# OneStat

Team 10: Zach Fulton, Nolan McDermott, Talon Smith

# OneStat summary:
# Baseball for All

Player Comparisons, Stat leaderboards,
Statistical definitions, WARLeagues, etc



# wRC+

Name: Weighted Runs Created Plus
Scale: 100 is average
        150 is 50% above average
        50 is 50% below average
Last Year's Leader:
        Juan Soto - 212 (112% above average)
Example: All Soto's 2020 Home Runs



Our take: wRC+ is a great and easy to understand stat for rating offensive performances. Similar to OPS+ but used wOBA so it better represents walks vs singles.
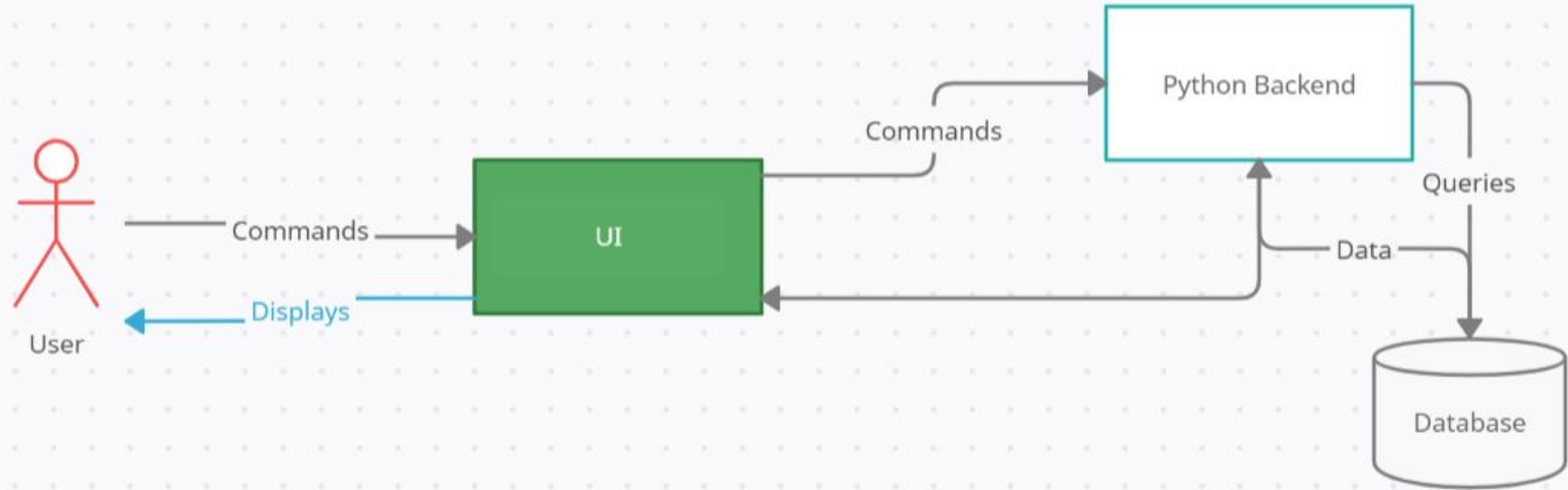
# System overview

Three-tier architecture

- Client server as the UI Tier
- Python Backend that organizes and communicates data as the middle tier
- OneStat database as the storage tier


- These three tiers will interact in order to provide the general users on the UI Tier with a quick and easy experience

Actors:

- Client server will be used by general users including WarLeague members and owners
- All three tiers will be used by the system managers in order to maintain and update the program and databases

