ARCHITECTURE NOTEBOOK

Audience Response System

CONTENTS

SPRING BOOT BACK-END SERVER UML CLASS DIAGRAM

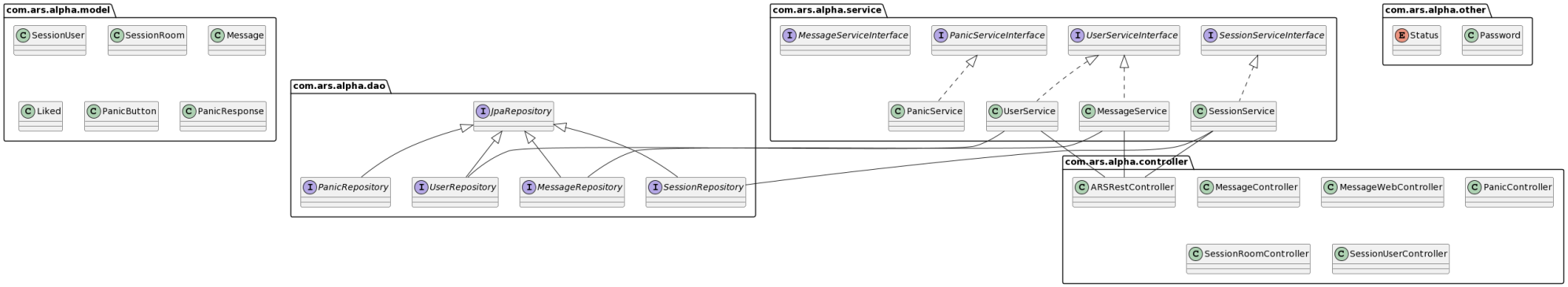


FIGURE 1- UML Class Diagram representation of back-end spring boot server. Does not include classes provided by libraries. Generated using PlantUML. A non-compressed version exists in the source document folder.

