

Project Management Plan

Week 1: Investigation and Database Design

Objective: Complete theoretical foundation and database architecture

Deliverables:

- Research summary with 2 referenced articles
- Entity identification and relationship analysis
- ERD using Crow's Foot notation (minimum 5 tables)
- Normalisation documentation (0NF to 3NF)
- Data dictionary for all tables
- Australian Privacy Principles research

Week 2: Database Implementation and Basic Application

Objective: Create functional database with basic web interface

Deliverables:

- SQL CREATE TABLE statements for all tables
- INSERT statements with 15-20 sample records
- Python Flask application setup
- Basic CRUD operations (Create volunteer, Read organisations, Update contact details, Delete events)
- ACID properties documentation
- SQLite database file (.db)

Week 3: Advanced Queries, Evaluation, and Reporting

Objective: Complete advanced features and comprehensive evaluation

Deliverables:

- Advanced SQL queries with JOINS, aggregates, and calculated fields
- Skill-based volunteer search functionality
- Multi-table queries (3+ tables)
- Data quality and cleaning procedures
- Final comprehensive report
- Ethical, legal, and security analysis (APP compliance)

Investigation

Article 1 Summary: Volunteer Management Challenges

Source: Oho. (2024). Volunteer Management: the Challenges of Recruitment and Retention. Retrieved from

<https://weareoho.com/2024/04/26/volunteer-management-the-challenges-of-recruitment-and-retention/>

Key Takeaways

Scale and Significance: Over 5 million Australians volunteer through organisations each year, representing almost one quarter (24.8%) of the population 15 and over. This demonstrates the massive scale and importance of volunteer management in Australia.

Primary Challenge: In 2023, volunteer recruitment emerged as the top challenge according to volunteer managers (Volunteering Australia). This directly supports the need for better volunteer coordination systems like the Community Connect platform.

Retention Crisis: The article reveals a concerning trend: from 2018 to 2021, the global volunteering workforce lost 44 percent of volunteers. This massive turnover creates a significant administrative burden for organisations.

The Volunteer Lifecycle: The research identifies five critical stages: attraction, recruitment, engagement/retention, recognition, and retirement, with each stage presenting unique challenges requiring tailored strategies.

Technology Solutions: Software systems are constantly evolving and have become more crucial to the recruitment, engagement and retention of volunteers. A volunteer management tool can help save time, reduce manual effort, provide data insights and help build your volunteer relationships.

Relevance to Community Connect: This article directly validates the need for a database-driven platform and highlights the specific challenges (recruitment, retention, administrative burden) that the Community Connect system aims to address.

Article 2 Summary: Importance of Community Engagement

Source: Granicus. (2024). Why is Community Engagement Important? Retrieved from <https://granicus.com/blog/why-is-community-engagement-important/>

Key Takeaways

Definition and Purpose: Community engagement is important as it takes action to influence stakeholders with government, political or funding power to implement public projects and policies that primarily benefit individual communities and drive social change.

Democratic Participation: The article emphasises that people should have a say over decisions that impact their everyday lives, supporting the democratic principle underlying volunteer coordination platforms.

Social Capital Creation: Community engagement is primarily part of a dialogue where organisations and communities can make decisions to create social capital. This directly relates to how volunteer platforms build connections between organisations and community members.

Empowerment and Inclusion: By including diverse voices, usually marginalised or overlooked voices are actively empowered within their community to participate in decision making that affects their everyday lives.

Government Efficiency: Community engagement helps governments improve the efficiency, legitimacy and transparency of their decision making when community input is incorporated into planning processes.

Growing Trend: Since the 1990s, there has been a rapid expansion in formal, state-based initiatives to facilitate public participation in decision making, where communities are invited to engage beyond voting.

Relevance to Community Connect: This article provides the theoretical foundation for why volunteer coordination platforms are essential - they facilitate community engagement, create social capital, and enable democratic participation in community improvement initiatives.

