**COIT20257** **Distributed Systems: Principles and Development**

**Term 2, 2019**

**Assessment 2:**

**Peer-to-Peer Content Distribution and Distributed Query**

**Program use and test instruction**

**Prepared by:**

Student Name: Sandesh Bhetwal

Student Number: 12096531

CQU email address: sandesh.bhetwal@cqumail.com

Campus lecturer/tutor: Partha Gangavalli Sydney B/01

Unit Coordinator: Wei Li

# **Compilation and Starting the Application**

Steps:

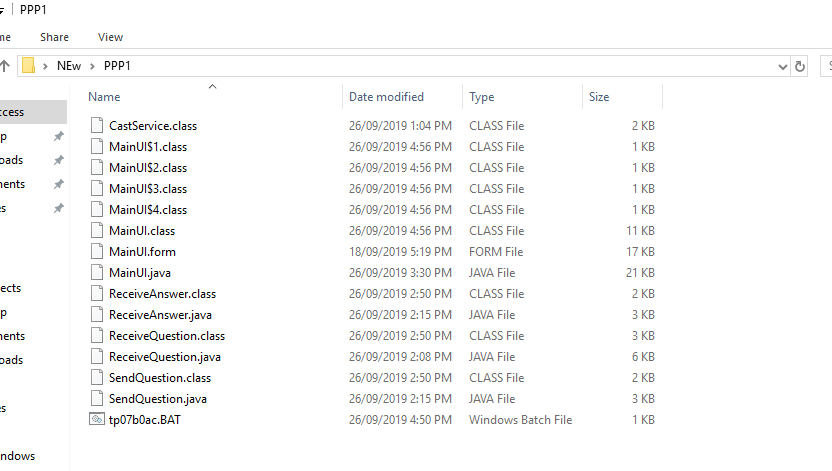
1. Extract the given file and save it in desktop.
2. Open the folder and the required files should be there as shown in Fig 1

Fig 1: Folder Structure for a Peer

1. Run a command prompt and navigate to the folder you just extracted the files. Then run the below code as shown in Fig 2

* javac MainUI.java
* java MainUI

*Note: To be able to run the program, we need to have Java installed in our system. If Java is not installed, go to* [*https://www.java.com/en/download/*](https://www.java.com/en/download/) *and download the Java.*

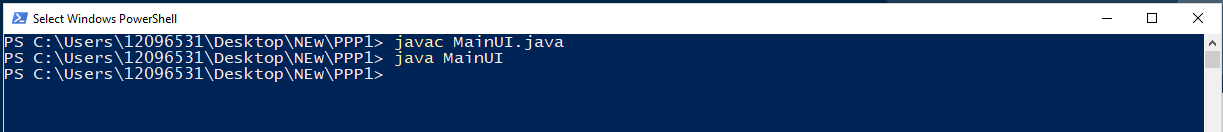


Fig 2: Command to run the MainUI

1. The main screen should be open as seen in Fig 3.

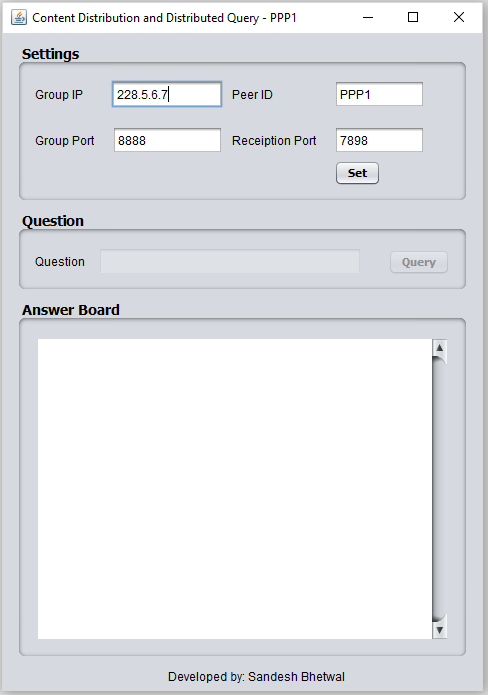
****

Fig 3: Main application

*Note: Initially only the Setting Panel at top is enabled, once we establish connection with peer overlay, the setting panel disables and the question panel is enabled. This helps in preventing any possible error before connection is established with peer overlay.*

*Also, other user at another location will start their peer and connect to the peer overlay.*

# **Using the system:**

Steps:

1. Firstly insert the correct value to input fields in the setting panel at the top.
2. In case a box is left empty, the program shows and error like Fig 4.