UML Use Case Diagram

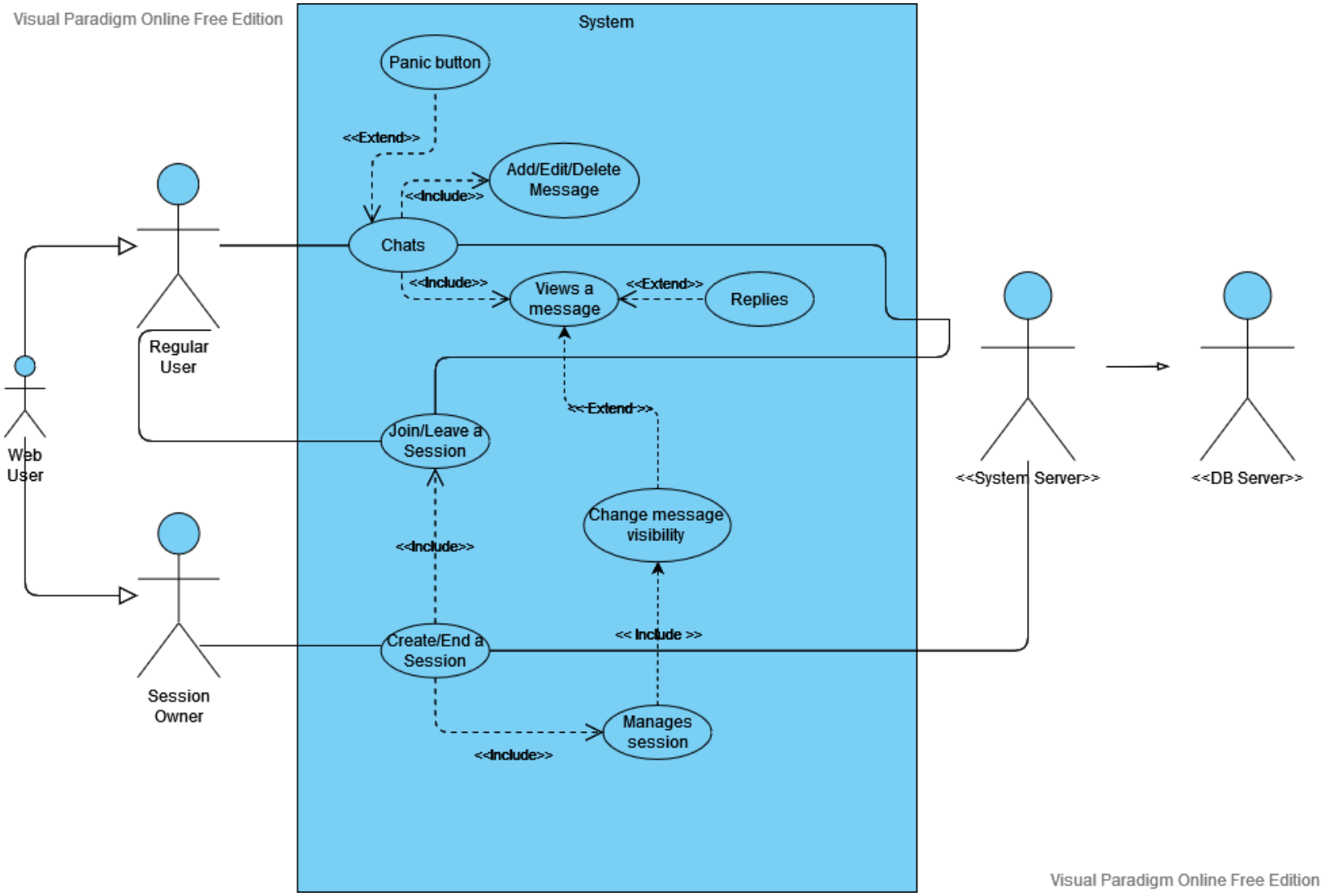


FIGURE 2-

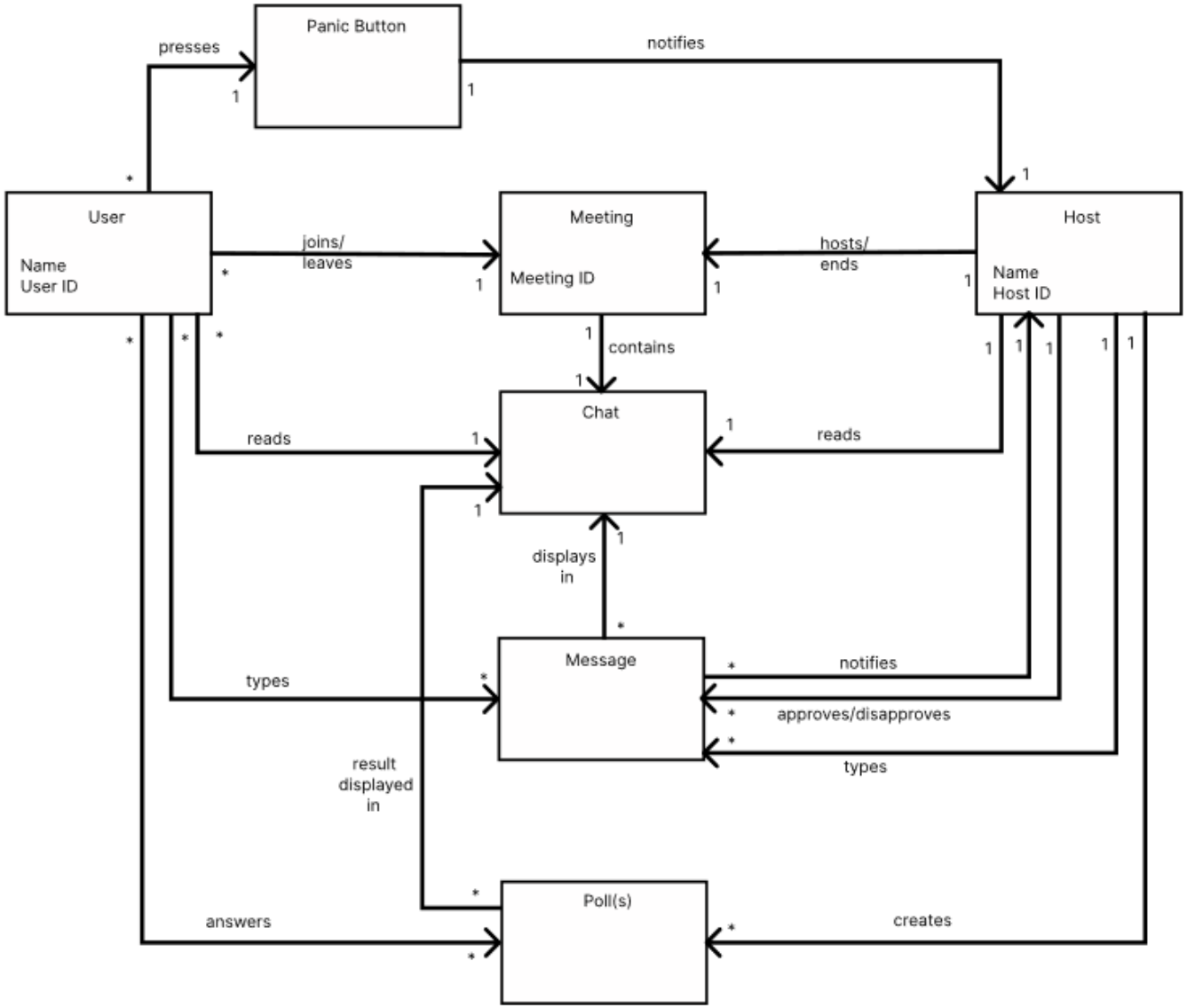
UML Use Case Diagram.

