Disaster Resource Management System

Submitted for Team Project (MC470303)
of
3rd Semester

Master of Computer Applications with specialization in Artificial Intelligence and Internet of Thing– CSE

Submitted by :

Rishi Kumar (2447031)
Prashant Kumar Mishra (2447021)
Satyam Bhardwaj (2447052)
Mayank Shaw (2447050)
Monil Mishra (2447017)
Biswajit Ghadei (2447036)

Submitted to:- Dr. Anand Shanker Tewari

Course Title :- Object Oriented Programming using JAVA

Department of Computer Science & Engineering

NATIONAL INSTITUTE OF TECHNOLOGY PATNA

University Campus, Bihar –

800005 July 2025 – Dec 2025

Introduction

When disasters strike, whether the devastating floods of Bihar or the catastrophic landslides in Uttarakhand, the immediate aftermath is a chaotic scramble for survival and aid. **The first 72 hours** are universally recognized as the "**golden window**" for rescue and relief, where every second counts. The efficiency of resource management during this period is very crucial.

This case study outlines the critical gaps observed in recent disasters in India. After that we have introduce a software solution for the Disaster Resource Management System designed to bridge these gaps by ensuring that the right resources reach the right people at the right time.

- **The Needs of the solution:**
    - The challenge is not one-off rescue but the **large-scale, efficient distribution of sustenance resources and a common operational picture**". A system that allows local officials to **update inventory** as they distribute relief and **automatically generate requests** when stocks are low would prevent shortages and ensure a **continuous supply chain** to the affected population.

