Software Engineering

System Design and Architecture

4+1 views

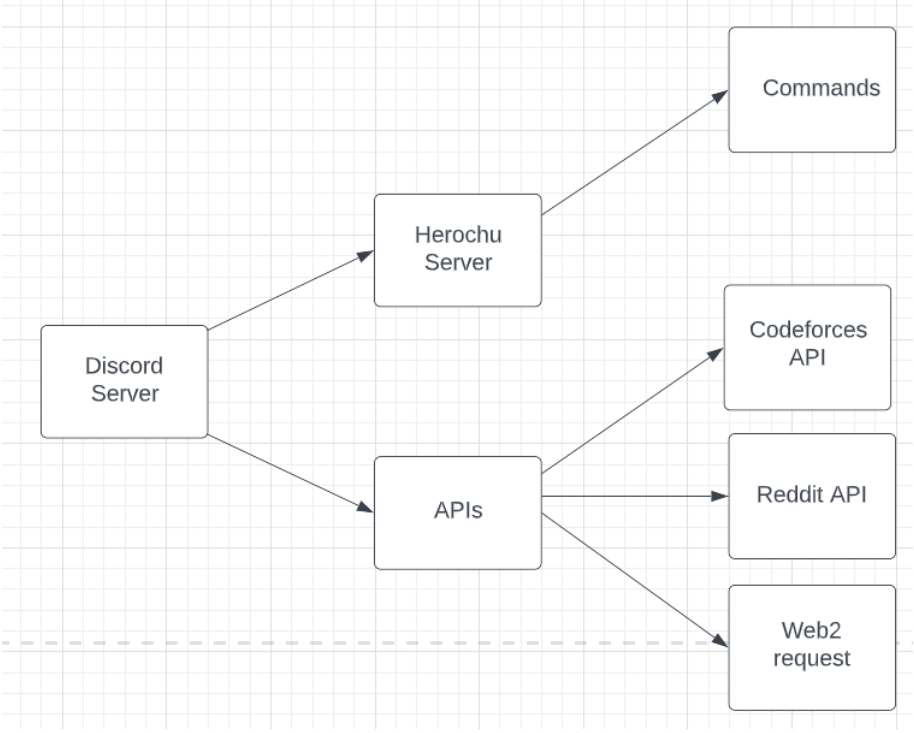
Tatvam (B20CS077)
Maniya Yash Rajeshbhai (B20CS033)

Introduction:
This document contains the information on the system design and architecture of our clist bot

System Architecture :

1. Physical View

This view tells about the physical layout of the system and its components. It refers to the structure of physical elements of the system.
This view is also called the Deployment view.



- The view shows how the components of our system interact with each other and their direct or indirect dependencies.
- The top-tier objects are User-actions, Discord and Heroku Servers.
- Low level objects and modules are those coming below the main runner code script.
- The above diagram will give a rough idea to software developers (us in this case) to implement and create the modules, and architecture via coding the corresponding elements.

2. Logical View :

This view tells about the functionality that the system will provide to the end-users.

The clist bot will provide the following functionalities to the end-user:

