



ADORAPLAN

A Eucharistic Adoration Planner

ABSTRACT:

Enhance the management and engagement of religious communities through streamlined technological solutions

Presented by: Bria Tran
CSC 3150 Systems Design

Table of Contents

Executive Summary	2
1.0 Introduction and Overview	2
Problem Statement	2
Project Vision and Scope	2
Requirements Summary	3
Stakeholders and Their Interests	3
Expected Costs and Benefits	4
Constraints	4
Document Overview	5
2.0 System Initiation	6
Project Initiation Request (PIR)	6
3.0 Feasibility Assessment	9
Introduction	9
Feasibility Analysis	9
Conclusion	10
4.0 Requirements Definition	10
Introduction	10
Functional Requirements	11
Data Requirements	12
Non-functional Requirements	13
5.0 Requirements Model	14
6.0 System Evolution	56
7.0 Conclusions and Recommendations.	56
Appendices	57
Glossary	58
Bibliography	59

Executive Summary

AdoraPlan, the Adoration Scheduler App, is a transformative solution designed to simplify the coordination of Eucharistic Veneration time slots within the St. Pius X Parish Community in Mountlake Terrace and potentially across the Archdiocese of Seattle. Developed in response to the challenges parishioners face in signing up for adoration slots and ensuring continuous presence with the host, this app aims to enhance the spiritual experience of adorers while upholding the sanctity of the sacrament.

Key features of the app include easy sign-up and management of adoration slots, automated notifications for scheduled times and changes, and the ability to request substitutes or trade times when needed. By providing a user-friendly interface accessible via web and mobile devices, the app promotes increased participation in adoration by providing a user-friendly interface accessible via mobile and integration with existing parish protocols, state data protection compliance, and legal regulations, ensuring a reliable and trustworthy platform.

In Summary, AdoraPlan represents a significant advancement in parish administration while promoting reverence for the sacrament of Eucharistic Adoration and strengthening the spiritual fabric of the church community in the Mountlake Terrace Religious Community.

1.0 Introduction and Overview

Problem Statement

The client is St. Pius X Parish, but it could expand into the Archdiocese of Seattle. Our task is to develop an Adoration Scheduler Scheduling App to address the parish's challenge of coordinating Eucharistic veneration time slots. Parishioners register for adoration using a paper sign-up sheet posted on the adoration chapel's entrance. It is incumbent upon individuals to find substitutes or arrange trades when they cannot fulfill their commitment. However, contacting other parishioners and securing coverage has proven to be challenging. This is concerning because continuous presence with the host is imperative for Eucharistic veneration. The purpose of adoration is to spend time in the presence of the Lord for at least one hour, aiming to be in a prayerful mindset. In addition, leaving the chapel unattended has previously invited unwanted individuals to enter a sacred space. Developing this Adoration Scheduler aims to establish a more efficient planning system for the church.

Project Vision and Scope

Our project is to develop an adoration scheduler for St. Pius X Parish and the Archdiocese of Seattle, streamlining and coordinating Eucharistic veneration time slots. This solution addresses the current challenges of parishioners struggling to find substitutes or trades for their adoration commitments. It ensures continuous presence with the host to prevent sacrilege and keep reverence for the sacrament. By creating a more efficient scheduling system, our goal is to enhance the spiritual experience of adorers and safeguard the sanctity of Eucharistic veneration within the church community.

To achieve this, we plan to develop an Adoration scheduler that provides a streamlined platform for parishioners to sign up for adoration slots, manage their commitments, and quickly find substitutes or trades when necessary. The scope of our solution includes creating a user-friendly interface accessible via mobile devices, with features such as automated notifications and a centralized database for tracking adoration schedules. However, our solution will not extend to managing other aspects of parish administration or broader church activities. The environment for our project development will involve close collaboration with St. Pius X Parish and other

stakeholders to ensure the scheduler meets their specific needs and integrates seamlessly with pre-existing church protocols and systems.

Requirements Summary

The system must enable parishioners to create an account quickly, manage their selected time slot(s), receive notifications automatically, and communicate with fellow adorers. It must prioritize security, user-friendliness, and integration with existing parish protocols while providing comprehensive technical support and scalability.

Stakeholders and Their Interests

Primary Stakeholders & Interests:

1. Parish Administrators
 - 1.1. Interested in an efficient and organized system for managing adoration schedules to reduce administrative burdens.
 - 1.2. Interested in a program that ensures smooth operation of the parish activities.
2. Priest and Religious missionaries
 - 2.1. Interested in a tool that facilitates adherence to the practice of Eucharistic veneration, ensuring continuous presence with the host to maintain the sanctity of the sacrament.
3. Donors
 - 3.1. Interested in supporting initiatives that enhance the spiritual life of the parish community.
 - 3.2. Expect transparency and accountability in how their contributions are used to develop and maintain their app.
 - 3.3. Are interested in knowing how their donations are allocated for the project.
4. Technical support volunteers and staff
 - 4.1. Interested in designing a user-friendly application that meets all the system requirements
 - 4.2. Interested in minimizing technical issues
 - 4.3. Provides adequate support resources for troubleshooting and maintenance
5. Adorers
 - 5.1. Interested in a convenient platform for signing up for adoration slots, managing their commitments, and quickly finding substitutes when needs arise.

Expected Costs and Benefits

Expected Costs according to (*Intelligent Project Sizing*):

1. Initial cost: about \$97.7k
2. Estimated time to develop: about 8 to 33 weeks

Intangible benefits:

1. Enhanced spiritual connection.
 - 1.1. Increased participation in Eucharistic veneration
2. Streamlined coordination.
 - 2.1. facilitates straightforward management of adoration slots, reducing administrative burdens and enhancing overall efficiency
3. Strengthened sense of accountability.
 - 3.1. encourages adorers to take ownership of their roles and maintain the adoration schedule

Constraints

1. Budget
 - 1.1. Must adhere to a predetermined budget, which may limit the scope of development or resources distributed to certain aspects of the app
 - 1.2. Reliant on the amount of donations from donors

Mitigation strategy: identify essential features for app functionality and explore multiple avenues for securing funding
2. Regulatory
 - 2.1. Must comply with data protection laws like the General Data Protection Regulation (GDPR)
 - 2.2. Must comply with the Fair Labor Standards Act (FLSA)

Mitigation strategy: conduct regular compliance checks and implement privacy and data protection measures during the development process
3. Resource & Allocation
 - 3.1. Limited availability of resources, including funds and technological infrastructure, may need to improve the speed of development, testing, and deployment phases.

Mitigation strategy: adopt agile development methodologies and optimize workflow
4. Capacity
 - 4.1. Must initially handle at least 200 users.
 - 4.2. Must handle increases in user activity and data storage demands and ensure scalability and performance under varying usage scenarios.

Mitigation strategy Suggested Solution: design an app with scalability in mind and conduct performance testing under various load conditions to optimize system performance and resource utilization.

Document Overview

The System Proposal outlines the development of the Adoration Scheduler App and is organized into several sections. This proposal is a comprehensive guide for stakeholders involved in implementing AdoraPlan and offers a detailed roadmap for achieving the project objectives.

1. Introduction
 - 1.1. identifies the project, purpose, scope, and significance and outlines the primary functionalities of AdoraPlan
2. System Initiation
 - 2.1. Identifies the Project Initiation Request (PIR)
3. Feasibility Analysis
 - 3.1. Identifies the technical, resource, schedule, organizational, legal, and contractual to assess its viability and potential risks
4. Requirements Definition
 - 4.1. Identifies the Functional, Nonfunctional, and Data requirements
5. Requirements Model
 - 5.1. Identify the functional use cases and provide detailed descriptions of the system functions
6. System Evolution
 - 6.1. Identifies how the system is to expand and act in the future
7. Conclusion
 - 7.1. Summarizes the critical findings of the System Proposal and further recommendations to optimize the system
8. Glossary
9. References
10. Appendices

2.0 System Initiation

Project Initiation Request (PIR)

PIR-00000 [PIR Number to be assigned by the Project Office]

Project Initiation Request (PIR) – Level 2 v2.0

Project Name: AdoraPlan

Student Name: Bria Tran

0. General Project Information

Project Name:	<i>AdoraPlan</i>
Two Sentence Request Description:	<i>AdoraPlan, an Adoration Planner, enables Church parishioners to schedule times for Eucharistic veneration on a digital platform to streamline Adoration reservation times. It allows for booking time slots, finding a substitution, providing contact information, and, most importantly, a Q&A section for communication between church members.</i>
Requested Launch Date(s):	<i>9/1/2026</i>
Department(s) Affected By Project:	<i>IT, HR, Customer Service, Management, and possibly the Finance department</i>
Project's Customers:	<i>Catholic or Christian Churches who practice adoration of the host</i>
Date Request Submitted:	<i>5/21/2024</i>

1. Project Sponsor and Manager

Project Sponsor

Name:	Andy Cameron
Title:	Professor
Department:	Computer Science - SPU
Email:	acameron@spu.edu

Business Project Manager & Requestor

Name:	<i>Bria Tran</i>
Title:	<i>Student</i>
Department:	<i>Information Systems - SPU</i>
Email:	<i>tranb10@spu.edu</i>

2. Business Problem or Opportunity: The motivation for this request

Within my church community, St. Pius X, the parish offers 24-hour Eucharistic adoration, a practice cherished deeply by Catholic Christians. Currently, parishioners sign up for their weekly adoration time slots by physically adding their names to a paper posted on the door of the chapel room. However, this method has proved to cause errors and confusion for the parishioners. Life's demands often intervene, causing parishioners to miss their scheduled adoration sessions, which poses a significant challenge for our parish. Finding a substitute becomes paramount because of the importance of having someone constantly present to accompany the host during adoration. Unfortunately, locating replacements can be cumbersome, as parishioners may not always know each other well enough to reach out. In addition, there is also a security issue. Multiple cases of outsiders intruding onto chapel grounds, claiming to be adored. There is no lock or personal identification in front of the door. This presents an extraordinary opportunity for technology to provide a solution or, at the very least, simplify the problems. While there are no immediate deadlines for addressing the issue, the sooner we implement a solution, the smoother Eucharistic adoration will become for our church and its members.

