HSC SDD Major Project Diagrams

CS:GO Case Opening Simulator

Oscar Reinitz

# Storyboard

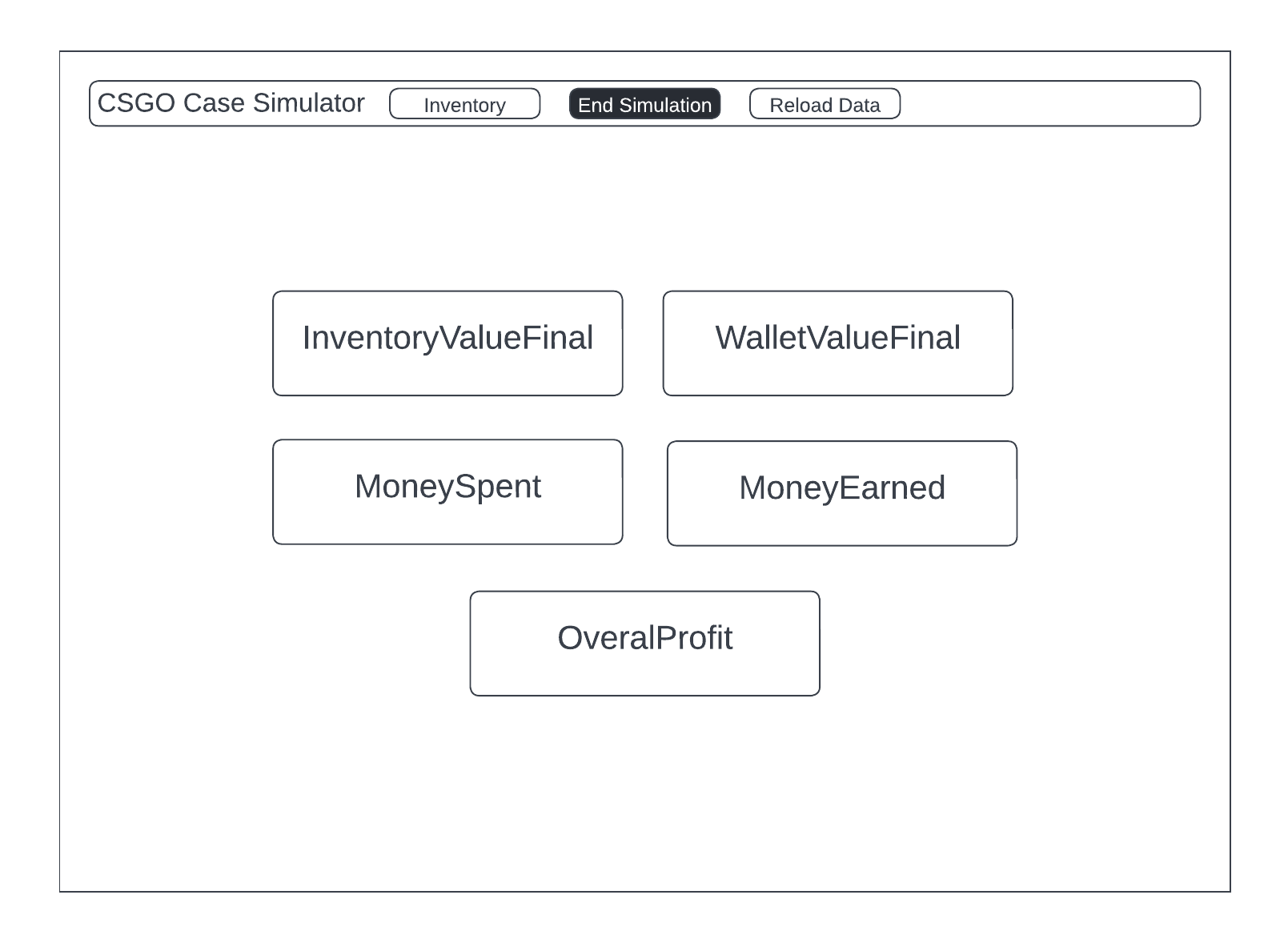
Diagram, PowerPoint

Description automatically generated with medium confidence

# Screen Designs

A picture containing graphical user interface

Description automatically generated



# Data Dictionary

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Data Format | Scope | Description |
| imgHashdata | Object | Global | JSON file retrieved from CSGO:Backpack that is converted into an object within the JS code. Contains the image hashes for every item in Counter Strike Global Offensive |
| itemData | Object | Global | JSON file retrieved from a sorted mirror of the CSGO:Backpack object. Is updated regularly by a GitHub bot, contains the price data and probability odds for each case and their items variants. |
| siteActive | Boolean | Global | Stored in local storage. Records if the simulation has begun or not. |
| inventoryData | Object Array | Global | Stored in local storage. Holds the data of items won from opened cases. |
| homeDiv | HTML Element | Global | Allows manipulation of homepage Div |
| mainDiv | HTML Element | Global | Allows manipulation of main case screen Div |
| simBarDiv | HTML Element | Global | Allows manipulation of header bar Div |
| loadingDiv | HTML Element | Global | Allows manipulation of the loading screen Div |
| inventoryDiv | HTML Element | Global | Allows manipulation of inventory Div |
| inventoryValue | Integer | Global | Stored in local storage. Holds current value of inventory. |
| totalSpend | Integer | Global | Stored in local storage. Holds how much money has been spent |
| totalSold | Integer | Global | Stored in local storage. Holds how much money has been made from sales. |
| totalProfit | Integer | Global | Calculated when simulation is finished, combines the value of sold item, the value of current inventory and how much money was spent to calculate the overall profit of the simulation |
| startValue | Integer | Global | Holds the initial starting value once selected by user. |
| myBalance | Integer | Global | Stored in local storage. Holds user’s current balance. |
| lenCase | Integer | Local | Length of the nameArray used within the populateCases function |
| nameArray | Object Array | Local | Object array created from a select part of the itemData object |
| imgHash | String | Local | URL string that is created using the current case image hash data. |

# IPO Charts

## Case Search

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| * Name of Desired case | * Binary search of an array containing the case names. * Hide DIVs of cases which names do not match the desired case name. | * Case matching the input name is shown to the user |