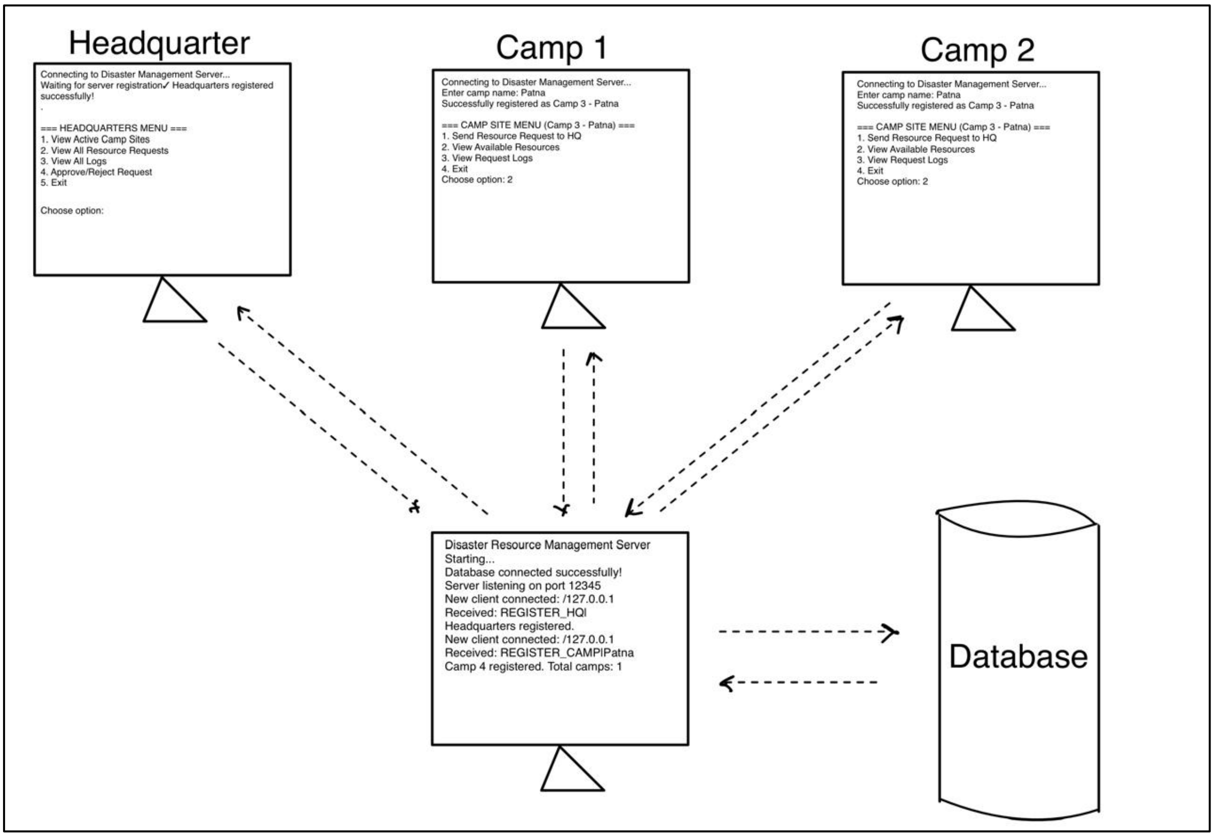
Actors

1. **Field Relief Camps & Local Authorities (The Requesters):**
    - **Who:** These are the front-line units like temporary camps that directly serving affected communities.
    - **How They Use the System**:
        - Log Local Inventory
        - Generate Automated Requests
        - Track Request Status
2. **Headquarters & Central Command Centres (The Approvers & Coordinators):**
    - **Who:** State and District Disaster Management Authorities (NDRF/SDRF).
    - **How They Use the System:**
        - View All Requests
        - Grant/Reject Requests
        - Monitor Live Inventory
        - Maintain Audit Logs

3. **Non-Governmental Organizations (NGOs) & Aid Agencies:**
   - **Who:** Organizations like the Red Cross, Oxfam, and local charities.
   - **How They Benefit:** They can be integrated as both field units (running camps) and resource providers. The system **prevents duplication of effort** .
4. **The Affected Population (The Ultimate Beneficiaries):**
   - **Who:** Individuals and families impacted by the disaster.
   - **How They Benefit:** While they may not directly use the software, **they are the ultimate stakeholders**.

<u>System Architecture</u>

Client-Server Model with a central server managing multiple concurrent clients using Java Sockets and multi-threading.



```
Connecting to Disaster Management Server...
Waiting for server registration✓ Headquarters registered
successfully!
.

=== HEADQUARTERS MENU ===
1. View Active Camp Sites
2. View All Resource Requests
3. View All Logs
4. Approve/Reject Request
5. Exit

Choose option:
```

```
Connecting to Disaster Management Server...
Enter camp name: Patna
Successfully registered as Camp 3 - Patna

=== CAMP SITE MENU (Camp 3 - Patna) ===
1. Send Resource Request to HQ
2. View Available Resources
3. View Request Logs
4. Exit
Choose option: 2
```

```
Connecting to Disaster Management Server...
Enter camp name: Patna
Successfully registered as Camp 3 - Patna

=== CAMP SITE MENU (Camp 3 - Patna) ===
1. Send Resource Request to HQ
2. View Available Resources
3. View Request Logs
4. Exit
Choose option: 2
```

```
Disaster Resource Management Server
Starting...
Database connected successfully!
Server listening on port 12345
New client connected: /127.0.0.1
Received: REGISTER_HQI
Headquarters registered.
New client connected: /127.0.0.1
Received: REGISTER_CAMPIPatna
Camp 4 registered. Total camps: 1
```

Fig 1 : System Architecture diagram for the DRMS

Workload Distribution

The workload is distributed as follows:

- **Server class Architecture -**
  - **Members:** Prashant Kumar Mishra
  - **Responsibilities:**
    1. Design and implementation of the central Server class.
    2. Managing multi-threading to handle concurrent clients.
- **Headquarter class Architecture -**
  - **Members:** Rishi Kumar
  - **Responsibilities:**
    1. Design and implementation of the Headquarter class.
    2. Managing camp request and activity log in real time.
- **Camp class Architecture**
  - **Members:** Satyam Bhardwaj, Mayank Shaw
  - **Responsibilities:**
    1. Designing and implementing the Camp class.
    2. Implementing the text-based menu-driven interface for Camp.
    3. Developing the logic for automatic request generation based on inventory thresholds.
- **Database Architecture**
  - **Members:** Biswajit Ghadei, Monil Mishra
  - **Responsibilities:**
    1. Implementing the database operations (MySql).
    2. Designing and creating the necessary database schema.