3. Justification, Impact, and Importance

Assumptions

- Secure database to house parishioner contact information
- Accessible through Wi-Fi or the use of data
- Comply with data protection regulations and standards
- Allows users to edit, delete, and share their information on the platform
- Be able to accommodate a growing amount of personal information over time
- Has the potential to integrate with other tools in the future

Competitive Landscape / Context

- Subsplash – the leading developer of mobile apps and digital solutions for churches and other ministry-based organizations
- Tithe.ly – another popular platform that provides digital solutions for churches like event registration and other church management tools
- Church Community Builder – another popular software platform designed to assist churches with volunteer management, communication tools, and other administrative tasks

Tangible Return, Opportunity, or Value One Time**On-Going**

▪ <i>Estimate one-time cost for setting up the app (unsure of ongoing fees) (Intelligent Project Sizing)</i>	\$97.7k	\$0
▪ Parishes will schedule their time slots for adoration, find covers, and communicate with fellow worshippers		
▪ Adorees will have a personal barcode, similar to a gym barcode, to scan themselves at the door lock and safely worship at the chapel		

Intangible Benefits Impact or Value

▪ <i>Improved communication between churchgoers</i>	\$0
▪ Enhanced accountability for adores	\$0

4. Product Requirements**4.1. Must Haves**

4.1.1. <i>Adoration time slot reservation capabilities</i>
4.1.2. Open Q&A section for communication with all adores
4.1.3. Account management
4.1.4. Adoration time slot modification capabilities
4.1.5. View personal and event or time slots

4.2. Could Haves (Nice to Haves)

4.2.1. <i>Personal Identification Barcode</i>
4.2.2. Other Volunteering Management Capabilities

4.3. Won't Haves (Don't Do's, aka Out of Scope)

4.3.1. <i>Event Planning</i>
4.3.2. Church Financial Management

3.0 Feasibility Assessment

Introduction

This section will summarize the feasibility analysis, outlining the factors considered and the criteria used to evaluate the viability of AdoraPlan's App Project. The feasibility analysis assesses the project's technical, resource, schedule, organizational, and contractual feasibility to determine its potential for success. (WO4–Feasibility and Risk)

The rating scale will be a rating scale with five levels:

1. Not feasible: shows that the system will experience significant setbacks to the project's success and render it unfeasible
2. Barely feasible: Suggests that many challenges must be addressed but may be manageable with measures
3. Moderately feasible: shows moderate challenges that require attention but are manageable within the project's scope
4. Feasible: Signifies minimal obstacles that may affect the project's success but can be effectively mitigated
5. Very feasible: Reflects minimal to no obstacles and shows strong potential for project success

Feasibility Analysis

Technical Feasibility

AdoraPlan's system is *moderately feasible*, although there is some risk.

1. There may be some basic navigational training needed to train the administrative staff and religious ministers, but it will be straightforward since most people know how to operate apps like Gmail and Facebook
2. The development group may need to be more familiar with the target technology because it is developing an app on iOS and Android devices. Using knowledge of software engineering classes and other algorithmic software, developers should be able to accomplish the task given some time to study the defined programming languages. The main compounds and structure are unlikely to change and should not cause too many complications down the road

Resource Feasibility

AdoraPlan's resource *feasibility is moderate*, given the risk, but they are manageable.

1. We have enough people to develop the app since it is a relatively simple and small scheduling app
2. We have all the hardware because the church owns both Mac and Windows operating systems available for use to develop the app on
3. If we cannot access those devices, we have the means to purchase the required operating systems to ensure the developers have the hardware and software needed

Schedule Feasibility

AdoraPlan's schedule feasibility is *very feasible*.

1. There are no time constraints or deadlines by which the app must be completed. However, when the app begins production, a year is recommended. However, more time may be allocated for the project since there are no stringent deadlines to be met. It is workable for this system to be created within a year, since it is small and does not require too many components.

Organizational Feasibility

From an organizational perspective, this project is *feasible*.

1. Yes, there are other competitors with similar Church help systems like Subsplash, Tithe.ly, and Church Community Builder; however, none but precisely for this issue. Those apps mainly focus on managing the financial aspect of Christian worship and funds
2. Building is better than buying because there are no applications like this on the market for Christian traditions that practice Eucharistic adoration
3. There are no real immediate consequences of not developing it; however, not developing an app for this issue would leave the church in inefficient processes and create a stressful environment for the administration
4. Developing AdoraPlan will allow users to prioritize the sanity of the Eucharist instead of constantly worrying about who is there to accompany the Lord
5. This is not an app that is going to be used on a subscription or one-time fee payment...it is a nonprofit app that is going to be developed and maintained by the donations set aside by the Catholic Archdiocese of Seattle

Legal and Contractual Feasibility

Both legally and contractually, it is *very feasible*.

1. Must comply with the data protection rules of the GDPR to protect the privacy and security of user data.

Conclusion

While the feasibility analysis highlights the project's strong potential, there are some reservations and risks that should be managed proactively. One concern is the potential for data privacy and security breaches, given the sensitive nature of personal information stored in the app. Stringent security measures and compliance with data protection regulations such as FLSA and GDPR will be essential to mitigate that risk. (Optimizing Financial Industry Awards Organizing with CRM Solution)

Suggestions for managing risks include:

1. Regulatory review and update of security protocols to address potential threats.
2. Implement a robust backup plan to safeguard against data loss or system downtime.
3. Provide comprehensive user training and support resources to promote effective app utilization.
4. Establish clear communication channels for stakeholders to report feedback and concerns.

4.0 Requirements Definition

Introduction

This section will define functional and non-functional requirements in simple terms and outline the detailed requirements for AdoraPlan. This section serves as a guide to provide stakeholders with a better understanding of the project's objectives to ensure clarity throughout the project lifecycle and the Minimum Viable Product (MVP) of this system. (Requirement, What We Are Seeking), (Feasibility Studies & Feasibility Study about Business)

Functional Requirements

1. Download App (Refer to Use Case 6)
 - 1.1. User must be able to download AdoraPlan on both iOS and Android mobile devices
2. User Account Creation
 - 2.1. After downloading the app, the system must prompt users to create a login account. (See Use Case 1)
3. User Login
 - 3.1. System must save login information of existing users (Refer to Use Cases 8 and 20)
 - 3.2. System must allow users to recover login information (Refer to Use Cases 13 and 14)
4. User Profile and Account Management
 - 4.1. User must be able to access the app function from a main menu (See Use Case 15)
 - 4.2. Saving user data is necessary, including name, photo (optional), email address, and phone number(s) (See Use Case 10).
 - 4.3. Uploaded photos must be in JPG or PNG format and cropped to a circular profile photo of 180 x 180 pixels, no larger than 5 megabytes (MB) (See Use Case 10)
 - 4.4. Display the user data on the user's profile, with the photo as the header and the name, email, and number(s) vertically displayed below (See Use Case 2).
5. Share contact information with other users
 - 5.1. Users must share their personal information with other clients within AdoraPlan (See Use Case 3))
 - 5.2. The system must allow the user to selectively share specific details, such as their name and any additional information they disclose
6. Time-slot Reservation (Refer to Use Cases 4, 16, 17, 18, and 19)
 - 6.1. Users can reserve hourly time slots on a 24/7 calendar for recurring adoration commitments.
 - 6.2. Options to cancel, drop, or trade time slots and events must be available.
7. Q&A General Chat (Refer to Use Case 5)
 - 7.1. The system must include an open Question and Answer chat (Q&A) that allows all members to communicate
8. Event Creation and Sign-Up
 - 8.1. Users must be able to create events with the following details: (See Use Case 7)
 - 8.1.1. name, description, date/time, location, host, and volunteer count.
 - 8.2. When new events are created, we must send notifications.
9. View Calendar
 - 9.1. User must be able to view events and time slots on a general calendar (See Use case 11)
 - 9.2. Users must be able to view the events and time slots they are taking part in on a personal calendar (See Use Case 12)
10. Generate Report
 - 10.1. The system must log all user interactions with the app and periodically generate a report to send to the Systems Administrator (See Use Case 9)
11. Barcode Scanner Integration (See System Evolution Section)
 - 11.1. Each user should have a randomly generated barcode for door lock access.
 - 11.2. A circular icon should display the barcode and expand when pressed for scanning.