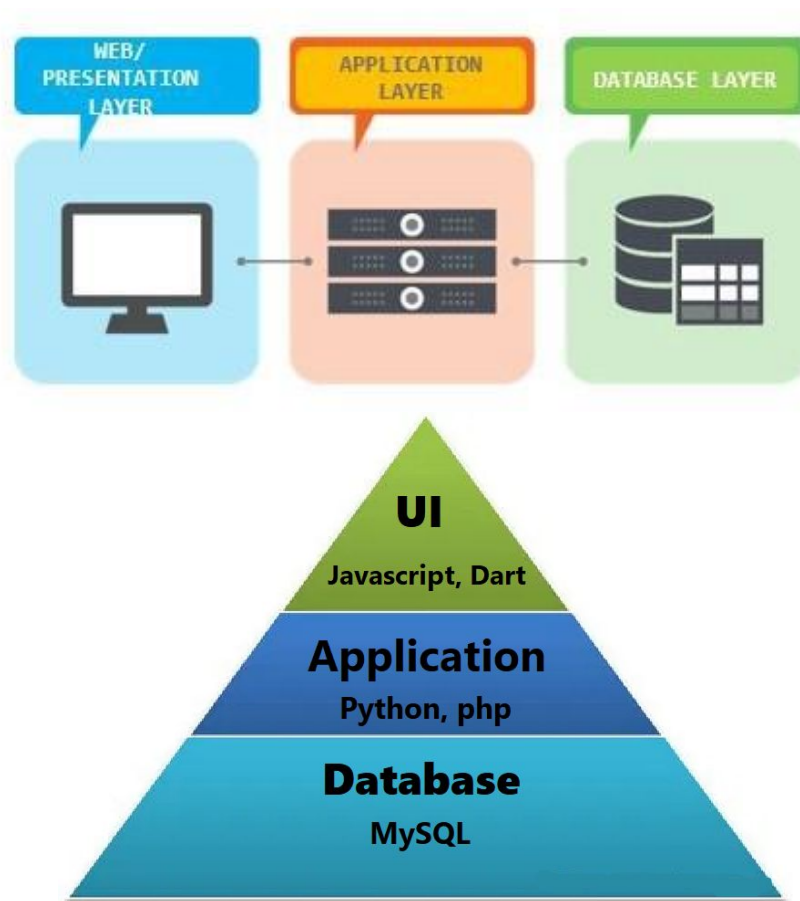# System's components w/ system diagram

# Actor Identification

- General User
  - Accesses OneStat from their homes
  - Very limited access to the software

- WarLeague Owner
  - Can invite users to their league and manage it
  - Has access to edit the league and manage the league and its players

- WarLeague Member
  - Can join a league
  - Chooses players to use and can compete
  - Has less access than the league owner

- System Manager
  - Verifies server information
  - Updates definitions
  - Assists general users
  - Significantly more access than general users

# Architectural Style
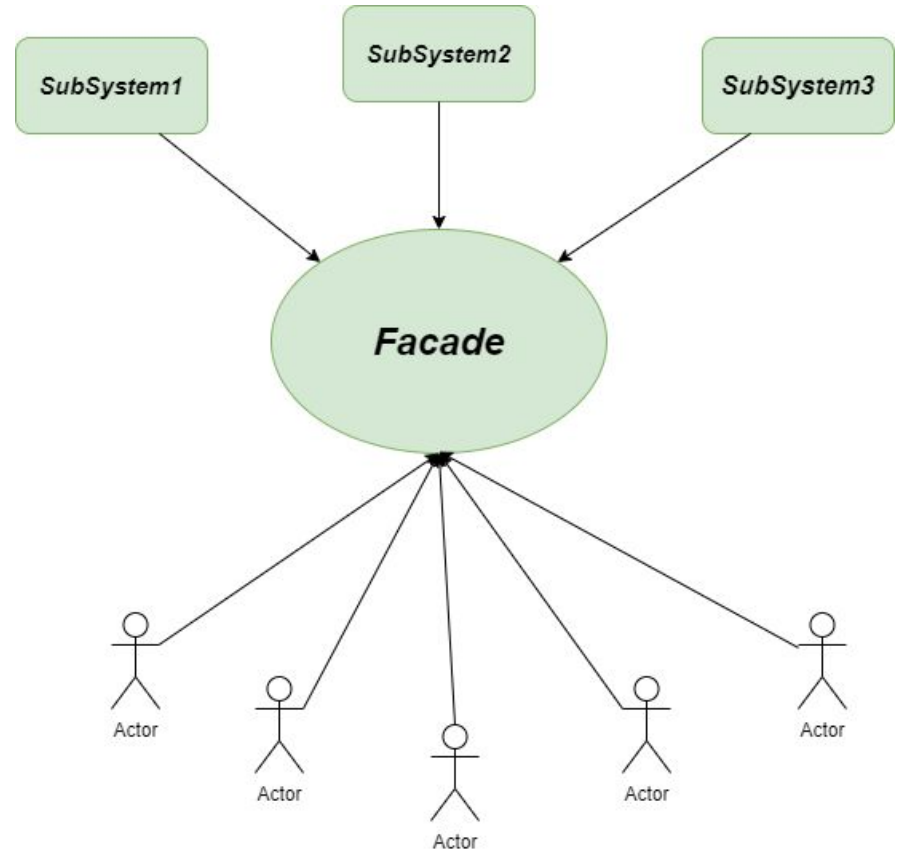
**Three Tier Structure**

- OneStat Database
  - Stores
    - Definitions
    - Leaderboards
    - Player Data
- Python based middle tier
  - Retrieves requests from user
  - Implement processes to retrieve data from database and send to user
- Client UI
  - Displays requested information neatly with OneStat's user interface

