Assignment 3

Advanced Programming 1

1. Project Proposal: Project Relevance

The modern streaming industry (Twitch, YouTube Live) is experiencing content overload. Viewers are becoming passive consumers, and maintaining their attention (retention rate) is becoming increasingly difficult. Standard gameplay is no longer enough – audiences demand interactivity and dopamine rewards.

Our project, CS2 LiveDrop, addresses this problem through gamification of viewing. We are creating a system that synchronizes in-game events in Counter-Strike 2 (via Game State Integration) with real-world rewards for viewers. The project's relevance stems from the growing demand for interactive mechanics (RNG, Betting) that allow viewers to feel a sense of ownership in the streamer's success.

When a streamer "Aces" (kills the entire opposing team), the viewer receives not just emotional stimulation, but also a chance to win a skin. This creates a mutually beneficial ecosystem: the streamer gains a loyal audience and donations to create "prize pools," while viewers experience excitement and rewards.

1. Project Proposal: Competitor Analysis

There are two types of competitors in the market, but neither offers our functionality:

1. Roulette sites (KeyDrop, Cybershock,CSGO-run etc.): These platforms allow you to win skins, but they are completely disconnected from the context of the stream. The user simply spins the wheel without interacting with the content creator.
2. Twitch Channel Points / Predictions: Twitch's built-in tools allow you to place bets, but the rewards are useless channel points that cannot be converted into real-world value.

Our advantage: We combine these two models. Our system operates in real time, responding to specific player skills (Headshot, Bomb Plant). This is a unique selling point (USP) that our competitors do not offer.

1. Project Proposal: Target Audience

1. CS2 Streamers (Micro & Mid-tier): They desperately need tools to quickly build an audience and increase chat activity. Our tool serves as a catalyst for growth. Before revealing your charisma, streamer needs to growth the audience, and only then make a content on him own, which will be the best way.

2. Viewers/Gamblers: People who enjoy CS2 and skins love the thrill of the game, but want transparent winning conditions tied to the streamer's skill, not hidden casino algorithms.

1. Project Proposal: Planned Features

To implement the technical requirements of the course, we are planning the following functionality:

1. CS2 GSI Listener: HTTP server for receiving JSON packets from the game client (handling Kill, Death, and Round Win events).
2. Live Lottery Engine: A module that instantly selects a winner from the list of active viewers when a trigger (event) occurs.
3. Crowdfunding Cases (CRUD): Streamers can create custom cases, viewers can add funds to their bank (Update), and when the required amount is reached, a drawing takes place.
4. User Authentication & Wallet: A system for registering users and managing their virtual balance/inventory.

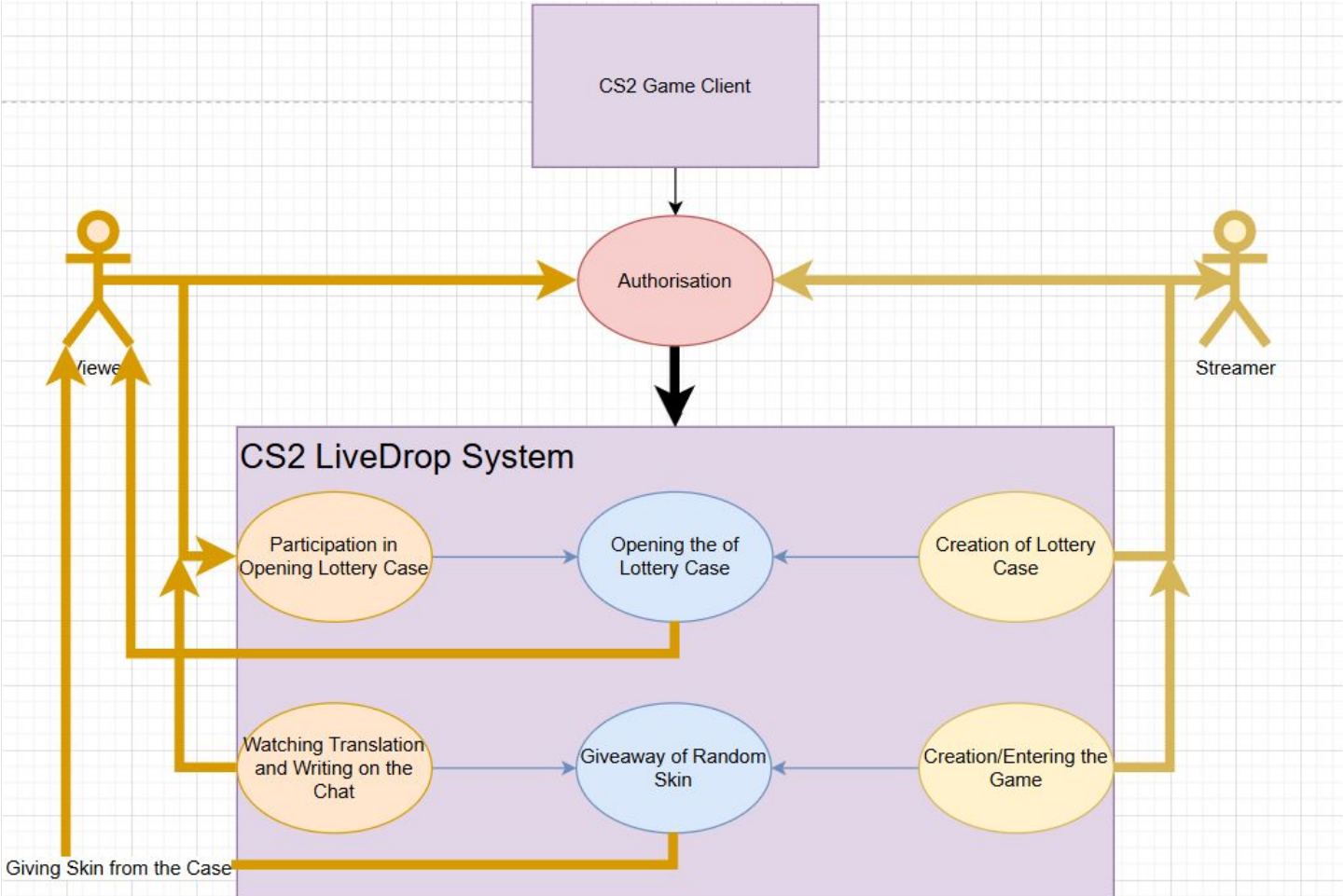
We are planning to create a fully-functional Telegram App to reach all of these goals, because it provides best environment to manipulate to user with relative UI, not even opening another app. But stillm work is on, and everything can be changed to still the WEb-service.