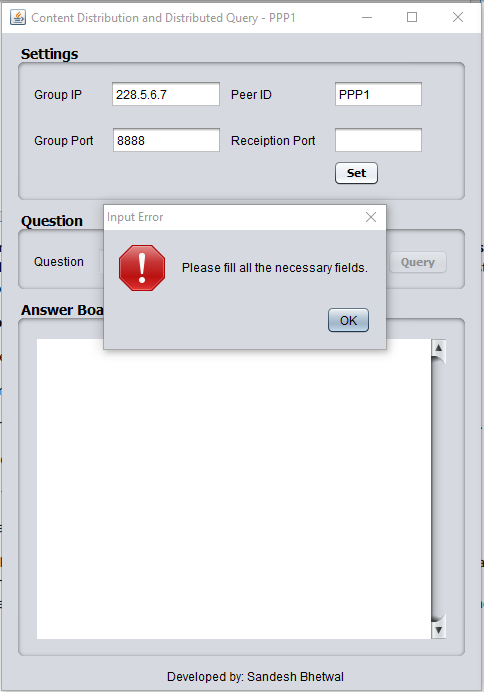


Fig 4: Error shown for empty input field in Setting Panel

1. If all the fields are filled correctly, then the program establish connection to the peer overlay and disable the setting panel for further edit. The main UI looks like Fig 5.

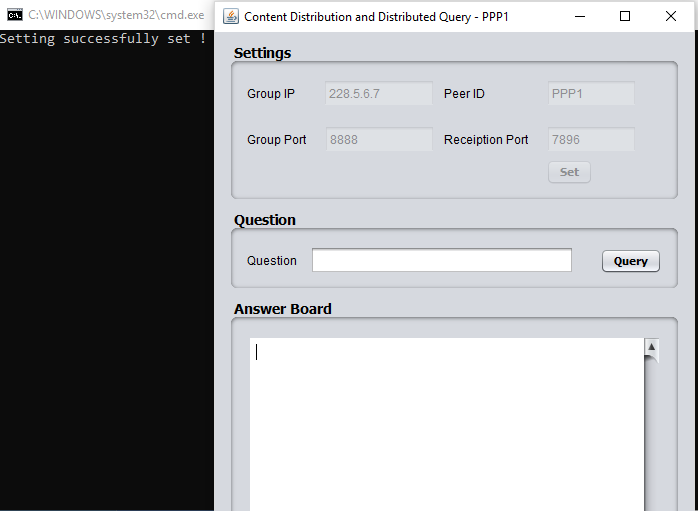


Fig 5: Successful set of setting and established connection with peer overlay

1. After connected with peer overlay, we can query our question using the question panel in the middle section. Question ranges from 1 to 9 only.

*The question should be in format of “Question 5”. ‘Question’ followed by a space and the question number.*

1. Ask the question. For example: try “Question 5” and press Query button.
2. If other peer are present in the group, the question is sent to the group and each peer answer you privately. The answer you received are displayed in Answer Board at the bottom as shown in Fig 6.

