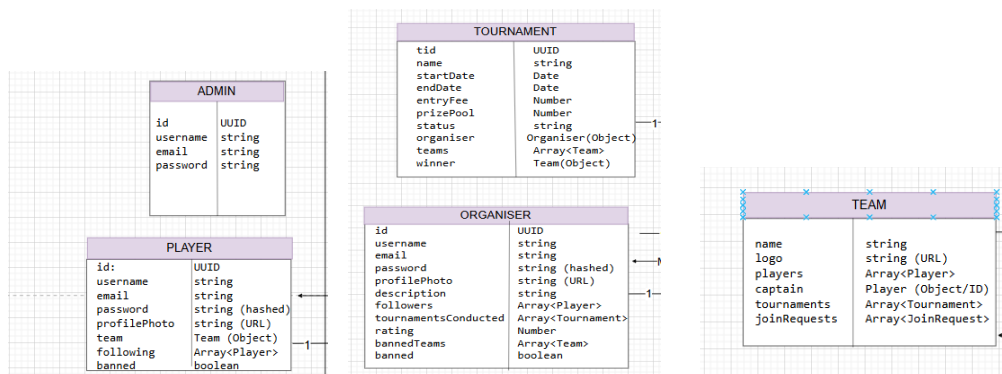# DDD DIAGRAM

GROUP:42

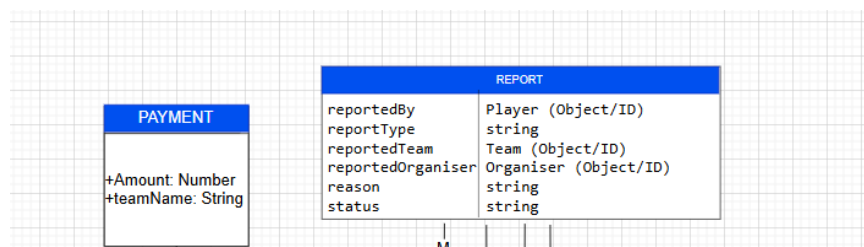**DOMAIN**: TOURNAMENT CONDUCTING WEBSITE

1. ENTITY:

   - An entity is a domain object that has a distinct identity and lifecycle
   - So in our tournament conducting website, the entities are those that have unique id
   - The entities are Admin, Player, Tournament, Organiser,   Team
   - Each of these entities again have attributes
   - Properties may change, but identity remains

TOURNAMENT	
tid	UUID
name	string
startDate	Date
endDate	Date
entryFee	Number
prizePool	Number
status	string
organiser	Organiser(Object)
teams	Array<Team>
winner	Team(Object)

ADMIN	
id	UUID
username	string
email	string
password	string

ORGANISER	
id	UUID
username	string
email	string
password	string (hashed)
profilePhoto	string (URL)
description	string
followers	Array<Player>
tournamentsConducted	Array<Tournament>
rating	Number
bannedTeams	Array<Team>
banned	boolean

TEAM	
name	string
logo	string (URL)
players	Array<Player>
captain	Player (Object/ID)
tournaments	Array<Tournament>
joinRequests	Array<JoinRequest>

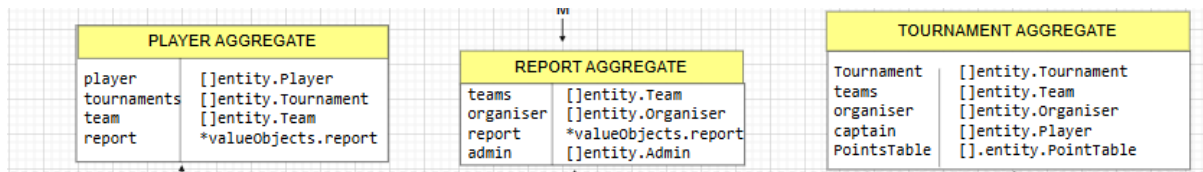| PLAYER	
id:	UUID
username	string
email	string
password	string (hashed)
profilePhoto	string (URL)
team	Team (Object)
following	Array<Player>
banned	boolean

2. VALUE OBJECTS :

   - A value object is a type of domain object that represents a value that is conceptually unchangeable.
   - These lack unique ID and also cannot changes after creation
   - These are just for short time
   - The value objects we have taken are Report and Payment

REPORT	
reportedBy	Player (Object/ID)
reportType	string
reportedTeam	Team (Object/ID)
reportedOrganiser	Organiser (Object/ID)
reason	string
status	string

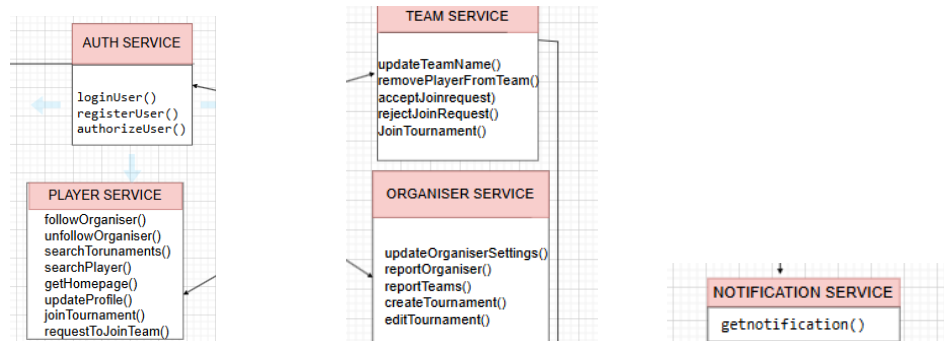| PAYMENT |
| --- |
| +Amount: Number |
| +teamName: String |

