**CHAPTER THREE**

**METHODOLOGY AND PROPOSED SYSTEM**

**3.1 Introduction**

This chapter describes the methodology used in developing the Church Database and Communication System. It discusses the selected software development process model, justifies its suitability, and explains the activities carried out in each phase of development. Furthermore, it introduces the proposed system, presents relevant models using UML diagrams, outlines the features of the system, and provides a review of its strengths and weaknesses.

**3.2 Methodology**

**3.2.1 Software Development Process Model**

For the development of the Church Database and Communication System, the **Agile Software Development Model** was adopted. Agile emphasizes iterative development, user involvement, and flexibility to adapt to changes, which makes it suitable for a system that requires feedback from stakeholders such as church leaders, secretaries, and members.

**3.2.2 Justification for Agile Model**

The Agile model was chosen because:

* The requirements of the church system were evolving and needed constant refinement.
* The system required continuous feedback from users to ensure that the functionalities met real needs.
* Agile allows for incremental releases, making it possible to test parts of the system before full deployment.
* It reduces the risks associated with rigid plan-driven approaches.

**3.2.3 Details of the Agile Process**

Agile was implemented through short iterations that involved planning, designing, coding, testing, and reviewing with users.

**3.2.4 Activities in Each Phase**

* **Planning Phase:** Requirements were gathered through interviews with church administrators and members. The main needs identified included membership management, contribution tracking, event scheduling, and communication.
* **Design Phase:** System models were created using UML diagrams (context model, sequence diagrams, structural and behavioral models) to represent system interactions.
* **Implementation Phase:** The system was developed in increments. Core features like database setup and user authentication were built first, followed by modules for contributions, events, and communication.
* **Testing Phase:** Each module was tested individually (unit testing) and then integrated into the system. User testing was conducted with church staff for feedback.
* **Review Phase:** Feedback from users guided refinements in usability, system navigation, and data reporting features.

**3.3 Proposed System**

**3.3.1 Introduction of Proposed System**

The Church Database and Communication System is designed to streamline church activities by maintaining accurate member records, managing financial contributions, scheduling events, and improving communication between church leaders and members.

**3.3.2 Models of the Proposed System**

The following UML models represent different views of the proposed system:

* **Context Model:** Shows the interaction between the system and external entities (e.g., Admin, Members, and Finance Team).
* **Interaction Model (Sequence Diagram):** Illustrates how a member registers, makes contributions, or receives notifications.
* **Structural Model (Class Diagram):** Displays the major classes such as *Member, Contribution, Event, User Account,* and their relationships.
* **Behavioral Model (Activity Diagram):** Describes workflows such as member registration, event scheduling, and sending announcements.

**3.3.3 New Features of the Proposed System**

* Automated tracking of contributions and tithe records.
* Event scheduling with reminders and notifications.
* Centralized member database with secure login.
* Communication tools (SMS/email alerts).
* User-friendly dashboard for church administrators.

**3.3.4 Development Tools and Environment**

* **Development Tools:** Visual Studio Code, MySQL, PHP/Laravel **Environment:** Windows OS with XAMPP server (Apache, MySQL, PHP), and a relational database management system.

**3.3.5 System Review**

* **Good Features:**
  + Secure login system.
  + Easy-to-use interface.
  + Real-time reporting of contributions.
  + Reliable member data management.

#### ****Features of the System****

* Membership management with customizable profiles.
* Event scheduling and online registrations.
* Volunteer and ministry scheduling tools.
* Online giving and donation tracking.
* Communication through bulk SMS, emails, and notifications.
* Integration with financial and email marketing platforms.
* Cloud-based with mobile accessibility.

### ****3.2 Review of the Good Features****

* User-friendly interface with intuitive navigation.
* Excellent integration with third-party apps.
* Strong volunteer and event scheduling capabilities.
* Secure cloud storage with role-based access.
* Multi-platform accessibility (web and mobile).

### ****3.3 Review of the Bad Features****

* Requires stable internet access, limiting offline usage.
* Subscription-based pricing may be expensive for small churches.
* Advanced customization options require technical knowledge.
* Limited flexibility in financial reporting compared to specialized accounting systems.

### ****3.4 Summary of System Review****

The Church Database and Communication System provides a reliable and efficient solution for managing church operations. While it is user-friendly and has multiple benefits, some limitations exist due to dependency on internet services and external APIs. Overall, the system is a significant improvement over manual methods and provides a strong foundation for future enhancements.