MMP Database Improvements

* **FAQ Section**
  + Agreed to not store questions in a database
* **Implementing the theme selection (Add to elderly\_details table)**
  + Colour theme ((Validation: Must be 1, 2, 3 or 4)
    - INT(1)
  + Text size (Validation: Must be ('Extra Large', 'Large', 'Medium', 'Small'))
    - ENUM('Extra Large', 'Large', 'Medium', 'Small')

ALTER TABLE elderly\_details

ADD COLUMN colour\_theme CHAR(1) CHECK (colour\_theme IN (1, 2, 3, 4)),

` ADD COLUMN text\_size ENUM('Extra Large', 'Large', 'Medium', 'Small');

A screenshot of a computer

Description automatically generated

**Updated statement to add registration details, now including theme/font size**

INSERT INTO elderly\_details (first\_name, last\_name, date\_of\_birth, phone\_number, email, carer\_id, emergency\_contact\_id, colour\_theme, text\_size)

VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?);

**Statement to update theme**

UPDATE elderly\_details

SET colour\_theme = ?

WHERE elderly\_id = ?;

**Statement to update text size**

UPDATE elderly\_details

SET text\_size = ?

WHERE elderly\_id = ?;

**Statement to update theme and textsize**

UPDATE elderly\_details

SET colour\_theme = ?, text\_size = ?

WHERE elderly\_id = ?;

**Statement to reset theme and textsize**

UPDATE elderly\_details

SET colour\_theme = NULL, text\_size = NULL

WHERE elderly\_id = ?;

**Statement to get user’s textsize**

SELECT text\_size

FROM elderly\_details

WHERE elderly\_id = ?;

**Statement to get user’s theme choice**

SELECT colour\_theme

FROM elderly\_details

WHERE elderly\_id = ?;

**Statement to get user’s text size and theme choice**

SELECT text\_size, colour\_theme

FROM elderly\_details

WHERE elderly\_id = ?;

* **Contact us form (Table Name: contact\_us)**
  + Query ID (Primary Key, auto increment)
    - PRIMARY KEY (AUTO-INCREMENT)
  + First Name (Validation: 20 characters or less)
    - VARCHAR(20)
  + Last Name (Validation: 20 characters or less)
    - VARCHAR(20)
  + Phone Number (Validation: Must be 11 numbers)
    - CHAR(11)
  + Email Address (Validation: 100 characters or less)
    - VARCHAR(100)
  + Message (Validation: 1000 characters or less)
    - VARCHAR(1000)
  + Responded to (Validation: Must be True/False, is automatically set to false)
    - BOOLEAN

CREATE TABLE contact\_us (

query\_id INT AUTO\_INCREMENT PRIMARY KEY,

first\_name VARCHAR(20) NOT NULL,

last\_name VARCHAR(20) NOT NULL,

phone\_number CHAR(11) NOT NULL,

email VARCHAR(100) NOT NULL,

message VARCHAR(1000) NOT NULL,

responded\_to BOOLEAN DEFAULT FALSE

);

A screenshot of a computer

Description automatically generated

**Statement to INSERT a query**

INSERT INTO contact\_us (first\_name, last\_name, phone\_number, email, message)

VALUES (?, ?, ?, ?, ?);

**Statement to get a query, by query ID**

SELECT \*

FROM contact\_us

WHERE query\_id = ?;

**Statement to get queries NOT responded to**

SELECT \*

FROM contact\_us

WHERE responded\_to = FALSE;

**Statement to get queries responded to**

SELECT \*

FROM contact\_us

WHERE responded\_to = TRUE;

**Statement to update responded\_to to TRUE**

UPDATE contact\_us

SET responded\_to = TRUE

WHERE query\_id = ?;