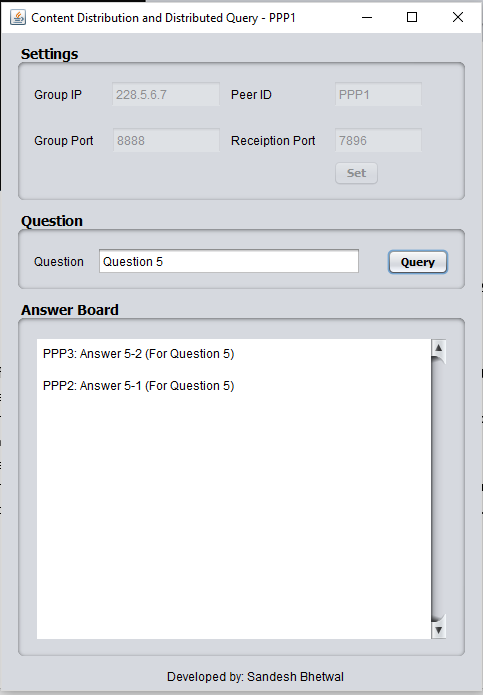


Fig 6: Question 5 asked by PPP1 and answer received

1. The question can be asked as many time as we want until the system is terminated or no more peer are present in our peer overlay.

# **Key issues on upgrading to Peer Overlay from client server:**

The main issue peer overlay solves is the distributions of total load over all the peers instead of just a server. In client server model the server handles all the load which will require more computing power and increase cost. The peer overlay removes this condition because in peer overlay each peer acts as both the client and the server at the same time.

The peer overlay make use of separate data unlike client-server model where all the required data are stored in the server. Also, peer overlay helps the user to achieve scalability and growth of the program.

In addition, peer overlay are economically beneficial because we do not need to setup separate server which cost a lot of money. Although peer overlay has some issues with security of data, as the whole peer overlay can see the data sent. This is overshadowed by numerous advantages of peer overlay.

# **Testing the System:**

Before moving on testing the system, let’s look at the set of question and answer each peer has. The question set is from Question 1 to Question 9. Question 4, 5 and 6 has multiple answers.

The answer each peer has is listed in the matrix below. So, PPP1 has answer for Question 1, 2, 4 and 6, PPP2 has answer for Question 3, 4, 5, 7 and PPP3 has answer for 5, 6, 8, 9.

Peers: Answers:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PPP1 | 1 | 2 | 4-1 | 6-1 |
| PPP2 | 3 | 4-2 | 5-1 | 7 |
| PPP3 | 5-2 | 6-2 | 8 | 9 |

So for simplicity, we take into consideration that for a single answer questions each Peer does not ask question which they already have answers to. So the question each pair should ask is listed below.

PPP1 can ask Question 3, 4, 5, 6, 7, 8, 9

PPP2 can ask Question 1, 2, 4, 5, 6, 8, 9

PPP3 can ask Question 1, 2, 3, 4, 5, 6, 7

*Note: In case the peer ask a single answer question which only they have the answer, there will be no response from other peer and the answer board is empty.*

