PROJECT DOCUMENTATION

1.Introduction-

a. Goals and Objectives-

In today's digital age, effective management of church activities and community involvement is becoming increasingly important. The current implementation of RockRMS for website management has shown to be beneficial for online church management requirements. There is, however, a considerable gap in offering a smooth, accessible, and interactive experience for church members who use mobile devices exclusively. This disconnect restricts effective communication, event management, donation processing, and real-time interaction with the church community.

To answer this need, a mobile application that uses the capabilities of RockRMS is essential. This app should have an easy-to-use interface that allows church members to participate in various church events, access resources, manage their contributions, and remain in touch with the church community while on the go. To provide a uniform experience and real-time data synchronization, the app should interface seamlessly with the existing RockRMS-based website.

b. Scope-

The goal of developing this mobile application is to improve the church's digital presence, quicken church management activities, and encourage a stronger, more connected church community. This project aims not just to embrace technology improvements in church management, but also to keep the church relevant and accessible in a quickly changing digital environment.

2. Background-

a. Target Audience-

The primary target audience for the mobile application is the members of Grace Church. This diverse group encompasses individuals of varying ages, technological proficiencies, and roles within the church community.

b. Current System-

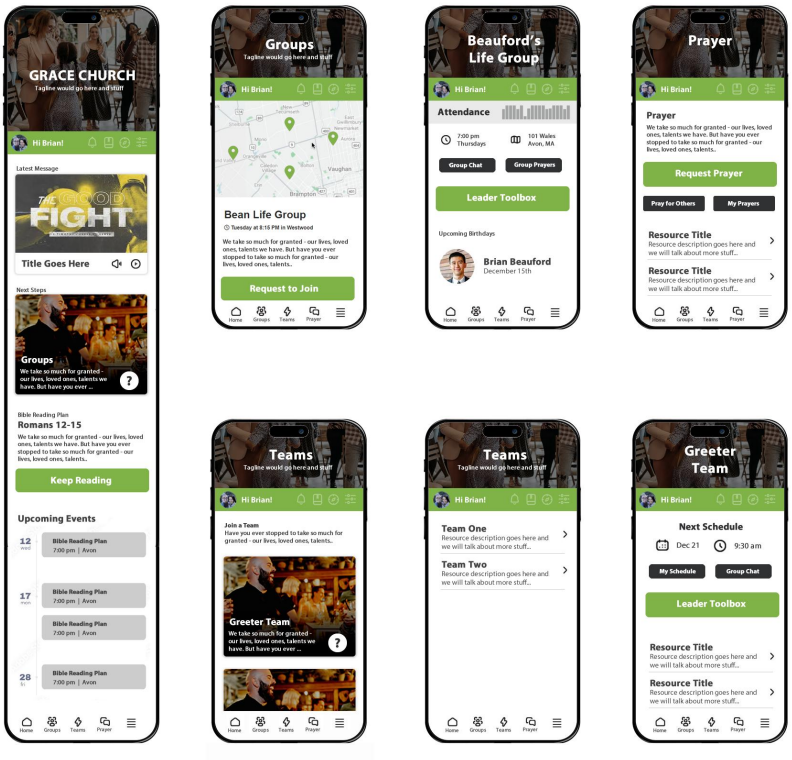
Currently, the church is in the process of developing a website using the same RockRMS platform. This effort represents the church's existing system and forms the basis for the forthcoming mobile application.

3. **Requirements**-

A. User Authentication and Profiles: Secure login for church members and staff, with profile management capabilities.
B. Event Management: Features for viewing church events and services.
C. Donation Management: A system for members to make and track their donations, including one-time and recurring contributions.
D. Communication Tools: Group-messaging keeps the community informed and engaged.
E. Resource Access: A section for accessing church-related resources.
F. Community Interaction: Features for members to interact, such as prayer
G. Data Synchronization with RockRMS Website: Uninterrupted integration with the RockRMS-based website for real-time data updates.
H. Privacy and Security Features: Ensuring user data protection.
I. User Analytics: Tools for church administrators to analyze app usage and member engagement.

4. Development-

a. **User Interface Design:** These are the mock-up's provided for the user interface.



b. **Technology Stack:**

- **Frontend (Mobile App Development):**

  I. XAML (Extensible Application Markup Language): XAML is widely used in various Microsoft frameworks for defining user interfaces. It's particularly relevant if you choose to develop the app using Xamarin.Forms for cross-platform development. Xamarin.Forms allow you to create native UIs for both Android and iOS from a single, shared codebase using C# and XAML.

  II. C#: As the primary programming language for Xamarin.Forms, C# is used to write the logic of your application. It's a versatile and powerful language with extensive support for modern programming techniques.

- **Backend Development:**

  I. C# with .NET Core/Framework: For the backend, you can use C# with .NET Core (or .NET Framework) to create a robust and scalable server-side application. .NET Core is a cross-platform framework, which is ideal for creating web APIs that the mobile application can interact with.

- **Integration with RockRMS:**

  I. Lava (RockRMS's templating language): Lava is a templating language used within RockRMS. If your mobile app needs to fetch and display data processed by Lava templates or needs to integrate closely with the RockRMS components that use Lava, understanding and utilizing this language will be essential.

  II. RockRMS API: Utilize RockRMS's API for seamless integration between the mobile app and the RockRMS platform, likely using C# for any server-side integration logic.