# Design patterns

**Façade**

- To allow for the complex organization of the system to be organized and stable when an actor is using it.
- Since there are multiple types of users it can distinguish be them and provide different levels of access for users
- Creates a false image that the system is simple when in reality it is very complex
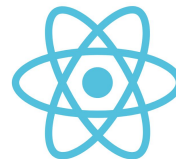
# Frameworks

- User framework
  - HTML, CSS, JavaScript for desktop
    - React framework for Javascript
  - Dart for mobile
- Middle tier
  - Python to retrieve data and inputs from the other tiers and process requests
- Database
  - MySQL to organize and store data securely
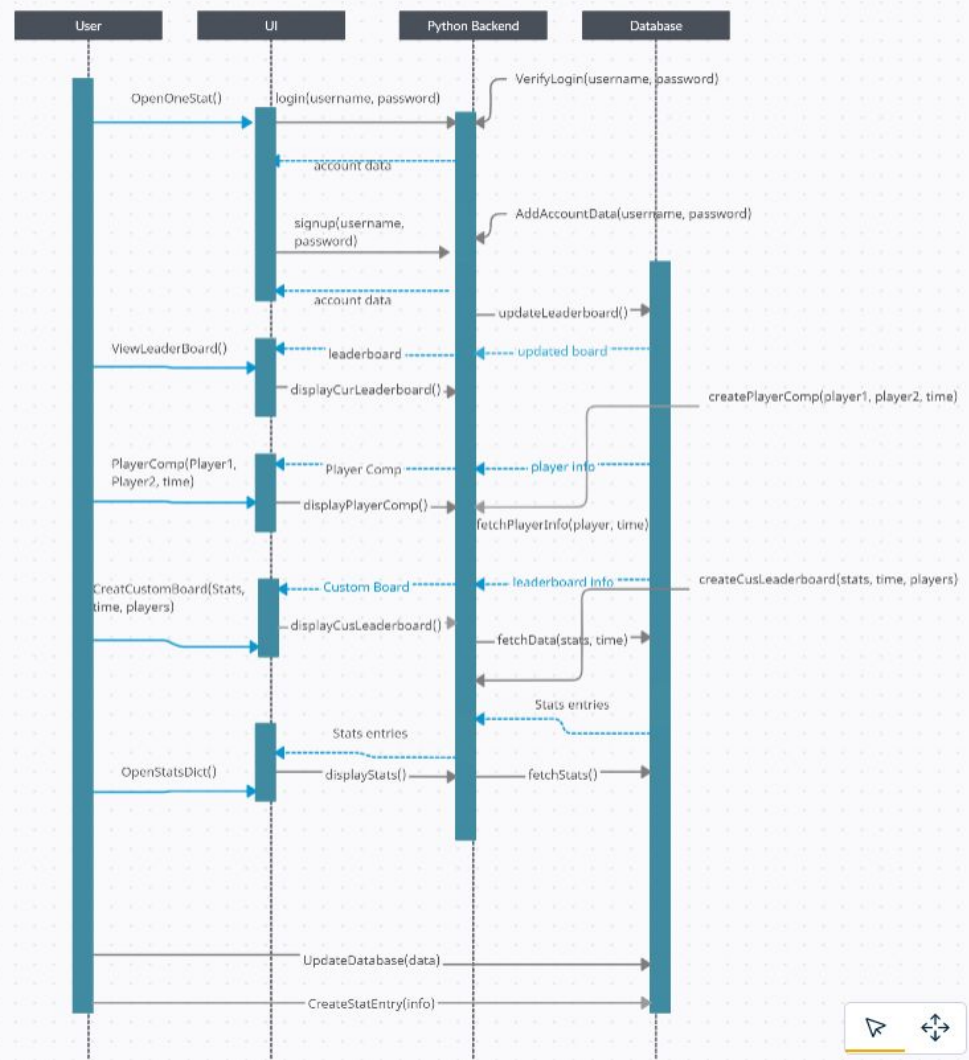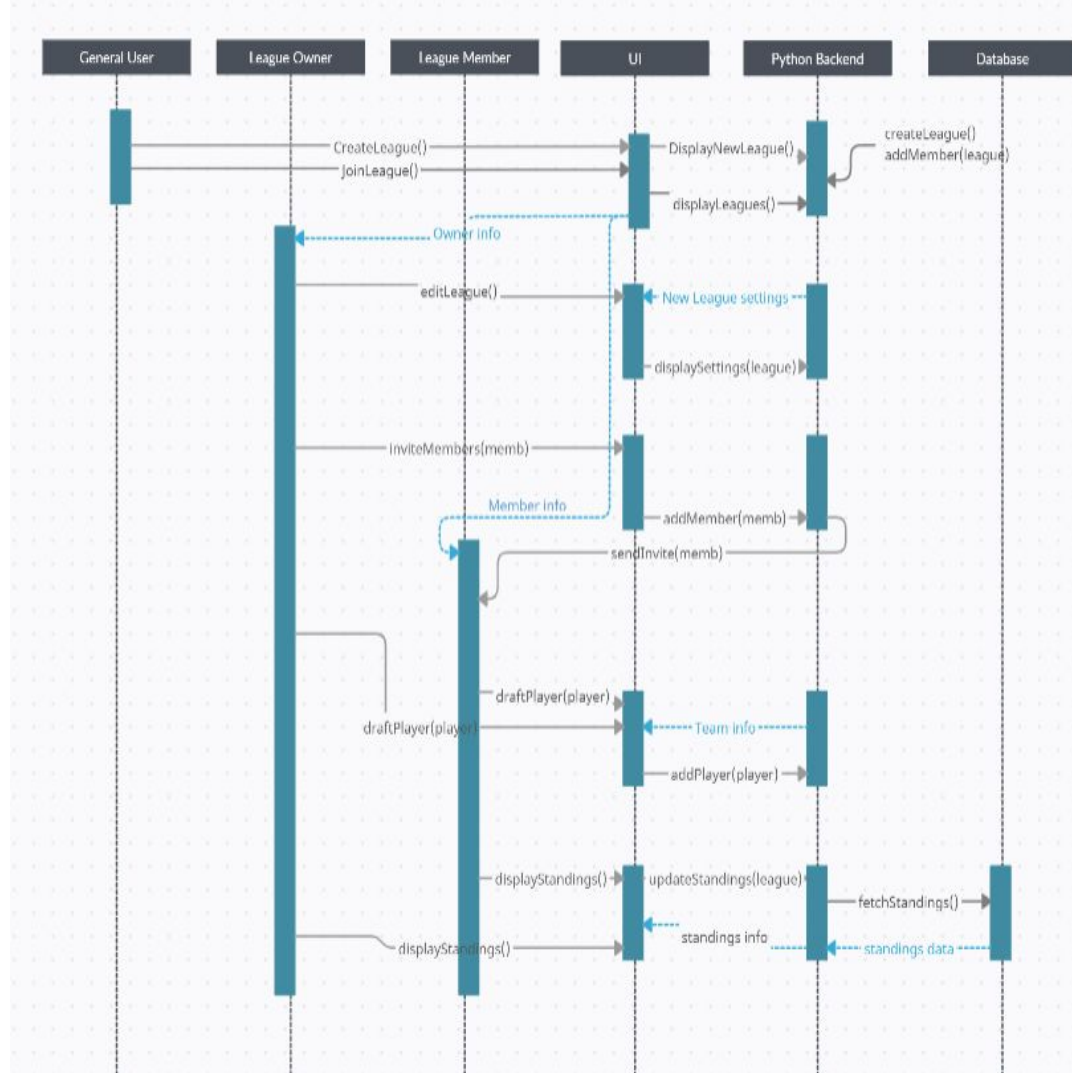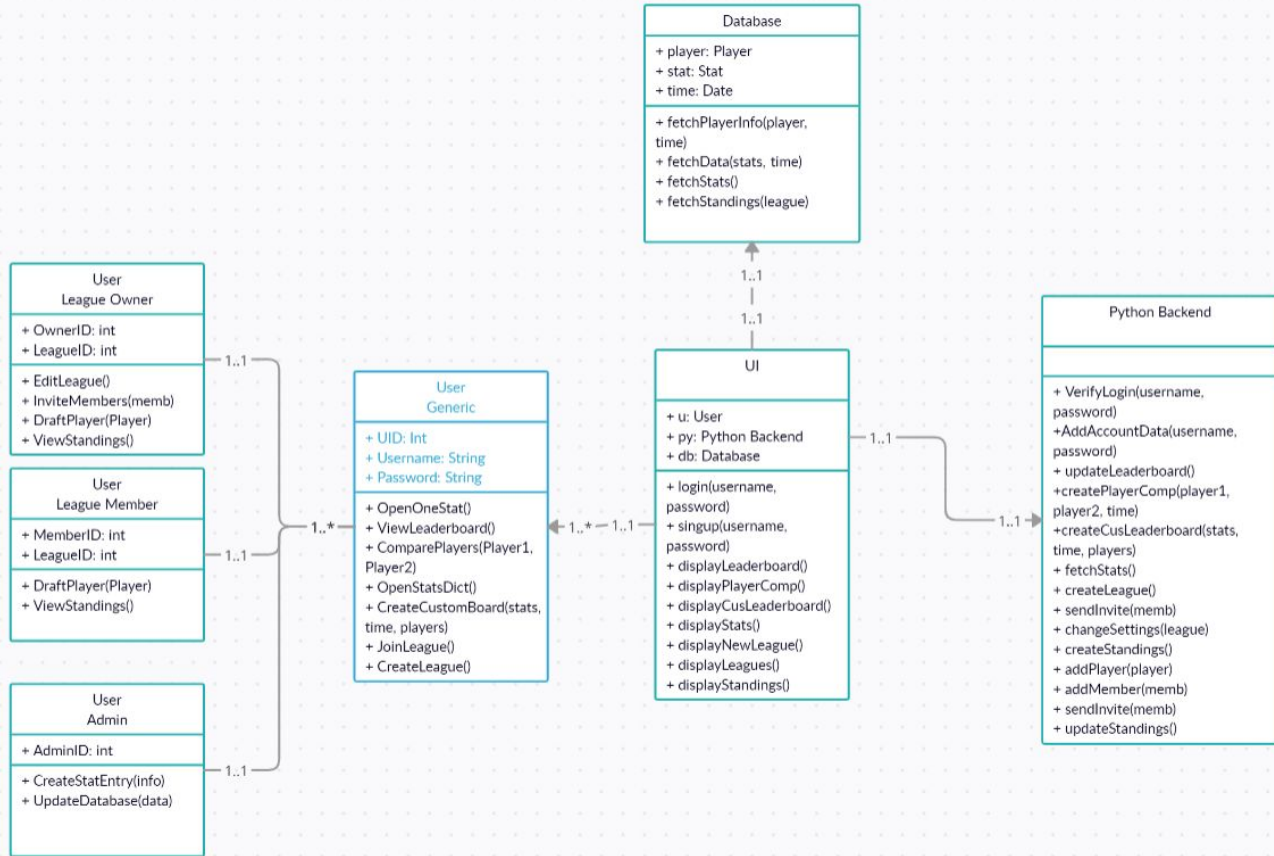  - Allows quick access and editing