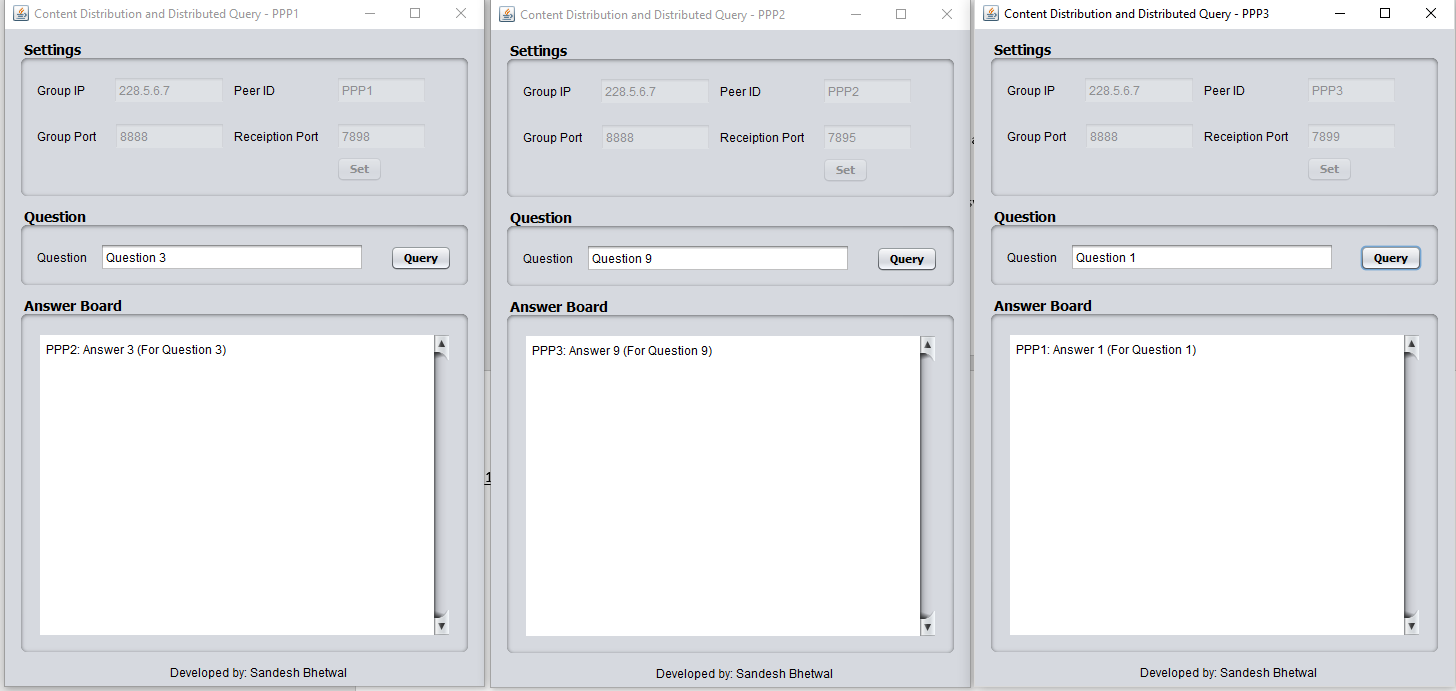
Test Case 1: Each Peer asks single answer question

Fig 7: Test case 1 – Asking