Functional Design

HIPO Diagram & Function Examples

By Group 24

Connected Devices

Function: Add Device
Inputs: @H, @IP4, @IP6, @Type, @MAC, @R_SSID
Outputs: None
Pseudocode:
Connect to the database
Query =

```
/*
There will be referential constraints on the table so a valid MAC address and Router are
already created
*/
INSERT INTO Connected_Devices
VALUES (@H, @IP4, @IP6, @Type, @MAC, @R_SSID);
```

Parse query
Execute query
Close connection to the database

Function: Delete Device
Inputs: @IP4, @IP6
Outputs: None
Pseudocode:
Connect to the database
Query =

```
/*
Table will be set up with cascading deletes so status, which references this table will also
be cleared of the device if this table doesn't have it
*/
DELETE * FROM Connected_Devices WHERE (IPv4 = @IP4 OR IPv6 = @IP6);
```

Parse query
Execute query
Close connection to the database

Function: Add Status
Inputs: @IP4, @IP6, @Uptime, @Status
Outputs: None

Pseudocode:
Connect to the database
Query =

```
/*
There will be referential constraints on status so a valid device is already in the connected
devices table before it can be added to this table
*/
INSERT INTO Status
VALUES (@IP4, @IP6, @Uptime, @Status);
```

Parse query
Execute query
Close connection to the database

Function: Get IP
Inputs: @MAC
Outputs: IPv4, IPv6
Pseudocode:
Connect to the database
Query =

```
SELECT IPv4, IPv6
FROM Connected_Devices
WHERE @MAC = MAC_address;
```

Parse query
Execute query
Close connection to the database

Function: Get MAC Address
Inputs: @IPv4, @IPv6
Outputs: MAC_address
Pseudocode:
Connect to the database
Query =

```
SELECT MAC_address
FROM Connected_Devices
WHERE (@IPv4 = IPv4) OR (@IPv6 = IPv6);
```

Parse query
Execute query
Close connection to the database

Function: Get Hostname
Inputs: @IPv4, @IPv6

Outputs: Hostname
Pseudocode:
Connect to the database
Query =

```
SELECT Hostname
FROM Connected_Devices
WHERE (@IPv4 = IPv4) OR (@IPv6 = IPv6);
```

Parse query
Execute query
Close connection to the database

Function: Get connection uptime
Inputs: @IPv4, @IPv6
Outputs: Connection_uptime
Pseudocode:
Connect to the database
Query =

```
SELECT Connection_uptime
FROM Status
WHERE (CD_IPv4 = @IPv4) OR (CD_IPv6 = @IPv6);
```

Parse query
Execute query
Close connection to the database

Function: Get Online/Offlinbe Status
Inputs: @IPv4, @IPv6
Outputs: Online/Offline
Pseudocode:
Connect to the database
Query =

```
SELECT Online_Offline
FROM Status
WHERE (CD_IPv4 = @IPv4) OR (CD_IPv6 = @IPv6);
```

Parse query
Execute query
Close connection to the database

Firewall

Function: Add Firewall
Inputs: @MAC, @Level

Outputs: None
Pseudocode:
Connect to the database
Query =

```
INSERT INTO FIREWALL
VALUES (@MAC, @Level);
```

Parse query
Execute query
Close connection to the database

Function: Delete FIrewall
Inputs: @MAC
Outputs: None
Pseudocode:
Connect to the database
Query =

```
/*
Delete will cascade so anything that references firewall will also be deleted
*/
DELETE * FROM FIREWALL WHERE HW_MAC_address = @MAC;
```

Parse query
Execute query
Close connection to the database

Function: Get security level
Inputs: @MAC
Outputs: Security_level
Pseudocode:
Connect to the database
Query =

```
SELECT Security_level
FROM FIREWALL
WHERE @MAC = HW_MAC_address;
```

Parse query
Execute query
Close connection to the database

Function: Set security level
Inputs: @MAC. @Level
Outputs: None

Pseudocode:
Connect to the database
Query =

```
UPDATE FIREWALL
SET Security_level = @Level
WHERE HW_MAC_address = @MAC;
```

Parse query
Execute query
Close connection to the database

Function: Get Blacklist
Inputs: @MAC
Outputs: Black_list_site
Pseudocode:
Connect to the database
Query =

```
SELECT Black_list_site
FROM Black_list
WHERE FW_HW_MAC_address = @MAC;
```

Parse query
Execute query
Close connection to the database

Function: Set blacklist
Inputs: @MAC, @Site
Outputs: None
Pseudocode:
Connect to the database
Query =

```
/*
Function itself will take an array of strings, and will then loop through them executing the
insert into command multiple times, but will only execute the delete command once.
*/
DELETE * FROM Black_list WHERE FW_HW_MAC_address = @MAC;
INSERT INTO Black_list
VALUES (@MAC, @Site);
```

