RADZEN SOFTWARE DOCUMENTATION

April 28th, 2023

PREPARED FOR / Center for Organ Recovery & Education

COMMUNITY PARTNER REPRESENTATIVE / Bill DiPlacido

CONSULTANT TEAM / Jim Chen, Rebecca Jiang, Anna Li, Victor Waddell

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Purpose

CORE partners with Donate Life America to help organize the annual Donate Life Month. Donate Life Month was created in 2003 to spread awareness for organ donation and is celebrated in April. The events of Donate Life Month encourage people to register as donors while honoring those who have saved lives through organ donations. CORE helps organize hundreds of events for the month, which includes finding speakers for each event, and must collaborate with numerous hospitals to fulfill event requests.

The year 2022 saw CORE hosting Donate Life event forms on Microsoft Forms/Excel. However, CORE's most recent technologies for event management included Gravity Forms, Excel, and VolunteerHub (an open API). They moved to Gravity Forms (WordPress plugin) and stored this event data in a SQL database.

Furthermore, CORE's efficiency was impeded by difficulties of leveraging its current software solution (VolunteerHub) fully. As a result, excessive labor exists due to relying on manual event coordination, tracking, and human error. Many staff members must manually communicate with stakeholders about event updates as VolunteerHub is not fully optimized. CORE must also manually update all involved platforms to reflect changes to events or information. CORE would like to centralize and automate some processes to use its technologies better.

In subsequent years of event management, CORE is determined to transition to centralizing its data and event management using the platform Radzen which its technical team will operate and manage. Our consulting team assisted in this transition by creating a minimum viable product (MVP) custom Radzen application for event management and following documentation on how to continue its development and sustainability. Automating and digitizing these processes would alleviate many of the issues regarding inefficiencies and errors during event coordination.

1.2 Intended Audience

CORE is responsible for managing the logistics of organizing and incorporating partner events into the DonateLife Month system. This involves community outreach, disseminating information about the events, arranging for the setup and dismantling of the events, and overseeing volunteer management. Before the solution, the staff relies on text messages or manual communication with volunteers, which can be time-consuming and challenging.

We compiled a list of user stories and needs that were the starting point for our solution development and helped us revolve our solution around the end users. Users of the software would be numerous users, such as event coordinators, general staff, volunteers, speakers, administrators, and partner organizations.

1.3 Intended Use

Our solution will help CORE digitize its event management process (particularly for DonateLife month). This will improve efficiency and reduce human error involved in manual event coordination while giving time back to CORE staff to focus on crucial objectives such as saving lives and educating the community. Volunteers and event organizers can receive timely notifications and contribute to more organized events. Better events will help enhance donation awareness and save more lives, accomplishing CORE's mission.

1.4 Features

The table below serves as an outline for what we were able to implement; we were able to cover all of the requirements given to us by the CORE team.

Feature	**Detail**
Login system	Varying permissions for admin, CORE staff, guests
View event status	If an event needs a volunteer speaker or leadership speaker that hasn't been filled yet, it will be labeled as

	incomplete
Create volunteer speaker	Details: phone number, address, accessibility requirements, etc.
View all volunteer speakers	See all volunteer speakers in the system
Create event	Partner organizations, or CORE staff input their event details and requests into the system
Edit event information	Mainly speaker procurement updates, CORE staff assignments, event date/times, location, etc.
Add volunteer speaker to event	
Add leadership speaker to event	
Delete events	Ability to delete event after creation
View all events	Chronological
View event details	CORE staff can view specific event details and see if they are staffed or not
See CORE attendance	See what CORE members are slated to attend an event
Filter/sort events	Ability to filter events by upcoming dates and if a speaker is still needed
Ability to “lock in” events	Prevent changes to an event while allowing only authorized members to make further changes. Admin only

Export event information to Excel & CSV	
Export volunteer information to Excel & CSV	

2. SYSTEM FEATURES AND REQUIREMENTS

2.1 Functional Requirements

Take time to define the functional requirements that are essential for the software to be built.

Minimum Hardware and Software Specifications for Radzen Blazor Studio

The following are the minimum hardware and software specifications required to install and run RADZEN:

Hardware Requirements

- Processor: Intel Core i3 or equivalent
- Memory: 4 GB RAM
- Storage: 1 GB free disk space

Software Requirements

- Operating System: Windows 7 SP1 or later, macOS 10.12 Sierra or later, or Ubuntu 18.04 LTS or later

Supported Platforms and Browsers

The following are the platforms and browsers supported by RADZEN:

Platforms

- Windows 7 SP1 or later (64-bit)
- macOS 10.12 Sierra or later
- Ubuntu 18.04 LTS or later

Browsers

- Google Chrome (latest stable version)
- Mozilla Firefox (latest stable version)

- Microsoft Edge (latest stable version)
- Safari (latest stable version)

SQL Server Support

- SQL Server Management Studio 17.x or later (Windows only)
- SQL Server Data Tools 17.x or later (Windows only)
- SQL Server 2012 or later

Azure Functions Support

- Azure Functions Core Tools 2.x or later

2.2 System Features

The platform we used to design the application was Radzen Studio. Radzen Studio has numerous features, hardware, software requirements, and documentation that are necessary to utilize its features thoroughly. Here are our requirements to setup and run the Radzen app:

- Mac or Windows OS
- SQL Server (we chose **Express** to facilitate the development, and it is preferable for development in a production environment and for smaller scale applications)
- .Net SDK 3.1.100 (Outdated .NET framework but supports the specifications set forth by CORE)
- Azure or SSMS (a database GUI is highly recommended either locally or cloud-based to establish the database and tables for the Radzen application)

Downloads to the requirements can be found throughout the installation guide.

2.3 Nonfunctional Requirements

There are also nonfunctional requirements to maintain the Radzen app and ensure it is up-to-date and sustainable:

- **Development Team** - The application will require a development team to update the application frequently as needed for end users.
- **Budget** - There are potential reasons that the app will require purchases of services through Radzen or other means to improve scalability or usability.
- **Additional Timeframes** - There would likely be the need for future sprints or further development of the app to meet the needs of CORE's end users. Many of the requirements and features have been documented in prior sections.

3. RADZEN INSTALLATION GUIDE AND RESOURCES

3.1 Installing Radzen Studio

1. Download Radzen Studio
    - MacOS: https://www.radzen.com/download/#macOS
    - WindowsOS: https://www.radzen.com/download/#windows
2. Open the setup file download and agree to install it.
3. Register with any email address, and Create a project with Blazor and .Net 7 selected. *We can change to .Net 3 later. Choose .Net 7 for now.*
4. Go to the project folder you created. Locate the **server** folder and delete it.
5. Go into the meta server folder while still in the project folder. Locate a file named **meta/app.json**. Open the file and change **serverVersion** to 3. Save your changes.
6. Download **.Net SDK 3.1.100**. The links below for each OS can automatically download the installer once clicked.
    - MacOS x64
    - Windows x64
7. Launch the .net installer, and proceed with the installation until it is completed.
8. Restart the Radzen Studio. You can now use Radzen Studio.

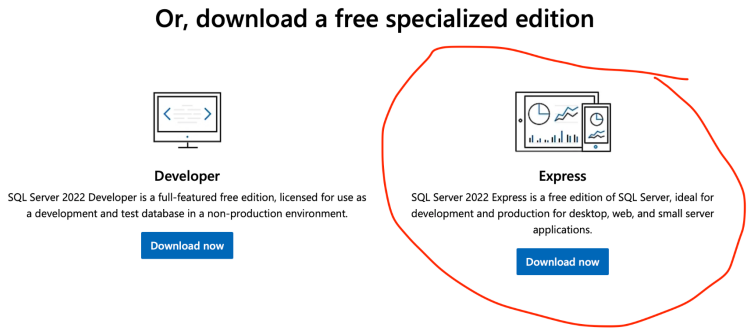
3.2 Local Database Creation for Windows OS (For Development Only)

Install SQL Server free version:

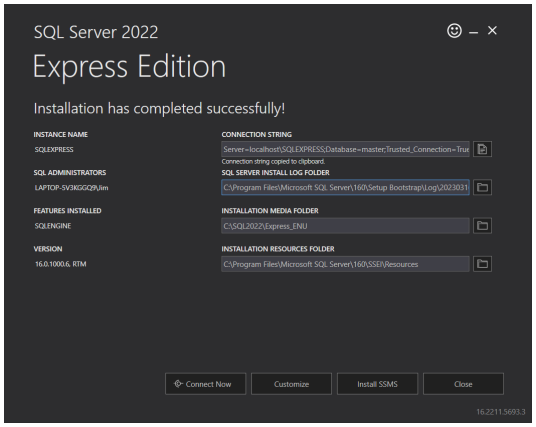
To continue with setting up the database for the development environment, we must download SQL Server from Microsoft:

1. Navigate to the SQL Server Downloads page on microsoft https://www.microsoft.com/en-us/sql-server/sql-server-downloads

2. Once on the sql server page, scroll down to where you see these options:



3. Choose the 'Express' version
4. Run the installed file and choose basic for installation. Proceed through the installer when prompted.
5. After installation, write down the name of the server path (**localhost\SQLEXPRESS** in this case), as we will need it later



Installing SQL Server Management Studio (SSMS)

In this section, we will install SSMS to operate the database in a GUI and later connect it to Radzen.

- Install SSMS and step through the installation

3.3 Azure Database Creation

An Azure Database can be set up with the official guide below:
https://learn.microsoft.com/en-us/azure/azure-sql/database/single-database-create-quickstart?view=azuresql&tabs=azure-portal

3.4 Database Schema Setup

Follow each of these steps in order as some tables have dependencies:

1. Create the database in SSMS or Azure

   The following query can be pasted inside of a **New Query** in SSMS or Azure and then press **Execute,** which will create a database named CORE (change instances of **CORE** to set a custom db name):

```
USE master
GO
IF NOT EXISTS (
  SELECT name
  FROM sys.databases
  WHERE name = N'CORE'
)
CREATE DATABASE [CORE]
GO
```

2. Copy and paste the text from each of these sql files, **in order**, inside separate **New Query**(s) and **Execute** to create each table:
   - Users table (tblADUsers.sql)
   - Hospitals table (tblHospitals.sql)
   - Events table (tblEvents.sql)
   - Speakers table (tblSpeakers.sql)
   - VolunteerSpeakerRequests table (tblVolunteerSpeakerRequests.sql)
   - COREAttendance table (tblCOREAttendance.sql)

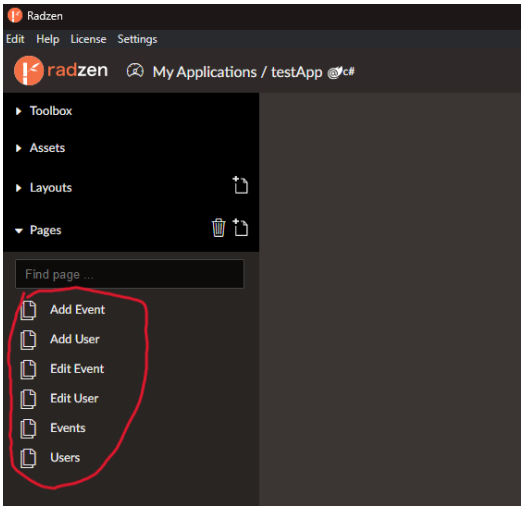
- Admin Users table (tblADUsers.sql)
- Event Logs table and trigger (tblEventLogs.sql)
- Speaker Logs table and trigger (tblSpeakerLogs.sql)
- VolunteerSpeakerRequest Logs table and trigger (tblVolunteerSpeakerRequests.sql)
- COREAttendance Logs table and trigger (tblCOREAttendanceLogs.sql)

3. Lastly, set up stored procedures. From the same folder from the previous step, use the same New Query and Execute process on the text inside the **sp_Event_StartNewEvent.sql** file.