a single answer question.

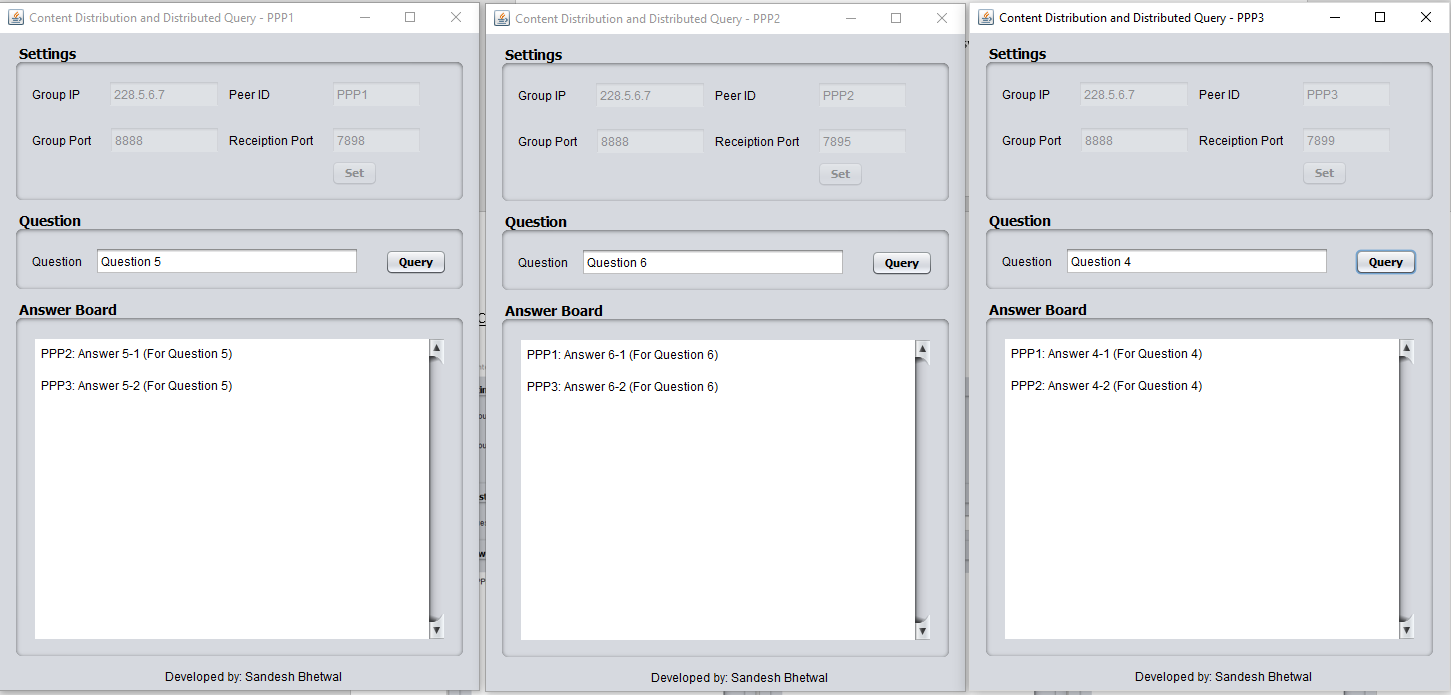
Test Case 2: Each Peer asks multiple answers question

