Joshua Van Deren, Jacob Lai, Grant Savage, Joy Tan

April 2, 2018

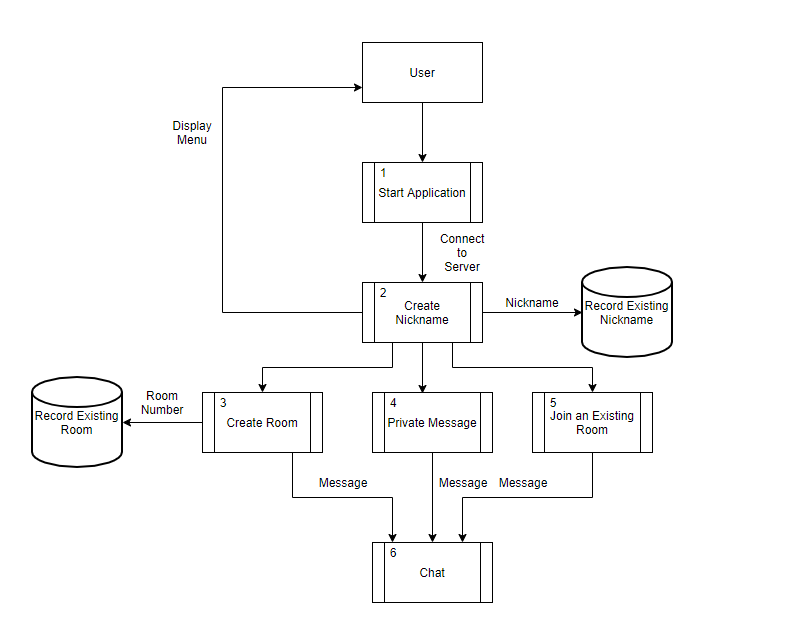
SE310

Siewert

**SE310, Analysis and Design of Software Systems**

**Exercise #5 – Architecture and Requirements Finalization for Proposed Project**

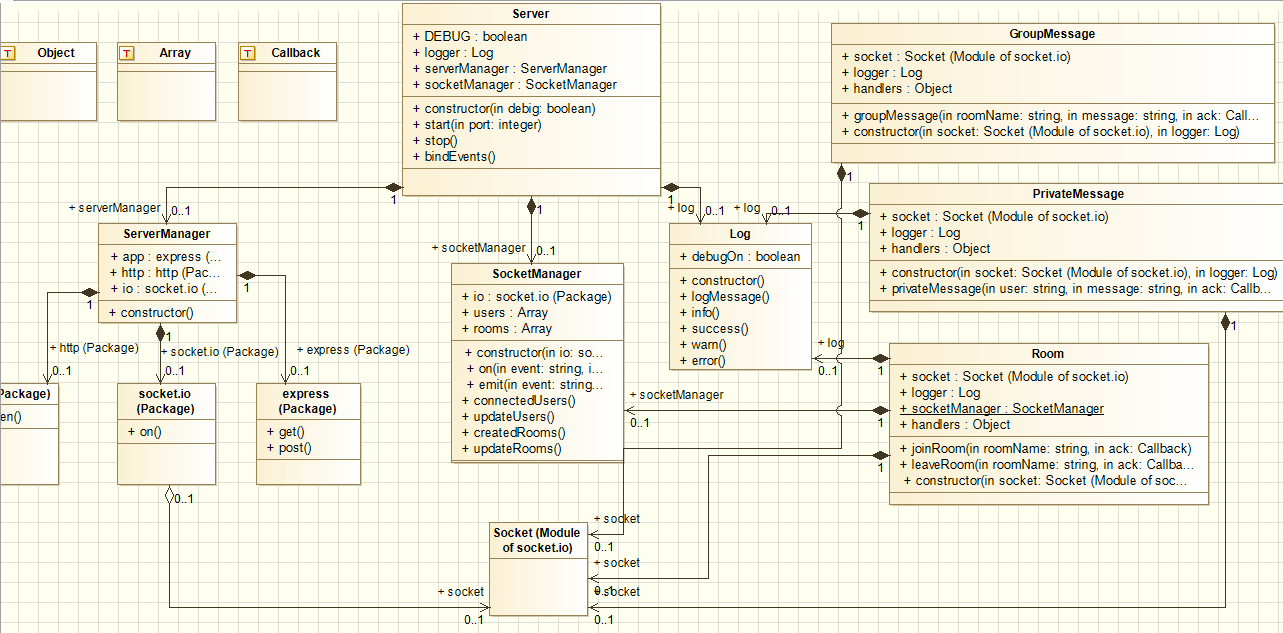
1. Data Flow Diagram



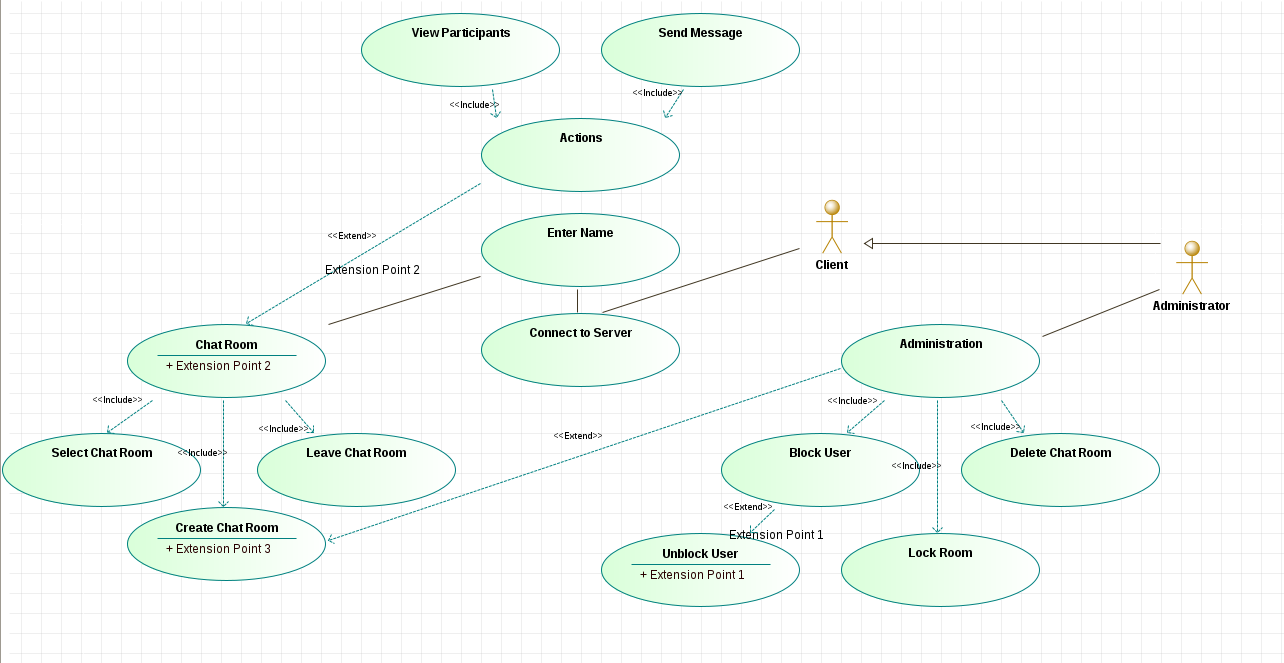
We decided to provide a Data Flow Diagram instead of the single page block diagram since we already have one on our Github page. The cylindrical symbol refers to data storage. The numbered rectangles refer to processes of the system.

1. We have updated our UML class diagram and OIM diagrams, they are now consistent with our prototype. Here are the diagrams we picked to further refine:

**Class diagram**



**Use case diagram**



Based on the feedbacks we received from code walkthroughs, we made some minor changes to our use case diagram. We added an Administrator actor which inherits from the Client actor (Since an administrator is a client), and it will handle management of the room which it is appointed to be an administrator of. We changed the use case “Management” to “Administration” based on the feedback.

1. Capability requirements updates:

* Must generate an ID made of a string of random characters and numbers for every user that connects to the server.
* Every user must have the ability to create a room.
* Every user must have the ability to leave a room.
* Every user must have the ability to send a message.
* Every user can join either a one on one chat or a group chat.
* A user can only be in one chat room or in none at any given moment.
* The user that creates a room becomes the administrator of that room.
* Once the administrator of the room leaves, all participants in the room are kicked out and the room is deleted.
* The name of any room is created by the user and no two rooms can have the same name.
* Every user has a designated random color for their name when their messages appear in the chat.
* Every room has a variable keeping track of the number of participants in the room and the ID of those participants.

