# Smart Office

Group 12

Setup Guide

# Server Setup Guide

The server itself is self-sufficient and does not require much in order to set it up. The only thing that changes from time to time is the IP address that it connects to.

The IP address will differ for different users based on what machine they are connected to and/or if they have a static IP address.

Servers are normally set up with static IP addresses so clients always know where to log in. In our case we did not have a static IP address so we relied on whatever IP address was assigned to our machine each time we connected to the network.

If someone does not have a static IP address, they need to change the server address each time they fire it to make sure the server listens at the correct port and address.

To do so, follow the steps below:

- open the SmartOfficeServer.sln file using Visual Studio.
- In the solution explorer, look for a file named TCPServer.cs
- Go to **line 109** in that file and look for a field called iP Address.
- Change the address to whatever address your machine is currently connected to(IPv4 Address)
- Build and run the server.

The client application that is included has a similar IP address field at **line 38(SmartOfficeClient.cs)**. When using the product, you must make sure that the IP addresses mentioned in both the fields are identical otherwise the client and the server won't be able to communicate

## Server Setup

Key Points

- Having a static IP address is great. If not make sure that the client and server have the same IP address or it won't work!
- Line 109 for the server IP address, Line 36 for the client

# Database Setup Guide

## Database Setup

Key Points:

- MySQL database.
- User: **root**
- DB Password: **123**
- DB name: **SmartOffice**

We used a MySQL database in our project. To setup a database on your local server machine, please follow the steps below:

- Have a working MySQL service on your machine. If not, please follow: http://dev.mysql.com/downloads/mysql/ to setup a local MySQL service.
- Make sure you're logged in as the **root** user with a database password of **123.** If not, please create a new user with the above details.
- Execute the following commands in Table DB-Commands to generate a working instance of the MySQL database

## Table DB-COMMANDS: Use the commands in this table to create a working instance of a MySQL database

| Command | Description |
|---|---|
| Create database SmartOffice; use SmartOffice | The database used by the server |
| CREATE TABLE `delivery` ( `id_Delivery` int(11) NOT NULL AUTO_INCREMENT, `id_User` bigint(20) NOT NULL, `id_reciver` bigint(20) NOT NULL, `Request_Type` int(11) NOT NULL COMMENT '1: ordering coffee \n2: delivering mail to another employee \n3: delivering mail outside the office', `Start_Time` datetime NOT NULL, `End_Time` datetime DEFAULT NULL, `Status` int(11) NOT NULL DEFAULT '2' COMMENT '0 for fail\n1 for finished\n2 for in progress', PRIMARY KEY (`id_Delivery`), KEY `idUser_idx` (`id_User`), CONSTRAINT `idUser` FOREIGN KEY (`id_User`) REFERENCES `user` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION, CONSTRAINT `id_reciver` FOREIGN KEY (`id_User`) REFERENCES `user` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION ) ENGINE=InnoDB AUTO_INCREMENT=145 DEFAULT CHARSET=latin1 | Delivery table used to contain data about all the deliveries made |
| CREATE TABLE `notification` ( `id_notification` bigint(20) NOT NULL AUTO_INCREMENT, `id_user` bigint(20) NOT NULL, `id_sender` bigint(20) NOT NULL, `subject` varchar(100) DEFAULT NULL, `description` varchar(500) DEFAULT NULL, `time` datetime NOT NULL, PRIMARY KEY (`id_notification`), KEY `id_user_idx` (`id_user`), KEY `id_sender_idx` (`id_sender`), CONSTRAINT `id_sender` FOREIGN KEY (`id_sender`) REFERENCES `user` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION, CONSTRAINT `id_user` FOREIGN KEY (`id_user`) REFERENCES `user` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION ) ENGINE=InnoDB AUTO_INCREMENT=236 DEFAULT CHARSET=latin1 | Table used to contain all the notifications of the users |
| CREATE TABLE `robot` ( `id_robot` int(11) NOT NULL AUTO_INCREMENT, `battery_status` int(11) NOT NULL, PRIMARY KEY (`id_robot`) ) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=latin1 | Details of all the robots in an office |
| CREATE TABLE `user` ( `id` bigint(20) NOT NULL AUTO_INCREMENT, `username` varchar(100) NOT NULL, `password` varchar(100) NOT NULL, `Name` varchar(100) NOT NULL, `Age` int(11) DEFAULT NULL, `Email` varchar(100) DEFAULT NULL, `Phone` varchar(45) DEFAULT NULL, `Department` varchar(45) NOT NULL, `ImagePath` varchar(300) NOT NULL, PRIMARY KEY (`id`), UNIQUE KEY `username_UNIQUE` (`username`) ) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=latin1 | Details of all the users in the office |

| | |
|---|---|
| insert into user (username, password, Name, age, email, phone, department, imagePath) values ("Thinh","123","Thinh Nguyen","21","abc@gmail.com","999-999-9999", "Admin","kitten-2.jpg"); | To add a new user to the database. Change values accordingly |

# Unity Setup Guide

We used a game development software called Unity 3D (supported by certain Windows and Mac Operating Systems) to simulate our Smart Office.

Click here for system requirements.

Once the software is installed follow the instructions below:

- Double click the Unity Scene File named "Office" in the Assets folder in the Unity Client folder, and the scene will open.
- On the **Hierarchy** tab on the left, click on "Bot."
- On the bottom of the **Inspector** pane on the right, there will be a section for the Movement.cs script.
    - This is where adjustments can be made as specified in the User Documentation in order to visualize the desired functions of the Smart Office System.