Generated using Visual Paradigm Online Free Edition. Shows interaction between front-end web users and the back-end Spring Boot server and the Microsoft SQL Server Database. Diagram excludes postponed Polling/Voting feature.

DOMAIN MODEL DIAGRAM

FIGURE 3-

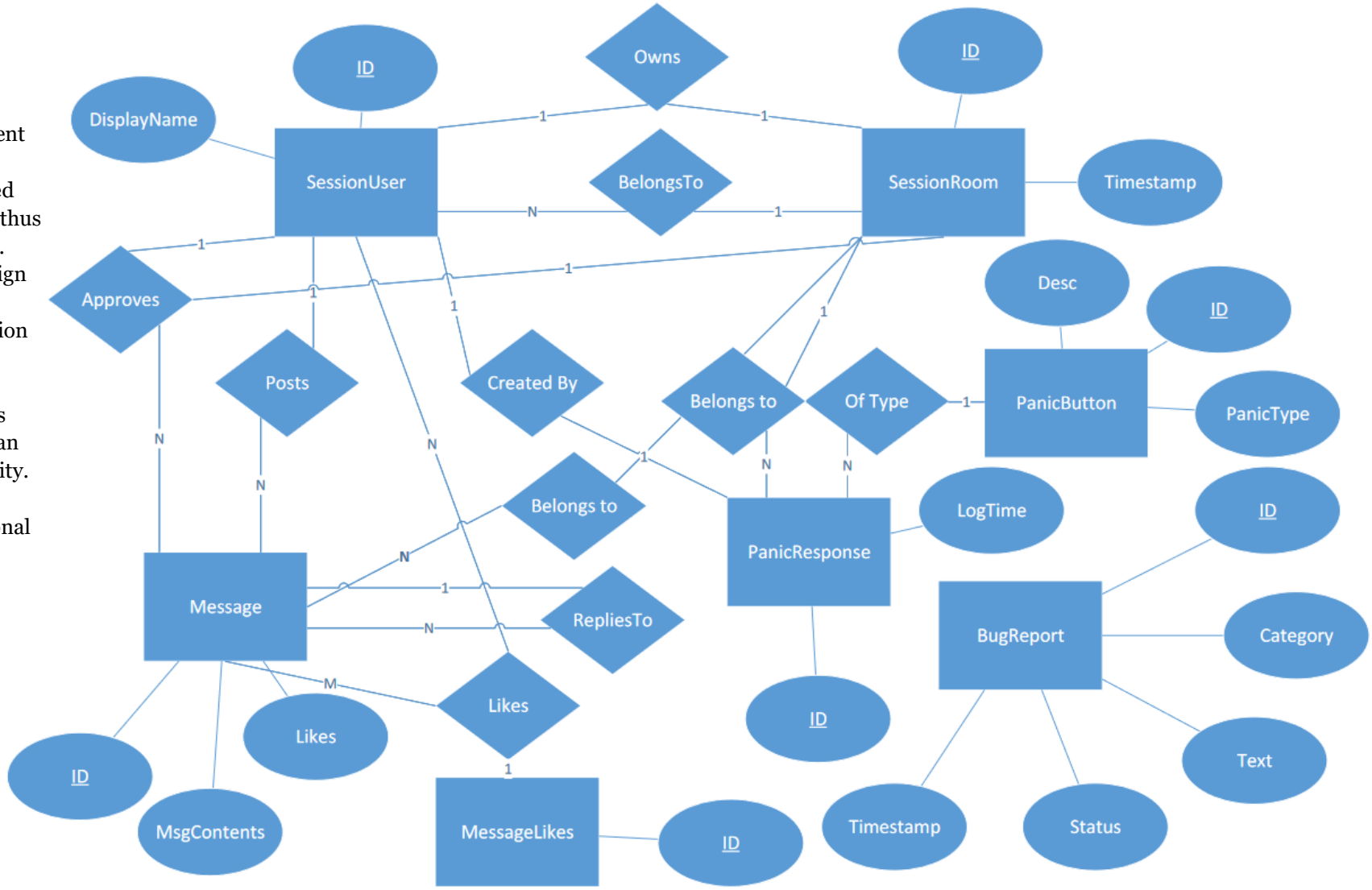
Domain Model Diagram between session host, session use, and the application. Arrows represent direction of action. The action is listed on each arrow. Numbers represent number of possible action actions per object, with * meaning an unlimited number. Rectangles represent objects with important information stored within.



