# BARANGAYLINK

Empowering Communities through Digital Connection and Compassionate Service

Caballero, Jireh Niel

Joven, Jowi Lorenz Nyl

Salvador, Paula Mae

Solas, Brian Kenneth


BSIT 3-B

# Rationale

In many barangays, residents experience barriers when trying to raise concerns or access services.

The traditional system requires residents to physically visit the barangay hall, leading to delays, inconvenience, and exclusion of vulnerable groups.

The Barangay Mobile App seeks to create a compassionate digital bridge between residents and officials by:

- Providing a safe and convenient way to report incidents and make service requests online.
- Delivering real-time updates to keep the community informed and empowered.
- Equipping barangay officials with organized tools for faster, transparent, and more humane service.

Goal: To strengthen local governance by improving community engagement, transparency, responsiveness, and care for residents' needs.

# Objectives

Streamline the barangay reporting and request process through an accessible mobile and web application.

Provide real-time notifications and status updates to residents about their reports and requests.

Foster transparency and accountability between barangay officials and their constituents.

Speed up the response and processing time for reports and requests to better serve the community.

Encourage greater resident participation in community safety, communication, and development efforts.

# Database Architecture

SalvaCorps utilizes a Client-Server Database Architecture, a widely adopted structure for modern applications.

**In a Client-Server setup:**

The client refers to the user's device (e.g., smartphone, tablet, or computer), which sends requests to access, submit, or retrieve data.

The server hosts the database and processes all incoming requests from clients. It retrieves, stores, and manages the data before sending back appropriate responses.

For instance, when a resident submits a report through the app, the client transmits the report details to the server. The server then processes the request, stores the information securely within the database, and responds with a confirmation or status update. Similarly, when barangay officials access the system to review submitted reports, their devices act as clients, interacting with the same centralized server.
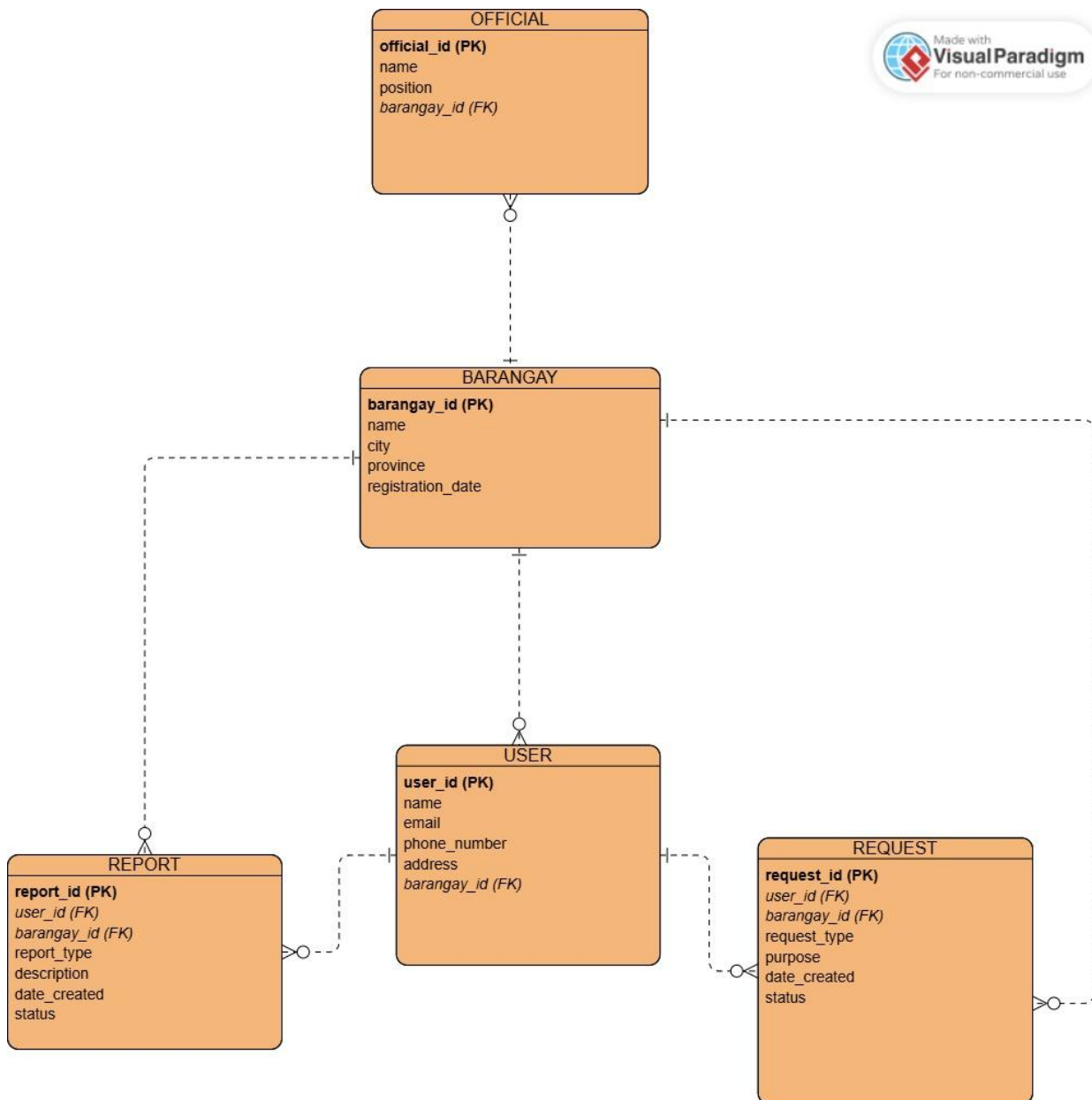
**This architecture allows:**

