# Initial Base Data Design

## Table of Contents

**Author**: S. Schmidt
**Version**: draft – in process
**Date**: 27/04/2022
This document is a work in progress as the overall specification for the app is still being developed.

**Version Changes:**
11/05/2022 – Updated document to reflect Associations table

# Initial Base Data Design

## Events Table.

```
Events

    eventID: Integer
    eventName: String
    Description: String
    dateTime: String
    Location: String
--------------------------------------------
    Endpoints
      /addevent?eventid=1
      /delevent?eventid=1
      /updevent?eventid=1
      /getevent?eventid=1
      /getevent
```

| Element | Details |
|---------|---------|
| eventid | ID assigned to event on insertion |
| eventName | Name of the event |
| Description | Description of the event |
| dateTime | Scheduled date time in for "**dd/mm/yyyy hh:mm am/pm NZST**" |
| Location | Location of event |

# Initial Base Data Design

## Staff Table.

```
Staff

  staffID: Integer
  firstName: String
  surname: String
  startDate: Date
  endDate: Date
  DOB: Date
  active: ENUM(Y,N)
-------------------------------------
  Endpoints
    /addstaff
    /deactivatestaff
    /updstaff
    /getstaff
```

| Element | Details |
|---|---|
| staffID | ID assigned to staff member on insertion |
| firstName | First Name of the staff member |
| surname | Surname of the staff member |
| startDate | Start date of employment "**dd/mm/yyyy hh:mm am/pm NZST**" |
| finishDate | Finish date of employment "**dd/mm/yyyy hh:mm am/pm NZST**" |
| DOB | Birthdate of employee "**dd/mm/yyyy**" |
| Active | ENUM(Y,N) |

# Initial Base Data Design

## Bubble Table.

```
Bubble

   bubbleID: Integer
   bubbleName: String
   Active: ENUM(Y,N)
---------------------------------------------------------
   Endpoints
     /addbubble
     /delbubble?bubbleID=1
     /updbubble?bubbleID=1
     /getbubble?bubbleID=1
```

| Element | Details |
|---|---|
| bubbleID | ID assigned to bubble on insertion |
| bubbleName | Name of the bubble |
| Active | ENUM(Y,N) |

# Initial Base Data Design

## Association Table.

```
┌─────────────────────────────────────┐
│ Associations                         │
├─────────────────────────────────────┤
│   assocID: Integer                   │
│   bubbleID: Integer                  │
│   staffID: Integer                   │
│ ------------------------------------ │
│   Endpoints                          │
│     /addassoc                        │
│     /updassoc                        │
│     /getassoc                        │
│     /delassoc                        │
└─────────────────────────────────────┘
```

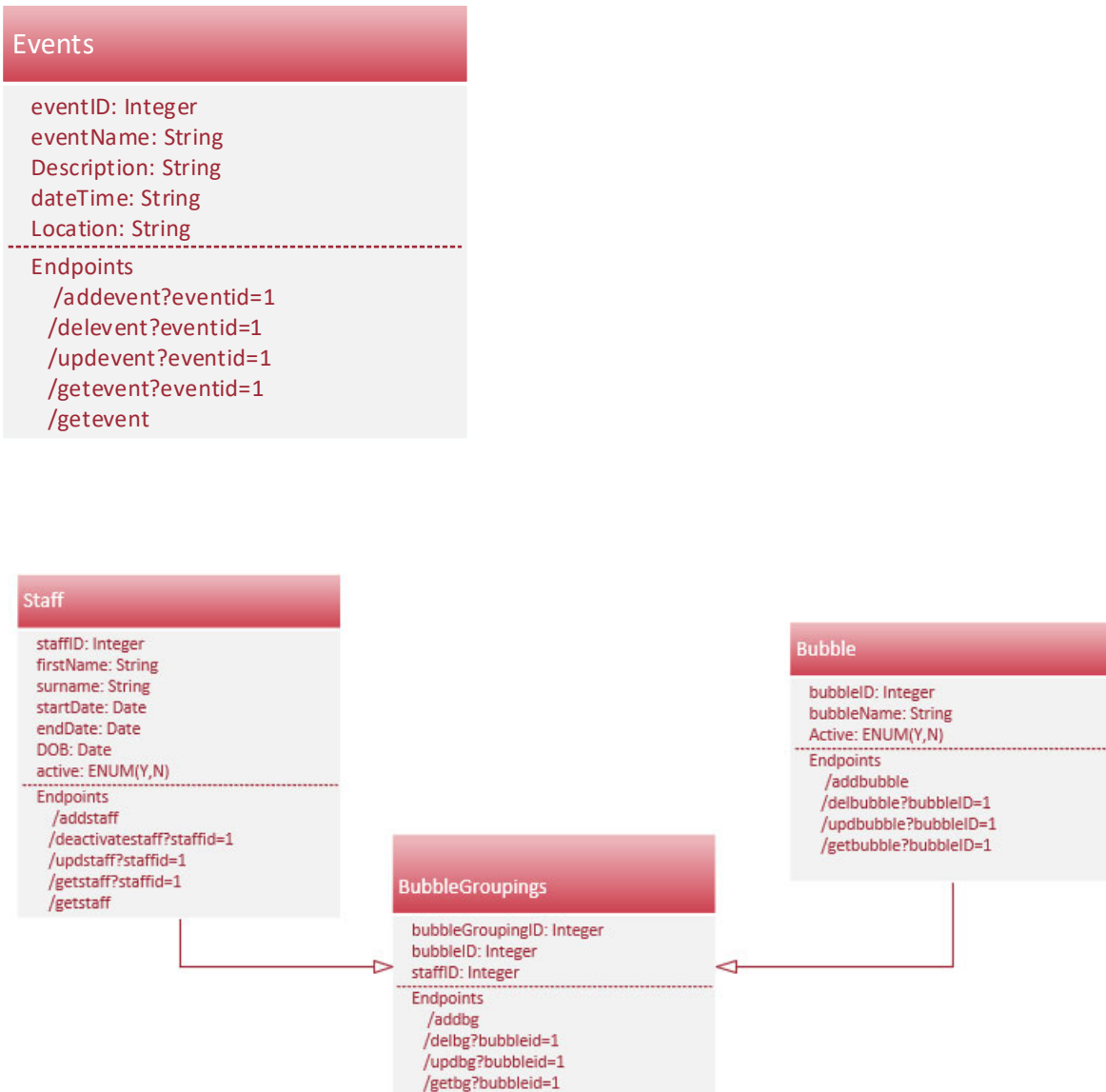| Element | Details |
|---------|---------|
| assocID | ID assigned to this record on insertion |
| bubbleID | ID of associated bubble record |
| staffID | ID of staff member |

This table is a link table. It connects staff to a bubble.
Each record records the staffID and the bubble the entry is associated.

# Initial Base Data Design

## Relationships.

**Events**

eventID: Integer
eventName: String
Description: String
dateTime: String
Location: String

---

Endpoints
  /addevent?eventid=1
  /delevent?eventid=1
  /updevent?eventid=1
  /getevent?eventid=1
  /getevent

**Staff**

staffID: Integer
firstName: String
surname: String
startDate: Date
endDate: Date
DOB: Date
active: ENUM(Y,N)

---

Endpoints
  /addstaff
  /deactivatestaff?staffid=1
  /updstaff?staffid=1
  /getstaff?staffid=1
  /getstaff

**Bubble**

bubbleID: Integer
bubbleName: String
Active: ENUM(Y,N)

---

Endpoints
  /addbubble
  /delbubble?bubbleID=1
  /updbubble?bubbleID=1
  /getbubble?bubbleID=1

**BubbleGroupings**

bubbleGroupingID: Integer
bubbleID: Integer
staffID: Integer

---

Endpoints
  /addbg
  /delbg?bubbleid=1
  /updbg?bubbleid=1
  /getbg?bubbleid=1

# Initial Base Data Design

## Client/Server Relationship and access path.



User Request

FrontEnd Application Server

BackEnd Server

Database Serer
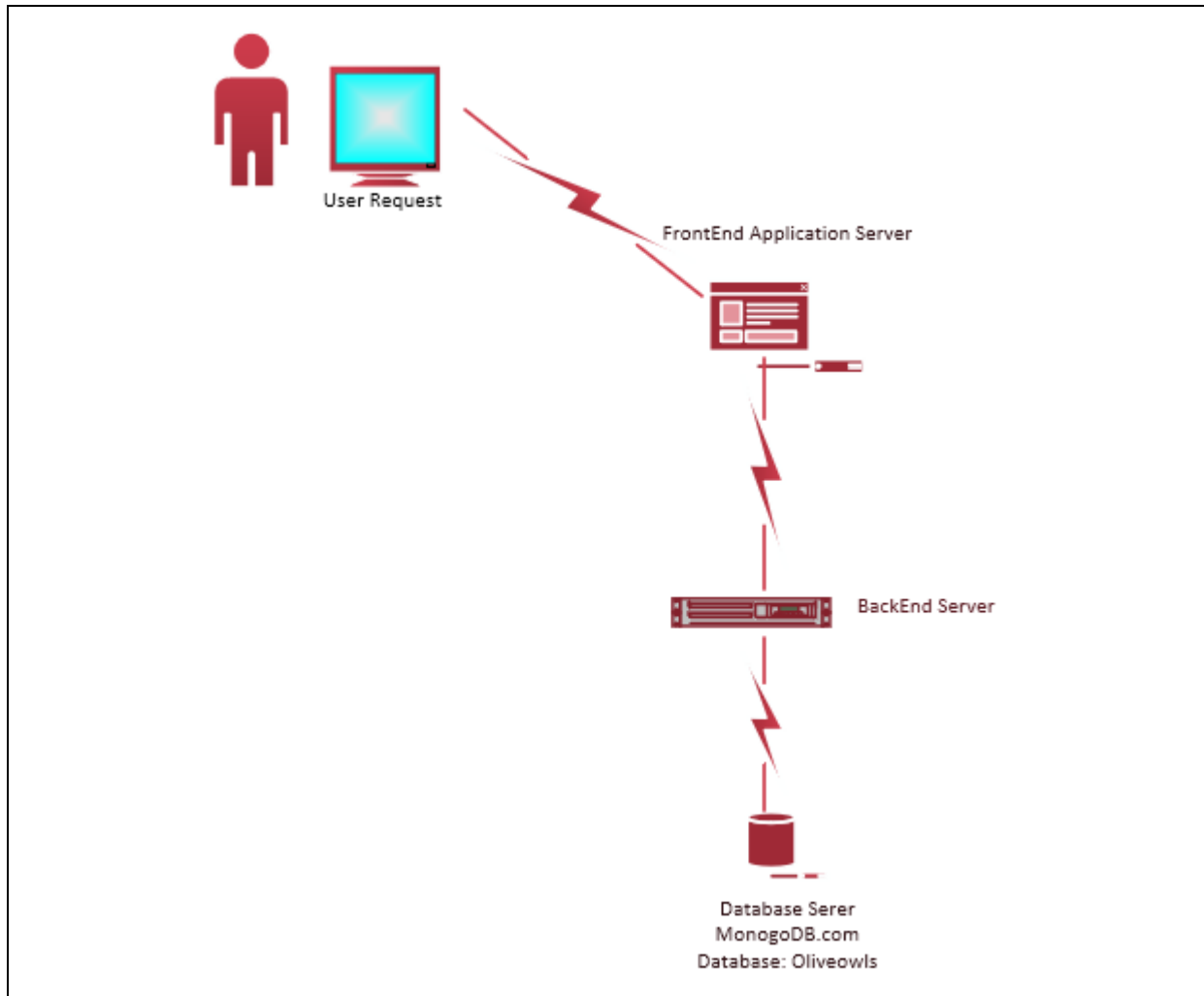MonogoDB.com
Database: Oliveowls

FIGURE 1

Application requests are made from the client to the frontend application system.
If data access is required, a defined request in the form of an endpoint is made from the frontend to the backend application system. The backend processes the endpoint requested and returns/processes the required information or returns an error message.

This allows the overall application to scale easily providing to areas where resource can be applied and managed. It also providers segregation of functions allowing for tighter, auditable security.