An abstract graphic on the left side of the slide, composed of numerous overlapping triangles in various shades of blue, ranging from deep navy to light sky blue. The triangles are arranged in a way that creates a sense of depth and movement, pointing towards the right.

Personal Twitch game resources engine

LU ZHU

MAY 2022

Personal Twitch game resources engine

Twitch provides *streams, videos, and clips* of games.

Twitch allows to follow gamers/creators but cannot save favorites.

Provide a list of the top 20 games by name

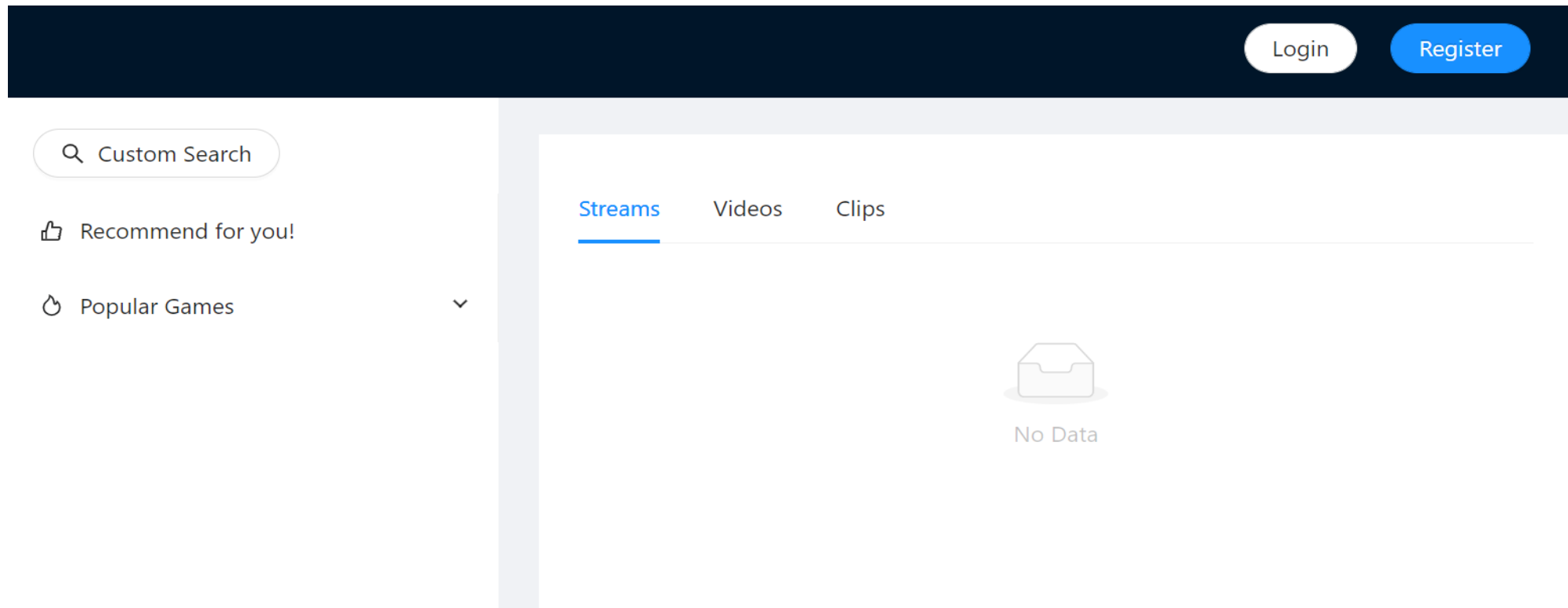
Provide a search engine for games by name

Resources (streams, videos, clips) of specific games can be displayed with preview and URL