Transaction flow diagram

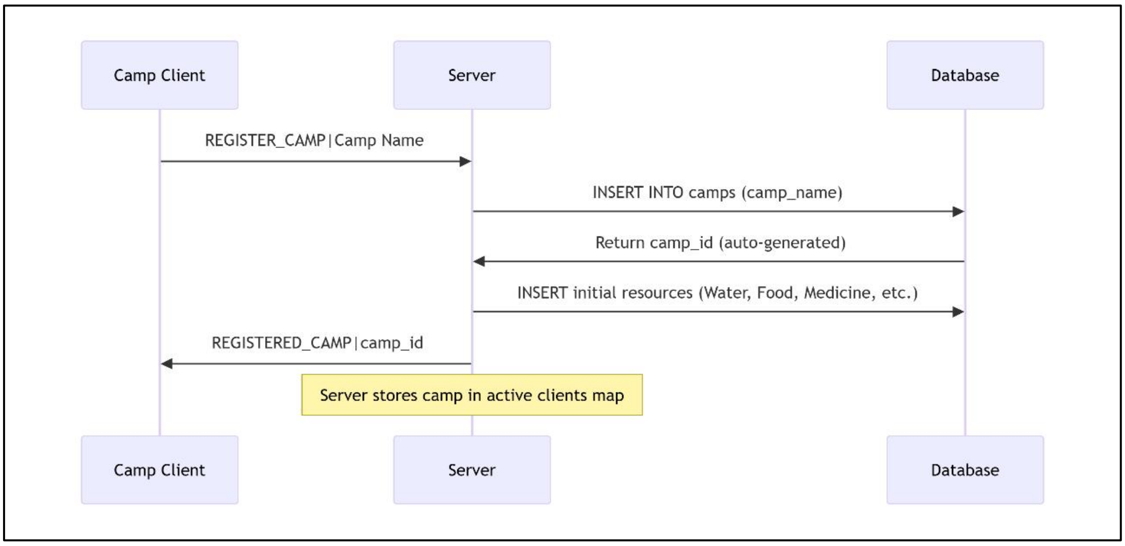

Camp Client
Server
Database
REGISTER_CAMP|Camp Name
INSERT INTO camps (camp_name)
Return camp_id (auto-generated)
INSERT initial resources (Water, Food, Medicine, etc.)
REGISTERED_CAMP|camp_id
Server stores camp in active clients map
Camp Client
Server
Database