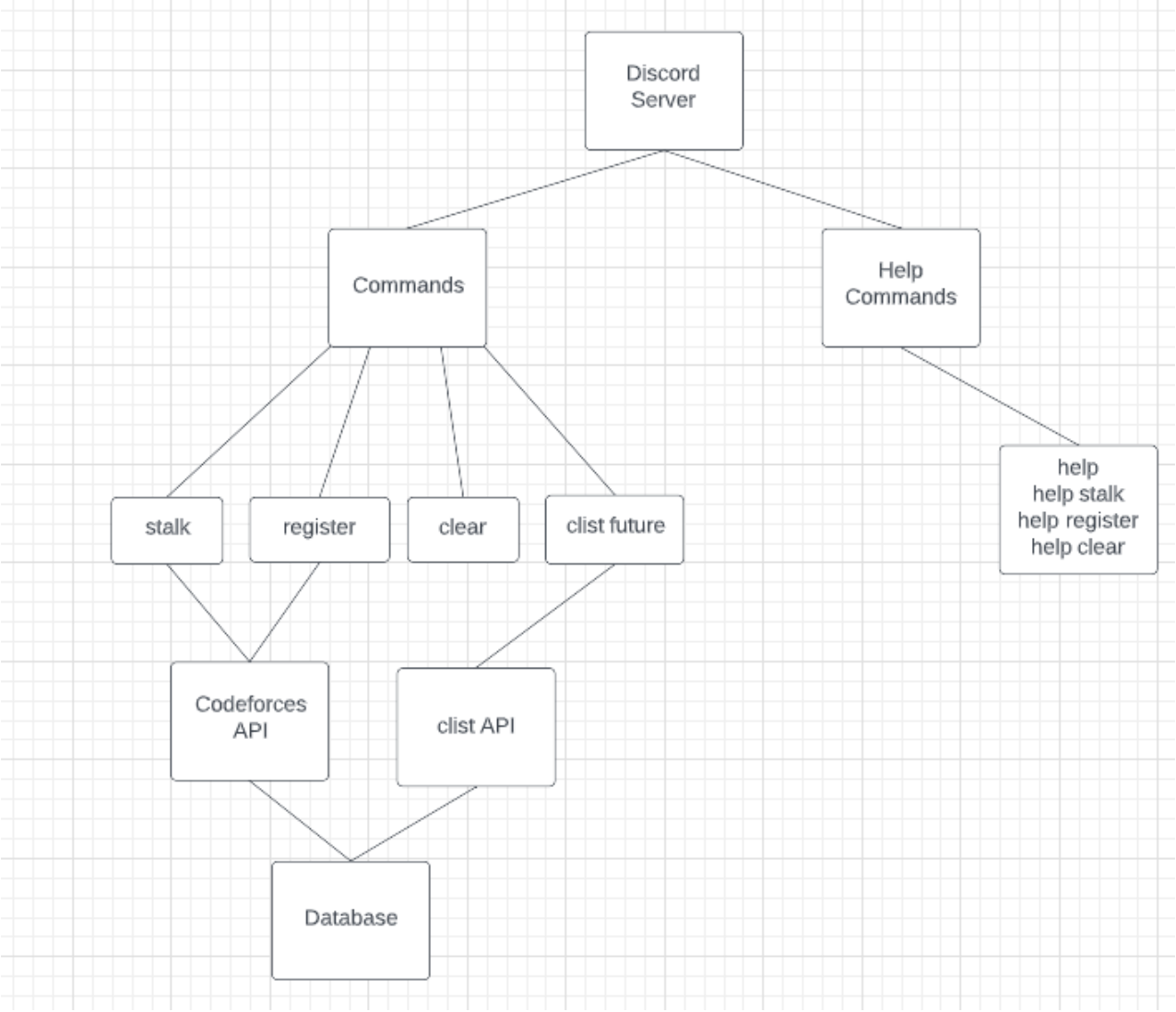
- User Commands that bot can handle :
    - ;clist future
    - ;clist future long
    - ;stalk <user_tag>
    - ;register <user_tag> <CF_handle>
    - ;userdata <user_tag>
    - ;stalk <CF_handle>
    - ;plot rating

- Help Menu :
    - Commands list
    - Commands brief description
    - Command syntax
    - Command aliases

- Server Access (to developer) :
    - Heroku - for hosting script
    - Discord - for auth and deployment

- API Access (via commands) :
    - CodeForces

- Reddit



3. Process View :

This view explains about the system processes and how they communicate, and focuses on the runtime behaviour of the system.

