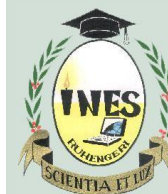


INSTITUT D'ENSEIGNEMENT SUPÉRIEUR DE RUHENGERI



Scientia et Lux

Accredited by Ministerial Order N°005/2010/MINÉDUC of 16 June 2010

25th February, 2025

Flood Emergency Response Advisor

Faculty: AFS

Department: Computer Science

Class: SWE A year3

Course: Artificial Intelligence

Group members:

| Name | RegNo |
|---|-----------------|
| ABARI ILIOR Aichetou | 23/21291 |
| CHOL Adut Gai | 23/21358 |
| NDAYISHIMIYE Abdul Aziz | 23/19390 |
| Uwera Gloriose | 23/20525 |
| ABDULAZEEZ Abubakar | 23/20989 |
| Muvandimwe Marie Divine (did not participate) | 23/20540 |

B.P. 155

Ruhengeri Rwanda

T : +250 788 90 30 30

F : +250 788 90 30 32

E : www.ines.ac.rw

Flood Emergency Response System Documentation- Day 2 Report

1. User Requirements:

- **Real-time Warnings** for heavy rain or rising water levels.
- **Evacuation Guidance based on flood severity:**
 - **RED ALERT** → **Immediate evacuation is recommended.**
 - **ORANGE ALERT** → **Be prepared for possible evacuation.**
- **Resource Distribution** post-flood (clean water, food, medicine).
- **Adaptive Rescue Operations** for road blockages.
- **Community Support** for isolated areas after infrastructure damage, by advising contact with emergency services (112 or 115) in case of severe infrastructure damage.