# Sequence Diagram

OneStat

# Sequence Diagram

WARLeagues

# Class Diagram



**Database**

+ player: Player
+ stat: Stat
+ time: Date

+ fetchPlayerInfo(player, time)
+ fetchData(stats, time)
+ fetchStats()
+ fetchStandings(league)

**User**
**League Owner**

+ OwnerID: int
+ LeagueID: int

+ EditLeague()
+ InviteMembers(memb)
+ DraftPlayer(Player)
+ ViewStandings()

**User**
**League Member**

+ MemberID: int
+ LeagueID: int

+ DraftPlayer(Player)
+ ViewStandings()

**User**
**Admin**

+ AdminID: int

+ CreateStatEntry(info)
+ UpdateDatabase(data)

**User**
**Generic**

+ UID: Int
+ Username: String
+ Password: String

+ OpenOneStat()
+ ViewLeaderboard()
+ ComparePlayers(Player1, Player2)
+ OpenStatsDict()
+ CreateCustomBoard(stats, time, players)
+ JoinLeague()
+ CreateLeague()

**UI**

+ u: User
+ py: Python Backend
+ db: Database

+ login(username, password)
+ singup(username, password)
+ displayLeaderboard()
+ displayPlayerComp()
+ displayCusLeaderboard()
+ displayStats()
+ displayNewLeague()
+ displayLeagues()
+ displayStandings()

**Python Backend**

+ VerifyLogin(username, password)
+ AddAccountData(username, password)
+ updateLeaderboard()
+ createPlayerComp(player1, player2, time)
+ createCusLeaderboard(stats, time, players)
+ fetchStats()
+ createLeague()
+ sendInvite(memb)
+ changeSettings(league)
+ createStandings()
+ addPlayer(player)
+ addMember(memb)
+ sendInvite(memb)
+ updateStandings()

1..1
1..1
1..1
1..1
1..*
1..1
1..*
1..1
1..1
1..1

# THANK YOU
## – OneStat Team

GitHub Link:
https://github.com/NolanMcD/CSC431