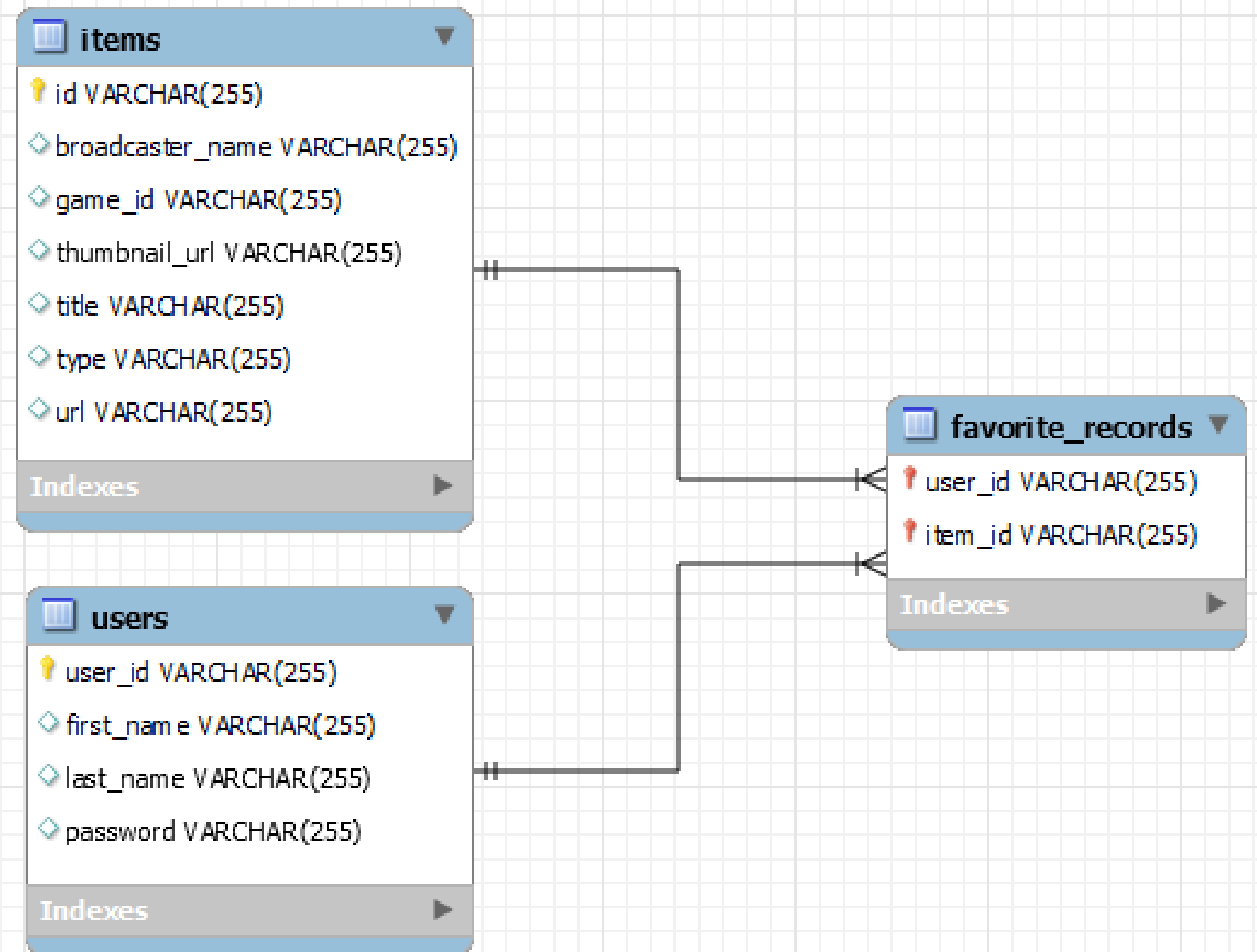
Custom favorite items can be saved if registered and logged in