Fig 8: Test case 2 - Asking multiple answers question

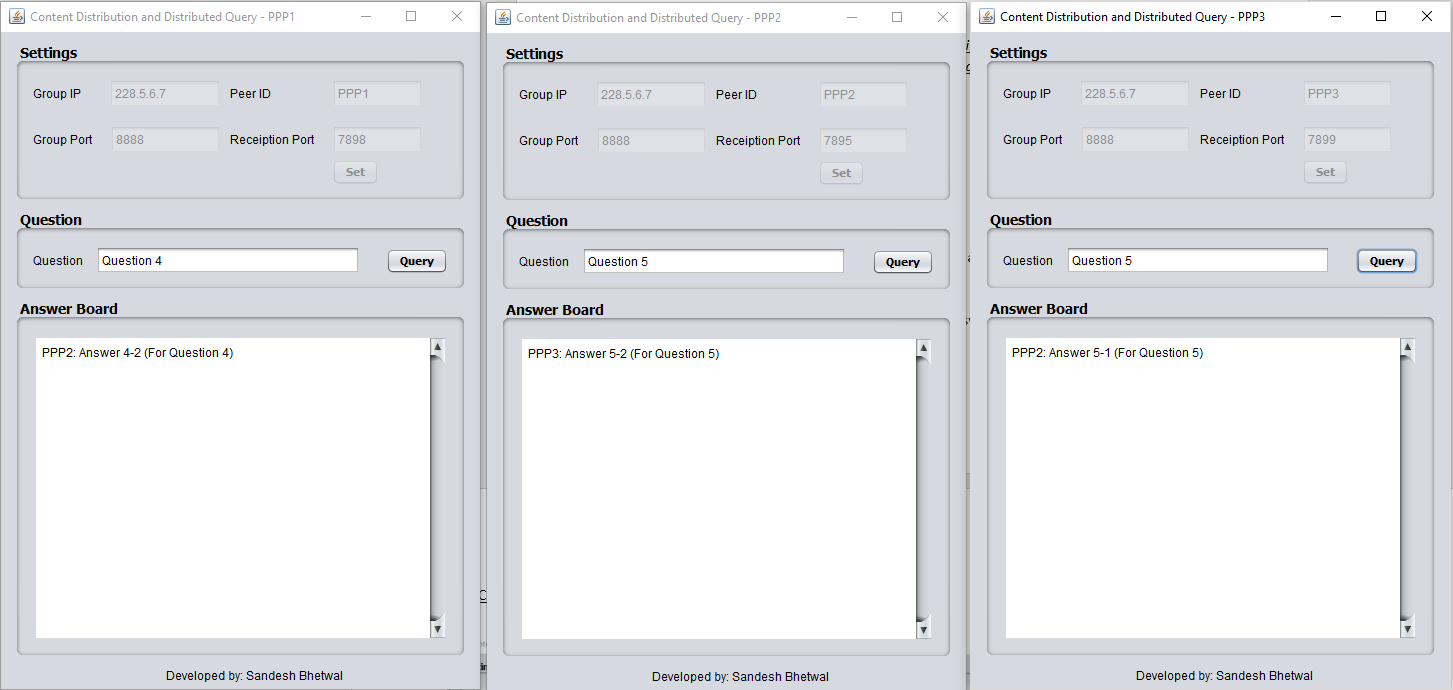
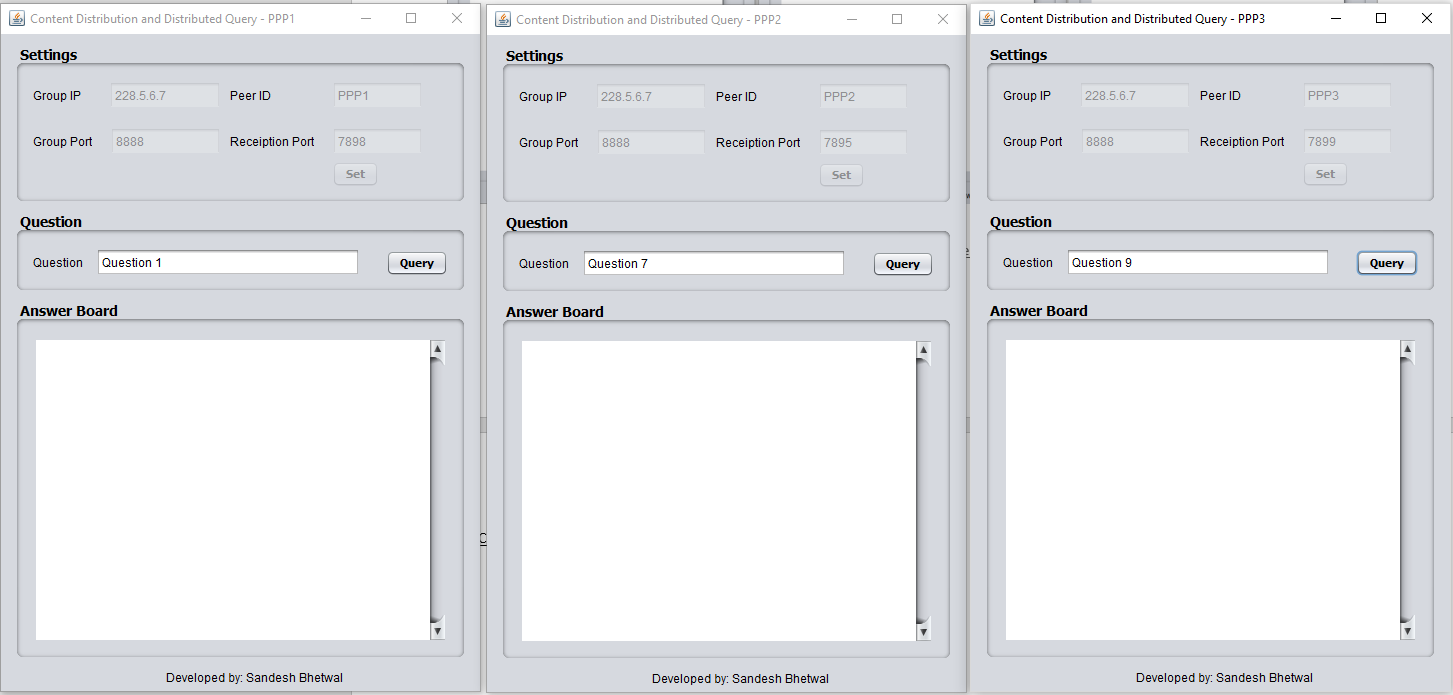
Test Case 3: Each peer asks multiple answer question but 1 answer with themselves

