

# Software Portfolio

## Mobile Phone App:

### QR Code Scanner for Attendance-taking at Event Reception



# Introduction

[\[Click here to see app demo video\]](#)

## Use-case:

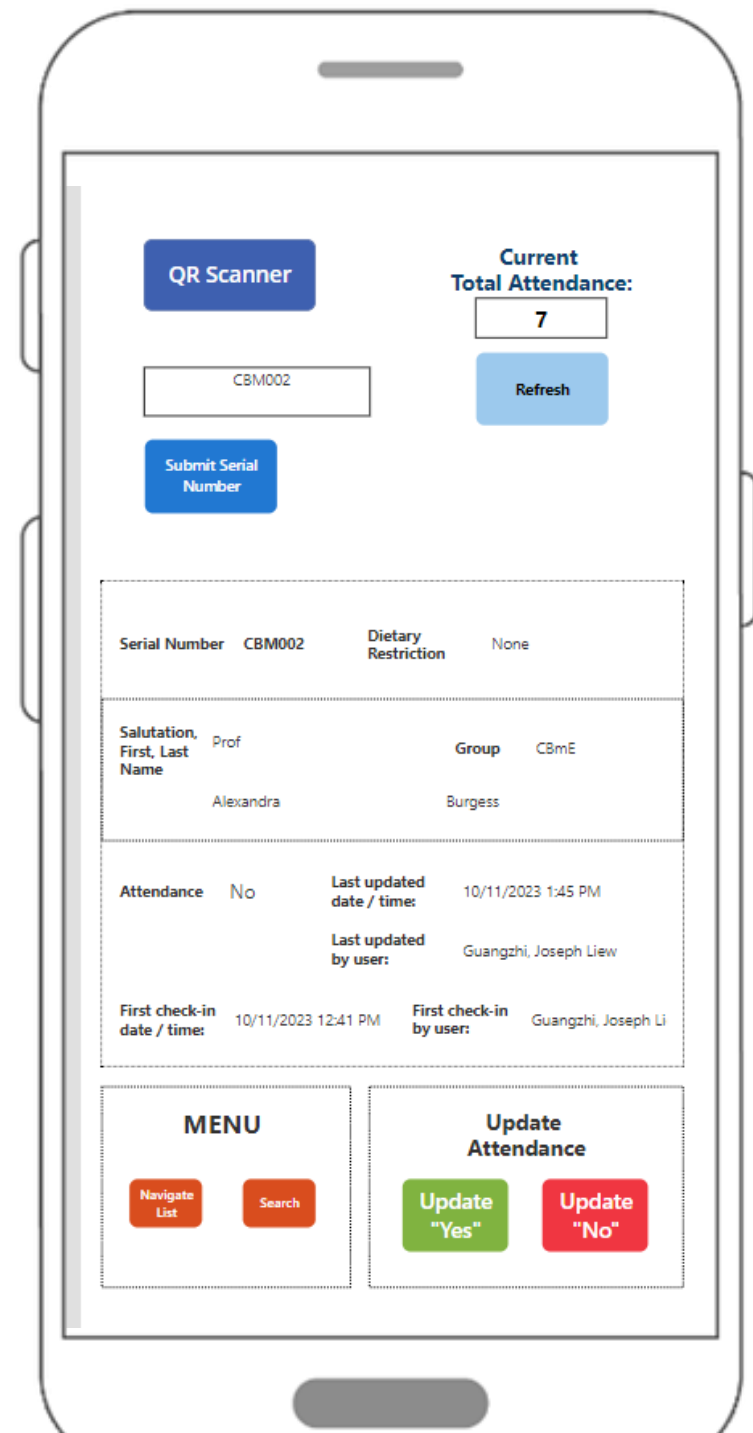
- Event reception staff who are checking-in participants and taking attendance

## Core Features

- Scan participant's QR code to automatically take attendance and display participant info.
- Participant misplaced QR code? Search participant info and update attendance
- Edit participant attendance and info
- Real-time data consolidation
- Audit trail
- Fast performance

## Software Design

- Object-oriented software design to overcome software and hardware limitation
- Client on-demand system to balance client and server load



# Problem Statement

## User's Problems

- During events, event organisers need to check-in and take attendance of participants at reception counter.
- Manual attendance tracking is time-consuming.
- Duplicate data entry can occur from multiple staff members checking in the same participant.
- No live update on total attendance.
- Staff lack participant information; unable to direct participant to seats.
- No record of attendee check-in time and audit trail.
- Difficulty updating participant information during the event, such as seating changes.