Recommendations can be provided based on customer favorites

React single-page-app



Database



Database

```
@Entity
@Table(name = "Users")
@JsonIgnoreProperties(ignoreUnknown = true)
@JsonInclude(JsonInclude.Include.NON_NULL)
public class User {
    private static final long serialVersionUID = 1L;

    @Id
    @Column(name = "user_id")
    @JsonProperty("user_id")
    private String userId;

    @JsonProperty("password")
    private String password;

    @Column(name = "first_name")
    @JsonProperty("first_name")
    private String firstName;

    @Column(name = "last_name")
    @JsonProperty("last_name")
    private String lastName;

    @ManyToMany(cascade = CascadeType.ALL, fetch = FetchType.EAGER)
    @JoinTable(name = "Favorite_records",
        joinColumns = { @JoinColumn(name = "user_id") },
        inverseJoinColumns = { @JoinColumn(name = "item_id") })
    Set<Item> itemSet = new HashSet<>();
}
```

```
@Entity
@Table(name = "items")
@JsonIgnoreProperties(ignoreUnknown = true)
@JsonInclude(JsonInclude.Include.NON_NULL)
public class Item implements Serializable {
    @Id
    @JsonProperty("id")
    private String id;

    @JsonProperty("title")
    private String title;

    @JsonProperty("url")
    private String url;

    @Column(name = "thumbnail_url")
    @JsonProperty("thumbnail_url")