3. AGGREGATES:
   - An aggregate is a cluster of domain objects that are treated as a single unit for the purpose of data consistency.
   - The Aggregate Root is the main entity that controls access to the other entities inside the aggregate.
   - The aggregate we took are :
     - 1.Player Aggregate : Root-Player Enity
     - 2.Report Aggreagate: Root-Admin Entity
     - 3.Tournament Aggreagate: Root-Organiser Enitity

PLAYER AGGREGATE	
player	[]entity.Player
tournaments	[]entity.Tournament
team	[]entity.Team
report	*valueObjects.report

REPORT AGGREGATE	
teams	[]entity.Team
organiser	[]entity.Organiser
report	*valueObjects.report
admin	[]entity.Admin

| TOURNAMENT AGGREGATE	
Tournament	[]entity.Tournament
teams	[]entity.Team
organiser	[]entity.Organiser
captain	[]entity.Player
PointsTable	[].entity.PointTable

4. SERVICES:
   - Operation that does not naturally belong to any specific entity or value object.
   - Contains business logic
   1. Admin Service: authenticate, register, login
   2. Team Service: updating Team name, accepting Requests, Join tournaments
   3. Player Service: follow Organiser, JoinTeams, searchPlayers,updateProfile
   4. OrganiserService: reportTeams, conductTournaments
   5. NotificationService: send notifications

AUTH SERVICE
loginUser()
registerUser()
authorizeUser()

TEAM SERVICE
updateTeamName()
removePlayerFromTeam()
acceptJoinrequest)
rejectJoinRequest()
JoinTournament()

PLAYER SERVICE
followOrganiser()
unfollowOrganiser()
searchTorunaments()
searchPlayer()
getHomepage()
updateProfile()
joinTournament()
requestToJoinTeam()

ORGANISER SERVICE
updateOrganiserSettings()
reportOrganiser()
reportTeams()
createTournament()
editTournament()

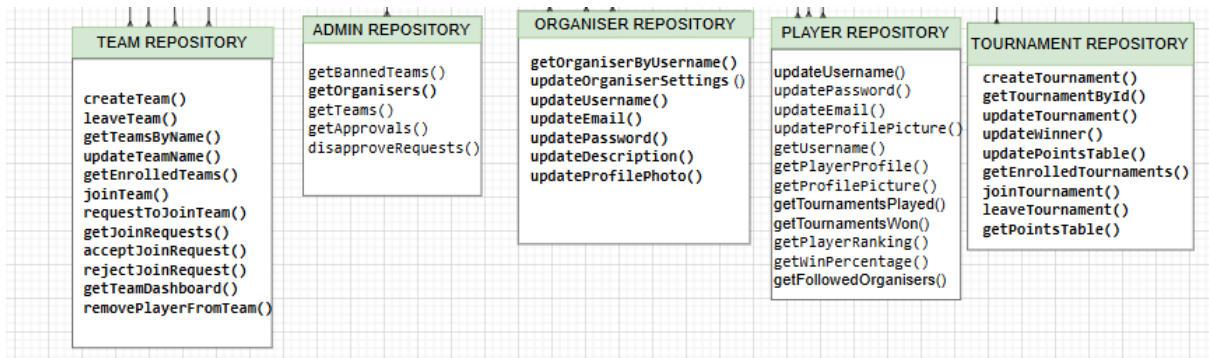| NOTIFICATION SERVICE |
|---|
| getnotification() |

5. HUB :
   - Manages domain events
   - Notifies different parts of the system when an important action occurs.
   - Helps in connection between different Services

HUB
Admin Service
Notification Service
Team Service
Player Service
Organiser Service
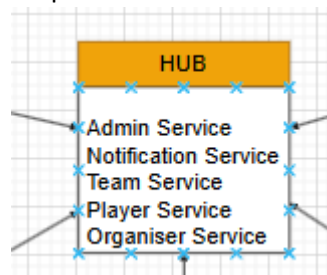
6. REPOSITORIES:

- o Repositories separate data access logic from the domain model.
- o They provide a consistent interface for querying and storing domain objects
- o It provides an abstraction layer, so the rest of the system does not deal directly with the database.
- o We have Team Repository, Admin Repository, Player Repository, Organiser Repository, Tournament Repository

TEAM REPOSITORY

```
createTeam()
leaveTeam()
getTeamsByName()
updateTeamName()
getEnrolledTeams()
joinTeam()
requestToJoinTeam()
getJoinRequests()
acceptJoinRequest()
rejectJoinRequest()
getTeamDashboard()
removePlayerFromTeam()
```

ADMIN REPOSITORY

```
getBannedTeams()
getOrganisers()
getTeams()
getApprovals()
disapproveRequests()
```

ORGANISER REPOSITORY

```
getOrganiserByUsername()
updateOrganiserSettings ()
updateUsername()
updateEmail()
updatePassword()
updateDescription()
updateProfilePhoto()
```

PLAYER REPOSITORY

```
updateUsername()
updatePassword()
updateEmail()
updateProfilePicture()
getUsername()
getPlayerProfile()
getProfilePicture()
getTournamentsPlayed()
getTournamentsWon()
getPlayerRanking()
getWinPercentage()
getFollowedOrganisers()
```

TOURNAMENT REPOSITORY

```
createTournament()
getTournamentById()
updateTournament()
updateWinner()
updatePointsTable()
getEnrolledTournaments()
joinTournament()
leaveTournament()
getPointsTable()
```

FINAL DDD: