**Create a simple and headless client-server Java application**

**Server**

* Create a headless server application in Java which listens on a configurable TCP port. The command line options to start the server are:
  + server -port <integer between 1 and 65535> -data <folder to store information> -proc\_count <positive integer>
  + -port option is optional. Default port is 3000. A valid TCP port number is between 1 and 65535.
  + -data option is the location of the folder where server keeps its data store. This is optional as well. If not specified, server stores data in "serverdata" folder in user's home directory. If the directory does not exist, it is created. If it exists, it is used. Server must deal with permissions to write to this folder and exit gracefully if it cannot write to, read from or create files/folders under the specified folder.
  + proc\_count is a positive integer and is optional. If it is not provided, the default value is 2. Its usage will become clearer further down in the specification.
* When server starts, it checks if it can access data folder and can bind to the specified port. If it can, it creates initial\_proc\_count number of worker threads to process requests from the clients.
* Server's job is to maintain a database of birds, their characteristics and sightings. In order to achieve this, server stores data in two XML files:
  + **birds.xml:** which contains a list of birds with their 4 characteristics: name, color, weight and height
  + **sightings.xml:** which contains the location (string) and time (date & time) for each bird sighting
* Server at start up loads the two files and stores data in any data-structure of programmer's choice.
* Server has a thread that periodically saves the contents of in-memory data-structure to two XML files.
* Server also saves any data on shutdown.

**Client-Server communication**

* Programmer is free to choose message format of the messages exchanged between the server and client to perform the actions mentioned in the Client section.

**Client**

* Client application is a headless Java application with following commandline syntax:
  + -serverPort <integer> -addbird -addsighting -listbirds -listsightings -remove -quit
  + -serverPort is an optional option and defaults to port 3000. It is used to initiate a TCP connection to the server.
  + -addbird, -addsighting, -listbirds, -listsightings, -remove and -quit are exclusing arguments. Only one of these can be provided on the command line at one time.
  + Client application quits after performing one of the operations mentioned above: addbird, addsighting, listbirds, listsightings, remove and quit.
* addbird option
  + When this option is specified, the client asks the user to enter the bird name, color, weight and height on the command line. The user enters these values on the command line and client takes this data and sends this using programmer's preferred message format over the TCP connection established with the server. Server parses the message and inserts the bird into its data structures. If the bird name is duplicate, server sends an error message to the client using programmer's preferred message format between the client and the server. If everything went well, server sends a success message and client displays "Bird <name> successfully added to the database" on the console.
* addsighting option
  + When this option is specified, the client asks the user to enter the bird name, location, date and time on the command line. Client sends the message to the server encapsulating this information and server performs the requested operation and replies with a success or failure message. Client prints the success or failure message on the command line.
* remove option
  + When this option is specified, client prompts the user for bird name only. Client sends a message to the server to remove the bird from the database. Server removes the bird from its store. Server then sends an appropriate message to the client and client displays a success or failure message on the console.
* quit option
  + When this option is specified, client sends a quit message to the server and server performs a shutdown operation and quits. See the server section on what the server has to do on shutdown.
* listbirds option
  + When this option is specified, the client sends a message to the server to send the entire bird list. Server sends this information back and client prints all the birds on the server in a table with a column for name, color, weight and height. The table is printed in alphabetical order of the bird names. Client also prints total number of birds present on the server.
* listsightings option
  + When this option is specified, the client asks the user to enter the bird name, start date and end date on the command line. The bird name can be a regular expression. Client takes this data and sends to the server a message to return the bird sightings. Server uses Java regular expression library to match the appropriate bird names and uses the date range to find the appropriate entries. Server sends this data to the client and client prints a table with following columns: name, date which is sorted by alphabetical order of the bird names and further sorted in order of time.