



# AIRLINE RESERVATION SYSTEM



Sneh C Dave

## Table of Contents

Overview .....	2
Get Started.....	3
Credentials to test application without creating new account .....	3
Application Setup Requirements: .....	4
Steps to setup IDE: .....	5
Steps to setup libraries for Database.....	6
Steps to setup libraries for REST .....	7
Steps to setup vcpkg .....	8
Steps to configure project properties.....	9
Run the Application from source .....	10
Run the application with Executables.....	11
Steps to use Application .....	12
Create a new account .....	12
Sign in to Existing account .....	12
Create a new flight.....	13
View Existing Flights.....	13
Delete a flight.....	13
Book a new flight .....	13
Book Ticket: Search All Flights .....	14
Book Ticket: Search by to and from destination.....	14
View Existing bookings.....	15

## Overview

- Three main parts of the application:
  - Client
  - Server
  - Database
- Client:
  - This part of the code handles the I/O operations, validations and communications with the server.
- Server:
  - This part of code handles the requests from the client and does the transactions with the database.
- Database:
  - The application uses PostgreSQL database hosted on AWS for storing all the data.
- Major Features of the application:
  - The client - server communication is based on REST architecture.
  - The server uses cpprestsdk which works on the modern asynchronous C++ task-based design to communicate with the client.
  - The database is hosted on AWS which provides data to server independent of location.
  - The database on AWS also provides extended data security and back up.
  - The server sends sql queries to the database hosted on AWS.
  - The application can handle the scenarios where multiple users are trying to book a same seat at same time. The application works on the first come first serve basis.
  - Administrator:
    - Create a new flight
    - View existing flights
    - Delete existing flight
  - Customer
    - Book a new flight
      - Search by all flights
      - Search by to and from destinations
    - View existing bookings

## Get Started

- [Run application by building from source](#)
- [Run application directly from executables](#)
- [Steps on how to run it](#)

## Credentials to test application without creating new account

Type	Username	Password
Administrator	testadmin	test123
Customer	testcustomer	test123

## Application Setup Requirements:

1. Operating system: Windows 7 64-bit
2. Tools: Git, Vcpkg
3. IDE: Microsoft Visual Studio 2015 (Version 14.0.25431.01 Update 3)
4. Database: libpq++ (C++ client API for PostgreSQL)  
<https://github.com/jtv/libpqxx>
5. REST: cpprestsdk (The C++ REST SDK is a Microsoft project for cloud-based client-server communication in native code using a modern asynchronous C++ API design.)  
<https://github.com/Microsoft/cpprestsdk>

## Steps to setup IDE:

1. Install Microsoft Visual Studio 2015 on a Window 7 64-bit machine.
2. Open Microsoft Visual Studio 2015 as an Administrator.
3. Make a new Visual C++ project. Select Empty Project. Enter the name of project as Server.
4. [Setup the Database libraries](#)
5. [Setup the Rest libraries.](#)
6. [Configure the project properties](#)
7. Add the .h and .cpp files into the project.
  - a. Right Click on project name Server in Solution Explorer.
  - b. Select Add Existing items.
  - c. Browse and select all the (.h) files in the Source\Server\include from the downloaded zip folder.
  - d. Browse and select all the (.cpp) files in the Source\Server\src from the downloaded zip folder.
8. Select Build solution from Build menu in Menu bar.
9. Once the build operation is completed, copy all (.dll) files in Source\DLL folder in the downloaded zip folder to the Release folder of the project.  
Ex: C:\Users\Sneh\Documents\Visual Studio 2015\Projects\Server\x64\Release
10. Make another new Visual C++ project. Select Empty Project. Enter the name of project as Client.
11. [Configure the project properties](#)
12. Add the .h and .cpp files into the project.
  - a. Right Click on project name Server in Solution Explorer.
  - b. Select Add Existing items.
  - c. Browse and select all the (.h) files in the Source\Client\include from the downloaded zip folder.
  - d. Browse and select all the (.cpp) files in the Source\Client\src from the downloaded zip folder.
13. Select Build solution from Build menu in Menu bar.
14. Once the build operation is completed, copy all (.dll) files in Source\DLL folder in the downloaded zip folder to the Release folder of the project.  
Ex: C:\Users\Sneh\Documents\Visual Studio 2015\Projects\Client\x64\Release

Go to [Run the application](#) section.