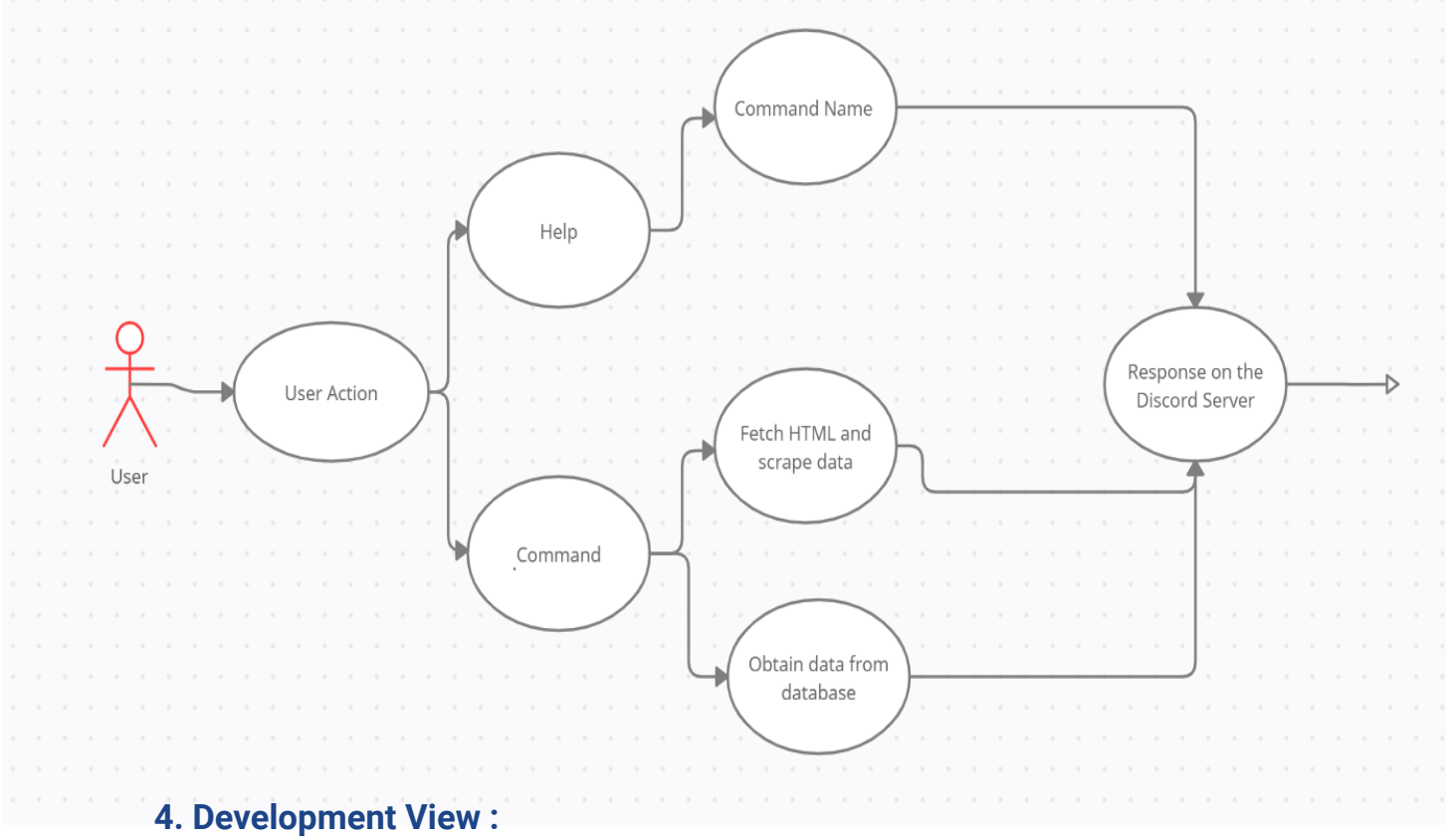
Here are some of the basic processes the bot will handel during the runtime execution of code:

- Command input :
    I. Script on Heroku gets command details
    II. Arguments are checked
    III. Returns error if detected
    IV. If no error : returns the corresponding output

V. The output is sent to channel via Discord server
VI. User accesses the required output

- Event Trigger :
  I. User triggers an event
  II. Script on Heroku recognizes the event category
  III. Processes the required output/backend calculations
  IV. Event ends