    private String thumbnailUrl;

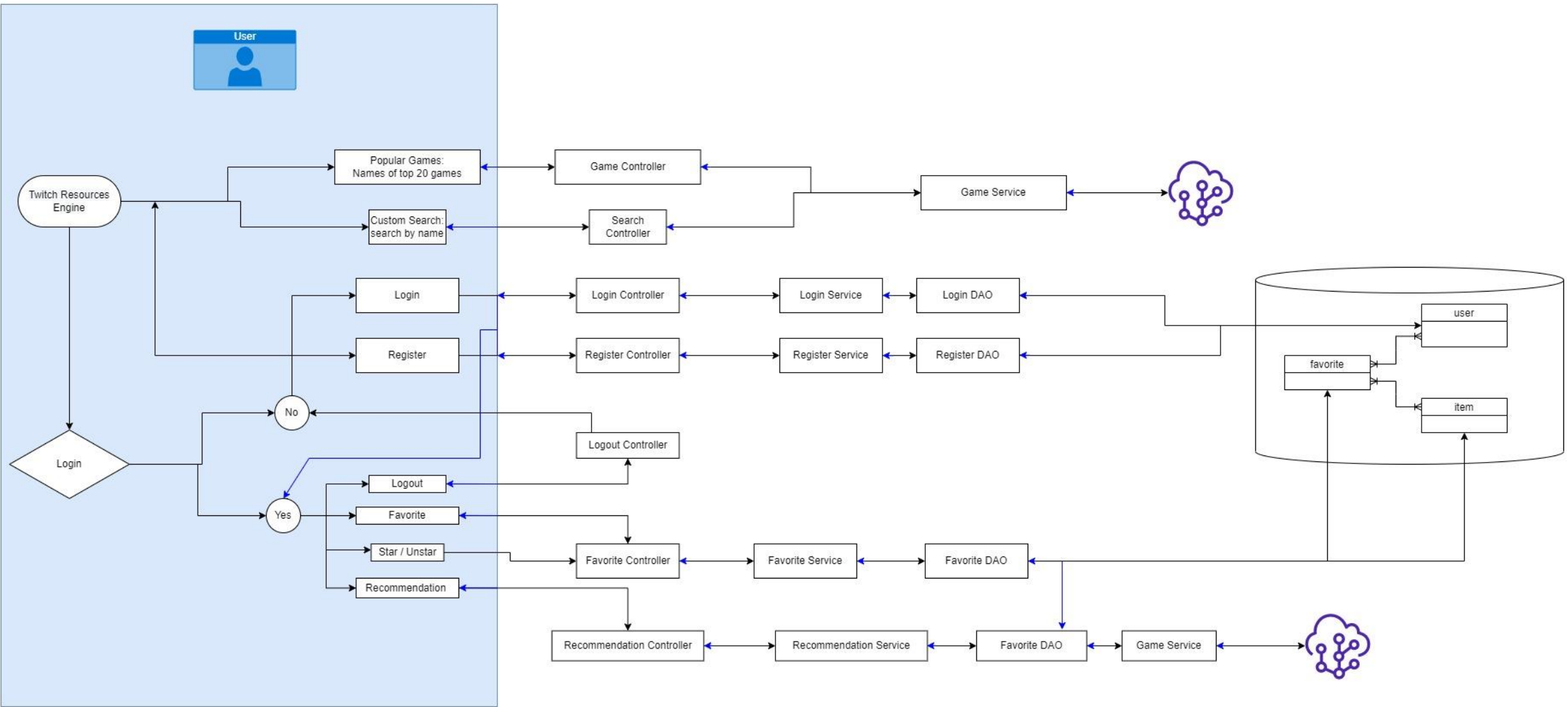
    @Column(name = "broadcaster_name")
    @JsonProperty("broadcaster_name")
    @JsonAlias({ "user_name" })
    private String broadcasterName;

    @Column(name = "game_id")
    @JsonProperty("game_id")
    private String gameId;

    @Enumerated(value = EnumType.STRING)
    @JsonProperty("item_type")
    private ItemType type;

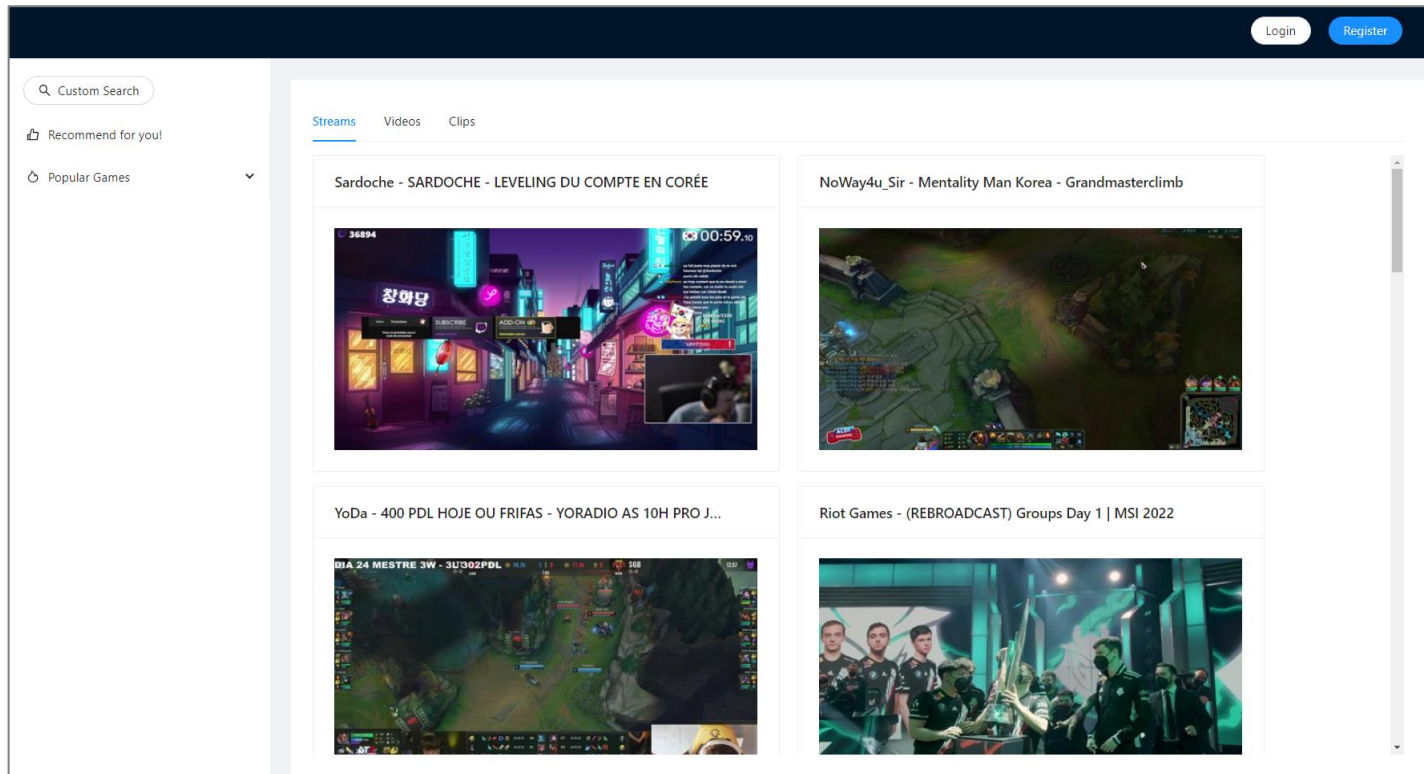
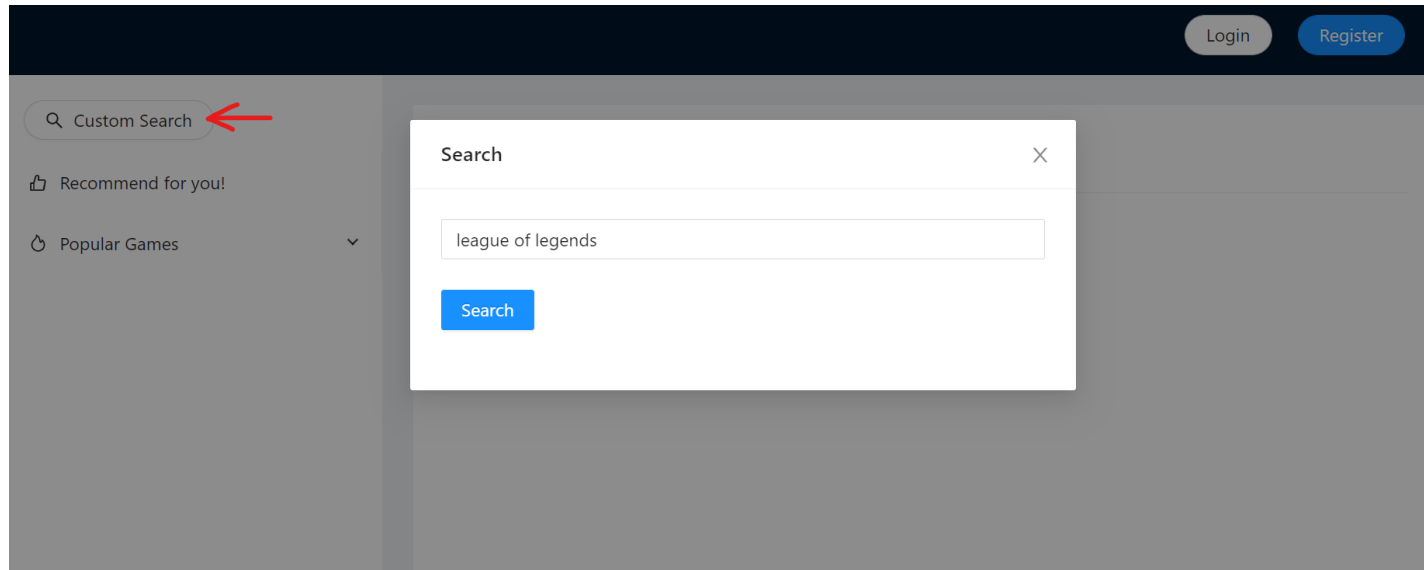
    @JsonIgnore
    @ManyToMany(mappedBy = "itemSet")
    private Set<User> users = new HashSet<>();
}
```

```
@Repository
public class FavoriteDao {
    @Autowired
    private SessionFactory sessionFactory;
    // Insert a favorite record to the database
    public void setFavoriteItem(String userId, Item item) {
        Session session = null;
        try {
            session = sessionFactory.openSession();
            User user = session.get(User.class, userId);
            user.getItemSet().add(item);
            session.beginTransaction();
            session.save(user);
            session.getTransaction().commit();
        } catch (Exception ex) {
            ex.printStackTrace();
            session.getTransaction().rollback();
        } finally {
            if (session != null) session.close();
        }
    }
}
```

