CSC 431

WARLeagues

System Architecture Specification (SAS)

Team 10

Nolan McDermott	Scrum Master, Concept Manager
Zach Fulton	Team member, Implementation Manager
Talon Smith	Team Member, Design Manager

Version History

Version	Date	Author(s)	Change Comments
1.0	3/30/21	Nolan	First Draft
2.0	4/1/21	Nolan & Talon	Diagrams and info
2.1	4/13/21	Zach	Tweaks to design
3.0	4/29/21	Talon	Diagram changes and adding class summaries

Table of Contents

Table of Tables

Table of Figures

1. System Analysis

1.1 System Overview

OneStat is software that combines the information from many baseball websites to be the main platform baseball fans can use. The software will define baseball statistics, create custom leaderboards, allow for users to compare players, and provide a competitive fantasy baseball platform called WARLeages.

The architecture design that will be used is a three-tier architecture because there will be the client-server as a UI tier, the OneStat database which will be the storage tier, and a middle tier that is a python backend in order to organize the information and send to the user to and from the database.

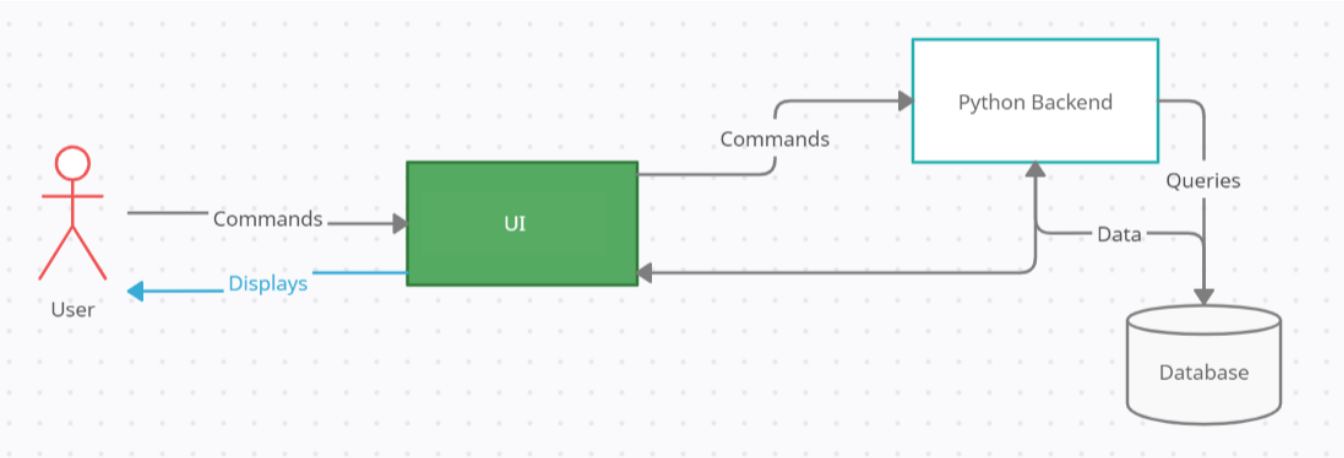
There are 4 main actors that can interact with OneStat. The first is the generic user, which accesses the software from their home and would be the average consumer. This user does not necessarily need an account unless they want to save the statistics they search for.

The next is a WarLeague owner, which initially creates a league and can invite people to their league as well as manage it. They manage the league players and makes sure their matches are running smoothly. An account needs to be created to have these abilities.

A WarLeague member also needs an account, but they are not in charge of the league and just plays within it. A War league owner is also considered a WarLeague member.

Lastly a system manager is part of the OneStat team and has the most access to the software. They can verify and modify server information, assist general users, as well as update information such as statistics and definitions.

1.2 System Diagram



1.3 Actor Identification

One actor is the general user. These are people that use the system from their homes through the first tier. These could include a user who just wants to look up statistics, a fantasy league owner, and a league member. All these roles will have different accessibility depending on if they created an account for the platform, especially if they participate in WARLeagues.

Another actor is the system manager, who verifies information on the servers, updates definitions, assists general users with technical difficulties, as well as makes sure the platform is running smoothly. They will have significantly more access than the general user, as they will be able to modify aspects in all three tiers, including data and system processes.