Data Requirements

1. Download App
 - 1.1. Platform-specific files for iOS and Android
 - 1.2. Metadata for App Store listing
2. User Account Creation
 - 2.1. User information: name, email address, phone number(s), username, password, home address
 - 2.2. Data Storage: Secure database for profile data and user credentials
3. User Login
 - 3.1. User credentials: username and password
 - 3.2. Password recovery mechanisms: password and username reset links
 - 3.3. Logs of login attempts and sessions
4. User Profile and Account Management
 - 4.1. User profile photo (JPG/ONG, 180x180 pixels, 5 MB max)
 - 4.2. User settings
 - 4.3. Storage and retrieval mechanisms for profile pictures
 - 4.4. User profile display
5. Share contact information with Other Users
 - 5.1. Permissions and settings for data sharing
6. Time-Slot Reservation
 - 6.1. Calendar data structure to store time slots (date, time, status: available/reserved)
 - 6.2. User reservations: user ID, time slot ID, reservation status
 - 6.3. Functions to manage cancelations, drops, and trades of time slots
 - 6.4. Notification system for reservation updates
7. Q&A General Chat
 - 7.1. Chat message: user ID, message content, timestamp
 - 7.2. Group Chat management: message history
 - 7.3. Data storage for chat logs
8. Event Creation and Sign up
 - 8.1. Events details: name, description, date/time, location, host, volunteer count
 - 8.2. User event participation, user ID, participation status
 - 8.3. Notification system for new event creation
 - 8.4. Calender integration for events
9. View Calendar
 - 9.1. General calendar data: available events and time slots
 - 9.2. Personal calendar data: user-specific events and reservations
 - 9.3. Mechanism to filter and display calendar entries
10. Generate Report
 - 10.1. Logs of user interactions: user ID, interaction type, timestamp
 - 10.2. Report generation algorithms: summarize interactions, format report
 - 10.3. Scheduled task to generate and send reports periodically
 - 10.4. Data store for historical reports
11. Barcode Scanner Integration
 - 11.1. User-specific barcodes: unique and randomly generated barcodes
 - 11.2. Function to display and expand barcode within the app
 - 11.3. Integration with door locking systems to validate barcodes
 - 11.4. Logs of barcode generation and usage
12. Data Security and Privacy

- 12.1. Encryption to secure user data storage
- 12.2. Compliance with Data Protection Regulation (GDPR)
- 12.3. Regular data backups and recovery plans

Non-functional Requirements

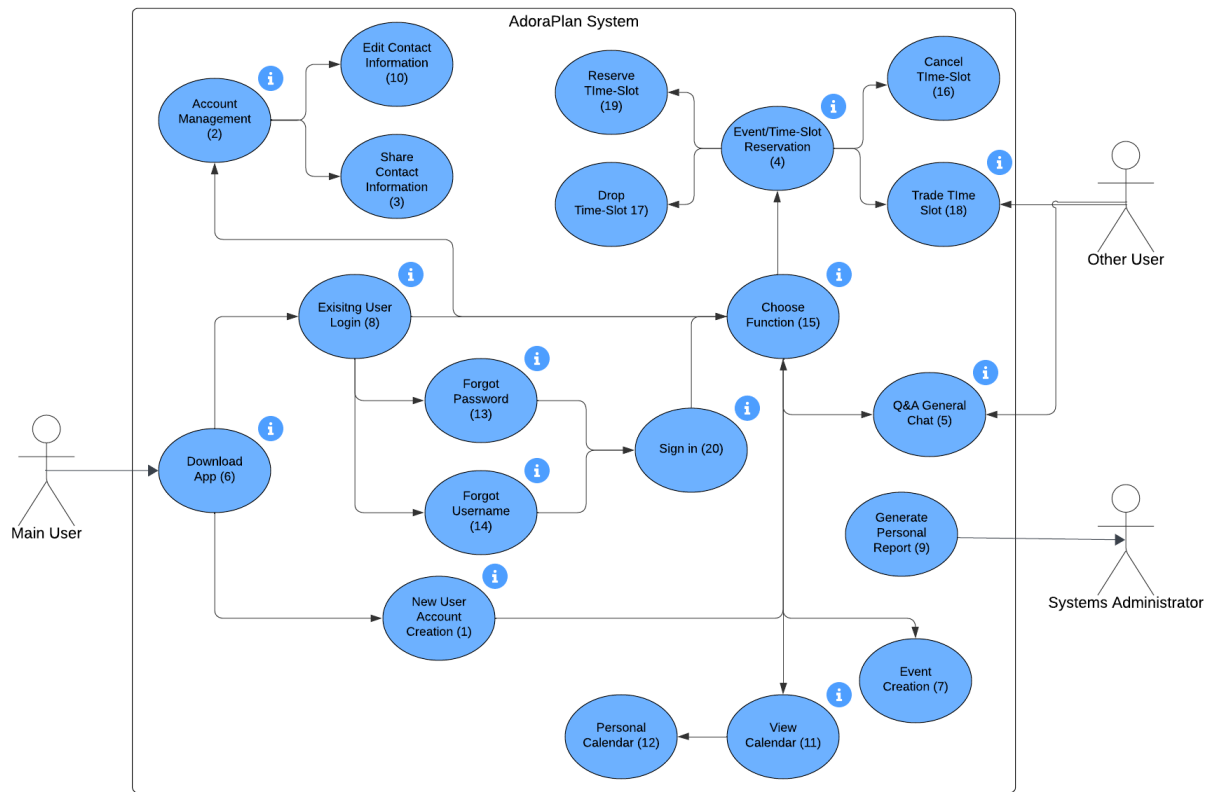
- 1. Platform Compatibility
 - 1.1. The app must run on both iOS and Android devices and be developed using Xcode and Android Studio
- 2. Reliability
 - 2.1. The app should achieve at least 99.9% uptime (three nines reliability)
- 3. Security
 - 3.1. Passwords must be blurred and only show one character at a time.
 - 3.2. Only personnel can access personal data.
 - 3.3. Data protection regulations, such as GDPR and FLSA compliance, must be maintained.
- 4. Scalability
 - 4.1. The system must handle a growing database of users and accommodate at least 200 simultaneous users.
- 5. Response Time
 - 5.1. Interface response time should be under 2 seconds.
- 6. Accessibility
 - 6.1. The user interface will display in larger-than-normal font sizes
 - 6.2. Screen magnification should be available
- 7. Data Backup
 - 7.1. Implement backup storage for user personal data
- 8. Distribution
 - 8.1. Distribute the app on the Apple App Store and Google Play Store

5.0 Requirements Model

Introduction

The AdoraPlan app streamlines operations and enhances user engagement on our platform. This overview introduces the app's core functionalities and the supporting documentation, ensuring that technical and non-technical users understand its processes. The documentation includes a Use Case Diagram to visually explain the app's features. Each oval represents a Use Case AdoraPlan must be able to handle, demonstrating how the functions interact. Actors are at the end of the diagram, showing how humans interact with the system. Use-case descriptions, which further explain the process for each use-case function, follow the Use-Case Diagram.

Use-Case Diagram



Use-Case Descriptions

Use Case Name: New User Account Creation		ID: UC-1	Importance: Must Have
Primary Actor: The User		Use Case Type: Overview, Essential	
Supporting Actors: System Administrators			
Stakeholders and Interests: Religious Leaders – interested in parishioners going to adoration services Parishioners – interested in a simple account creation process Church Administrators/Staff – interested in parishioners going to adoration services App Developers – interested in ensuring that their databases can hold user accounts IT Support – interested in a simple account creation process			
Brief Description: The user selects the New User login portal, and the portal directs them to a page where they can create a username and password for their account. The systems automatically create a profile for them, sign them in, and direct them to the application home page. They will create a login profile with their name, phone number, email, and other relevant details.			
Trigger: During the Download App Use Case, the user downloads AdoraPlan and encounters a prompt to sign in.			
Type (mark one): <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal (rare)			
Relationships: Association: User Include: Use Case 9 Extend: N/A Generalization: N/A			
The Normal Flow of Events: <ol style="list-style-type: none"> 1. User selects the "New User Sign In" option 2. Users receive directions to a form where they can create their account 3. User enters required new account credentials 4. User enters optional Photo 5. System creates a new user account 			
Sub-flows: User Selects "New User Sign In" <ol style="list-style-type: none"> 1.1 Display: "New User Sign In" 1.2 User clicks "New User Sign In" User is redirected to the Create New Account Form <ol style="list-style-type: none"> 2.1 Display New Account Form 2.2 Prompts user for "First Name" and "Last Name" 2.3 Prompts the user for a phone number in "XXX-XXX-XXXX" format 2.4 Prompts User for email like mark.peters@example.com 2.5 Prompts user for username 2.6 Prompts user for a password 			

User enters the required credentials

- 3.1 User fills out the entire form
- 3.2 System checks fields are not empty
- 3.3 Systems validate the format of each field

Users can Optionally Add a Profile Picture

- 4.1 Display option to upload a profile picture
- 4.2 Display text below "Can change the photo at any time."
- 4.3 User uploads photo or skips
- 4.4 The system checks the file size and format if a photo is uploaded to the correct size.