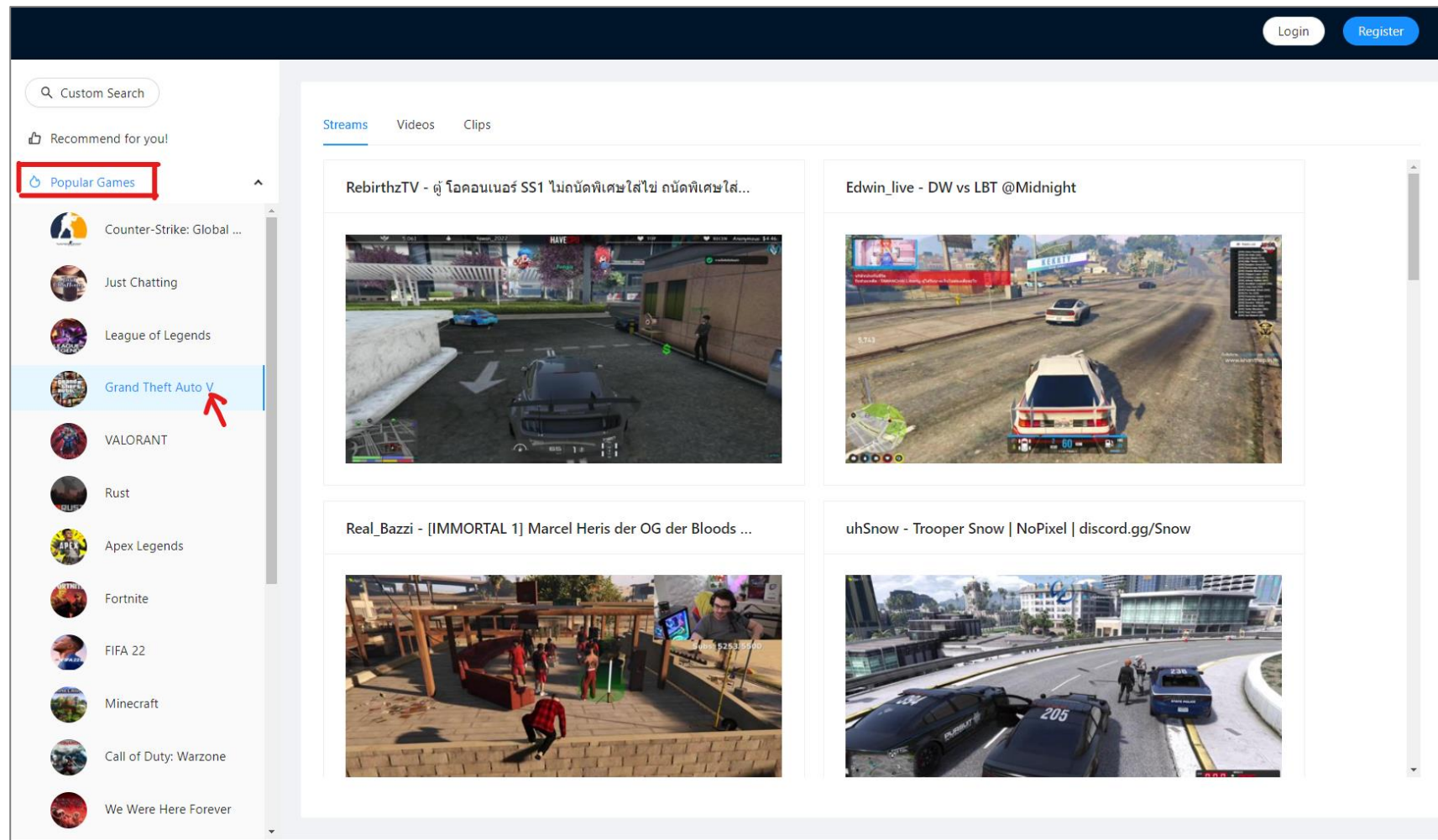


Custom Search

- Search by name ↔
Game Controller ↔
Game Service ↔
Twitch Resources search



Popular Games



- Popular Games ↔ Game Controller ↔ Game Service ↔ Twitch Resources search

Game Controller & Service

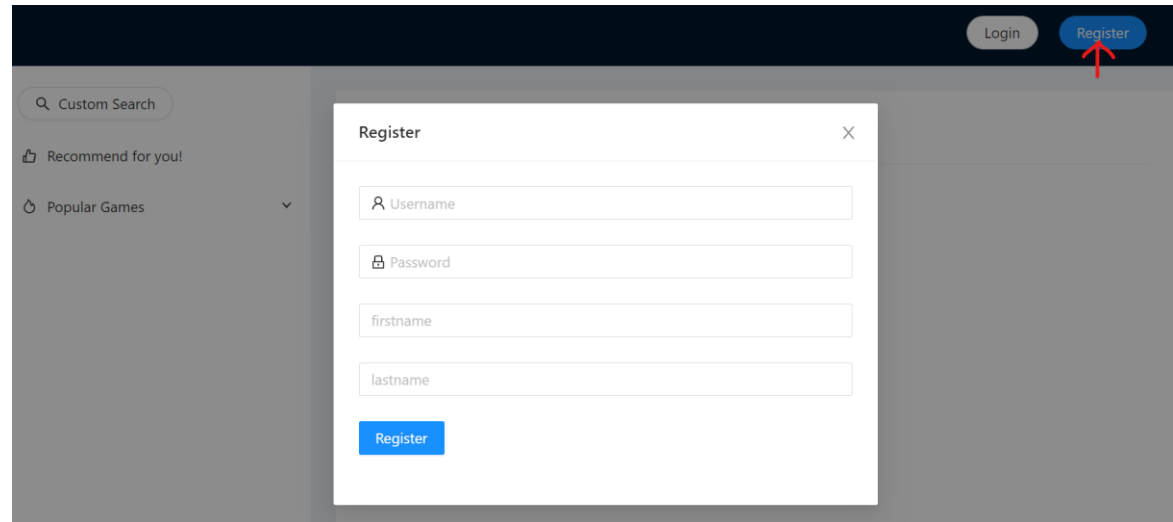
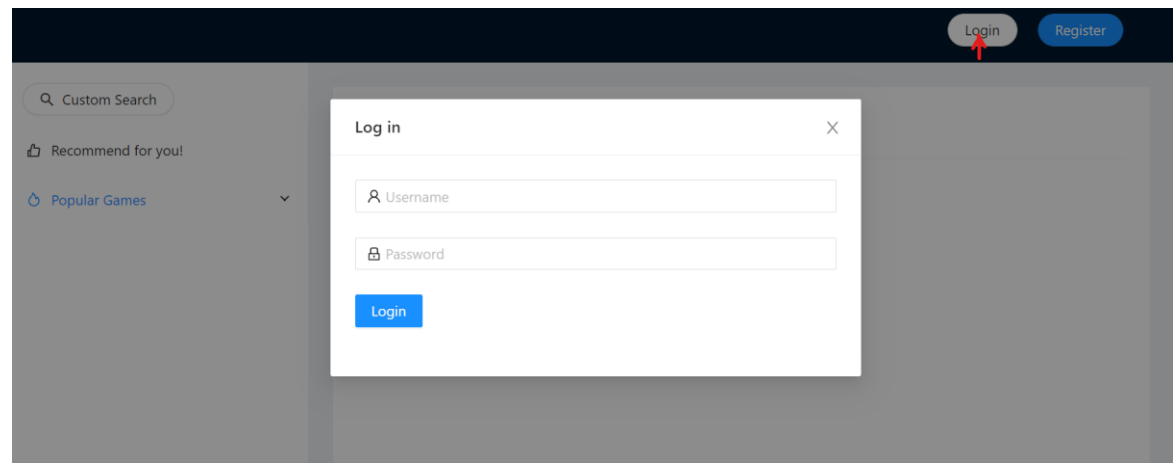
```
@Controller
public class GameController {
    @Autowired
    private GameService gameService;

    public GameController (GameService gameService) { this.gameService = gameService; }

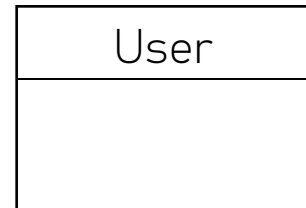
    @RequestMapping(value = "/game", method = RequestMethod.GET)
    public void getGame(@RequestParam(value = "game_name", required = false) String gameName,
                        HttpServletResponse response) throws IOException, ServletException {
        response.setContentType("application/json;charset=UTF-8");
        try {
            // Return the dedicated game information if gameName is provided in the request URL,
            // otherwise return the top x games.
            if (gameName != null) {
                response.getWriter().print(new ObjectMapper().writeValueAsString(gameService.searchGame(gameName)));
            } else {
                response.getWriter().print(new ObjectMapper().writeValueAsString(gameService.topGames( limit: 0)));
            }
        } catch (TwitchException e) {
            throw new ServletException(e);
        }
    }
}
```

