DIOCESE MANAGEMENT SYSTEM

2016-2018

Submitted in partial fulfillment of the requirements for the award of the degree of Master of Computer Applications of APJ Abdul Kalam Technological University, Thiruvananthapuram.

Submitted By

BISMI MARIA JOSE

Reg No:SJC16MCA-D9

MASTER OF COMPUTER APPLICATIONS

ST.JOSEPHS COLLEGE OF ENGINEERING AND TECHNOLOGY, PALAI

CHOONDACHERRY P.O, KOTTAYAM

KERALA

ST.JOSEPHS COLLEGE OF ENGINEERING AND TECHNOLOGY, PALAI

(An ISO 9001: 2008 Certified College)

CHOONDACHERRY P.O, KOTTAYAM KERALA

CERTIFICATE

This is to certify that the project work entitled **DIOCESE MANAGEMENT SYSTEM** submitted by **BISMI MARIA JOSE** student of **Fourth** semester **MCA 2nd Year Direct** at **ST.JOSEPHS COLLEGE OF ENGINEERING AND TECHNOLOGY, PALAI** in partial fulfillment for the award of Master of Computer Applications .

Mrs. Rinu Rachel Varghese	**Mrs. Rinu Rachel Varghese**	**Dr.Jainendrakumar TD**
(Project Guide)	**(Project Coordinator)**	**(HOD-MCA)**

Submitted for the Viva-Voce Examination held on .....

Examiner 1:

Examiner 2:

ST. JOSEPH'S COLLEGE OF
ENGINEERING & TECHNOLOGY, PALAI
(Approved by AICTE and affiliated to APJ Abdul Kalam Technological University / Mahatma Gandhi University, Kottayam)
An ISO 9001: 2008 certified college

12/04/2018

CERTIFICATE

This is to certify that Ms. BISMI MARIA JOSE, MCA 2nd Year Direct student of St Joseph's College Of Engineering And Technology, Palai has successfully completed the project work entitled "DIOCESE MANAGEMENT SYSTEM" in this college, as part of their curriculum under the guidance of Mr. Rinu Rachel Varughese, Asst. Professor, SJCET during the period from 4th January 2018 to 10th April 2018.We found that she was very sincere and methodical in her approach towards the tasks given.

We wish all the best for her future endeavors.

Dr. Jainendrakumar T.D
Head of the Department,
Dept. of MCA ,
ST Joseph's College of Engg. And Tech., Palai

ST. JOSEPH'S COLLEGE OF ENGINEERING & TECHNOLOGY
CHOONDACHERRY P.O.
KOTTAYAM DIST.

Choondacherry P.O., Kottayam Dist., Kerala - 686 579
Phone : 04822-239301, 239700 Fax: 04822-239307
E-mail : info@sjcetpalai.ac.in
Website : www.sjcetpalai.ac.in

DECLARATION

I, Bismi Maria Jose, hereby declare that the project titled **Diocese Management System** is a record of work carried out under the guidance of **Mrs. Rinu Rachel Varghese, Asst. Professor**, Department of Computer Science and Applications, SJCET, Palai as per the requirement of the curriculum of Master of Computer Applications program of APJ Abdul Kalam Technological University, Thiruvananthapuram. Further, I also declare that this report has not been submitted, full or part thereof, in any University / Institution for the award of any Degree / Diploma.

Place: Choondacherry BISMI MARIA JOSE

Date: Reg No.:- SJC16MCA-D9

ACKNOWLEDGEMENT

The success of any project depends largely on the encouragement and guidelines of many others. I 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.

I also express sincere thanks, from the bottom of our heart, to our parents for their encouragement and support in all our endeavors and especially in this project.

Words are inadequate to express our deep sense of gratitude to **Dr.K.Vijay Kumar, Principal, SJCET, Palai** for allowing us to utilize all the facilities of our college and also for his encouragement.

I extend our sincere gratitude to **Dr. Jainendra Kumar TD, Head of the Department of Computer Science and Applications, SJCET, Palai** who has been a constant source of inspiration and without his tremendous help and support this project would not have been materialized.

I owe a particular debt of gratitude to our internal project guide **Mrs.Rinu Rachel Varghese ,Asst. Professor, Department of Computer Science and Applications, SJCET, Palai** 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 this project.

I extend my sincere thanks to all of our teachers and non-teaching staff of SJCET, Palai for the knowledge they have imparted to us over the last two years. I would also like to express my appreciation to all our friends for their comments, help and support.

SYNOPSIS

DIOCESE MANAGEMENT SYSTEM

Diocese Management System is a web application used to share the data between members of a Diocese and aims to provide web services with better visibility and efficiency. It is mainly used to automate the manual task of members of Diocese (Bishop, Vicar, Asst. Vicar and Parish Members).This system consists of the Super admin, Sub admin and the Parish members. It helps Super admin to manage their Subadmin and Parish members in a single click.

The Subadmin is usually the Bishop who is the Head of the Diocese. He can add parishes, priests and transfer priest that comes under his control. The Subadmin is the Vicar or Assistant Vicar that belongs to individual parishes in that Diocese. Subadmin can add family, generate certificates requested by parish members and non-parish members, approve and reject the requests of parish members.

Parish members can request the Subadmin for carrying out the Parish Ceremonies like Baptism, Confirmation, Marriage, Death, and Community Certificate. They can offer donations in the event of Parish ceremonies via online. The Subadmin and analyze records and generate reports that is to be presented to Superadmin.

Contents

List of Tables

List of Figures

1 INTRODUCTION

1.1 Problem Definition

The project Diocese Management System is concerned with the development of a web application that helps the Arch Diocese of Thalasseri to automate the year old manual tasks of the Diocese. It is aimed to digitize the paper works that have been recorded in the churches under the Diocese.

The project enables to add parishes, assign priests to each parish, and transfer priests from one parish to another. The higher authority of the diocese, usually the Bishop is given the privilege to handle things relating the parish and priests.

The vicar or assistant vicar handles things regarding the parish and its members like user login registration, create events, donations, managing events like approval or rejecting marriage, baptism, communion, death record.

One of the main problems with the current system is that there is no platform for getting certificates online. The requester has to contact with the Parish priest to acquire certificates due to security reasons. The existing system is a manual system and it needs a number of staffs and a lot of time consumption for the process.

As technology is advancing, each and every works are computerizing. Everyone prefers online management which is faster and time conserving. So in this context prefer to modify the current system.

1.2 Objective of the project

The main objective of the project is to create a website that enables the members of Arch Diocese like Bishop, Priests, and Members of Parish to automatically carry out manual tasks and to reduce the burden of waiting long to request and respond to the activities.

2 LITERATURE SURVEY

2.1 Initial Investigation

The purpose of this document is to give a clear picture of the module designs of the project Diocese Management System. It is a system that automates the manual activities of the Diocese, Parishes, and Parish members. In the manual system, records were not digitized. Thus the storage, retrieval and processing of records were too hectic and overloaded. The need to smoothly carryout the ongoing tasks of Church has led to the automation of existing system.

2.2 Existing System

The study of the existing system is a pre-requisite for developing any software sys-tem. The study of the system reveals many features of the existing system. This gives analyst an insight into the working of the system and helps the developer to design an appropriate system, which will eliminate the many limitations present in the existing system.

Limitations of Existing System are:

- It is a manual system
- Process is by means of manual search
- Difficult to keep all the paper records
- A lot of time is consumed in searching records
- Chances of loss in document containing important details
- Time consuming and miss handling of reports

2.3 Proposed System

The proposed system computerization is developed using SQL server as back-end and ASP.net as front-end. The ASP.net framework is managed, type safe environment for application, development and execution. The software is developed as a simulated system and the complex procedures are avoided to make the system easy to use. The proposed system is user friendly and has simplicity and security. In the proposed system the data redundancy can be avoided to certain extend and the data consistency can be maintained.

2.3.1 Advantages of proposed system

- Give solution to the current system problems
- Less time consuming and more efficient
- Result will be very precise and accurate
- Searching is easy and accurate
- Efficient and effective storage and retrieval.
- Easy searching and storing documents
- Eliminate chances for errors and reduce effort

2.3.2 Features of the Proposed System

- Access to the system and database as per user identification
- The maximum security ensured
- User-friendly and flexible in all aspects
- Data entry updates is quite easy
- Effective table manipulation as facilitated by the rich SQL Good validation checking
- Good validation checking
- Easy maintenance
- Removes chances of leakage of information
- Provides a better record keeping system

2.4 Feasibility Study

An important outcome of the preliminary investigation is the determination that the system requested is feasible. Feasibility study is carried out to select the best system that meets the performance requirements.

Feasibility study is both necessary and prudent to evaluate the feasibility of the project at the earliest possible time. It involves preliminary investigation of the project and examines whether the designed system will be useful to the organization. Months or years of effort, thousand for millions of money and untold professional embarrassment can be averted if an in-conceived system is recognized early in the definition phase.

Feasibility study is a procedure that identifies, describes and evaluates candidate system and selects the best system for the job. An estimate is made of whether the identified user needs may be satisfied using current software and hardware technologies. The study will decide if the proposed system will be cost effective from a business point of view and if it can be developed given existing budgetary constraints. The key considerations involved in the feasibility analysis are economic, technical, behavioral and operational. During feasibility analysis for this project, following primary areas of interest are to be considered. Investigation and generating ideas about a new system does this.
Eight steps involved in the feasibility analysis are:

- Form a project team and appoint a project leader.
- Prepare system flowcharts.
- Enumerate potential proposed system.
- Define and identify characteristics of proposed system.
- Determine and evaluate performance and cost effective of each proposed system.
- Weight system performance and cost data.
- Select the best-proposed system.

2.4.1 Technical Feasibility

According to feasibility analysis procedure the technical feasibility of the system is analyzed and the technical requirements such as software facilities,procedure,inputs,are identified. While considering the problems of existing system,it is sufficient to implement the new system.The proposed system can be implemented to solve issues in the existing system.It includes the evaluation of and how it meets the proposed system. This system use PHP as front end technology and MYSQL Server as backend tech- nology.

2.4.2 Economic Feasibility

Economic feasibility deals about the economical impact faced by the organization to implement a new system. Financial benefits must equal or exceed the costs. The cost of

conducting a full system, including software and hardware cost for the class of application being considered should be evaluated. Economic Feasibility in this project:

- The cost to conduct a full system investigation is possible.
- There is no additional manpower requirement.
- There is no additional cost involved in maintaining the proposed system.

Economic justification is generally the Bottom Line consideration for most systems. Economic justification includes a broad range of concerns that includes cost benefit analysis. In this we weight the cost and the benefits associated with the candidate system and if it suits the basic purpose of the organization i.e. profit making, the project is making to the analysis and design phase. The financial and the economic questions during the preliminary investigation are verified to estimate the followings:

- The cost to conduct a full system investigation.
- The cost of hardware and software for the class of application being considered.
- The benefits in the form of reduced cost.

The proposed system will give the minute information; as a result the performance is improved.This feasibility checks whether the system can be developed with the available funds. The Diocese management system does not require enormous amount of money to be developed. This can be done economically if planned judicially, so it is economically feasible.

2.4.3 Operational Feasibility

This feasibility test asks if the system will work when it is developed and installed.The proposed system of Diocese management system offers greater level of user-friendliness.The system operation is one of the important phase in the development life cycle of a system. So, operational feasibility should be given much importance. The users of the system dont need thorough training on the system. All they are expected to know to operate the system is the basic net surfing knowledge. It has user-friendly interfaces.

All the behavioral aspects are considered carefully and have found that the project is behaviorally feasible. Thus this project passes these entries, tests for feasibility and thus found feasible. The proposed system produces best results and gives high performance. It can be implemented easily .So this project is operationally feasible.

2.4.4 Behavioral Feasibility

In todays world,computer is an inevitable entity.As per the definition of behavior design, many valid points are recognized in this study.This system behavior changes according to different environment.In order to ensure proper authentication and authorization and security of sensitive data of the admin or farmers,log-in facilities are provided.These are the main feasibility studies tested in this application.

3 SYSTEM ANALYSIS AND DESIGN

System analysis is a management technique which helps us in designing a new system or improving an existing system . System analysis is a process of gathering and interpreting facts, diagnosing problems and using the information to recommend improvement of the system.During these places, the analyst and the user come to a detailed agreement on whet function the proposed system has to perform. This one contains:

- Inputs to be supplied.
- Output to be produced.
- Procedures to get the outputs from the given inputs.
- Data to be retained.

3.1 Requirement Analysis

Requirements analysis encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product or project, taking account of the possibly conflicting requirements of the various stakeholders, analyzing, documenting, validating and managing software or system requirements.

Requirements analysis is critical to the success or failure of a systems or software project.The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design.

- **Eliciting requirements:** business process documentation, and stakeholder interviews. This is sometimes also called requirements gathering or requirements discovery.
- **Analyzing requirements:** determining whether the stated requirements are clear, complete, consistent and unambiguous, and resolving any apparent conflicts.
- **Recording requirements:** Requirements may be documented in various forms, usually including a summary list and may include natural-language documents, use cases, user stories, process specifications and a variety of models including data models.

Requirements analysis can be a long and tiring process during which many delicate psychological skills are involved. Large systems may confront analysts with hundreds or thousands of system requirements. New systems change the environment and relationships between people, so it is important to identify all the stakeholders, take into account all their needs and ensure they understand the implications of the new systems. Analysts can employ several techniques to elicit the requirements from the customer. These may include the development of scenarios (represented as user stories in agile methods), the identification of use cases,

the use of workplace observation or ethnography, holding interviews, or focus groups (more aptly named in this context as requirements workshops, or requirements review sessions) and creating requirements lists. Prototyping may be used to develop an example system that can be demonstrated to stakeholders.

User requirements gathering and analysis. The aim of our user-focused tools and methods is to provide a clear understanding of requirements as an early input to development projects. We help focus design work on those issues that are central to the success of a product or system in the eyes of the end-users.

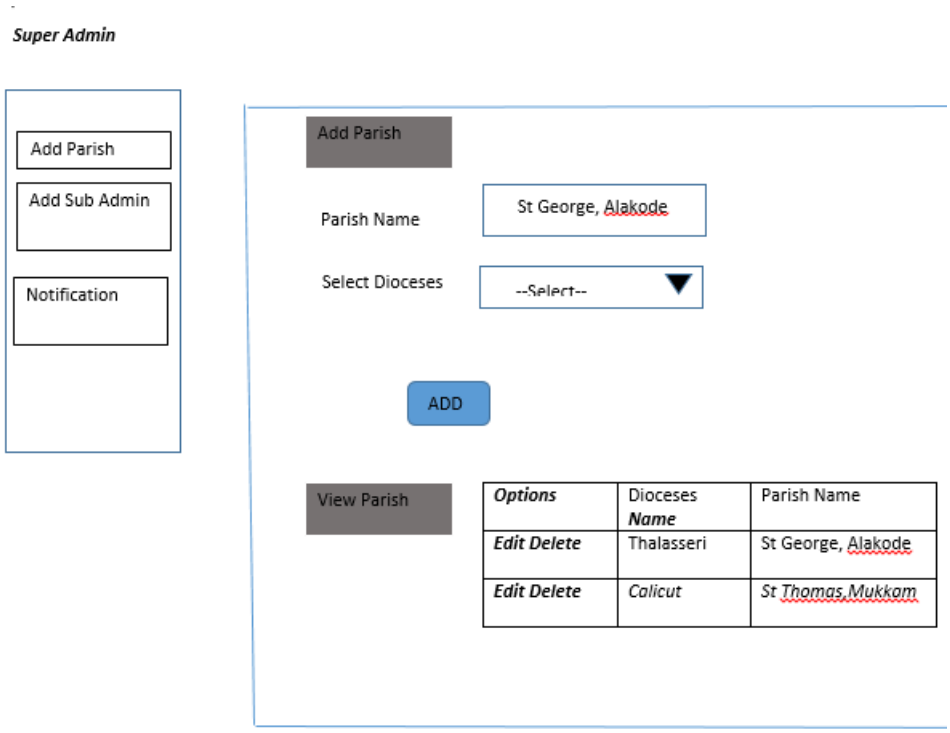
User interface design

User interface design (UI) or user interface engineering is the design of user interfaces for machines and software, such as computers, home appliances, mobile devices, and other electronic devices, with the focus on maximizing usability and the user experience. The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals (user-centered design).

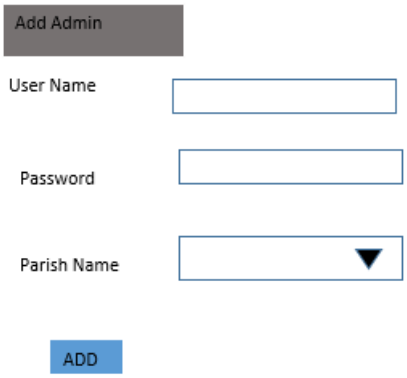
Good user interface design facilitates finishing the task at hand without drawing unnecessary attention to itself. Graphic design and typography are utilized to support its usability, influencing how the user performs certain interactions and improving the aesthetic appeal of the design; design aesthetics may enhance or detract from the ability of users to use the functions of the interface.The design process must balance technical functionality and visual elements (e.g., mental model) to create a system that is not only operational but also usable and adaptable to changing user needs.

3.1.1 Sample Screen

super admin Login



add admin

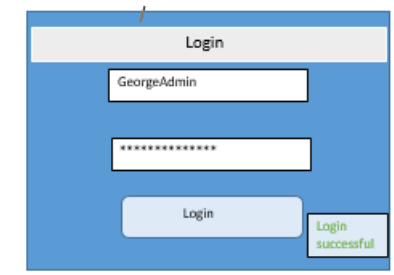


super admin view

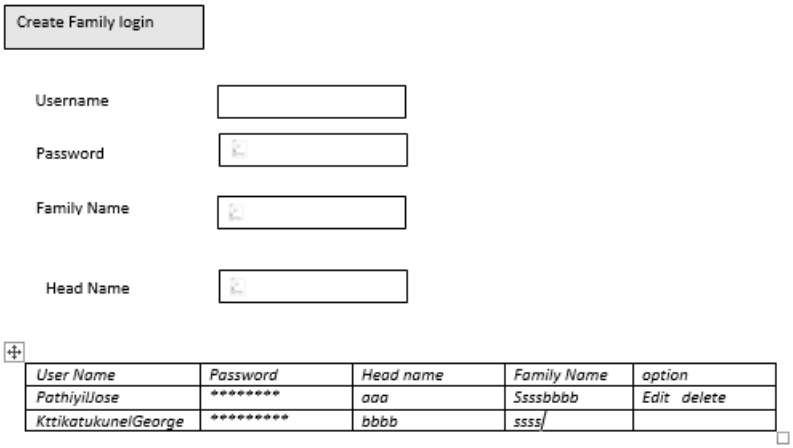
View admin/

Options	Username	Password	Parish Name
Edit Delete	GeorgeAdmin	********	St George, Alakode
Edit Delete	ThomasAdmin	********	*St Thomas,Mukkam*

Sub admin Homepage



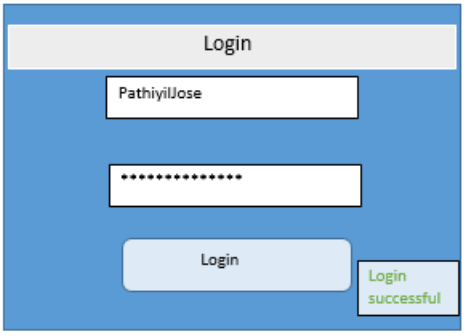
Sub admin create family



Member Homepage

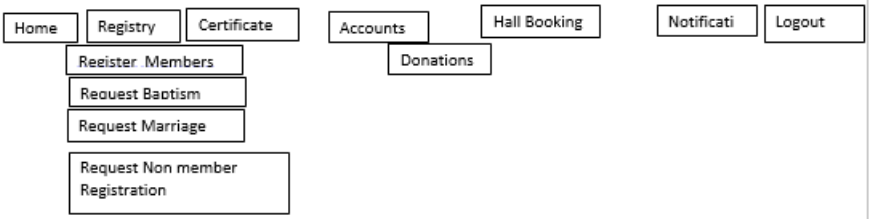
Member login page

Main login page



Home Page

Welcome PathiyilJose(Member)



3.2 Software And Hardware Requirement Specification

3.2.1 Hardware Requirements

- Processor : Intel Pentium IV 2.4 GHZ or above
- Clock speed : 2.33 MHZ
- System bus : 64 bits
- RAM : 4.00 GB of RAM
- HDD : 500 GB or higher
- Monitor : SVGA COLOR
- Keyboard : 108 keys
- Mouse : 2 button mouse

3.2.2 Software Requirements

- Operating System - Windows 10
- Web Browser - Google Chrome
- Front End - ASP.NET
- Back End - MS SQL SERVER 2015

3.3 System Design

Designing the system in an effective way leads to the smooth working of any softwares. System design is the process of developing specification for a candidate system that meet the criteria established in the system analysis. Major step in the system design is the preparation of the input forms and output reports in a form applicable to the user. The main objective of the system design is to use the package easily by any computer operator. System design is the creative act of invention, developing new inputs, and database, offline files, method, procedure and output for processing business to meet an organization objective. System design builds information gathered during the system analysis. This system is designed neatly so that user will never get ambiguity while using the system.

3.3.1 Modular Design

Mainly this project consists of 3 Modules:

- Admin module
- User Module
- Users can log-in to the system using his User-name, Password,Parish Name and User Type.
- Super admin: - can add parishes,priests,transfer priest from one parish to another.
- Sub admin: - can create log-in for each family.Create events,add ward details,issue certificates when demanded by members.
- Members: - can fill details, add family members,request for ceremony like baptism,marriage,betrothal,de and also request for issuing certificates of the ceremony.

3.3.2 Input Design

Input Design deals with what data should be given as input, how the data should be arranged or code, the dialog to guide the operating personnel in providing input, methods for preparing input validations and steps to follow when error occur. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow.

3.3.3 Output Design

A quality output is one, which meets the requirements of the end user and presents the information clearly. The objective of output design is to convey information about past activities, current status or projections of the future, signal important events, opportunities, problems, or warnings, trigger an action, confirm an action etc. Efficient, intelligible output design should improve the systems relationship with the user and helps in decisions making. In output design the emphasis is on displaying the output on a CRT screen in a predefined format. The primary consideration in design of output is the information requirement and objectives of the end users. The major formation of the output is to convey the information and so its layout and design need a careful consideration.
Two phases of the output design are:
1. Output definition.
2. Output specification.

3.3.4 Database Design

A database is an organized mechanism that has the capability of storing Information through which a user can retrieve stored information in an effective and efficient Manner. The data is the purpose of any database and must be protected.The database design is a two level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual Database Management System (DBMS).

In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design.

Database Design

Table 1 **PositionTable**
Stores Family Tree.

Table 1: Position

Column	**Type**	**Attributes**	**Descriptions**
PositionId	int	Primary Key	It is used to store the unique id
PositionValue	varchar(50)	Not Null	Should not be null

Table 2 **BloodListTable**
Stores Blood Group.

Table 2: Blood Group

Column	**Type**	**Attributes**	**Descriptions**
BloodgroupId	int	Primary Key	It is used to store the unique id
BloodGroupValue	varchar(50)	Not Null	Should not be null

Table 3 **Sup_Parish_Table**
Super Admin adds new Username and Password for Parishes.

Table 3: Parish Name

Column	**Type**	**Attributes**	**Descriptions**
FIELDS	DATA TYPE	CONSTRAINT	DESCRIPTION
ParishId	int	Primary Key	It is used to store the unique id
ParishName	varchar(50)	Not Null	Should not be null
Place	varchar(50)	Not Null	Should not be null
UserName	varchar(50)	Not Null	Should not be null
Password	varchar(50)	Not Null	Should not be null

Table 4 **Sup_Priest_Table**
Super Admin adds Parish Priests.

Table 4: Parish Priests

Column	**Type**	**Attributes**	**Descriptions**
Parish_Priest_Id	int	Primary Key	It is used to store the unique id
Parish_Priest_Name	varchar(50)	Not Null	Should not be null
Parish_Priest_Image	varchar(50)	Null	Can be null
Contact_Number	varchar(50)	Null	Can be null
OrdinationDate	varchar(50)	Null	Can be null
Designation	varchar(50)	Null	Can be null
CurrentParish_id	int	Not Null	Should not be null

Table 5 **Sup_Priest_TransferTable**
Super Admin Transfers Parish Priests .

Table 5: Transfer Parish Priests

Column	**Type**	**Attributes**	**Descriptions**
Transfer_Id	int	Primary Key	It is used to store the unique id
Priest_id	int	Foreign Key	Should not be null
Parish_id	int	Foreign Key	Should not be null

Table 6 **Sub_WardTable**
Sub Admin adds new Ward.

Table 6: Ward Details

Column	**Type**	**Attributes**	**Descriptions**
WardID	int	Primary Key	It is used to store the unique id
WardValue	varchar(50)	Not Null	Should not be null
Parishid	int	Foreign Key	Should not be null

Table 7 **Sub_CreateFamily_Table**
Sub Admin Creates Family Login.

Table 7: Family Login

Column	**Type**	**Attributes**	**Descriptions**
FamilyId	int	Primary Key	It is used to store the unique
FamilyName	varchar(50)	Not Null	Should not be null
Wardid	int	Foreign Key	Should not be null
Parish$_id$	int	Foreign Key	Should not be null
UserName	varchar(50)	Not Null	Should not be null
Password	int(11)	Not Null	Should not be null
FamilyNo	int	Not Null	Should not be null
ContactNo	varchar(50)	Not Null	Should not be null
HeadName	varchar(50)	Not Null	Should not be null

Table 8 **Sub_ParishHallTable**
Sub Admin adds new Parish Hall

Table 8: Parish Hall

Column	**Type**	**Attributes**	**Descriptions**
Hallid	int	Primary Key	It is used to store the unique id
HallName	varchar(50)	Not Null	Should not be null
Parishid	int	Foreign Key	Should not be null

Table 9 **: Sub_ RequestTable**
Stores Requests of Users.

Table 9: Request

Column	**Type**	**Attributes**	**Descriptions**
RequestId	int	Primary Key	It is used to store the unique id
EventName	varchar(50)	Not Null	Should not be null
Usertype	int	Not Null	Should not be null
ProposedTime	varchar(50)	Not Null	Should not be null
ProposedDate	varchar(50)	Not Null	Should not be null
Status	varchar(50)	Not Null	Should not be null
RequestTime	varchar(50)	Not Null	Should not be null
Parishid	int	Foreign Key	Should not be null
Memberid	int	Foreign Key	Should not be null

Table 10 **MemberDetailTable**
Stores details of Members in the Parish.

Table 10: MemberDetails

Column	**Type**	**Attributes**	**Descriptions**
MemberId	int	Primary Key	It is used to store the unique id
OfficialName	varchar(50)	Not Null	Should not be null
Address	varchar(50)	Not Null	Should not be null
PhoneNo	varchar(50)	Not Null	Should not be null
Dob	varchar(50)	Not Null	Should not be null
Email	varchar(50)	Null	Can be null
GenderValue	int	Not Null	Should not be null
Father_sName	varchar(50)	Not Null	Should not be null
Mother_sName	varchar(50)	Not Null	Should not be null
Occupation	varchar(50)	Not Null	Should not be null
Parish	varchar(50)	Not Null	Should not be null
PreviousParish	varchar(50)	Not Null	Should not be null
BaptismName	varchar(50)	Not Null	Should not be null
HouseNob	varchar(50)	Not Null	Should not be null
WardID	int	Foreign Key	Should not be null
BloodgroupId	int	Foreign K	Should not be null
PatronSaint	varchar(50)	Not Null	Should not be null
Image	varchar(50)	Not Null	Should not be null
WifesBaptismName	varchar(50)	Not Null	Should not be null
WifesOfficialName	varchar(50)	Not Null	Should not be null
BaptismStatus	int	Not Null	Should not be null
Registered_Status	int	Not Null	Should not be null

Table 11 **DeathTable**
Stores details of Dead Members in the Parish.

Table 11: Death Details

Column	**Type**	**Attributes**	**Descriptions**
DeathId	int	Primary Key	It is used to store the unique id
Memberid	int	Foreign Key	Should not be null
Death_Date	varchar(50)	Not Null	Should not be null
Buried_Date	varchar(50)	Not Null	Should not be null
ParishID	int	Foreign Key	Should not be null

Table 12 **BaptismTable**
Stores Baptism Details of Members.

Table 12: Baptism Details

Column	**Type**	**Attributes**	**Descriptions**
BaptismId	int	Primary Key	It is used to store the unique id
BaptismName	varchar(50)	Not Null	Should not be null
Dobaptism	varchar(50)	Not Null	Should not be null
NativeParish	varchar(50)	Not Null	Should not be null
Father_sOName	varchar(50)	Not Null	Should not be null
Mother_OName	varchar(50)	Not Null	Should not be null
ChiefPriest	varchar(50)	Not Null	Should not be null
GodFather	varchar(50)	Not Null	Should not be null
GodMother	varchar(50)	Not Null	Should not be null
PlaceOBirth	varchar(50)	Not Null	Should not be null
OfficialName	varchar(50)	Not Null	Should not be null
Memberid	int	Foreign Key	Should not be null
Usertype	int	Not Null	Should not be null
GFProof	varchar(50)	Not Null	Should not be null
GMProof	varchar(50)	Not Null	Should not be null
FProof	varchar(50)	Not Null	Should not be null
MProof	varchar(50)	Not Null	Should not be null
UrProof	varchar(50)	Not Null	Should not be null

Table 13 **Marriage Table**
Stores Marriage Details of Members.

Table 13: Marriage Details

Column	**Type**	**Attributes**	**Descriptions**
MarriageId	int	Primary Key	It is used to store the unique id
GFamilyName	varchar(50)	Not Null	Should not be null
GParishName	varchar(50)	Not Null	Should not be null
GDiocese	varchar(50)	Not Null	Should not be null
GOfficialName	varchar(50)	Not Null	Should not be null
GBaptismName	varchar(50)	Not Null	Should not be null
GFName	varchar(50)	Not Null	Should not be null
GMName	varchar(50)	Not Null	Should not be null
BFamilyName	varchar(50)	Not Null	Should not be null
BParishName	varchar(50)	Not Null	Should not be null
BDiocese	varchar(50)	Not Null	Should not be null
BOfficialName	varchar(50)	Not Null	Should not be null
BBaptismName	varchar(50)	Not Null	Should not be null
BFName	varchar(50)	Not Null	Should not be null
BMName	varchar(50)	Not Null	Should not be null
PremarriageCerti	varchar(50)	Not Null	Should not be null
BetrothalParishName	varchar(50)	Not Null	Should not be null
BetrothalDate	varchar(50)	Not Null	Should not be null
Bann1	varchar(50)	Not Null	Should not be null
Bann2	varchar(50)	Not Null	Should not be null
Bann3	varchar(50)	Not Null	Should not be null
Betrothal_BlessedBy	varchar(50)	Not Null	Should not be null
Betrothal_witness1	varchar(50)	Not Null	Should not be null
Betrothal_witness2	varchar(50)	Not Null	Should not be null
MarriageParish	int	Not Null	Should not be null
Marriage_BlessedBy	varchar(50)	Not Null	Should not be null
Usertype	int	Not Null	Should not be null
Memberid	int	Not Null	Should not be null

3.3.5 Data Flow Diagram

Data flow diagram is the graphical representation of the system. It is a network that uses special symbols to describe the flow of data and process that transforms data throughout the system. Data flow diagram is a way of representing system requirements in a graphic form. A DFD also known as Bubble Chart has the purpose of clarifying system requirements and identifies major transformations that will become program in system design. So it is the starting point of design phase that functionally decomposes the requirements specifications down to the lowest level of details. A DFD consist of series of bubbles joined by lines. The bubbles represent data transformation and the lines represent data flow in the system.

Level 0 DFD

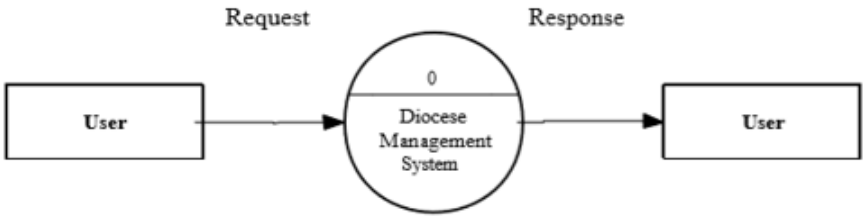


Figure 1: Level 0 Data Flow Diagram

Level 1 for Super Admin

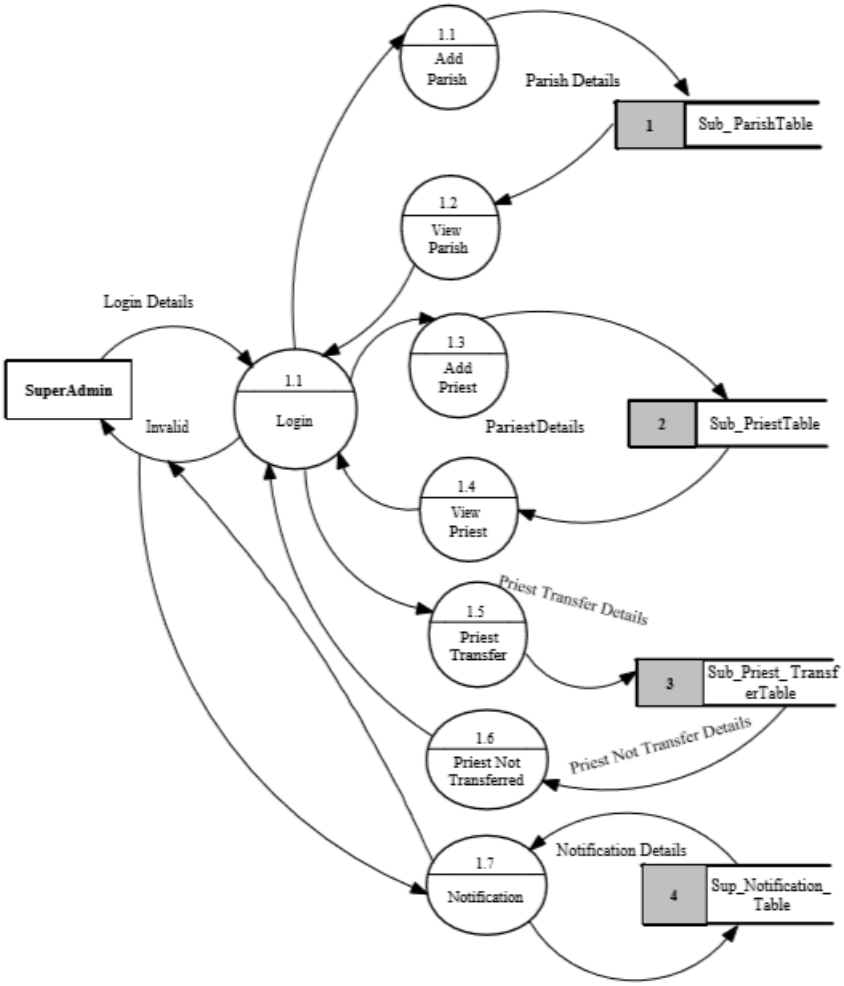


Figure 2: Level 1 Data Flow Diagram For Super Admin

Level 1 for Sub Admin

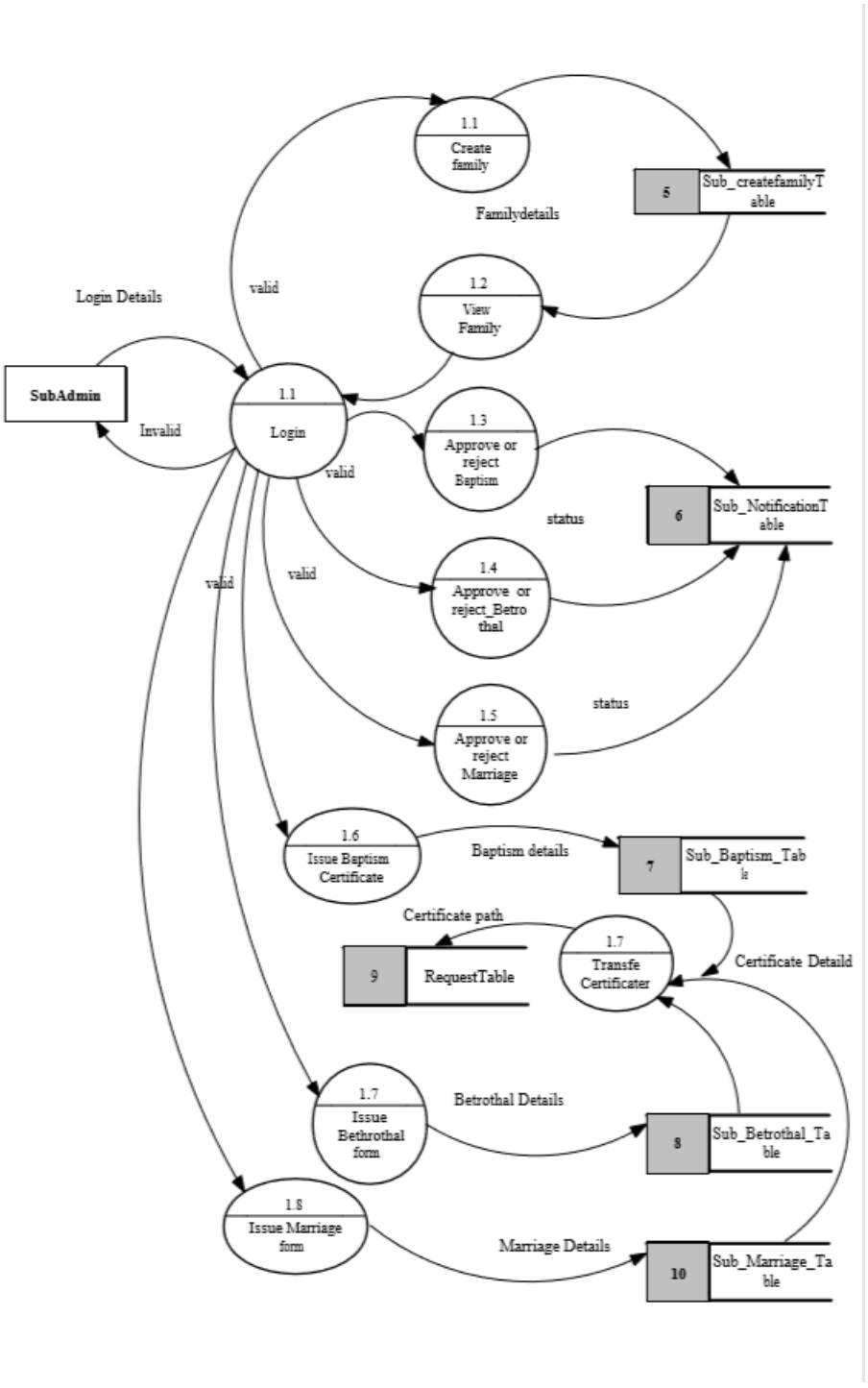


Figure 3: Level 1 Data Flow Diagram For Sub Admin

Level 1 for Parish Members

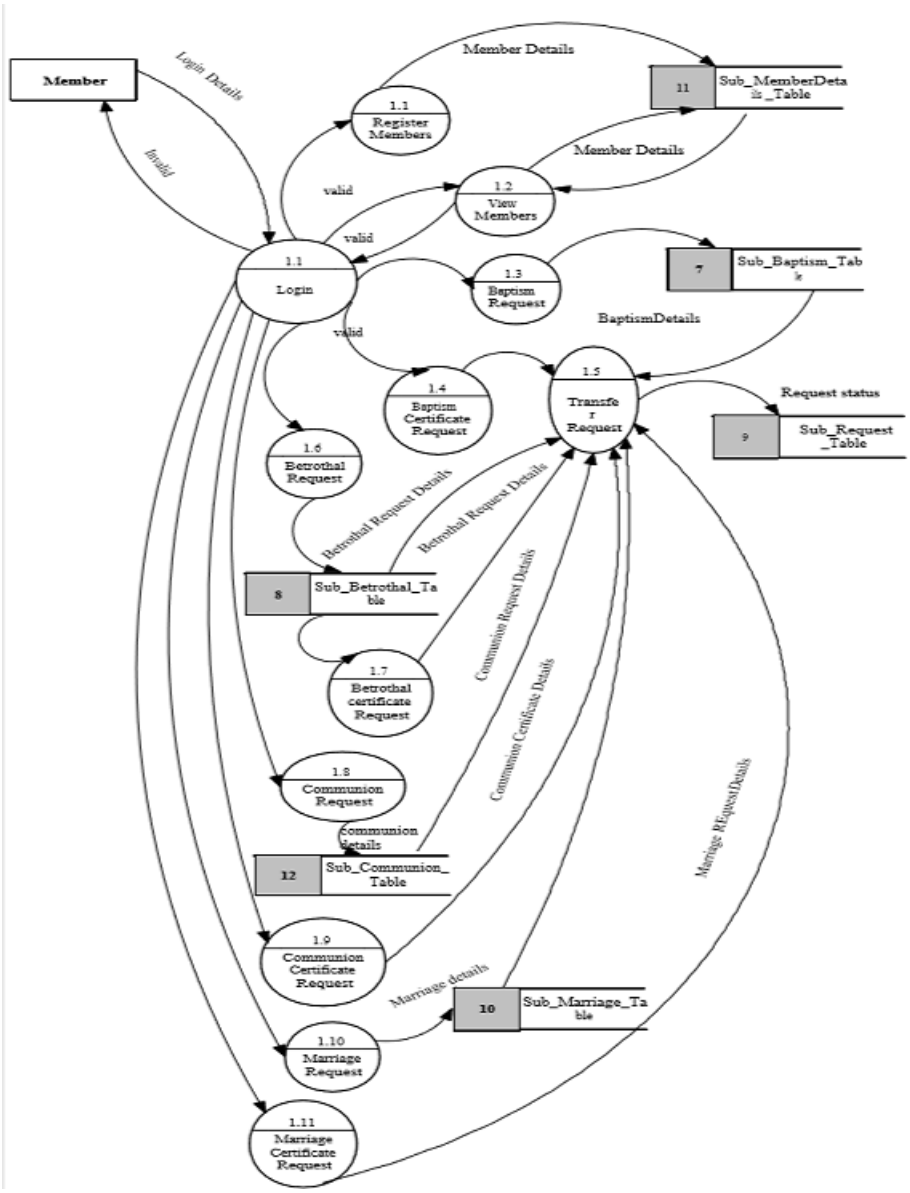


Figure 4: Level 1 Data Flow Diagram For Parish Members

3.4 Unified Modelling Language (UML)

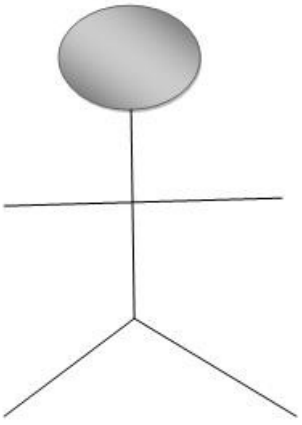
UML is a way of visualizing a software program using a collection of diagrams.The notation has evolved from the work of GradyBooch, JamesRumbaugh, IvarJAcobson and the Rational Software Corporation to be used for object - oriented design, butithas since been extended to cover a wider variety of software engineering projects.

3.5 Use Case Diagram

To model a system, the most important aspect is capture the dynamic behavior. To modify a bit in details, dynamic behavior of the system when it is running or operating. So only behavior is not sufficient to model a system rather dynamic behavior is more important than static behavior. In UML there are five diagrams available to model dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature there should be some internal or external factors for making the interaction. These internal and external agents are known as actors. So use case diagram consists of actors, use case and their relationships. The diagram is used to model the system of an application. A single use case diagram captures a particular functionality of a system.

Use case Diagram objects:

- Actor



Actor is a use case diagram in an entity that performs a role in one given system.

This could be a person, organization or an external system usually drawn like skeleton.

- Use case System

  A use case represents a function or an action within the system. Its drawn as an oval and named with the function.

- Package
  Package is another optional element that is extremely useful in complex diagrams. Similar to use class diagrams, packages are used to group together use cases

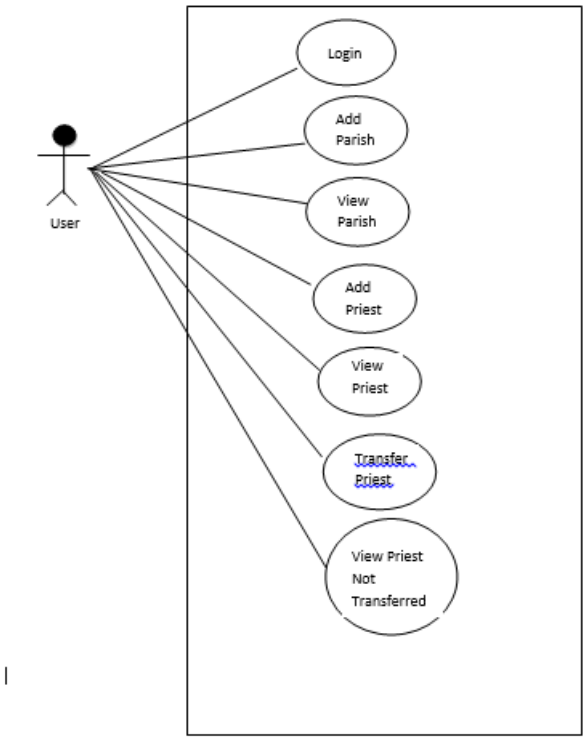


Figure 5: Usecase Diagram for Super Admin

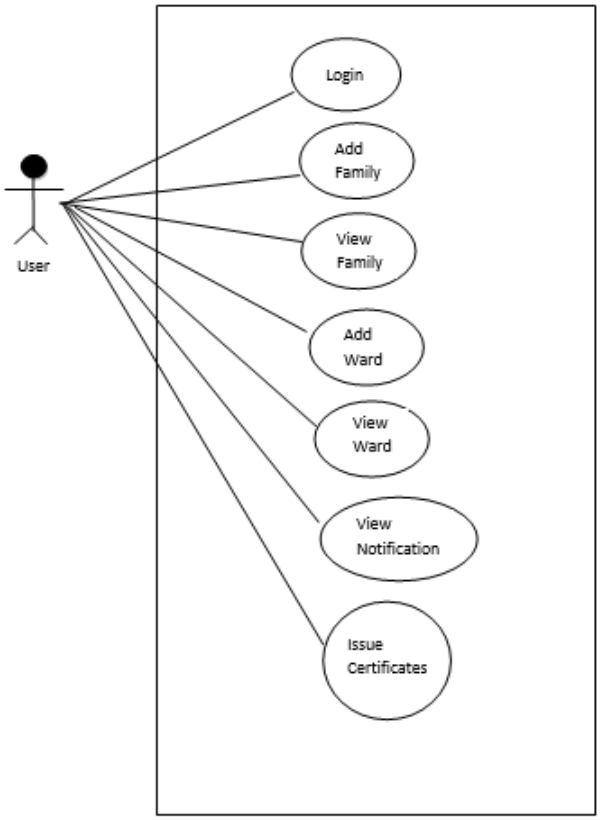


Figure 6: Usecase Diagram for Sub Admin

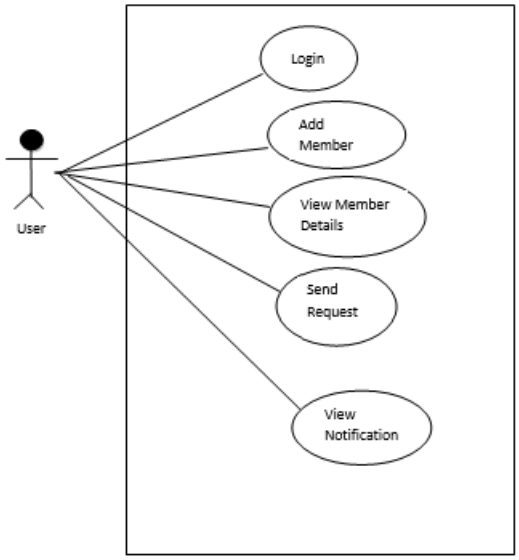


Figure 7: Usecase Diagram for Member

3.5.1 Sequence Diagram

UML sequence diagrams are used to represent or model the flow of messages,events and actions between the objects or components of a system.Time is represented in the vertical direction showing the sequence of interaction of the header elements.

Sequence Diagrams are used primarily to design,document and validate the architecture, interfaces and logic of the system by describing the sequence of actions that need to be performed to complete a task.UML sequence diagrams are useful design tools because they provide a dynamic view of the system behavior which can be difficult to extract from static diagrams or specifications.

Super Admin

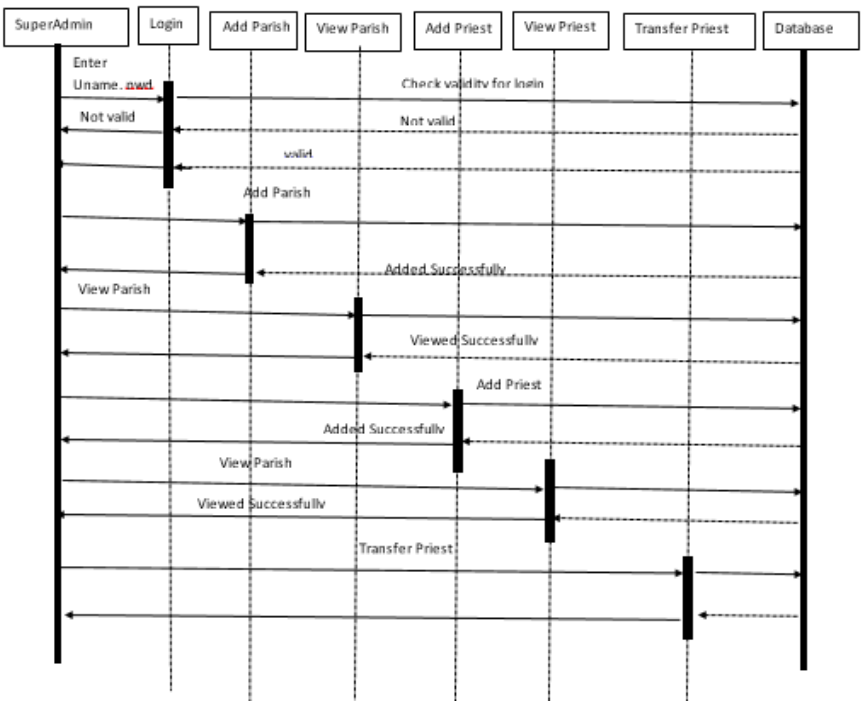


Figure 8: Sequence Diagram For Super Admin

Sub Admin

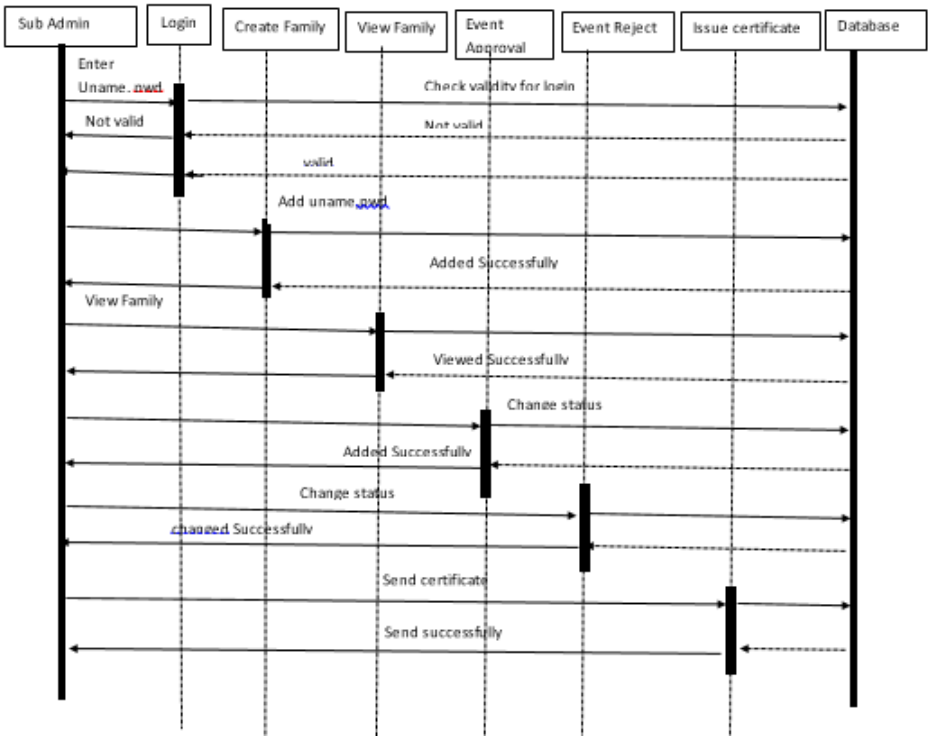


Figure 9: Sequence Diagram For Sub Admin

Parish Member

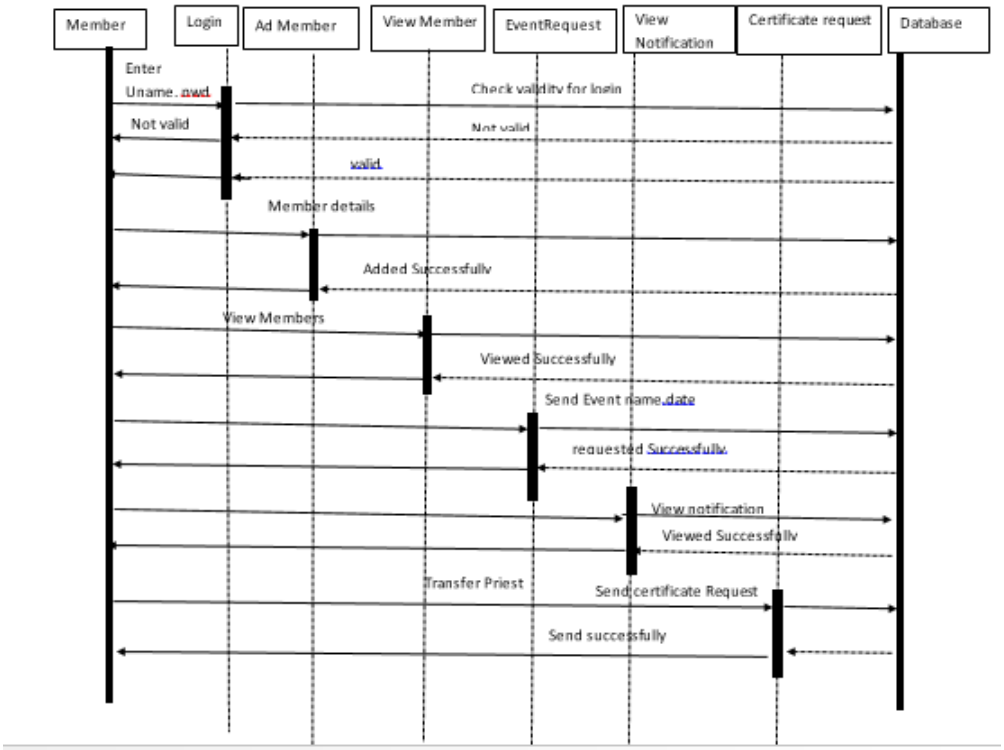


Figure 10: Sequence Diagram For Parish Member

3.5.2 Activity Diagram

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

The basic purposes of activity diagrams is similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.

It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not. It shows different flows such as parallel, branched, concurrent, and single.

Super Admin

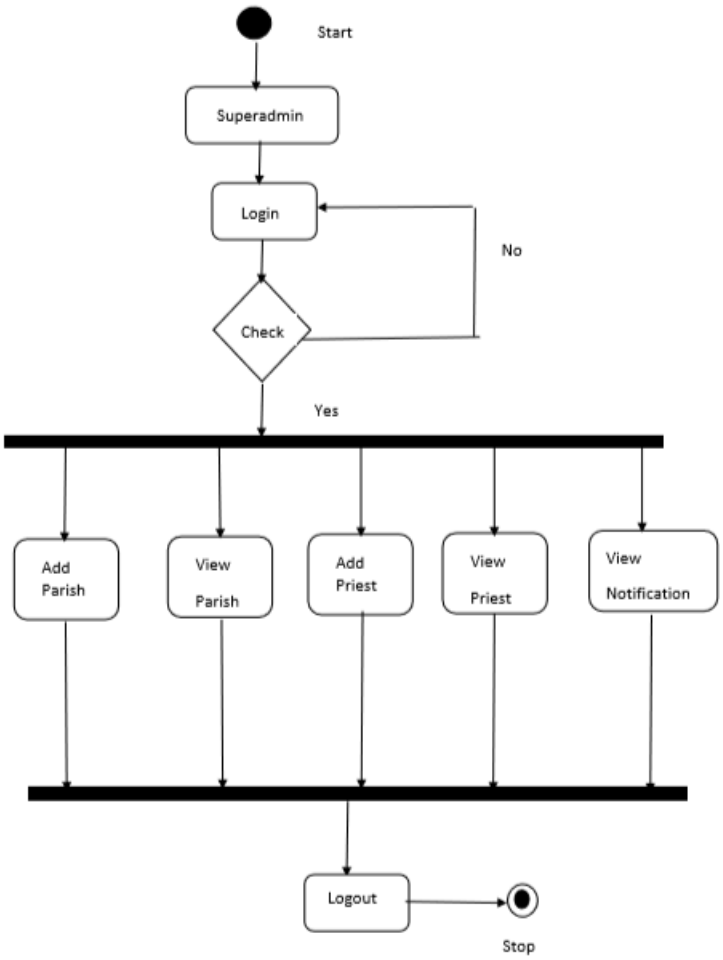


Figure 11: Activity Diagram For Super Admin

Sub Admin

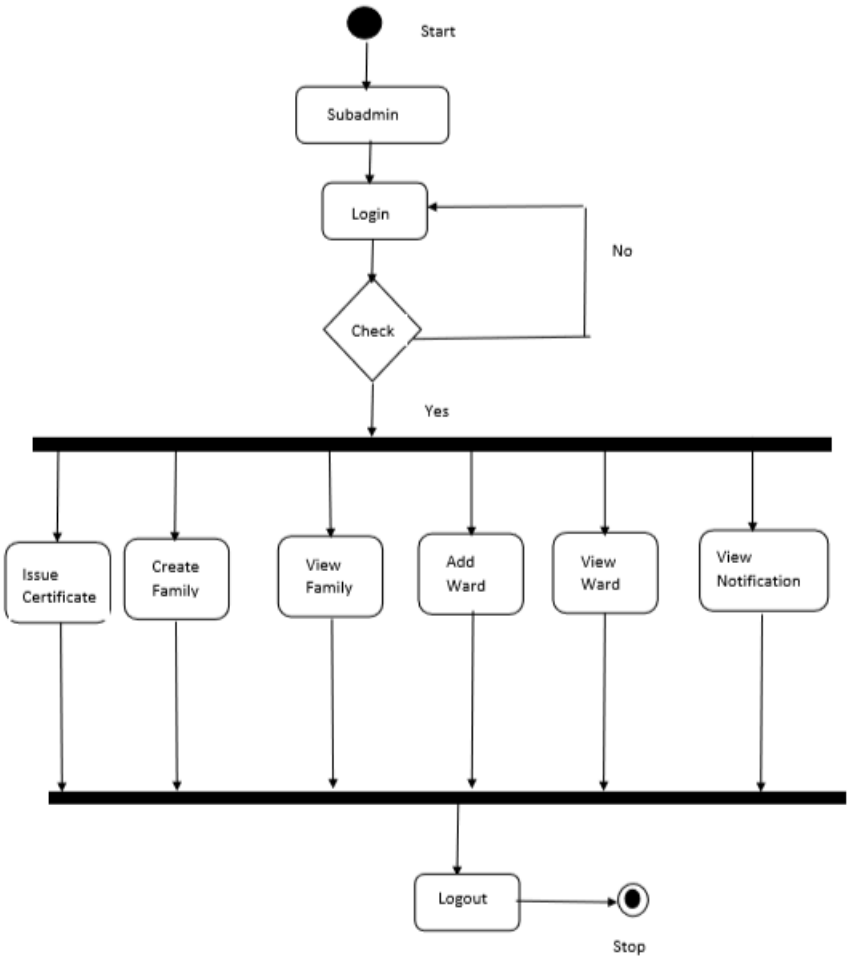


Figure 12: Activity Diagram For Sub Admin

Parish Member

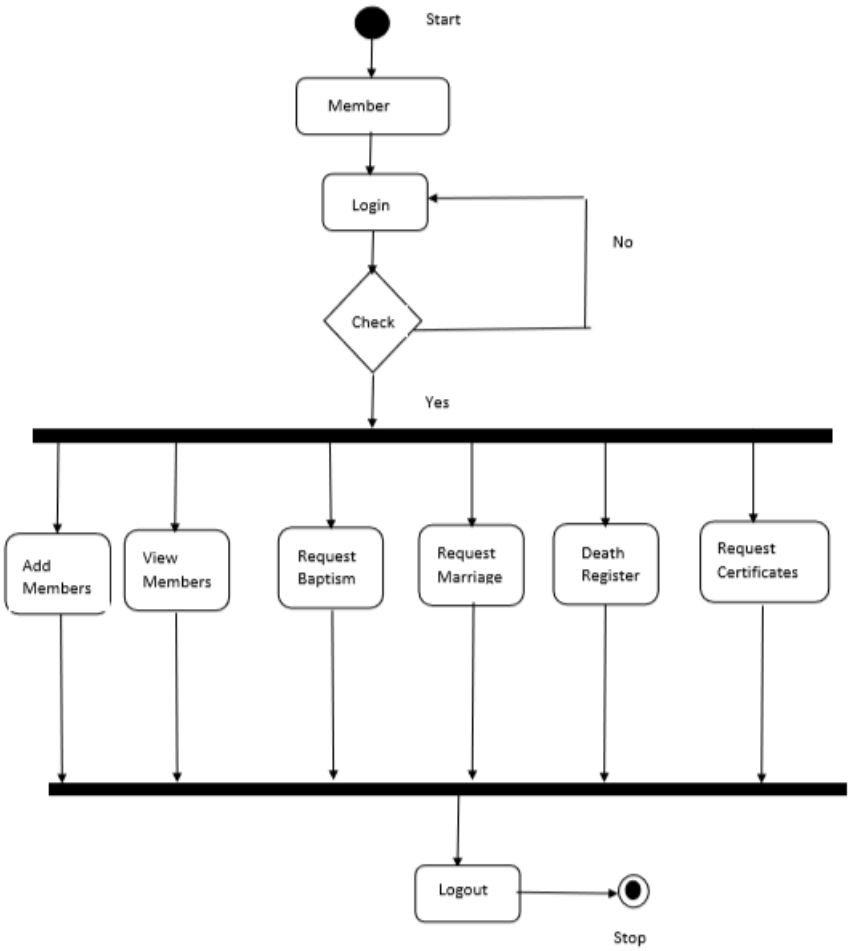


Figure 13: Activity Diagram For Parish Member

4 Tools And Platforms

4.0.1 Introduction to Microsoft.Net

Microsoft .Net is the umbrella term for the Microsofts Strategy of to move from a client centric model to a network centric model. It can be best described as the initiative that will allow the Internet to the basic of a new operating system. It is free from the constraints of hardware by making user data available from the Internet. It is important to developer because it will change the way. They develop applications by allowing them to hook on web services. The vision of .NET is globally distributed system that use XML as the universal glue to allow functions running on different computers across the world to come together in a single application. In this vision, systems from servers to wireless palmtops, will share the same general platform, with versions of.NET available for all of them, and with each of them able to collaborate with others.

4.0.2 The .NET Platform

The.Net platform is the developers perspective in the views.NET as an amalgam of a set of services, specification, guidelines and tools for incorporating the .NET vision. It includes the .NET infrastructure and tools to build and operate a new generation of smart Internet devices. As of the developer .NET platform is something that helps to put the .NET vision into a reality and hence helps the developers in providing the user with the .NET experience.

4.0.3 The .NET Products

The .NET products will include a whole range of tools and servers that rely of on XML as a language to describe data and SOAP (Simple Object Access Protocols) as protocol for transmission of data between products. This includes Microsoft Windows.NET, MSN.NET, Personal subscription services, Microsoft visual studio.NET and Microsoft centric for .NET.

4.0.4 The .NET Services

The .NET services will include all the web services and other corporate services provide by the third party vendors. A vast range of partners and developers will have the opportunity to produce corporate and vertical services built on the .NET platform.

4.0.5 The .NET Framework

The .NET framework is an environment for building, deploying and running web services and other application. Microsoft.NET framework is a standard that aims at integrating web application and services development to enable deployment and maintenance of HTTP and XML. It goes beyond development to enable deployment and maintenance of application and services along with handling their scalability and reliability.

4.0.6 Common Language Runtime (clr)

The .NET framework provides a runtime environment called the Common Language Runtime, which manages the execution of code and provides services that make the deployment process easier. Compilers and tools expose the run times functionality and enable you to write code that benefits from this managed execution environment. The Common Language Runtime makes it easy to design components and applications whose objects interact across language. Objects written in different language can communicate with each other and their behaviors can be tightly integrated.

4.0.7 Web Services

Web Services are the applications that delivered ad services that can be integrated with other web services using Internet Standards. In other words, it is an URL addressable resource that returns information to client who wants to use it and component base programming techniques are being increasingly used to develop web application.

4.0.8 ASP.NET

Active Server Page is the server side technology for creating dynamic web pages. The technology basically used scripting interspersed with in HTML to generate information on the fly, whenever a client requested, an ASP page, the script embedded in the HTML pages was interpreted by scripting engine on the web server and execute to generate HTML.

This HTML tags and displayed the output on the browser section. ASP.NET is server side web technology that can be used to develop dynamic and scalable web applications. These applications can connect to database, interact with user and provide asynchronous services. It is used on a server to build powerful web applications. ASP.NET offers several important advantages over previous web development model.

4.1 SQL SERVER 2015

SQL Server is Database Management System (DBMS) developed and marketed by Microsoft. This system is the most important part of Microsoft back office an enterprise suite of client server application. The most important aspects of SQL SERVER are: - SQL Server is easy to use, SQL Server scales from a laptop to symmetric multiprocessor systems, SQL Server provides data warehousing feature that until now have only been available in oracle and other more expensive DBMS. It is relatively easy to manage through the use of a graphical computing environment for almost every task of system and database administration.

4.2 GITHUB

Version control is a system that manages changes to a file or files.These changes are kept as logs in a history, with detailed information on what file(s) was changed, what was changed within the file, who changed it, and a message on why the change was made. This is extremely useful, especially when working in teams.To understand how incredibly powerful version control is How many files of different versions of a manuscript or thesis do you have laying around after getting feedback from your supervisor or co-authors?

Have you ever wanted to experiment with your code or your manuscript and need to make a new file so that the original is not touched ?Have you ever deleted something and wish you hadnt ? Have you ever forgotten what you were doing on a project ? All these problems are fixed by using version control (git)!

Git Hub History

In this project **"DIOCESE MANAGEMENT SYSTEM"**, maintained a Git Hub Repository to store the whole project details to know about the changes made. https://github.com/bismimariajose123

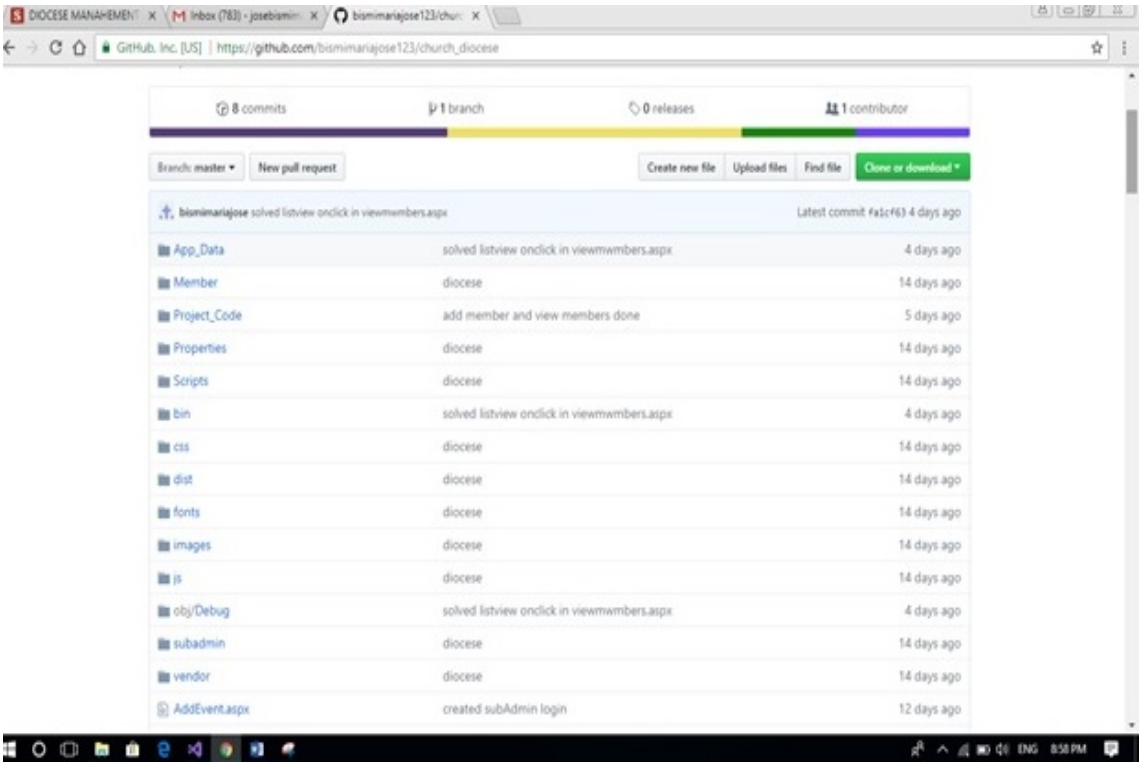


Figure 14: Git History:Figure1

5 SYSTEM TESTING

5.1 Testing Methodologies And Strategies

Software testing is an integral part of to ensure software quality, some software organizations are reluctant to include testing in their software cycle, because they are afraid of the high cost associated with the software testing .There are several factors that attribute the cost of software testing. Creating and maintaining large number of test cases is a time consuming process. Furthermore, it requires skilled and experienced testers to develop great quality test cases.

Even with the wide availability of automation tools for testing, the degree of automation mostly remains at the automated test script level and generally significant amount of human intervention is required in testing. In addition data collected, as testing is conducted provides a good indication of software quality as a while. The debugging process is the most unpredictable part of testing process. Testing begins at the module level and work towards the integration of entire computer based system. No testing is completed without verification and validation part.

The goal of verification and validation activities are to access and improve the quality of work products generated during the development and modification of the software. Testing plays a vital role in determining the reliability and efficiency of the software and hence is very important stage in software development. Tests are to be conducted on the software to evaluate its performance under a number of conditions. Ideally, it should do so at the level of each module and also when all of them are integrated to form the completed system.

In the project DIOCESE MANAGEMENT SYSTEM the testing has been successfully handled with the modules. The test data was given to each and every module in all respect and got the desired output. Each module that has been tested is found working properly.

5.1.1 Unit Testing

Here we test each module individually and integrated the overall system. Unit testing focuses verification efforts even in the smallest unit of software design in each module. This is known as module testing. The modules of the Law Helper are tested separately. This testing is carried out in the programming style itself. In this testing each module is focused to work satisfactorily as regard to expected output from the module. There are some validation checks for the fields. Unit testing gives stress on the modules of Law Helper independently of one another, to find errors. Different modules are tested against the specifications produced during the design of the modules. Unit testing is done to test the working of individual modules with test servers. Program unit is usually small enough that the programmer who developed it can test it in a great detail. Unit testing focuses first on that the modules to locate errors. These errors are verified and corrected and so that the unit perfectly fits to the project.

5.2 Test Cases

S.no	Page	Test Step	Expected Result	Actual Result	Status	Remarks (if any)
1	Admin Home Page	Login in to Admin Page	Should display homepage of admin with menus	Displays Admin home page	Pass	None
2	Admin Home Page	Click on Home menu	Should display homepage of admin with menus	Displays Admin home page	Pass	None
3	Admin Home Page	Click on Control Panel menu	Redirect to Control panel page	Displays Control Panel page with options	Pass	None
4	Control Panel Page	Click on Home Option	Redirect to Admin home page	Displays Admin home page	Pass	None
5	Control Panel Page	Click on Add Ward Option	Displays textbox and button	Displays textbox and button	Pass	None
6	Control Panel Page	Enter Ward Name and click on Add button	Ward Name get added and displays in the gridview .	Ward Name get added and displays in the gridview .Edit and Delete options appears.	Pass	None
7	Control Panel Page	Click on Edit option and change the Ward name. Then click Update	The Ward Name get updated	The Ward Name get updated and displayed in gridview	Pass	None
8	Control Panel Page	Click on Edit option and change the Ward name. Then click Cancel	Should make no changes to values.	No changes were made.	Pass	None
9	Control Panel Page	Click on Delete option.	Should delete the row.	Deletes the selected row.	Pass	None

S.no	Page	Test Step	Expected Result	Actual Result	Status	Remarks (if any)
10	Control Panel Page	Click on Add Family Option.	Displays textbox, button and dropdown list.	Displays textbox, button and dropdown list.	Pass	None
11	Control Panel Page	Enter username, password and select user type. Then Click Insert Button.	Value should get inserted in Database and should display in the gridview.	Value gets inserted and displays in gridview.	Pass	None
12	Control Panel Page	Click on Edit option and change the username, password .Then click Update.	The value gets updated.	Value gets updated and displays in gridview.	Pass	None
13	Control Panel Page	Click on Edit option and change the Values. Then click Cancel.	Should make no changes to values.	No changes were made.	Pass	None
14	Control Panel Page	Click on Delete option.	Should delete the row.	Deletes the selected row.	Pass	None
15	Go to Home.	Click on Notification.	Should display values if present Else nothing is displayed.	Nothing is displayed.	Pass	None
16	Notification Page	Click on Logout.	Should return to Login page.	Returns to Login Page.	Pass	None
17	Member Page	Login as Registered Member.	Should display Home Page of Member.	Displays Home Page.	Pass	None
18	Member Page	Click on Home Option.	Redirect to Member home page.	Displays Member home page.	Pass	None
19	Member Page	Click on Registry Option.	Should display a list of dropdown menus.	Displays a list of dropdown menus.	Pass	None

S.no	Page	Test Step	Expected Result	Actual Result	Status	Remarks (If any)
20	Member Page	Click on List All Family in dropdown menu	Should display a search box, button and gridview with values like username,usertyp eand two buttons.	Displays a search box, button and gridview with values like username, user-type and two buttons.	Pass	None
21	List All Family Page	Click on View Members button in gridview.	Should redirect to View_members page.	Redirect to View_members page.	Pass	None
22	View Members Page	Click on List All Family <-Back Link.	Should redirect to List All Family Page.	Redirect to List All Family Page.	Pass	None
23	List All Family Page	Click on Register Members button in gridview.	Should redirect to Registration page.	Redirect to Registration page.	Pass	None
24	Registration page	Fill in all the details and click Register Button.	Should redirect to List All Family Page and details get displayed.	Redirect to List All Family Page.	Pass	None
25	List All Family Page.	Click on Edit option and change the Values .Then click Update.	The value gets updated.	Value gets updated and displays in gridview.	Pass	None
26	List All Family Page.	Click on Edit option and change the Values. Then click Cancel.	Should make no changes to values.	No changes were made.	Pass	None
27	List All Family Page.	Click on Delete option.	Should delete the row.	Deletes the selected row.	Pass	None
28	Member Page (Baptism Request)	Select Baptism Request from Registry menu Option.	Should redirect to Baptism Request page.	Redirects to Baptism Request page.	Pass	None

S.no	Page	Test Step	Expected Result	Actual Result	Status	Remarks (if any)
29	Baptism Request page	Fill details and click on Request button.	Should sent a request to Admin and redirect to Notification Page. Status=null.	Redirects to Notification Page and displays the data in gridview. Status=null	Pass	None
30	Notification Page	Click on Logout.	Should redirect to Login Page.	Redirects to Login Page.	Pass	None
31	Login Page	Login as Admin.	Should redirect to Admin Home Page.	Redirects to Admin Home Page.	Pass	None
32	Admin Home Page	Click on View Notifications option.	Should display the notifications / request (Baptism Request).	Displays the request (Baptism Request) with Status "null".	Pass	None
33	View Notifications Page	Approve Baptism Request. Click on Edit option in gridview, choose Approve from dropdown- list under Status and click update.	Should update the Status to Approved.	Status changed to Approved.	Pass	None
34	View Notifications Page	Click on Logout.	Should redirect to Login Page.	Redirects to Login Page.	Pass	None
35	Login Page	Login as Member.(who requested baptism)	Should redirect to Member Home Page.	Redirects to Member Home Page.	Pass	None
36	Member Home Page	Click on Notification Option.	Should redirect to Notification page and display a link "Click to register" and Status is Approved.	Displays a link "Click to register" and Status is Approved.	Pass	None

S.no	Page	Test Step	Expected Result	Actual Result	Status	Remarks (if any)
37	Notification page	Click on "Click to register" link.	Should redirect to Baptism Form page and auto fills some details about the person to be baptized.	Redirects to Baptism Form and auto fills details.	Pass	None
38	Baptism Form Page	Fill the details and click submit button.	Should Redirect to List All Family Page.	Redirects to List All Family Page.	Pass	None
39	Notification Page	Click on Notification menu.	Should redirect to Notification Page. Status is "Baptism registered".	Status is "Baptism registered".	Pass	None
40	List All Family Page	Click on Register Members button in gridview.	Should redirect to Member Registration Form.	Redirects to member registration form.	Pass	None
41	Member Registration Form	Type the Baptized Child's Name.	Should Auto fill some details. Since the child is a member after baptism.	Auto fill some details.	Pass	None
42	Member Registration Form	Click on Register button.	Child must be Added as Member. Redirects to List All Family Page. Displays the data of child in gridview..	Displays the data about child in gridview. The Count of family members is increased.	Pass	None
43	List All Family Page	Click on View more details button.	Redirects to view more details page. Should display all details about the person selected.	Displays all details about the person selected.	Pass	None
44	View more details page	Click on Back link.	Should redirects to previous page.	Redirects to previous page.	Pass	None

S.no	Page	Test Step	Expected Result	Actual Result	Status	Remarks (if any)
45	Notification Page (Baptism Certificate download)	Click on Notification menu. And click on Certificate link appears.	Should download a pdf of Baptism Certificate and store in D: /.	Downloads certificate onto D: /.	Pass	None
46	View Members Page (Death Certificate Request).	Choose "Death" from Option dropdownlist in gridview and click on Send Request button.	Should redirect to Death Request Page.	Redirects to Death Request Page.	Pass	None
47	Death Request Page.	Fill details and click Cancel button.	Should Return to View Members Page (previous page).	Returns to previous page.	Pass	None
48	Death Request Page.	Fill details and click Request button.	Should send a request to Admin.	Returns to View all Members page after sending a request to admin.	Pass	None
49	Admin Home Page	View Notification option.	Should displays the Request for death certificate.	Displays the request for death certificate.	Pass	None
50	View Notification Page (Approve death certificate request)	Click on edit option and select "Death Certificate Approved" option from dropdownlist.	Status must be changed to Death Certificate Approved. Register Death button appears.	Status changed to Death Certificate Approved. Button is visible.	Pass	None
51	View Notification Page (Add member to Death table)	Click on Register Death button.	Should store values to Death Table. Status changed to Death Certificate Issued.	Status changed to Death Certificate Issued.	Pass	None
52	Member Home Page	Login as Member. Click on Notification option.	Should display a certificate.	Displays a certificate.	Pass	None

53	Notification Page(Download death certificate)	Click on Certificate link.	Should download Death certificate pdf document.	Downloads Death certificate pdf document.	Pass	None
54	View Members Page	Death person identification.	Should display Death Persons name in Red color and must hide the Option Cell in gridview.	Displays Death Persons name in Red color and Option Cell becomes invisible.	Pass	None

6 SYSTEM IMPLEMENTATION

The implementation is one phase of software development. Implementation is that stage in the project where theoretical design is turned into working system. Implementation involves placing the complete and tested software system into actual work environment. Implementation is concerned with translating design specification with source code. The primary goal of implementation is to write the source code to its specification that can be achieved by making the source code clear and straight forward as possible. Implementation means the process of converting a new or revised system design into operational one. The three types of implementation are:-implementation of a computerized system to replace a manual system, implementation of a new system to replace existing one and implementation of a modified system to replace an existing one.

The implementation is the final stage and it is an important phase. It involves the individual programming ; system testing, user training , and the operational running of developed proposed system that constitute the application subsystem. The implementation phase of the software development is concerned with translating design specification in the source code. The user tests the developed system and the changes are according to the needs. Before implementation, Several tests have been conducted to ensure no errors encountered during the operation.

The implementation phase ends with an evaluation of the system after placing it into operation of time. The validity and proper functionality of all the modules of the developed application is assured during the process of implementation. Implementation is the process of assuring that the information system is operational and then allowing user to take over its operation for use and evaluation. Implementation is the stage in the project where the theoretical design is turned into a working system. The implementation phase constructs ,installs and operated the new system. The most crucial stage in achieving a new successful system is that it works effectively and efficiently.

7 SCOPE OF FUTURE ENHANCEMENT

Making enhancements is all about perfecting maintenance. It means adding, modifying or redeveloping the code to support changes in the specifications. It is necessary to keep up with changing user needs and the operational environment. More money and time is spend on perfective maintenance than on corrective or adaptive maintenance together.

In the proposed system **Diocese Management System,** future enhancements are possible in the following areas:

- Ability to transfer certificates to requester with validation .
- Email messaging.
- More events can be included.
- More up gradations are possible.

8 CONCLUSION

The project titled DIOCESE MANAGEMENT SYSTEM, has been developed for the members of the Diocese,Parish priests and Bishop to easily carry out the activities relating to churches. The project is developed as a Web Application by using ASP.NET as the front end and SQL Server 2015 as the back end.

The system has been developed in an interactive manner; the search by the system are perfect and clear. The system is flexible, user friendly and has its own full data security and all data recovery facility. The developed system has mainly three modules Super administrator, Sub Administrator and Parish Members. It is developed using ASP .NET and SQL Server.

9 APPENDIX

9.1 APPENDIX A

9.1.1 Screen Shots

Figure 15: Login Page

Figure 16: Super Admin Home Page

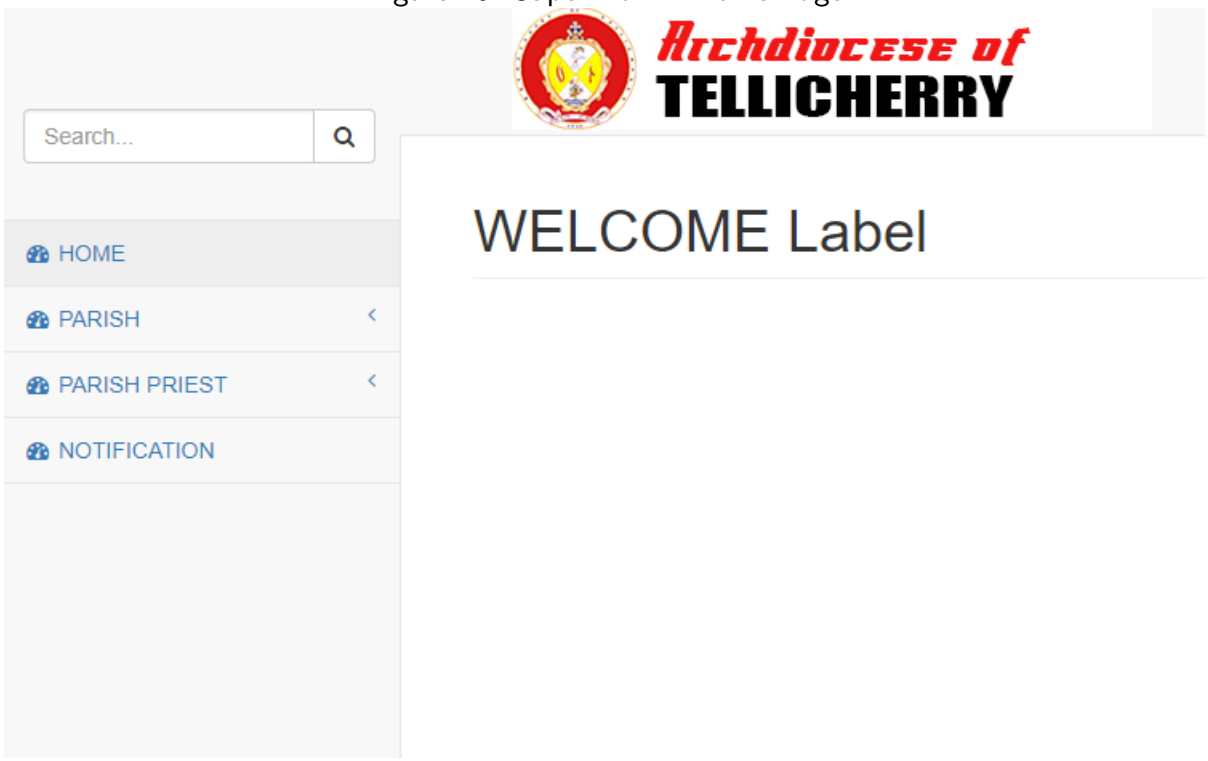


Figure 17: Super Admin Add Parish

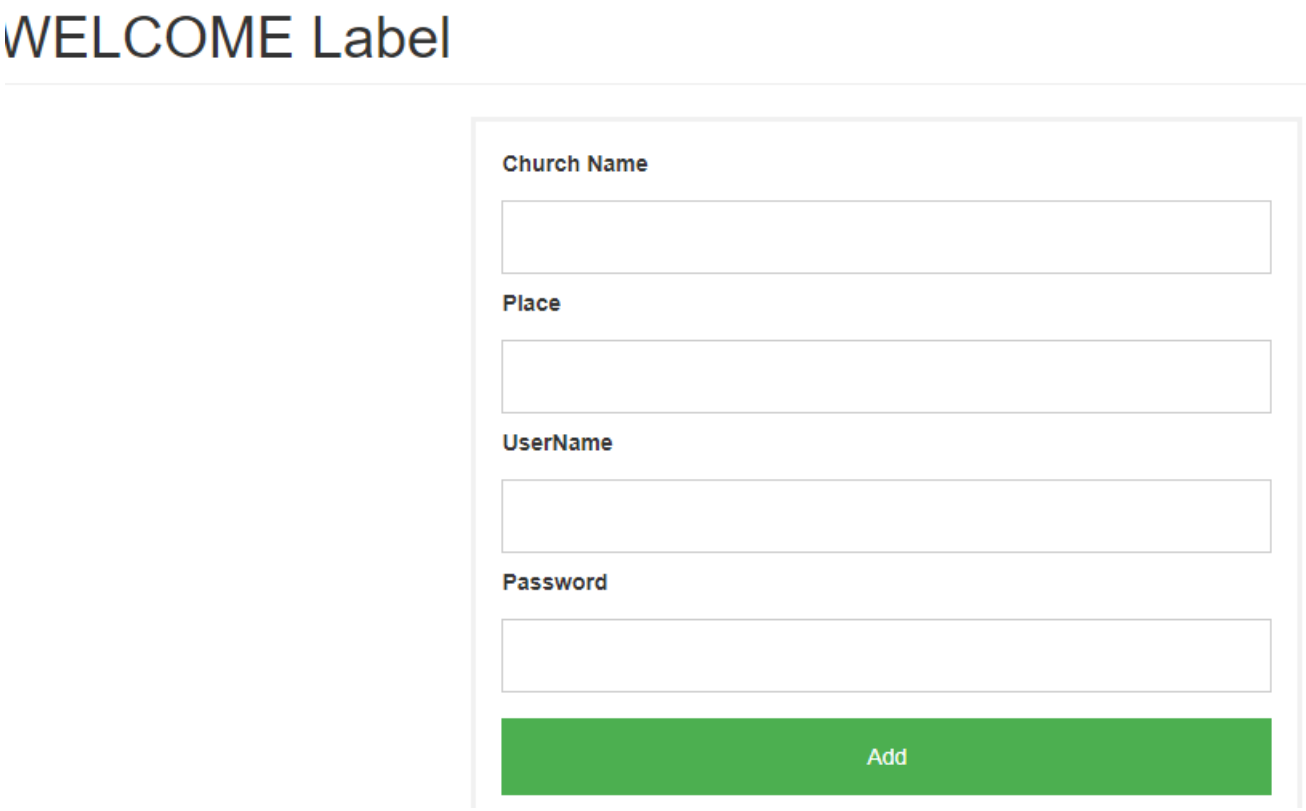
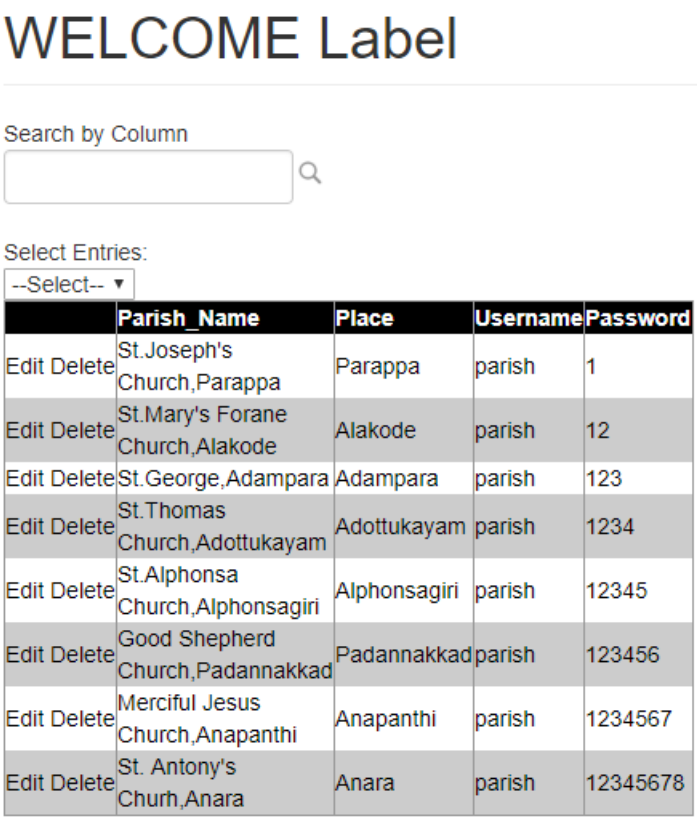


Figure 18: Super Admin View Parish



	Parish_Name	Place	Username	Password
Edit Delete	St.Joseph's Church,Parappa	Parappa	parish	1
Edit Delete	St.Mary's Forane Church,Alakode	Alakode	parish	12
Edit Delete	St.George,Adampara	Adampara	parish	123
Edit Delete	St.Thomas Church,Adottukayam	Adottukayam	parish	1234
Edit Delete	St.Alphonsa Church,Alphonsagiri	Alphonsagiri	parish	12345
Edit Delete	Good Shepherd Church,Padannakkad	Padannakkad	parish	123456
Edit Delete	Merciful Jesus Church,Anapanthi	Anapanthi	parish	1234567
Edit Delete	St. Antony's Churh,Anara	Anara	parish	12345678

Figure 19: Super Admin Add Parish Priest

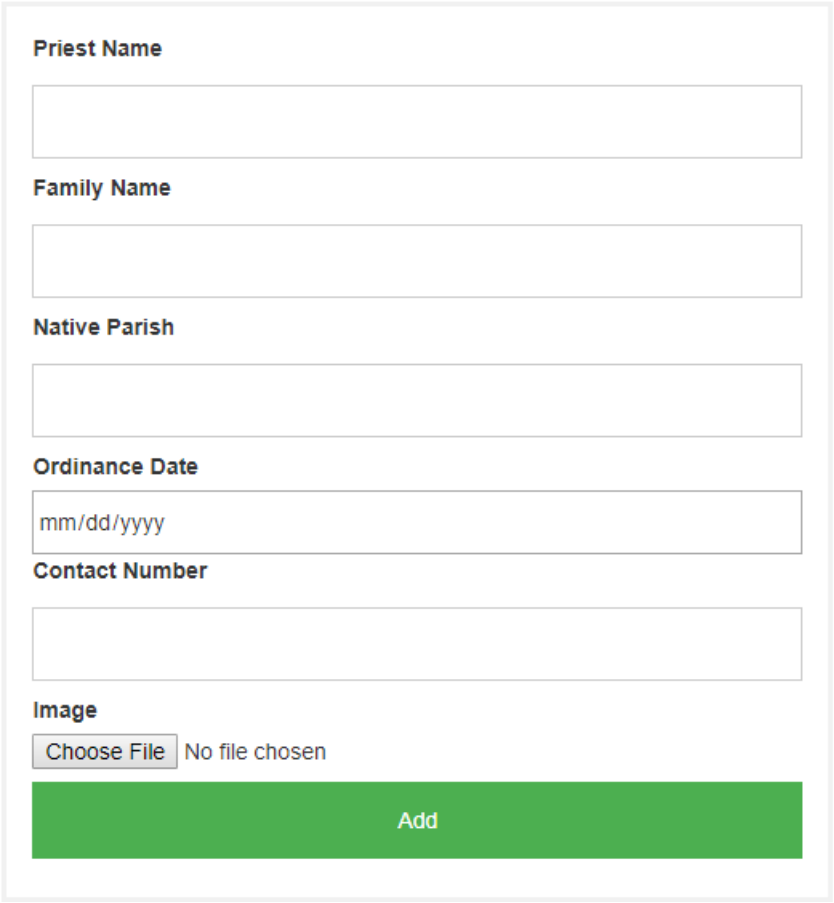


Figure 20: Super Admin View Parish Priest

WELCOME Label

Search by Column

Select Entries:

--Select--

	Priest Name	Phone No	Ordination Date	Native Place	Designation	Image	Current Parish
Edit Delete	Fr. Antony (Siby)	94476424 07	30/12/1992	Mangode St. Mary's Church	Parish Priest		St.Mary's Forane Church,Alakode
Edit Delete	Fr. Joseph (Lijo)	94472865 01	26/12/2008	Pathenpara St.Antony's Church	Parish Priest		St.George,Adampara
Edit Delete	Fr. Mathew (Sherin)	85470872 76	27/12/2012	Parappa St. Joseph's Church	Asst Vicar		
Edit Delete	Fr. Sebastian Chennoth	94478665 24	30/12/1997	Paisakary Devamatha Forane Church	Parish Priest		St.Joseph's Church,Parappa

Figure 21: Super Admin Transfer Priest

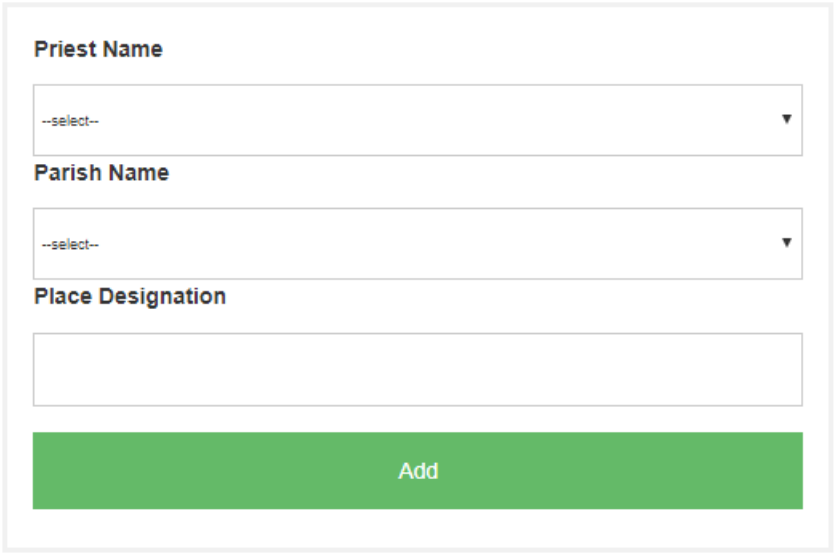


Figure 22: Super Admin Priest Not Transferred

WELCOME Label

Search by Column

Select Entries:
--Select--

	Priest Name	Phone No	Ordination Date	Native Place	Designation	Image	Current Parish
Edit Delete	Fr. Mathew (Sherin)	8547087276	27/12/2012	Parappa St. Joseph's Church	Asst Vicar		

Figure 23: Sub Admin Home Page

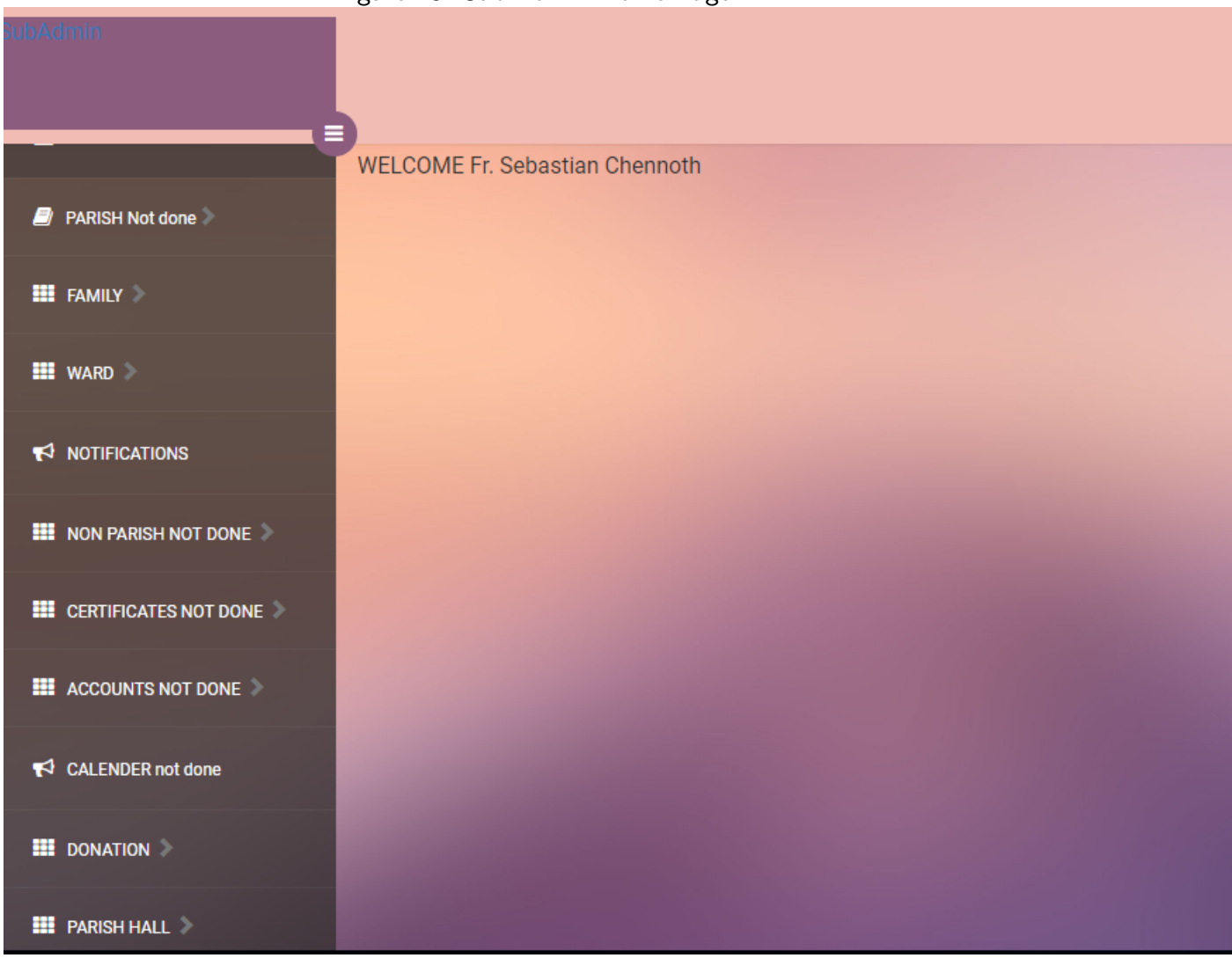


Figure 24: Sub Admin Add Parish Priest

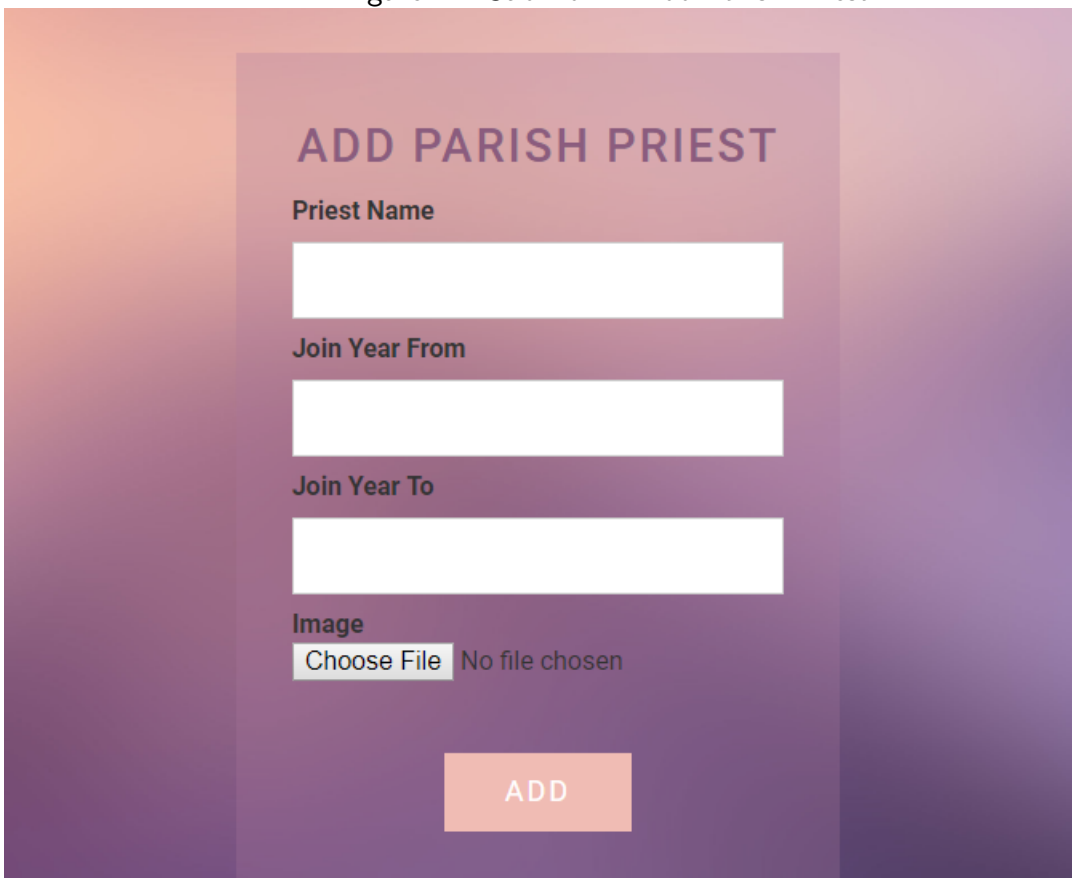


Figure 25: Sub Admin Create Family

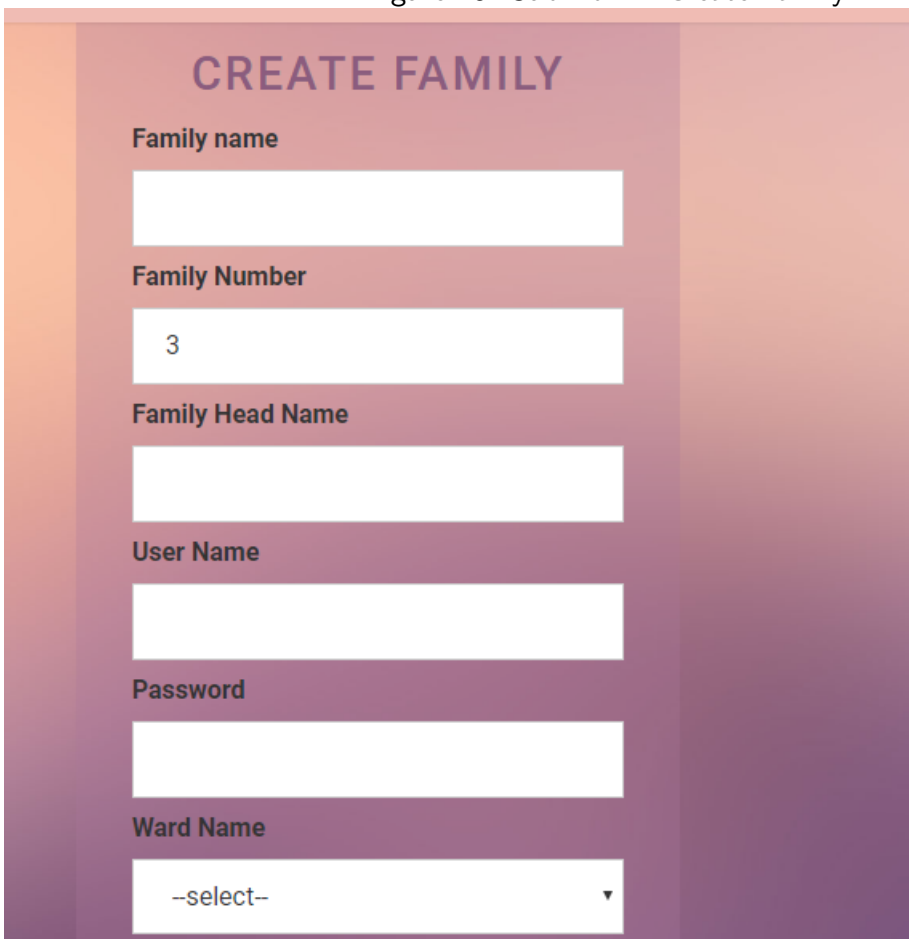
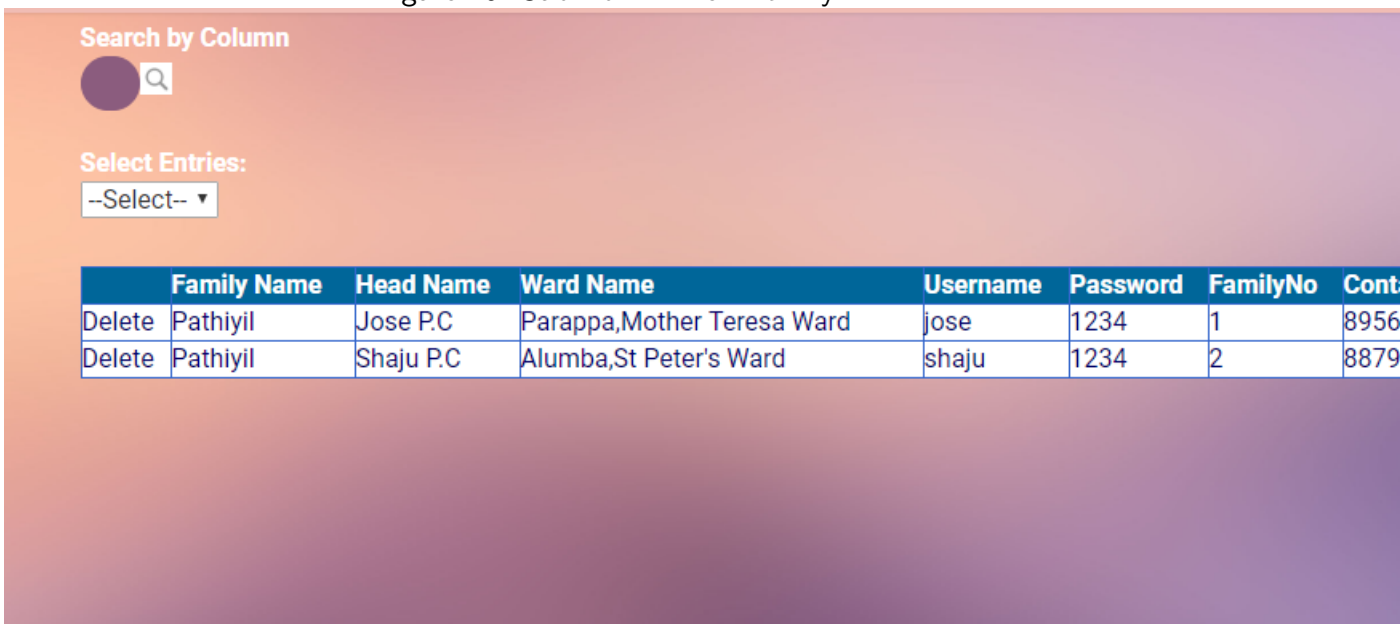


Figure 26: Sub Admin View Family



	Family Name	Head Name	Ward Name	Username	Password	FamilyNo	Cont
Delete	Pathiyil	Jose P.C	Parappa,Mother Teresa Ward	jose	1234	1	8956
Delete	Pathiyil	Shaju P.C	Alumba,St Peter's Ward	shaju	1234	2	8879

Figure 27: Sub Admin Create Ward

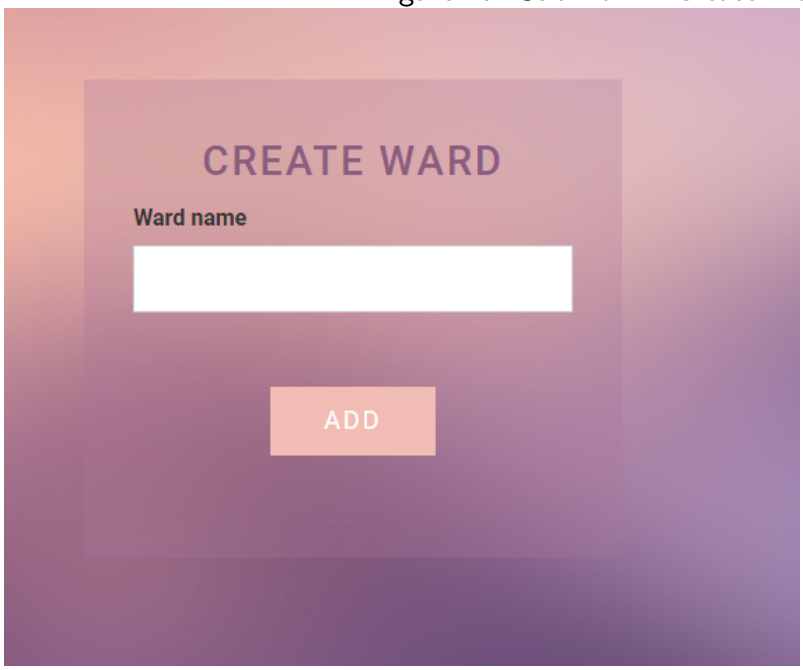


Figure 28: Sub Admin View Event

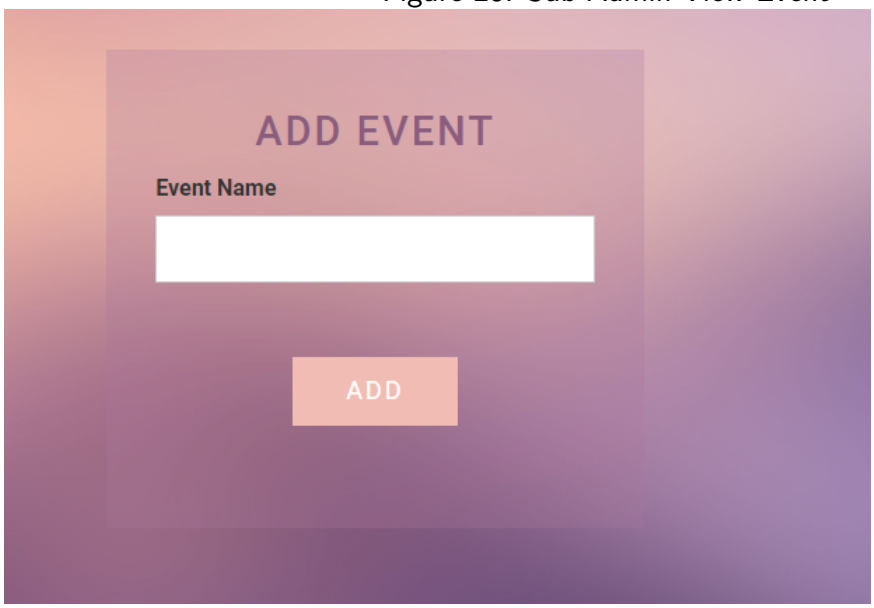


Figure 29: Sub Admin Add Hall

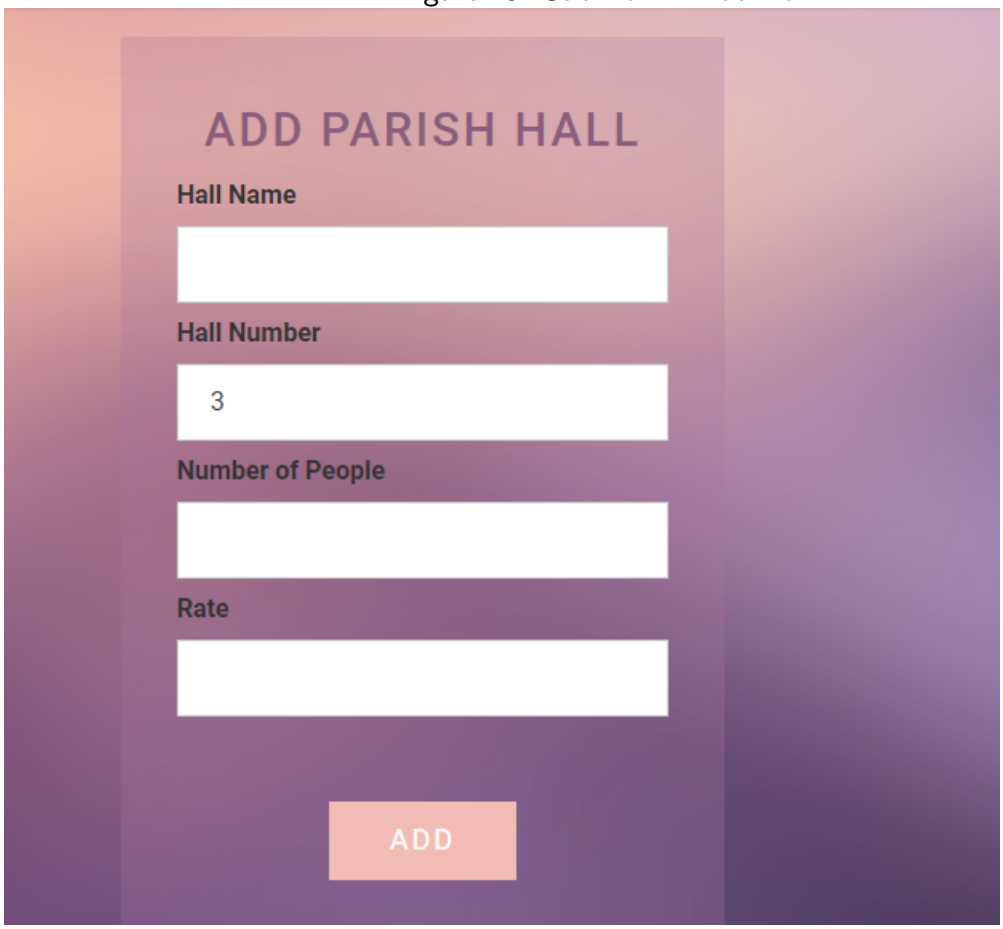


Figure 30: Member Home Page

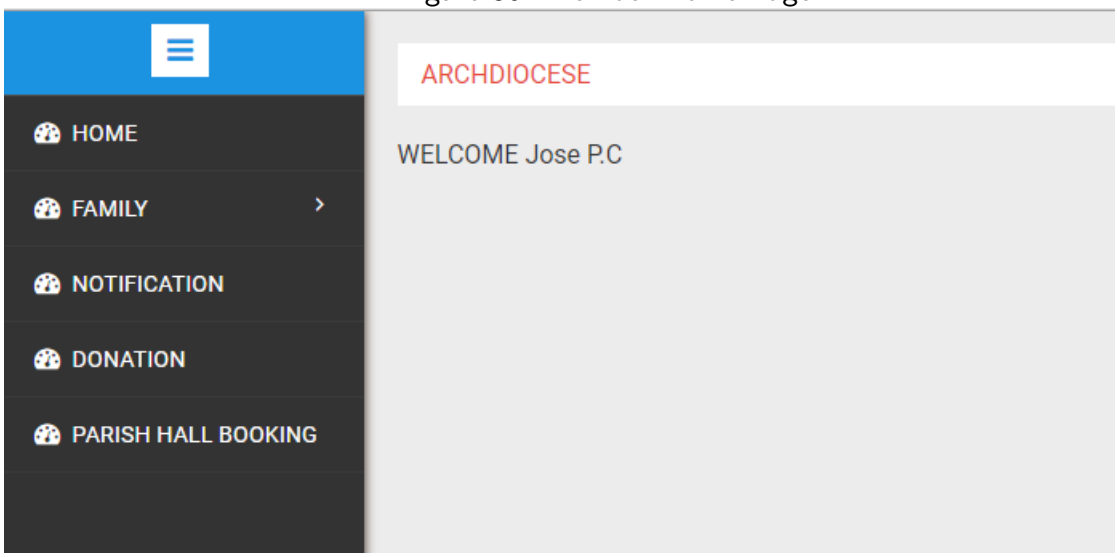


Figure 31: Member Registration

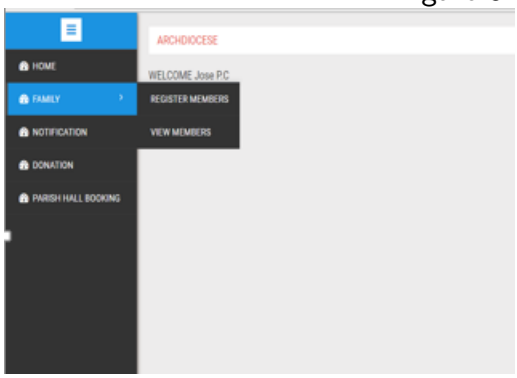


Figure 32: Enter Member Details

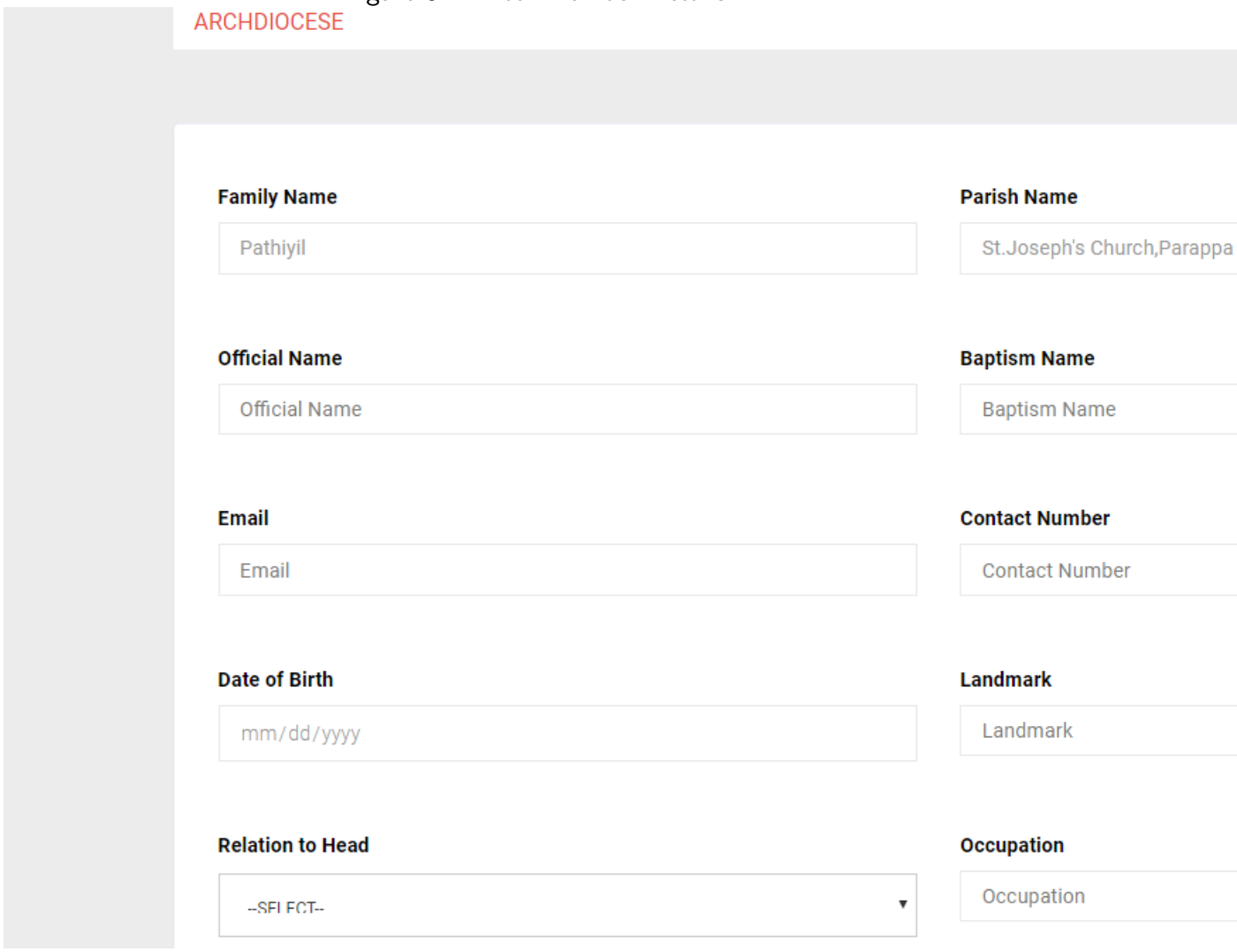


Figure 33: Enter Member Details

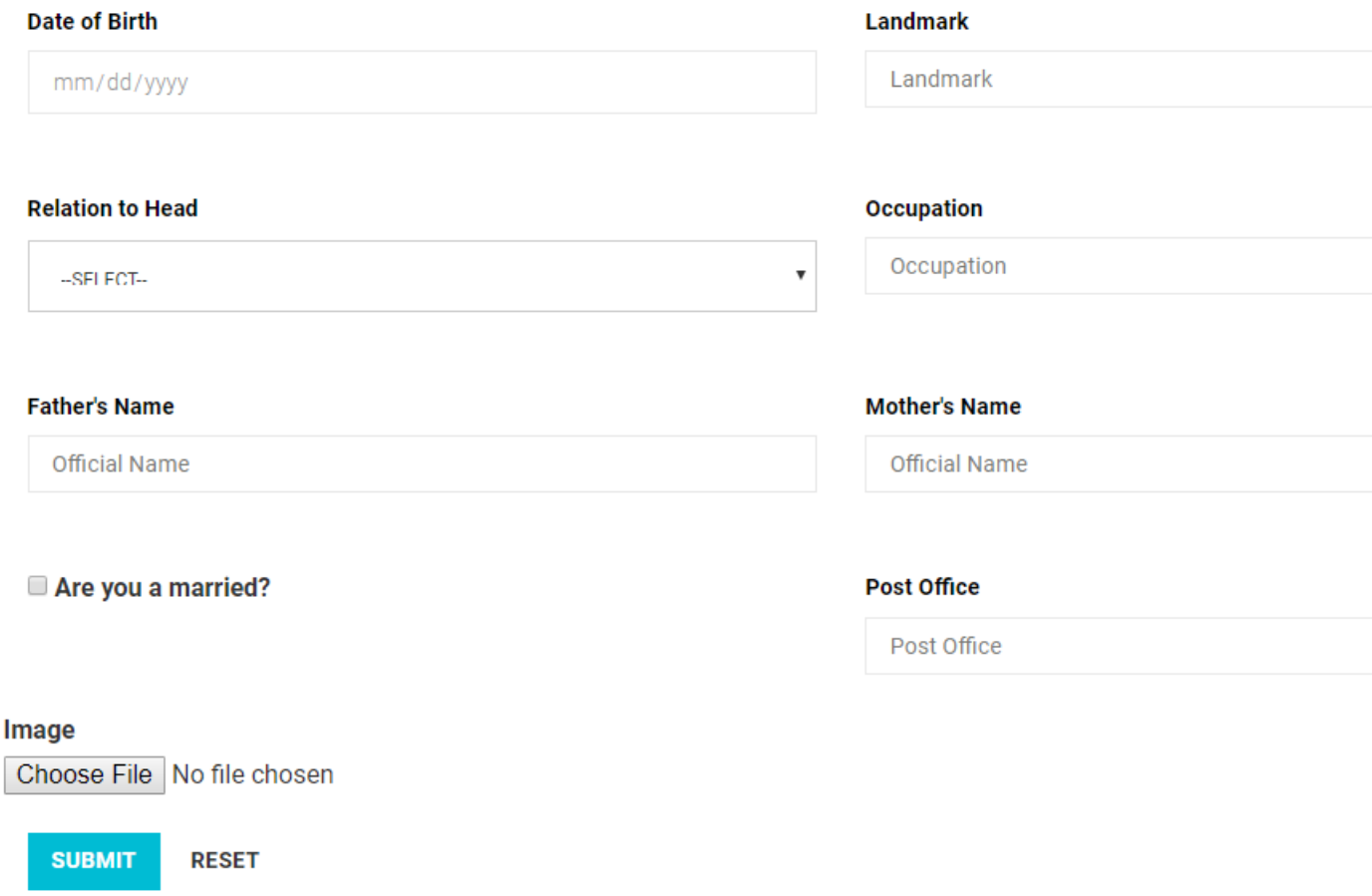


Figure 34: Communion Form

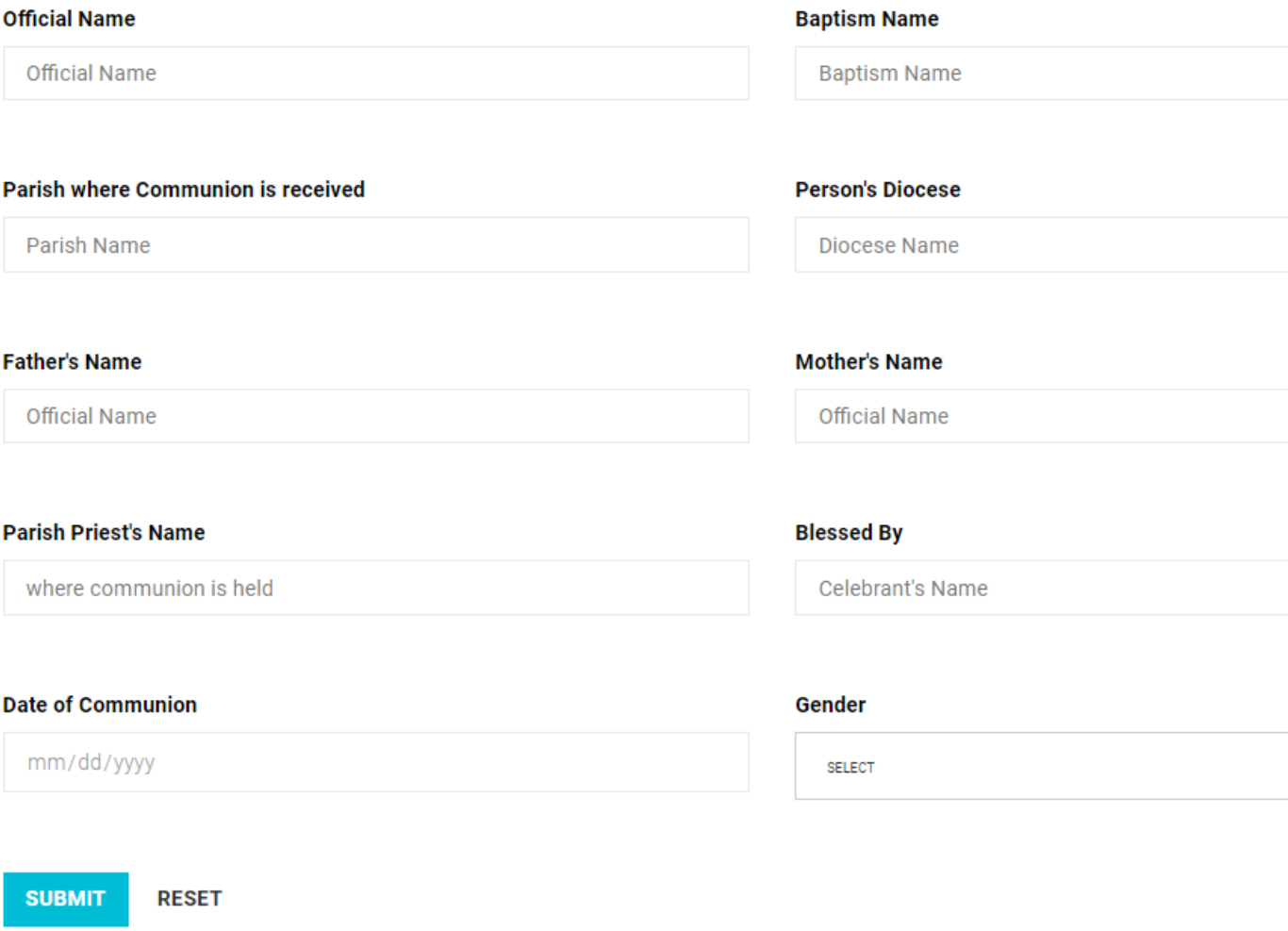
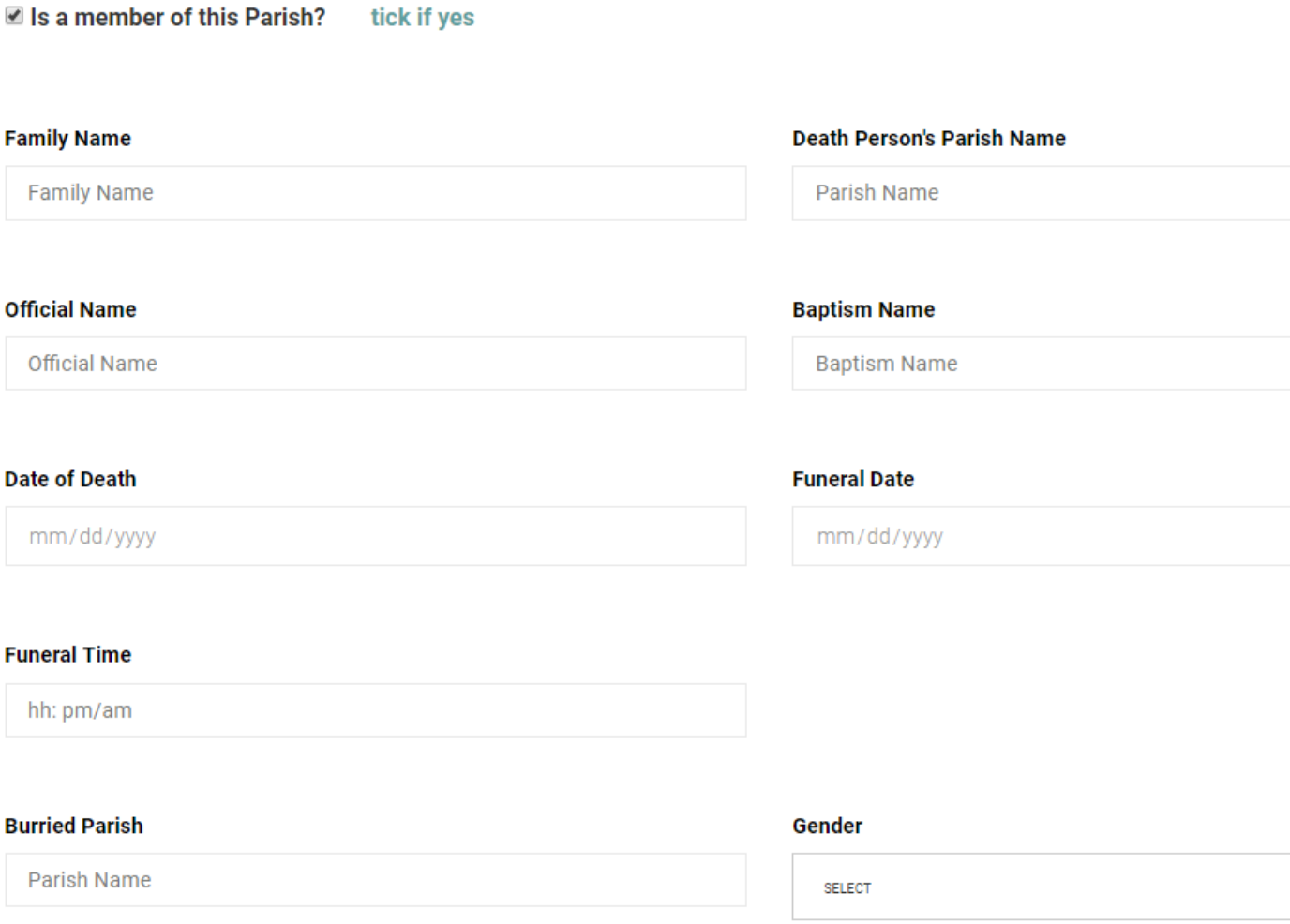


Figure 35: Death Form



9.2 APPENDIX B

9.2.1 Sample Source Code / Pseudo Code

```
using Diocese.Project_Code.SuperAdmin;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Diocese
{
    public partial class AddParish : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!Page.IsPostBack)
            {

            }
        }

        protected void BtnAddParish_Click(object sender, EventArgs e)
        {
            int result=0;
            Parish_BO objParishDetails = new Parish_BO();
            Parish_BLL objPariahDetailsBLL = new Parish_BLL();
            objParishDetails.ParishName = TBChurchName.Text;
            objParishDetails.ParishPlace = TBPlace.Text;
            objParishDetails.UName = TBusername.Text;
            objParishDetails.Passwd = TBPassword.Text;
           result=objPariahDetailsBLL.InsertParishDetails(objParishDetails);
            Response.Redirect("AddParish.aspx");

        }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace Diocese.Project_Code.SuperAdmin
{
    //Declaring  Variables
    public class Parish_BO
    {
        private string Parish_Name;
        private string Place;
        private string Username;
        private string Password;

        // Get and set values
        public string ParishName

        {
            get
            {
                return Parish_Name;
            }

            set
            {
                Parish_Name = value;
            }
        }
        public string ParishPlace

        {
            get
            {
                return Place;
            }

            set
            {
                Place = value;
            }
        }
```

```
        public string UName

        {
            get
            {
                return Username;
            }

            set
            {
                Username = value;
            }
        }
        public string Passwd

        {
            get
            {
                return Password;
            }

            set
            {
                Password = value;
            }
        }



    }

}
```

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Web;

namespace Diocese.Project_Code.SuperAdmin
{
    public class Parish_BLL
    {
        Parish_DAL objParishDAL = new Parish_DAL(); // Creating object of Dataccess

        public int InsertParishDetails(Parish_BO objParishDetails_BO) // passing Buss
        {
              try
            {
                return objParishDAL.InsertParishInformation(objParishDetails_BO); //
            }
            catch (Exception ex)
            {
                throw ex;
            }
            finally
            {
                objParishDAL = null;
            }
        }
        public DataTable GetParishDetails()
        {
            return objParishDAL.GetParishInformation();
        }

        public int Delete_Parish(int id)
        {
            try
            {
               return objParishDAL.DeleteParish(id);
            }
            catch (Exception ex)
            {
                throw ex;
            }
```

```
        }

        public int UpdateParish(Parish_BO objParishDetails_BO, int id)
        {
            try
            {
                return objParishDAL.UpdateParish(objParishDetails_BO,id);
            }
            catch (Exception ex)
            {
                throw ex;
            }

        }

        public DataTable Get_Search_ParishDetails(string searchstr)
        {
            return objParishDAL.Get_Search_ParishInformation(searchstr);
        }
    }

}
```

```
using System;
using System.Collections.Generic;
using System.Configuration;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Text.RegularExpressions;
using System.Web;

namespace Diocese.Project_Code.SuperAdmin
{
    public class Parish_DAL
    {
        //SQL Connection string
        string ConnectionString = ConfigurationManager.ConnectionStrings["MyConnectio
        public int InsertParishInformation(Parish_BO objParishDetails_BO)
        {
            SqlConnection con = new SqlConnection(ConnectionString);
            con.Open();
            SqlCommand cmd = new SqlCommand("insert into Sup_ParishTable values(@Pari
            try
            {
                cmd.Parameters.AddWithValue("@ParishName", objParishDetails_BO.Parish
                cmd.Parameters.AddWithValue("@Place", objParishDetails_BO.ParishPlace
                cmd.Parameters.AddWithValue("@Username", objParishDetails_BO.UName);
                cmd.Parameters.AddWithValue("@Password", objParishDetails_BO.Passwd);
                int Result = cmd.ExecuteNonQuery();
                con.Close();
                return Result;
            }

            catch (Exception ex)
            {
                throw ex;
            }
            finally
            {
                cmd.Dispose();
                con.Close();
                con.Dispose();
            }
        }
```

```
public int DeleteParish(int id)
{
    SqlConnection con = new SqlConnection(ConnectionString);
    con.Open();
    SqlCommand cmd = new SqlCommand("delete from Sup_ParishTable where Parish
    cmd.Parameters.AddWithValue("@id", id);
    int i = cmd.ExecuteNonQuery();
    con.Close();
    return i;
}

public  DataTable Get_Search_ParishInformation(string searchstr)
{
    SqlConnection con = new SqlConnection(ConnectionString);
    con.Open();
    string query = "select * from Sup_ParishTable where Parish_Name like '%"+

    SqlCommand cmd = new SqlCommand(query, con);
    SqlDataAdapter sda = new SqlDataAdapter(cmd);
    DataTable dt = new DataTable();
    sda.Fill(dt);
    return dt;
}

public int UpdateParish(Parish_BO objParishDetails_BO, int id)
{
    SqlConnection con = new SqlConnection(ConnectionString);
    con.Open();
    SqlCommand cmd = new SqlCommand("update Sup_ParishTable set Parish_Name=@
    cmd.Parameters.AddWithValue("@ParishName", objParishDetails_BO.ParishName
    cmd.Parameters.AddWithValue("@Place", objParishDetails_BO.ParishPlace);
    cmd.Parameters.AddWithValue("@Username", objParishDetails_BO.UName);
    cmd.Parameters.AddWithValue("@Password", objParishDetails_BO.Passwd);
    cmd.Parameters.AddWithValue("@id",id);
    int Result = cmd.ExecuteNonQuery();
    con.Close();
    return Result;
}

public DataTable GetParishInformation()
{
    SqlConnection con = new SqlConnection(ConnectionString);
    con.Open();
```

```
            SqlCommand cmd = new SqlCommand("select * from Sup_ParishTable", con);
            SqlDataAdapter sda = new SqlDataAdapter(cmd);
            DataTable dt = new DataTable();
            sda.Fill(dt);
            return dt;

        }
    }
}
```

9.3 APPENDIX C

References

[1] Manjush Talmale, Abhishek Kinhekar, Akshay Saraf, Milind Bhajan Mobile Phone Cloning: History, Present Scenario and Precautionary Techniques, Nagpur, 2013International Journal of Application or Innovation in Engineering and Management (IJAIEM) 2013

[2] William Penberthy, Beginning ASP.NET for Visual Studio,2015.

[3] Rob Miles,C sharp Programming Yellow Book, Cheese,2016.

[4] Marino Posadas, Mastering C sharp and .Net Framework, Paperback,2016.

[5] http://www.google.co.in.

[6] http://www.wikipedia.org.