Parse query
Execute query
Close connection to the database

Function: Add to blacklist
Inputs: @MAC, @Site
Outputs: None
Pseudocode:
Connect to the database
Query =

```
INSERT INTO Black_list
VALUES (@MAC, @Site);
```

Parse query
Execute query
Close connection to the database

Function: Get whitelist
Inputs: @MAC
Outputs: White_list_site
Pseudocode:
Connect to the database
Query =

```
SELECT White_list_site
FROM White_list
WHERE FW_HW_MAC_address = @MAC;
```

Parse query
Execute query
Close connection to the database

Function: Set whitelist
Inputs: @MAC, @Site
Outputs: None
Pseudocode:
Connect to the database
Query =

```
/*
Function itself will take an array of strings, and will then loop through them executing the
insert into multiple times, but only execute the delete command once.
*/
DELETE * FROM White_list WHERE FW_HW_MAC_address = @MAC;
INSERT INTO White_list
VALUES (@MAC, @Site);
```

Parse query
Execute query

Close connection to the database

Function: Add to whitelist
Inputs: @MAC, @Site
Outputs: None
Pseudocode:
Connect to the database
Query =

```
INSERT INTO White_list
VALUES (@MAC, @Site);
```

Parse query
Execute query
Close connection to the database

Function: Get blocked ports
Inputs: @MAC
Outputs: Blocked_port
Pseudocode:
Connect to the database
Query =

```
SELECT Blocked_port
FROM Blocked_ports
WHERE @MAC = FW_HW_MAC_address;
```

Parse query
Execute query
Close connection to the database

Function: Set blocked ports
Inputs: @MAC, @Port
Outputs: None
Pseudocode:
Connect to the database
Query =

```
/*
Function will receive an array of strings and will execute the insert into command for each
entry in array, and will execute the delete command once
*/
DELETE * FROM Blocked_ports WHERE @MAC = FW_HW_MAC_address;
INSERT INTO Blocked_ports
VALUES (@MAC, @Port);
```

Parse query
Execute query
Close connection to the database

Function: Add blocked port
Inputs: @MAC, @Port
Outputs: None
Pseudocode:
Connect to the database
Query =

```
INSERT INTO Blocked_ports
VALUES (@MAC, @Port);
```

Parse query
Execute query
Close connection to the database

Router

Function: Add Router
Inputs: @SSID, @MAC, @IP4, @IP6, @Pass, @SNO, @V, @Up, @Vend, @Num
Outputs: None
Pseudocode:
Connect to the database
Query =

```
INSERT INTO HARDWARE
VALUES @MAC, @SNO, @V, @Up, @Vend, @IP4, @IP6, @Num);
INSERT INTO Router
VALUES (@SSID, @MAC, @IP4, @IP6, @Pass);
```

Parse query
Execute query
Close connection to the database

Function: Delete Router
Inputs: @SSID, @MAC
Outputs: None
Pseudocode:
Connect to the database
Query =

```
/*
Delete will cascade and everything that references this entity will be deleted also
*/
```

```
DELETE * FROM Router WHERE WiFi_SSID = @SSID;
DELETE * FROM HARDWARE WHERE MAC_address = @MAC;
```

Parse query
Execute query
Close connection to the database

Function: Get Serial No
Inputs: @SSID
Outputs: Serial_No
Pseudocode:
Connect to the database
Query =

```
SELECT Serial_No
FROM HARDWARE, Router
WHERE @SSID = WiFi_SSID AND HW_MAC_Address = MAC_address;
```

Parse query
Execute query
Close connection to the database

Function: Get software version
Inputs: @SSID
Outputs: Version
Pseudocode:
Connect to the database
Query =

```
SELECT Version
FROM HARDWARE, Router
WHERE WiFi_SSID = @SSID AND HW_MAC_Address = MAC_address;
```

Parse query
Execute query
Close connection to the database

Function: Update Software Version
Inputs: @MAC, @Version
Outputs: None
Pseudocode:
Connect to the database
Query =

```
UPDATE HARDWARE
SET Version = @Version
```

```
WHERE MAC_address = @MAC;
```

Parse query
Execute query
Close connection to the database

Function: Get uptime
Inputs: @SSID
Outputs: Connection_Uptime
Pseudocode:
Connect to the database
Query =

```
SELECT Connection_Uptime
FROM Status, Connected_Devices, Router
WHERE WiFi_SSID = @SSID AND HW_MAC_Address = MAC_address AND (IPv4 = CD_IPv4 OR IPv6 =
CD_IPv6);
```