2. Architecture & Design: Modules Description

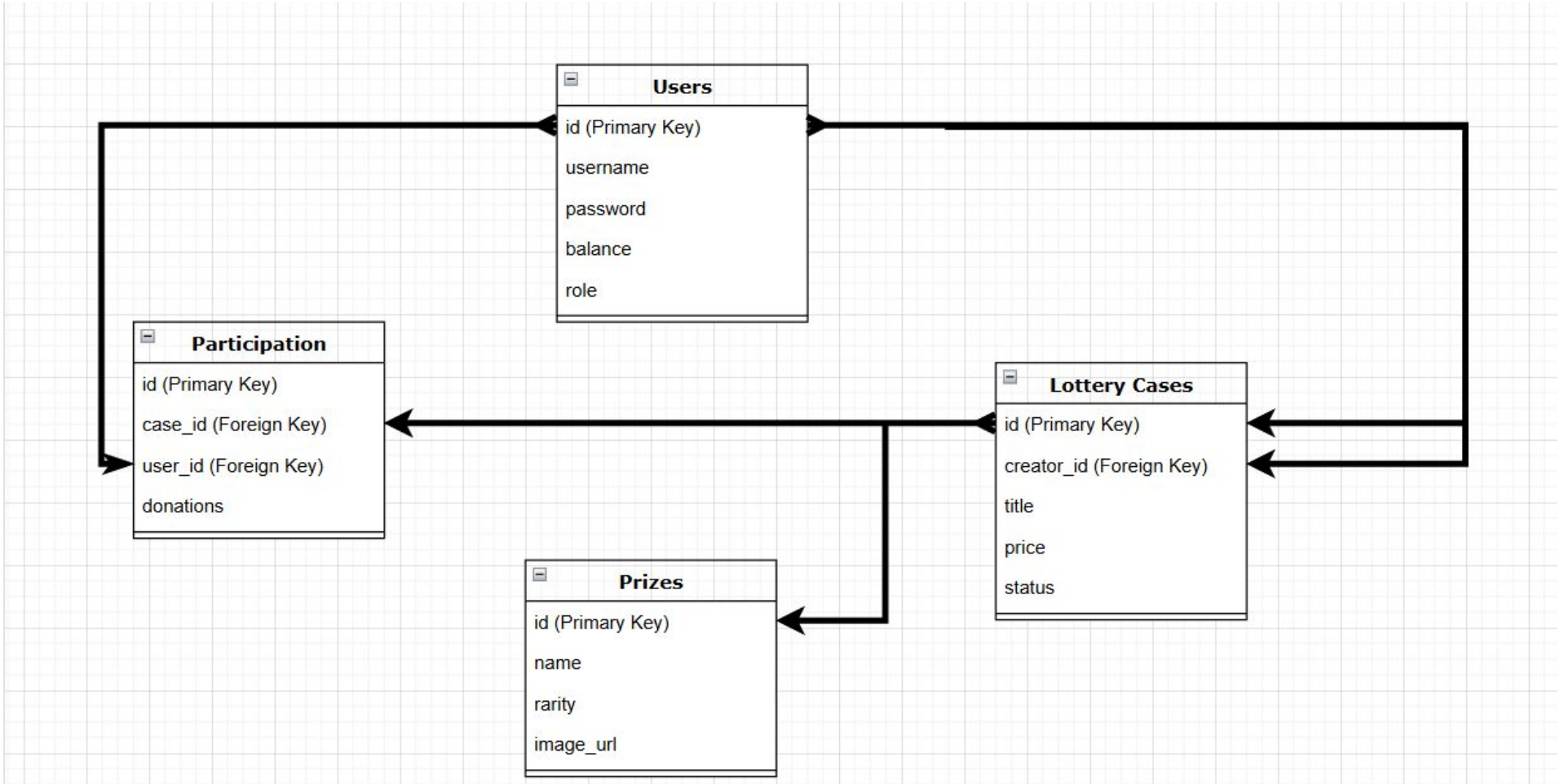
We use a monolithic architecture in Go, at least for now, because it provides ease of deployment and high performance for handling real-time HTTP requests from the game.

