## TASK 1:

## Problem Requirements:

**Problem Description:**

Problem: Many people struggle to keep their plants alive, which is mainly due to the fact they are not sure what the proper instructions are for their plant and easily forget to carry out the instructions. It may be hard to find specific instructions on each plant without a database that keeps track of it.

Objective: A plant care tracker database that helps plant keepers in managing their plant collections effectively. It lets them store information about different plants, track plant condition, manage care activities such as activities like watering, fertilizing, or repotting for each plant. It helps set reminders so that plant care requirements are met. It can also help record events such as recording occurrences like flowering, pruning, or pest infestations.

Nouns: *plants, users, instructions, reminders, tracker, locations, plant collections, activity, events, plant condition*
Verbs: *store, manage, keep, add, update, record, monitor*

**Aggregated nouns and verbs**
*Nouns:*

- Plants Table:
    - plant_id (Primary Key)
    - Name
    - Description
    - care_requirements (e.g., watering frequency, sunlight exposure, temperature range)
- Plant Condition Table:
    - condition_id (Primary Key)
    - plant_id (Foreign Key referencing Plants table)
    - condition_date: Date when the condition was assessed.
    - condition_status: Indicates the condition status of the plant (e.g., good, fair, poor).
- Plant Collection Table:

    - collection_id (Primary Key)
    - plant_id (Foreign Key referencing Plants table)
    - user_id (Foreign Key references Users table)
    - Collection name
- Care Logs Table:
    - log_id (Primary Key)
    - plant_id (Foreign Key referencing Plants table)
    - activity (e.g., watering, fertilizing, repotting, pruning)
    - timestamp
- Reminders Table:
    - reminder_id (Primary Key)
    - user_id (Foreign Key referencing Users table)
    - plant_id (Foreign Key referencing Plants table)
    - task (e.g., watering, fertilizing)
    - frequency (e.g., daily, weekly, monthly)
- Users Table:
    - user_id (Primary Key)
    - Username
    - Email
    - Password
- Events Table:
    - event_id (Primary Key)
    - plant_id (Foreign Key referencing Plants table)
    - event_type: Describes the type of event (e.g., flowering, pruning, transplanting, pest infestation).
    - event_date: Date when the event occurred.

## Notes:

➔ Users can manage plant collections, allowing plants to belong to multiple collections, with each collection linked to the user's account. Plant conditions are tracked to optimize care efforts, while care activities are logged with timestamps. Reminders for care tasks are associated with specific plants and user accounts. Events such as flowering, pruning, or pest infestations are tracked.

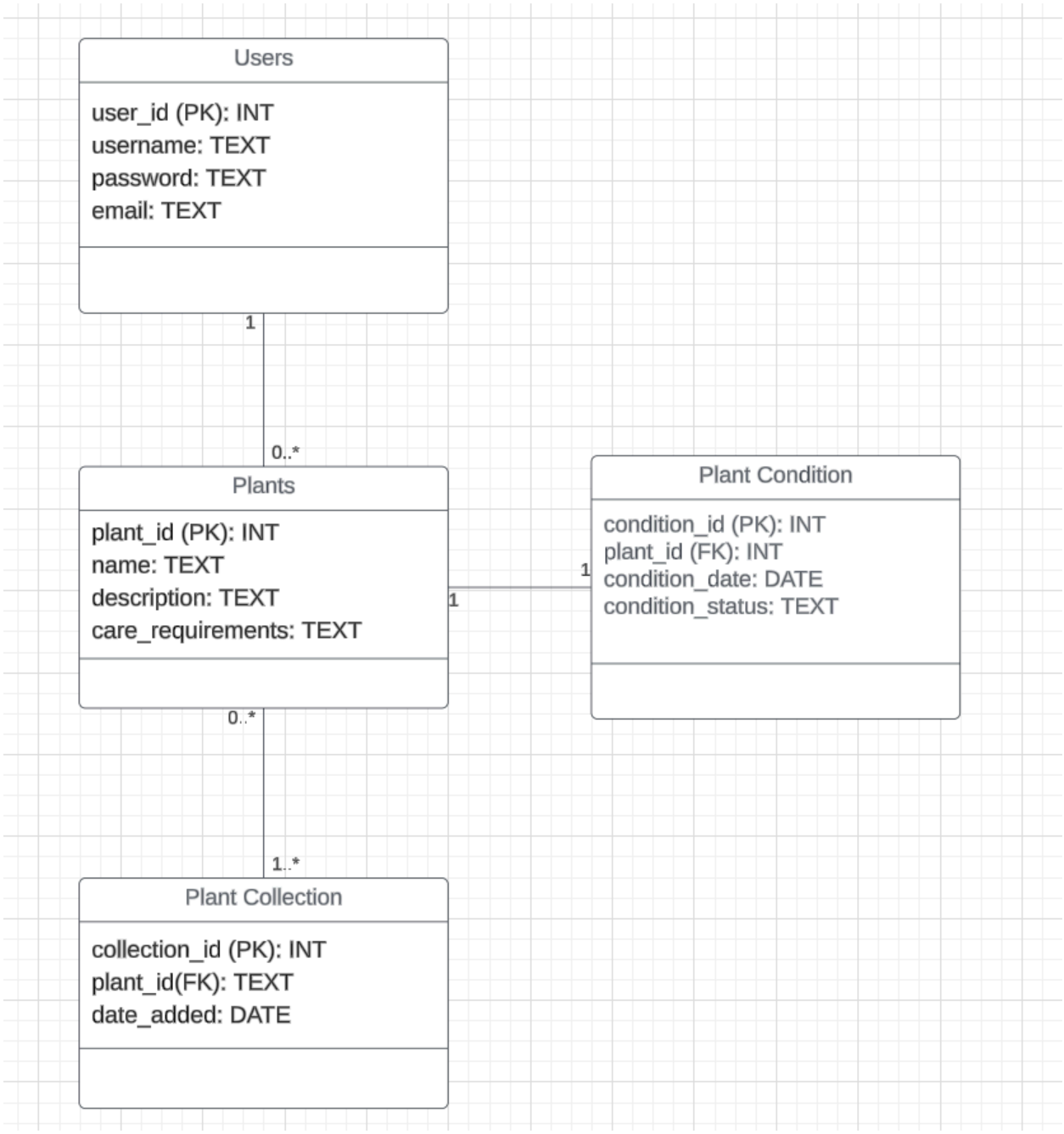
## **Rules**:

- **Plant** must have a unique identifier in the system.
- **Users** can **add** and **manage** information about different **plants**
- **Plants** can be a part of many **Plant Collections**
- **Plant Collections** need more than one plant
- **Plant conditions** are **tracked** to optimize **care routines** for each **plant**.
- **Users** can **log care activities** for each **plant**
- **Reminders** are set up for specific care tasks, associated with individual **plants** and **user** accounts.
- **Events** impacting plant growth are **recorded** for **plants** affected.

# In-memory key functionalities:

Plant Management: Users can add, update, and remove plants from their collections using hashes which will make retrieval and updates fast . They can also view details about each plant, such as care requirements and current condition.
Plant Collection Management: Users can create and manage multiple plant collections, as sets in Redis which will also let them organize their plants based on preferences or locations.
Care Activity Logging: Users can log care activities for each plant, such as watering, fertilizing, or pruning, along with timestamps.
Reminders: Users can set reminders for specific care tasks associated with individual plants or collections.
Event Tracking: Events impacting plant growth, such as flowering, pruning, or pest infestations, are recorded and displayed for users.

## UML Class diagram:



## TASK 2:

Plant Management:
Data Structure: Redis Hash
Implementation: A hash with the key plant:{plant_id} to store details about each plant where fields are name, description, care requirements, and plant condition. This lets me easily retrieve and update individual plants and their information.

Plant Collection Management:
Data Structure: Redis Set
Implementation: Sets with keys like collection:{collection_id} to store sets of plant IDs representing plant collections. Each collection can contain multiple plant IDs, which will allow users to organize their plants based on what they want.

Care Activity Logging:
Data Structure: Redis List
Implementation: Lists with keys like careLog:{plant_id} to store lists of care activities for each plant along with timestamps. This will help to keep track of the logs and the retrieval of care activities for individual plants.

Reminders:
Data Structure: Redis Sorted Set
Implementation:  A Sorted Set named reminders is used to manage reminders for users. The member in the set is a unique identifier for the reminder (e.g., reminder_123). The score associated with the member represents the timestamp for the next reminder.

Event Tracking:
Data Structure: Redis Hash
Implementation: Hashes with keys like event_{event_id} are used to store details about events impacting plant growth. Each hash field represents an event attribute:
plant_id: Stores the ID of the plant associated with the event.
event_type: Describes the type of event (e.g., "flowering", "pruning", "pest infestation").
event_date: Captures the date when the event occurred.

# TASK 3:

## 1. Plant Management (Hash):

**To Create a Plant:**
HSET plant_123 name "Snake Plant" description "Air-purifying succulent" care_requirements "Water every 2-3 weeks, needs bright indirect light"

**To Read All Plant Details:**
HGETALL plant_123
Output:
name: "Snake Plant"
description: "Air-purifying succulent"
care_requirements: "Water every 2-3 weeks, needs bright indirect light"

**To Update Plant Description:**
HSET plant_123 description "Needs even brighter indirect light now"

**To Delete a Plant:**
HDEL plant_123

## 2. Plant Collection (Set):

**To Create a Collection:**
SADD collection_1 plant_123 plant_456

**To Read All Plants in the Collection:**
SMEMBERS collection_1
Output:
1) "plant_123"
2) "plant_456"

**To Update the Collection:**
SADD collection_1 plant_789
SREM collection_1 plant_788

**To Remove a Plant from the Collection:**
SREM collection_1 plant_456

**To Delete the Collection (Remove all plants):**
SREM collection_1

## 3. Care Activity Logging (List):

**To Log/Create a Care Activity (Watering):**
RPUSH care_log_123 "2024-04-20:Watering"

**Get/Read All Care Activity Logs for a Plant:**
LRANGE care_log_123 0 -1
Output:
1) "2024-04-20:Watering"

**To Delete Care Activity Log:**
LTRIM care_log_123 0

## 4. Reminders (Sorted Set):

**To Create a Reminder (Fertilize Plant):**
ZADD reminders user_1 4-25-2024 reminder_123
Here, 4-25-2025 is the timestamp for the next reminder.

**To Get/Read Upcoming Reminders for a User:**
ZREVRANGE reminders user_1 0 -1
This will return a list of reminders for user_1 sorted by their timestamps (the one with the closest timestamp will be at the top).

**To Update a Reminder (Change timestamp):**
Remove the old reminder:
ZREMRANGEBYSCORE reminders user_1 "-inf" "<old_timestamp>"
Add a new reminder with the updated timestamp:

ZADD reminders user_1 {new_timestamp} reminder_123

**To Delete a Reminder:**
ZREM reminders user_1 reminder_123

## 5. Events (Hash):

**To Create an Event (Flowering):**
HSET event_987 plant_id 123 event_type "Flowering" event_date "2024-04-19"

**To Get All Details About an Event:**
HGETALL event_987
Output:
plant_id: "123"
event_type: "Flowering"
event_date: "2024-04-19"

**To Update Event Date:**
HSET event_987 event_date "2024-04-20"  # Update event date to today

**To Delete an Event:**
HDEL event_987