- Popular Games ↔ Game Controller ↔ Game Service ↔ Twitch Resources search

Login Register



- Login
- Register



Logged in

The screenshot shows the Twitch website interface. At the top, a dark blue navigation bar contains a "My Favorites" button (star icon) and a "Logout" button. Below the navigation bar, there is a search bar labeled "Custom Search" and a "Recommend for you!" section. The main content area is divided into two columns. The left column shows a list of popular games, including "Just Chatting", "Counter-Strike: Global ...", "Grand Theft Auto V", "Apex Legends", "League of Legends", "VALORANT", "Fortnite", "Minecraft", "Call of Duty: Warzone", "FIFA 22", "Slots", and "Rust". The right column displays a grid of stream thumbnails. Each thumbnail has a star icon in the top left corner, indicating whether it is a favorite. Red arrows point to the star icons, with labels: "Star - added to Favorites" and "Unstar - not added / removed".

My Favorites Logout

Custom Search

Recommend for you!

Popular Games

Just Chatting

Counter-Strike: Global ...

Grand Theft Auto V

Apex Legends

League of Legends

VALORANT

Fortnite

Minecraft

Call of Duty: Warzone

FIFA 22

Slots

Rust

Streams Videos Clips

Star - added to Favorites

Unstar - not added / removed

TSMI ImperialHal - SEASON 13 WITH TSM DEBUT | !aimlab !...

11121immy - NEW SEASON. NEW GRIND. NEW ME. // NEW !...