Parse query
Execute query
Close connection to the database

Function: Get IP
Inputs: @SSID
Outputs: IPv4, IPv6
Pseudocode:
Connect to the database
Query =

```
SELECT IPv4. IPv6
FROM Router
WHERE WiFi_SSID = @SSID;
```

Parse query
Execute query
Close connection to the database

Function: Set IP
Inputs: @SSID, @IPv4, @IPv6
Outputs: None
Pseudocode:
Connect to the database
Query =

```
/*
Router table will have cascading updates so anything that references the table will also be
```

```
updated
*/
UPDATE Router
SET IPv4 = @IPv4, IPv6 = @IPv6
WHERE WiFi_SSID = @SSID;
```

Parse query
Execute query
Close connection to the database

Function: Get all connected devices
Inputs: @SSID
Outputs: IPv4, IPv6
Pseudocode:
Connect to the database
Query =

```
SELECT IPv4, IPv6
FROM Connected_Devices
WHERE R_SSID = @SSID;
```

Parse query
Execute query
Close connection to the database

Function: Get Firewall Settings
Inputs: @SSID
Outputs: Security_level, Black_list_site, White_list_site, Blocked_port
Pseudocode:
Connect to the database
Query =

```
SELECT Security_level, Black_list_site, White_list_site, Blocked_port
FROM FIREWALL, Black_list, White_list, Blocked_ports, Router
WHERE WiFi_SSID = @SSID AND (Router.HW_MAC_Address = FIREWALL.HW_MAC_address OR
FW_HW_MAC_address = HW_MAC_address) AND FIREWALL.HW_MAC_address =
Black_list.FW_HW_MAC_address AND FIREWALL.HW_MAC_address = White_list.FW_HW_MAC_address AND
FIREWALL.HW_MAC_address = Blocked_ports.FW_HW_MAC_address
ORDER BY WiFi_SSID;
```

Parse query
Execute query
Close connection to the database

Function: Get WiFi SSID
Inputs: @IPv4, @IPv6
Outputs: WiFi_SSID

Pseudocode:
Connect to the database
Query =

```
SELECT WiFi_SSID
FROM Router
WHERE IPv4 = @IPv4 OR IPv6 = @IPv6;
```

Parse query
Execute query
Close connection to the database

Function: Set WiFi SSID
Inputs: @IPv4, @IPv6, @SSID
Outputs: None
Pseudocode:
Connect to the database
Query =

```
/*
Router table will have cascading updates so anything that references the table will also be
updated
*/
UPDATE Router
SET WiFi_SSID = @SSID
WHERE (@IPv4 = IPv4 OR @IPv6 = IPv6);
```

Parse query
Execute query
Close connection to the database

Function: Get WiFi Password
Inputs: @SSID
Outputs: WiFi_password
Pseudocode:
Connect to the database
Query =

```
SELECT WiFi_password
FROM Router
WHERE WiFi_SSID = @SSID;
```

Parse query
Execute query
Close connection to the database

Function: Set WiFi Password
Inputs: @SSID, @Passowrd
Outputs: None
Pseudocode:
Connect to the database
Query =

```
UPDATE Router
SET WiFi_password = @Password
WHERE WiFi_SSID = @SSID;
```

Parse query
Execute query
Close connection to the database

Function: Get MAC address
Inputs: @SSID
Outputs: HW_MAC_Address
Pseudocode:
Connect to the database
Query =

```
SELECT HW_MAC_address
FROM Router
WHERE WiFi_SSID = @SSID;
```

Parse query
Execute query
Close connection to the database

Modem

Function: Add Modem
Inputs: @IP4, @IP6, @MAC, @SNO, @V, @Up, @Vend, @Num
Outputs: None
Pseudocode:
Connect to the database
Query =

```
INSERT INTO HARDWARE
VALUES @MAC, @SNO, @V, @Up, @Vend, @IP4, @IP6, @Num);
INSERT INTO Modem
VALUES (@IP4, @IP6, @MAC);
```

Parse query
Execute query

Close connection to the database

Function: Delete Modem
Inputs: @IP4, @IP6, @MAC
Outputs: IPv4, IPv6
Pseudocode:
Connect to the database
Query =

```
/*
Delete will cascade and anything that references this entity will also be deleted
*/
DELETE * FROM Modem WHERE (@IP4 = IPv4 OR @IP6 = IPv6);
DELETE * FROM HARDWARE WHERE MAC_address = @MAC;
```