## Steps to setup libraries for Database

1. Copy Source\libpqxx-master folder in downloaded zipped folder to path C:\libpqxx-master

## Steps to setup libraries for REST

1. [Setup Vcpkg](#)
2. Open cmd as Administrator and change directory to root where vcpkg was installed.  
Ex: C:\Users\Sneh\Documents\Visual Studio 2015\Projects\Server\vcpkg
3. Run below command to setup all libraries of cpprestsdk. This step takes some time as it unpacks as required libraries.

```
vcpkg install cpprestsdk cpprestsdk:x64-windows
```



## Steps to setup vcpkg

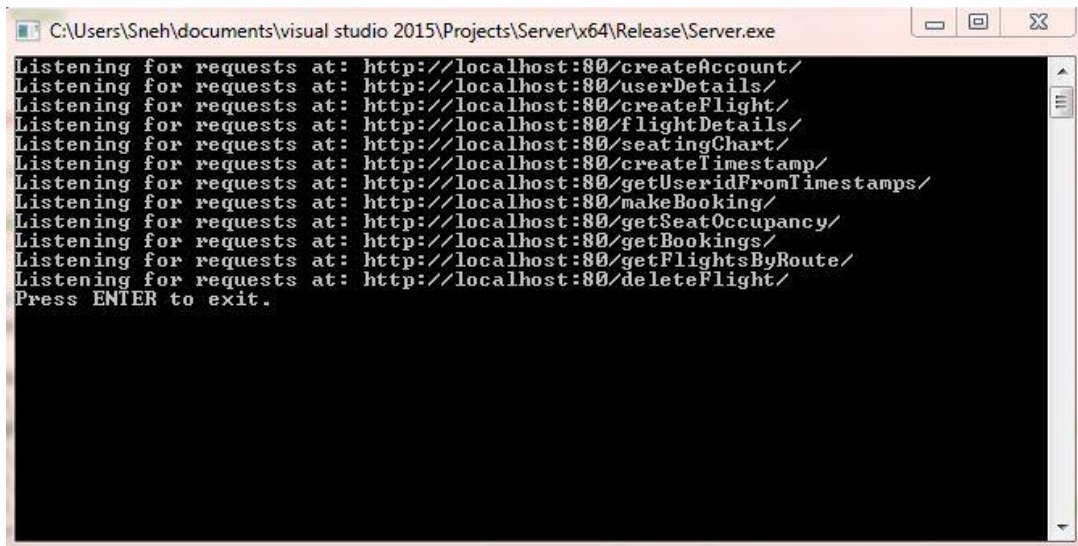
1. Install Git.
2. Open cmd as Administrator and change directory to Project root directory which was made in Visual Studio  
Ex: C:\Users\Sneh\Documents\Visual Studio 2015\Projects\Server  
Server is the name of the project
3. Git clone the master branch of vcpkg.  
git clone https://github.com/Microsoft/vcpkg.git
4. Change directory to vcpkg  
cd vcpkg
5. Run command given below to start installation of vcpkg. This step will take some time to complete.  
.bootstrap-vcpkg.bat
6. Run command given below to hook up user wide integration  
.vcpkg integrate install

## Steps to configure project properties

1. Set the configuration to Release mode.
2. Set the platform as x64.
3. In the solution explorer, right-click on project name Server and select properties.
4. Select General under tree structure: Configuration properties -> C/C++
5. Add below line in Additional Include Directories field.  
`C:\libpqxx-master\include;%(AdditionalIncludeDirectories);$(VcpkgRoot)\include`
6. Select General under tree structure: Configuration properties -> Linker
7. Add below line in Additional Library Directories field.  
`C:\libpqxx-master\lib;%(AdditionalLibraryDirectories);$(VcpkgRoot)\lib;$(VcpkgRoot)\lib\manual-link`
8. Select Input under tree structure: Configuration properties -> Linker
9. Add below line in Additional Dependencies field.  
`kernel32.lib;user32.lib;gdi32.lib;winspool.lib;comdlg32.lib;advapi32.lib;shell32.lib;ole32.lib;oleaut32.lib;uuid.lib;odbc32.lib;odbccp32.lib;libpqxx.lib;%(AdditionalDependencies)`
10. Click on Apply and Ok.