Fig 2: Camp Registration & Initial Setup

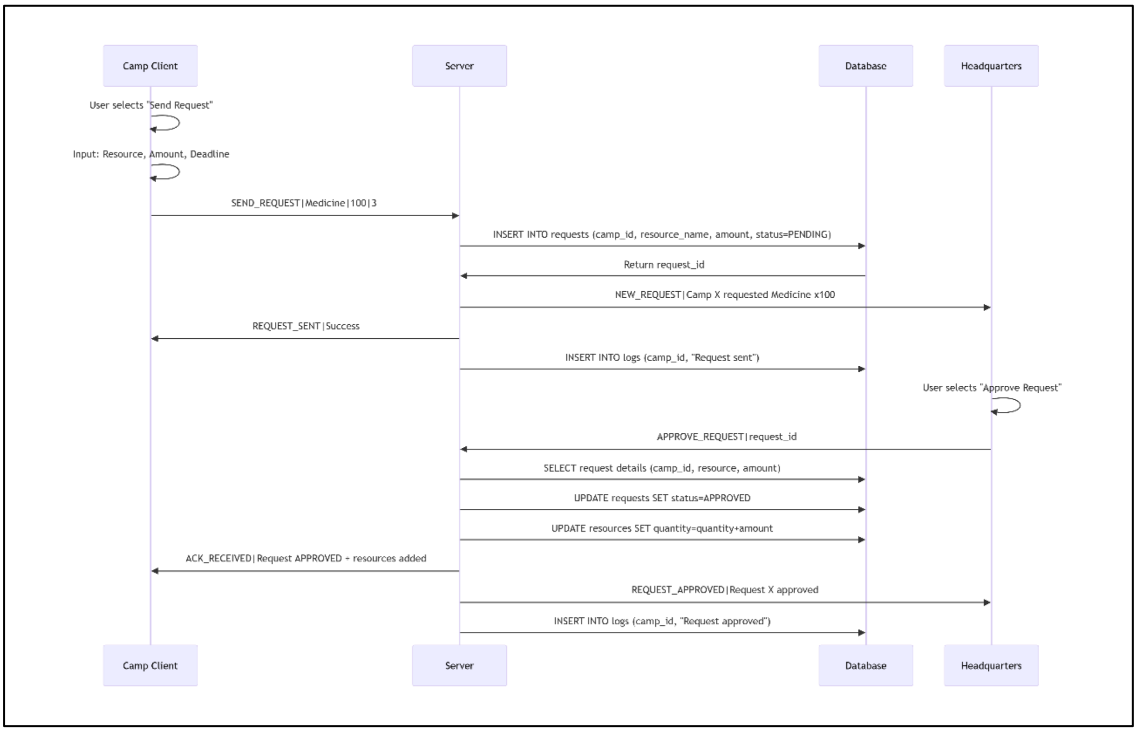

Camp Client
Server
Database
Headquarters
User selects "Send Request"
Input: Resource, Amount, Deadline
SEND_REQUEST|Medicine|100|3
INSERT INTO requests (camp_id, resource_name, amount, status=PENDING)
Return request_id
NEW_REQUEST|Camp X requested Medicine x100
REQUEST_SENT|Success
INSERT INTO logs (camp_id, "Request sent")
User selects "Approve Request"
APPROVE_REQUEST|request_id
SELECT request details (camp_id, resource, amount)
UPDATE requests SET status=APPROVED
UPDATE resources SET quantity=quantity+amount
ACK_RECEIVED|Request APPROVED - resources added
REQUEST_APPROVED|Request X approved
INSERT INTO logs (camp_id, "Request approved")
Camp Client
Server
Database
Headquarters