- Scalability: Multiple clients (residents and officials) can simultaneously use the system
- Centralized Management: Data is consistently maintained and managed from a single source.
- Security: Sensitive data is stored securely in one location rather than scattered across multiple devices.

For improved accessibility and scalability, the system can be further enhanced by adopting a Cloud Database Architecture, where the server is hosted on a secure cloud platform. This ensures that residents and officials can access the system anytime and anywhere via the internet, without relying on local infrastructure.

| Table Name | Description |
| --- | --- |
| User | Stores residents' profiles (name, email, phone number, address, barangay). |
| Barangay | Contains information about barangays registered in the system. |
| Official | Records the barangay officials like Captains, Kagawads, etc. |
| Report | Captures reports filed by residents (e.g., blotters, complaints). |
| Request | Records service requests made by residents (e.g., clearance, permits). |

# Entity Relationship Diagram

**OFFICIAL**
- **official_id (PK)**
- name
- position
- *barangay_id (FK)*

**BARANGAY**
- **barangay_id (PK)**
- name
- city
- province
- registration_date

**USER**
- **user_id (PK)**
- name
- email
- phone_number
- address
- *barangay_id (FK)*

**REPORT**
- **report_id (PK)**
- *user_id (FK)*
- *barangay_id (FK)*
- report_type
- description
- date_created
- status

**REQUEST**
- **request_id (PK)**
- *user_id (FK)*
- *barangay_id (FK)*
- request_type
- purpose
- date_created
- status

# Queries

--Q#1

INSERT INTO BARANGAY (name, city, province, registration_date) VALUES

 ('Barangay Uno', 'Iloilo City', 'Iloilo', '2023-01-15'),

('Barangay Dos', 'Iloilo City', 'Iloilo', '2023-02-10');

--Q#2

INSERT INTO USER (name, email, phone_number, address, barangay_id) VALUES

 ('Juan Dela Cruz', 'juan@example.com', '09171234567', 'Zone 1, Barangay Uno', 1),

 ('Maria Santos', 'maria@example.com', '09181234567', 'Zone 2, Barangay Dos', 2);

--Q#3

INSERT INTO OFFICIAL (name, position, barangay_id) VALUES

 ('Kap. Ricardo Reyes', 'Barangay Captain', 1),

('Kag. Liza Dizon', 'Kagawad', 1),

('Kap. Benito Ramos', 'Barangay Captain', 2);