Deconstruction

Primary Entities

VOLUNTEER

- Represents individuals who want to volunteer
- Key attributes: volunteer_id (PK), first_name, last_name, date_of_birth, email, phone, address, registration_date

ORGANISATION

- Represents groups that need volunteer assistance
- Key attributes: org_id (PK), org_name, contact_email, phone, address, org_type, description

EVENT

- Represents specific volunteering opportunities
- Key attributes: event_id (PK), event_name, description, start_date, end_date, location, max_volunteers, organisation_id (FK)

SKILL

- Repository of volunteer capabilities and event requirements
- Key attributes: skill_id (PK), skill_name, skill_description, skill_category

Junction Tables (Resolving M:N Relationships)

VOLUNTEER_EVENT

- Links volunteers to events they've registered for
- Key attributes: volunteer_id (FK), event_id (FK), registration_date, attendance_status

VOLUNTEER_SKILL

- Links volunteers to their possessed skills
- Key attributes: volunteer_id (FK), skill_id (FK), proficiency_level, years_experience

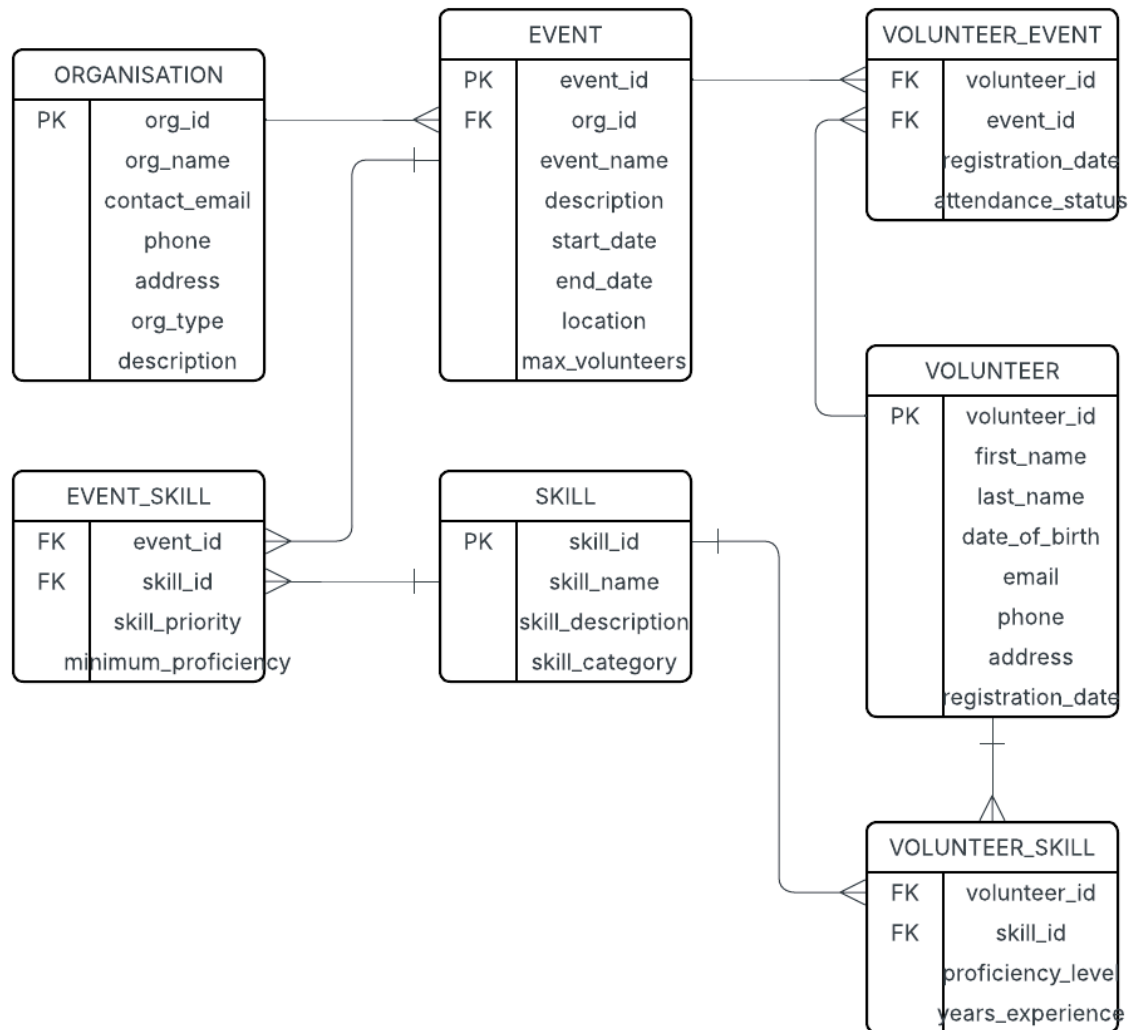
EVENT_SKILL

- Links events to required skills
- Key attributes: event_id (FK), skill_id (FK), skill_priority, minimum_proficiency

Key Relationships

1. organisations → Events: One-to-Many
 - One organisation can post multiple events
2. Volunteers ↔ Events: Many-to-Many (via VOLUNTEER_EVENT)
 - Volunteers can register for multiple events
 - Events can have multiple volunteers
3. Volunteers ↔ Skills: Many-to-Many (via VOLUNTEER_SKILL)
 - Volunteers can possess multiple skills
 - Skills can be held by multiple volunteers
4. Events ↔ Skills: Many-to-Many (via EVENT_SKILL)
 - Events can require multiple skills
 - Skills can be needed for multiple events

Entity Relationship Diagram (ERD)



Normalisation

Starting Point: Unnormalised Table (0NF)

The Community Connect system initially might store all data in a single flat file structure:

VOLUNTEER_REGISTRATION_DATA

volunteer_id, volunteer_name, volunteer_email, volunteer_phone, volunteer_dob,
volunteer_address, organisation_name, org_contact_person, org_email, org_phone,
