# JAYHAWK GO

# Odds and Events
# Use-Case Specifications

**Version 1.1**

# Revision History

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| 29/09/22 | 1.0 | Initial draft of use-case specifications | Michael, Nathan, Thomas, Mark, Wyatt |
| 09/10/22 | 1.1 | Formatting improvements and changes | Mark, Nathan |
| | | | |
| | | | |

# Table of Contents

# Use-Case Specifications

## 1. Use-Case Model

### 1.1 Introduction

The web application will be available to any user with an internet connection. Users of the web application fall into two types: unregistered and registered. When an unregistered user visits the site, they can choose to create a registered account with a username and password. When the user then logs in to the web application with that account, they become a registered user with access to personalization features.

### 1.2 General Actors Descriptions

1.2.1 Unregistered User

Anyone using the Internet to access the online web application. Can access the same basic data as if it were a regular web page. Lack of user registration lessens interactive ability and personalization within the web application.

1.2.2 Registered User

A user that logs in to the web application via username and password. Preferences for these users are saved within the system, allowing the user to better curate their experience to fit their personal needs.

1.2.3 Database Manager

An external actor that allows other roles or actors, including itself, to manipulate and use data in the database.

1.2.4 Data Sources

The range of websites and API that will be accessed in order to import raw information into the database. Only these sources will be used to read the data.

1.2.5 Account Management

An external actor that authenticates user account credentials during registration and login.

### 1.3 Use-Case Model Hierarchy

**Needs Picture**

*1.3.1 Use-Case Package One*
- **Description**
- **Use Cases**
- **Actors**
- **Relationships**
- **Packages Owned**

1.3.1.1 Packages Diagram

1.3.1.2 Sub Package One
- **Description**
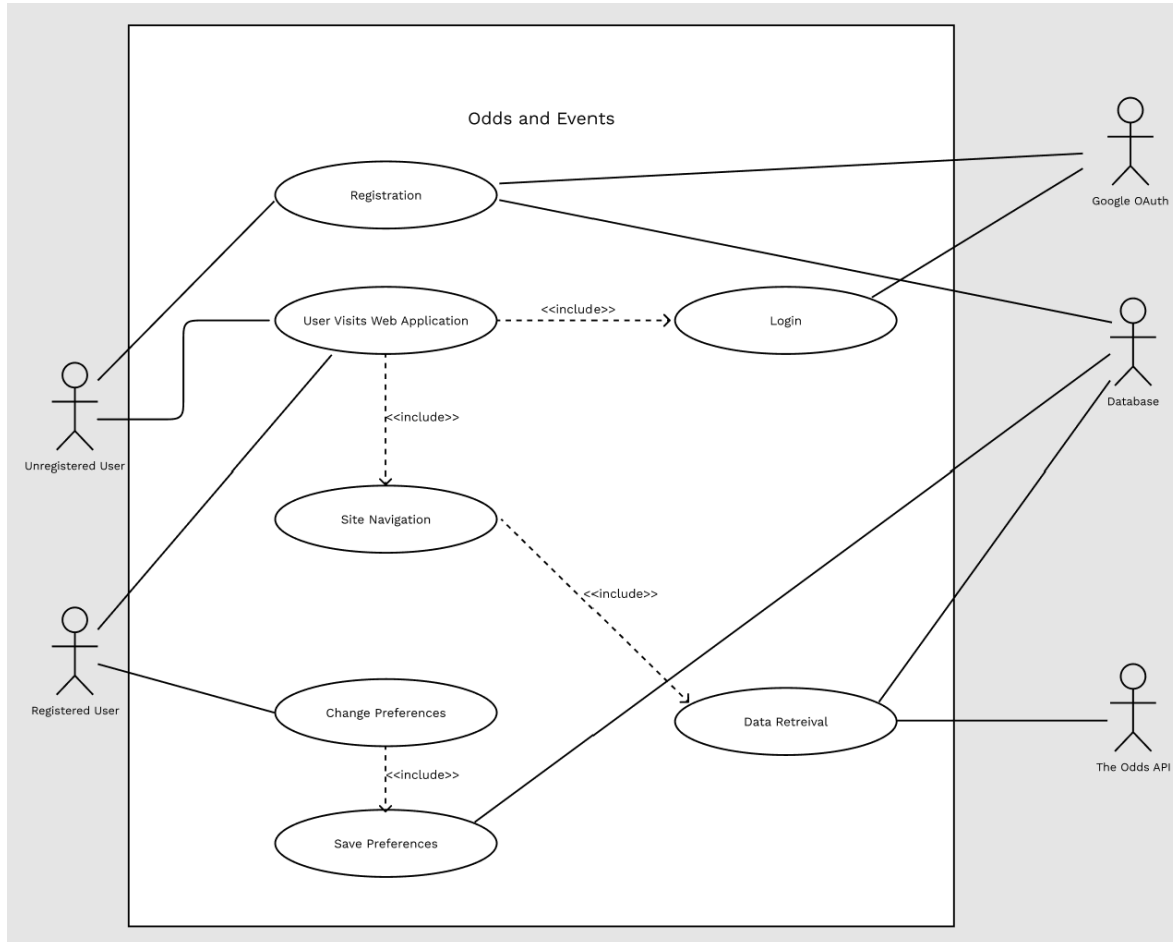- **Use Cases**
- **Actors**
- **Relationships**