1. cmd/server: The entry point to the application. Initializes the connection to the database and starts the HTTP server.
2. internal/auth: Responsible for user registration and authentication (Session/JWT).
3. internal/gsi (Game State Integration): Parses incoming JSON requests from the Valve CS2 Client, filters unnecessary data, and highlights key events (kills, match state).
4. internal/lottery: Contains the RNG (Random Number Generation) business logic. Determines winners based on weighted probabilities (the higher the donation/activity, the higher the chance).
5. internal/db: PostgreSQL, or Sqllite with the MySQL, integration layer.

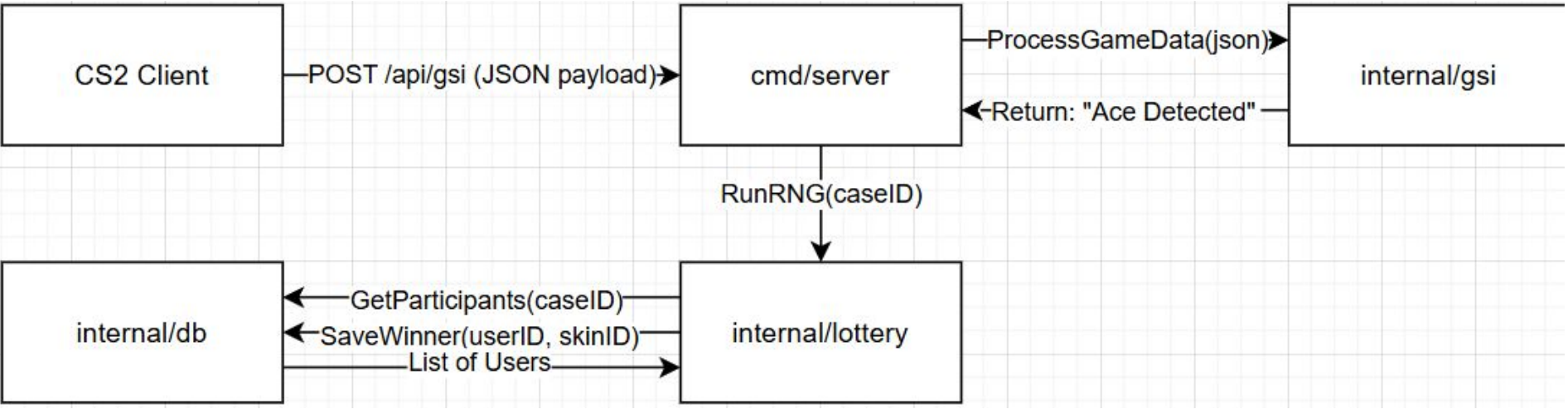
2. Architecture & Design: Use-case Diagram



2. Architecture & Design: ERD-Diagram



2. Architecture & Design: UML Diagram



3. Project Plan

Sanzhar: Backend Core, GSI Integration, Lottery Logic

Nurali: Auth System, User Management (CRUD), Database Setup.

Weeks	Task	Responsible	What's exactly
Week 7	Project Setup & Architecture	Team	Initializing a Git repository, designing a database, writing a Proposal.
Week 8	GSI Receiver Implementation	Sanzhar	Creating an HTTP handler to receive data from CS2. Parsing JSON.
Week 8	Authentication System	Nurali	Implementing Registration/Login. Creating middleware to protect routes. (Requirements: Auth).
Week 9	Lottery RNG Logic	Sanzhar	Developing a winner selection algorithm. Implementing the "Event -> Reward" connection.
Week 9	User CRUD & Wallet	Nurali	Profile endpoints: top up balance, change settings, view history. (CRUD is mandatory!).
Week 10	Case/Event Management	Sanzhar	Streamer admin panel: creating and editing giveaway conditions.
Week 10	Frontend Integration & Testing	Nurali	Layout of HTML templates (html/template) and display of data from the database.

4. Github repository:

https://github.com/2006michigun2006-hub/cs2-livedrop