org_address, org_type, event_name, event_description, event_date, event_location,
event_max_volunteers, skills_list, skill_levels, registration_date, attendance_status

Sample Data:

volunteer_id	volunteer_name	volunteer_email	organisation_name	event_name	event_date	skills_list	skill_levels
1	John Smith	john@email.com	Red Cross	Blood Drive	2025-09-15	First Aid, CPR, Communication	Expert, Intermediate, Beginner
1	John Smith	john@email.com	Community Kitchen	Food Service	2025-09-20	First Aid, Food Handling	Expert, Beginner

Problems in 0NF:

- Repeating groups (skills_list, skill_levels)
- Data redundancy (volunteer details repeated)
- Update anomalies
- Insertion and deletion anomalies

First Normal Form (1NF)

Rule: Eliminate repeating groups and ensure atomic values.

Solution: Split the skills into separate rows and ensure each cell contains only atomic values.

VOLUNTEER_REGISTRATION_DATA

volunteer_id, volunteer_name, volunteer_email, volunteer_phone, volunteer_dob,
volunteer_address, organisation_name, org_contact_person, org_email, org_phone,
org_address, org_type, event_name, event_description, event_date, event_location,
event_max_volunteers, skill_name, skill_level, registration_date, attendance_status

Sample Data:

volunteer_id	volunteer_name	volunteer_email	organisation_name	event_name	skill_name	skill_level
1	John Smith	john@email.com	Red Cross	Blood Drive	First Aid	Expert
1	John Smith	john@email.com	Red Cross	Blood Drive	CPR	Intermediate
1	John Smith	john@email.com	Community Kitchen	Food Service	First Aid	Expert
1	John Smith	john@email.com	Community Kitchen	Food Service	Food Handling	Beginner

Achievement: No repeating groups, all attributes are atomic.

Second Normal Form (2NF)

Rule: Must be in 1NF AND all non-key attributes must be fully functionally dependent on the entire primary key.

Analysis: The composite key would be (volunteer_id, event_name, skill_name). However:

- volunteer_name depends only on volunteer_id
- organisation_name depends only on event_name
- skill_level depends on volunteer_id + skill_name

Solution: Decompose into multiple tables.