We have decided to take out a few requirements and defer them, they are: “Any user within a room can choose to lock the room so that no other user can enter that specific chat room”, “Any user within a locked room can choose to unlock that room”, “Any user has the ability to block another user or to unblock those they have blocked.” We will judge accordingly if we need to implement these features in the future.

Performance requirements:

* The server application must be deployed on a Virtual Private Server that has at least 1GB of RAM, 1 vCPU, and 25 GB of disk space.
* The client must be able to connect to the server within 10 seconds of startup. Otherwise, the client will abort.
* Upon client disconnection the client shall attempt to reconnect to the server within 10 seconds. Otherwise, the client will abort.
* The delay time between message sending and receiving in a room shall be the same across all clients regardless of how many clients are connected to a room.

We have decided to take out the real-time requirement for now because we do not have deadlines constraints for the system.

Derived requirements:

We do not have changes for derived requirements at this point.

We use the Github Kanban to manage the progress of the project, we have been constantly updating the project Kanban and are progressing with a good pace. Please refer to our Github page for details. We are going to upload a separate document that states our individual contribution to the project.

1. Use-case to Requirements tracing analysis:

R1: Send Message  
R2: Join Room  
R3: Leave Room  
R4: Colored Name  
R5: Create Nickname  
  
U1: Customize Nickname  
U2: Peer-to-Peer Blocking  
U3: Unblock User from Room  
U4: Block User from Room

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Requirements | Priority Weights | U1 | U2 | U3 | U4 |
| R1 | 4 |  |  |  |  |
| R2 | 3 |  |  |  |  |
| R3 | 3 |  |  |  |  |
| R4 | 5 | X | X | X | X |
| R5 | 1 | X |  |  |  |
| R6 | 1 | X |  |  |  |
| Score | | 7 | 5 | 5 | 5 |

Therefore, we will work on U1 first at this stage.