Fig 3: Resource Request Workflow

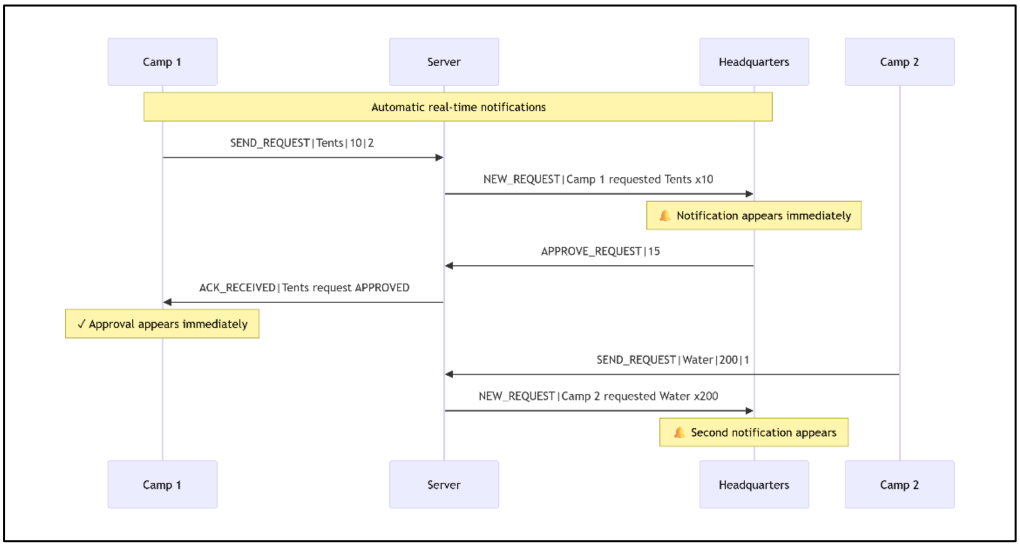

Camp 1
Server
Headquarters
Camp 2
Automatic real-time notifications
SEND_REQUEST|Tents|10|2
NEW_REQUEST|Camp 1 requested Tents x10
🔔 Notification appears immediately
APPROVE_REQUEST|15
ACK_RECEIVED|Tents request APPROVED
✓ Approval appears immediately
SEND_REQUEST|Water|200|1
NEW_REQUEST|Camp 2 requested Water x200
🔔 Second notification appears
Camp 1
Server
Headquarters
Camp 2

Fig 4: Real-time Notification System

Database Design

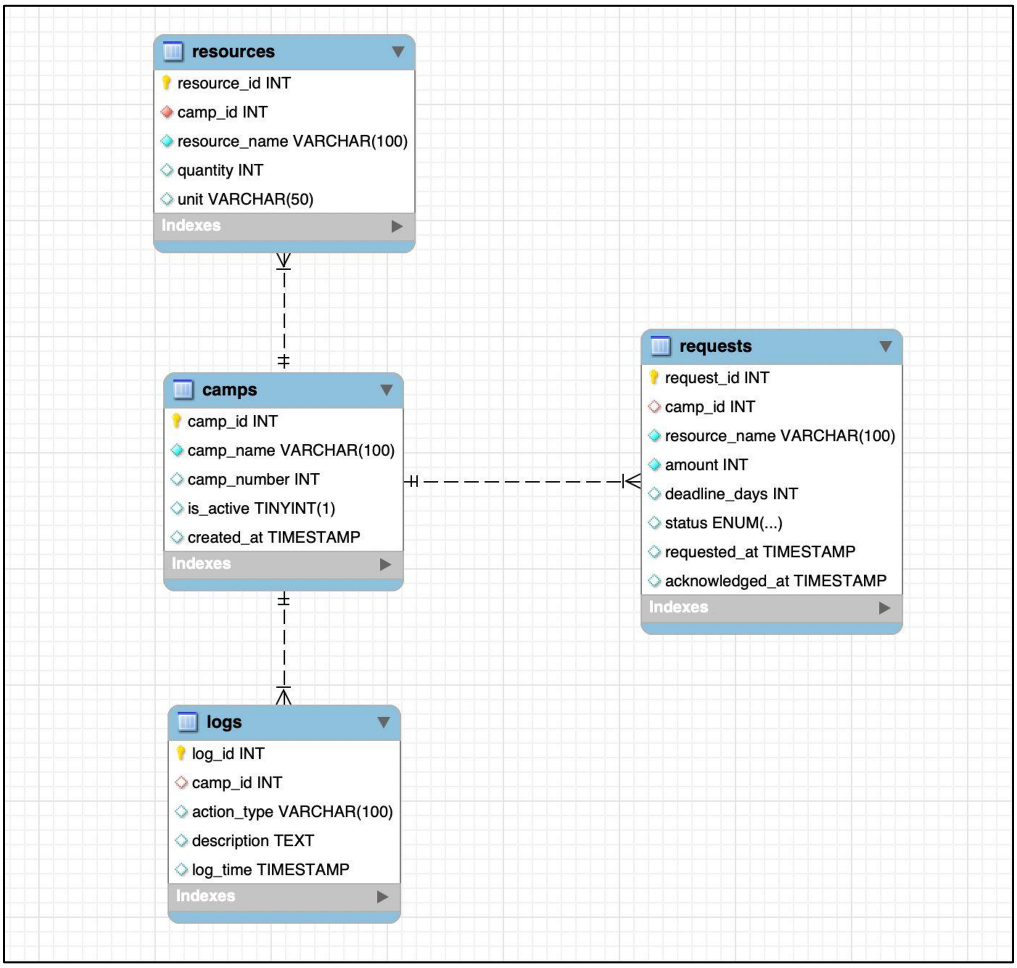
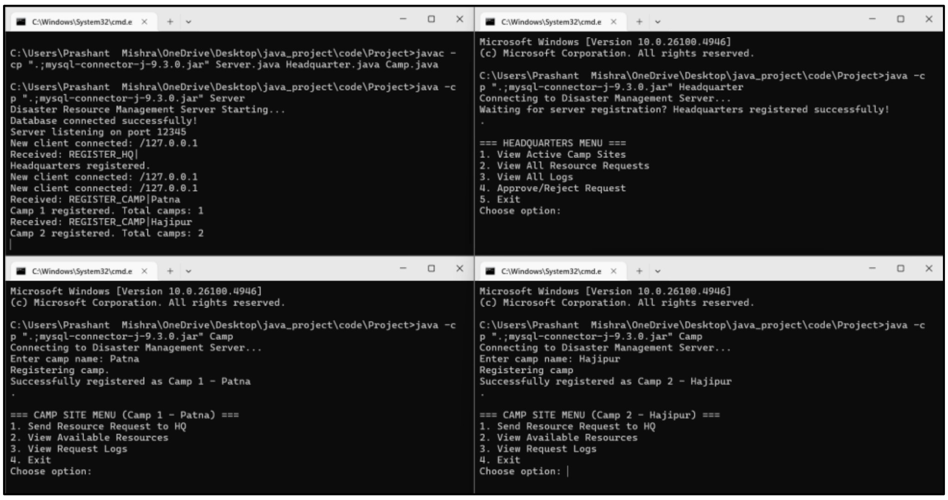


