

Middleware Architecture (IS3108 / SCS3203) - Assignment

Group Details and Task Implementation Document

Group Name: The Websockets

Group Members

Name	Registration No.	Index No.	Email Address
Benshekniel T.	2022/CS/023	22000232	2022cs023@stu.ucsc.cmb.ac.lk
Shimhan M.H.M	2022/CS/187	22001875	2022cs187@stu.ucsc.cmb.ac.lk
John Manuel S.P	2022/CS/081	22000811	2022cs081@stu.ucsc.cmb.ac.lk
Simaak M.N.M	2022/CS/191	22001913	2022cs191@stu.ucsc.cmb.ac.lk
Bawa M.M.A.J	2022/CS/022	22000224	2022cs022@stu.ucsc.cmb.ac.lk
Abdulla S.A	2022/CS/002	22000021	2022cs002@stu.ucsc.cmb.ac.lk
Jabir M.N.N	2022/IS/036	22020365	2022is036@stu.ucsc.cmb.ac.lk
Abdulla M.Z	2022/IS/001	22020012	2022is001@stu.ucsc.cmb.ac.lk

Project Overview

For This assignment our Group implemented a complete Publish Subscribe (Pub/Sub) middleware system using Java socket programming. Our system demonstrates a simple client-server architecture in Task 1 to a complex topic based message distribution system in Task 3.

Task Implementations

Task 1: Simple Client-Server Communication

Implementation Summary:

- The server listens on a specified port.
- The client connects via server IP and port.
- Messages sent by the client are displayed on the server terminal.
- The client disconnects when the message “terminate” is typed.

Task 2: Publishers and Subscribers

Implementation Summary:

- Server supports multiple client connections.
- Clients identify as a PUBLISHER or SUBSCRIBER.
- Messages from PUBLISHERS are broadcast to all SUBSCRIBERS.
- Publishers don't receive each other's messages.

Task 3: Topic Based Filtering of messages

Implementation Summary:

- Each publisher and subscriber is associated with a specific topic.
- The server maintains subscriber lists topic wise.
- Messages are only sent to subscribers of the same topic.

Task 4: Documentation of improved Architecture

To eliminate the single point of failure, a **distributed Pub/Sub architecture** is proposed with,

- Multiple clustered Servers
- Load balancing mechanisms to avoid Failure
- Message Broker for delivery of Messages
- Service Detection and Registration

Key Benefits:

- High availability
- Fault tolerance
- Highly Scalable

Conclusion

Our project demonstrates a working Pub/Sub middleware also including an improved and scalable distributed system. Each team member contributed to the implementation, testing, and documentation.