* **Medication Information (Add to elderly\_medication)**
  + Dosage (Validation: Must be out of: '5 mg', '10 mg', '20 mg', '50 mg', '100 mg', '250 mg', '500 mg', '1 g')
    - ENUM('5 mg', '10 mg', '20 mg', '50 mg', '100 mg', '250 mg', '500 mg', '1 g')
  + Frequency (Validation: Must be out of: 'Once Daily', 'Twice Daily', 'Three Times Daily', 'Four Times Daily', 'Every 6 Hours', 'Every 4 Hours', 'Every 3 Hours', 'Every 2 Hours', 'Every Hour')
    - ENUM('Once Daily', 'Twice Daily', 'Three Times Daily', 'Four Times Daily', 'Every 6 Hours', 'Every 4 Hours', 'Every 3 Hours', 'Every 2 Hours', 'Every Hour')

ALTER TABLE elderly\_medication

ADD COLUMN dosage ENUM('5 mg', '10 mg', '20 mg', '50 mg', '100 mg', '250 mg', '500 mg', '1 g'),

ADD COLUMN frequency ENUM('Once Daily', 'Twice Daily', 'Three Times Daily', 'Four Times Daily', 'Every 6 Hours', 'Every 4 Hours', 'Every 3 Hours', 'Every 2 Hours', 'Every Hour');

A screenshot of a phone

Description automatically generated

**Statement to insert all columns**

INSERT INTO elderly\_medication (elderly\_id, medication\_id, dosage, frequency)

VALUES (?, ?, ?, ?);

**Statement to insert dosage**

INSERT INTO elderly\_medication (dosage)

VALUES (?);

**Statement to insert frequency**

INSERT INTO elderly\_medication (frequency)

VALUES (?);

**Statement to get dosage**

SELECT dosage

FROM elderly\_medication

WHERE elderly\_id = ? AND medication\_id = ?;

**Statement to get frequency**

SELECT frequency

FROM elderly\_medication

WHERE elderly\_id = ? AND medication\_id = ?;

**Join statement to get elderly\_id, medication name, dosage and frequency**

SELECT e.elderly\_id, m.medication\_name, em.dosage, em.frequency

FROM elderly\_medication em

JOIN medication m ON em.medication\_id = m.medication\_id

JOIN elderly\_details e ON em.elderly\_id = e.elderly\_id

WHERE e.elderly\_id = ?;

Test (with elderly\_id = 7) :  
A screenshot of a medical prescription

Description automatically generated

* **Medication Log (medication\_log)**
  + User ID PK FK
    - Composite primary key
    - Composite foreign key
  + Medication ID PK FK
    - Composite primary key
    - Composite foreign key

Foreign key constraint ensures the combination of elderly\_id and medication\_id exists in the elderly\_medications table

* + Time Taken (Validation: Can’t be NULL, format -YYYY-MM-DD HH:MM:SS )
    - DATETIME
  + Time Logged (Validation: Can’t be NULL, entered automatically - YYYY-MM-DD HH:MM:SS)
    - DATETIME

CREATE TABLE medication\_log (

elderly\_id INT(11),

medication\_id INT(11),

time\_taken DATETIME NOT NULL,

time\_logged DATETIME DEFAULT CURRENT\_TIMESTAMP,

PRIMARY KEY (elderly\_id, medication\_id),

FOREIGN KEY (elderly\_id, medication\_id) REFERENCES elderly\_medication(elderly\_id, medication\_id)

);A screenshot of a computer

Description automatically generated

**Insert statement**

INSERT INTO medication\_log (elderly\_id, medication\_id, time\_taken)

VALUES (?, ?, ?);

**Statement to get the time taken, based on elderly\_id and medication\_id**

SELECT time\_taken

FROM medication\_log

WHERE elderly\_id = ? AND medication\_id = ?;

**JOIN statement to get medication name and time taken**

SELECT m.medication\_name, ml.time\_taken

FROM medication\_log ml

JOIN elderly\_medication em ON ml.elderly\_id = em.elderly\_id AND ml.medication\_id = em.medication\_id

JOIN medication m ON em.medication\_id = m.medication\_id

WHERE ml.elderly\_id = ?;