# Technical Challenges

- Limited infrastructure

Available infrastructure	Limitation
Database: SharePoint Lists	<ul style="list-style-type: none"><li>• No support for server-side computation</li><li>• Computation reliant on client device</li><li>• Low cap for concurrent database calls</li></ul>
Client device: Personal mobile phone	<ul style="list-style-type: none"><li>• Personal device with low compute power</li></ul>
Software: PowerApps	<ul style="list-style-type: none"><li>• Limited development options. No OOP.</li></ul>

# Solution

Limitation	Solution
<ul style="list-style-type: none"><li>• No support for server-side computation</li><li>• Computation reliant on client device</li><li>• Low cap for concurrent database calls</li></ul>	<ul style="list-style-type: none"><li>• Eliminate unnecessary database calls using client cache</li><li>• Conserve client and server load. Use pull-based design to deliver updated data on demand.</li></ul>
<ul style="list-style-type: none"><li>• Personal device with low compute power</li></ul>	<ul style="list-style-type: none"><li>• Only cache data required for computation</li><li>• Use modular design and cache data on client to avoid data duplication</li></ul>
<ul style="list-style-type: none"><li>• Limited development options. No OOP.</li></ul>	<ul style="list-style-type: none"><li>• Use object-structured design with global variables as cache</li></ul>

# Object-Structured Design

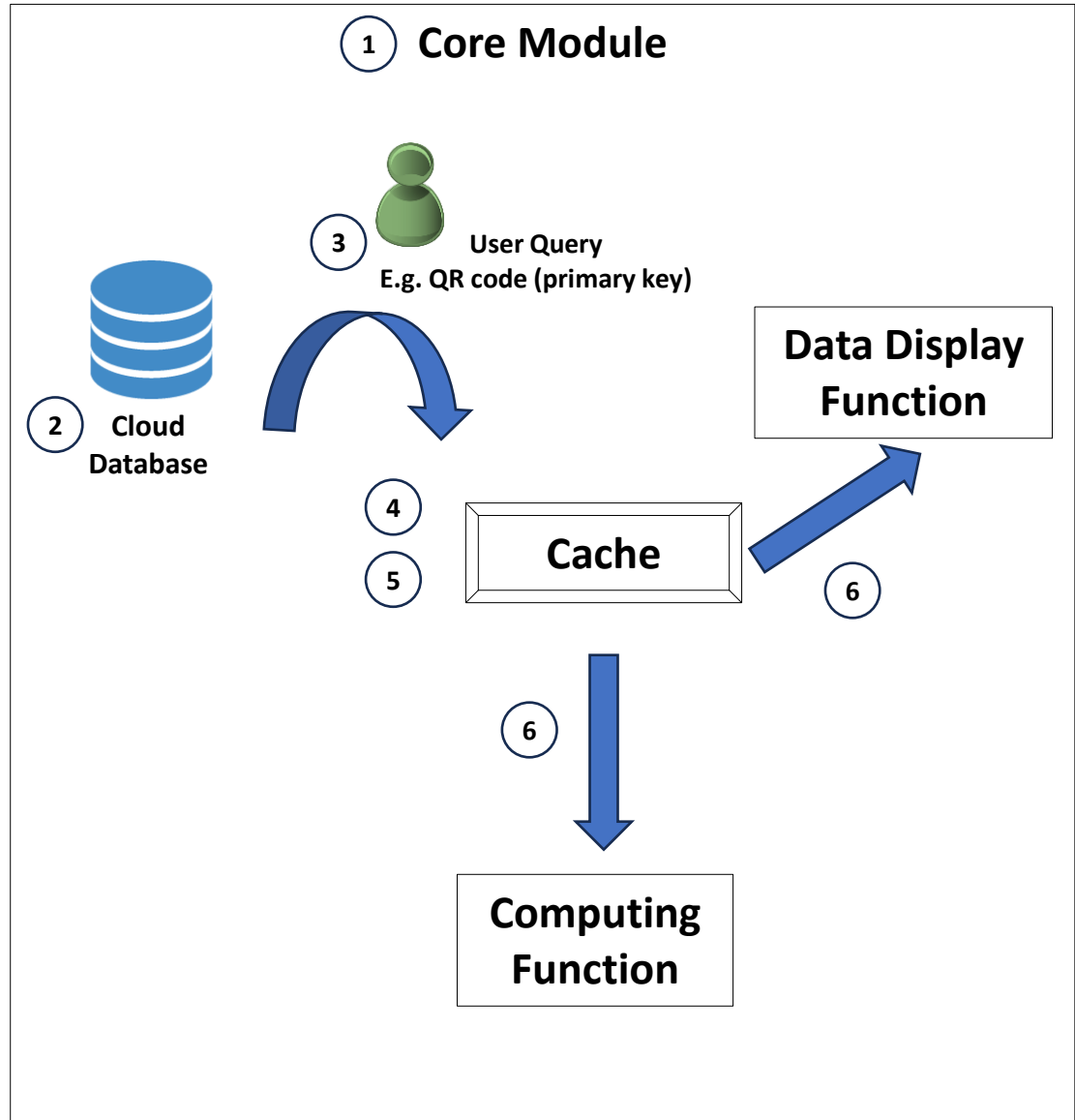
- Split app to 3 core modules, matching 3 different user workflows
  - Query / update via QR code
  - Query / update via search
  - Query / update via data table navigation
- Each module has own environment and cache
  - Client only loads the required environment selected by user
  - Conserves client memory for necessary cache
- Never load entire database
  - Only data of individual participant that the user has selected will be loaded into cache and displayed for modification

# Pseudocode Flowchart

Object structured design allows efficient reuse of common cache for device computation

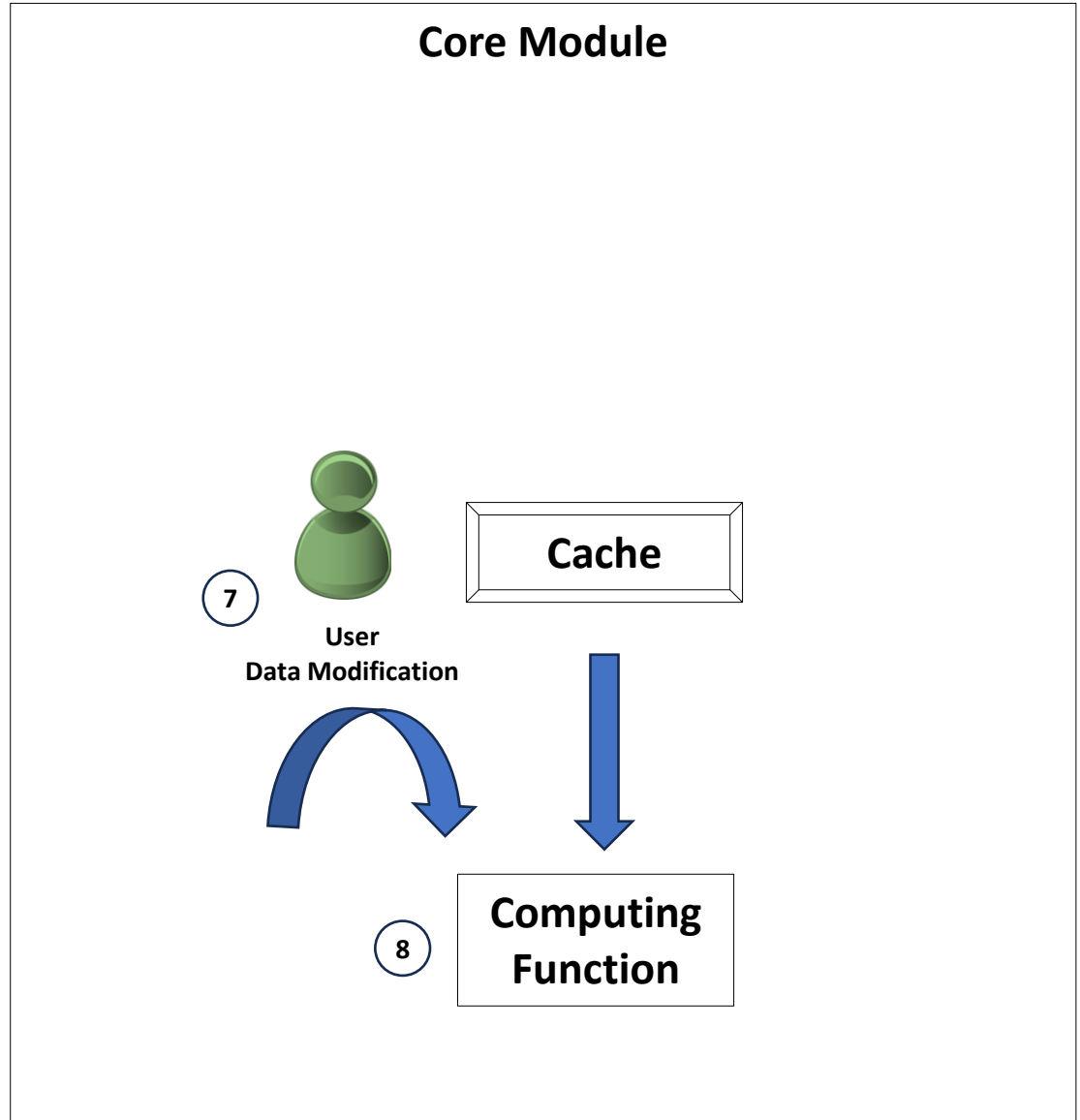
## Data flow

1. User selects desired **Core Module**
2. Data is stored in **Cloud Database**
3. Primary key inputted by user input translated into **User Query**
4. Device pulls query from **Cloud Database** and stores data in device's **Cache**
5. **Cache** provides data for current query instance, providing data required by different functions in the module.  
(Prevents duplicate database calls.)
6. Cache feeds data to **Data Display Function** for user to view data and **Computing Function** for user to modify data



# Pseudocode Flowchart

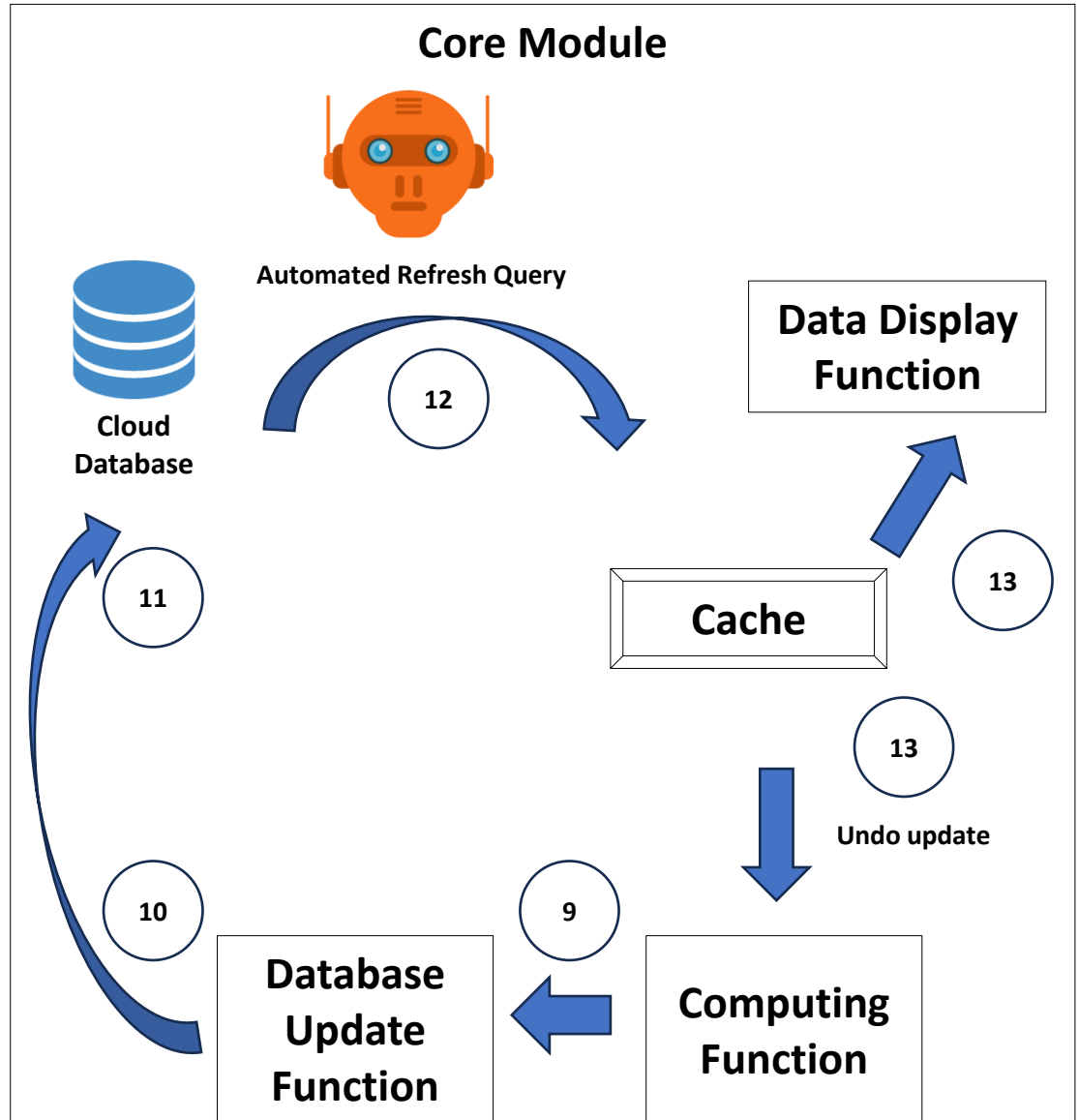
7. If there is **User Data Modification**
8. Then **Computing Function** processes changes to cached data





# Pseudocode Flowchart

9. **Computing Function** selects only modified data from cache and push to **Database Update Function**
10. **Database Update Function** identifies primary key and patches changes to **Cloud Database**
11. In this way, all devices will have synced data when they refresh the same query from **Cloud Database**
12. In current device, **Automated Refresh Query** refreshes **Cache** data  
(This allows the cache to be hydrated by most recent update. The most recent update might be made from another device.)
13. **Cache** hydrates the data displayed in the **Data Display Function**.  
Or, if user wants to undo change, feeds data to the **Computing Function** to process the undo update.



# Core Modules

[\[Click here to see app demo video\]](#)

QR scanning for attendance taking and on-demand info retrieval

Press to use phone camera to scan QR code

Attendance is automatically updated with no duplicates

Total attendees automatically refreshed when attendance is taken

Also refreshed on-demand when user selects refresh.

The screenshot displays the app's interface with several key components:

- QR Scanner Section:** Includes a blue 'QR Scanner' button, a text input field containing 'CBM002', a blue 'Submit Serial Number' button, and a 'Current Total Attendance: 7' display with a 'Refresh' button below it. Red arrows point from the explanatory text boxes to these elements.
- User Details Table:**

Serial Number	CBM002	Dietary Restriction	None
Salutation, First, Last Name	Prof Alexandra	Group	CBmE Burgess
- Attendance and Update Info:**

Attendance	No	Last updated date / time:	10/11/2023 1:45 PM
		Last updated by user:	Guangzhi, Joseph Liew
First check-in date / time:	10/11/2023 12:41 PM	First check-in by user:	Guangzhi, Joseph Li
- Bottom Navigation:**
  - MENU:** Contains 'Navigate List' and 'Search' buttons.
  - Update Attendance:** Contains 'Update "Yes"' and 'Update "No"' buttons.

QR Scanner

Current  
Total Attendance:

7

CBM002

Refresh

Submit Serial  
Number

Participant  
attendance and  
details displayed

Serial Number	CBM002	Dietary Restriction	None
Salutation, First, Last Name	Prof Alexandra	Group	CBmE Burgess
Attendance	No	Last updated date / time:	10/11/2023 1:45 PM
		Last updated by user:	Guangzhi, Joseph Liew
First check-in date / time:	10/11/2023 12:41 PM	First check-in by user:	Guangzhi, Joseph Li

Attendance check-in  
time and audit-trail

MENU

Navigate  
List

Search

Update  
Attendance

Update  
"Yes"

Update  
"No"

# On-demand Pull-based Search

## Search box:

If participant misplaced QR code, easily find search the database to find participant.

User then selects participant to view data / update attendance.

Search results with minimum identifier data downloaded from cloud to device cache.

Full data of participant only queried from database when user selects a search result.

This is staged data hydration to cache.

Avoid server lag and device memory overload caused by unnecessary download of entire dataset.

Mr  
Joan Nolan  
Group: Participant

Dr  
Ryan Nolan  
Group: Speakers

Serial\_number  
ATT051

Group  
Participant

Salutation  
Mr

First\_name  
Joan

Last\_name  
Nolan

Email  
Joalan@Paul.com

Dietary  
Vegetarian

Updated\_time

Updated\_name  
Guangzhi, Joseph Liew

First\_Recorder\_name

First\_record\_time

**MENU**

Navigate List

Scanner

Update Values

Current Total Attendance:  
7

# On-demand Pull-based Navigation and Update

User can also select participant to view or update via table navigation

Minimum identifier data downloaded from cloud to device cache and shown on table.

Full participant data of participant only queried from database when user selects a participant.

This is staged data hydration to cache.

Avoid server lag and device memory overload caused by unnecessary download of entire dataset.

The screenshot displays a mobile application interface. On the left, a vertical list of participants is shown, each with a name and a group (CBmE). The participants listed are Prof Alexander Butler, Prof Alexandra Burgess, Prof Amanda Thomson, Prof Angela Bower, Prof Cameron Paterson, Prof Charles Parr, Prof Colin McGrath, and Prof David Dyer. A red arrow points from the text 'Minimum identifier data downloaded from cloud to device cache and shown on table.' to the list of participants. Another red arrow points from the text 'Full participant data of participant only queried from database when user selects a participant.' to the detailed view of Prof Charles Parr. The detailed view on the right shows fields for Salutation (Prof), First\_name (Alexandra), Last\_name (Burgess), Email (Aleess@Irene.com), Dietary (None), First\_record\_time (10/11/2023 12:41 PM), First Recorder Name (Guangzhi, Joseph Liew), Updater Name (Guangzhi, Joseph Liew), Updated\_time (10/11/2023 1:45 PM), and Attendance (No). At the bottom, there is a 'MENU' section with 'Search' and 'Scanner' buttons, and a 'Current Total Attendances' section showing a value of 7.

Prof Alexander Butler	Group: CBmE
Prof Alexandra Burgess	Group: CBmE
Prof Amanda Thomson	Group: CBmE
Prof Angela Bower	Group: CBmE
Prof Cameron Paterson	Group: CBmE
Prof Charles Parr	Group: CBmE
Prof Colin McGrath	Group: CBmE
Prof David Dyer	Group: CBmE

**MENU**

Search

Scanner

Update Values

Refresh

Current Total Attendances: 7

# Results

## **Better event management**

- Live update of total attendees enabled event organiser to move participants from reception to event venue at optimum time.

## **Reduced manpower**

- Manpower at check-in reception reduced by half.
- Faster check-in: staff can process more check-ins in shorter period.
- In app search function eliminated the need to re-direct participant with no registration number or QR code a dedicated check-in counter.

## **Increased efficiency**

- Attendance automatically consolidated into single database.
- Simplified logistics:
  - All-in-one phone app for update, search, info retrieval
  - Replaced computer station setup at reception