**Design Document: "Zookeeper's Challenge"**

**Components:**

* Main Program
  + Purpose: Orchestrates the execution of the application.
* Animal Class
  + Purpose: Represents individual animals with essential attributes.
* Subclasses (Hyena, Lion, Tiger, Bear)
  + Purpose: Specific subclasses representing different species of animals.
* File IO Module
  + Purpose: Handles File I/O operations for reading input files and generating output files.

**Data Structures:**

* Vector
  + Purpose: Stores Animal objects for further processing.
* Map
  + Purpose: Maintains a species count map for tracking the number of animals of each species.

**Interactions:**

* Main Program and File IO Module
  + Main program interacts with the File IO Module to read data from input files.
* Animal Objects and Vector
  + Animal objects are stored in a Vector for further processing and manipulation.
* Animal Objects and Species Count Map
  + Species count maps are updated based on the data extracted from Animal objects.
* Report Generation
  + Report is generated by writing species counts to an output file.

**Sample Design Flow:**

1. Initialize a Vector for storing Animal objects to manage data efficiently throughout the program.
2. Read input files using the File IO Module to extract necessary information about arriving animals.
3. Create Animal objects based on the data extracted from input files and add them to the Vector for further processing.
4. Update the species count map by iterating through the Animal objects in the Vector and incrementing the count for each species encountered.
5. Generate a report by writing species counts to an output file, providing a comprehensive overview of the zoo's population.
6. Close input and output files to finalize the operation and ensure proper resource management.