Fig 5: ER Diagram of the database

Output


C:\Windows\System32\cmd.e
C:\Windows\System32\cmd.e
C:\Windows\System32\cmd.e
C:\Windows\System32\cmd.e

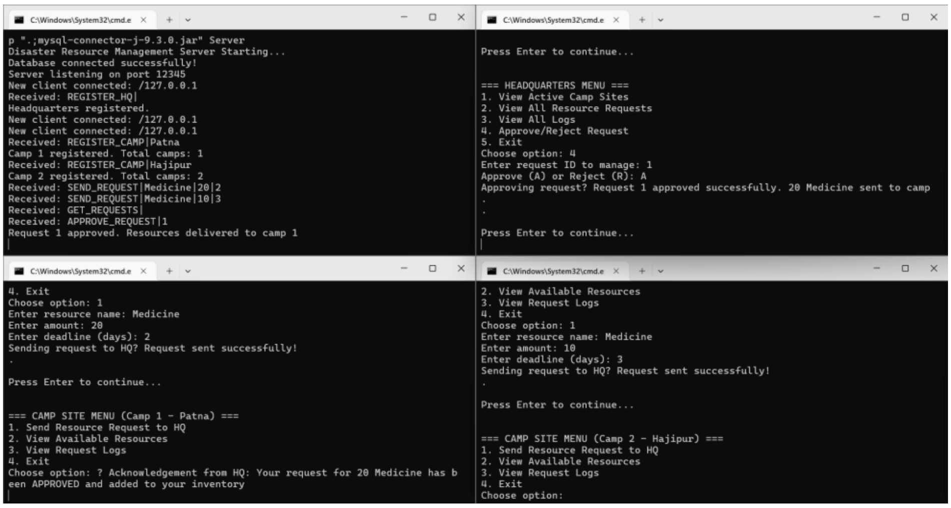
```
C:\Users\Prashant  Mishra\OneDrive\Desktop\java_project\code\Project>javac -
cp ".;mysql-connector-j-9.3.0.jar" Server.java Headquarter.java Camp.java

C:\Users\Prashant  Mishra\OneDrive\Desktop\java_project\code\Project>java -c
p ".;mysql-connector-j-9.3.0.jar" Server
Disaster Resource Management Server Starting...
Database connected successfully!
Server listening on port 12345
New client connected: /127.0.0.1
Received: REGISTER_HQ|
Headquarters registered.
New client connected: /127.0.0.1
New client connected: /127.0.0.1
Received: REGISTER_CAMP|Patna
Camp 1 registered. Total camps: 1
Received: REGISTER_CAMP|Hajipur
Camp 2 registered. Total camps: 2
```

```
Microsoft Windows [Version 10.0.26100.4946]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Prashant  Mishra\OneDrive\Desktop\java_project\code\Project>java -c
p ".;mysql-connector-j-9.3.0.jar" Headquarter
Connecting to Disaster Management Server...
Waiting for server registration? Headquarters registered successfully!
.

=== HEADQUARTERS MENU ===
1. View Active Camp Sites
2. View All Resource Requests
3. View All Logs
4. Approve/Reject Request
5. Exit
Choose option:
```

```
Microsoft Windows [Version 10.0.26100.4946]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Prashant  Mishra\OneDrive\Desktop\java_project\code\Project>java -c
p ".;mysql-connector-j-9.3.0.jar" Camp
Connecting to Disaster Management Server...
Enter camp name: Patna
Registering camp.
Successfully registered as Camp 1 - Patna
.

=== CAMP SITE MENU (Camp 1 - Patna) ===
1. Send Resource Request to HQ
2. View Available Resources
3. View Request Logs
4. Exit
Choose option:
```

```
Microsoft Windows [Version 10.0.26100.4946]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Prashant  Mishra\OneDrive\Desktop\java_project\code\Project>java -c
p ".;mysql-connector-j-9.3.0.jar" Camp
Connecting to Disaster Management Server...
Enter camp name: Hajipur
Registering camp
Successfully registered as Camp 2 - Hajipur
.

=== CAMP SITE MENU (Camp 2 - Hajipur) ===
1. Send Resource Request to HQ
2. View Available Resources
3. View Request Logs
4. Exit
Choose option:
```