3.5 Connect the SQL Server (SSMS or Azure) to Radzen

Once complete with the previous steps, we can now connect the database we just created in SSMS to our Radzen application.

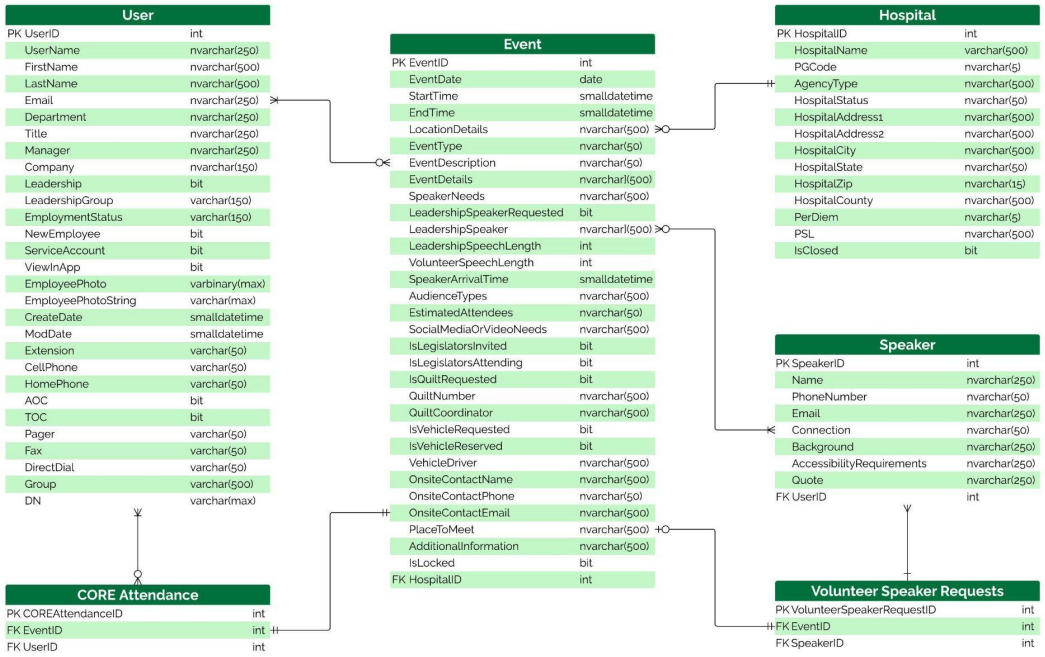
1. Reopen the Radzen application that you created.
2. Click **data** in the upper right corner of the app:
3. Ensure **MS SQL SERVER** is selected.
4. Paste **localhost\SQLEXPRESS** (or the server name you choose) into the Server field.
5. Put the name of the database you chose (we used **CORE**)
6. Choose **Windows Authentication** from the **Authentication** drop down. (OR if Azure is used, enter the username and password for the Azure DB.)
7. Press the Infer Schema button, this will load a page where we can customize page generation.
8. Check all buttons pertaining to **Automatic data-source and page generation**. Additionally, ensure that the tables and stored procedures buttons under **Entities** are checked.
9. The installation guide is complete. When observing the application, there should now be auto-generated CRUD pages after following the Radzen installation steps and the connect to database steps:


Radzen
Edit Help License Settings
radzen
My Applications / testApp
Toolbox
Assets
Layouts
Pages
Find page ...
Add Event
Add User
Edit Event
Edit User
Events
Users