AIMLAB

KOVAK'S

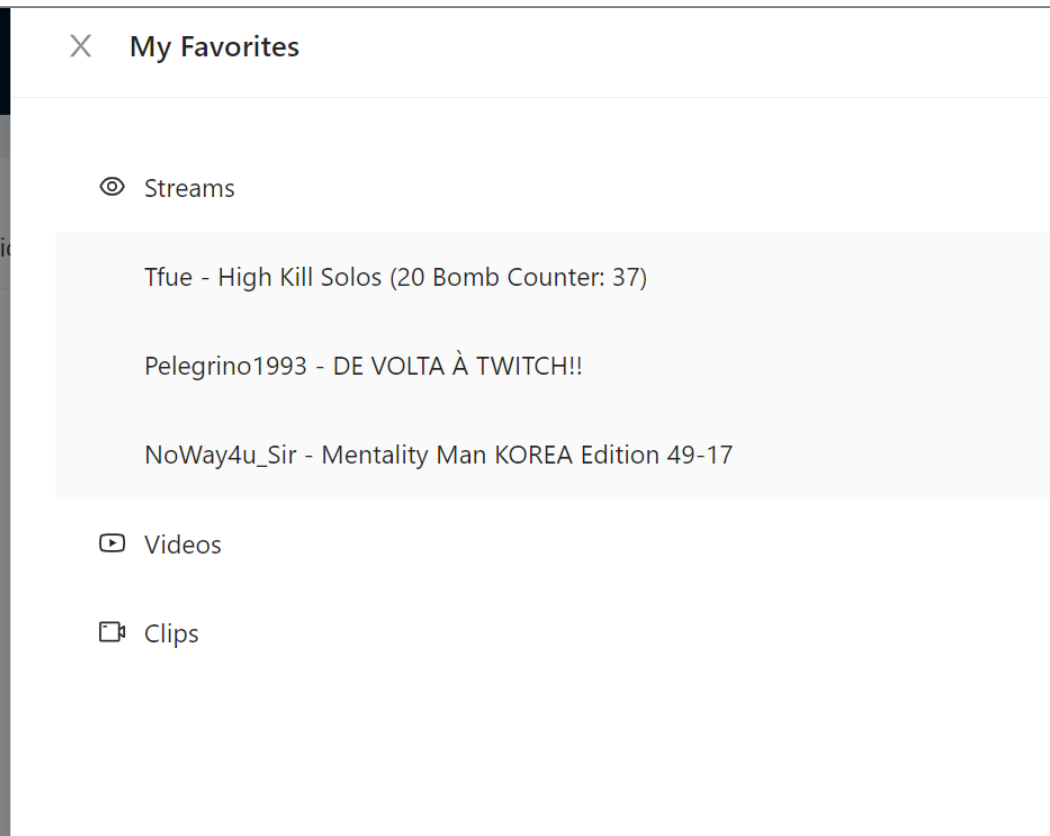
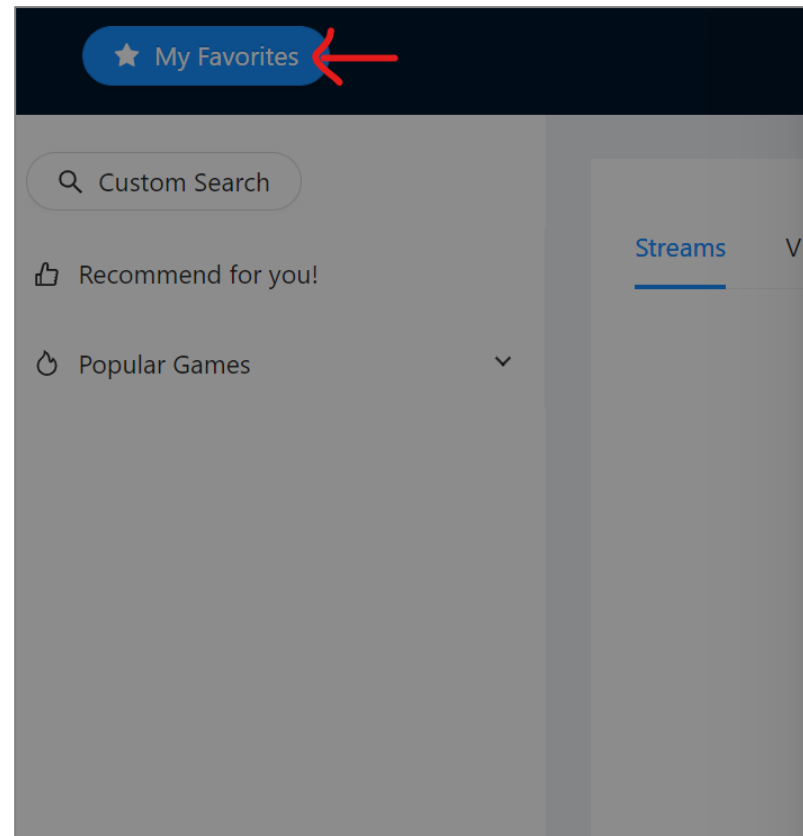
4507

00:45:21

NICKMERC'S - Apex Pro Plays New Apex Season | Code: MFA...