Fig 6: Initial Server, Headquarter and Camp Initialization


C:\Windows\System32\cmd.e
C:\Windows\System32\cmd.e
C:\Windows\System32\cmd.e
C:\Windows\System32\cmd.e

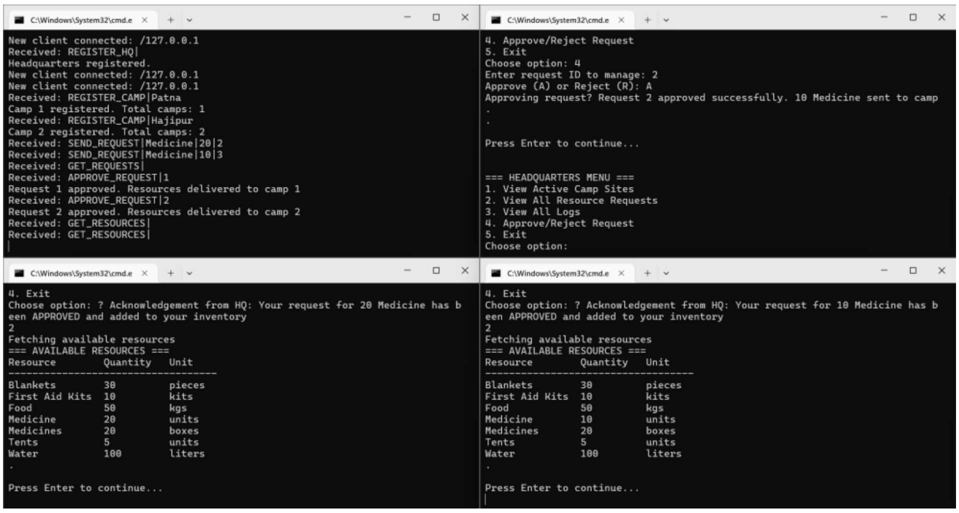
```
p ".;mysql-connector-j-9.3.0.jar" Server
Disaster Resource Management Server Starting...
Database connected successfully!
Server listening on port 12345
New client connected: /127.0.0.1
Received: REGISTER_HQ|
Headquarters registered.
New client connected: /127.0.0.1
New client connected: /127.0.0.1
Received: REGISTER_CAMP|Patna
Camp 1 registered. Total camps: 1
Received: REGISTER_CAMP|Hajipur
Camp 2 registered. Total camps: 2
Received: SEND_REQUEST|Medicine|20|2
Received: SEND_REQUEST|Medicine|10|3
Received: GET_REQUESTS|
Received: APPROVE_REQUEST|1
Request 1 approved. Resources delivered to camp 1
```

```

Press Enter to continue...


=== HEADQUARTERS MENU ===
1. View Active Camp Sites
2. View All Resource Requests
3. View All Logs
4. Approve/Reject Request
5. Exit
Choose option: 4
Enter request ID to manage: 1
Approve (A) or Reject (R): A
Approving request? Request 1 approved successfully. 20 Medicine sent to camp
.
.

Press Enter to continue...
```

```
4. Exit
Choose option: 1
Enter resource name: Medicine
Enter amount: 20
Enter deadline (days): 2
Sending request to HQ? Request sent successfully!
.

Press Enter to continue...


=== CAMP SITE MENU (Camp 1 - Patna) ===
1. Send Resource Request to HQ
2. View Available Resources
3. View Request Logs
4. Exit
Choose option: ? Acknowledgement from HQ: Your request for 20 Medicine has b
een APPROVED and added to your inventory
```

```
2. View Available Resources
3. View Request Logs
4. Exit
Choose option: 1
Enter resource name: Medicine
Enter amount: 10
Enter deadline (days): 3
Sending request to HQ? Request sent successfully!
.

Press Enter to continue...


=== CAMP SITE MENU (Camp 2 - Hajipur) ===
1. Send Resource Request to HQ
2. View Available Resources
3. View Request Logs
4. Exit
Choose option:
```