A New User Account is created, and the User is Automatically Signed in

- 5.1 User review information and submits form
- 5.2 System creates a new account in the database
- 5.3 The app automatically signs the user in
- 5.4 Display welcome screen

Alternate/Exceptional Flows:

Invalid Data Entry

1. User enters required credentials
2. Systems checks fields are not empty
3. If any required field is empty
 - 3.1 Display an error message showing the field(s) that need to be filled
 - 3.2 Return the user to the form to complete the missing fields
4. The system validates the format of each field
5. If any field is in an incorrect format
 - 5.1 Display an error message showing the specific format issues
 - 5.2 Return the user to the form to correct the errors
6. The user corrects the errors and resubmits the form
7. Continue normal flow

Username Already Exists

1. User enters a username that already exists in the system database
 - 1.1 Display error message indicating the username is already in use
 - 1.2 Prompt the user to choose a different username
2. The user enters a new username
3. Continue normal flow

Optional Profile Photo Upload Fails

1. User uploads profile picture
2. System checks file size and format
3. If the file size is bigger than the limit or the format is incorrect
 - 3.1 displays an error message indicating the issue with the upload
 - 3.2 prompt user for valid picture files
4. User uploads a valid photo file or ships the upload
5. Continue normal flow

Special Requirements:

Security Requirements

1. The system must ensure that passwords are stored securely using encryption algorithms
2. All user data must be over a secure (SSL/TLS) connection

Usability Requirements

1. The account creation form must be accessible and usable on both iOS and Android mobile devices

Performance Requirements

1. The system must process account creation requests within 2 seconds to ensure a smooth user experience

2. Profile picture uploads only support JPG and PNG files up to 5 MB in size and cropped to 180 x 180 pixels

Backup Requirements

1. Ensure user data is backed up regularly to prevent information loss
2. Implement a recovery mechanism to restore user data in case of system failures

Scalability Requirements

1. The system must handle simultaneous account creation requests without decreased performance
2. Ensure that the system database architecture can scale to accommodate the growing number of user accounts

Compliance Requirements

1. The system must comply with relevant data protection regulations, like the GDPR, regarding the handling of personal data and storage
2. Allow users to review and manage their personal information as per legal requirements

Error Handling Requirements

1. Logs errors and notify system administrators if there are critical issues

To do/Issues:

N/A

Use Case Name: Account Management		ID: UC 2	Importance: Must Have
Primary Actor: The User		Use Case Type: Overview, Essential	
Supporting Actors: Systems Administrator			
Stakeholders and Interests: Religious Leaders – interested in parishioners staying engaged with church activities Parishioners – interested in a simple account management process Church Administrators/Staff – interested in parishioners staying involved with church activities App Developers – interested in error-free account management feature IT Support – interested in easy maintenance and updates to the account management system			
Brief Description: The user chooses to manage the information in their account, which is directed toward their personal profile.			
Trigger: Choose Function Use case – The user decides what they want to do in the application			
Type (mark one): <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal (rare)			
Relationships: Association: The user Include: Use Case 19 Extend: N/A Generalization: N/A			
The Normal Flow of Events: <ol style="list-style-type: none"> 1. User Navigates to Account Management 2. App Displays available options 2. Display Edit Contact Information 3. Display Share Contact Information 4. Review and Save Changes 			
Sub-flows: User Navigates to Account Management <ol style="list-style-type: none"> 1.1 User selects the profile icon of "Account Setting" from the menu Edit Contact Information <ol style="list-style-type: none"> 3.1 The user selects "Edit Contact Information" 3.2 The app displays the current contact information in a form Edit Shared Contact Information <ol style="list-style-type: none"> 4.1 The user selects "Share Contact Information" 4.2 The app displays the current shared information and a form to change the contacts Review Changes <ol style="list-style-type: none"> 5.1 The user is shown the updated information 5.2 The user confirms that the information is correct 			
Alternate/Exceptional Flows: System Errors During Saving Changes			

1. User confirms that the update information is correct
2. System attempts to solve the changes
3. If there is a database or server error
 - 3.1 Display an error message indicating the issue
 - 3.2 Log the error for system administrators
 - 3.3 Prompt the user to try again late or contact support
4. User decides to try again or contact support
5. Continue with the normal flow if retrying "Review Changes"

Special Requirements:

Security Requirements

1. The system must ensure that passwords are stored securely using encryption algorithms
2. All user data must be over a secure (SSL/TLS) connection

Usability Requirements

1. The account creation form must be accessible and usable on both iOS and Android mobile devices
2. Provide instructions and real-time feedback to guide users through the account management process
3. Ensure the interface has clear labeling of actions and options

Performance Requirements

1. The system must process account creation requests within 2 seconds to ensure a smooth user experience
2. Profile picture uploads only support JPG and PNG files up to 500 MB in size and cropped to 180 x 180 pixels

Backup Requirements

1. Ensure user data is backed up regularly to prevent information loss
2. Implement a recovery mechanism to restore user data in case of system failures

Scalability Requirements

1. The system must handle simultaneous account creation requests without decreased performance
2. Ensure that the system database architecture can scale to accommodate the growing number of user accounts

Compliance Requirements

1. The system must comply with relevant data protection regulations, like the GDPR, regarding the handling of personal data and storage
2. Allow users to review and manage their personal information as per legal requirements

Error Handling Requirements

1. Logs errors and notify system administrators if there are critical issues

To do/Issues:

N/A

Use Case Name: Share Contact Information		ID: UC 3	Importance: Must Have
Primary Actor: The User		Use Case Type: Overview, Essential	
Supporting Actors: Systems Administrator			
Stakeholders and Interests: Religious Leaders – interested in parishioners staying engaged with the church community Parishioners – interested in a simple account management process Church Administrators/Staff – interested in parishioners staying engaged with the church community App Developers – interested in error-free account management feature IT Support – interested in easy maintenance and updates to the account management system			
Brief Description: The user will choose what information to make public for the other users on the application to view, such as their phone number or email address.			
Trigger: Account Management Use case – User chooses what they want to do in their account profile			
Type (mark one): <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal (rare)			
Relationships: Association: The User Include: Use case 2 Extend: N/A Generalization: N/A			
The Normal Flow of Events: <ol style="list-style-type: none"> 1. Select Share Contact Information 2. Display and Select Information to Share 3. Review and Save Changes 4. Display Information on Public Profile 5. Return to Account Management 			
Sub-flows: Display and Select Information to Share <ol style="list-style-type: none"> 2.1 The app displays a sharing interface with options for selecting contact information to share 2.2 The user selects/deselects fields like phone number, email, and other relevant contact information Review and Save Changes <ol style="list-style-type: none"> 3.1 The user reviews the interface with the selected fields to share 3.2 The user confirms that the information has been shared correctly Display Information on <ol style="list-style-type: none"> 4.1 System updates the user's public profile with the information to be shared 			
Alternate/Exceptional Flows: System Errors During Saving Changes <ol style="list-style-type: none"> 1. User reviews the interface with the selected fields to share 2. User confirms that the information is correct 3. System attempts to update the user's public profile 			

4. If there is a database or server error
 - 4.1 displays an error message indicating the issue
 - 4.2 log the error for system administrators
 - 4.3 prompt the user to try again later or contact support
5. user decides to try again or contact support
6. continue normal flow from "Review and Save Changes."

Special Requirements:

Security Requirements

1. All user data must be over a secure (SSL/TLS) connection

Usability Requirements

1. The contact sharing form must be accessible and usable on both iOS and Android mobile devices

Performance Requirements

1. The system must process requests to share contact information within 2 seconds to ensure a smooth user experience

Backup Requirements

1. Ensure user data is backed up regularly to prevent information loss
2. Implement a recovery mechanism to restore user data in case of system failures

Scalability Requirements

1. The system must handle simultaneous account creation requests without decreased performance
2. Ensure that the system database architecture can scale to accommodate the growing number of user accounts

Compliance Requirements

2. The system must comply with relevant data protection regulations like the GDPR regarding the handling of personal data and storage
3. Allow users to review and manage their personal information as per legal requirements

Error Handling Requirements

2. Logs errors and notify system administrators if there are critical issues

To do/Issues:

N/A

Use Case Name: Time-Slot Reservation		ID: UC 4	Importance: Must Have
Primary Actor: The User		Use Case Type: Overview, Essential	
Supporting Actors: Systems Administrator			
Stakeholders and Interests: Religious Leaders – interested in parishioners staying engaged with the church activities Parishioners – interested in a simple time-slot reservation process Church Administrators/Staff – interested in parishioners staying engaged with the church activities App Developers – interested in error-free reservation capabilities IT Support – interested in easy maintenance and updates to the time-slot reservation database			
Brief Description: The user selects the time they want and decides whether to reserve, drop, trade, or cancel the time reservation.			
Trigger: Choose Function Use Case – The user decides what they want to do in the application			
Type (mark one): <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal (rare)			
Relationships: Association: The User Include: Use Case 19 Extend: N/A Generalization: N/A			
The Normal Flow of Events: <ol style="list-style-type: none"> 1. The user navigates to Time-Slot Reservation 2. The user browses and selects a date 3. The user reviews and selects a time-slot 4. Choose a Time-Slot Reservation Action 5. Follow the selected action 			
Sub-flows: Browse and Select a Date <ol style="list-style-type: none"> 2.1 The user browses through the calendar and selects a desired date Review and Select Time-Slot <ol style="list-style-type: none"> 3.1 The user reviews the available time slots and selects the preferred slot Choose a Time-Slot Reservation Action <ol style="list-style-type: none"> 4.1 The app displays four options to modify the time slot 4.2 The user chooses to either reserve, drop, cancel, or trade the slot Follow the Selected Action <ol style="list-style-type: none"> 5.1 The user follows the actions required from their chosen action 5.2 After completing the Time-Slot Reservation Action, the user is brought back to their calendar 5.3 The user can review the reservation details (date, time) 			
Alternate/Exceptional Flows: N/A			
Special Requirements:			

Security Requirements

1. All user data must be over a secure (SSL/TLS) connection

Usability Requirements

1. The account creation form must be accessible and usable on both iOS and Android mobile devices

Performance Requirements

1. The system must process time-slot reservation requests within 2 seconds to ensure a smooth user experience
2. The system must be able to handle multiple simultaneous reservation requests without slowed performance