1.3.1.3  Sub Package Two

*1.3.2  Use-Case Package Two*

## 1.4  Diagrams of the Use-Case Model

## 2.  Login

### 2.1    Brief Description

This use case describes how a user logs in to the Odds and Events web application. The actors starting in this use case are unregistered users that have yet to sign into their registered user account, and by the end of the login process will either be logged in to their registered user account or redirected to the account registration page if they did not previously create an account with Odds and Events.

### 2.2    Flow of Events

#### 2.2.1   Basic Flow

1. The use case begins when a user types in the URL for the Odds and Events login page, clicks on the login button or otherwise accesses the login page.

2. The website displays the login page.

3. The user logs in via their Google account.

4. Google OAuth Credentials checks for valid credentials.

5. If the credentials correspond to a registered user, then the user is logged in.

6. The website redirects the user to the homepage.

#### 2.2.2   Alternative Flow
#### 2.2.2.1   New User

If Google OAuth Credentials reports that the user does not have a registered account, the website prompts the user to go through the registration process. Once the account is created, the user is signed in and considered a registered user.

### 2.3    Special Requirements

#### 2.3.1   Google OAuth Credentials

Must be compatible with Google OAuth Credentials in order to properly validate user login.

### 2.4    Preconditions

#### 2.4.1   Internet Access

The web application requires internet access to communicate with Google OAuth Credentials for login.

#### 2.4.1   User Exists

To successfully log in, the user must have already gone through the registration process.

### 2.5    Postconditions

#### 2.5.1   Logged In

If the user had a valid account, the user is logged in and redirected to the web application's homepage.

#### 2.5.2   Registration

If the user does not have a valid account, they are redirected to the registration page, where they can set up an account.

### 2.6    Extension Points

#### 2.6.1   Redirected to Registration Page

The user can be redirected to the registration page if their account does not exist in the system.

### 2.7    Relationships

User login requires communication with Google OAuth in order to connect to the user's Google account.

### 2.8    Use-Case Diagrams

See use case diagram above in section 1.4.

### 2.9 Other Diagrams

Necessity of Other Diagrams is TBD

## 3. Registration

### 3.1 Brief Description

This use case describes how a user creates a registered user account for the Odds and Events web application. The actors starting in this use case are unregistered users, and by the end of the registration process will either have created an account with Odds and Events and be signed into that account or will be redirected to the login page if they previously created an account with Odds and Events using the Google account they attempted to provide during registration.

### 3.2 Flow of Events

#### 3.2.1 Basic Flow

1. The use case begins when a user types in the URL for the Odds and Events registration page, clicks on the register button or otherwise accesses or is redirected to the registration page.

2. The website displays the registration page.

3. The user is asked to provide a Google account to register their account.

4. Google OAuth checks for valid user credentials.

5. If the credentials correspond to a valid Google account holder, then the email of that account is saved for later reference for the user's Odds and Events username.

6. Once logged in for the first time via Google OAuth Credentials, the user is prompted to create a username for their account. This username is saved alongside the email for the same account.

7. The user is then redirected to the Odds and Events home page, now as a registered user.

#### 3.2.2 Alternative Flows
#### 3.2.2.1 User Has Pre-Existing Registered Account

If Google OAuth reports that they already have an account registered for Odds and Events, this is reported back to the user, telling them to instead go through the login process to continue signing in.

### 3.3 Special Requirements

#### 3.3.1 Google OAuth

Must be compatible with Google OAuth Credentials to properly validate user login.

### 3.4 Preconditions

#### 3.4.1 Internet Access

The web application requires internet access to communicate with Google OAuth Credentials for registration.

### 3.5 Postconditions

#### 3.5.1 Account Created

If the user has a valid Google account and did not have an account set up with Odds and Events, then a new Odds and Events account is created for that user.

#### 3.5.2 Redirected to Login

If the user had a valid account, they are redirected to the login page, where they can sign into their account.

### 3.6 Extension Points

#### 3.6.1 Redirected to Login Page

The user can be redirected to the login page if their account already exists in the system.

### 3.7 Relationships

Registration requires communication with Google OAuth in order to create their OAE account, as well as communication with the database in order to store the account's email and username information.

### 3.8 Use-Case Diagrams

See use case diagram above in section 1.4.

### 3.9 Other Diagrams

Necessity of Other Diagrams is TBD.

## 4.    Site Navigation

### 4.1    Brief Description

This use case describes how a user navigates the web application. Both unregistered and registered users can browse the basic level content, while some functionality is only provided to registered users.