Fig 7: Request Initialization and Request approval


C:\Windows\System32\cmd.e
C:\Windows\System32\cmd.e
C:\Windows\System32\cmd.e
C:\Windows\System32\cmd.e

```
New client connected: /127.0.0.1
Received: REGISTER_HQ|
Headquarters registered.
New client connected: /127.0.0.1
New client connected: /127.0.0.1
Received: REGISTER_CAMP|Patna
Camp 1 registered. Total camps: 1
Received: REGISTER_CAMP|Hajipur
Camp 2 registered. Total camps: 2
Received: SEND_REQUEST|Medicine|20|2
Received: SEND_REQUEST|Medicine|10|3
Received: GET_REQUESTS|
Received: APPROVE_REQUEST|1
Request 1 approved. Resources delivered to camp 1
Received: APPROVE_REQUEST|2
Request 2 approved. Resources delivered to camp 2
Received: GET_RESOURCES|
Received: GET_RESOURCES|
```

```
4. Approve/Reject Request
5. Exit
Choose option: 4
Enter request ID to manage: 2
Approve (A) or Reject (R): A
Approving request? Request 2 approved successfully. 10 Medicine sent to camp
.
.

Press Enter to continue...


=== HEADQUARTERS MENU ===
1. View Active Camp Sites
2. View All Resource Requests
3. View All Logs
4. Approve/Reject Request
5. Exit
Choose option:
```

```
4. Exit
Choose option: ? Acknowledgement from HQ: Your request for 20 Medicine has b
een APPROVED and added to your inventory
2
Fetching available resources
=== AVAILABLE RESOURCES ===
Resource        Quantity   Unit
-----------------------------------
Blankets        30         pieces
First Aid Kits  10         kits
Food            50         kgs
Medicine        20         units
Medicines       20         boxes
Tents           5          units
Water           100        liters
.

Press Enter to continue...
```

```
4. Exit
Choose option: ? Acknowledgement from HQ: Your request for 10 Medicine has b
een APPROVED and added to your inventory
2
Fetching available resources
=== AVAILABLE RESOURCES ===
Resource        Quantity   Unit
-----------------------------------
Blankets        30         pieces
First Aid Kits  10         kits
Food            50         kgs
Medicine        10         units
Medicines       20         boxes
Tents           5          units
Water           100        liters
.

Press Enter to continue...
```

Fig 8: Resource monitoring