Parse query
Execute query
Close connection to the database

Function: Get IP
Inputs: @MAC
Outputs: IPv4, IPv6
Pseudocode:
Connect to the database
Query =

```
SELECT IPv4, IPv6
FROM Modem
WHERE HW_MAC_Address = @MAC;
```

Parse query
Execute query
Close connection to the database

Function: Set IP
Inputs: @MAC, @IPv4, @IPv6
Outputs: None
Pseudocode:
Connect to the database
Query =

```
/*
Modem will have cascading updates, so by updating Modem, other tables that reference it will
be updated aswell
*/
UPDATE Modem
```

```
SET IPv4 = @IPv4, IPv6 = @IPv6
WHERE HW_MAC_Address = @MAC;
```

Parse query
Execute query
Close connection to the database

Gateway

Function: Add Gateway
Inputs: @IP4, @IP6
Outputs: None
Pseudocode:
Connect to the database
Query =

```
INSERT INTO Gateway
VALUES (@IP4, @IP6);
```

Parse query
Execute query
Close connection to the database

Function: Delete Gateway
Inputs: @IP4, @IP6
Outputs: None
Pseudocode:
Connect to the database
Query =

```
/*
Delete will cascade and anything that references this entity will also be deleted
*/
DELETE * FROM Gateway WHERE (IPv4 = @IP4 OR IPv6 = @IP6);
```

Parse query
Execute query
Close connection to the database

Function: Get IP
Inputs: @SSID
Outputs: G_IPv4, G_IPv6
Pseudocode:
Connect to the database
Query =

```
SELECT G_IPv4, G_IPv6
FROM External_connections
WHERE R_WiFi_SSID = @SSID;
```

Parse query
Execute query
Close connection to the database

Function: Set IP
Inputs: @SSID, @IPv4, @IPv6
Outputs: None
Pseudocode:
Connect to the database
Query =

```
/*
Gateway will have cascading updates, so by updating gateway, other tables that reference it
will be updated aswell
*/
UPDATE External_connections
SET G_IPv4 = @IPv4, G_IPv6 = @IPv6
WHERE R_WiFi_SSID = @SSID;
```

Parse query
Execute query
Close connection to the database

Network Report

Function: Get Report
Inputs: @DT, @SSID
Outputs: Total_Data_Used, Total_Connected_Users, Average_Speed, Average_Connection_Quality
Pseudocode:
Connect to the database
Query =

```
SELECT *
FROM Network_Report, Generates
WHERE WiFi_SSID = @SSID AND Date_Time = @DT;
```

Parse query
Execute query
Close connection to the database

Function: Generate Report
Inputs: @SSID, @DT, @TD, @TCU, @AS, @ACQ
Outputs: None

Pseudocode:
Connect to the database
Query =

```
/*
The web server will ping the router and get all the relevant information and will then insert
that info into the DB
*/
INSERT INTO Network_Report
VALUES (@DT, @TD, @TCU, @AS, @ACQ);
INSERT INTO Generates
VALUES (@SSID, @DT);
```

Parse query
Execute query
Close connection to the database

Parental Control

Function: Add Parental Control
Inputs: @ID, @SSID, @EN, @Time
Outputs: None
Pseudocode:
Connect to the database
Query =

```
/*
Due to referential constraints on the table, this query will not work if there is no user
with @ID or router with @SSID
*/
INSERT INTO Parental_Control
VALUES (@ID, @SSID, @EN, @Time);
```

Parse query
Execute query
Close connection to the database

Function: Delete Parental Control
Inputs: @ID, @SSID
Outputs: None
Pseudocode:
Connect to the database
Query =

```
/*
Delete will cascade and anything that references this entity will be deleted also
```

```
*/
DELETE * FROM Parental_Control WHERE User_ID = @ID AND R_WiFi_SSID = @SSID;
```

Parse query
Execute query
Close connection to the database

Function: Get Blocked Sites
Inputs: @ID
Outputs: URL, When
Pseudocode:
Connect to the database
Query =

```
SELECT *
FROM Blocked_Sites
WHERE PC_User_ID = @ID;
```

Parse query
Execute query
Close connection to the database

Function: Set Blocked Sites
Inputs: @ID, @URL, @DT
Outputs: None
Pseudocode:
Connect to the database
Query =

```
/*
Function will receive an array of URLs and for each one will execute the insert command, and
will only execute the delete command once to clear the sites for that user
*/
DELETE * FROM Blocked_Sites WHERE PC_User_ID = @ID;
INSERT INTO Blocked_Sites
VALUES (@ID, @URL, @DT);
```

Parse query
Execute query
Close connection to the database

Function: Add Blocked Site
Inputs: @ID, @URL, @DT
Outputs: None
Pseudocode:
Connect to the database
Query =

```
INSERT INTO Blocked_Sites
VALUES (@ID, @URL, @DT);
```

