JAYHAWK GO

Odds and Events
Software Architecture Document

Version 1.0

Revision History

Date	Version	Description	Author
10/23/22	1.0	Initial draft of software architecture document	Thomas, Mark, Wyatt, Michael, Nathan

Table of Contents

Software Architecture Document

1. Introduction

1.1 Purpose

This document provides a comprehensive architectural overview of the Odds and Events system. With numerous architectural views presented, there will be a different perspectives shown of the system. The goal is to show and capture any significant architectural decisions made to the system with illustrations provided.

1.2 Scope

This Software Architecture Document provides an architectural overview of Odds and Events. It can gather sports betting lines across multiple events for easy viewing and analysis. The data is compiled from requests to The Odds API, processed. and then stored in the system's SQLite database. With this information, users can make more educated betting decisions by comparing multiple sportsbooks lines for favorable advantages. Access to the web application is done via a web browser and the Internet, with no other specific software requirements or downloads needed by the user. Registration provides additional functionality for users, including settings preferences and odds trend lines.

1.3 Definitions, Acronyms, and Abbreviations

OAE Odds and Events

TBD To Be Determined

SRS Software Requirements Specifications

API Application Programming Interface

URL Uniform Resource Locator

UML Unified Modeling Language

1.4 References

- UPEDU SAD Template
- Vision and Scope Document
- Software Requirements Specification
- Use Case Specification
- Use Case Realization

1.5 Overview

This document contains information on Architectural Representation, Architectural Goals and Constraints, Use-Case View, Logical View, Interface Description, Size and Performance, and Quality of the Odds and Events system.

2. Architectural Representation

This document presents the architectural representation of Odds and Events. The views included are the use case view, deployment view, and implementation view utilizing Unified Modeling Language (UML).

3. Architectural Goals and Constraints

The Odds and Events web application that can be hosted and accessed through the Internet. The system consists of three major components: the web server that displays the web app to the Internet, the database that stores information, and the clients. The server and database will be hosted to the Internet. The web app itself can be accessed using a web browser on a laptop or mobile device.

Refer to Software Requirements Specifications document.

4. Use-Case View

The Use Case View is important to the selection of the set of scenarios and/or use cases that are the focus of an iteration. It describes the set of scenarios and/or use cases that represent some significant, central functionality. It also describes the set of scenarios and/or use cases that have a substantial architectural coverage or that stress or illustrate a specific, delicate point of the architecture.

Refer to the Use-Case Specifications document.

4.1 Use-Case Realizations

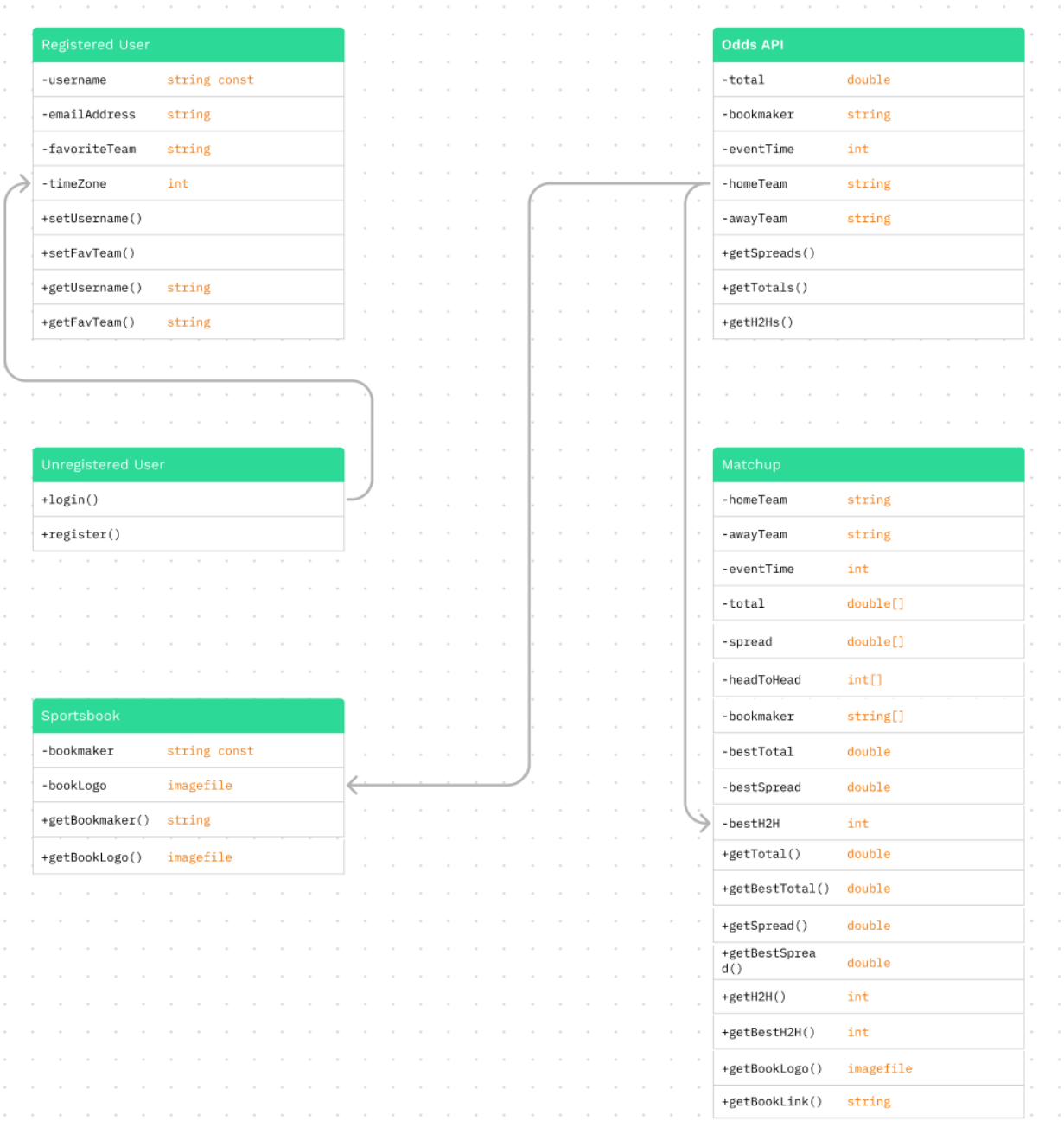
Refer to the Use Case Realization document.