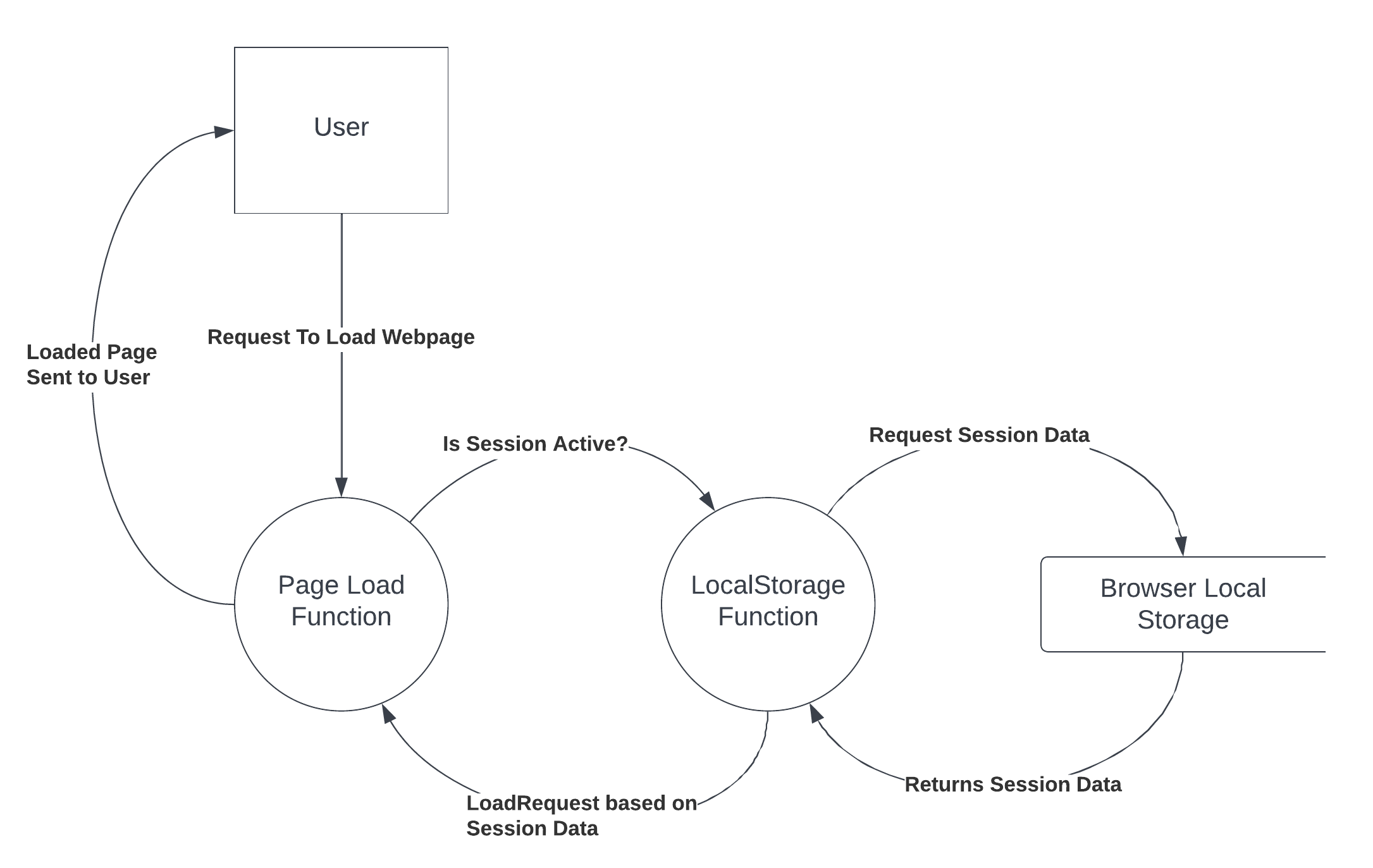
## Case Roll

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| * User Balance * Case Data | * Deduct case price from user balance * Retrieve case odds from case data * Retrieve item list * Randomise an item based upon case odds * Store won item in local storage | * Display won item to user |

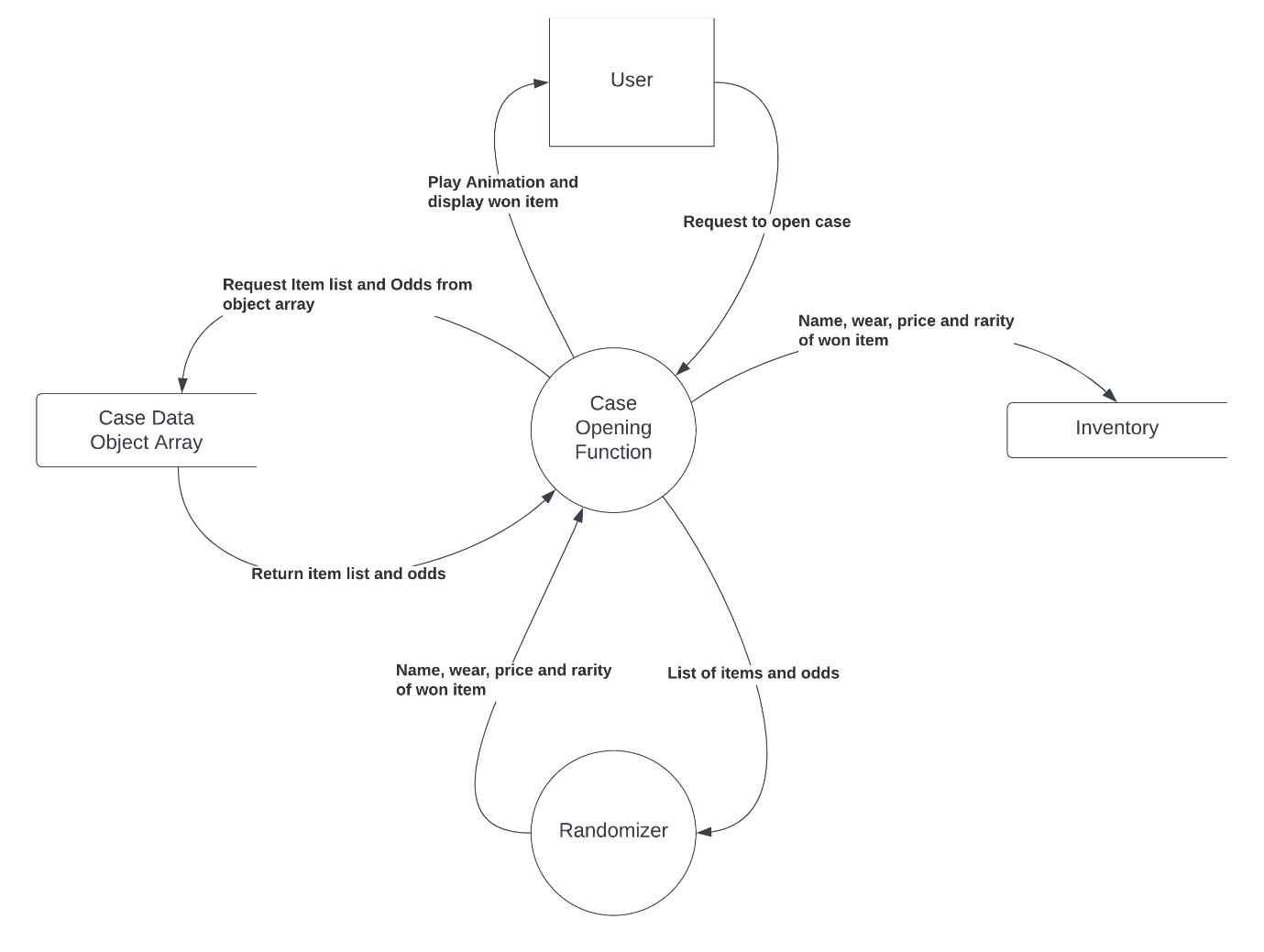
## Case Page Loading

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| * Image Hash Data * Case Data | * Toggle loading screen div while populating * Determine number of cases from data * Read name of case from data and populate HTML element * Read image hash from data and populate HTML IMG source * Read price data of case from data and populate HTML element | * Loading screen shown while cases populate the website * Once loaded, a grid of each case, their price and image is shown to user. |