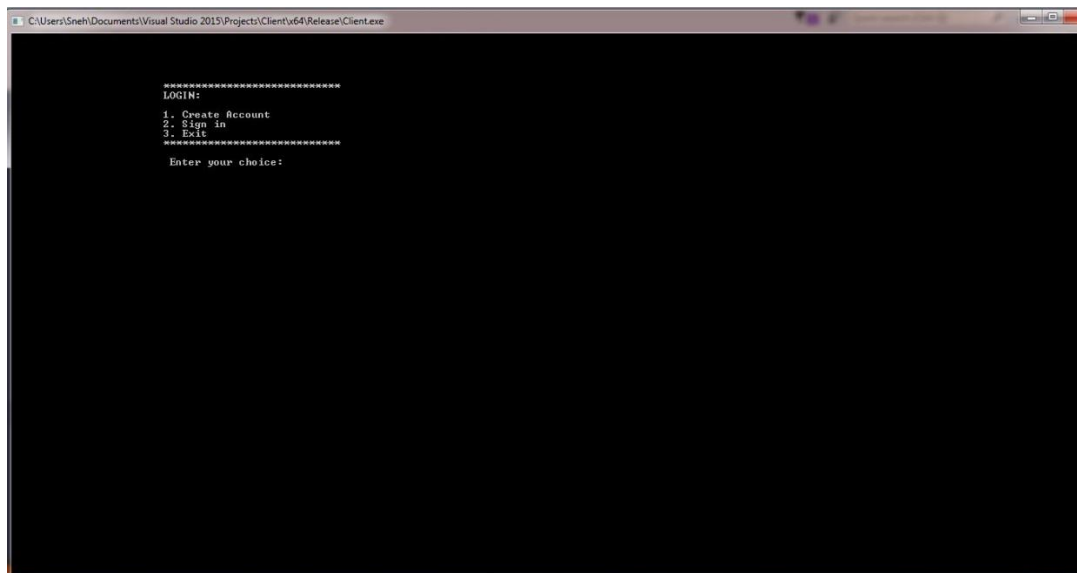
## Run the Application from source

1. After do all steps in [Setup the IDE](#), our project has been built and ready to run.
2. Go to Release folder of project Server.  
Ex: C:\Users\Sneh\Documents\Visual Studio 2015\Projects\Server\x64\Release
3. Right click on Server.exe and select run as Administrator (Mandatory step)
4. Now the Server should have started and should look as below.



```
C:\Users\Sneh\documents\visual studio 2015\Projects\Server\x64\Release\Server.exe
Listening for requests at: http://localhost:80/createAccount/
Listening for requests at: http://localhost:80/userDetails/
Listening for requests at: http://localhost:80/createFlight/
Listening for requests at: http://localhost:80/flightDetails/
Listening for requests at: http://localhost:80/seatingChart/
Listening for requests at: http://localhost:80/createTimestamp/
Listening for requests at: http://localhost:80/getUserIdFromTimestamps/
Listening for requests at: http://localhost:80/makeBooking/
Listening for requests at: http://localhost:80/getSeatOccupancy/
Listening for requests at: http://localhost:80/getBookings/
Listening for requests at: http://localhost:80/getFlightsByRoute/
Listening for requests at: http://localhost:80/deleteFlight/
Press ENTER to exit.
```

5. Go to Release folder of project Client.  
Ex: C:\Users\Sneh\Documents\Visual Studio 2015\Projects\Client\x64\Release
6. Right click on Client.exe and select run as Administrator (Mandatory step)
7. Now the Client should have started and should look as below.



```
C:\Users\Sneh\Documents\Visual Studio 2015\Projects\Client\x64\Release\Client.exe

*****
LOGIN
1. Create Account
2. Sign in
3. Exit
*****
Enter your choice:
```

## Run the application with Executables

- Prerequisites: Microsoft Visual Studio 2015 Redistributable x64 and x86, Microsoft Visual Studio 2013 Redistributable x64 and x86
1. If you have prerequisites installed, then go to Step 2. If not, install them with included setup files in MicrosoftRuntimeSetups folder in downloaded zipped folder.
  2. Open Server folder. Right click on Server.exe and select run as Administrator (Mandatory step)
  3. Server must be running for client to work.
  4. Open Client folder. Right click on Client.exe and select run as Administrator (Mandatory step)
  5. If the console window doesn't open maximized, then maximize the console window:
    - a. Right click on Title bar of the console window and select Properties.
    - b. Go to Layout tab.
    - c. Set the width of the window size to 180 and click ok.
  6. Now, client and server will be running.

Go to [Steps to use Application](#) section to see how to use it.

## Steps to use Application

- [Creating new account](#)
- [Sign in to existing account](#)
- [Create new flight](#)
- [View existing flights](#)
- [Delete a flight](#)
- [Book a new flight](#)
- [View Existing bookings](#)

### Create a new account

1. Enter 1 at main menu to create a new account.
2. Choose type of account:
  - a. Enter 1 for Administrator
  - b. Enter 2 for Customer
3. Enter the account details:
  - a. Enter the name of user (Only alphabetic input with spaces allowed)
  - b. Enter the age of user (Only integer from 1 to 125 allowed)
  - c. Enter the email id of user (Format must be of valid email id. Ex: [abc@pqr.xyz](#))
  - d. Enter the username (Only alphabetic string allowed without spaces. Username must be unique as code checks its existence in database)
  - e. Enter the password (Only alphanumeric input is allowed {A-Z, a-z, 0-9})
  - f. Confirm the password again (Password must match the previously entered password)
    - i. User gets three chances for entering the correct password. If not account creation will be failed.
4. Account will be created if all above steps are followed correctly and "Account Created!" message will be displayed.