5. Logical View

5.1 Overview

This section describes the decomposition of the design model in terms of class diagrams and their relationships.

5.2 Architecturally Significant Design Packages



Property	Description
Name	Matchup
Description	Page for sports event matchups
Relations	Dependency from Odds API
Attributes	homeTeam: home team awayTeam: away team eventTime: time and data of event total: array of over/under totals from sportsbooks spread: array of spreads from sportsbooks

	headToHead: array of head-to-head moneylines from sportsbooks bookmaker: array of bookmakers offering bets bestTotal: most advantageous over/under total bet bestSpread: most advantageous spread bet bestH2H: most advantageous head-to-head moneyline bet
Operations	getTotal(): receive matchup over/under totals from database getBestTotal(): receive best over/under total getSpread(): receive matchup spreads from database getBestSpread(): receive best spread getH2H(): receive matchup head-to-head moneylines from database getBestH2H(): receive best head-to-head moneyline getBookLogo(): receive logo image for sportsbook getBookLink(): receive URL link to sportsbook
Requirements	Requires data from Odds API

Property	Description
Name	Odds API
Description	API to receive bet data from sportsbooks
Relations	Dependency for Matchup and Sportsbook
Attributes	total: over/under for game bookmaker: sportsbook offering bet eventTime: time and date of event homeTeam: home team awayTeam: away team
Operations	getSpreads(): API call to receive spreads for event getTotals(): API call to receive over/under for event getH2Hs(): API call to receive moneylines for event
Requirements	Requires the Odds API and associated sportsbooks provide needed data

Property	Description
Name	Registered User
Description	An authenticated user who has registered and logged into application
Relations	Association with Unregistered User
Attributes	username: username emailAddress: email address favoriteTeam: favorite team timeZone: time zone
Operations	setFavTeam(): updates favorite team attribute getUsername(): gets the username for user getFavTeam(): gets the favorite team for user
Requirements	Requires an unregistered user to register and login

Property	Description
Name	Unregistered User
Description	A user that has not registered or logged into application
Relations	Association with Registered User
Attributes	N/A

Operations	login(): user logs into application and becomes a registered user register(): user registers account with application
Requirements	N/A

Property	Description
Name	Sportsbook
Description	Company offering sports bets
Relations	Dependency from Odds API
Attributes	bookmaker: sportsbook offering bets bookLogo: image for bookmaker
Operations	getBookmaker(): gets the name of bookmaker getBookLogo(): gets image for bookmaker
Requirements	Requires data from Odds API

6. Interface Description

The architecture of this system should support interface requirements through the implementation of a client-server model. The Odds and Events web application portion is to be implemented as an interactive and informative platform.

7. Size and Performance

The selected architecture supports sizing and timing requirements through the implementation of the client-server architecture. The client portion will be implemented through a client's device via a web browser connected to the Internet on a laptop or mobile device. The client can login to their account with a Google email, where the server will handle the storage of user information and processing data between the database and the web application. The system will be designed to handle varying types of connectivity with no issues.

8. Quality

The software architecture supports the quality requirements as stipulated in the Software Requirements Specification and Supplementary Specification portions. The software quality is held to a high standard to ensure the stability of the overall system. The architecture will minimize inefficiency and unreliability in order to maintain a pleasant user experience with a quick and reactive web application. Extensibility is involved by utilizing additional data from another section of the Odds API. Portability will be provided through its accessibility in a web browser from any device connected to the internet.