Parse query
Execute query
Close connection to the database

Function: Get Blocked Keywords
Inputs: @ID
Outputs: Keyword, When
Pseudocode:
Connect to the database
Query =

```
SELECT *
FROM Blocked_Keywords
WHERE PC_User_ID = @ID;
```

Parse query
Execute query
Close connection to the database

Function: Set Blocked Keywords
Inputs: @ID, @Keyword, @DT
Outputs: None
Pseudocode:
Connect to the database
Query =

```
/*
Function will receive an array of keywords and for each one will execute the insert command,
and will only execute the delete command once to clear the sites for that user
*/
DELETE * FROM Blocked_Keywords WHERE PC_User_ID = @ID;
INSERT INTO Blocked_Keywords
VALUES (@ID, @Keyword, @DT);
```

Parse query
Execute query
Close connection to the database

Function: Add Blocked Keyword
Inputs: @ID, @Keyword, @DT
Outputs: None
Pseudocode:

Connect to the database
Query =

```
INSERT INTO Blocked_Keywords
VALUES (@ID, @Keyword, @DT);
```

Parse query
Execute query
Close connection to the database

Function: Get Trusted Computers
Inputs: @ID
Outputs: IP, Trusted, Computer_Name
Pseudocode:
Connect to the database
Query =

```
SELECT *
FROM Trusted_Computers
WHERE PC_User_ID = @ID;
```

Parse query
Execute query
Close connection to the database

Function: Set Trusted Computers
Inputs: @ID, @IP, @T, @Name
Outputs: None
Pseudocode:
Connect to the database
Query =

```
/*
Function will receive an array of trusted computers and for each one will execute the insert
command, and will only execute the delete command once to clear the sites for that user
*/
DELETE * FROM Trusted_Computers WHERE PC_User_ID = @ID;
INSERT INTO Trusted_Computers
VALUES (@ID, @IP, @T, @Name);
```

Parse query
Execute query
Close connection to the database

Function: Add Trusted Computer
Inputs: @ID, @IP, @T, @Name
Outputs: None

Pseudocode:
Connect to the database
Query =

```
INSERT INTO Trusted_Computers
VALUES (@ID, @IP, @T, @Name);
```

Parse query
Execute query
Close connection to the database

Users

Function: Router Ban User
Inputs: @ID, @SSID
Outputs: None
Pseudocode:
Connect to the database
Query =

```
DELETE * FROM Can_Connect WHERE Client_ID = @ID AND R_WiFi_SSID = @SSID;
```

Parse query
Execute query
Close connection to the database

Function: Delete User
Inputs: @ID, @SSID
Outputs: None
Pseudocode:
Connect to the database
Query =

```
DELETE * FROM Can_Connect WHERE Client_ID = @ID;
```

Parse query
Execute query
Close connection to the database

Function: Add Admin
Inputs: @ID, @UName, @Pass
Outputs: None
Pseudocode:
Connect to the database
Query =

```
/*
Add User will be called for this so Admin references a valid and existing user
*/
INSERT INTO Admin
VALUES (@ID, @UName, @Pass);
```

Parse query
Execute query
Close connection to the database

Function: Delete Admin
Inputs: @ID
Outputs: None
Pseudocode:
Connect to the database
Query =

```
/*
Admin table will cascade delete so they will also be removed from can modify and so forth
*/
DELETE * FROM Admin WHERE ID = @ID;
```

Parse query
Execute query
Close connection to the database

Function: Select User
Inputs: @ID
Outputs: ID, Name, Username, Password
Pseudocode:
Connect to the database
Query =

```
SELECT *
FROM Users, Admin, Client
WHERE ID = @ID;
```

Parse query
Execute query
Close connection to the database

Function: Add User
Inputs: @ID
Outputs: None
Pseudocode:

Connect to the database
Query =

```
INSERT INTO Users
VALUES (@ID);
```

Parse query
Execute query
Close connection to the database

Function: Add Client
Inputs: @ID, @Name
Outputs: None
Pseudocode:
Connect to the database
Query =

```
/*
Add user will previously be called and added client will then correctly reference the
corresponding user
*/
INSERT INTO Client
VALUES (@ID, @Name);
```

Parse query
Execute query
Close connection to the database

Function: Delete Client
Inputs: @ID
Outputs: None
Pseudocode:
Connect to the database
Query =

```
DELETE * FROM Client WHERE ID = @ID;
```

Parse query
Execute query
Close connection to the database

Function:
Inputs:
Outputs:
Pseudocode: