

CHURCH.IN

Project Report submitted to the Mahatma Gandhi University, Kottayam in
partial fulfillment of the requirement for the degree of

BACHELOR OF COMPUTER APPLICATION

Submitted By

Aksamol Tomichan (180021097951)

Albina Chacko (180021097952)

UNDER THE GUIDANCE OF

Mrs. Jyothish Abraham

(Assistant Professor, Department of Computer Application)



Puliyannmala, Kattappana,
Idukki Dt., Kerala – 685 508, Ph: 04868 270400

Web: www.christcollegekattappana.org,
Email.christcollegekattappana@gmail.com

2018 – 2021

Christ College, Puliyanmala

Puliyanmala, Kattappana, Kerala



CERTIFICATE

*This is to certify that the report titled **Church.in** was submitted by Ms. Akshamol Tomichan (180021097951) and Ms. Albina Chacko (180021097952), under our guidance and supervision was submitted in practical fulfillment of the requirement for the award of the degree of bachelor of computer applications during the academic year 2018-2021.*

Mrs. Jyothish Abraham

(Head of the department)

Mrs. Jyothish Abraham

(Project Guide)

Rev. Dr. Alex Louis CMI

Principal,

Christ College Puliyanmala

Submitted for the Viva –Voice Examination held on _____

Examiner 1

Examiner 2

Vazhithala

15/12/2020

CERTIFICATE

This is to certify that **Ms. Albina Chacko** (Reg. No 180021097952), Student of **Christ College, Puliyanmala P.O., Kattappana, Idukki-685515, Kerala**, attended a software development programme from **14th September 2020 to 30th October 2020** by using **PHP**. She has completed the project titled **Church.in** successfully in partial fulfillment of her BCA mini project under the guidance and supervision of **Santhisoft Technologies, Thodupuzha, Kerala**.

She has successfully completed the project and during the period, she was disciplined and hardworking.

We wish her success in all her future endeavors.

For Santhisoft Technologies,



Mr. Jismon Augustine
Project Guide



Fr. Dr. Boby Antony CMI
Director

Vazhithala

15/12/2020

CERTIFICATE

This is to certify that **Ms. Aksamol Tomichan** (Reg. No 180021097951), Student of **Christ College, Puliyanmala P.O., Kattappana, Idukki-685515, Kerala**, attended a software development programme from **14th September 2020 to 30th October 2020** by using **PHP**. She has completed the project titled **Church.in** successfully in partial fulfillment of her **BCA** mini project under the guidance and supervision of **Santhisoft Technologies, Thodupuzha, Kerala**.

She has successfully completed the project and during the period, she was disciplined and hardworking.

We wish her success in all her future endeavors.

For Santhisoft Technologies,



Mr. Jismon Augustine
Project Guide



Fr. Dr. Bobby Antony CMI
Director

DECLARATION

We, Aksamol Tomichan and Albina Chacko hereby declare that the project report “**CHURCH.IN**” is a record of work carried out under the valuable guidance of Mrs. Jyothish Abraham Assistant Professor Department of Computer Application, Christ College, Puliyanmala, as per the requirements of the curriculum of Bachelor of Computer Application programs of Mahatma Gandhi University, Kottayam. Further, we also declare that this report has not been submitted, full or part thereof, in any University/Institution for the award of any Degree/Diploma.

Aksamol Tomichan 180021097951

Albina Chacko 180021097952

ACKNOWLEDGEMENT

The success of any project depends largely on the encouragement and guidelines of many others. We would like to take this opportunity to express our gratitude to those people who have been instrumental in the successful completion of this project.

First and foremost, we give all glory, honour and praise to **God Almighty** who gave us wisdom and enabled us to complete the project successfully.

Words are inadequate to express our deep sense of gratitude to **Rev. Dr. Alex Louis CMI, Principal, Christ College, Puliyanmala**, for allowing us to utilize all the facilities of our college and also for his encouragement.

We extend our sincere gratitude to **Mrs. Jyothish Abraham, Head of the Department of Computer Application, Christ College, Puliyanmala**, who has been a constant source of inspiration and without her tremendous help and support this project would not have been materialized.

We own a particular debt of gratitude to our internal project guide, **Mrs. Jyothish Abraham, Department of Computer Application, Christ College, Puliyanmala**, for all the necessary help and support that she has extend to us. Her valuable suggestions, corrections and the sincere efforts to accomplish our project even under a tight time schedule were crucial in the successful completion of this project.

We would also like to express our appreciation to all friends for their comments, help and support.

TABLE OF CONTENTS

1. Introduction	2
1.1 Background and Motivation.....	3
1.2 The Proposed System.....	3
1.3 Project Scope.....	4
1.3.1 Existing System	4
1.3.2 Disadvantages of Existing System.....	5
1.3.3 Advantages of Proposed System.....	5
2. System Analysis	6
2.1 Introduction	7
2.2 Stake holders of this Project	7
2.2.1 Administrator.....	7
2.2.2 Family.....	7
2.3 Software Requirement Specification	8
2.3.1 Features Provided to Administrator.....	8
2.3.2 Features Provided to Family.....	8
2.4 Feasibility Study	8
2.4.1 Technical Feasibility.....	9
2.4.2 Economic Feasibility.....	10
2.4.3 Operational feasibility.....	10
2.5 Software Development Life Cycle Model.....	11
2.6 Hardware and Software Requirements.....	11
2.6.1 Software Specification.....	11
2.6.1.1 PHP.....	12
2.6.1.2 MYSQL	12
2.6.1.3 WAMP Server	14
2.6.1.4 Adobe Dreamweaver.....	15
2.6.1.5 Window 10.....	16
2.6.1.6 Microsoft Word.....	16

2.6.1.7 Smart Draw.....	17
2.6.2 Hardware Requirements	18
3. System Design	19
3.1 System Architecture.....	20
3.2 Module Design.....	21
3.3 Database Design	21
3.3.1 Normalization	22
3.3.2 Table Structure.....	24
3.3.3 Data Flow Diagrams.....	31
3.3.3.1 Introduction to Data Flow Diagrams.....	31
3.3.3.2 Data Flow Diagram.....	33
3.4 Interface Design	42
3.4.1 User Interface Screen Design.....	42
3.4.2 Output Design.....	43
4. Implementation	46
4.1 Coding Standards	47
5. Testing	50
5.1 Test Cases.....	51
5.1.1 Unit Testing.....	52
5.1.2 Integration Testing	53
5.1.3. Black Box Testing	55
5.1.4. White Box Testing.....	56
5.1.5. Validation Testing.....	57
5.1.6. User Acceptance Testing	59
5.2 Test Case Documents	59
6. Conclusion	60
6.1 Future Enhancements	61
7. Bibliography	62

8. Appendix	64
8.1 Screenshots.....	65
8.2 Sample Code.....	74

ABSTRACT

The project named as "**Church.in**" is a website of a church. It is designed to maintain the details about the family. Since the existing system is being controlled manually it faces a lot of limitations in reaching the expected performance. The proposed system handles these limitations and maintains the day to day activities in a systematic manner. The church management system includes Ward registration, Family Registration, Member registration, Death registration, Wedding registration, etc. 'Vicar' and 'Family' are the two stakeholders included in this system. Vicar is the prime stakeholder who has the privilege to control the entire system. Vicar can add, delete, edit and view the details in the software. The stakeholder 'Family' is limited in using many of the functionalities of the system compared to the 'Vicar'. Family can request the certificate for a particular date and the admin can generate the certificate.

INTRODUCTION

1.1 BACKGROUND AND MOTIVATION

In recent years there has been emphasis on the need for development of sound accounting practice in not-for-profit organization. The quality of data provided by church information systems has taken a small but important step forward in the past decade. During this period more churches have been recognizing the need for a conscientious effort aimed at developing a well designed information system which will provide data for helping each church achieve the most effective program possible with the use of a given amount of money.

In spite of this small step, the opinion still exists that few churches have recognize the need for improved data and even fewer churches have made serious efforts at improvement in this area. For purposes of this study church is defined as a “Local assembly or organization of baptized believers, banded together for the full performance of the will of Christ”.

Church management software is a specialized system which reduces the cost of operations and helps promote growth by assisting religious organizations and churches to automate daily administrative, management and reporting tasks.

1.2 THE PROPOSED SYSTEM

The software application which avoids more manual hours that need to spend in record keeping and generating reports. This application keeps the data in a centralized way which is available to all the users simultaneously. It is very easy to manage historical data in database. No specific training is required for the distributors to use this application. They can easily use the tool that decreases manual hours spending for normal things and hence increases the performance. Since the new project is aimed to manage the different phases in church like member’s registration, family details, ward details etc. A preliminary study was conducted in detail and several fact-finding techniques like interviewing, record searching; observation, comparison etc. were used to reach a better decision. The current system for this each activity was deeply studied and analysed. All the forms and other printed or non-printed formats for data collection were checked. Observation was done to a great extent to find the difficulties of the process and time delay in finding results.

The existing system is a manual and it is very difficult to handle. The proposed system is a computerized system, which is a menu driven system. We include a number of special features for the proper functioning of the system. We

have designed the system as user friendly. Security is an important feature of the system. The proposed system is a computerized church information system. The main objective of the system is to overcome the drawbacks of the existing system. We can achieve these objectives by computerizing the whole activities that are carried out manually. In the proposed system there are provisions to store all details in different tables. Speed and accuracy are the main features of the proposed system. The new system has been proposed to handle the existing functions in an efficient manner. The back end of the Project is worked on the basis of the MySQL. There are total 2 login for the vicar and the family. The data entering, data updation and data validation all are done in the background session. Also there is a technique for the filtering of the data on the basis of the different components, that is also done in the background. For doing all these services we must create a background connection with the database.

1.3 PROJECT SCOPE

The scope of the project is determined by the allocated time, resources and the client's requirements. Thus, the scope of the proposed Web based Church Information System can be summarized as follows:

The proposed system will provide an overall solution to handle the day-to-day activities related to the Church.in. The 'Church Information System' software is being developed as accurate and efficient online software for the user such as family and also the administrator that is the management of the church. In this system the record of the each request details are preserved along with their status. The system is also made secured as all the updates of the system can be done by the authorized person that is the administrator only.

1.3.1 Existing System

In earlier days the computer systems were not widely used. In the existing system, most of the work depends on human. The reports such as ward reports, family reports, member detail report etc., are done physically. In the current system security was not apt. As per the existing system the articles may not be adequate to meet the requirements of the members. It is difficult to manage the details of the family to retrieve and store the data in register.

Usually the management of church is done manually which is a difficult process. At present all the transactions one here are involved in ward registration, family registration, member registration and views are also recorded. All this are done

manually. This manual system had a number of disadvantages. Also the records are kept in books that make them difficult to maintain, difficult to handle. It was really time consuming and took away a lot of man-hours. This brought out the need of an advanced system for the management activities.

1.3.2 Disadvantages of Existing System

The study of the existing system helps to determine what the problems are:

After we studied the system we found some of the problems in the current scenario:

- All the information regarding to church and members are not secure and accurate.
- The family details are saved in different files.
- Member details are not accurate.
- Monthly and yearly report generation will take more time.
- Hand written contract documents
- Inflexibility of finding details

1.3.3 Advantages of Proposed System

The main advantages of the proposed '**Church.in**' compared to the existing system are:

- The security of the data :
The data is encapsulated and protected for the unwanted intrusions.
- User-friendly and Reduction of time :
By using this proposed system we can reduce the time consuming for making various data details like member registration, family details etc., so it can work as user friendly software.
- Minimized paperwork :
By the suggested system we can decrease the usage of papers and binding registers for entering ward details, family details, member details follows the book-keeping manner.
- Error Reduction and Improved Data Quality :
This planned system maintain the accurate data and also enhance the quality of data, in the case of church details we can produce the report in a better quality, error free for understanding in an easy way.
- Increased data security :
By the new proposed system we are ensuring our details security as the

ward details, family details, member details on the church management system.

- Increased Reliability and contain reports to view Reports daily basis :

Data reports sort the record and provide the better display of current data in the system.

SYSTEM ANALYSIS

2.1 INTRODUCTION

Church.in is a web application which is designed using PHP and MySQL. The major objective of our project is to provide more information about members. It is proposed to manage details about the family. The church information system based application that records the details of the family, member and report the details. This software help to register all the ward, family, member details etc.

The prime objective of computerization in any field is to make man's job easier. The ability of a computer to compute, store data and retrieve the stored data quickly has made it an effective tool to solve the drawbacks of manual labour .The objective of church management system is to enhance the currently working system more user friendly and flexible application in terms storage and retrieval of data .We have seen that computers must be used to reduce the time and complexity.

The technological development and impact of computers and internet on our lives has been verified over time affected various sectors of activity. And almost every task today is being run through computers. Church Information System Project Platform encompasses church information services. All in a single platform for easy administration and communication. This application is debuted to give information access to emerging technology to be more effective and efficient in reaching out to more people, thus expanding the kingdom. It is integrated solution for church administration.

2.2 STAKEHOLDERS OF THIS PROJECT

In this project we have Two types of stakeholders Administrator and Family. Administrator is the prime stakeholder who has the privilege to control the entire system. The stakeholder 'Family' is limited in using many of the functionalities of the system compared to the 'Administrator'.

2.2.1 Administrator

Administrator is the person who manages the software. He is the person who focuses on the data of the software. Administrator is mainly managing this software. The Administrator who have permission to access the details of family. The Administrator who have the permission to approved the details given by the family.

2.2.2 Family

They can access this software from anywhere at any time. The family can login to the software with specific username and password. And also they can register

their details. 'Family' is limited in using many of the functionalities of the system compared to the 'Administrator'.

2.3 SOFTWARE REQUIREMENTS SPECIFICATION

2.3.1 Features provided to Administrator

- This system should have the provision for login based on username and password.
- This system should have the provision for home page.
- This system should have the provision for add/modify/remove/view details in that church.
- The system should have the provision to generate the certificate and send to the family.
- The system should have the provision to logout.
- This system should have the provision for view/approval member details, wedding details etc in that church.

2.3.2 Features provided to Family

- This system should have the provision for login based on username and password.
- This system should have the provision for home page.
- This system should have the provision for registration by own details.
- This system should have provision for apply and view their own certificate.
- The system should have the provision to change password.
- The system should have the provision to logout.

2.4 FEASIBILITY STUDY

Feasibility is defined as the practical extent to which a project can be performed successfully. To evaluate feasibility, a feasibility study is performed, which determines whether the solution considered to accomplish the requirements is practical and workable in the software. Information such as resource availability, cost estimation for software development, benefits of the software to the organization after it is developed and cost to be incurred on its maintenance are considered during the feasibility study. The objective of the feasibility study is to establish the reasons for developing the software that is acceptable to users, adaptable to change and conformable to established standards. Various other objectives of feasibility study are

listed below.

- To analyze whether the software will meet organizational requirements.
- To determine whether the software can be implemented using the current technology and within the specified budget and schedule.
- To determine whether the software can be integrated with other existing software.

When our project guide as well as our client Mrs. Jyothish Abraham told us regarding the main project and about Income Tax Calculator for getting the desired product developed, it comes up with rough idea about what all functions the software must perform and which all features are expected from the software.

Referencing to this information, the we did studies and discussions about whether the desired system and its functionality are feasible to develop and the output of this phase is a feasibility study report that should contained adequate comments and recommendations. Various types of feasibility that we checked include technical feasibility, operational feasibility, and economic feasibility.

2.4.1 Technical feasibility

Technical feasibility assesses the current resources (such as hardware and software) and technology, which are required to accomplish user requirements in the software within the allocated time and budget. For this, the software development team ascertains whether the current resources and technology can be upgraded or added in the software to accomplish specified user requirements. Technical feasibility also performs the following tasks.

- Analyzes the technical skills and capabilities of the software development team members.
- Determines whether the relevant technology is stable and established.
- Ascertains that the technology chosen for software development has a large number of users so that they can be consulted when problems arise or improvements are required.

Our project is technically feasible, because we are using latest programming languages in the market, JSP and MySQL. Also we are the pioneers in this area. We use the latest versions of JSP and MySQL for application development; such that, the product's compatibility even with the latest systems are guaranteed. Our software is also developed following the latest and up to date industry standards. It

also maintains latest designs and user interfaces. As we are using these latest technologies which is currently trending and used by a number of developers across the globe, we can say that our project is technically feasible.

2.4.2 Economic Feasibility

Economic feasibility determines whether the required software is capable of generating financial gains for an organization. It involves the cost incurred on the software development team, estimated cost of hardware and software, cost of performing feasibility study, and so on. For this, it is essential to consider expenses made on purchases (such as hardware purchase) and activities required to carry out software development. In addition, it is necessary to consider the benefits that can be achieved by developing the software. Software is said to be economically feasible if it focuses on the issues listed below.

- Cost incurred on software development to produce long-term gains for an organization.
- Cost required to conduct full software investigation (such as requirements elicitation and requirements analysis).
- Cost of hardware, software, development team, and training.

It is estimated that my project is economically feasible, because this is a small project, and only a minor amount is contributed to its production. The software is developed at very minimum budgets. Limited amount of resources has been utilized so as to reduce the gross production cost. Moreover this is a student project and didn't want to appoint any employees, so as a result the total cost of the project is very minimal.

2.4.3 Operational Feasibility

Operational feasibility assesses the extent to which the required software performs a series of steps to solve business problems and user requirements. This feasibility is dependent on human resources (software development team) and involves visualizing whether the software will operate after it is developed and be operative once it is installed. Operational feasibility also performs the following tasks.

- Determines whether the problems anticipated in user requirements are of high priority.
- Determines whether the solution suggested by the software development team is acceptable.

- Analyzes whether users will adapt to a new software.
- Determines whether the organization is satisfied by the alternative solutions proposed by the software development team.

We found that our project is operationally feasible as our project has been tested to ensure its acceptance in the current industry. We totally ensured that our project can compete with any other software product in the market. Advanced and sophisticated verification & confirmation techniques have been implemented in order to ensure the best security we can offer to the customers. User interfaces have been re-imagined to provide the simple and best user interaction with the product. During the development stage of our project we always interact with the user, so we able to provide the user with new facilities.

2.5 SOFTWARE DEVELOPMENT LIFE CYCLE MODEL

One of the basic notions of the software development process is SDLC models which stand for Software Development Life Cycle models. SDLC – is a continuous process, which starts from the moment, when it's made a decision to launch the project, and it ends at the moment of its full remove from the exploitation. Software development lifecycle (SDLC) is a framework that defines the steps involved in the development of software. It covers the detailed plan for building, deploying and maintaining the software. SDLC defines the complete cycle of development i.e. all the tasks involved in gathering a requirement for the maintenance of a product.

Some of the common SDLC models are Waterfall Model, V-Shaped Model, Prototype Model, Spiral Model, Iterative Incremental Model, Big Bang Model, Agile Model. We used Agile Model for our Project.

2.6 HARDWARE AND SOFTWARE REQUIREMENTS

2.6.1 Software Specification

This project is built upon the latest technology software.

Front end	:	PHP
Development tool	:	SQL
Database	:	My SQL
Web server	:	WAMP Server
Operating System	:	Windows 10

2.6.1.1 PHP

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. As of January 2013, PHP was installed on more than 240 million websites (39% of those sampled) and 2.1 million web servers. Originally created by Rasmus Lerdorf in 1994, the reference implementation of PHP is now produced by The PHP Group. While PHP originally stood for Personal Home Page, it now stands for PHP: Hypertext Pre-processor, a recursive acronym.

PHP code can be simply mixed with HTML code, or it can be used in combination with various templating engines and web frameworks. PHP code is usually processed by a PHP interpreter, which is usually implemented as a web server's native module or a Common Gateway Interface (CGI) executable. After the PHP code is interpreted and executed, the web server sends resulting output to its client, usually in form of a part of the generated web page - for example, PHP code can generate a web page's HTML code, an image, or some other data. PHP has also evolved to include a command-line interface (CLI) capability and can be used in standalone graphical applications. PHP is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

2.6.1.2 MySQL

MySQL is the world's most popular open source database software, with over 100 million copies of its software downloaded or distributed throughout its history. With its superior speed, reliability, and ease of use, MySQL has become the preferred choice for Web, Web 2.0, SaaS, ISV, Telecom companies and forward-thinking corporate IT Managers because it eliminates the major problems associated with downtime, maintenance and administration for modern, online applications.

Many of the world's largest and fastest-growing organizations use MySQL to save time and money powering their high-volume Web sites, critical business systems, and packaged software — including industry leaders such as Yahoo!, Alcatel-Lucent, Google, Nokia, YouTube, Wikipedia, and Booking.com.

The flagship MySQL offering is MySQL Enterprise, a comprehensive set of production-tested software, proactive monitoring tools, and premium support services available in an affordable annual subscription. MySQL is a key part of LAMP (Linux, Apache, MySQL, PHP / Perl / Python), the fast-growing open source enterprise software stack. More and more companies are using LAMP as an alternative to

expensive proprietary software stacks because of its lower cost and freedom from platform lock-in.

MySQL is a database management system. A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

MySQL databases are relational. A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and —pointers between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data.

The SQL part of —MySQL stands for —Structured Query Language. SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax. SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, —SQL-92 refers to the standard released in 1992,

—SQL:1999 refers to the standard released in 1999, and —SQL:2003 refers to the current version of the standard.

We use the phrase —the SQL standard to mean the current version of the SQL Standard at any time. MySQL software is Open Source. Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs.

The MySQL Database Server is very fast, reliable, scalable, and easy to use. If that is what you are looking for, you should give it a try. MySQL Server can

run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention. If you dedicate an entire machine to MySQL, you can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available. MySQL can also scale up to clusters of machines, networked together.

MySQL Server was originally developed to handle large databases much faster than existing solutions and has been successfully used in highly demanding production environments for several years. Although under constant development, MySQL Server today offers a rich and useful set of functions. Its connectivity, speed, and security make MySQL Server highly suited for accessing databases on the Internet. MySQL Server works in client/server or embedded systems. The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs). We also provide MySQL Server as an embedded multi-threaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.

A large amount of contributed MySQL software is available. MySQL Server has a practical set of features developed in close cooperation with our users. It is very likely that your favourite application or language supports the MySQL Database Server.

2.6.1.3 WAMP Server

WAMP Server is a Windows web development environment. It allows you to create web applications with Apache2, PHP and a MySQL database. Alongside, PhpMyAdmin allows you to manage easily your databases. WAMP Server refers to a software stack for the Microsoft Windows operating system, created by Romain Bourdon and consisting of the Apache web server, Open SSL for SSL support, MySQL database and PHP programming language. WAMP Server is a Web development platform on Windows that allows you to create dynamic Web applications with Apache2, PHP, MySQL and MariaDB. Wamp Server automatically installs everything you need to intuitively develop Web applications. You will be able to tune your server without even touching its setting files. Best of all, Wamp Server is available for free (under GPML license) in both 32 and 64 bit versions. Wamp Server is not compatible with Windows XP, SP3, or Windows Server 2003.

WAMP Server's functionalities are very complete and easy to use so we won't explain here how to use them.

With a left click on WAMP Server's icon, you will be able to:

- Manage your Apache and MySQL services
- Switch online/offline (give access to everyone or only localhost)
- Install and switch Apache, MySQL and PHP releases
- Manage your server's settings
- Access your logs
- Access your settings files
- Create alias

2.6.1.4 Adobe Dreamweaver

Adobe Dreamweaver is a proprietary web development tool from Adobe Inc. It was created by Macromedia in 1997 and developed by them until Macromedia was acquired by Adobe Systems in 2005. Adobe Dreamweaver is available for the macOS and Windows operating systems. Adobe Dreamweaver CC is a web design and development application that uses both a visual design surface known as Live View and a code editor with standard features such as syntax highlighting, code completion, and code collapsing as well as more advanced features such as real-time syntax checking and code introspection for generating code hints to assist the user in writing code. Combined with an array of site management tools, Dreamweaver allows for its users to design, code and manage websites, as well as mobile content. Dreamweaver is an Integrated Development Environment (IDE) tool. You can live preview of changes for the frontend. Dreamweaver is positioned as a versatile web design and development tool that enables visualization of web content while coding. Dreamweaver, like other HTML editors, edits files locally then uploads them to the remote web server using FTP, SFTP, or WebDAV. Dreamweaver CS4 now supports the Subversion (SVN) version control system.

2.6.1.5 Windows 10

Operating System is defined as a program that manages the computer hardware. An operating system can be viewed as a scheduler, where it has resources for which it has charge. Resources include CPU, memory, I/O device and disk space. In another view, the operating system is a new machine. The third view is that operating system is a multiplexer which allows sharing of resources provides

protection from interference and provides a level of cooperation between users. This project is developed using Windows 10 as the operating system and supports its latest versions. Windows 10 is a series of personal computer operating systems produced by Microsoft as part of its Windows NT family of operating systems. It is the successor to Windows 8.1, and was released to manufacturing on July 15, 2015, and to retail on July 29, 2015. One of Windows 10's most notable features is support for universal apps. Windows 10 also introduced the Microsoft Edge web browser, a virtual desktop system, a window and desktop management feature called Task View, support for fingerprint and face recognition login, new security features for enterprise environments, and DirectX12. Windows 10 received mostly positive reviews upon its original release in July 2015. Critics praised Microsoft's decision to provide a desktop-oriented interfacing line with previous versions of Windows, contrasting the tablet-oriented approach of 8, although Windows 10's touch-oriented user interface mode was criticized for containing regressions upon the touch-oriented interface of Windows 8. Critics also praised the improvements to Windows 10's bundled software over Windows 8.1, Xbox Live integration, as well as the functionality and capabilities of the Cortana personal assistant and the replacement of Internet Explorer with Microsoft Edge. However, media outlets have been critical of changes to operating system behaviours, including mandatory update installation, privacy concerns over data collection performed by the OS for Microsoft and its partners and the adware-like tactics used to promote the operating system on its release.

2.6.1.6 Microsoft Word

Microsoft Word (or simply Word) is a word processor developed by Microsoft. It was first released on October 25, 1983 under the name Multi-Tool Word for Xenix systems. Subsequent versions were later written for several other platforms including IBM PCs running DOS (1983), Apple Macintosh running the Classic Mac OS (1985), AT&T Unix PC (1985), Atari ST (1988), OS/2 (1989), Microsoft Windows (1989), SCO Unix (1994), and macOS (formerly OS X; 2001).

Commercial versions of Word are licensed as a standalone product or as a component of Microsoft Office, Windows RT or the discontinued Microsoft Works suite. Unlike most MS-DOS programs at the time, Microsoft Word was designed to be used with a mouse. Advertisements depicted the Microsoft Mouse, and described Word as a WYSIWYG, windowed word processor with the ability to undo and display bold, italic, and underlined text, although it could not render fonts. It was not initially

popular, since its user interface was different from the leading word processor at the time, WordStar. However, Microsoft steadily improved the product, releasing versions 2.0 through 5.0 over the next six years. In 1985, Microsoft ported Word to the classic Mac OS (known as Macintosh System Software at the time). This was made easier by Word for DOS having been designed for use with high resolution displays and laser printers, even though none were yet available to the general public. Following the precedents of Lisa Write and MacWrite, Word for Mac OS added true WYSIWYG features. It fulfilled a need for a word processor that was more capable than MacWrite. After its release, Word for Mac OS's sales were higher than its MS-DOS counterpart for at least four years.

2.6.1.7 SmartDraw

SmartDraw is a diagram tool used to make flowcharts, organization charts, mind maps, project charts, and other business visuals. SmartDraw has two versions: an online edition and a downloadable edition for Windows desktop. SmartDraw integrates with Microsoft Office products including Word, PowerPoint, and Excel and G Suite applications like Google Docs and Google Sheets. SmartDraw has apps for Atlassian's Confluence, Jira, and Trello. SmartDraw is compatible with Google Drive, Dropbox, Box, and OneDrive.

Since 1994, the mission of SmartDraw Software has been to expand the ways in which people communicate so that we can clearly understand each other, make informed decisions, and work together to improve our businesses and the world. We accomplish this by creating software and services that make it possible for people to capture and present information as visuals, while being a pleasure to use. In 2019, we took this to the next level by launching Visual Script, which makes it easy to visualize data in relational formats like trees, flows, and timelines, automatically, without any human input. Visual Script is a relationship visualization platform that empowers organizations to visualize data across siloed ecosystems and gain critical insights in real-time. Today, SmartDraw Software is one of the most sophisticated digital marketing organizations in the world with over 90,000 unique visitors to our website each business day and in excess of 3,000,000 installations of our apps each year. SmartDraw is used by more than half of the Fortune 500 and by over 250,000 public and private enterprises of all sizes around the world. Privately held, SmartDraw Software is headquartered in San Diego, California.

2.6.2 Hardware requirements

The selection of hardware configuring is a very task related to the software development, particularly inefficient RAM may affect adversely on the speed and corresponding on the efficiency of the entire system. The processor should be powerful to handle all the operations. The hard disk should have the sufficient to solve the database and the application.

Hardware used for development:

CPU : Intel i3 Processor
Memory : 4 GB
Cache : 6 MB
Hard Disk : 1 TB
Monitor : 15.6" Monitor
Keyboard : Standard 108 keys Enhanced Keyboard
Mouse : Optical Mouse

Minimum Hardware Required For Implementation:

CPU : Pentium IV Processor
Memory : 256 MB Above
Cache : 512 KB Above
Hard Disk : 20 GB Above
Monitor : Any
Keyboard : Any
Mouse : Any

SYSTEM DESIGN

3.1 SYSTEM ARCHITECTURE

A system architecture or systems architecture is the conceptual model that defines the structure, behaviour, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures of the system.

System architecture can comprise system components, the externally visible properties of those components, the relationships (e.g. the behaviour) between them. It can provide a plan from which products can be procured, and systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture; collectively these are called architecture description languages (ADLs).

The system architecture can best be thought of as a set of representations of an existing (or to be created) system. It is used to convey the informational content of the elements comprising a system, the relationships among those elements, and the rules governing those relationships. The architectural components and set of relationships between these components that architecture describes may consist of hardware, software, documentation, facilities, manual procedures, or roles played by organizations or people. System architecture is primarily concerned with the internal interfaces among the system's components or subsystems, and the interface between the system and its external environment, especially the user.

The structural design reduces complexity, facilitates change and result in easier implementation by encouraging parallel development of different parts of the system. The procedural design transforms structural elements of program architecture into a procedural description of software components. The architectural design considers architecture as the most important functional requirement. The system is based on the three-tier architecture. The first level is the user interface (presentation logic), which displays controls, receives and validates user input. The second level is the business layer (business logic) where the application specific logic takes place. The third level is the data layer where the application information is stored in files or database. It contains logic about to retrieve and update data. The important feature about the three-tier design is that information only travels from one level to an adjacent level.

3.2 MODULE DESIGN

Modular programming is a software design technique that emphasizes separating the functionality of a program into independent, interchangeable modules, such that each contains everything necessary to execute only one aspect of the desired functionality. Conceptually, modules represent a separation of concerns, and improve maintainability by enforcing logical boundaries between components.

Different modules of this project includes

- **User Authentication**

In this module we have included the login verification and the user management. Admin

and family have different username and password. They can login the website.

- **Ward and family Management**

In this module we include Ward registration and Family registration. The admin can register the details about ward and family members. They can complete register for Birth Year, Death Year, Wedding Year of the family members.

- **Member Management**

In this module the complete details about all the members in the church are registered. The member details such as wedding registration, engagement registration etc. The admin can view and approve the details given by the members.

- **Activity and Certificate Management**

This module contain the details about certificate management. Family can send request to the admin to retrieve the certificate in a required date. The admin can generate the certificate and send back to the family.

- **Report Generation**

Admin can easy to view the ward wise family details, member details according to the family and count the family according to each ward. This module have the report of the church based on the ward details, family details etc.. are provided. This module also allows the members to see their individual details such as Birth certificate.

3.3 DATABASE DESIGN

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to

make information access easy, quick, inexpensive and flexible for the users. The general theme behind a database is to integrate all information. Database design is recognized as a standard of management information system and is available virtually for every computer system. In database design several specific objectives are considered:

- Ease of learning and use
- Controlled redundancy
- Data independence
- More information at low cost
- Accuracy and integrity
- Recovery from failure
- Privacy and security
- Performance

A database is an integrated collection of data and provides centralized access to the data. Usually the centralized data managing the software is called RDBMS. The main significant difference between RDBMS and other DBMS is the separation of data as seen by the program and data has in direct access to stores device. This is the difference between logical and physical data.

3.3.1 Normalization

Designing a database is complete task and the normalization theory is a useful aid in the design process. The process of normalization is concerned with transformation of conceptual schema into computer representation form. There will be need for most databases to grow by adding new attributes and new relations. The data will be used in new ways. Tuples will be added and deleted. Information stored may undergo updating also. New association may also be added. In such situations the performance of a database is entirely depend upon its design. A bad database design may lead to certain undesirable things like:

- Repetition of information
- Inability to represent certain information
- Loss of information

To minimize these anomalies, Normalization may be used. If the database is in a normalized form, the data can be growing without, in most cases, forcing the rewriting application programs. This is important because of the excessive and

growing cost of maintaining an organization's application programs and its data from the disrupting effects of database growth. As the quality of application programs increases, the cost of maintaining the without normalization will rise to prohibitive levels. A normalized database can also encompass many related activities of an organization thereby minimizing the need for rewriting the applications of programs. Thus, normalization helps one attain a good database design and there by ensures continued efficiency of database.

Normalization theory is built around the concept of normal forms. A relation is said to be in normal form if it satisfies a certain specified set of constraints. For example, a relation is said to be in first normal form (1NF) if it satisfies the constraint that it contains atomic values only. Thus every normalized relation is in 1NF. Numerous normal forms have been defined. Codd defined the first three normal forms.

All normalized relations are in 1NF, some 1NF relations are also in 2NF and some 2NF relations are also in 3NF. 2NF relations are more desirable than 1NF and 3NF are more desirable than 2NF. That is, the database designer should prefer 3NF than 1NF or 2NF. Normalization procedure states that a relation that is in some given normal form can be converted into a set of relations in a more desirable form. We can define this procedure as the successive reduction of a given collection of relations to some more desirable form. This procedure is reversible. That is, it is always possible to take the output from the procedure and convert them back into input. In this process, no information is lost. So it is also called "no loss decomposition".

First Normal Form

A relation is in first normal form (1NF) if and all its attributes are based on single domain. The objective of normalizing a table is to remove its repeating groups and ensure that all entries of the resulting table have at most single value.

Second Normal Form

A table is said to be second Normal Form (2NF), when it is in 1NF and every attribute in record is functionally dependent upon the whole key, and not just a part of the key.

Third Normal Form

A table is in third Normal Form (3NF), when it is in 2NF and every non-

key attribute is functionally dependent on just the primary key.

3.3.2 Table Structure

Table is a collection of complete details about a particular subject. These data are saved in rows and Columns. The data of each Row are different units. Hence, rows are called RECORDS and Columns of each row are called FIELDS.

Data is stored in tables, which is available in the backend the items and data, which are entered in the input, form id directly stored in this table using linking of database. We can link more than one table to input forms. We can collect the details from the different tables to display on the output. There are mainly 12 tables in the project. They are,

1. tbl_admin
2. tbl_ward
3. tbl_family
4. tbl_member
5. tbl_priest
6. tbl_vicar
7. tbl_holycommunion
8. tbl_engagement
9. tbl_wedding
10. tbl_death
11. tbl_certificatemaster
12. tbl_certificaterequest

Table design

1. Table Name : tbl_admin

Primary key : Nil

Foreign key : Nil

Table 3.1 tbl_admin

SL No	Field Name	Data type	Constraint

1	Username	VARCHAR(30)	Not Null
2	Password	VARCHAR(30)	Not Null

2. Table Name :
tbl_ward

Primary key : WardId

Foreign key : Nil

Table 3.2 tbl_ward

SL No	Field Name	Data type	Constraint
1	WardId	INT	Primary Key
2	WardName	VARCHAR(20)	Not Null
3	WardShortName	VARCHAR(10)	Not Null
4	WardDiscription	VARCHAR(100)	Not Null
5	WardStatus	INT(10)	Not Null

3. Table Name : tbl_family

Primary key : FamilyId

Foreign key : WId

Table 3.3 tbl_family

SL No	Field Name	Data type	Constraint
1	FamilyId	INT	Primary Key
2	WId	INT	Foreign Key
3	FamilyRegNo	INT	Not Null
4	FamilyName	VARCHAR(30)	Not Null
5	FamilyHead	VARCHAR(20)	Not Null
6	UserName	VARCHAR(20)	Not Null
7	Pwd	VARCHAR(20)	Not Null
8	Contact	BIGINT	Not Null

9	FamilyStatus	INT(10)	Not Null
---	--------------	---------	----------

4. Table Name : tbl_member

Primary key : MemberId

Foreign key : Fid

Table 3.4 tbl_member

SL No	Field Name	Data type	Constraint
1	MemberId	INT	Primary key
2	MemberNm	VARCHAR(30)	Not Null
3	MFName	VARCHAR(30)	Not Null
4	MMName	VARCHAR(30)	Not Null
5	MEmail	VARCHAR(50)	Not Null
6	MGender	VARCHAR(20)	Not Null
7	MStatus	VARCHAR(50)	Not Null
8	MDob	DATE	Not Null
9	Fid	INT	Foreign Key
10	RWithHead	VARCHAR(30)	Not Null
11	BName	VARCHAR(30)	Not Null
12	BDate	DATE	Not Null
13	BPriest	VARCHAR(30)	Not Null
14	BChurch	VARCHAR(50)	Not Null
15	BDetails	VARCHAR(30)	Not Null
16	ConNo	BIGINT	Not Null
17	MemberStatus	INT	Not Null

5. Table Name : tbl_priest

Primary key : PriestId

Foreign Key : FReg

Table 3.5 tbl_priest

SL No	Field Name	Data type	Constraint
1	PriestId	INT	Primary key
2	PriestName	VARCHAR(30)	Not Null
3	DOB	DATE	Not Null
4	HouseNm	VARCHAR(50)	Not Null
5	FReg	INT	Foreign Key
6	Dio	VARCHAR(20)	Not Null
7	Congregation	VARCHAR(50)	Not Null
8	DateON	DATE	Not Null
9	PN	VARCHAR(20)	Not Null
10	Con	BIGINT(10)	Not Null
11	PriestStatus	INT	Not Null

6. Table Name : tbl_vicar

Primary key : VicarId

Foreign key : Nil

Table 3.6 tbl_vicar

SL No	Field Name	Data type	Constraint
1	VicarId	INT	Primary Key
2	VicarName	VARCHAR(30)	Not Null
3	HouseName	VARCHAR(30)	Not Null
4	Diocese	VARCHAR(30)	Not Null
5	YearofJoining	BIGINT(10)	Not Null
6	YearofLeaving	BIGINT(10)	Not Null

7	DateofBirth	DATE	Not Null
8	FeastDay	DATE	Not Null
9	ContactNo	BIGINT(10)	Not Null
10	VicarStatus	INT	Not Null

7. Table Name : tbl_holycommounion

Primary key : HolyId

Foreign key : MemId

Table 3.7 tbl_holycommounion

SL No	Field Name	Data type	Constraint
1	HolyId	INT	Primary Key
2	HolyDate	DATE	Not Null
3	MemId	INT	Foreign Key
4	HolyPriest	VARCHAR(50)	Not Null
5	HolyChurch	VARCHAR(50)	Not Null
6	HolyStatus	INT	Not Null

8. Table Name : tbl_engagement

Primary key : EngId

Foreign key : EngMID

Table 3.8 tbl_engagement

SL No	Field Name	Data type	Constraint
1	EngId	INT	Primary Key
2	EngMID	INT	Foreign Key
3	EngPN	VARCHAR(30)	Not Null
4	EngDate	Date	Not Null

5	EngPFN	VARCHAR(30)	Not Null
6	EngPMN	VARCHAR(30)	Not Null
7	EngPBN	VARCHAR(20)	Not Null
8	EngPriest	VARCHAR(30)	Not Null
9	EngChurch	VARCHAR(50)	Not Null
10	EngWitness	VARCHAR(30)	Not Null
11	EngStatus	VARCHAR(20)	Not Null

9. Table Name : tbl_wedding

Primary key : WeddId

Foreign key : FamId

Table 3.9 tbl_wedding

SL No	Field Name	Data type	Constraint
1	WeddId	INT	Primary Key
2	FamId	INT	Foreign Key
3	Groom	VARCHAR(30)	Not Null
4	Bride	VARCHAR(30)	Not Null
5	BParish	VARCHAR(30)	Not Null
6	BDiocese	VARCHAR(30)	Not Null
7	WChurch	VARCHAR(30)	Not Null
8	BFamilyName	VARCHAR(30)	Not Null
9	WDate	DATE	Not Null
10	WPriest	VARCHAR(30)	Not Null
11	WW1	VARCHAR(30)	Not Null
12	WW2	VARCHAR(50)	Not Null
13	WStatus	VARCHAR(20)	Not Null

10. Table Name : tbl_death

Primary key : DeathId

Foreign key : DMId

Table 3.10 tbl_death

SL No	Field Name	Data type	Constraint
1	DeathId	INT	Primary Key
2	DeathDate	DATE	Not Null
3	DeathFdate	DATE	Not Null
4	DPriest	VARCHAR(50)	Not Null
5	DChurch	VARCHAR(50)	Not Null
6	Comment	VARCHAR(60)	Not Null
7	DMId	INT	Foreign Key
8	DeathStatus	VARCHAR(20)	Not Null

11. Table Name : tbl_certificatemaster

Primary key : Primary Key

Foreign key : Nil

Table 3.11 tbl_certificatemaster

SL No	Field Name	Data type	Constraint
1	CertificateId	INT	Primary Key
2	CertificateType	VARCHAR(20)	Not Null
3	CertificateName	VARCHAR(50)	Not Null
4	CDescription	VARCHAR(50)	Not Null
5	CStatus	INT	Not Null

12. Table Name : tbl_certificaterequest

Primary key : CertRequestId

Foreign key : CMId, CType

Table 3.12 tbl_certificaterequest

SL No	Field Name	Data type	Constraint
1	CertRequestId	INT	Primary key
2	CMId	INT	Foreign key
3	CType	VARCHAR(30)	Foreign key

4	RequestPurpose	VARCHAR(30)	Not Null
5	RequiredDate	DATE	Not Null
4	CertStatus	VARCHAR(30)	Not Null

3.3.3 Data Flow Diagram

3.3.3.1 Introduction to Data Flow Diagrams

Data Flow Diagram is a network that describes the flow of data and processes that change, or transform, data throughout the system. This network is constructed by use a set of symbols that do not imply a physical implementation. It is a graphical tool for structured analysis of the system requirements. DFD models a system by using external entities from which data flows to a process, which transforms the data and creates, output-data-flows which go to other processes or external entities or files. Data in files may also flow to processes as inputs. There are various symbols used in a DFD. Bubbles represent the processes. Named arrows indicate the data flow. External entities are represented by rectangles. Entities supplying data are known as sources and those that consume data are called sinks. Data are stored in a data store by a process in the system. Each component in a DFD is labelled with a descriptive name. Process names are further identified with a number. The Data Flow Diagram shows the logical flow of a system and defines the boundaries of the system. For a candidate system, it describes the input (source), outputs (destination), database (files) and procedures (data flow), all in a format that meet the user's requirements. The main merit of DFD is that it can provide an overview of system requirements, what data a system would process, what transformations of data are done, what files are used, and where the results flow. This network is constructed by use a set of symbols that do not imply a physical implementation. It is a graphical tool for structured analysis of the system requirements. DFD models a system by using external entities from which data flows to a process, which transforms the data and creates, output-data-flows which go to other processes or external entities or files. External entities are represented by rectangles. Entities supplying data are known as sources and those that consume data are called sinks. Data are stored in a data store by a process in the system. It is a graphical tool for structured analysis of the system requirements. DFD models a system by using external entities from which data flows to a process, which


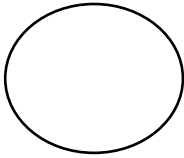


transforms the data and creates, output-data-flows which go to other processes or external entities or files. Data in files may also flow to processes as inputs.

Rules for constructing a Data Flow Diagram

1. Arrows should not cross each other
2. Squares, circles and files must bear names.
3. Decomposed data flow squares and circles can have same time
4. Choose meaningful names for data flow

Draw all data flows around the outside of the diagram

Basic Data Flow Diagram Symbols

	<p>A data flow is a route, which enables packets of data to travel from one point to another. Data may flow from a source to a process and from data store or process. An arrow line depicts the flow, with arrow head pointing in the direction of the flow.</p>
	<p>Circles stands for process that converts data in to information. A process represents transformation where incoming data flows are changed into outgoing data flows.</p>
	<p>A data store is a repository of data that is to be stored for use by a one or more process may be as simple as buffer or queue or sophisticated as relational database. They should have clear names. If a process merely uses the content of store and does not alter it, the arrowhead goes only from the store to the process. If a process alters the details in the store then a double-headed arrow is used.</p>
	<p>A source or sink is a person or part of an organization, which enters or receives information from the system, but is considered to be outside the contest of data flow model.</p>

3.3.3.2 Data Flow Diagram

Each component in a DFD is labelled with a descriptive name. Process name are further identified with number. Context level DFD is draw first. Then the process is decomposed into several elementary levels and is represented in the order of importance. A DFD describes what data flow (logical) rather than how they are processed, so it does not depend on hardware, software, and data structure or file organization.

A DFD methodology is quite effective; especially when the required design.

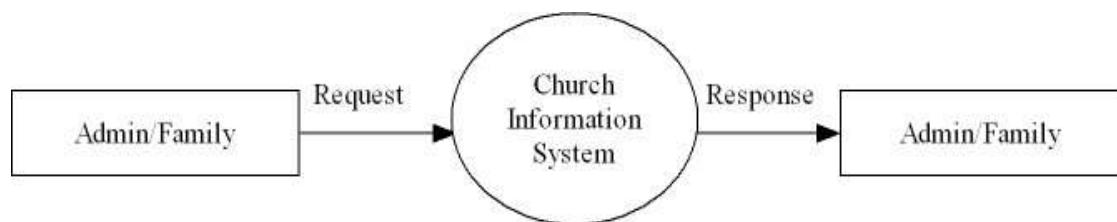


Fig 3.1 Level 0 DFD for Church Information System

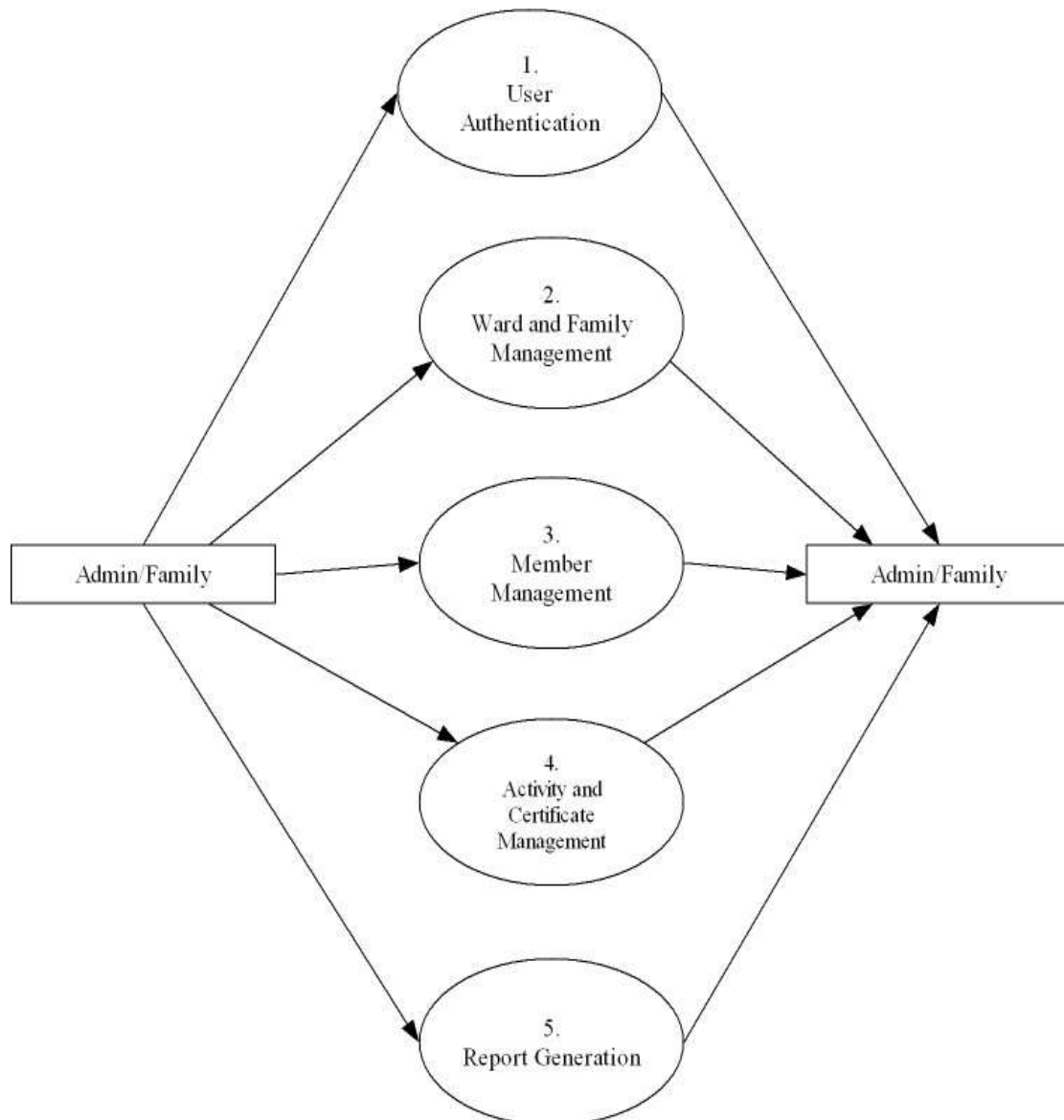


Fig 3.2 First Level Data Flow Diagram for Church Information System

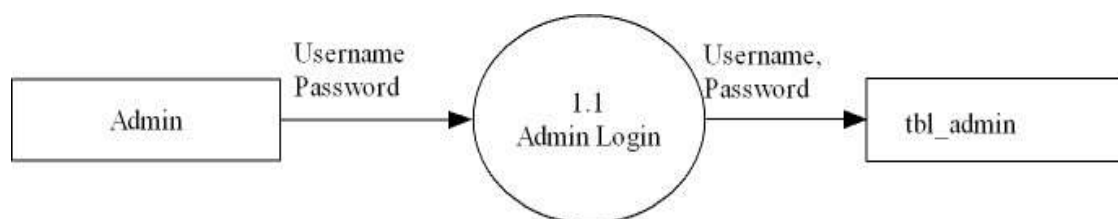


Fig 3.3 Second Level Data Flow Diagram for Admin Login

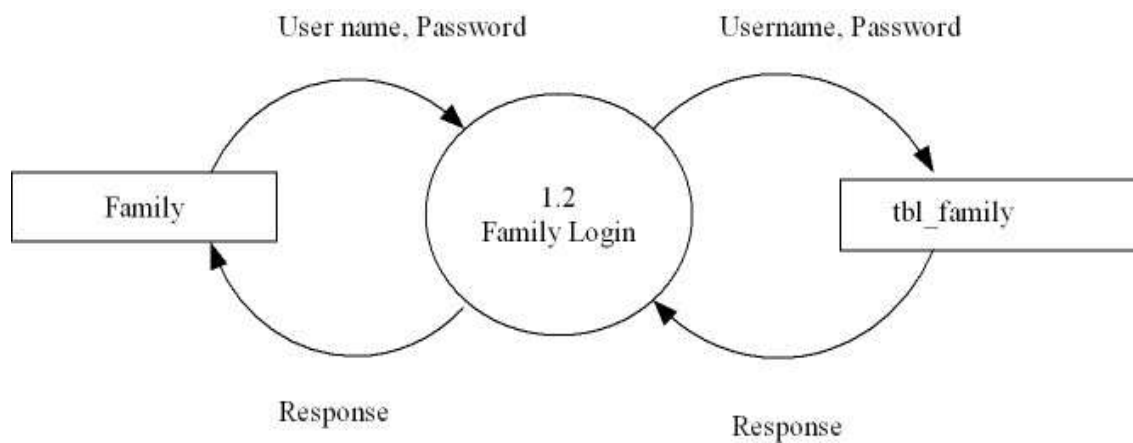


Fig 3.4 First Level Data Flow Diagram for Family Login

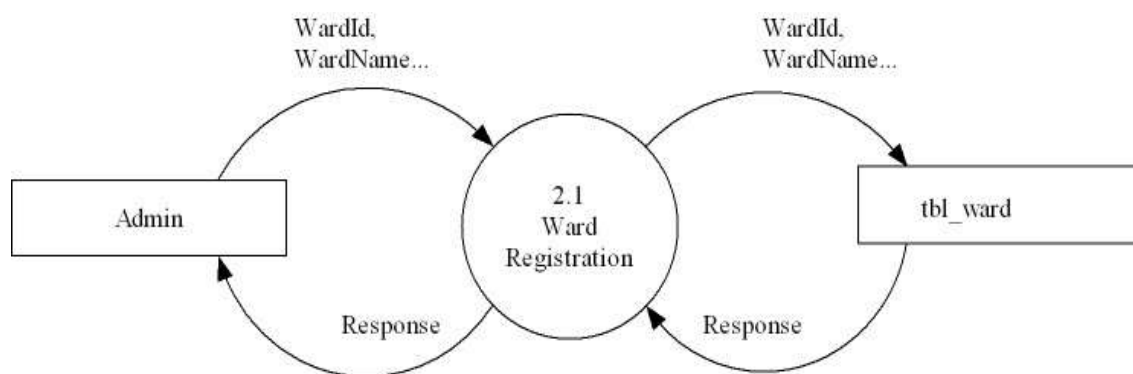


Fig 3.5 Second Level Data Flow Diagram for Ward Registration

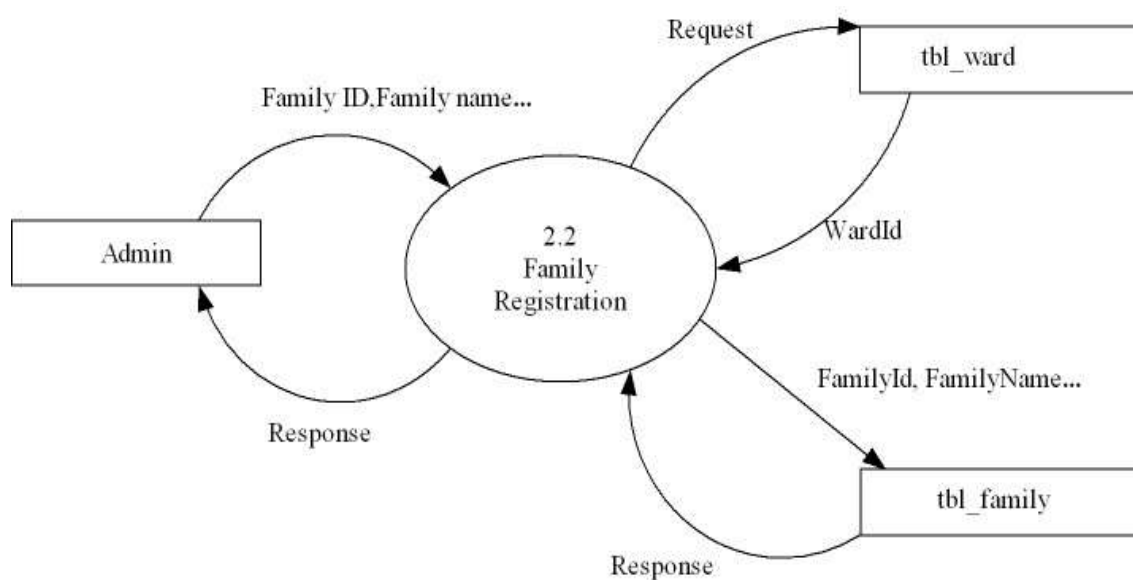


Fig 3.6 Second Level Data Flow Diagram for Family Registration

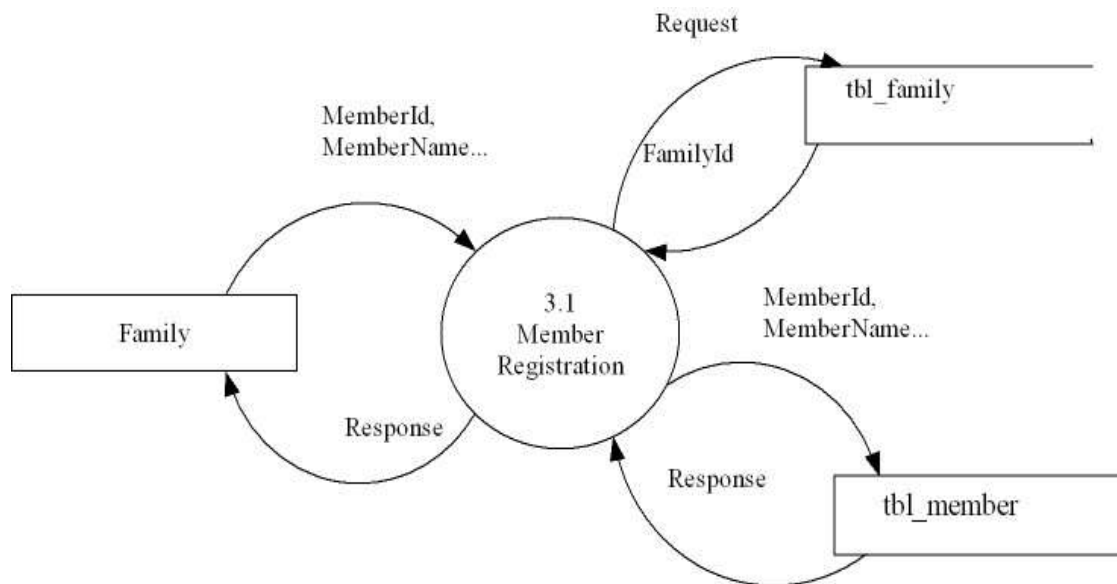


Fig 3.7 Second Level Data Flow Diagram for Member Registration

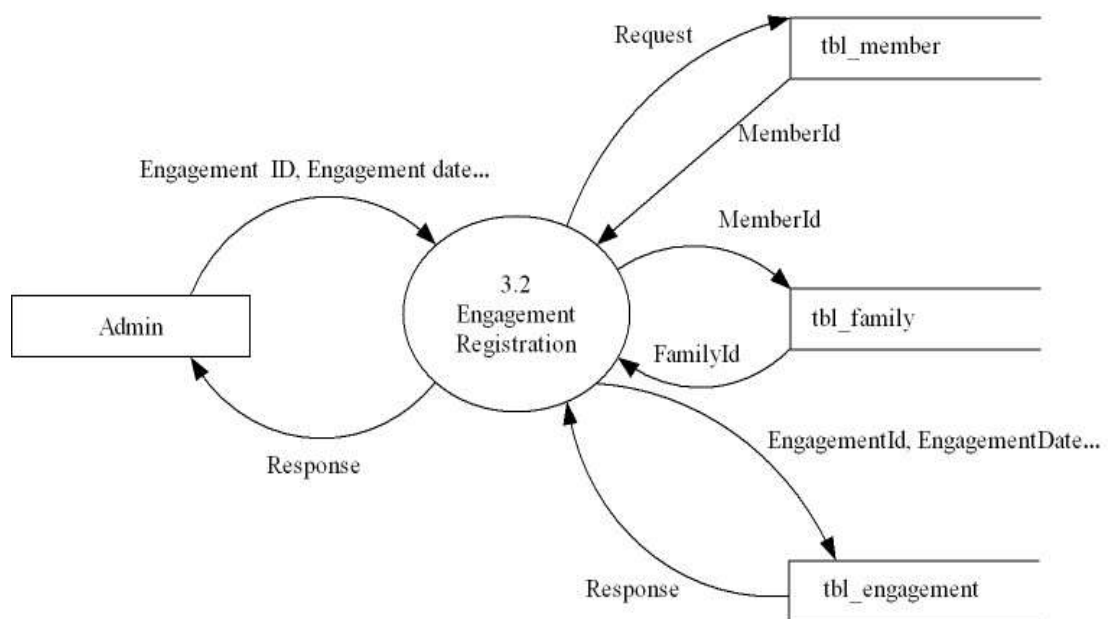


Fig 3.8 Second Level Data Flow Diagram for Engagement Registration

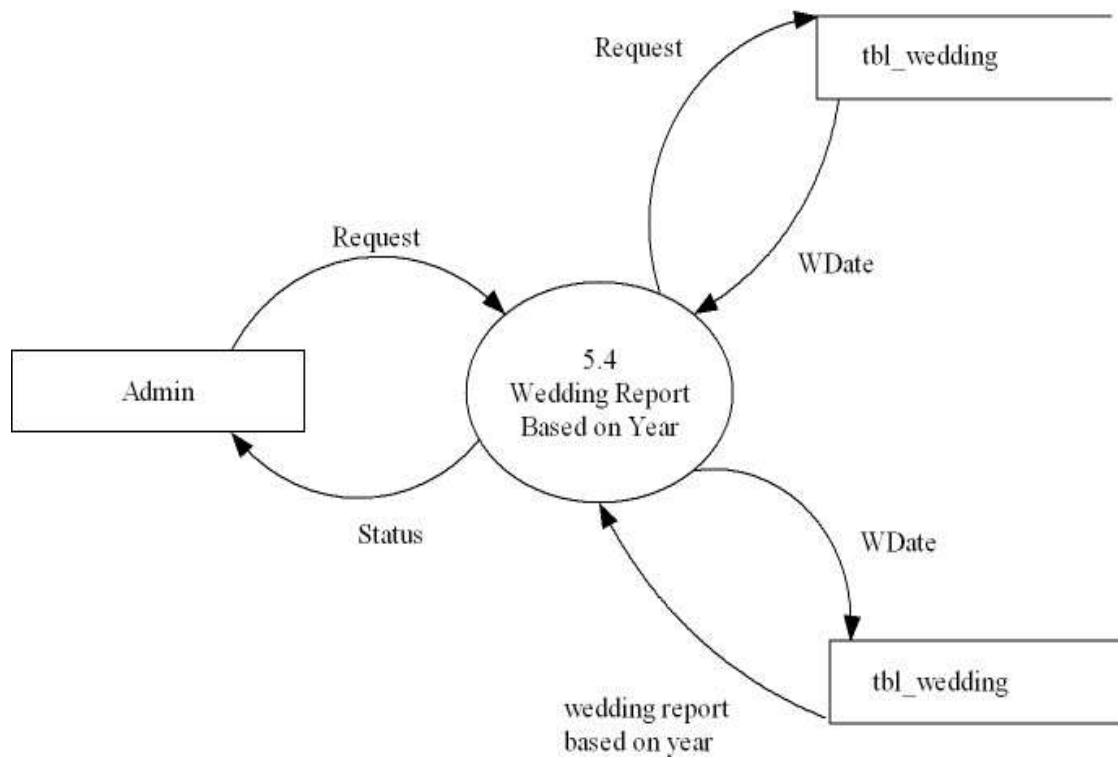


Fig 3.9 Second Level Data Flow Diagram for Wedding Registration

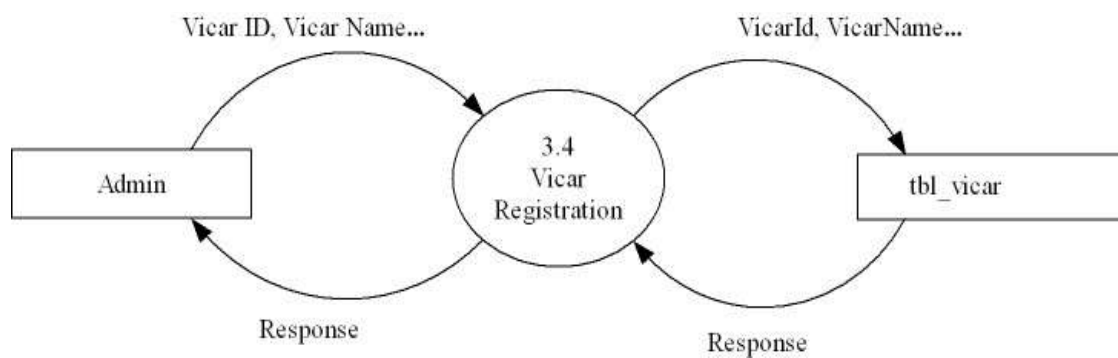


Fig 3.10 Second Level Data Flow Diagram for Vicar Registration

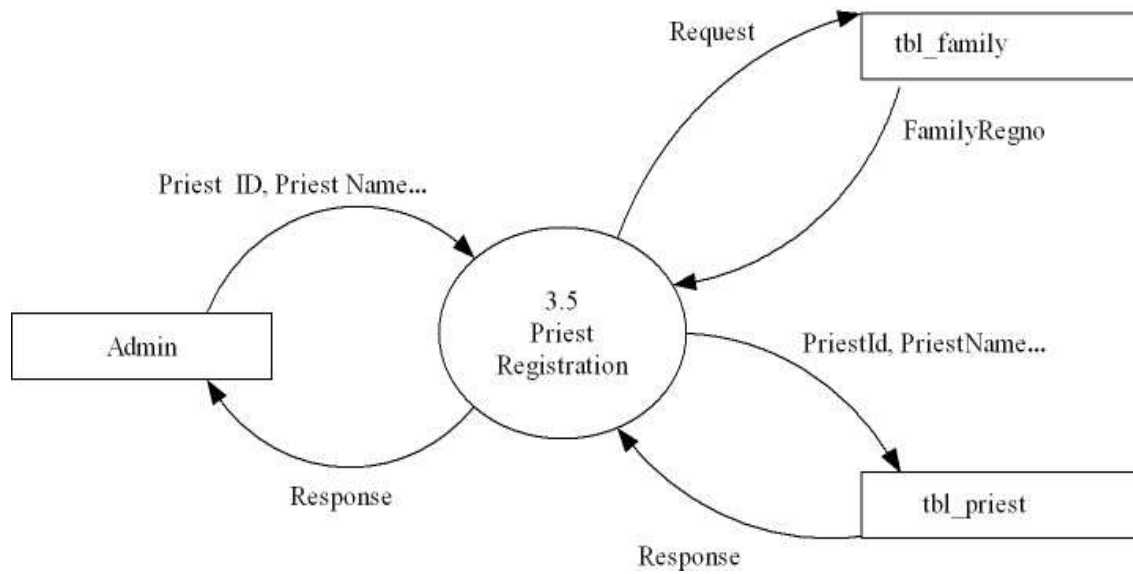


Fig 3.11 Second Level Data Flow Diagram for Priest Registration

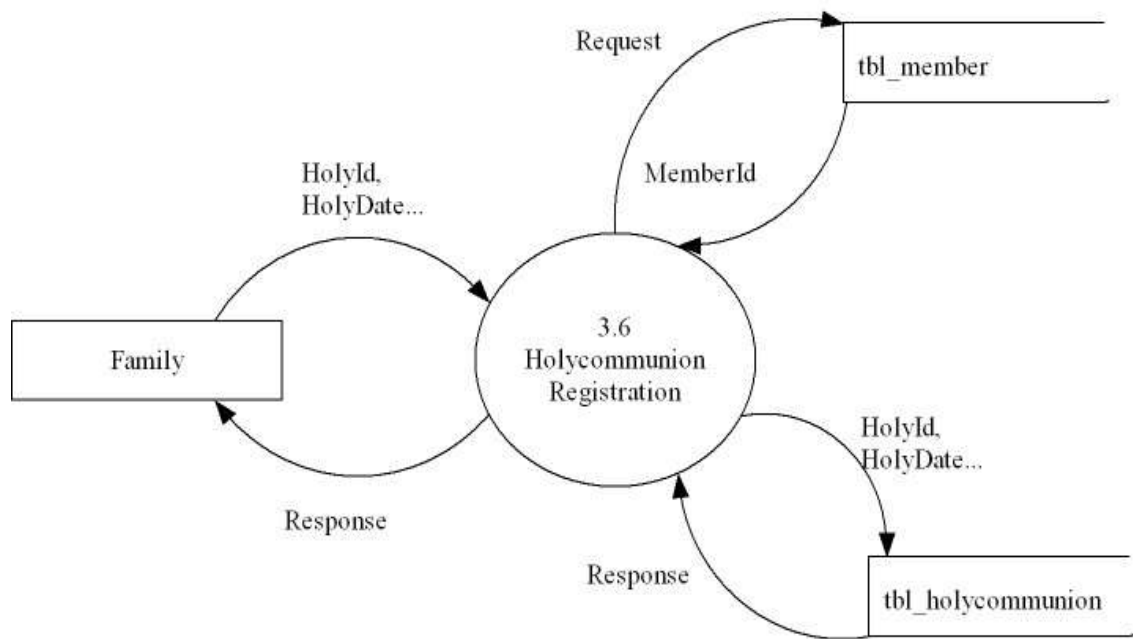


Fig 3.12 Second Level Data Flow Diagram for Holycommunion Registration

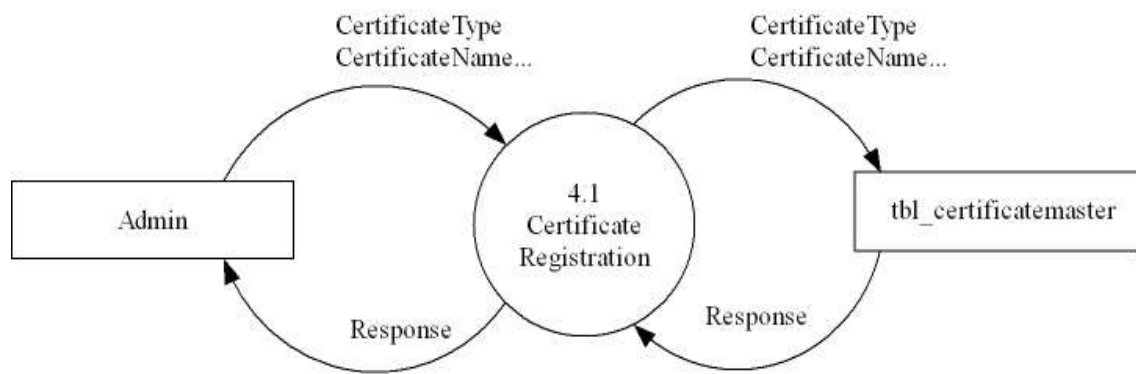


Fig 3.13 Second Level Data Flow Diagram for Certificate Registration

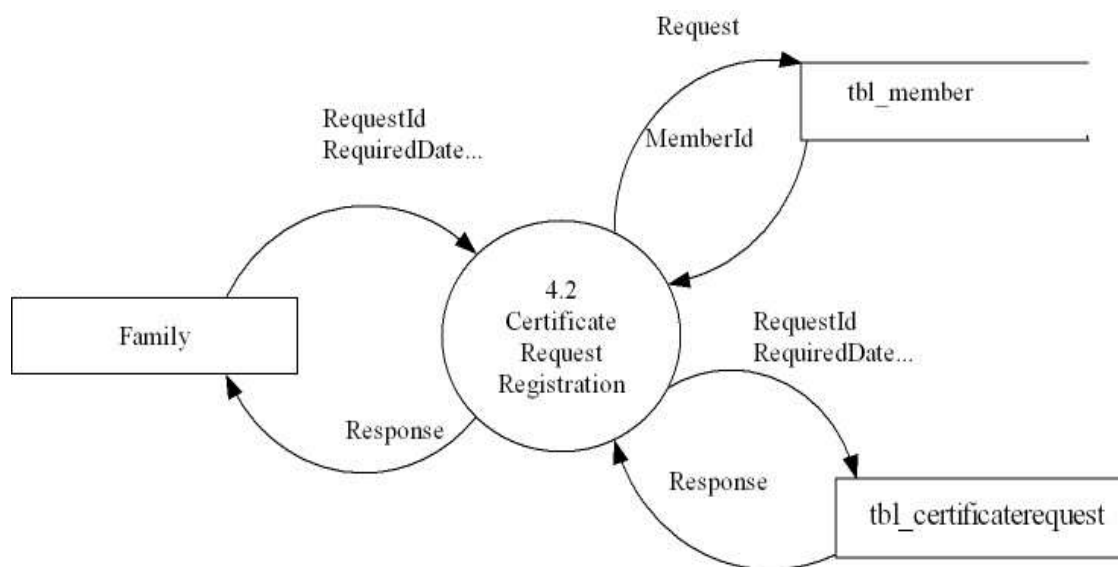


Fig 3.14 Second Level Data Flow Diagram for Certificate Request Registration

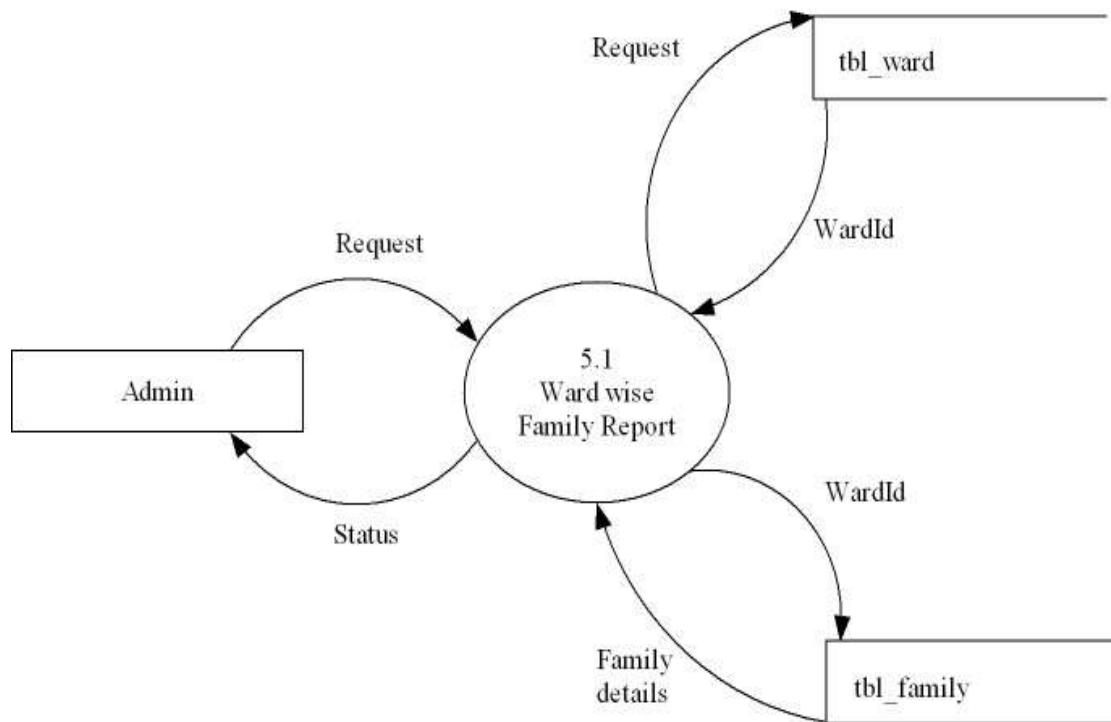


Fig 3.15 Second Level Data Flow Diagram for Ward wise Family Report

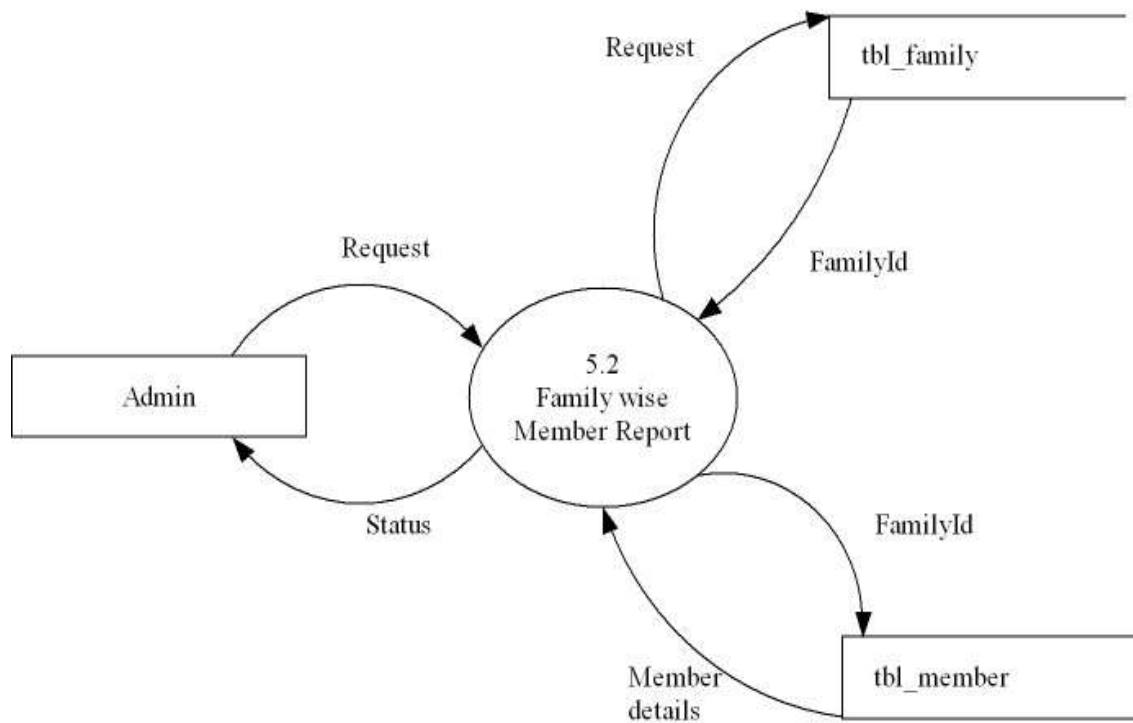


Fig 3.16 Second Level Data Flow Diagram for Family wise Member Report

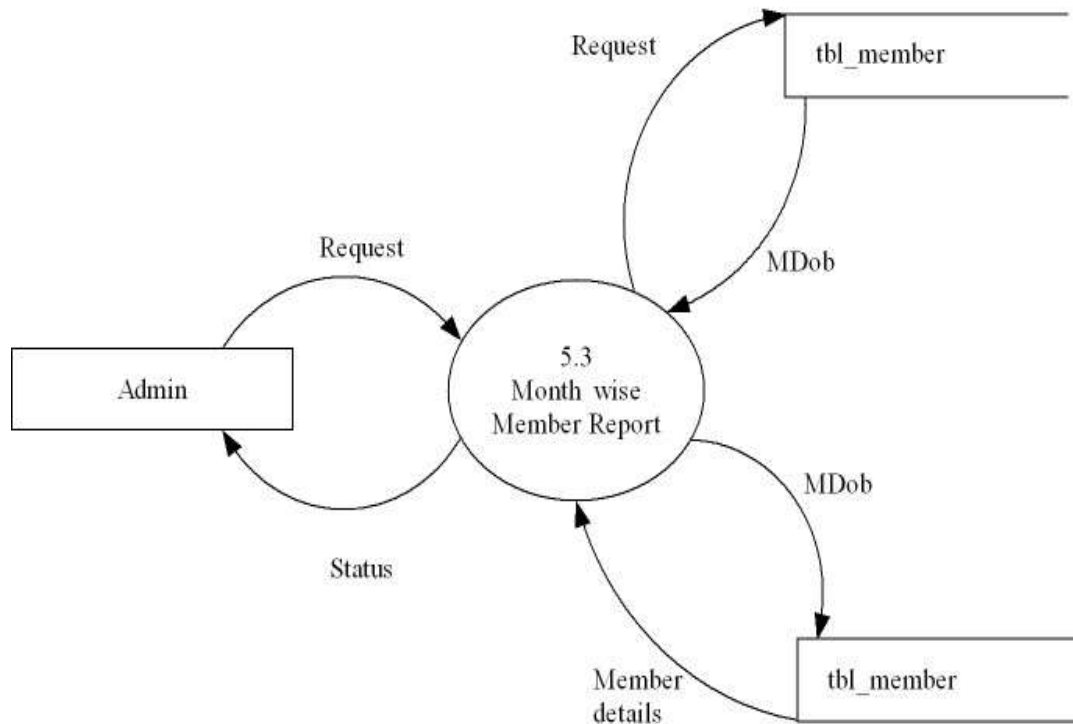


Fig 3.17 Second Level Data Flow Diagram for Month wise Member Report

3.4 INTERFACE DESIGN

These modules can apply to hardware, software or the interface between a user and a machine. An example of a user interface could include a GUI, a control panel for a nuclear power plant, or even the cockpit of an aircraft. In systems engineering, all the inputs and outputs of a system, subsystem, and its components are listed in an interface control document often as part of the requirements of the engineering project. The development of a user interface is a unique field.

3.4.1 User Interface Screen Design

The user interface design is very important for any application. The interface design describes how the software communicates within itself, to system that interpreted with it and with humans who use it. The input design is the process of converting the user-oriented inputs into the computer based format. The data is fed into the system using simple inactive forms. The forms have been supplied with messages so that the user can enter data without facing any difficulty. They data is validated wherever it requires in the project. This ensures that only the correct data have been incorporated into system. The goal of designing input data is to make the automation as easy and free from errors as possible. For providing a good input

design for the application easy data input and selection features are adopted. The input design requirements such as user friendliness, consistent format and interactive dialogue for giving the right messages and help for the user at right are also considered for development for this project.

Input Design is a part of the overall design. The input methods can be broadly classified into batch and online. Internal controls must be established for monitoring the number of inputs and for ensuring that the data are valid. The basic steps involved in input design are:

- Review input requirements.
- Decide how the input data flow will be implemented.
- Decide the source document.
- Prototype on line input screens.
- Design the input screens.

The quality of the system input determines the quality of the system output. Input specifications describe the manner in which data enter the system for processing. Input design features can ensure the reliability of the system and produce results from accurate data. The input design also determines whether the user can interact efficiently with the system.

Here is a sample input form:

Next Big Event: 999 days 10 hours 59 minutes 41 seconds

Faith

Home Registration Approvals Certificate Requests Reports Logout

FAMILY REGISTRATION

Family Name: Marathi

Family Register Number: 1009

Name: St.Joseph

Family Head: Jay

Contact Number: 997689678

Username: Jay

Password: ****

Save

Fig 3.22 Family registration form

This input form is for the registration of new user. It contains textboxes for inputting name, house name, phone number, district, location, email id and a password box for giving the password. The new user can use the facilities provided to them only after becoming a part of the system. After clicking the submit button the user will get a registration successful mail and after the approval of admin. After signing up the user can login to the system directly through the login page with the email id and password.

3.4.2 Output Design

A quality output is one, which meets the requirements of end user and presents the information clearly. In any system result of processing are communicated to the user and to the other system through outputs. In the output design it is determined how the information is to be displayed for immediate need.

It is the most important and direct source information is to the user. Efficient and intelligent output design improves the system's relationships with the user and helps in decision -making. The objective of the output design is to convey the information of all the past activities, current status and to emphasis important events. The output generally refers to the results and information that is generated from the system. Outputs from computers are required primarily to communicate the results of processing to the users.

Output also provides a means of storage by copying the results for later reference in consultation. There is a chance that some of the end users will not actually operate the input data or information through workstations, but will see the output from the system.

Two phases of the output design are:

1. Output Definition
2. Output Specification

Output Definition takes into account the type of output contents, its frequency and its volume, the appropriate output media is determined for output. Once the media is chosen, the detail specification of output documents are carried out. The nature of output required from the proposed system is determined during logical design stage. It takes the outline of the output from the logical design and produces output as specified during the logical design phase.

In a project, when designing the output, the system analyst must accomplish the following:

- Determine the information to present.
 - Decide whether to display, print, speak the information and select the output medium.
 - Arrange the information in acceptable format.
 - Decide how to distribute the output to the intended receipt.
 - Thus by following the above specifications, a high quality output can be generated.
- In my projects outputs are generated as excel files and pie charts.

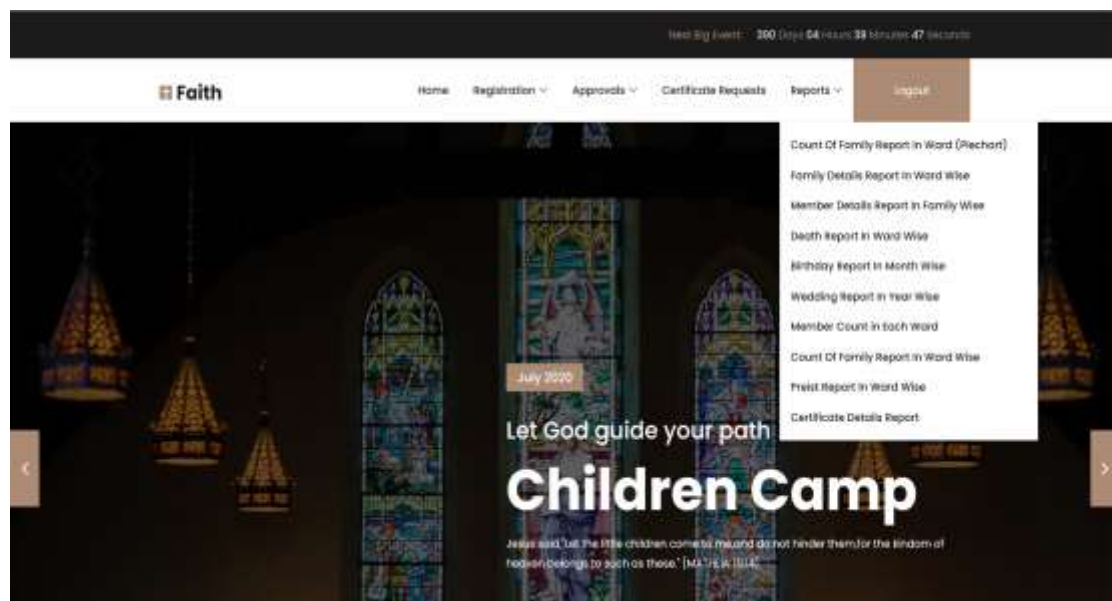


Fig 3.23 Various reports on the admin page

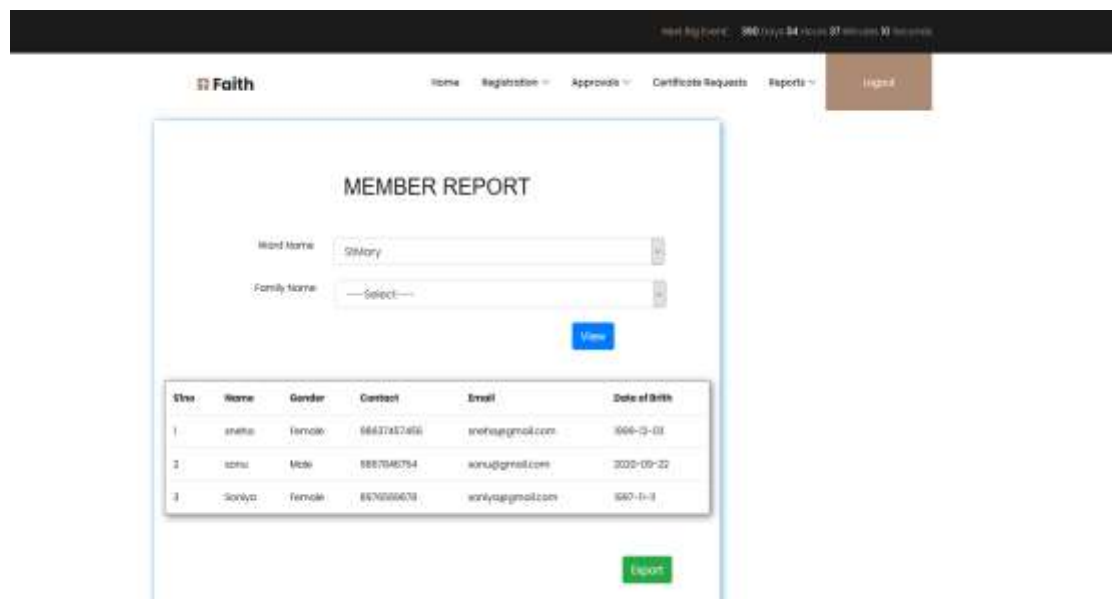


Fig 3.24 Reports on member details in family wise

IMPLEMENTATION

Implementation is the stage of the project when the theoretical design is turned into a working system. The implementation stage is a systems project in its own right. It includes careful planning, investigation of current system and its constraints on implementation, design of methods to achieve the changeover, training of the staff in the changeover procedure and evaluation of changeover method.

4.1 CODING STANDARDS

PHP follows few rules and maintains its style of coding. As there are many coders and developers all over the world, so each of them can follow different coding styles and standards but this would have raised great confusion and difficulty for a developer to understand another developers code. It would have been hard to manage and store the code for future reference. Here is where the coding standards come into play. This not only makes a code easy to read but also makes the code very easy to refer in the future. This makes the code understandable and clearer to decipher, just like a blueprint. This also makes the code more formal and industry or software oriented. Below mentioned are few guidelines that one must follow in order to maintain the standard of PHP coding.

1. PHP tags: One must use the PHP standard tags(), rather than the shorthand tags() to delimit the PHP code.

2. Commenting: Use of standard C and C++ commenting style i.e., (//) – for single line and (/* */) – for multi-line, is highly encouraged and use of Python or Perl style of commenting i.e., (#), is discouraged.

3. Line length and Indentation: It is a standard recommendation to not exceed more than 75-85 characters per line of code. One must not use tabs for indentation instead use 4 spaces as it is the standard indenting method in most of the programming languages.

4. Structuring the control flow statements: The control flow or conditional statements must be written in such a way so that it could be differentiated from function call statements. While writing if, for, while, switch and other control flow statements there must be one space between the keyword and the opening parenthesis.

Example:

```
filter_none  
edit  
play_arrow
```



```
brightness_4
<?php
$n=5;
if($n> 0)
{
echo "Positive";
}
elseif ($n < 0)
{
echo "Negative";
}
Else {
echo "Zero";
}
?>
```

Output:

Positive

5. Function Calls: While writing a function call statement, there must be no space between the function name and the opening parenthesis.

Example:

```
filter_none
edit
play_arrow
brightness_4
<?php
echo testFunc(5, 6);
function testFunc($num1, $num2) {
$val = $num1 + $num2;
return $val;
}
?>
```

Output: 11

6. Naming Variables:

Here are few conventions that one must follow in order to name the variables:

- Use of lower case letters to name the variables.
- Use of ‘_’ to separate the words in a variable.
- Static variable names may be started with a letter ‘s’.
- Global variable names must start a with letter ‘g’.
- Use of upper-case letters to define global constants with ‘_’ as a separator.

7. Block alignment: Every block of code and curly braces must be aligned.

8. Short Functions: All functions and methods must limit themselves to a single page and must not be lengthy.

TESTING

Coding conventions are a set of guidelines for a specific programming language that recommend programming style, practices and methods for each aspect of a piece program written in this language. These conventions usually cover file organization, indentation, comments, declarations, statements, white space, naming conventions, programming practices, programming principles, programming rules of thumb, architectural best practices, etc. These are guidelines for software structural quality. Software programmers are highly recommended to follow these guidelines to help improve the readability of their source code and make software maintenance easier.

5.1 TEST CASES

The objective of system testing is to ensure that all individual programs are working as expected, that the programs link together to meet the requirements specified and to ensure that the computer system and the associated clerical and other procedures work together. The initial phase of system testing is the responsibility of the analyst who determines what conditions are to be tested, generates test data, produced a schedule of expected results, runs the tests and compares the computer produced results with the expected results with the expected results. The analyst may also be involved in procedures testing. When the analyst is satisfied that the system is working properly, he hands it over to the users for testing. The importance of system testing by the user must be stressed. Ultimately it is the user must verify the system and give the go-ahead.

During testing, the system is used experimentally to ensure that the software does not fail, i.e., that it will run according to its specifications and in the way users expect it to. Special test data is input for processing (test plan) and the results are examined to locate unexpected results. A limited number of users may also be allowed to use the system so analysts can see whether they try to use it in unexpected ways. It is preferably to find these surprises before the organization implements the system and depends on it. In many organizations, testing is performed by person other than those who write the original programs. Using persons who do not know how certain parts were designed or programmed ensures more complete and unbiased testing and more reliable software.

Parallel running is often regarded as the final phase of system testing. Since the parallel operation of two systems is very demanding in terms of user

resources it should be embarked on only if the user is satisfied with the results of testing -- it should not be started if problems are known to exist. Testing is the major quality control measure during software development. Its basic function is to detect errors in the software. Thus the goal of testing is to uncover requirement design and coding errors in the program.

Testing is the process of correcting a program with intends of finding an error. Different types of testing are,

1. Unit Testing
2. Integrated Testing
3. Black Box Testing
4. White Box Testing
5. Validation Testing
6. User Acceptance Testing

5.1.1 Unit Testing

In computer programming, unit testing is a method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures are tested to determine if they are fit for use In this testing we test each module individual and integrated the overall system. Unit testing focuses verification efforts on the smaller unit of software design in the module. This is also known as module testing. The modules of the system are tested separately. The testing is carried out during programming stage itself. In this testing step each module is found to working satisfactory as regard to the expected output from the module. There are some validation checks for verifying the data input given by the user which both the formal and validity of the entered. It is very easy to find error debug the system.

Fig 5.1 Unit Testing

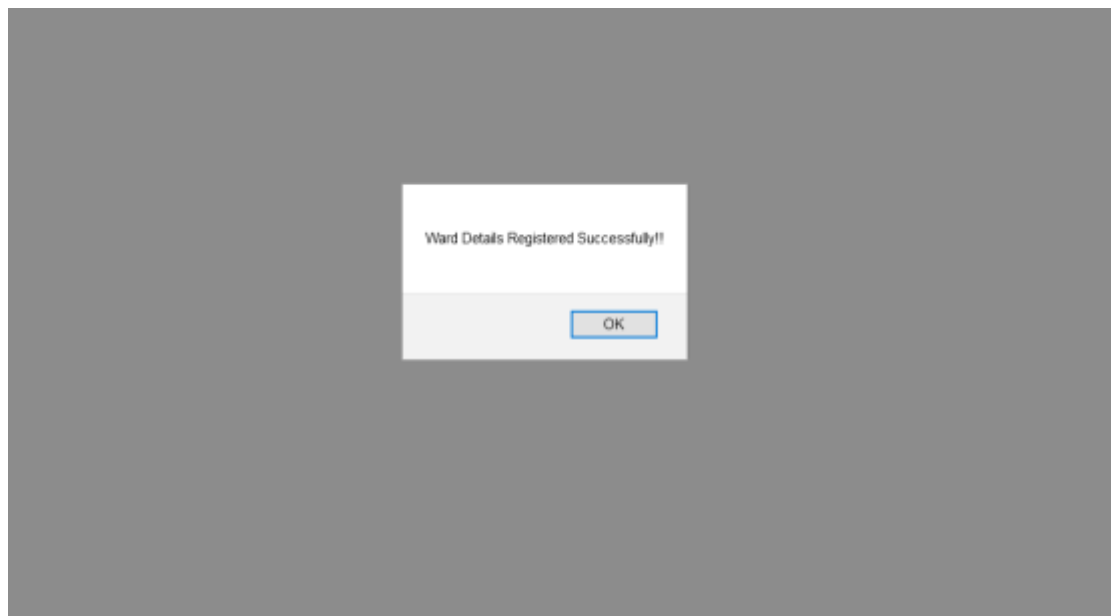


Fig 5.2 Unit Testing Result

We have continued Unit Testing from the starting of the coding phase itself. Whenever we completed one small sub module, some amount of testing was done based on the requirements to see if the functionality is aligned to the gathered requirements.

5.1.2 Integration Testing

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. Software components may be integrated in an iterative way or all together ("big bang"). Normally the former is considered a better

practice since it allows interface issues to be located more quickly and fixed. Data can be lost across an interface; one module can have an adverse effect on the other sub functions when combined by, may not produce the desired major functions. Integrated testing is the systematic testing for constructing the uncover errors within the interface. This testing was done with sample data. The developed system has run success full for this sample data. The need for integrated test is to find the overall system performance.

Integration testing is a logical extension of unit testing. In its simplest form, two units that have already been tested are combined into a component and the interface between them is tested. A component, in this sense, refers to an integrated aggregate of more than one unit. Integration testing identifies problems that occur when units are combined. By using a test plan that requires you to test each unit and ensure the viability of each before combining units, you know that any errors discovered when combining units are likely related to the interface between units. This method reduces the number of possibilities to a far simpler level of analysis. Progressively larger groups of tested software components corresponding to elements of the architectural design are integrated and tested until the software works as a system.

The screenshot shows a web application interface for 'Faith'. At the top, there is a navigation bar with links: Home, Registration, Approvals, Certificate Requests, Reports, and a Login button. Below the navigation bar, the main content area is titled 'FAMILY REGISTRATION'. It contains a form with the following fields and values:

Field	Value
Family Name	Mavall
Family Register Number	1000
Name	St. Joseph
Family Head	Joy
Contact Number	6679885678
Username	Joy
Password	****

A blue 'Save' button is located at the bottom right of the form.

Fig 5.3 Integration Testing



Fig 5.4 Integration Testing Result

We have performed integration testing whenever we have combined two modules together. When two modules are combined we have checked whether the functionality works correctly or not through integration testing.

5.1.3 Black Box Testing

Black-box testing is a method of software testing that examines the functionality of an application (e.g. what the software does) without peering into its internal structures or workings. This method of test can be applied to virtually every level of software testing: unit, integration, system and acceptance. It typically comprises most if not all higher level testing, but can also dominate unit testing as well. In black box testing the structure of the program is not considered. Test cases are decided solely on the basis of the requirements or the specification of the program or module, and the internals of the module or program are not considered for selection of the test cases.

In the Black Box testing tester only knows the input that can be given to the system and what output the system should give. In other words, the basis of deciding test cases in functional testing is requirements or specifications of the system or module. This form of testing is also called functional or behavioural testing. One advantage of the black box technique is that no programming knowledge is required. Whatever biases the programmers may have had, the tester likely has a different set and may emphasize different areas of functionality. On the other hand, black-box testing has been said to be "like a walk in a dark labyrinth without a flashlight."

Because they do not examine the source code, there are situations when a tester writes many test cases to check something that could have been tested by only one test case, or leaves some parts of the program untested.

5.1.4 White Box Testing

White-box testing (also known as clear box testing, glass box testing, and transparent box testing and structural testing) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality. In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit, e.g. in-circuit testing (ICT).

While white-box testing can be applied at the unit, integration and system levels of the software testing process, it is usually done at the unit level. It can test paths within a unit, paths between units during integration, and between subsystems during a system-level test. Though this method of test design can uncover many errors or problems, it might not detect unimplemented parts of the specification or missing requirements. White Box testing is concerned with testing the implementation of the program. The intent of this testing is not to exercise all the different input or output conditions but to exercise the different programming structures and data structures used in the program.

White-box test design techniques include:

- Control flow testing
- Data flow testing
- Branch testing
- Path testing
- Statement coverage
- Decision coverage

5.1.5 Validation Testing

At the culmination of Black Box testing, software is completely assembled as a package, interface errors have been uncovered and corrected and final series of software tests, Validation tests begins. Validation testing can be defined many ways but a simple definition is that validation succeeds when the software functions in a

manner that can be reasonably accepted by the customer. After validation test has been conducted one of the two possible conditions exists.

1. The function or performance characteristics confirm to specification and are accepted.
2. A derivation from specification uncovered and a deficiency list is created.

The screenshot shows the 'Faith' website's 'MEMBER REGISTRATION' form. The form is titled 'MEMBER REGISTRATION' and contains several input fields. The 'Email' field is highlighted with a red border and a red error message: 'Please enter an email address.' Below the error message is a small 'x' icon. The form also includes fields for 'Name', 'Address', 'Date of Birth', 'Gender', 'Phone Number', 'Email', 'Password', 'Confirm Password', and 'Ten digits code'. The 'Email' field contains the text 'ten@'. The 'Date of Birth' field contains '04/01/2001'. The 'Gender' field contains 'Male'. The 'Phone Number' field contains '9876543210'. The 'Email' field contains 'ten@'. The 'Password' field contains '12345678'. The 'Confirm Password' field contains '12345678'. The 'Ten digits code' field contains '1234567890'. The form is set against a light blue background with a white border.

Fig 5.5 Email id validation

MEMBER REGISTRATION

Member Name:

Member / Other Name:

Member Mobile Number:

Email:

Gender: ☒ Male ☐ Female ☐ Other

Ethnicity:

Date of Birth:

Marital Status:

Baptism Name:

Baptism Date:

Baptism Place:

Baptism Church:

Baptism Pastor:

Baptism Parents:

Contact Number:

Please match the requested format: Ten digits code.

Fig 5.6 Ten Digit Code Validation

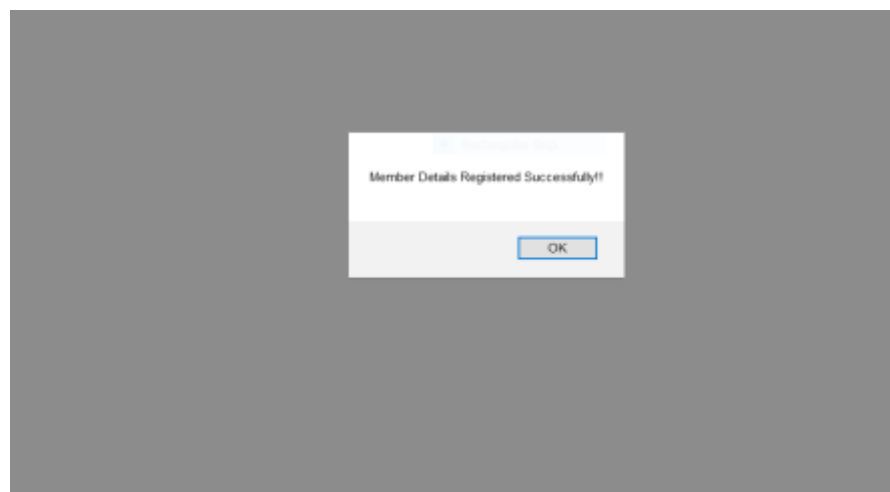


Fig 5.7 Validation Testing Result

We have given various validations in our forms so that there will be a neat format for the data's that are entered on to the website. We have also given an already existing validation so that the data redundancy is reduced; same data is not entered twice.

5.1.6 User Acceptance Testing

Acceptance Testing is a level of the software testing process where a system is tested for acceptability. User Acceptance testing is the software testing process where system tested for acceptability & validates the end to end business flow. Such type of testing executed by client in separate environment & confirms whether system meets the requirements as per requirement specification or not.

UAT is performed after System Testing is done and all or most of the major defects have been fixed. This testing is to be conducted in the final stage of Software Development Life Cycle (SDLC) prior to system being delivered to a live environment. UAT users or end users are concentrating on end to end scenarios & typically involves running a suite of tests on the completed system.

User Acceptance testing also known as Customer Acceptance testing (CAT), if the system is being built or developed by an external supplier. The CAT or UAT are the final confirmation from the client before the system is ready for production. The business customers are the primary owners of these UAT tests. These tests are created by business customers and articulated in business domain languages. So ideally it is collaboration between business customers, business analysts, testers and developers. It consists of test suites which involve multiple test cases & each test case contains input data (if required) as well as the expected output. The result of test case is either a pass or fail.

5.2 TEST CASE DOCUMENTS

A test case is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly. The process of developing test cases can also help find problems in the requirements or design of an application. A sample of test case document format is given below.

CONCLUSION

The project was successfully completed within the time span allotted. All the modules are tested separately and put together to form the main system. Finally, the modules are tested with real data and it worked successfully. Thus the system has fulfilled the entire objective defined.

This project will help people to gasp information that are from various parts under a single roof. Our goal of developing this “CHURCH.IN” has come to a good result without many defects.

6.1. FUTURE ENHANCEMENTS

The system has been designed in such a way that it can be modified with very little effort when such needs arise in the future. New features can be added with slight modifications of software which make it easy to expand the scope of this project. Though the system is working on various assumptions, it can be modified easily to any kind of requirements.

This program would enhance the running of the church information. The existing system will be used alongside the new system to ensure that the church does not loose valuable information when switching to the new system.

The result of this project leads to the conclusion that if this software is introduced and implemented, it would help the church achieve the objectives above and also help eradicate the paper work from the system.

BIBLIOGRAPHY

- [1] K K Aggarwal, Yogesh Singh – Software Engineering Third Edition
- [2] Programming PHP, Ramus Lerdorf and Kevin Tatore
- [3] Beginning PHP, Dave W Mercer, Allan Kent, Steven D Nowicki, David Mercer, Dan Squier, Wankyu Choi, Wiley Publishing, Inc
- [4] Roger S Pressman-Software Engineering: A Practitioner's Approach, Sixth Edition, McGraw- Hill Higher Education
- [5] Pamkaj Jalote-An Integrated approach to Software Engineering, Second Edition, Narosa Publishing Company
- [6] <https://www.visual-paradigm.com>
- [7] https://en.wikipedia.org/wiki/Operating_system
- [8] <http://ecomputernotes.com/software-engineering/feasibilitystudy>
- [9] https://en.wikipedia.org/wiki/Third_normal_form
- [10] <https://searchsoftwarequality.techtarget.com/definition/PERT-chart>
- [11] <https://www.guru99.com/user-acceptance-testing.html>
- [12] https://www.tutorialspoint.com/system_analysis_and_design.html
- [13] <http://softwaretestingfundamentals.com/test-case/>
- [14] <http://softwaretestingfundamentals.com/black-box-testing/>
- [15] <https://www.geeksforgeeks.org/php-coding-standards/>
- [16] <https://www.geeksforgeeks.org/software-engineering-sdlc-v-model/>
- [17] https://www.tutorialspoint.com/sdlc/sdlc_quick_guide.htm
- [18] <http://www.coddletech.com/php>
- [19] <http://www.datatreesystems.in/technologies/mysql/>
- [20] <http://www.wampserver.com/en/>
- [21] https://en.wikipedia.org/wiki/Adobe_Dreamweaver
- [22] https://en.wikipedia.org/wiki/Microsoft_Word
- [23] <https://en.wikipedia.org/wiki/SmartDraw>

APPENDIX

8.1 SCREENSHOTS



Fig 8.1 Guest Page

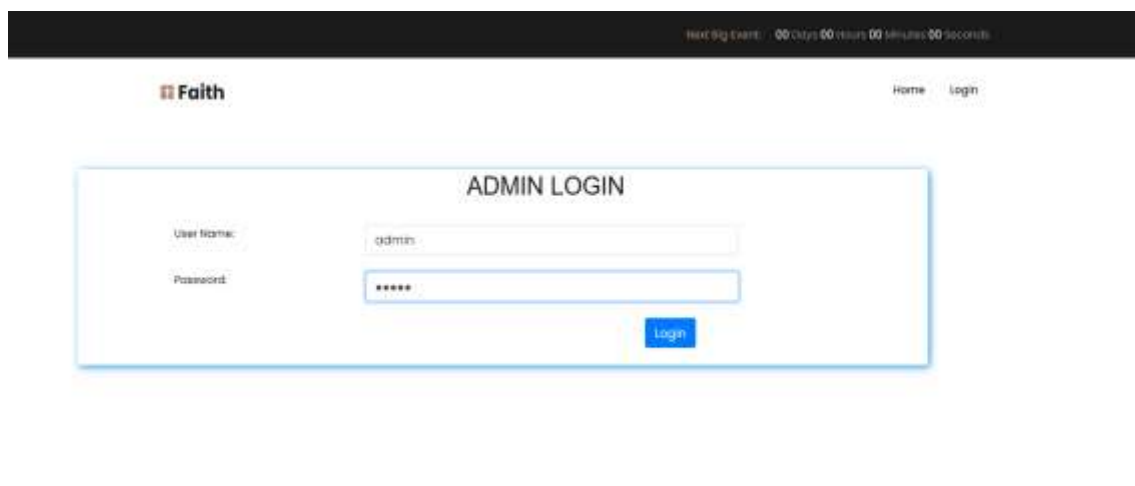


Fig 8.2 Admin Login Page

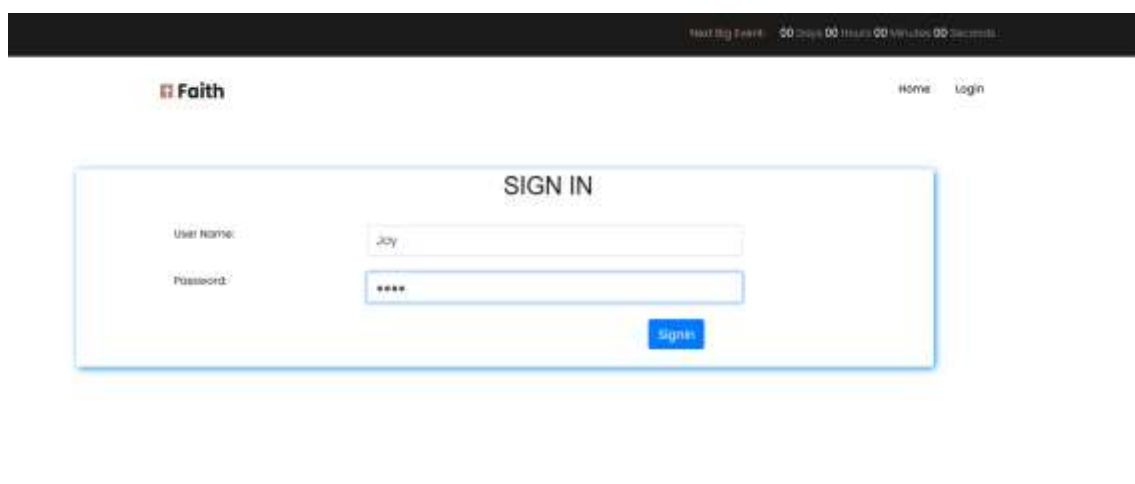


Fig 8.3 Family Login Page



Fig 8.4 Admin Home Page



Fig 8.5 Family Home Page

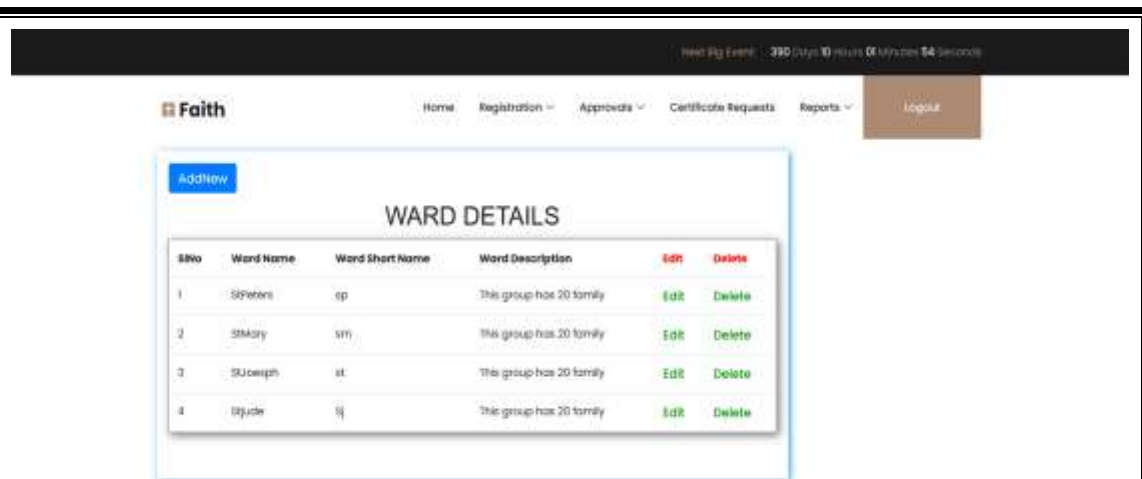


Fig 8.6 Ward Detail

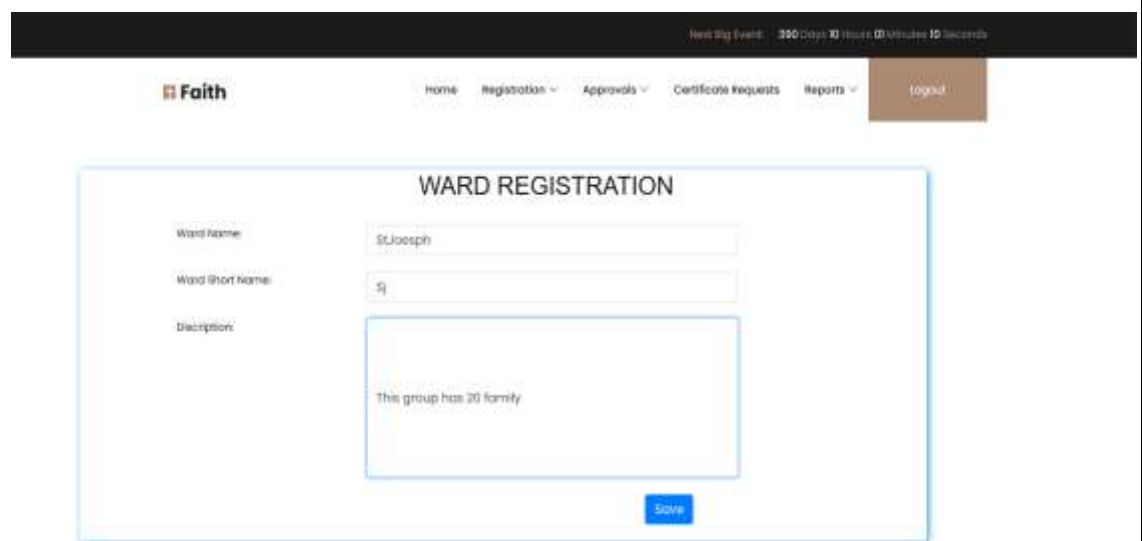


Fig 8.7 New Ward Registration

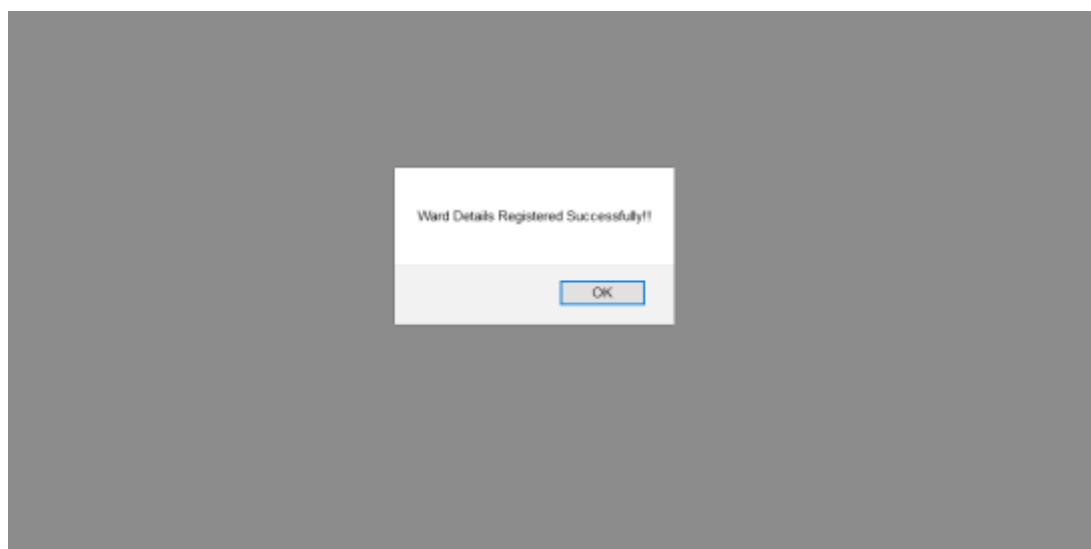


Fig 8.8 Ward Registration Result

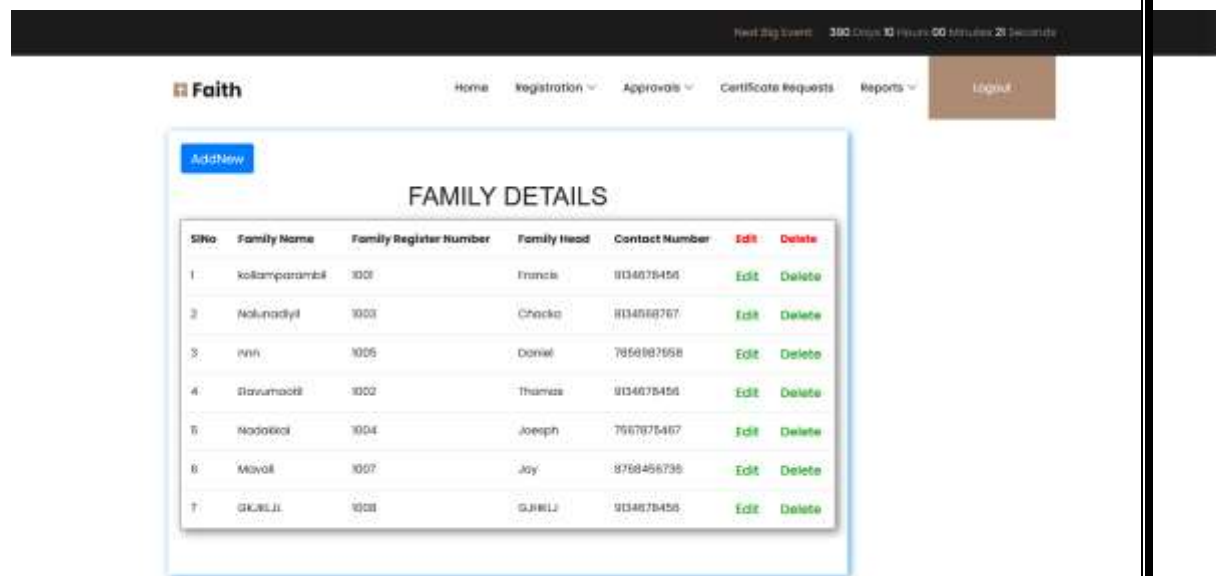


Fig 8.9 Family Details

Family Name: Mavalil

Family Register Number: 1009

Name: St.Joseph

Family Head: Joy

Contact Number: 8876886678

Username: Joy

Password: ****

Save

Fig 8.10 New Family Registration

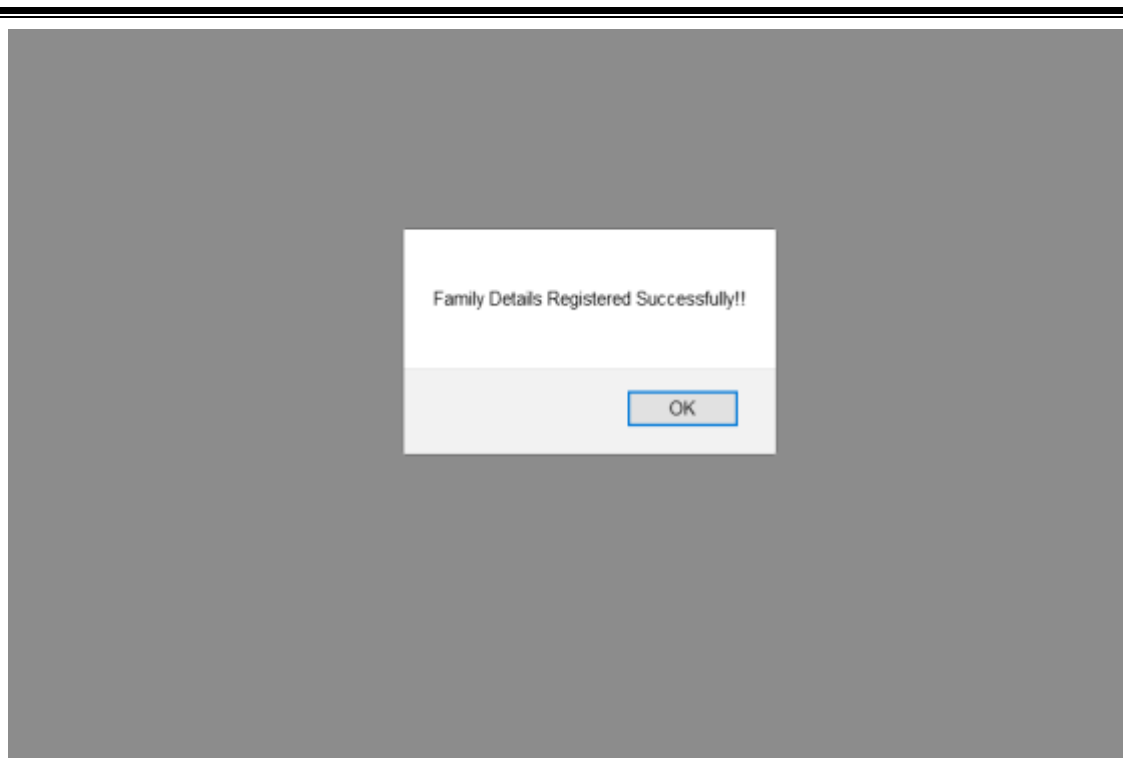


Fig 8.11 Family Registration Result

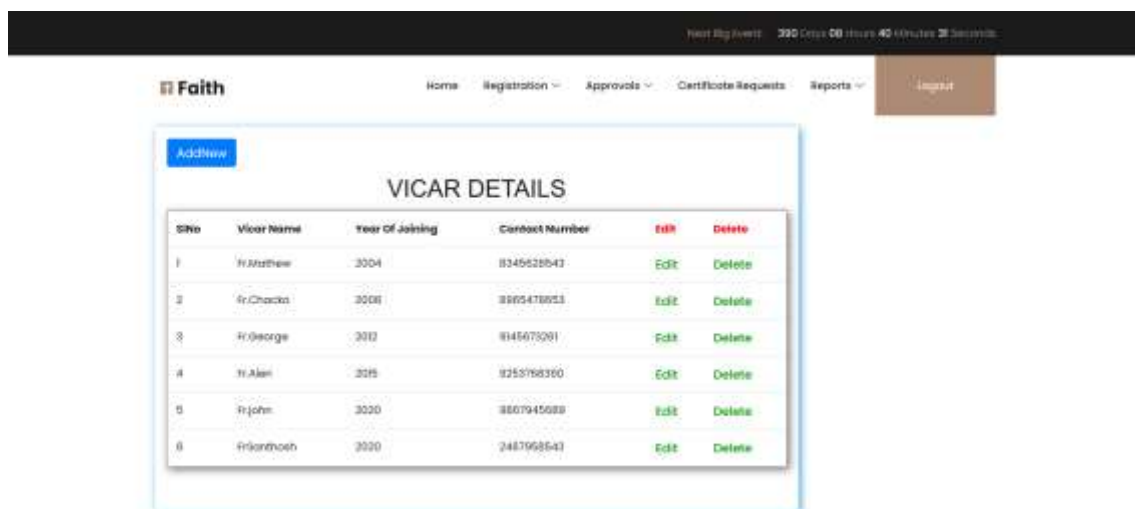
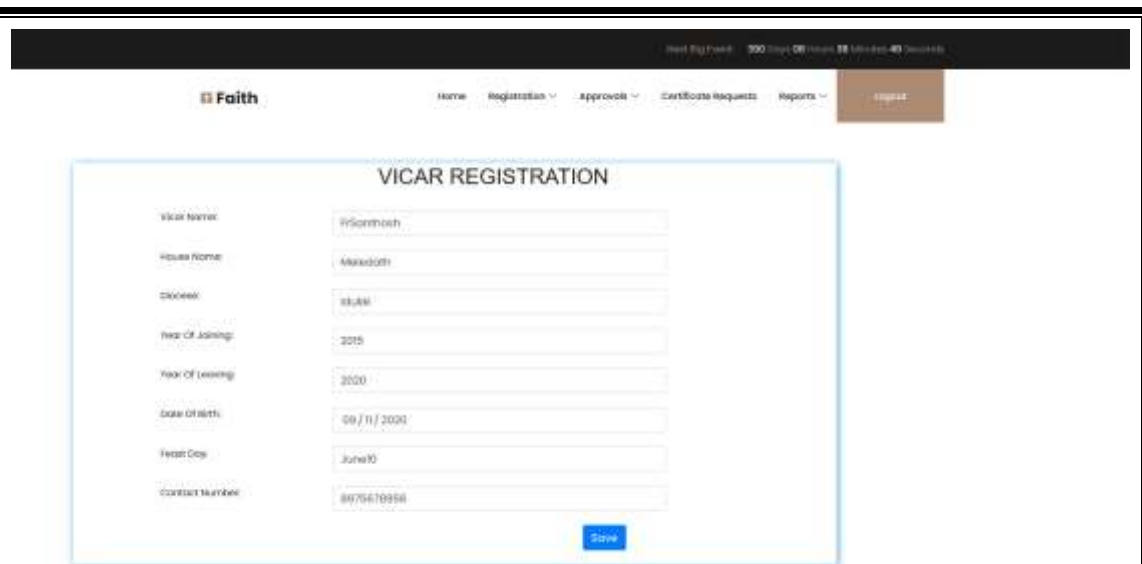


Fig 8.12 Vicar Details



The screenshot shows the 'VICAR REGISTRATION' form within the Church.IN web application. The form is titled 'VICAR REGISTRATION' and contains several input fields for registration details. The fields are as follows:

Field Label	Value
Vicar Name	Prithosh
House Name	Makodoti
Gender	Male
Year Of Joining	2015
Year Of Leaving	2020
Date Of Birth	09 / 11 / 2000
Feast Day	June 10
Contact Number	8075678956

A 'Save' button is located at the bottom right of the form.

Fig 8.13 New Vicar Registration

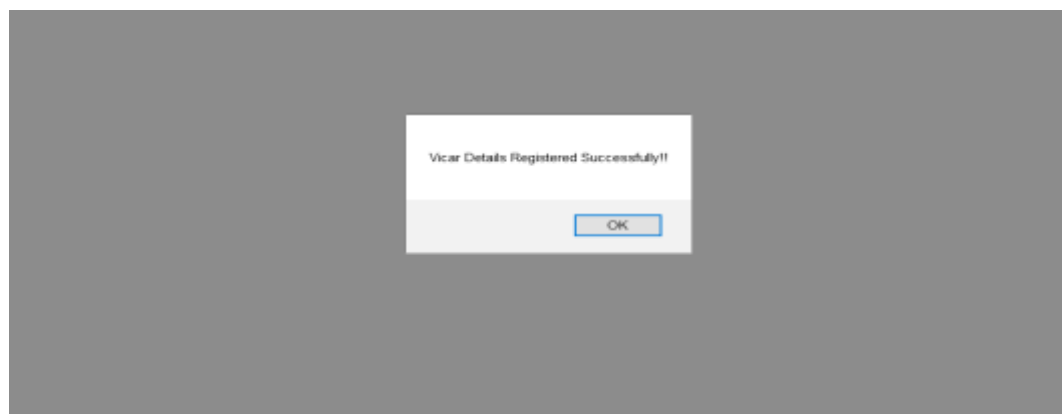
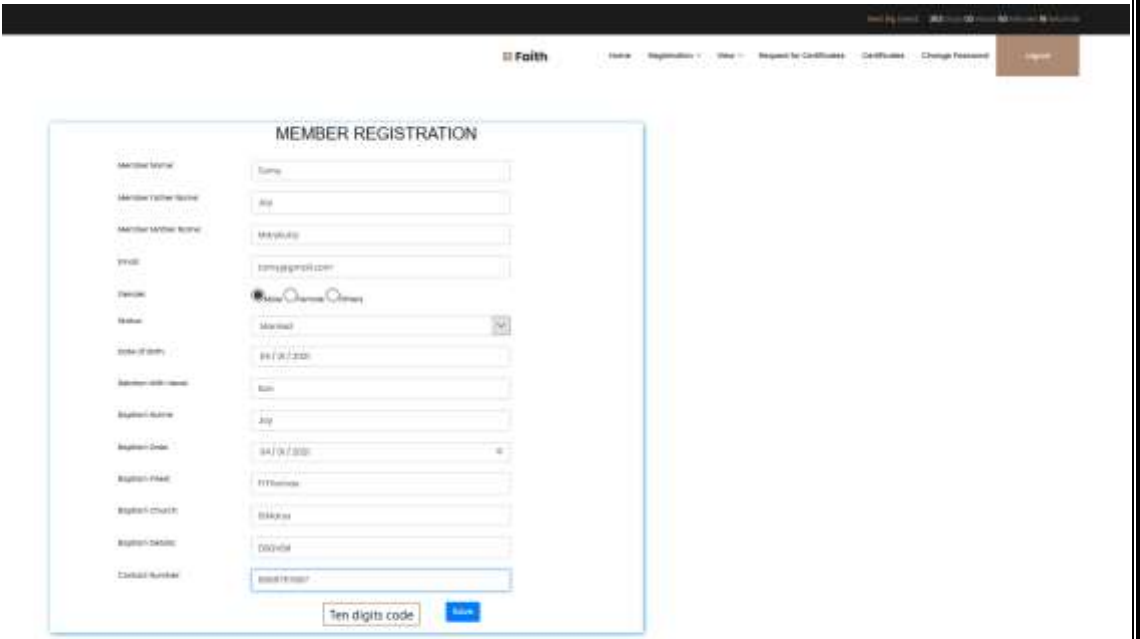


Fig 8.14 Vicar Registration Result



The screenshot shows the 'MEMBER REGISTRATION' form within the Church.IN web application. The form is titled 'MEMBER REGISTRATION' and contains the following fields:

- Member Name: [Text input]
- Member Father Name: [Text input]
- Member Mother Name: [Text input]
- Phone: [Text input]
- Gender: ☒ Male ☐ Female ☐ Other
- Marital: ☐ Married ☒ Single
- Date of Birth: [Date input: DD / MM / YYYY]
- Member Address: [Text input]
- Baptism Name: [Text input]
- Baptism Date: [Date input: DD / MM / YYYY]
- Baptism Place: [Text input]
- Baptism Church: [Text input]
- Baptism Pastor: [Text input]
- Cardinal Number: [Text input]

Below the form, there is a 'Ten digits code' field and a 'Save' button.

Fig 8.15 Member Registration

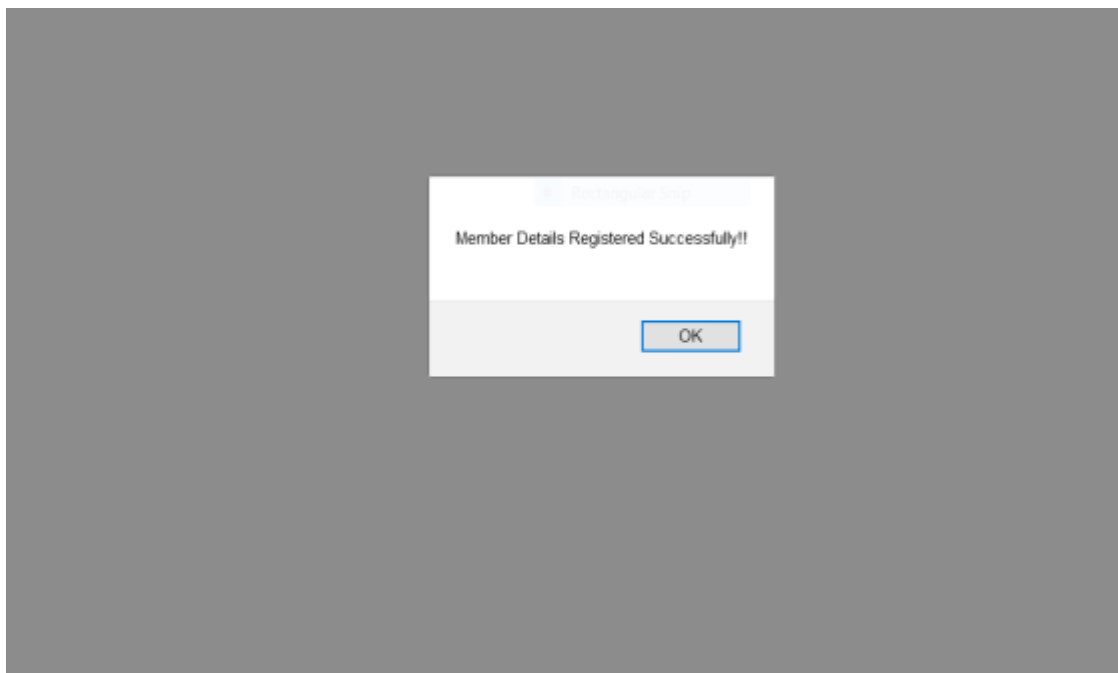
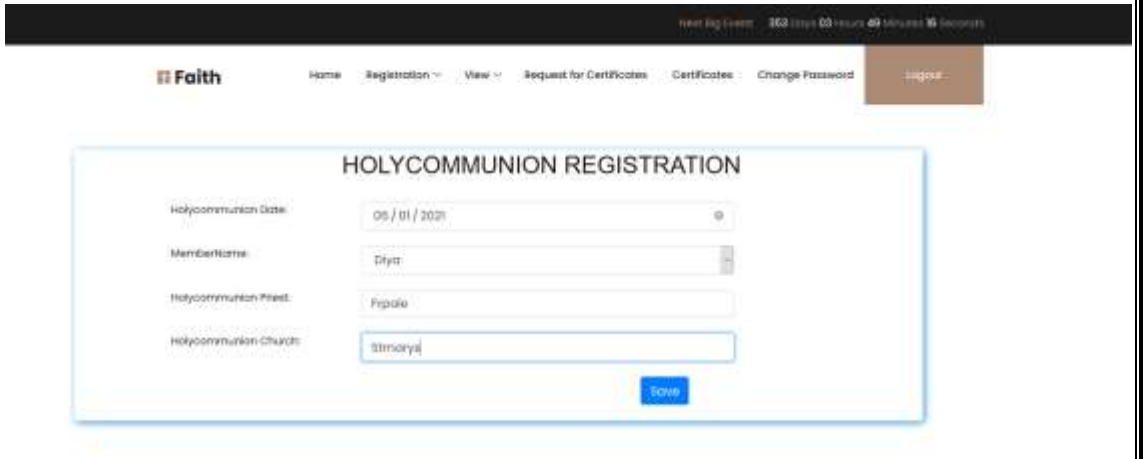


Fig 8.16 Member registration Result



The screenshot shows a web application interface for 'Faith'. At the top, there is a navigation bar with links: Home, Registration, View, Request for Certificates, Certificates, Change Password, and a login button. A timer at the top right indicates 'Next Big Event: 362 days 09 hours 49 minutes 16 seconds'. The main content area features a 'HOLYCOMMUNION REGISTRATION' form with the following fields: 'Holycommunion Date' (05 / 01 / 2021), 'MemberName' (Dipta), 'Holycommunion Priest' (Priya), and 'Holycommunion Church' (Stmarys). A 'Save' button is located at the bottom right of the form.

HOLYCOMMUNION REGISTRATION

Holycommunion Date: 05 / 01 / 2021

MemberName: Dipta

Holycommunion Priest: Priya

Holycommunion Church: Stmarys

Save

Fig 8.17 Holycommunion Registration

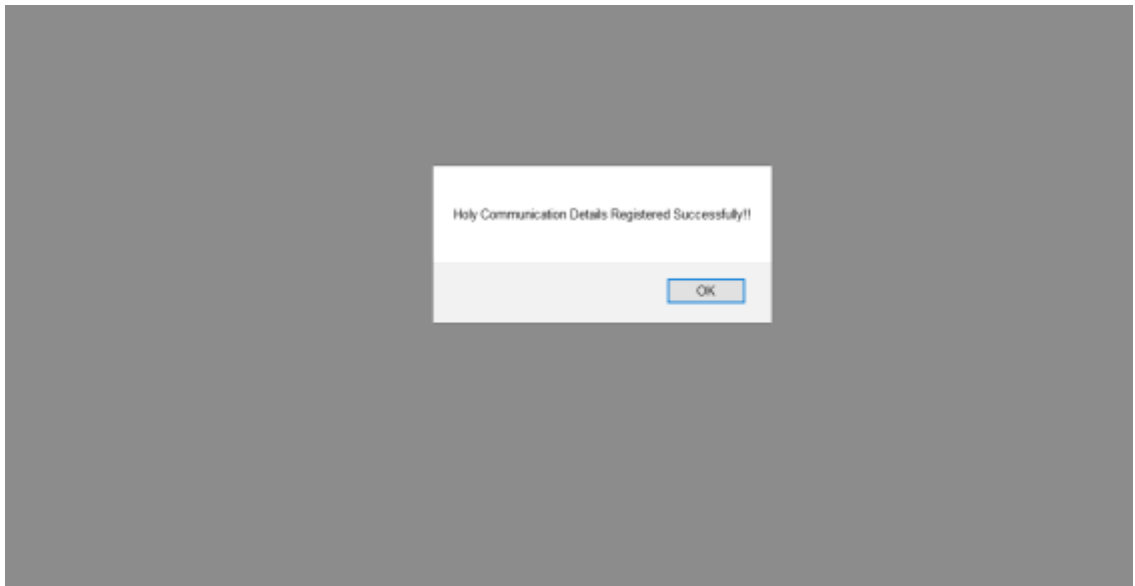
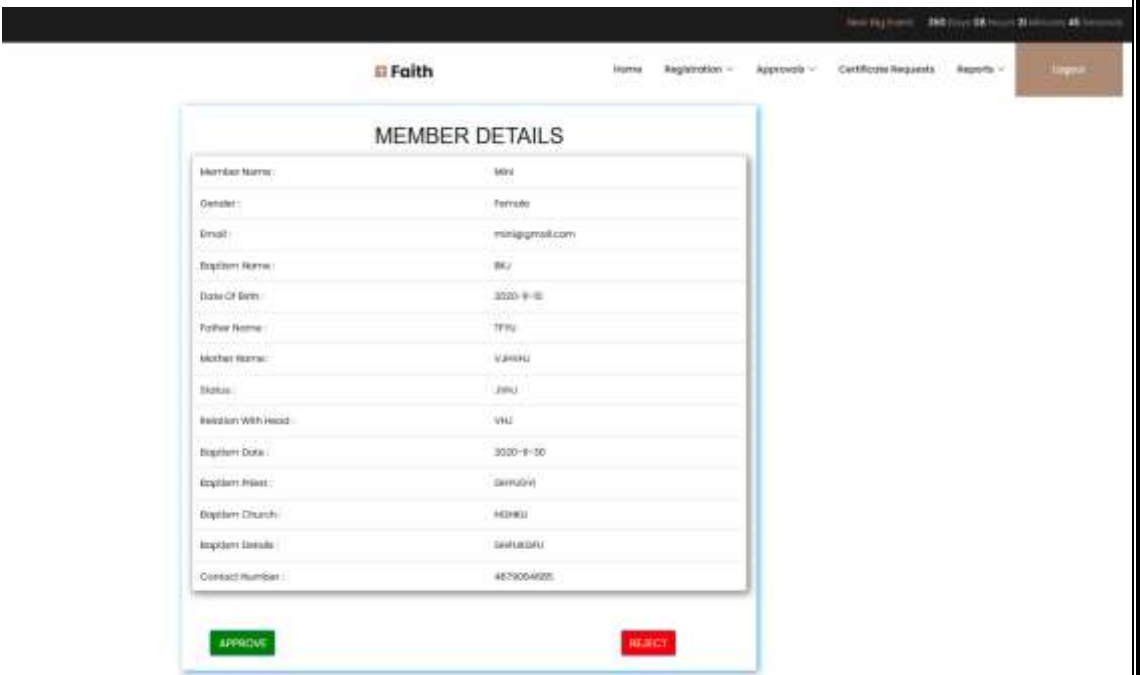


Fig 8.18 Holycommunion Registration Result



The screenshot displays the 'Faith' admin interface. At the top, there is a navigation bar with links: Home, Registration, Approvals, Certificate Requests, Reports, and Logout. The main content area is titled 'MEMBER DETAILS' and contains a form with the following fields:

Field	Value
Member Name	MINI
Gender	Female
Email	mini@gmail.com
Baptism Name	BAZ
Date Of Birth	2020-9-02
Father Name	THIRU
Mother Name	VAISHU
Status	BAU
Relation With Head	WIFE
Baptism Date	2020-9-02
Baptism Place	CHURCH
Baptism Church	CHURCH
Baptism Details	CHURCH
Contact Number	4279004020

At the bottom of the form, there are two buttons: 'APPROVE' (green) and 'REJECT' (red).

Fig 8.19 Admin Approval Page

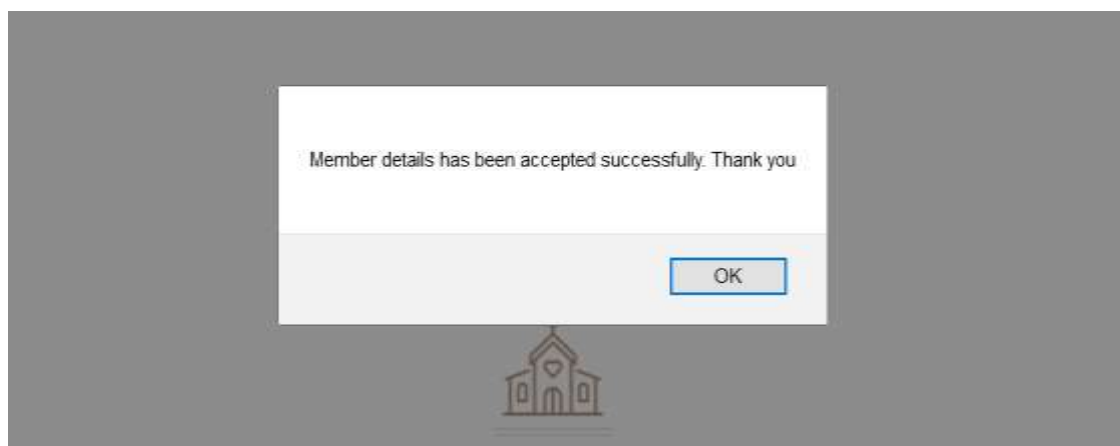


Fig 8.20 Admin Approval Result

8.2. SAMPLE CODE

Fig 4.1 Sample Code

```
<?php
include("header.php");
?>
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Untitled Document</title>
</head>
<body>
<?php
include("config.php");
?>
<form action="familysigninaction.php" method="post" enctype="multipart/form-
data">
<div class="container" style="margin-left: 93px; margin-bottom: 10%; padding-
left:130px; box-shadow: 2px 2px 10px #1b93e1; border-radius: 4px; top: 14px;
margin-top: 3%;">
    <h2 style="text-align: center; margin-top: 6%; font-family: fantasy ;">SIGN
IN</h2> <br>
    <div class="row">
        <div class="col-md-3" style="text-align: left">
```

```

        <label>User Name:</label>
    </div>
    <div class="col-md-6">
        <input      type="text"      class="form-control"      name="txt_uname"
style="width:500px;" placeholder="Username" required>
    </div>
</div>
<br>
<div class="row">
    <div class="col-md-3" style="text-align: left">
        <label>Password:</label>
    </div>
    <div class="col-md-6">
        <input      type="password"      class="form-control"      name="txt_pwd"
style="width:500px;" placeholder="Password" required autofocus>
    </div>
</div>
<br>
<div class="row">
    <input type="submit" name="btnsubmit" value="Signin" class="btn btn-primary"
style="margin-left:63%">
</div>
<br>
</div>
</form>
</body>
</html>
<?php
Include ("footer.php");
?>

```