--Q#4

INSERT INTO REPORT (user_id, barangay_id, report_type, description, date_created, status) VALUES

(1, 1, 'Garbage Issue', 'Garbage hasn't been collected for 3 days in Zone 1.', '2024-05-15', 'pending'),

 (2, 2, 'Street Light Problem', 'No street lights working in Zone 4.', '2024-05-16', 'resolved');

--Q#5

 INSERT INTO REQUEST (user_id, barangay_id, request_type, purpose, date_created, status)
VALUES

(1, 1, 'Barangay Clearance', 'Job application requirement.', '2024-05-17', 'pending'), (2, 2,
'Certificate of Indigency', 'For scholarship application.', '2024-05-18', 'approved');

--Q#6

SELECT

 R.report_id,

 U.name AS resident_name,

B.name AS barangay_name, R.report_type,

 R.description,

 R.date_created,

 R.status

FROM REPORT R

JOIN USER U ON R.user_id = U.user_id

JOIN BARANGAY B ON R.barangay_id = B.barangay_id

ORDER BY R.date_created DESC;

```sql
--Q#7

 SELECT

RE.request_id,

U.name AS resident_name,

B.name AS barangay_name,

 RE.request_type,

 RE.purpose,

RE.date_created,

RE.status

FROM REQUEST RE

JOIN USER U ON RE.user_id = U.user_id

JOIN BARANGAY B ON RE.barangay_id = B.barangay_id

ORDER BY RE.date_created DESC;

--Q#8

SELECT

report_id,

report_type,

 description,

date_created

FROM REPORT WHERE status = 'pending'

 ORDER BY date_created DESC;
```

```sql
--Q#9

SELECT

 request_id,

 request_type,

purpose,

date_created

FROM REQUEST WHERE status = 'pending'

ORDER BY date_created DESC;


 --Q#10

 SELECT

B.name AS barangay_name,

COUNT(R.report_id) AS total_reports

FROM BARANGAY B

 LEFT JOIN REPORT R ON B.barangay_id = R.barangay_id

 GROUP BY B.barangay_id;


--Q#11

 SELECT

B.name AS barangay_name,

 COUNT(RE.request_id) AS total_requests

FROM BARANGAY B

 LEFT JOIN REQUEST RE ON B.barangay_id = RE.barangay_id
```

```sql
    GROUP BY B.barangay_id;


--Q#12

SELECT

 status,

COUNT(*) AS total

 FROM REPORT

 GROUP BY status;


--Q#13

SELECT

 report_id,

report_type,

 date_created,

status

 FROM REPORT

WHERE status = 'pending'

AND date_created >= CURDATE() - INTERVAL 7 DAY;
```

```sql
--Q#14

 SELECT

 request_id,

request_type,

 purpose,

date_created

FROM REQUEST

 WHERE barangay_id = 1 AND status = 'pending';


--Q#15

SELECT

U.user_id,

U.name,

COUNT(R.report_id) AS total_reports

 FROM USER U

 JOIN REPORT R ON U.user_id = R.user_id

 GROUP BY U.user_id

HAVING total_reports > 0;
```

```sql
--Q#16

SELECT

U.user_id,

U.name,

COUNT(RE.request_id) AS total_requests

 FROM USER U

 JOIN REQUEST RE ON U. user_id = RE.user_id

GROUP BY U.user_id

HAVING total_requests > 0;


 --Q#17

SELECT

 U.user_id,

 U.name,

COUNT(DISTINCT R.report_id) AS reports_count,

COUNT(DISTINCT RE.request_id) AS requests_count

 FROM USER U

LEFT JOIN REPORT R ON U.user_id = R.user_id

LEFT JOIN REQUEST RE ON U.user_id = RE.user_id

GROUP BY U.user_id;
```

# DB-FIDDLE

https://www.db-fiddle.com/f/5m9grngDMHxPLMbTsKhLUr/0