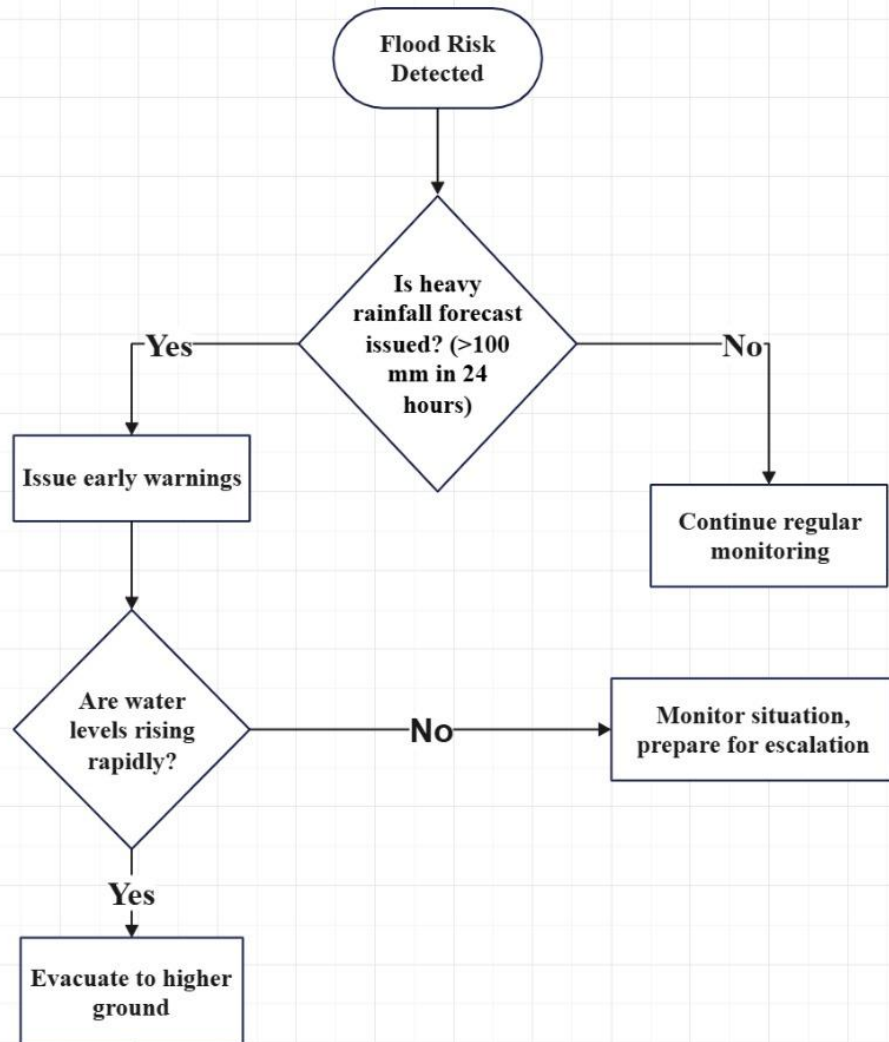
2. System Architecture

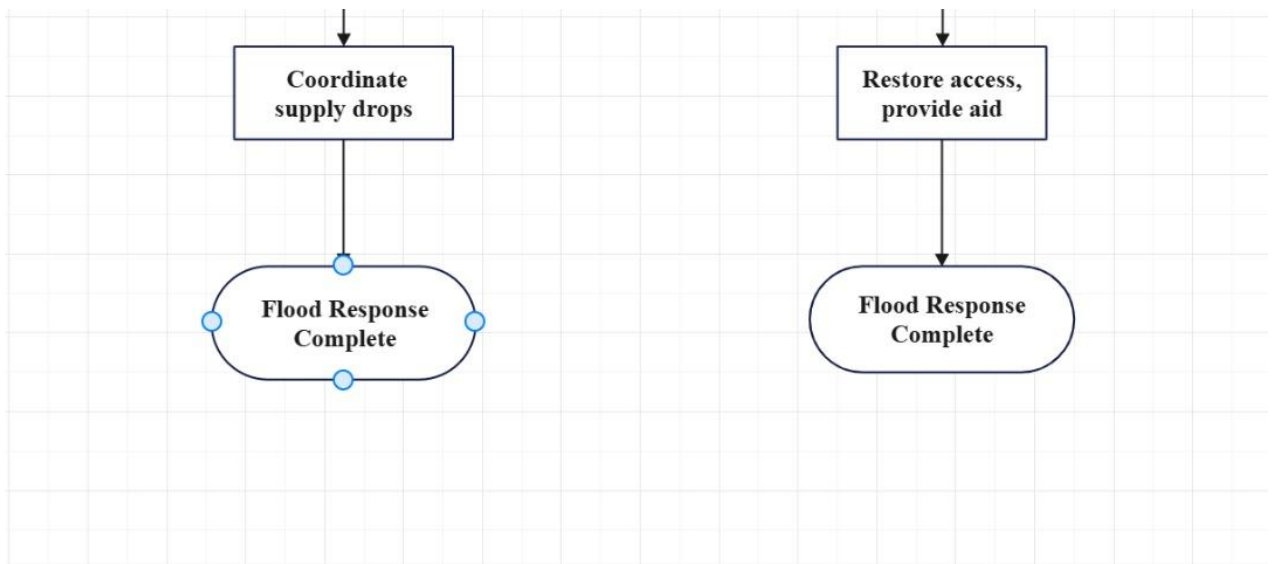
