# CSC 431 – R (Spring 2025) UScore - The Sports Score & Noise Tracker System Architecture Specification (SAS)

**Team Number 4**

| Shane O’Neill | Project Manager/Software Engineer |
| --- | --- |
| Samuel Schell | Project Manager/Software Engineer |
| Elliot Wright | Project Manager/Software Engineer |
|  |  |

# Version History

| Version | Date | Author(s) | Change Comments |
| --- | --- | --- | --- |
| 1 | 3/15/25 | Shane O’Neill | Created document |
| 2 | 4/1/25 | Elliot Wright, Samuel Schell | Added requirements and architectural style |
| 3 | 4/15/25 | Shane O'Neill | Sequence Diagram |
| 4 | 4/22/25 | Samuel Schell, Shane O’Neill | Completed full draft and reviewed |

# Table of Contents

[1. System Analysis 6](#_heading=h.cc6xg5sgqv68)

5[1.1 System Overview 6](#_heading=h.xg90khwhihyu)

5[1.2 System Diagram 6](#_heading=h.g0r95h42r9do)

5[1.3 Actor Identification 6](#_heading=h.kuzozslwchgo)

5[1.4 Design Rationale 6](#_heading=h.bijhl7q10jft)

5[1.4.1 Architectural Style 6](#_heading=h.trp5e25p2d5t)

5[1.4.2 Design Pattern(s) 6](#_heading=h.72zu5obpeshv)

5[1.4.3 Framework 6](#_heading=h.usmzdm949wqt)

5[2. Functional Design 7](#_heading=h.q8tgz4wbv001)

5[2.1 Diagram Title 7](#_heading=h.ndv6y8co822i)

5[3. Structural Design 8](#_heading=h.kptw9t9dqj5b)

# Table of Tables

<Generate table here>

| Table 1 | Version History | Page 2 |
| --- | --- | --- |
| Table 2 | Actor Identification | Page 5 |
| Table 3 | System Requirements Overview | Page 6 |
| Table 4 | Functional Requirements | Page 6 |

# Table of Figures

| 3.1.1 | Use-case diagram of Live Score Updates |
| --- | --- |
| 3.1.2 | Use-case diagram of Noise Meter |
| 3.1.3 | Use-case diagram of Quick-Reaction Chat System |
| 3.1.4 | Use-case diagram of Game Recaps |
| 3.1.5 | Use-case diagram of Customizable Feeds |
| 52.1 | Sequence diagram of Live Game Interaction Flow |
| 53.1 | Class diagram |

### System Analysis

#### System Overview

*U-Score is a live sports score tracker web application that enhances fan experience through three major features:*

*- Real-time score updates*

*- Noise Meter that reflects fan excitement*

*- Live fan chat feature for interaction*

*Its architecture is designed to be scalable and lightweight, using a client-server model hosted on AWS.*

