Prayer Tracking App Proposal

---

1. Project Name

Prayer Tracking

2. Team Members

- Eyad Amgad
- Gamal Azzam
- Mohamed Ahmed
- Aisha Mahmoud

---

3. Project Idea & General Description

Prayer Tracking is a comprehensive prayer tracking application designed to help Muslims maintain consistency in their daily prayers. The app will serve as **a personal prayer companion that reminds users of prayer times, tracks prayer completion, and provides a spiritual dashboard to monitor consistency.**

The application will offer features such as:

- Accurate prayer time calculations based on user location
- Customizable prayer notifications
- Prayer tracking and streak maintenance
- Qibla direction finder

Prayer Tracking aims to seamlessly integrate into users' daily lives, making prayer tracking intuitive and rewarding while fostering a deeper connection to their spiritual practice.

---

4. Technologies to be Used

Frontend:

- Flutter (for cross-platform mobile development)

Storage & Caching:

- Shared Preferences (for storing onboarding data locally on the device)

- SQfLite (for storing Data like Status for prayer and date)

DevOps & Tools:

- Git & GitHub (version control)
- Figma (design)
- Flutter Test (testing)
- Flutter DevTools (performance monitoring)

Libraries:

- flutter_svg: ^2.0.17
- flutter_screenutil: ^5.9.3
- hijri: ^3.0.0
- geolocator: ^13.0.3
- adhan_dart: ^1.1.2
- shared_preferences: ^2.5.2
- get: ^4.7.2
- permission_handler: ^11.4.0
- sensors_plus: ^6.1.1
- sqflite: ^2.4.2
- flutter_qiblah: ^3.1.0+1

---

5. Work Plan & Team Roles

- **Aisha Mahmoud**
  - Write and prepare the project proposal and presentation
  - Develop the onboarding pages of the application
  - Responsible for documentation
- **Eyad Amgad**
  - Design and develop the Statistics page
  - Implement prayer tracking functionality
  - Design UI components for consistent app experience
- **Gamal Azzam**
  - Develop the Splash Screen for initial app loading
  - Build the Home page as the main interface
  - Integrated Sqflite with the application for local data storage and offline data management..
- **Mohamed Abed**
  - Develop the Qibla direction finder feature
  - Implement location services integration
  - Ensure accurate Qibla calculations

TimeLine (Gantt chart):

Task	Week 1	Week 2	Week 3	Week 4	Week 5
Requirement Gathering	●				
UI Design using Figma	●				
Implement UI with Code		●			
Connect UI with Logic			●		
Integrate SqfLite& Fetch Prayer Times				●	
Debugging & Testing					●

6. Functional Requirements

Prayer Time Management:

- Accurate calculation of five daily prayer times based on location
- Support for various calculation methods (e.g., Muslim World League, ISNA)
- Automatic location updates with manual override
- Adjustable parameters for Fajr & Isha angles

Prayer Tracking:

- Simple interface to mark prayers as completed, missed, or made up later
- Calendar view to review prayer history
- Statistics dashboard showing prayer consistency

Notifications & Reminders:

- Customizable notifications before prayer times
- Adhan (call to prayer) options with multiple recitations
- Smart reminders based on user schedule and habits
- Vibration & silent modes for professional settings

Qibla Finder:

- Accurate Qibla direction using device compass
- Map view showing Qibla direction from the current location

7. Non-Functional Requirements

Performance:

- App startup time < 3 seconds
- Prayer time calculations completed within 1 second
- Smooth transitions between screens (60fps)
- Efficient battery usage (<5% daily consumption)
- Optimization for Flutter-specific performance metrics

Reliability:

- Offline functionality for core features
- Data synchronization when connectivity is restored
- Automatic backup of user prayer history
- System uptime of 99.9%
- Graceful error handling & recovery in Flutter UI

Security:

- End-to-end encryption for user data
- Compliance with data protection regulations (GDPR, etc.)
- Secure authentication with biometric options
- Regular security audits & updates
- Secure local storage for sensitive data using Flutter Secure Storage

Usability:

- Intuitive interface requiring minimal learning
- Accessibility features for users with disabilities
- Support for multiple languages (initially Arabic & English)
- Dark mode & adjustable font sizes using Flutter Themes
- One-handed operation of core features

Scalability:

- Support for up to 1M concurrent users
- Efficient database queries with response times <200ms
- Microservices architecture for future expansion
- API design allowing feature additions
- Modular Flutter architecture for scalability

Compatibility:

- Support for iOS 12+ & Android 7.0+
- Responsive design for various screen sizes using Flutter’s adaptive layouts
- Compatibility with different device capabilities
- Optimization for low-end devices in developing regions
- Support for various screen densities & orientations

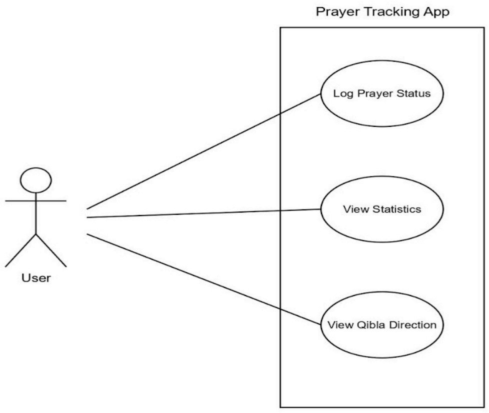
Localization:

- Full support for RTL & LTR languages using Flutter’s built-in internationalization
- Location-specific prayer calculation adjustments
- Cultural adaptations for different Islamic traditions
- Region-specific content & features

8. System Analysis & Design

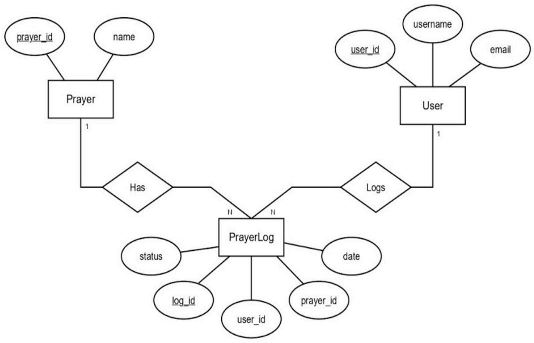
- **Problem Statement & Objectives:**

Use Case Diagram



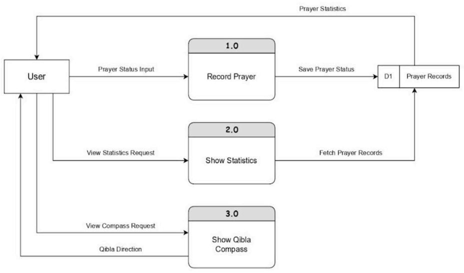
- **Database Design & Data Modeling:**

Entity-Relationship Diagram

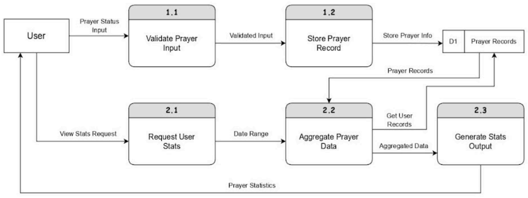


- ## Data Flow & System Behavior:

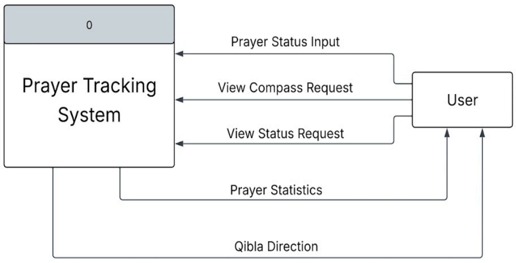
Level-0
DFD



Level-1
DFD



Context-Level DFD



9. Github Repository:

https://github.com/gamalazzam5/prayer-tracking-App/tree/master

10. Project Presentation:

https://drive.google.com/file/d/1CoW-MhvGixqB4Uvv29SLKU0wy1a9Q9cc/view?usp=sharing