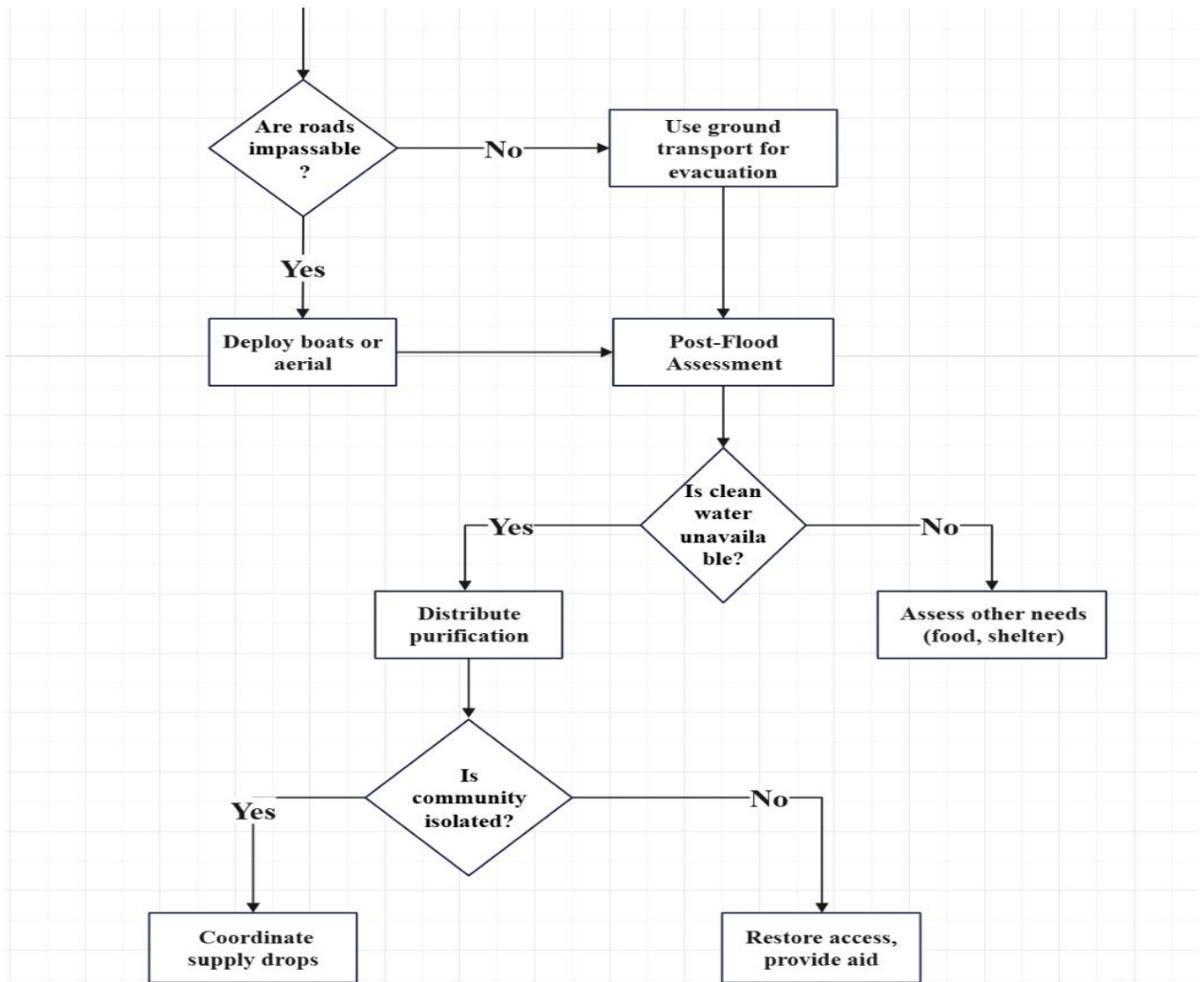
Updated System Architecture