### Sign in to Existing account

1. Enter 2 at main menu to sign in to existing account.
2. Choose type of account:
  - a. Enter 1 for Administrator
  - b. Enter 2 for Customer
3. Enter the Login details:
  - a. Enter username (Only alphabetic input allowed without spaces)
  - b. Enter password (Only alphanumeric input is allowed {A-Z, a-z, 0-9})
4. If entered details matches the entries in the database, sign in will be successful.
  - a. Sign in will be unsuccessful if following scenarios:
    - i. Customer tries to login as administrator and vice-versa
    - ii. Password doesn't match
    - iii. Username doesn't exist

### Create a new flight

1. Sign in as Administrator.
2. Enter 1 at menu which says "Create New Fight"
3. Enter flight details:
  - a. Enter flight name (Only alphabetic string allowed with spaces)
  - b. Enter departure airport (Only alphabetic string allowed with spaces)
  - c. Enter arrival airport (Only alphabetic string allowed with spaces. Must not be same as departure airport)
  - d. Enter departure time hours (Only integer from 0 to 23 is allowed)
  - e. Enter departure time minutes (Only integer from 0 to 59 is allowed)
  - f. Enter arrival time hours (Only integer from 0 to 23 is allowed)
  - g. Enter arrival time minutes (Only integer from 0 to 59 is allowed)
  - h. Enter seating capacity for economy class (Only integer allowed between 10 to 525)
  - i. Enter seating capacity for first class (Only integer allowed between 1 to 100)
  - j. Enter economy price (Only integer is allowed greater than 0)
  - k. Enter first class price (Only integer is allowed greater than 0)
4. After all details are entered "Flight Created!" message will be displayed

### View Existing Flights

1. Sign in as Administrator.
2. Enter 2 at menu which says "Display All Flights"
3. List of all flights in database will be displayed

### Delete a flight

1. Sign in as Administrator.
2. Enter 3 at menu which says "Delete Existing Flight"
3. List of all available flight will be displayed
4. Enter the flight id to delete (Only integer from available flight ids is allowed)
5. After successful deletion of flight, message "Flight Deleted" will be displayed.

### Book a new flight

1. Sign in as Customer.
2. Enter 1 at menu which says "Book a New Flight"
3. Enter the way you want to book the ticket:
  - a. Enter 1 for [Search All Flight](#)
  - b. Enter 2 for [Search by to and from destination](#)

### Book Ticket: Search All Flights

1. List of all flights in database will be displayed
2. Enter the flight id you want to book (Only integer from available flight ids is allowed)
3. Choose the class type:
  - a. Enter 1 for Economy class
  - b. Enter 2 for First class
4. Enter the seat number. (Only integer from available seat numbers is allowed)
  - a. Seats open are marked with O and reserved are marked with X
  - b. If seat is occupied, then error message will be displayed
5. Final Booking Summary will be displayed.
6. Confirm to make a booking:
  - a. Enter 1 for YES
  - b. Enter 2 for No
7. If you Enter 1:
  - a. Code will check for any other users trying to book ticket at same time.
  - b. Code will check for Seat occupancy again
  - c. After checking everything final booking will be done.
8. If you Enter 2:
  - a. You will come back at the Customer options menu

### Book Ticket: Search by to and from destination

1. Enter the departure airport (Only alphabetic input with spaces is allowed)
2. Enter arrival airport (Only alphabetic input with spaces is allowed. Cannot be same as departure airport)
3. If any flights exist between two places, then list of all such flights will be shown.
4. Enter the flight id you want to book (Only integer from available flight ids is allowed)
5. Choose the class type:
  - a. Enter 1 for Economy class
  - b. Enter 2 for First class
6. Enter the seat number. (Only integer from available seat numbers is allowed)
  - a. Seats open are marked with O and reserved are marked with X
  - b. If seat is occupied, then error message will be displayed, and you will come back at Customer options menu
7. Final Booking Summary will be displayed.
8. Confirm to make a booking:
  - a. Enter 1 for YES
  - b. Enter 2 for No
9. If you Enter 1:
  - a. Code will check for any other users trying to book ticket at same time.
  - b. Code will check for Seat occupancy again
  - c. After checking everything final booking will be done.
10. If you Enter 2:
  - a. You will come back at the Customer options menu

### View Existing bookings

1. Sign in as Customer
2. Enter 2 which says "View Existing Bookings"
3. List of all previous bookings will be shown