- VOLUNTEER (volunteer_id (PK), volunteer_name, volunteer_email, volunteer_phone, volunteer_dob, volunteer_address)
- ORGANISATION (organisation_id (PK), organisation_name, org_contact_person, org_email, org_phone, org_address, org_type)
- EVENT (event_id (PK), event_name, event_description, event_date, event_location, event_max_volunteers, organisation_id (FK))
- SKILL (skill_id (PK), skill_name)
- VOLUNTEER_SKILL (volunteer_id (FK), skill_id (FK), skill_level)
- VOLUNTEER_EVENT (volunteer_id (FK), event_id (FK), registration_date, attendance_status)

Achievement: All non-key attributes are fully functionally dependent on their respective primary keys.

Third Normal Form (3NF)

Rule: Must be in 2NF AND no transitive dependencies (non-key attributes should not depend on other non-key attributes).

Analysis: Examining for transitive dependencies:

- In VOLUNTEER: No transitive dependencies detected
- In ORGANISATION: No transitive dependencies detected
- In EVENT: No transitive dependencies detected
- In SKILL: Could add skill_category, but skill_description doesn't depend on skill_name transitively
- In junction tables: No transitive dependencies

Refined Solution (3NF):

- VOLUNTEER (volunteer_id, first_name, last_name, email, phone, date_of_birth, address, registration_date)
- ORGANISATION (organisation_id, organisation_name, contact_person, email, phone, address, organisation_type, description)
- EVENT (event_id, event_name, description, start_date, end_date, location, max_volunteers, organisation_id)
- SKILL (skill_id, skill_name, skill_description, skill_category)
- VOLUNTEER_SKILL (volunteer_id, skill_id, proficiency_level, years_experience)
- VOLUNTEER_EVENT (volunteer_id, event_id, registration_date, attendance_status)
- EVENT_SKILL (event_id, skill_id, skill_priority, minimum_proficiency)

Achievement: All tables are in 3NF with no transitive dependencies, ensuring data integrity and minimal redundancy.

Benefits of Normalisation

- Eliminated Data Redundancy: Volunteer and organisation details stored once
- Resolved Update Anomalies: Changes to volunteer details only require one update
- Prevented Insertion Anomalies: Can add skills without requiring event registration
- Avoided Deletion Anomalies: Deleting an event registration doesn't lose volunteer data
- Improved Data Integrity: Referential integrity enforced through foreign keys
- Enhanced Flexibility: System can easily accommodate new requirements and relationships

Data Dictionary

VOLUNTEER Table

Purpose: Stores information about individuals who register to volunteer

Attribute	Data Type	Length	Constraints	Description
volunteer_id	INTEGER	-	PRIMARY KEY, NOT NULL, AUTO INCREMENT	Unique identifier for each volunteer
first_name	TEXT	50	NOT NULL	Volunteer's first name
last_name	TEXT	50	NOT NULL	Volunteer's surname
date_of_birth	DATE	-	NOT NULL	Volunteer's birth date for age verification
email	TEXT	100	NOT NULL, UNIQUE	Contact email address
phone	TEXT	15	NOT NULL	Contact phone number
address	TEXT	200	NOT NULL	Residential address
registration_date	DATE	-	NOT NULL, DEFAULT CURRENT_DATE	Date volunteer registered on platform

ORGANISATION Table

Purpose: Stores information about organisations that post volunteer opportunities

Attribute	Data Type	Length	Constraints	Description
org_id	INTEGER	-	PRIMARY KEY, NOT NULL, AUTO INCREMENT	Unique identifier for each organisation
org_name	TEXT	100	NOT NULL	Official organisation name
contact_email	TEXT	100	NOT NULL	Primary contact email
phone	TEXT	15	NOT NULL	Contact phone number
address	TEXT	200	NOT NULL	Organisation's address
org_type	TEXT	50	NOT NULL	Type of organisation (e.g., Charity, Community Group)
description	TEXT	500	-	Brief description of organisation's mission