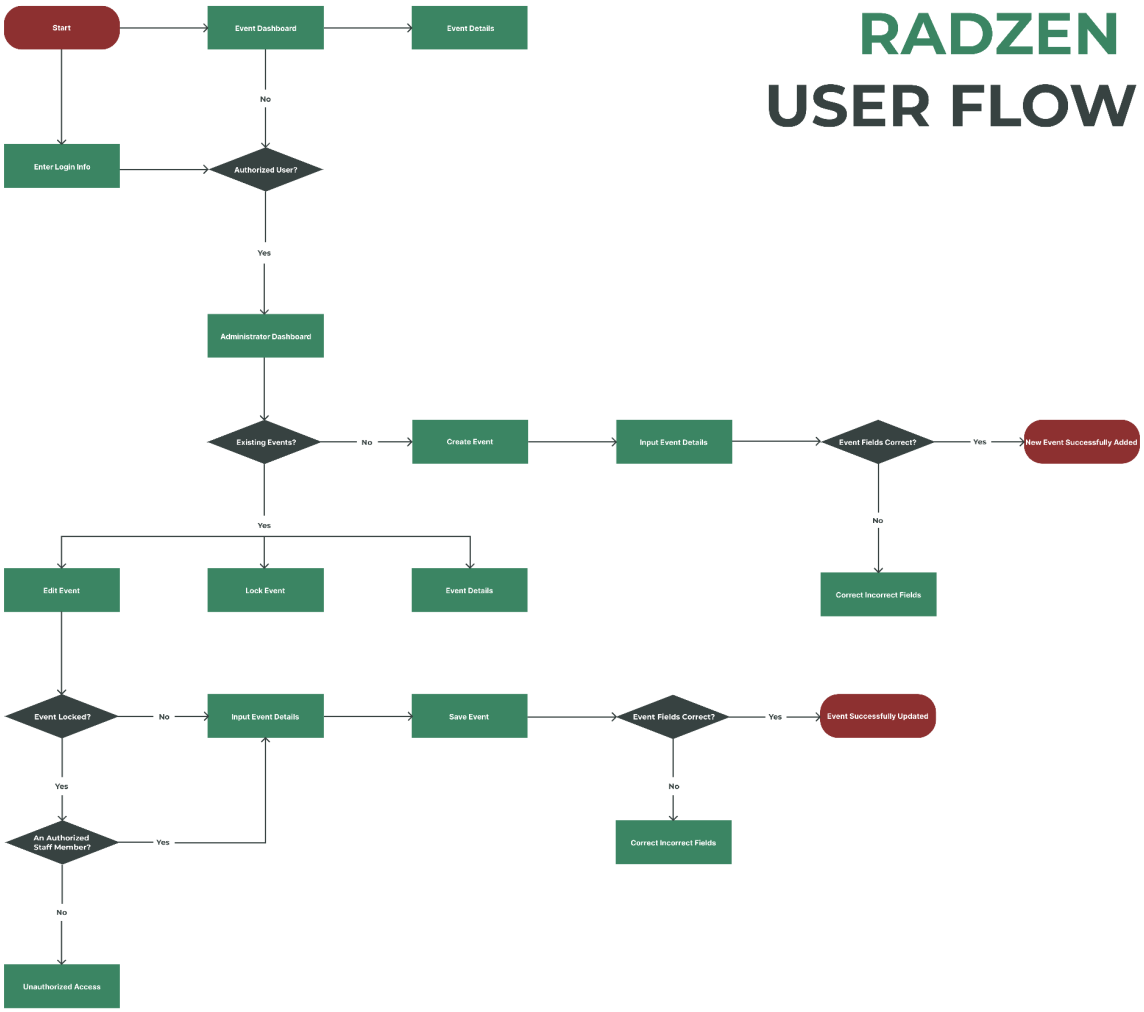
4. TECHNICAL DOCUMENTATION

4.1 CORE Entity Relationship Diagram - Link

CORE ERD



4.2 Radzen User Flow - Link



5. USER DOCUMENTATION

5.1 End-User Documentation

The end users of this web app include CORE staff, event speakers, and volunteers, administrators, and partner organizations. There are two types of users and each has access to different data.

Admin User

- Full access to read and edit all events
- Lock events

Normal User

- Access to read and edit unlocked events

Link to Radzen User Demo Video