ENTITY-RELATIONSHIP (ER) DIAGRAM

FIGURE 4-

ER Diagram of the MSSQL Database. Contains all current tables. Tables about Polling/Voting were dropped from the databases and are thus excluded from this diagram. Due to limitation of the design tool, this diagram does not accurately reflect participation constraints. Assume all relationships require participation except in cases were doing so would cause an issue with referential integrity. This ER Diagram is further elaborated on in the Relational Schema.



DATABASE RELATIONAL SCHEMA

FIGURE 5-

Relation Schema to explain the ER-Diagram above. Each item denotes a table with its attributes inside parentheses. Underlined attributes are primary keys. The foreign key section details every existing foreign key in the database. For example, UserSession(SessionID)->SessionRoom(ID) means that the attribute "SessionID" in the SessionUser table has a foreign key to the "ID" attribute in the SessionRoom table.

Relational Schema

SessionUser(ID, DisplayName, SessionID)

SessionRoom(ID, OwnerID, SessionPassword, Timestamp)

Message(ID, SessionID, MsgContents, PosterID, ReplyTo, Likes, Approved)

BugReport(ID, Category, BugText, Status, Timestamp)

PanicButton(ID, PanicType, Desc)

PanicResponse(ID, PanicType, Panicker, SessionRoom, LogTime)

MessageLikes(ID, LikerID, MessageID)

Foreign Keys:

SessionUser(SessionID) -> SessionRoom(ID)

SessionRoom(OwnerID) -> SessionUser(ID)

Message(SessionID) -> SessionRoom(ID)

Message(PosterID) -> UserSession(ID)

Message(ReplyTo) -> Message(ID)

PanicResponse(PanicType)->PanicButton(ID)

PanicResponse(Panicker)->SessionUser(ID)

PanicResponse(SessionRoom)->SessionRoom(ID)

MessageLikes(LikerID)->SessionUser(ID)

MessageLikes(MessageID)->Message(ID)