EVENT Table

Purpose: Stores volunteer events/opportunities posted by organisations

Attribute	Data Type	Length	Constraints	Description
event_id	INTEGER	-	PRIMARY KEY, NOT NULL, AUTO INCREMENT	Unique identifier for each event
org_id	INTEGER	-	FOREIGN KEY, NOT NULL	References ORGANISATION(org_id)
event_name	TEXT	100	NOT NULL	Name/title of the volunteer event
description	TEXT	1000	NOT NULL	Detailed description of the event
start_date	DATE	-	NOT NULL	Event start date
end_date	DATE	-	NOT NULL	Event end date
location	TEXT	200	NOT NULL	Event venue/location
max_volunteers	INTEGER	-	NOT NULL, CHECK (max_volunteers > 0)	Maximum number of volunteers needed

Referential Integrity:

- org_id REFERENCES ORGANISATION(org_id) ON DELETE CASCADE

SKILL Table

Purpose: Master table of all volunteer skills and capabilities

Attribute	Data Type	Length	Constraints	Description
skill_id	INTEGER	-	PRIMARY KEY, NOT NULL, AUTO INCREMENT	Unique identifier for each skill
skill_name	TEXT	50	NOT NULL, UNIQUE	Name of the skill
skill_description	TEXT	200	-	Description of what the skill involves
skill_category	TEXT	50	NOT NULL	Category grouping (e.g., Medical, Administrative, Physical)

VOLUNTEER_EVENT Table

Purpose: Junction table linking volunteers to events they've registered for (resolves M:N relationship)

Attribute	Data Type	Length	Constraints	Description
volunteer_id	INTEGER	-	FOREIGN KEY, NOT NULL	References VOLUNTEER(volunteer_id)
event_id	INTEGER	-	FOREIGN KEY, NOT NULL	References EVENT(event_id)
registration_date	DATE	-	NOT NULL, DEFAULT CURRENT_DATE	Date volunteer registered for event
attendance_status	TEXT	20	NOT NULL, DEFAULT 'Registered'	Status: Registered, Attended, No-Show, Cancelled

Composite Primary Key: (volunteer_id, event_id)

Referential Integrity:

- volunteer_id REFERENCES VOLUNTEER(volunteer_id) ON DELETE CASCADE
- event_id REFERENCES EVENT(event_id) ON DELETE CASCADE

VOLUNTEER_SKILL Table

Purpose: Junction table linking volunteers to their skills (resolves M:N relationship)

Attribute	Data Type	Length	Constraints	Description
volunteer_id	INTEGER	-	FOREIGN KEY, NOT NULL	References VOLUNTEER(volunteer_id)
skill_id	INTEGER	-	FOREIGN KEY, NOT NULL	References SKILL(skill_id)
proficiency_level	TEXT	20	NOT NULL	Skill level: Beginner, Intermediate, Advanced, Expert
years_experience	INTEGER	-	CHECK (years_experience >= 0)	Years of experience with this skill

Composite Primary Key: (volunteer_id, skill_id)

Referential Integrity:

- volunteer_id REFERENCES VOLUNTEER(volunteer_id) ON DELETE CASCADE
- skill_id REFERENCES SKILL(skill_id) ON DELETE CASCADE

EVENT_SKILL Table

Purpose: Junction table linking events to required skills (resolves M:N relationship)

Attribute	Data Type	Length	Constraints	Description
event_id	INTEGER	-	FOREIGN KEY, NOT NULL	References EVENT(event_id)
skill_id	INTEGER	-	FOREIGN KEY, NOT NULL	References SKILL(skill_id)
skill_priority	TEXT	20	NOT NULL	Priority level: Essential, Preferred, Bonus
minimum_proficiency	TEXT	20	NOT NULL	Minimum required proficiency level

Composite Primary Key: (event_id, skill_id)

Referential Integrity:

- event_id REFERENCES EVENT(event_id) ON DELETE CASCADE
- skill_id REFERENCES SKILL(skill_id) ON DELETE CASCADE

Database Design Summary

Total Tables: 7 (4 entity tables + 3 junction tables) Total Relationships: 6 relationships (3 one-to-many, 3 many-to-many resolved) Normalization Level: Third Normal Form (3NF)

Key Design Decisions:

- All M:N relationships properly resolved with junction tables
- Composite primary keys used for junction tables
- Referential integrity enforced with foreign key constraints
- CHECK constraints ensure data validity
- DEFAULT values provided where appropriate
- ON DELETE CASCADE maintains data consistency