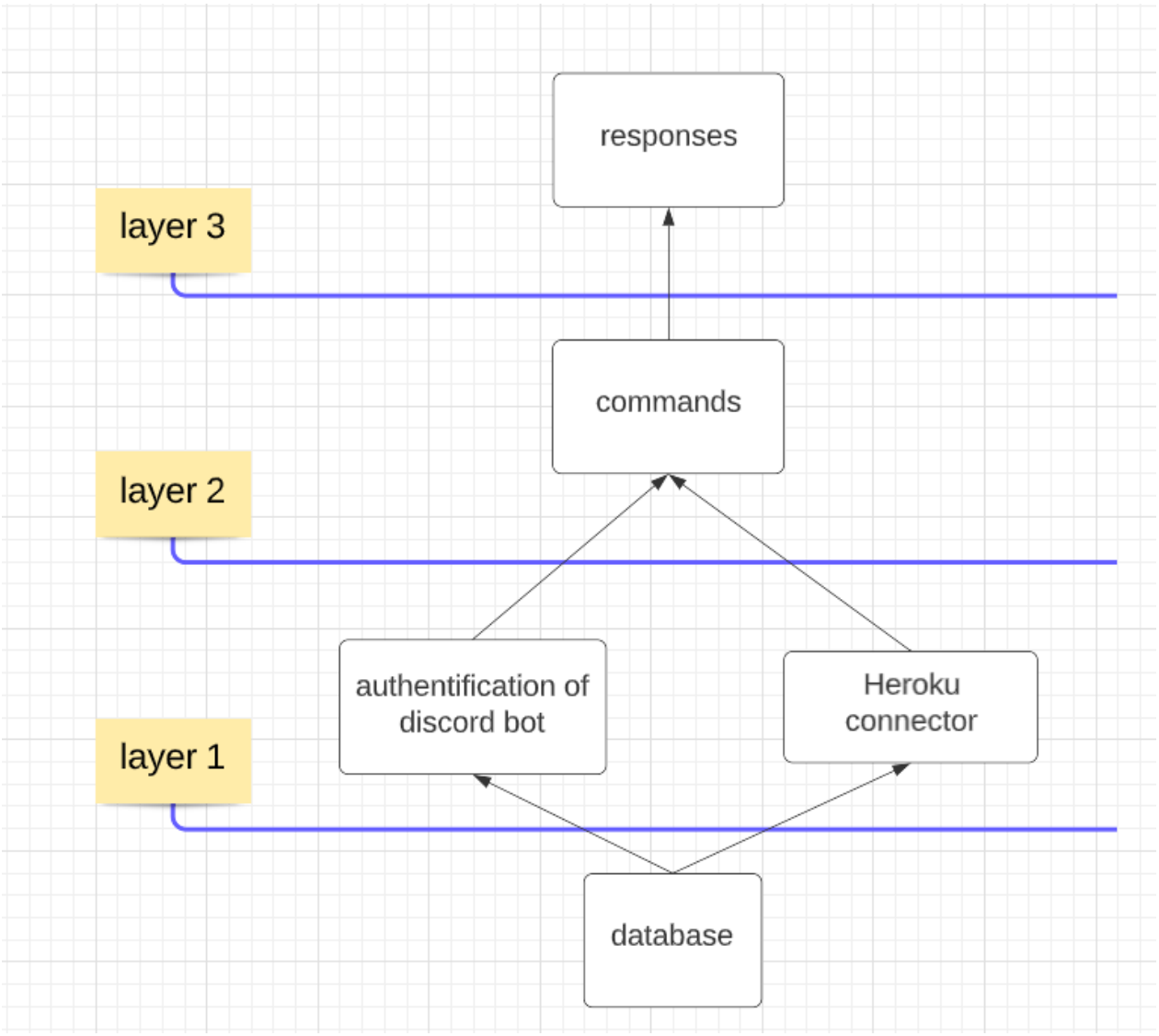
- User requests API information :
  I. Script on Heroku interprets request
  II. Returns error if detected
  III. If not, requests the information to corresponding API
  IV. API sends response in return
  V. Script does processing
  VI. Required output is sent to Discord Server
  VII. Discord then outputs the information to the User



4. Development View :

The development view is focused with software management and

represents a system from the perspective of a programmer. This view is also known as the implementation view.
It uses the UML Component diagram to describe system components. One of the UML diagrams used to describe the development view is the Package diagram.



5. (4+1) Use-Case View :

- A limited number of use cases, or scenarios, are used to demonstrate the architecture description in the fifth view. They are used to detect

architectural features as well as to show and evaluate architectural design.

- Suppose the user wants to stalk some account. To do that the user will have to use the command to stalk another user i.e., the ;stalk command. The user provides a handle in the command, suppose the user types ;stalk xyz. The bot will go to the codeforces website and search for the username given by the user(xyz). If there is any registered user account named “xyz”, the bot will provide the information about that account but if there is no such registered account the bot will show an error message showing that no such account exists.

- Another case is if the user wants to register himself/herself to the bot.For this the user will use the ;register command followed by the user tag in the discord server and the codeforces handle of the user. The bot on receiving the command will start to check whether the user tag and the codeforces handle provided by the user are valid or not. If not an error will be shown regarding the same and if everything is correct there will be a message for authentication before registering the codeforces handle with the user tag. If the authentication of the same fails then the bot will not register the user and if it passes the user will be registered by the bot along with his/her handle.