1.4 Design Rationale

1.4.1 Architectural Style

The architectural style that will be used is three-tier architecture, as this will allow us to develop the software in distinct groups. First will be the OneStat database that will store definition, leaderboard, and player data. Next will be the middle tier that will implement processes that will retrieve information from the database and send it back to the user's server. This last client-server will have the information neatly laid out in the OneStat website user interface.

1.4.2 Design Pattern(s)

The façade design pattern will be used to provide the general user with a clean and friendly interface despite the system being very complex. The general user does not need full functional control of the system, therefore this pattern will allow for them to only have access to features they would want to use. Additionally, the system will have many layers to it, so the general user will perceive the software as easy to use despite its complexity.

1.4.3 Framework

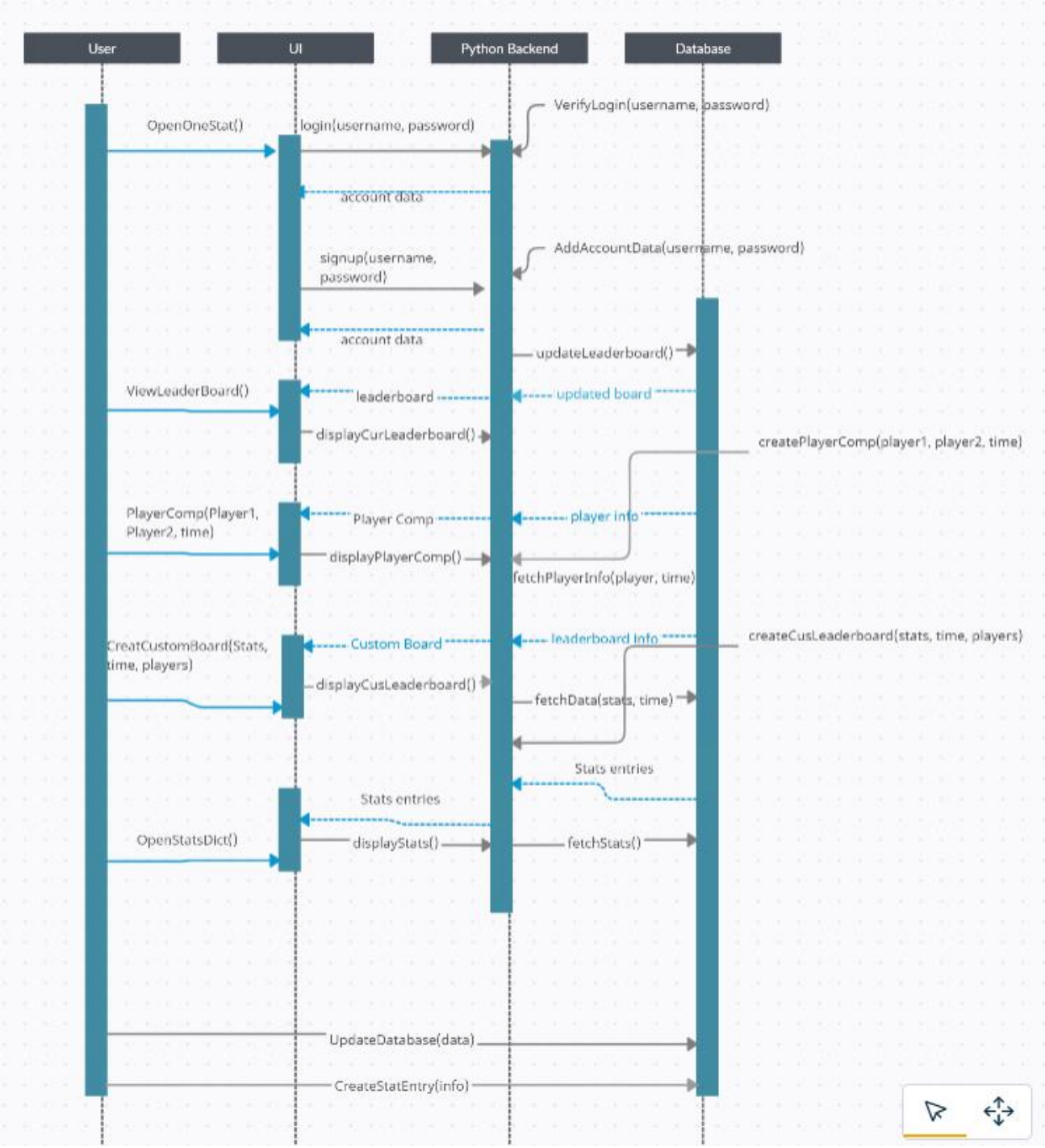
The user framework will be HTML, CSS, and JavaScript for desktop and Dart for the mobile as this will provide a simple way of presenting data to the user on multiple platforms and operating systems.