# Data Flow Diagram Page Loading

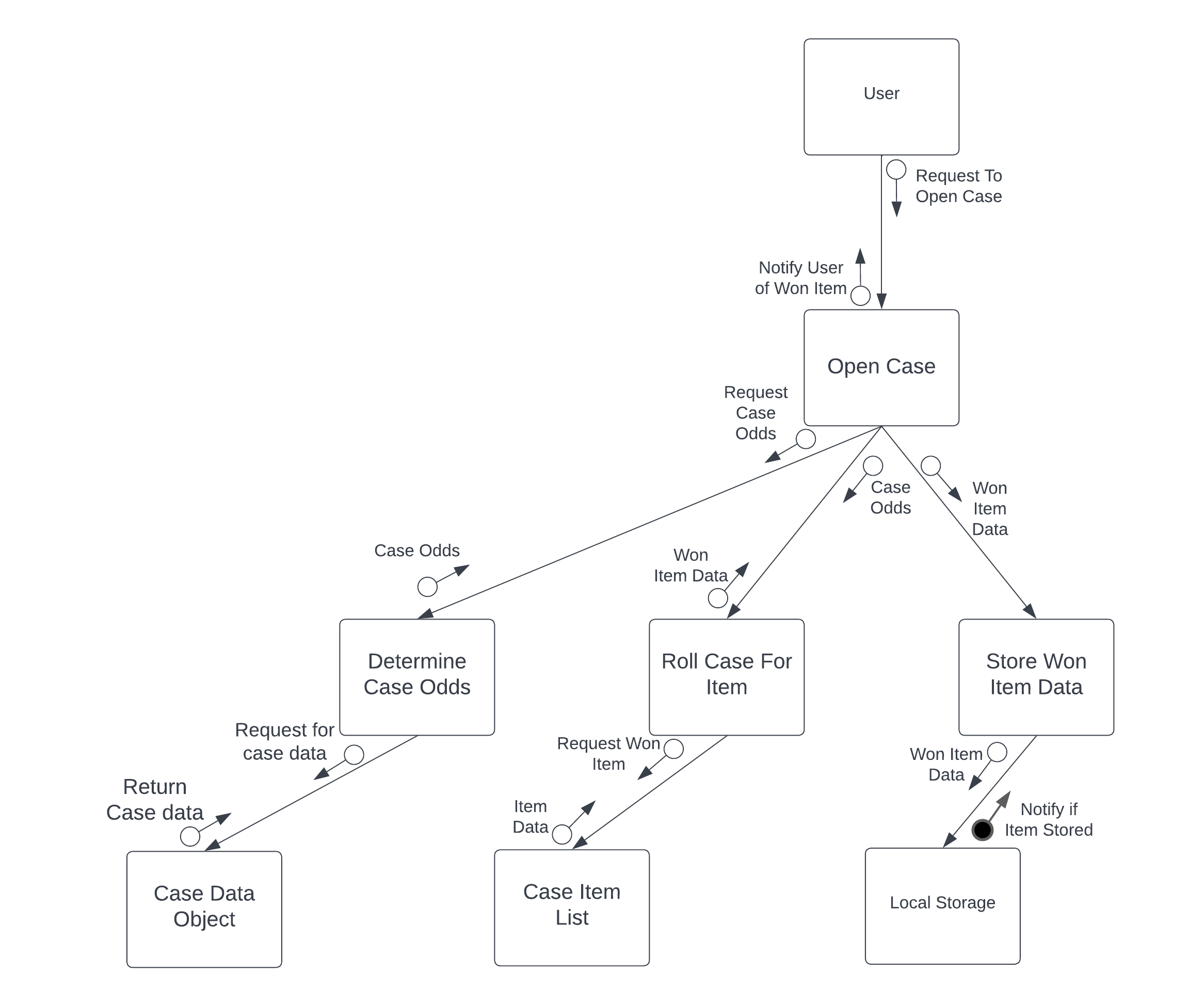


## Case Opening



# Structure Diagram

## Case Opening



# System Flowcharts

## Case Search

Diagram

Description automatically generated

## Case Opening