### 4.2    Flow of Events

The use case begins when the web application home page is accessed via the Internet.

#### 4.2.1  Basic Flow

1. The home page is accessed displaying upcoming and most in demand sporting events odds.

2. If the user logs into the system with their credentials, the event page is shown providing full details on bets and trends for a sports matchup.

3. Link to a sportsbook is clicked to fulfill a bet through their site and a new tab opens to that associated bet.

#### 4.2.2  Alternative Flows

##### 4.2.2.1   Navigating Through League Page

1. The home page is accessed displaying summarized details for upcoming and most in demand sporting events odds.

2. The sports league page is accessed providing a list of the upcoming events with summarized event betting odds.

3. Link to a sportsbook is clicked to fulfill a bet through their site and a new tab opens to that associated bet.

### 4.3    Special Requirements

None

### 4.4    Preconditions

#### 4.4.1  Registered User

To access an event page the user must be registered and login.

### 4.5    Postconditions

None

### 4.6    Extension Points

#### 4.6.1  Navigating to External Sportsbook

When the user clicks an external link to a sportsbook to place a bet a new tab is created to the associated bet provided by that sportsbook.

### 4.7    Relationships

None

### 4.8    Use-Case Diagrams

See use case diagram above in section 1.4.

### 4.9    Other Diagrams

Necessity of Other Diagrams is TBD.

## 5. Data Retrieval

### 5.1 Brief Description

In routine intervals or by user request, the back end will request API data to update constantly changing values.

### 5.2 Flow of Events

#### 5.2.1 Basic Flow

1. The use case begins when a user visits a sportsbook page for the Odds and Events.

2. The user is registered and views a specific game that requires live data.

3. An API request is made upon visiting the web page and is displayed to the user.

4. Data that is requested from the API is formatted to be ready for storing in the local database.

#### 5.2.2 Alternative Flows

#### 5.2.2.1 Request Interval

1. The use case begins when the defined time interval is met.

2. The API is requested every interval to constantly update older data.

### 5.3 Preconditions

#### 5.3.1 Internet Access

The web application needs internet connection to request data from the online API.

### 5.4 Postconditions

#### 5.4.1 Request is Valid

The request to the API was successful and valid. The data is obtained and displayed on the web application.

#### 5.4.2 Request is Invalid

The request to the API was unsuccessful due to the service being down or an invalid request was made. The web application will remain unchanged with older data until a valid request is made.

### 5.5 Extension Points

#### 5.5.1 Display Older Data

In the case where the request to the API is invalid, older data stored in the database will be retrieved and displayed.

### 5.6 Relationships

Data Retrieval requires communication with The Odds API in order to gather data, as well as with the database in order to store the requested data into OAE's local database.

### 5.7 Use-Case Diagrams

See use case diagram above in section 1.4.

# 6. Data Storage

## 6.1 Brief Description

This use case describes how data is stored for the Odds and Events web application. The actors starting in this use case are the registered users, unregistered users, and the database. By the end of this use case, data has been stored within the Odds and Events web application database.

## 6.2 Flow of Events

### 6.2.1 Basic Flow

1. A user visits the web application.

2. A user inputs information including registration, preferences, or site navigation.

3. This data is then stored within the Odds and Events database.

## 6.3 Preconditions

The user must input data to be stored.

## 6.4 Postconditions

Input data is stored in the database.

## 6.5 Relationships

## 6.6 Use-Case Diagrams

See use case diagram above in section 1.4.

## 7.    Save Preferences

### 7.1    Brief Description

The purpose of this use case is to allow the user to save preferences to their account however they want. This includes data sort preferences, light/ dark theme, etc. That will be saved for the next time they log in.

### 7.2    Flow of Events

#### 7.2.1  Basic Flow

1. A user visits the web browsing page.
2. They login or create an account.
3. Once login is complete and verified, the user can navigate to the account preferences page.
4. Here they can select and change their preferences to their liking.

#### 7.2.2  Alternative Flows

1. If a user does not have an account, they must create one.

#### 7.2.2.1   Filter By Preference

In addition to account preferences like light or dark theme and favorite team, the user can filter the data based on their preference.

### 7.3    Special Requirements

#### 7.3.1  Existing Account

The user will not be able to save and change preferences if they do not have a valid account. The user can be redirected to the account creation page if necessary.

### 7.4    Preconditions

#### 7.4.1  Internet Connection

The user must be able to connect to the internet as well as have an internet browser.

#### 7.4.2  Existing Account

The user must currently be logged in and have an account.

### 7.5    Postconditions

To access an event page the user must be registered and login.

### 7.6    Extension Points

None

### 7.7    Relationships

The user can only edit and save information if they have an account and are logged in.

Saving certain preferences will interact and change how the user browses the website.

### 7.8    Use-Case Diagrams

See use case diagram above in section 1.4.

### 7.9    Other Diagrams

Necessity of Other Diagrams is TBD.