The middle tier will use python to retrieve data and inputs from the other tiers and process requests.

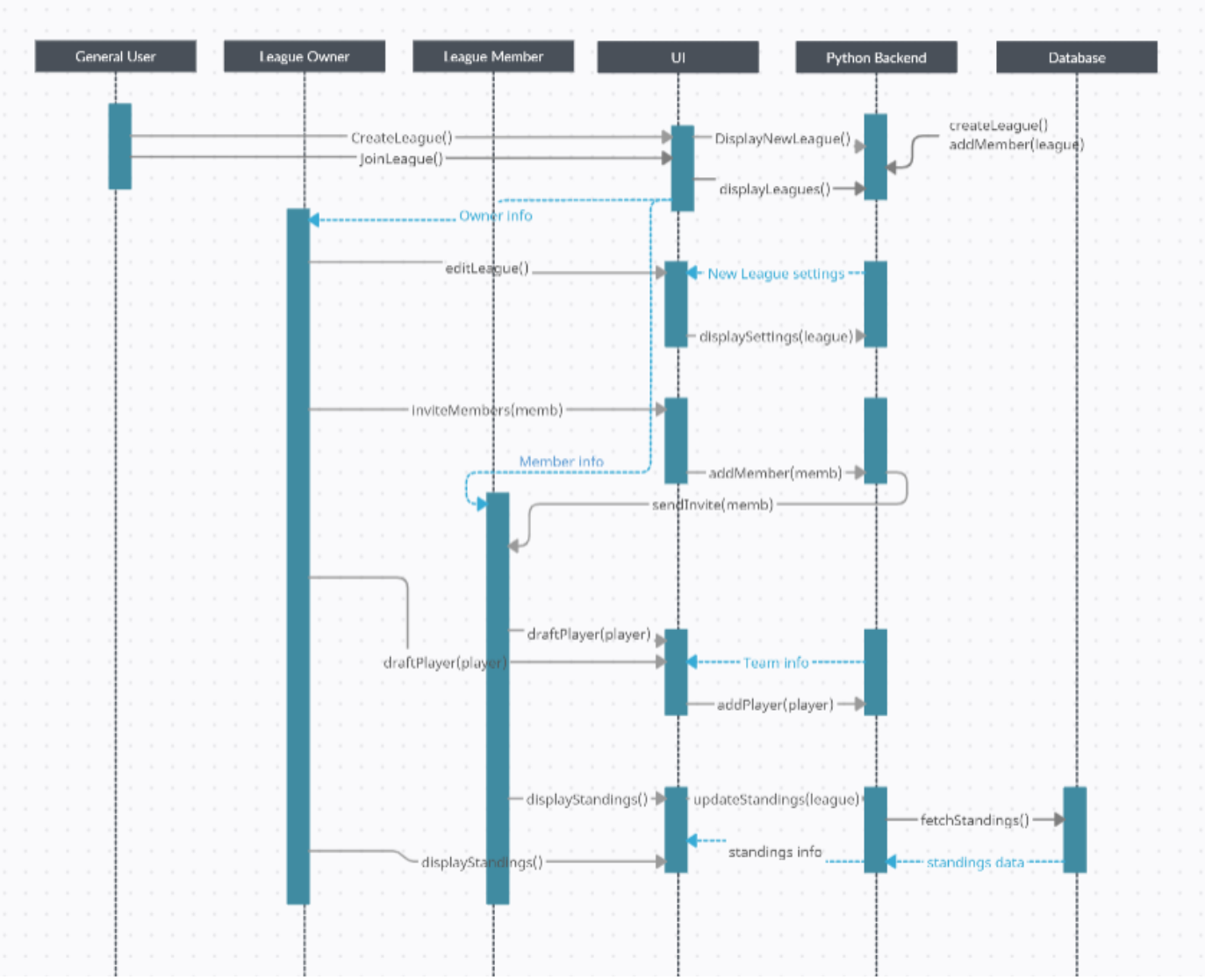
The database will be MySQL to organize and store data securely on the servers with quick retrieval and editing.

2. Functional Design

2.1 OneStat Sequence Diagram



2.2 WARLeagues Sequence Diagram



3. Structural Design