Fig 9: Test case 3 – Asking multiple answers questions but one answer with self

Test Case 4: Each peer asks single answer question with answer only with them

No Answer

No Answer

No Answer

Fig 10: Test case 4 – Asking questions that only self has the answer

Test Case 5: Each peer asks out of bound question

No Answer

No Answer

No Answer

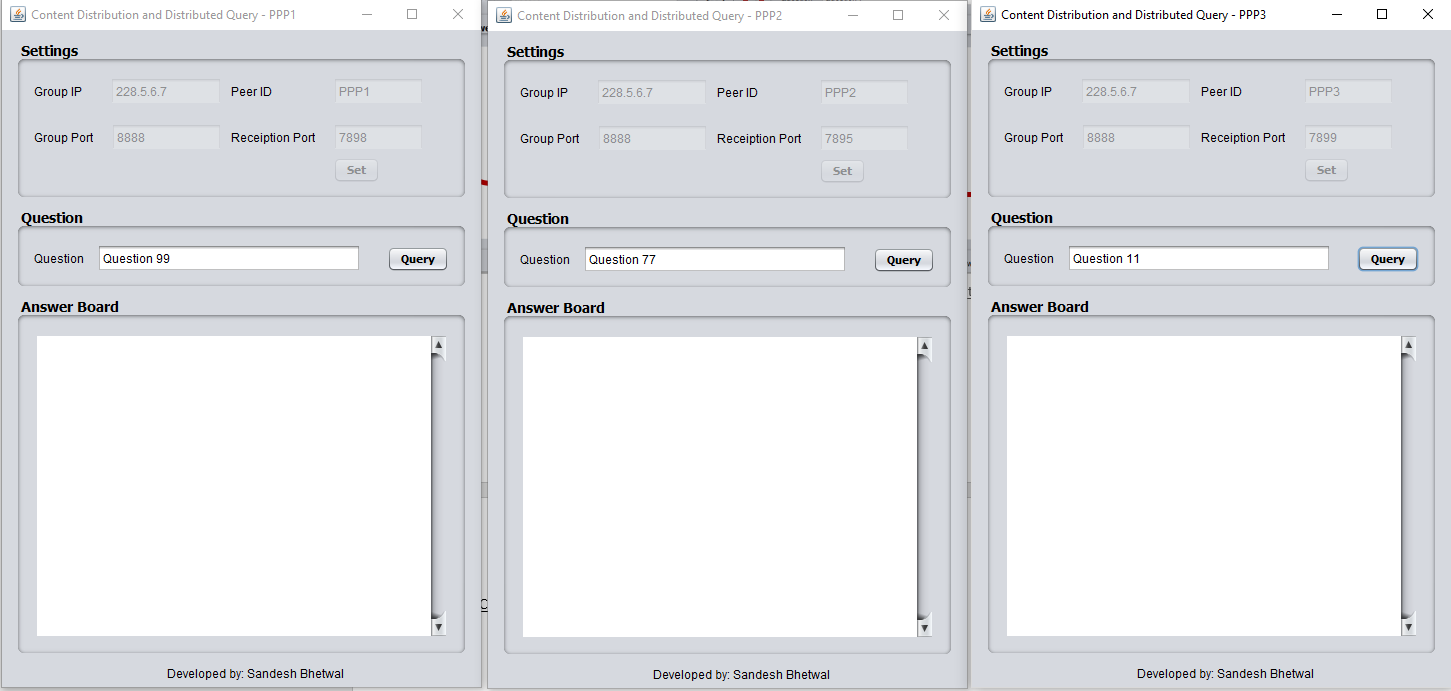


Fig 11: Asking out of bound question