT logo and name are shown, along with the tagline 'The Center for Convergence and Emerging Networking Technologies'. Below this, a navigation bar contains four buttons: 'ADD NEW RECORD', 'HISTORY RECORDS' (which is highlighted), 'BORROWED DEVICE', and 'DEVICE IN DELAY'. On the left side, there is a vertical sidebar with six buttons: 'WELCOME', 'VIEW ALL DEVICES', 'VIEW ALL PROJECTS', 'RENT MANAGEMENT', 'PROJECT MANAGEMENT', and 'DEVICE MANAGEMENT'. The main content area is titled 'HISTORY RECORDS' and includes a sub-header 'TO VIEW THE HISTORY RENT RECORDS'. Below this is a table with the following data:

NO	SUID	FIRST NAME	LAST NAME	EMAIL	SERVICE TAG	SERIAL NO	BORROW DATE	QUANTITY	RETURN DATE
111	609523	ANN	DXCV	ANN@SYR.EDU		FHK0609N03D	2012-02-29	1	2012-03-06
222	609523	ANN	DXCV	ANN@SYR.EDU		FHK0609N03D	2012-03-20	1	
333	609523	ANN	DXCV	ANN@SYR.EDU		FHKNDLOW03D	2012-03-07	1	2012-03-07
555	609523	ANN	DXCV	ANN@SYR.EDU		FHK0626X03E	2012-04-17	1	

At the bottom of the table area, it says 'Tuesday, April 17, 2012' on the left and 'Page 1 of 1' on the right.


All users have the privilege to view these pages.

The history record page displays the history rent/return record. And the borrowed device page displays all the borrowed devices. And the devices in delay page displays all the devices in delay.

These three pages are report based pages. Users cannot modify any data in these pages.

- ***Project Management***

The fifth button on the left navigation bar is 'PROJECT MANAGEMENT'. This page has two sub pages: 'DIRECTOR', and 'PROJECT'. Only the manager has the privilege to view these two pages.



CCENT  
The Center for Convergence and Emerging Networking Technologies

WELCOME

VIEW ALL DEVICES

VIEW ALL PROJECTS

RENT MANAGEMENT

PROJECT MANAGEMENT

DEVICE MANAGEMENT


DIRECTORPROJECT

CCENT DIRECTORS

NO	FIRST NAME	LAST NAME	EMAIL	PHONE	OFFICE	HOME PAGE
01	Dave	Molta	djmolta@syr.edu	3154434549	206B Hinds Hall	
03	Carlos	Caicedo	ccaicedo@syr.edu	315443-1363	215 Hinds Hall	

AddDeleteSaveExit

The director page is a form based page which allows the user to add, delete and save update on the director information using the buttons on the bottom.



CCENT  
The Center for Convergence and Emerging Networking Technologies

DIRECTORPROJECT

WELCOME

VIEW ALL DEVICES

VIEW ALL PROJECTS

RENT MANAGEMENT

PROJECT MANAGEMENT

DEVICE MANAGEMENT

PROJECT AND TEAM MEMBERS

TO MANAGE PROJECT INFORMATION AND TEAM MEMBERS

PROJECT NO001

PROJECT TITLEIPv6

DIRECTORCarlos

DESCRIPTION

Researchers in this area investigate the inner-workings of the IPv6 protocol with a focus on its security vulnerabilities. IPv6 is the updated version of one of the main protocols that makes the internet work. They have configured a small IPv6 network in which they perform several security attacks in order to characterize their impact on a network and their possible countermeasures. The study of similar issues on Mobile IPv6 networks is under way this semester. This work will also result in the creation of demos and lab exercises about the IPv6 protocol.


Add NewPreviousNext

SUID	FIRST NAME	LAST NAME	PROJECT	EMAIL	PHONE	PROGRAM
609523845	Lu	Zou	001	luzou@syr.edu		TNM
			001			

The project page is a form based page which displays the project information and its team members. The user could use the ‘Previous’ and ‘Next’ button to navigate the projects. And the team member subform allows the user to add, delete and update team members under this project.

- **Device Management**

The sixth button on the left navigation bar is ‘DEVICE MANAGEMENT’. This page has three sub pages: ‘TYPE’, and ‘VENDOR’, and ‘DEVICE’. Only the manager has the privilege to view these pages.



**CCENT**  
The Center for Convergence and Emerging Networking Technologies

WELCOME

VIEW ALL DEVICES

VIEW ALL PROJECTS

RENT MANAGEMENT

PROJECT MANAGEMENT

DEVICE MANAGEMENT

VENDOR

TYPE


DEVICE

DEVICE TYPE

TYPE NO	TYPE DESCRIPTION
111	Router
222	Server
333	Switch

Save

Both the type page and the vendor are form based pages. They display all the type and vendor information in one page, and also allow the user to modify the data in the page. The user could use the ‘Save’ button to save the changes.



CCENT  
The Center for Convergence and Emerging Networking Technologies

VENDORTYPEDEVICE

WELCOME

VIEW ALL DEVICES

VIEW ALL PROJECTS

RENT MANAGEMENT

PROJECT MANAGEMENT

DEVICE MANAGEMENT

DEVICE INFORMATION

TO ADD DEVICES, DELETE DEVICES, AND MODIFY THE DEVICE INFORMATION

SERIAL NO

FAVCDLOW03D

DEVICE NAME

ROUTER R200

DEVICE TYPE

Router

PROJECT

001

LOAN PERIOD

30

Days

STATUS

AVAILABLE

QUANTITY

1

PRICE

200

INVENTORY NO

VENDOR

Dell

MAINTAINANCE

2012-12-25

AQUISITION DATE

AQUISITION SOURCE

Technical DetailsAdd NewPreviousNext

The device page is also a form based page. It lists the basic attribute of a piece of device. The user could use the combo box to choose the device type, belonging project, status and vendor. Also the 'Next' and 'Previous' buttons at the bottom allow the user to navigate all the devices. Another button 'Technical Details' is directed to another page which displays the details of this piece of device. Once changes happen to this page, the 'Save' button could be used to save the changes.

## 7. Questions

For any questions about the Database system, please contact luzou@syr.edu