#### System Diagram

*<Insert System Diagram>*

#### Actor Identification

*- \*\*End Users\*\*: Fans who use U-Score for live scores and chat*

*- \*\*System Admins\*\*: Maintain system performance and deploy updates*

*- \*\*Third-party APIs\*\*: External providers of live sports data*

#### Design Rationale

##### Architectural Style

*U-Score uses a \*\*3-tier architecture\*\*:*

*- Presentation Tier: HTML/CSS/JavaScript frontend*

*- Logic Tier: Server-side scripts hosted on AWS*

*- Data Tier: Database or cache for storing scores and chat*

##### Design Pattern(s)

*- \*\*Observer Pattern\*\*: For pushing live data (scores and Noise Meter)*

*- \*\*Model-View-Controller (MVC)\*\*: Separates logic, UI, and data handling*

##### Framework

*- No formal frontend/backend framework used*

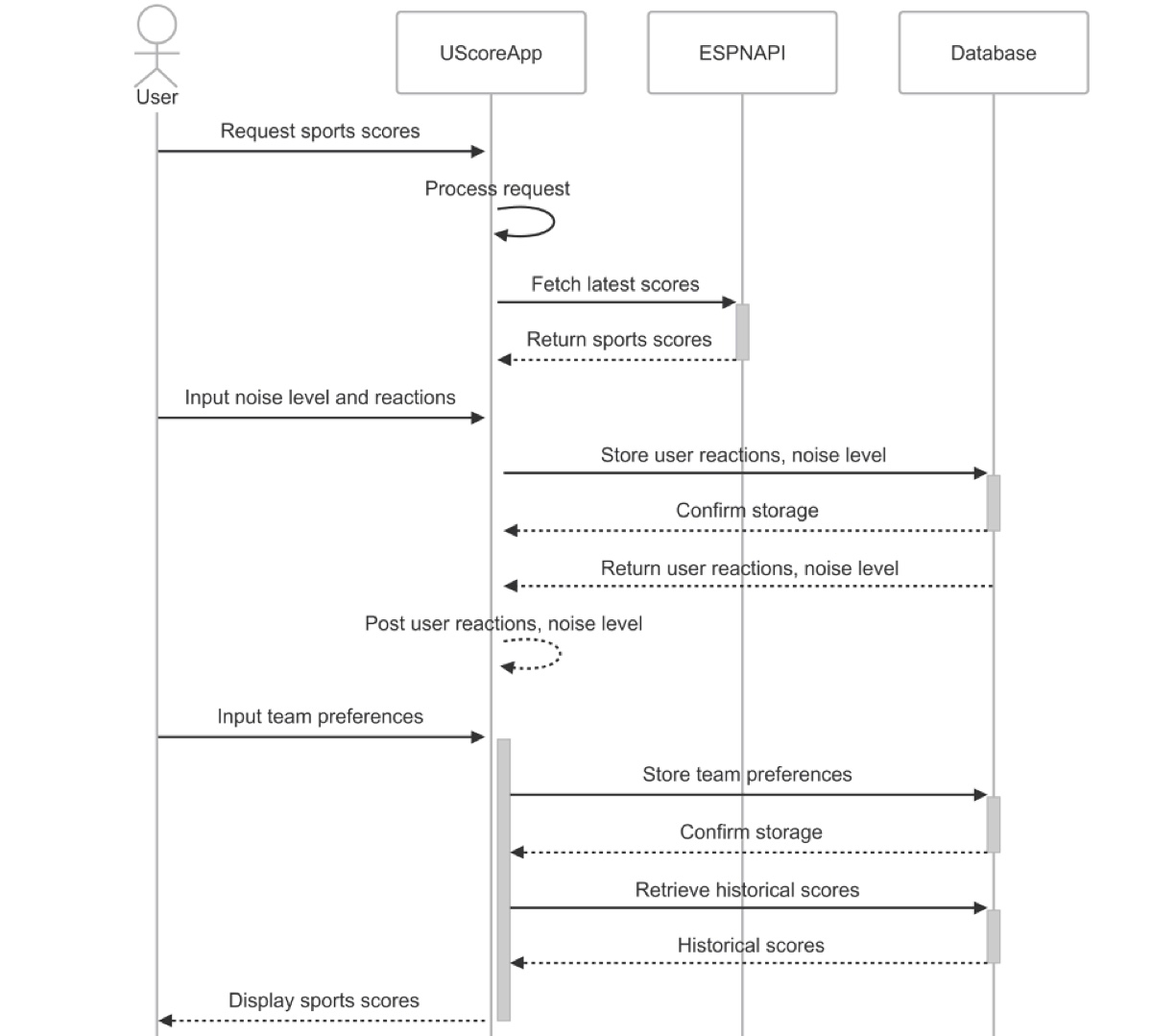
*- Vanilla HTML, CSS, JS for frontend*

*- AWS services and GitHub for hosting and version control*

### Functional Design

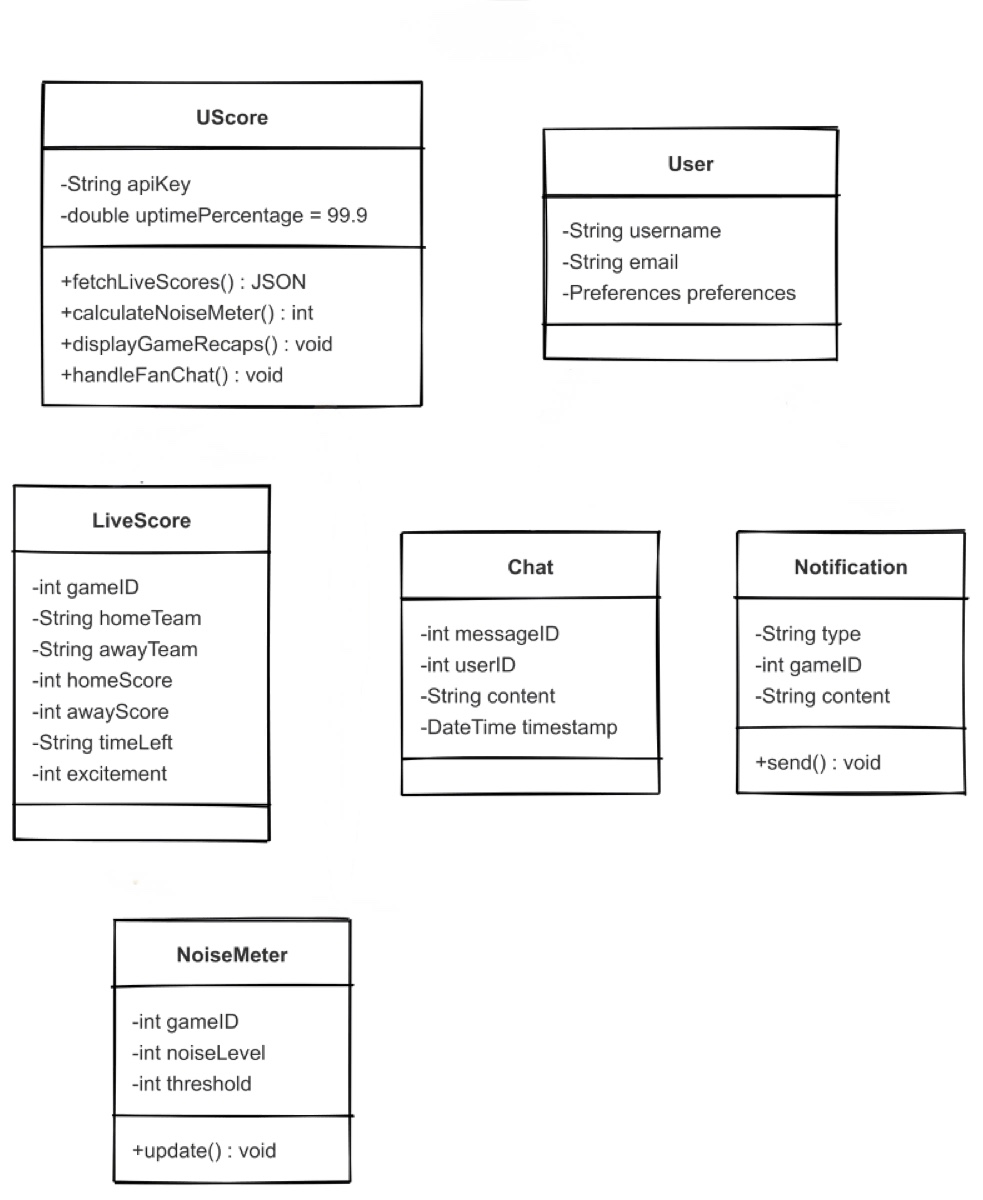
*<Identify all significant workflows as sequence diagrams using the following format>*

#### Diagram Title: Live Game Interaction Flow



### Structural Design

Class Diagram

**