Flowchart:

1. Start → Check Weather Forecast
2. IF heavy rainfall is predicted (>100mm) → Issue ORANGE ALERT (Be prepared for evacuation)
3. IF both heavy rainfall and rising water levels → Issue RED ALERT (Immediate evacuation recommended).
4. IF roads are flooded → Advise avoiding them and seeking alternative routes
5. IF roads are fully blocked → Deploy boats or aerial rescue
6. IF clean water is not available after the flood → Advise using purification tablets or boiling water
7. IF the community is isolated → Advise contacting emergency services (112 or 115)
8. End

Flood Emergency Response Advisor



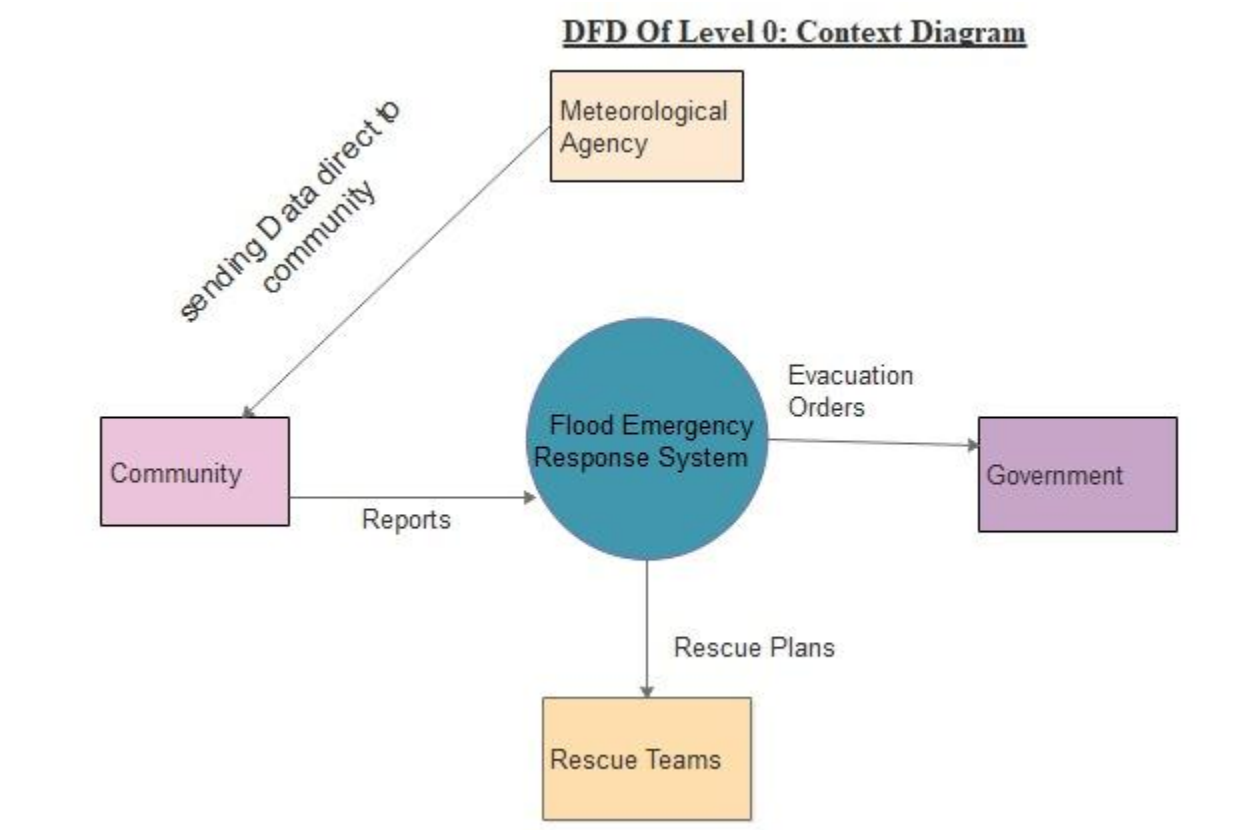


Data Flow Diagram (DFD):

Level 0: Context Diagram

- **External Entities:** Meteorological Agency, Communities, Government Agencies
- **System:** Flood Emergency Response Advisor
- **Outputs:** Warnings, Evacuation Orders, Resource Distribution

Level 1 DFD Screenshot:

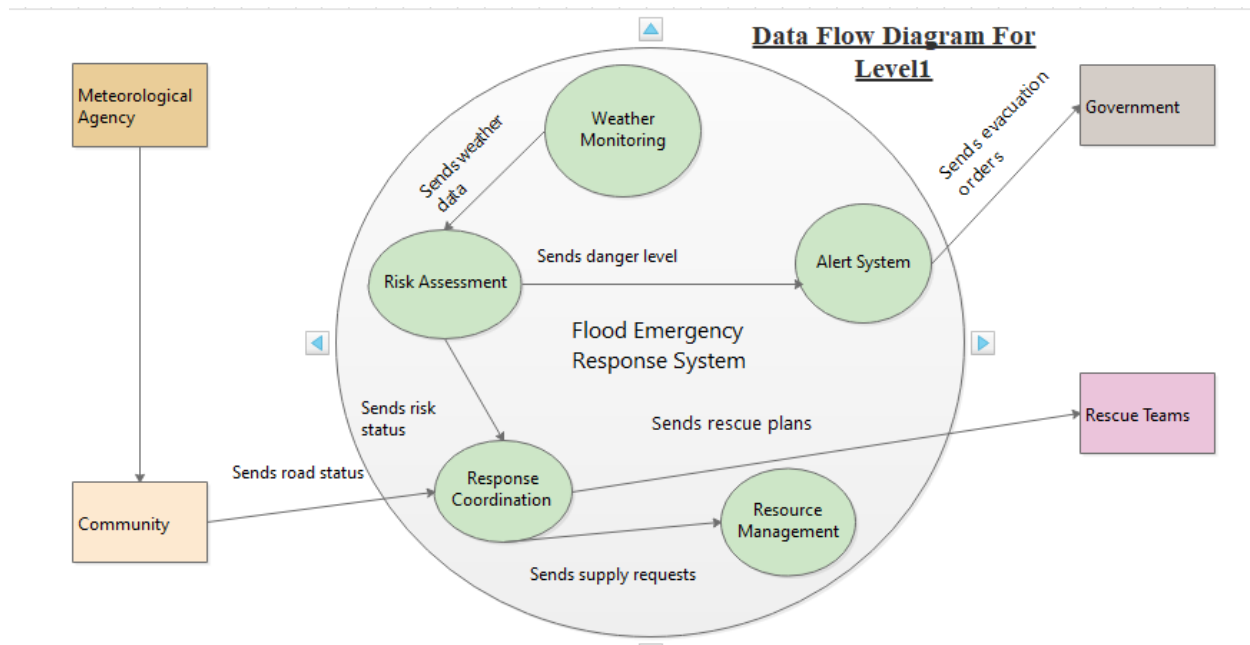


Level 1:

1. **Weather Monitoring:**
 - Input: Weather Forecasts
 - Process: Analyze Weather Data
 - Output: Risk Level, Warnings
2. **Community Alerts:**
 - Input: Community Reports
 - Process: Update Risk Status
 - Output: Dynamic Risk Assessment
3. **Response Coordination:**
 - Input: Risk Assessment, Road Conditions
 - Process: Decide and Deploy Appropriate Action

- Output: Evacuation Orders, Resource Deployment

Level 1 DFD Screenshot:



3. Initial Python Development

Rule-Based Logic

The initial Python implementation includes the following rules:

1. Heavy Rain Warning:

- **IF** heavy rainfall is forecast (rainfall > 100mm) **THEN** issue an **ORANGE ALERT** (be prepared for evacuation).

2. Evacuation Trigger:

- **IF** both **rainfall** and **water levels are rising** **THEN** issue a **RED ALERT** (immediate evacuation recommended).

3. Road Blockage:

- **IF** roads are flooded **THEN** advise residents to avoid them and seek alternative routes.

4. Clean Water Scarcity:

- **IF** clean water is not available **THEN** advise using purification tablets or boiling water.

5. Community Isolation

- **IF** the community is isolated **THEN** advise contacting emergency services (112 or 115).

Python Code Screenshot (Rules Implementation)

```
isolation = request.POST.get('isolation') == 'on'

message = ""
# Logic to determine the alert level

if rainfall and water_levels:
    message = "RED ALERT: Immediate evacuation is recommended! Seek higher ground.\n"
elif rainfall or water_levels:
    message = "ORANGE ALERT: Be prepared for evacuation and stay informed on weather updates.\n"
else:
    message = "GREEN ALERT: No immediate danger detected, stay informed.\n"

# Additional messages based on other conditions

if not clean_water:
    message += " Use purification tablets or boiled water.\n"

if roads:
    message += " Avoid flooded roads and seek alternative routes.\n"

if isolation:
    message += " Contact emergency services for assistance at '112' or '115'\n"

return JsonResponse({'message': message})

else:
    return JsonResponse({'error': 'Invalid request method'}, status=400)
```

In the Jupyter Notebook:

```
def flood_alert_system(rainfall, water_levels, clean_water, roads, isolation):
    if rainfall and water_levels:
        message = "🔴 RED ALERT: Immediate evacuation is recommended! Seek higher ground.\n"
    elif rainfall or water_levels:
        message = "🟡 ORANGE ALERT: Be prepared for evacuation and stay informed on weather updates.\n"
    else:
        message = "🟢 GREEN ALERT: No immediate danger detected, stay informed.\n"

    if not clean_water:
        message += "💧 Use purification tablets or boiled water.\n"
    if roads:
        message += "🛑 Avoid flooded roads and seek alternative routes.\n"
    if isolation:
        message += "📞 Contact emergency services for assistance at '112' or '115'\n"

    return message
```

4. Testing & Documentation

Test Cases & Results

| Test Scenario | Input | Expected Output |
|---------------------------|----------------------|--|
| Heavy Rain Forecast | Rainfall: 120mm | Warning: Heavy rainfall predicted! Alert communities and prepare for evacuation. |
| Rising Water Levels | Water Level: 95% | RED ALERT: Immediate evacuation is recommended! Seek higher ground. |
| Road Blockage | Road Status: Blocked | Avoid flooded roads and seek alternative routes. |
| Contaminated Water Supply | Water Supply: Dirty | Use purification tablets or boiled water. |
| Isolated community | Community isolated | Contact emergency services for assistance at '112' or '115' |

Test Results, see video