

CMPE 207 FINAL PROJECT
PUBLIC- TRANSPORT INTERACTIVE CIENT-SERVER DESIGN README FILE

+++++

Steps to setup and execute server and client:

\$./servers SJ

Above command, starts the server of San Jose City (SJ). This will accept requests from the San Jose city users.

\$./testGlade SJ

This command opens up the UI for type 1 client. Upon execution of this command, User authentication UI opens up. Credentials for the user are added manually using MySQL commands.

\$./type2Client

This command displays UI for type 2 client.

\$./type3

This command opens up UI for type 3 client where in user can generate report of bus's location and time.

To simulate real time change in bus's location we have implemented a stub which changes bus's current location in our database. Below is the command to execute stub.c file

\$./stub.c

Complete Deployment instructions of our project:

Prerequisites: Before running the code, host PC should be installed with GTK3+ and Glade.

To install these, follow below instructions and command.

\$ sudo apt-get install libgtk-3-dev

Above command installs gtk (Gnome ToolKit) version 3 which enables PC to support Graphical User Interface.

\$ sudo apt-get install glade

Above command installs Glade which is used to build the User Interface of our project.

How to compile the code?

We have implemented Makefile to our project to compile all the files.

\$ make

This command compiles all the files and errors out if there is any compilation issue.

\$ make clean

Above command removes all the object files from the folder. Thus in order to execute code again, we need to compile again by executing make command.

We have tested above all the commands and found it successful in Ubuntu 14.04.3 version (linux).

SQL Commands:

We used SQL Database to store the information which is required to be processed for the client convenience. To install the MySQL from the linux command, we need the following functions,

```
sudo apt-get purge mysql-client-core-5.6
sudo apt-get autoremove
sudo apt-get autoclean
sudo apt-get install mysql-client-core-5.5
sudo apt-get install mysql-server
```

By using this, we can get the MySQL database access.

For setting up the SQL, we need to set up some validations as,

User: "root"

Password: "root"

Command is,

mysql -u root -p

To create a database, we need the following query as,

CREATE DATABASE proj207;

To use the database created,

USE proj207;

To create the table in a database for bus details entry,

CREATE TABLE bus_details(bus_no int, bus_location varchar(255), city varchar(255));

To create the table in the database for bus records,

CREATE TABLE bus_record(bus_number int, bus_location varchar(255), bus_time

To create the table in the database for user_details,

```
CREATE TABLE user_details(username varchar(255),password varchar(255));  
INSERT INTO user_details("test", "test" );
```

To insert elements into the table bus_details,

```
INSERT INTO bus_details(72,"Bassett", "SJ" );  
  
INSERT INTO bus_details(73, "Santa Clara", "SJ");  
  
INSERT INTO bus_details(66,"Central" ,"SJ");  
  
INSERT INTO bus_details(323, "Walmart", "SJ");  
  
INSERT INTO bus_details(81, "Cisco centre", "SJ");  
  
INSERT INTO bus_details(68, "Sunnywale", "SJ");  
  
INSERT INTO bus_details(181, "Palo Alto" ,"SJ");  
  
INSERT INTO bus_details(93, "Downtown", "SJ");  
  
INSERT INTO bus_details(522, "SAP Centre", "SJ");  
  
INSERT INTO bus_details(23,"Light rail Station", "SJ" );
```

To link the MySQL database with the C code to be executed, we need the following command in the command-line prompt,

```
gcc -o stub $(mysql_config --cflags) stub.c $(mysql_config --libs)
```