加藤純一です - APEX鍛える 4

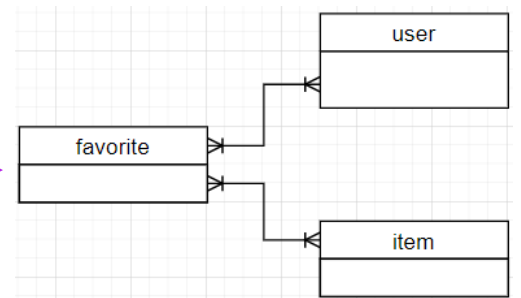
Favorites



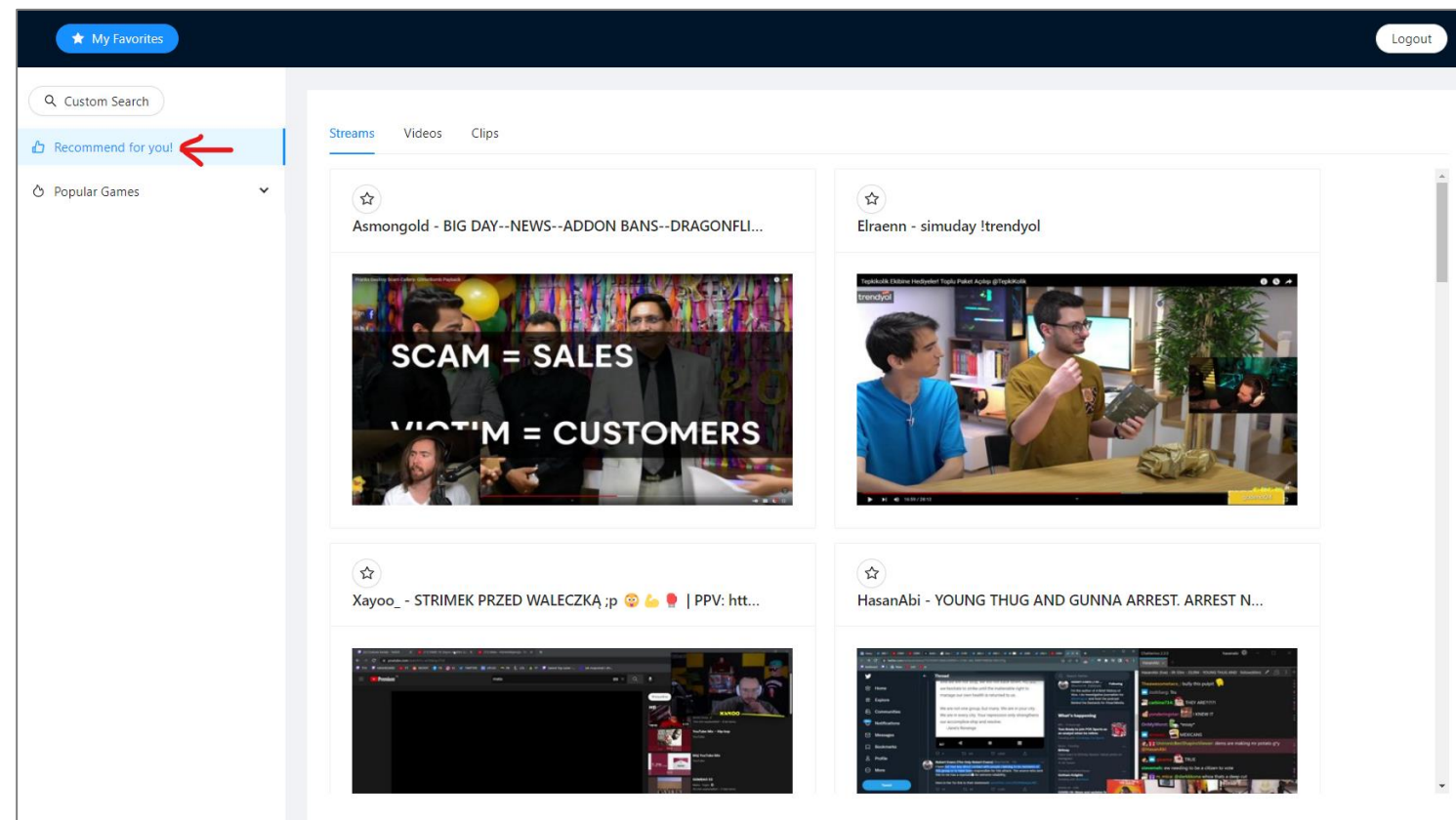
- Star / Unstar
- Favorites



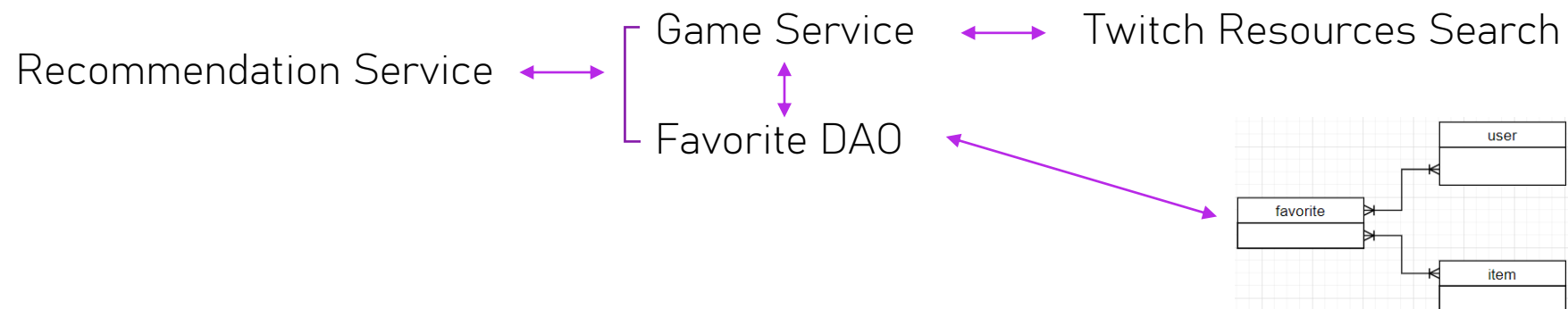
↔ Favorite Controller ↔ Favorite Service ↔ Favorite DAO ↔



Recommendation



- Recommended for you ! ↔ Recommendation Controller ↔



Recommendation

```
@Controller
public class RecommendationController {

    @Autowired
    private RecommendationService recommendationService;

    @RequestMapping(value = "/recommendation", method = RequestMethod.GET)
    @ResponseBody
    public Map<String, List<Item>> recommendation(HttpServletRequest request) throws RecommendationException {
        HttpSession session = request.getSession(false);
        Map<String, List<Item>> itemMap;
        try {
            if (session == null) {
                itemMap = recommendationService.recommendItemsByDefault();
            } else {
                String userId = (String) request.getSession().getAttribute("userId");
                itemMap = recommendationService.recommendItemsByUser(userId);
            }
        } catch (RecommendationException e) {
            throw new ServletException(e);
        }
        return itemMap;
    }
}
```

```
@Service
public class RecommendationService {

    private static final int DEFAULT_GAME_LIMIT = 3;
    private static final int DEFAULT_PER_GAME_RECOMMENDATION_LIMIT = 10;
    private static final int DEFAULT_TOTAL_RECOMMENDATION_LIMIT = 20;

    @Autowired
    private GameService gameService;

    @Autowired
    private FavoriteDao favoriteDao;

    //...

    private List<Item> recommendByTopGames(ItemType type, List<Game> topGames) throws RecommendationException {
        //...
    }

    private List<Item> recommendByFavoriteHistory(
        Set<String> favoritedItemIds, List<String> favoriteGameIds, ItemType type) throws RecommendationException {
        //...
    }

    public Map<String, List<Item>> recommendItemsByUser(String userId) throws RecommendationException {
        //...
    }

    public Map<String, List<Item>> recommendItemsByDefault() throws RecommendationException {
        //...
    }
}
```