SAMPLE API CALL DIAGRAM

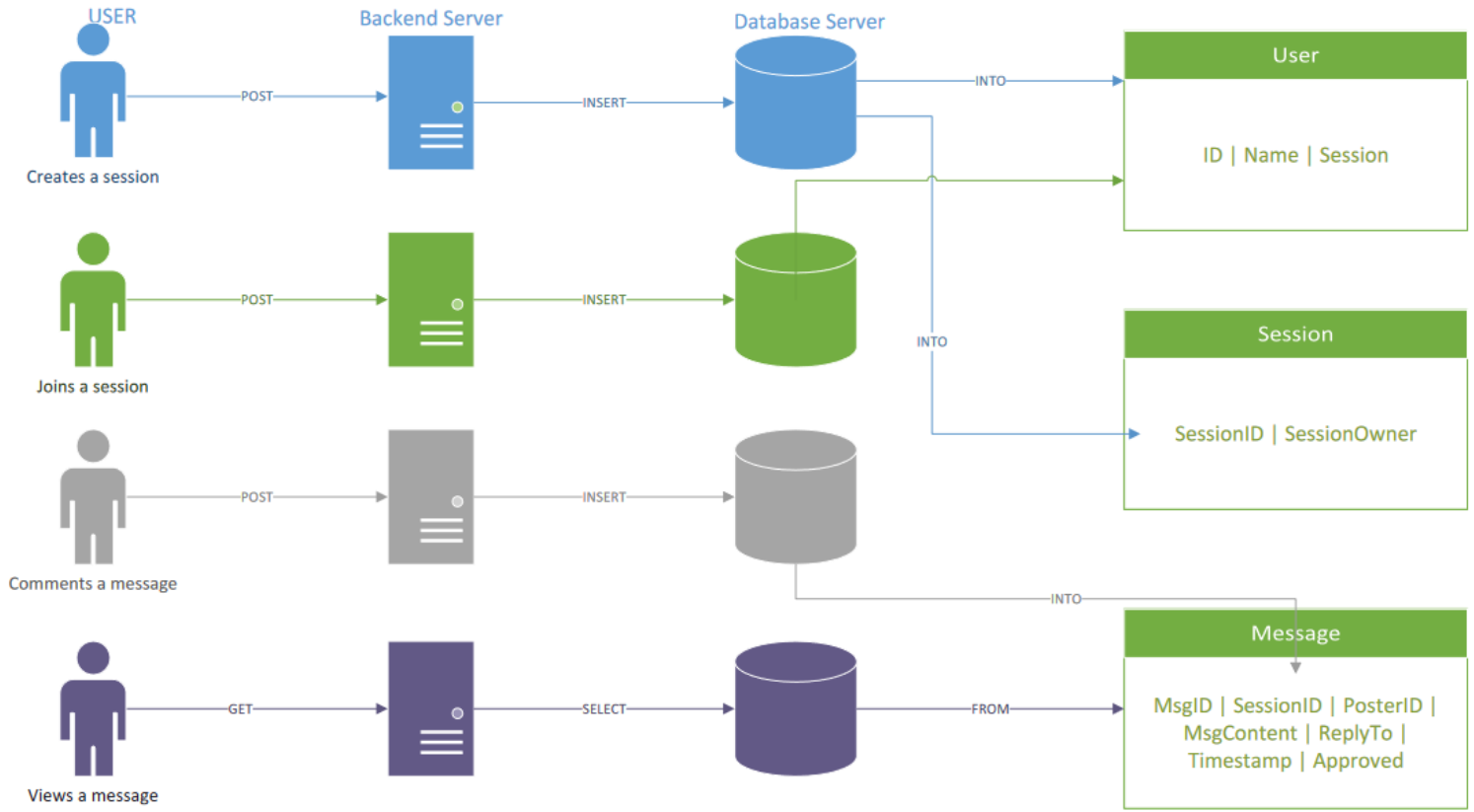


FIGURE 6-

This internal diagram shows a few basic API calls to the back-end server and the subsequent back-end calls to the Database server. Each color denotes a different action. Each line shows direction with the API Call or SQL Statement denoted on the line. The PERSON icon represents a user the RECTANGULAR SERVER icon represents the Spring Boot server, the CIRCULAR SERVER icon represents the database, and the RECTANGULAR BOXES represent database tables.