Backup Requirements

1. Ensure user data is backed up regularly to prevent information loss
2. Implement a recovery mechanism to restore user data in case of system failures

Scalability Requirements

1. The system must handle simultaneous time-slot reservation requests without decreased performance
2. Ensure that the system database architecture can scale to accommodate the growing number of user accounts

Compliance Requirements

1. The system must comply with relevant data protection regulations like the GDPR regarding the handling of personal data and storage
2. Allow users to review and manage their personal information as per legal requirements

Error Handling Requirements

1. Logs errors and notify system administrators if there are critical issues

To do/Issues:

N/A

Use Case Name: Q&A General Chat		ID: UC-5	Importance: Must Have
Primary Actor: Main User		Use Case Type: Overview, Essential	
Supporting Actors: Another user, the Systems Administrator			
Stakeholders and Interests: Religious Leaders – interested in parishioners staying engaged with the church community Parishioners – interested in simple communication channels with fellow church-goers Church Administrators/Staff – interested in parishioners staying engaged with the church community App Developers – interested in creating a powerful and efficient Q&A channel IT Support – interested in easy maintenance of communication channel			
Brief Description: The User enters the General Chat, composed of all the users on the application, and can message the chat to ask questions or communicate with other users.			
Trigger: Choose Function Use Case – The user decides what they want to do in the application			
Type (mark one): <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal (rare)			
Relationships: Association: The User Include: Use Case 19 Extend: N/A Generalization: N/A			
The Normal Flow of Events: <ol style="list-style-type: none"> 1. Navigate to Q&A Channel 2. Type Comments Into Chat 3. Review Previous Chats between other Parishioners 4. Get Notifications for New Chats 			
Sub-flows: Navigate to the Q&A Channel <ol style="list-style-type: none"> 1.1 The user selected the "Q&A Channel from the main menu Type Questions/Statements into the chat <ol style="list-style-type: none"> 2.1 The user clicks the text cursor, enters their questions/comments into the chat, and submits it Review Previous Chats between other Parishioners <ol style="list-style-type: none"> 3.1 The user can scroll backward to view previous conversations between other users and reply Get Notification for New Chats <ol style="list-style-type: none"> 4.1 The user receives a notification about new answers, comments, or questions 			
Alternate/Exceptional Flows: Chat moderation <ol style="list-style-type: none"> 1. The user types comments into the chat 2. User decides to edit or delete the comment 3. Continue normal flow if the comment is edited or deleted 			

System error During Chat interaction

1. User types comments into the chat
2. System attempts to submit the comment
3. If there is a database or server error during the submission
 - 3.1 displays an error message indicating the issue
 - 3.2 log the error for system administrators
 - 3.3 prompt the user to try again later or contact support
4. User decides to try again or contact support
5. continue normal flow if retrying to submit the comment

Notification Failure

1. user receives notifications about new answers, comments, or questions
2. if there is a failure in delivering the notification
 - 2.1 log the notification failure for system administrators
 - 2.2 display a message to the user indicating the issue with notifications
 - 2.3 provides an option for the user to manually check for new chat activities
3. user manually checks for new city
4. continue normal flow from reviewing new chats

Special Requirements:**Security Requirements**

1. allow users to report inappropriate content or behavior for review by system administrators

Usability Requirements

1. implement features such as emojis, gifs, and other multimedia support to enhance user engagement in the chat
2. provide options for a user to customize their chat setting (e.g., notification preferences)

Performance Requirements

1. The system must process chat interactions within 2 seconds to ensure a responsive user experience

Backup Requirements

1. Ensure user data is backed up regularly to prevent information loss
2. Implement a recovery mechanism to restore user data in case of system failures

Scalability Requirements

1. The system must handle simultaneous chat interactions from multiple users without decreased performance
2. Ensure that chat infrastructure can scale to accommodate the growing number of user accounts

Compliance Requirements

1. The system must comply with relevant data protection regulations like the GDPR regarding the handling of personal data and storage

Error Handling Requirements

1. Logs errors and notify system administrators if there are critical issues

To do/Issues:

N/A

Use Case Name: Download App		ID: UC-6	Importance: Must Have
Primary Actor: The User		Use Case Type: Overview, Essential	
Supporting Actors: N/A			
Stakeholders and Interests: Religious Leaders – interested in parishioners engaging with church activities Parishioners – interested in streamlined adoration scheduling and parishioner communication Church Administrators/Staff – interested in parishioners involved with church activities App Developers – interested in the accessibility of an efficient adoration scheduling app IT Support – interested in simple app set-up			
Brief Description: The user downloads the application on their mobile device and is prompted to the login page.			
Trigger: N/A			
Type (mark one): <input type="checkbox"/> External <input type="checkbox"/> Temporal (rare)			
Relationships: Association: The User Include: N/A Extend: N/A Generalization: N/A			
The Normal Flow of Events: 1. Go to App store 2. Search for AdoraPlan 3. Download the app			
Sub-flows: Go to the App Store 1.1 Go to App Store for iOS devices 1.2 Go to Google Play Store for Android Devices			
Alternate/Exceptional Flows: Insufficient Storage Space 1. The user attempts to download the app but has insufficient storage space on the device 1.1 System displays an error message indicating insufficient storage 1.2 User is provided with instructions to free up space Internet Connectivity Issue 1. User attempts to download the app without an internet connection 1.1 system displays an error message indicating the lack of internet connectivity 1.2 user is prompted to check their internet connection and retry			
Special Requirements: Security Requirements 1. All user data must be over a secure (SSL/TLS) connection 2. Implement app security measures to prevent unauthorized access			

Usability Requirements

1. Ensure the app is user-friendly with straightforward navigation and instructions for users

Performance Requirements

1. The app must load within 5 seconds of opening
2. Optimize app performance and reduce loading times for users

Backup Requirements

1. Implement automatic data backup features to prevent data loss in case of app uninstallation or device failure

Scalability Requirements

1. Ensure that the app infrastructure can scale to accommodate the growing number of user accounts

Compliance Requirements

1. The system must comply with relevant data protection regulations like the GDPR regarding the handling of personal data and storage
2. Comply with app store guidelines or regulations to avoid removal from app stores and maintain app availability

Error Handling Requirements

1. Log errors and notify system administrators if there are critical issues

To do/Issues:

N/A

Use Case Name: Event Creation		ID: UC-7	Importance: Must Have
Primary Actor: The User		Use Case Type: Overview, Essential	
Supporting Actors: Systems Administrator			
Stakeholders and Interests: Religious Leaders – interested in parishioners staying engaged with the church community Parishioners – interested in simple, customizable event creation Church Administrators/Staff – interested in parishioners staying engaged with the church community App Developers – interested in user engagement and retention with event participation IT Support – interested in the reliability of event creation without causing crashes			
Brief Description: The User selects event creation for the Choose Function Use Case, is prompted for the date and other event details, and posts the event on the General Calendar.			
Trigger: Choose Function Use Case – The user decides what they want to do in the application			
Type (mark one): <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal (rare)			
Relationships: Association: The User Include: Use Case 19 Extend: N/A Generalization: N/A			
The Normal Flow of Events: <ol style="list-style-type: none"> 1. Navigate to the Event Creation tab from the main menu 2. Create Event 3. Publish Event on the Calendar 4. Notify Other Parishioners on the Q&A Channel 5. Wait For People to Sign-up 			
Sub-flows: Create Event <ol style="list-style-type: none"> 2.1 The user fills in the event details (title, description, date, time, location, and other relevant information) Publish Even on Calendar <ol style="list-style-type: none"> 3.1 The user completes the event creation form and clicks the "Publish" button 3.2 The user receives a confirmation message that the event has been published Notify Other Parishioners <ol style="list-style-type: none"> 4.1 The app automatically generates a message with the event details in the Q&A channel Wait for People to Sign-Up <ol style="list-style-type: none"> 5.1 The app tracks sign-ups and displays the number of participants 5.2 The event creator can manage and view the list of attendees 			
Alternate/Exceptional Flows: Incomplete Event Details <ol style="list-style-type: none"> 1. User attempts to publish the event without filling in all required details <ol style="list-style-type: none"> 1.1 system displays an error message indicating the missing information 			

- 1.2 user is prompted to complete all required fields before publishing

Duplicate Event

1. user attempts to create an event with details identical to an existing event
 - 1.1 system checks for duplicates and displays a warning message if a similar event is found
 - 1.2 user is given the option to review the existing event or proceed with making a new event

Event Publishing Failure

1. user completes the vent-creating form but encounters a failure when publishing
 - 1.1 system displays an error message indicating the publishing failure
 - 1.2 user is prompted to retry or contact support if the issue persists

Special Requirements:

Security Requirements

1. All user data must be over a secure (SSL/TLS) connection

Usability Requirements

1. The event creation form must be accessible and usable on both iOS and Android mobile devices

Performance Requirements

1. The system must process event creation requests within 2 seconds to ensure a smooth user experience

Backup Requirements

1. Ensure user data is backed up regularly to prevent information loss
2. Implement a recovery mechanism to restore user data in case of system failures

Scalability Requirements

1. The system must handle simultaneous event creation requests without decreased performance

Compliance Requirements

1. The system must comply with relevant data protection regulations like the GDPR regarding the handling of personal data and storage

Error Handling Requirements

1. Logs errors and notify system administrators if there are critical issues

To do/Issues:

N/A

Use Case Name: Existing User		ID: UC-8	Importance: Must Have
Primary Actor: The user		Use Case Type: Overview, Essential	
Supporting Actors: Systems Administrator			
Stakeholders and Interests: Religious Leaders – interested in parishioners staying engaged with the church community Parishioners – interested in the app saving information from previous sessions Church Administrators/Staff – interested in parishioners staying engaged with the church community App Developers – interested in creating a database that accurately contains user data IT Support – interested in easy maintenance of communication channel			
Brief Description: The User selects the Existing User login portal and enters their login information (username and password) to access their account on the application.			
Trigger: Login to Account Use Case - Directs the user to the login page of their respective user type			
Type (mark one): <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal (rare)			
Relationships: Association: The User Include: Use Case 11 Extend: N/A Generalization: N/A			
The Normal Flow of Events: <ol style="list-style-type: none"> 1. The user selects the existing user option 2. Sign into Account 3. Redirect the User to the Home Page 			
Sub-flows: Sign into account <ol style="list-style-type: none"> 2.1 Enter Username 2.2 Enter Password 			
Alternate/Exceptional Flows: Forgotten password <ol style="list-style-type: none"> 1. User clicks the "Forgot Password" option 2. Redirect to the Forgotten Password Flow 3. Follow Option Instructions In step 2, in the case of forgotten username <ol style="list-style-type: none"> 1. User clicks the "Forgot Username" option 2. Redirect to the Forgotten Username Flow 3. Follow Option Instructions Invalid Login Information <ol style="list-style-type: none"> 1. System displays an error message indicating incorrect username or password 2. User is prompted to re-enter login information or use the "Forgot Username" or "Forgot Password" options if necessary 			

Special Requirements:**Security Requirements**

1. The system must ensure that passwords are stored securely using encryption algorithms
2. All user data must be over a secure (SSL/TLS) connection

Usability Requirements

1. The login form must be accessible and usable on both iOS and Android mobile devices

Performance Requirements

1. The system must process login requests within 2 seconds to ensure a smooth user experience

Backup Requirements

1. Ensure user data is backed up regularly to prevent information loss
2. Implement a recovery mechanism to restore user data in case of system failures

Scalability Requirements

1. The system must handle simultaneous login requests without decreased performance

Compliance Requirements

1. The system must comply with relevant data protection regulations like the GDPR regarding the handling of personal data and storage

Error Handling Requirements

1. Logs errors and notify system administrators if there are critical issues

To do/Issues:

N/A

Use Case Name: Generate Personal Report		ID: UC-9	Importance: Must Have
Primary Actor: Systems Administrator		Use Case Type: Overview, Essential	
Supporting Actors: N/A			
Stakeholders and Interests: Religious Leaders – interested in how parishioners engage with the church community Parishioners – interested in the app saving information from previous sessions Church Administrators/Staff – interested in how parishioners engage with the church community App Developers – interested in a database that accurately generates a report with all user data IT Support – interested in accurate and consistent report data			
Brief Description: The AdoraPlan System automatically compiles all user interactions with the app and composes a personal report to send to the Systems Administrator every 3 months.			
Trigger: Choose Function Use Case – The user decides what they want to do in the application			
Type (mark one): <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal (rare)			
Relationships: Association: Systems Administrator Include: N/A Extend: N/A Generalization: N/A			
The Normal Flow of Events: <ol style="list-style-type: none"> 1. Data Collection 2. Data Processing 3. Report Configuration 4. Report Generation 5. Distribution 6. Backup 			
Sub-flows: Data Collection <ol style="list-style-type: none"> 1.1 Connect to all the User Options 1.2 Extract Data Data Processing <ol style="list-style-type: none"> 2.1 Sort All Data and User Actions Report Configuration <ol style="list-style-type: none"> 3.1 Organize Data into Valid Format 3.2 Arrange Data into A Report Report Generation <ol style="list-style-type: none"> 4.1 Generate Full Report Distribution <ol style="list-style-type: none"> 5.1 Transfer Report to Systems Administrator 			

5.2 Notify Stakeholders

Backup

- 6.1 Archive Old Reports
- 6.2 Backup All Reports

Alternate/Exceptional Flows:

Data Corruption

1. If the system encounters inconsistent or corrupted data
2. System logs the data issues
3. The system notifies the systems administrator of the issue
4. System attempts to correct the data inconsistency automatically
5. If the automatic correction fails, the system flashes the data for manual review

Report generation failure

1. Is the system fail to generate the report
2. The system logs the error and notifies the systems administrator
3. The system retries the report generation process
4. If the retry fails, the system flags the report for manual intervention

Distribution Failure

1. The system fails to transfer the report to the systems administrator
2. System logs the distribution failure
3. The system retries the distribution process
4. If the retry fails, the system notifies the systems administrator and flags the problem for manual intervention

Special Requirements:

Security Requirements

1. All user data must be transmitted over a secure (SSL/TLS) connection

Usability Requirements

1. The personal report generation process must be accessible and usable on both iOS and Android mobile devices

Backup Requirements

1. Ensure user data is backed up regularly to prevent information loss
2. Implement a recovery mechanism to restore user data in case of system failures

Scalability Requirements

1. The system must handle simultaneous report generation processes without decreased performance

Compliance Requirements

1. The system must comply with relevant data protection regulations like the GDPR regarding the handling of personal data and storage

Error Handling Requirements

1. Logs errors and notify system administrators if there are critical issues

To do/Issues:

N/A

Use Case Name: Edit Contact Information		ID: UC-10	Importance: Must Have
Primary Actor: The User		Use Case Type: Overview, Essential	
Supporting Actors: Systems Administrator			
Stakeholders and Interests: Religious Leaders – interested in parishioners staying engaged with the church community Parishioners – interested in the app saving information from previous sessions Church Administrators/Staff – interested in parishioners staying engaged with the church community App Developers – interested in creating and maintaining security and confidentiality of login IT Support – interested in security maintenance of login information			
Brief Description: The user views and can modify their contact information, such as phone number, email address, name, and other relevant communication channels.			
Trigger: Account Management Use case – The user selects which account management function they want to execute			
Type (mark one): <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal (rare)			
Relationships: Association: The user Include: Use Case 2 Extend: N/A Generalization: N/A			
The Normal Flow of Events: <ol style="list-style-type: none"> 1. The user selects "Edit Contact Information" 2. User Modifies Contact Information 3. User Submits Changes 4. System Updates Information 5. Systems logs the Event 			
Sub-flows: Contact Information Modification 2.1 System switches the display to editable fields to modify contact information 2.2 The user updates the necessary fields (e.g., email address, phone number, home address) User Submits Changes 3.1 The user clicks the "Save" button to apply the changes System Updates the Information 4.1 The system saves the updated contact information to the database 4.1 The system confirms that the update was successful and provides a success message to the user			
Alternate/Exceptional Flows: Invalid Input Data <ol style="list-style-type: none"> 1. The user enters invalid data (e.g., incorrect email or phone number format) 2. The system displays an error message next to the invalid field 3. Prevents the user from submitting until the error is corrected Database Error			

1. Issue updating the database
2. The system displays an error message to the user and logs the error
3. Notifies IT support to investigate the error

Special Requirements:

Security Requirements

1. All user data must be over a secure (SSL/TLS) connection

Usability Requirements

1. The contact information modification form must be accessible and usable on both iOS and Android mobile devices

Performance Requirements

1. The system must process contact information updates within 2 seconds to ensure a smooth user experience

Backup Requirements

1. Ensure user data is backed up regularly to prevent information loss
2. Implement a recovery mechanism to restore user data in case of system failures

Scalability Requirements

1. The system must handle simultaneous contact information update requests without decreased performance

Compliance Requirements

1. The system must comply with relevant data protection regulations like the GDPR regarding the handling of personal data and storage
2. Allow users to review and manage their personal information as per legal requirements

Error Handling Requirements

1. Logs errors and notify system administrators if there are critical issues

To do/Issues:

N/A

Use Case Name: View Calendar		ID: UC-11	Importance: Must Have
Primary Actor: The User		Use Case Type: Overview, Essential	
Supporting Actors: Systems Administrator			
Stakeholders and Interests: Religious Leaders – interested in parishioners staying engaged with the church community Parishioners – interested in viewing ongoing events and time-slot availability or need Church Administrators/Staff – interested in parishioners staying engaged with the church community App Developers – ensuring the calendar can sync with other devices and app functions IT Support – intuitive and user-friendly interface for users to navigate and use the calendar			
Brief Description: The user views the calendar with all events and can choose to view either the general calendar with all events or their personal calendar with only the events/time slots they are participating in			
Trigger: Choose Function Use Case – The user decides what they want to do in the application			
Type (mark one): <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal (rare)			
Relationships: Association: The User Include: Use Case 19 Extend: N/A Generalization: N/A			
The Normal Flow of Events: <ol style="list-style-type: none"> 1. The user selects the "Calendar" option from the main menu 2. View General Calendar 3. Optionally View Personal Calendar 			
Sub-flows: View General Calendar <ol style="list-style-type: none"> 2.1 The user can scroll forward or backward to view all events and open time slots 2.2 Click on the act and get redirected to Use Case 4 for Time-Slot Reservation Optionally View Personal Calendar <ol style="list-style-type: none"> 3.1 The user selects the "Personal Calendar" tab on the menu 3.2 Views the events and time slots they signed up for 3.3 Click on the act and get redirected to use Case 4 for Time-Slot Reservation 			
Alternate/Exceptional Flows: No Events of Time Slots Available <ol style="list-style-type: none"> 1. System displays a message indicating no available events or time slots 			
Special Requirements: Security Requirements <ol style="list-style-type: none"> 1. All user data must be over a secure (SSL/TLS) connection Usability Requirements <ol style="list-style-type: none"> 1. The calendar interface must be accessible and usable on both iOS and Android mobile devices 			

Performance Requirements

1. The system must process account creation requests within 2 seconds to ensure a smooth user experience

Backup Requirements

1. Ensure user data is backed up regularly to prevent information loss
2. Implement a recovery mechanism to restore user data in case of system failures

Scalability Requirements

1. The system must handle simultaneous calendar view requests without decreased performance

Compliance Requirements

1. The system must comply with relevant data protection regulations like the GDPR regarding the handling of personal data and storage

Error Handling Requirements

1. Logs errors and notify system administrators if there are critical issues

To do/Issues:

N/A

Use Case Name: Personal Calendar		ID: UC-12	Importance: Must Have
Primary Actor: The User		Use Case Type: Overview, Essential	
Supporting Actors: Systems Administrator			
Stakeholders and Interests: Religious Leaders – interested in parishioners staying engaged with the church community Parishioners – interested in the app saving information from previous sessions Church Administrators/Staff – interested in parishioners staying engaged with the church community App Developers – interested in creating and maintaining security and confidentiality of login IT Support – interested in security maintenance of login information that the tax deduction is made from each employee's salary as per the tax table.>			
Brief Description: The user views their personal calendar containing all the events and adoration time slots that they are participating in			
Trigger: View Calendar Use Case – the user views the general calendar			
Type (mark one): <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal (rare)			
Relationships: Association: The user Include: Use Case 13 Extend: N/A Generalization: N/A			
The Normal Flow of Events: <ol style="list-style-type: none"> 1. User selects the "Personal Calendar" tab in the calendar menu 2. User Views Calendar Events 3. User Can Modify Event 			
Sub-flows: User Views Calendar Events <ol style="list-style-type: none"> 2.1 User can scroll forward and backward to view all events Users can Modify the Event <ol style="list-style-type: none"> 3.1 User clicks on the option they want to modify 3.2 The App redirects the user to Use Case 4 to modify Time-Slot Reservation 			
Alternate/Exceptional Flows: No Events Found <ol style="list-style-type: none"> 1. The System displays a message indicating that the user is not participating in any events or time slots 2. User is prompted to return to the main menu and choose a time slot function 			
Special Requirements: Security Requirements <ol style="list-style-type: none"> 1. All user data must be over a secure (SSL/TLS) connection Usability Requirements <ol style="list-style-type: none"> 1. The personal calendar interface must be accessible and usable on both iOS and Android mobile devices 			

Performance Requirements

1. The system must process account creation requests within 2 seconds to ensure a smooth user experience

Backup Requirements

1. Ensure user data is backed up regularly to prevent information loss
2. Implement a recovery mechanism to restore user data in case of system failures

Scalability Requirements

1. The system must handle simultaneous personal calendar view requests without decreased performance

Compliance Requirements

1. The system must comply with relevant data protection regulations like the GDPR regarding the handling of personal data and storage

Error Handling Requirements

1. Logs errors and notify system administrators if there are critical issues

To do/Issues:

N/A

Use Case Name: Forgot Password		ID: UC-13	Importance: Must Have
Primary Actor: The User		Use Case Type: Overview, Essential	
Supporting Actors: Systems Administrator			
Stakeholders and Interests: Religious Leaders – interested in parishioners staying engaged with the church community Parishioners – interested in the app security of password Church Administrators/Staff – interested in parishioners staying engaged with the church community App Developers – interested in creating and maintaining security and confidentiality of login IT Support – interested in security maintenance of login information			
Brief Description: The user enters their email or phone number and receives a password reset link. The system updates the user's profile and sends them to the sign-in page.			
Trigger: Existing User Use Case – the user wants to log in as an existing user and is asked for their login information			
Type (mark one): <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal (rare)			
Relationships: Association: The User Include: Use Case 8 Extend: N/A Generalization: Use Case 18			
The Normal Flow of Events: <ol style="list-style-type: none"> 1. Ask User for an Email Or Phone Number Associated with their Account 2. System Sends a Recovery link to the Chosen Recovery Method 3. User resets password 4. System updates user password 5. System redirects the user to the login page 			
Sub-flows: Ask for the Preferred Recovery Method <ol style="list-style-type: none"> 1.1 Allow user to choose between email or phone number for recovery method 1.2 User enters the information needed for the preferred recovery method 1.3 Validate the input to ensure it matches the format of an email or phone number User resets Password <ol style="list-style-type: none"> 3.1 User clicks on the link sent to them 3.2 System gives a time limit of ten minutes for the user to reset their password 3.3 User enters the new password 3.4 User click the "Save New Password" button The System Updates the User's Password <ol style="list-style-type: none"> 4.1 Update the user's password in the system's database 4.2 Ensure the password is securely hashed before storing it System Redirects the User to the Login Page			

- 5.1 Display a success message confirming that the password has been reset successfully
- 5.2 Redirect User to login page

Alternate/Exceptional Flows:

Invalid Recovery Information

1. Enter an email or phone number not associated with any account
2. The system displays an error message indicating that the provided information is invalid
3. User is prompted to re-enters the information or contact system administrator/IT support for more assistance

Expired Recovery Link

1. User clicks the recovery link after the time limit has expired
2. System displays a message indicating that the link has expired
3. User is prompted to request a new recovery link

Special Requirements:

Security Requirements

1. The system must ensure that passwords are stored securely using encryption algorithms
2. All user data must be over a secure (SSL/TLS) connection
3. Recovery links must be time-sensitive and expire after ten minutes for security reasons

Usability Requirements

4. The password reset form must be accessible and usable on both iOS and Android mobile devices

Performance Requirements

5. The system must process password reset requests within 2 seconds to ensure a smooth user experience
6. Email or SMS containing the recovery link must be sent within 5 seconds of the request

Backup Requirements

7. Ensure user data is backed up regularly to prevent information loss
8. Implement a recovery mechanism to restore user data in case of system failures

Scalability Requirements

9. The system must handle simultaneous password reset requests without decreased performance

Compliance Requirements

10. The system must comply with relevant data protection regulations like the GDPR regarding the handling of personal data and storage

Error Handling Requirements

11. Logs errors and notify system administrators if there are critical issues

To do/Issues:

N/A

Use Case Name: Forgot Username		ID: UC-14	Importance: Must Have
Primary Actor: The User		Use Case Type: Overview, Essential	
Supporting Actors: Systems Administrator			
Stakeholders and Interests: Religious Leaders – interested in parishioners staying engaged with the church community Parishioners – interested in the app security of password Church Administrators/Staff – interested in parishioners staying engaged with the church community App Developers – interested in creating and maintaining security and confidentiality of login IT Support – interested in security maintenance of login information			
Brief Description: The user enters their email or phone number and receives a link with their username. The system then sends them to the sign-in page after they recover their username.			
Trigger: Existing User Use Case – the user wants to log in as an existing user and is asked for their login information			
Type (mark one): <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal (rare)			
Relationships: Association: The User Include: Use Case 8 Extend: N/A Generalization: N/A			
The Normal Flow of Events: <ol style="list-style-type: none"> 1. Ask User for an Email Or Phone Number Associated with their Account 2. System Sends a link containing the username to the Chosen Recovery Method 3. The user clicks the link, viewing the username 4. System redirects the user to the login page 			
Sub-flows: Ask for the Preferred Recovery Method <ol style="list-style-type: none"> 1.1 Allow user to choose between email or phone number for recovery method 1.2 User enters the information needed for the preferred recovery method 1.3 Validate the input to ensure it matches the format of an email or phone number View Username <ol style="list-style-type: none"> 3.1 System gives a time limit of ten minutes for the user to view the username 3.2 User clicks the link and views the username 			
Alternate/Exceptional Flows: Invalid Recovery Information <ol style="list-style-type: none"> 1. Enter an email or phone number not associated with any account 2. The system displays an error message indicating that the provided information is invalid 3. User is prompted to re-renters the information or contact system administrator/IT support for more assistance Expired Recovery Link <ol style="list-style-type: none"> 1. User clicks the recovery link after the time limit has expired 2. System displays a message indicating that the link has expired 			

3. User is prompted to request a new recovery link

Special Requirements:

Security Requirements

1. The system must ensure that passwords are stored securely using encryption algorithms
2. All user data must be over a secure (SSL/TLS) connection
3. Recovery links must be time-sensitive and expire after ten minutes for security reasons

Usability Requirements

1. The account recovery process must be accessible and usable on both iOS and Android mobile devices

Performance Requirements

1. The system must process account recovery requests within 2 seconds to ensure a smooth user experience
2. Email or SMS containing the recovery link must be sent within 5 seconds of the request

Backup Requirements

1. Ensure user data is backed up regularly to prevent information loss
2. Implement a recovery mechanism to restore user data in case of system failures

Scalability Requirements

1. The system must handle simultaneous account creation requests without decreased performance

Compliance Requirements

1. The system must comply with relevant data protection regulations like the GDPR regarding the handling of personal data and storage

Error Handling Requirements

1. Logs errors and notify system administrators if there are critical issues

To do/Issues:

N/A

Use Case Name: Choose Function		ID: UC-15	Importance: Must Have
Primary Actor: The User		Use Case Type: Overview, Essential	
Supporting Actors: Systems Administrator			
Stakeholders and Interests: Religious Leaders – interested in parishioners staying engaged with the church community Parishioners – interested in the different features of the app Church Administrators/Staff – interested in parishioners staying engaged with the church community App Developers – interested in developing an app with all the functional requirements IT Support – interested in the features acting in the way they were made to function			
Brief Description: The main menu on the homepage allows the user to access and perform different features on the app, like Time-Slot Reservation, Account Management, Scanning Barcode, Viewing Calendar, Event Creation, Q&A General Chat, and accessing App assistance.			
Trigger: Existing User Use Case – existing users log into their account and are directed to the homepage. Account Creation Use Case – A new user creates an account, logins, and is directed to the homepage			
Type (mark one): <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal (rare)			
Relationships: Association: The User Include: Use Case 8, Use Case 1 Extend: N/A Generalization: N/A			
The Normal Flow of Events: <ol style="list-style-type: none"> 1. Display Main Menu 2. User Chooses Action From Main Menu 3. Systems redirects the user to the display of the action selected 4. The user interacts with the function selected 			
Sub-flows: Display Main Menu <ol style="list-style-type: none"> 1.1 User selects the main menu tab in the top center of the app 1.2 System retrieves the available functions for the user 1.3 System displays the main menu with the list of available functions User Choose Action From Main Menu <ol style="list-style-type: none"> 2.1 User reviews menu options 2.2 User selects a function by tapping the corresponding menu item Redirect User to Action Selected <ol style="list-style-type: none"> 3.1 System retrieves necessary data or setting required to execute the chosen function 3.2 System loads the appropriate modules and components associated with the function 3.3 Systems render the screen related to the chosen function Display Function Screen <ol style="list-style-type: none"> 4.1 User interacts with the function as intended 			

4.2 The system responds to user interaction appropriately

Alternate/Exceptional Flows:

Function Not Available

1. User selects a function that is temporarily unavailable (e.g., under maintenance)
2. The system displays a message indicating that the function is currently unavailable and suggests trying again later

Special Requirements:

Security Requirements

2. All user data must be over a secure (SSL/TLS) connection

Usability Requirements

2. The main menu and its features must be accessible and usable on both iOS and Android mobile devices
3. The interface must be user-friendly to ensure easy navigation through app functions

Performance Requirements

3. The system must process user requests within 2 seconds to ensure a smooth user experience
4. The main menu should load and display available functions within 1 second of selection

Backup Requirements

5. Ensure user data is backed up regularly to prevent information loss
6. Implement a recovery mechanism to restore user data in case of system failures

Scalability Requirements

7. The system must handle simultaneous requests to access and use different app functions without decreased performance

Compliance Requirements

8. The system must comply with relevant data protection regulations like the GDPR regarding the handling of personal data and storage

Error Handling Requirements

3. Logs errors and notify system administrators if there are critical issues

To do/Issues:

N/A

Use Case Name: Cancel Time-Slot		ID: UC-16	Importance: Must Have
Primary Actor: The User		Use Case Type: Overview, Essential	
Supporting Actors: Systems Administrator			
Stakeholders and Interests: Religious Leaders – interested in parishioners staying engaged with church activities Parishioners – interested in easy time slots and event modification Church Administrators/Staff – interested in parishioners staying involved with church activities App Developers – interested in creating seamless time slot and event modification functions IT Support – interested in the features acting in the way they were made to function			
Brief Description: Cancels the user's adoration time slot indefinitely, removing them from their weekly time commitment.			
Trigger: Time-Slot Reservation Use case – The user decides what to do with the time slot they selected			
Type (mark one): <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal (rare)			
Relationships: Association: The User Include: Use Case 4 Extend: N/A Generalization: N/A			
The Normal Flow of Events: <ol style="list-style-type: none"> 1. Review the time slot or event chosen 2. Submit Cancellation 3. System Removes a user from event indefinitely 4. System republishes time-slot on general calendar 5. The system notifies all users in the Q&A channel 			
Sub-flows: Review Time Slot or Event Chosen <ol style="list-style-type: none"> 1.1 Display the details of the selected event or time slot for confirmation 1.2 Prompt the user to confirm the cancellation Submit Cancellation <ol style="list-style-type: none"> 2.1 User confirms the cancellation 2.2 System records the cancellation request System Removes User from Event Indefinitely <ol style="list-style-type: none"> 3.1 System processes Cancellation request 3.2 System removes the user from the event or time slot 3.3 Update the event of time slot status to reflect the user's removal System Republishes Event/Time Slot on General Calendar <ol style="list-style-type: none"> 4.1 System marks the time slot/event as available 4.2 Update the general calendar to reflect the newly available time slot 4.3 Ensure the availability is visible to all potential users 			

System Notifies all users in the Q&A Channel

- 5.1 System generates a notification about the cancellation
- 5.2 Notification includes details about the newly available time slot of the canceled event
- 5.3 Post the notification in the Q&A channel
- 5.4 Ensure the notification is visible to all users in the app

Alternate/Exceptional Flows:

User Changes Mind During Cancellation Process

1. User decides not to cancel the time slot after reviewing the details
2. User cancels the cancellation request
3. The system returns the user to the main menu without making changes

System Fails to Process Cancellation Request

1. The system encounters an error while processing the cancellation
2. The system displays an error message
3. User is prompted to try again or contact IT support/Systems Administrator

Special Requirements:

Security Requirements

1. All user data must be over a secure (SSL/TLS) connection

Usability Requirements

1. The cancellation process must be accessible and usable on both iOS and Android mobile devices
2. The interface must provide clear instructions throughout the cancellation process

Performance Requirements

1. The system must process cancellation requests within 2 seconds to ensure a smooth user experience
2. Notifications about cancellations must be sent within 5 seconds of submission

Backup Requirements

1. Ensure user data is backed up regularly to prevent information loss
2. Implement a recovery mechanism to restore user data in case of system failures

Scalability Requirements

1. The system must handle simultaneous cancellation requests without decreased performance

Compliance Requirements

1. The system must comply with relevant data protection regulations like the GDPR regarding the handling of personal data and storage

Error Handling Requirements

1. Logs errors and notify system administrators if there are critical issues

To do/Issues:

N/A

Use Case Name: Drop Time-Slot		ID: UC 17	Importance: Must Have
Primary Actor: Main User		Use Case Type: Overview, Essential	
Supporting Actors: Another user, the Systems Administrator			
Stakeholders and Interests: Religious Leaders – interested in parishioners staying engaged with church activities Parishioners – interested in easy time slots and event modification Church Administrators/Staff – interested in parishioners staying involved with church activities App Developers – interested in creating seamless time slot and event modification functions IT Support – interested in the features acting in the way they were made to function			
Brief Description: The user decides that they want to drop a one-time slot. This option cancels the selected time slot and re-publishes it on the time slot reservation calendar for someone else to cover			
Trigger: Time-Slot Reservation Use case – The user decides what to do with the time slot they selected			
Type (mark one): <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal (rare)			
Relationships: Association: The User Include: Use Case 4 Extend: N/A Generalization: N/A			
The Normal Flow of Events: <ol style="list-style-type: none"> 1. Review the time slot or event chosen 2. Submit Drop 3. System Removes a user from event indefinitely 4. System republishes time-slot on general calendar 5. The system notifies all users in the Q&A channel 			
Sub-flows: Review Time Slot or Event Chosen <ol style="list-style-type: none"> 1.1 Display the details of the selected event or time slot for confirmation 1.2 Prompt the user to confirm the drop Submit Drop <ol style="list-style-type: none"> 2.1 User confirms the Drop 2.2 System records the cancellation request System Removes User from Event One-Time <ol style="list-style-type: none"> 3.1 System processes Cancellation request 3.2 System removes the user from the event or time slot 3.3 Update the event of time slot status to reflect the user's removal System Republishes Event/Time Slot on General Calendar <ol style="list-style-type: none"> 4.4 System marks the time slot/event as available 4.5 Update the general calendar to reflect the newly available time slot 			

4.6 Ensure the availability is visible to all potential users

System Notifies all users in the Q&A Channel

- 5.1 System generates a notification about the drop
- 5.2 Notification includes details about the newly available time slot of the canceled event
- 5.3 Post the notification in the Q&A channel
- 5.4 Ensure the notification is visible to all users in the app

Alternate/Exceptional Flows:

User Changes Mind During Drop Process

1. User decides not to drop the time slot after reviewing the details
2. User cancels the drop request
3. The system returns the user to the main menu without making changes

System Fails to Process Drop Request

1. The system encounters an error while processing the drop
2. The system displays an error message
3. User is prompted to try again or contact IT support/Systems Administrator

Special Requirements:

Security Requirements

1. All user data must be over a secure (SSL/TLS) connection

Usability Requirements

1. The drop process must be accessible and usable on both iOS and Android mobile devices
2. Interface must provide clear instructions throughout the drop process

Performance Requirements

1. The system must process drop requests within 2 seconds to ensure a smooth user experience
2. Notifications about drops must be sent within 5 seconds of submission

Backup Requirements

1. Ensure user data is backed up regularly to prevent information loss
2. Implement a recovery mechanism to restore user data in case of system failures

Scalability Requirements

1. The system must handle simultaneous drop requests without decreased performance

Compliance Requirements

1. The system must comply with relevant data protection regulations like the GDPR regarding the handling of personal data and storage

Error Handling Requirements

1. Logs errors and notify system administrators if there are critical issues

To do/Issues:

N/A

Use Case Name: Trade Time Slot		ID: UC 18	Importance: Must Have
Primary Actor: The User		Use Case Type: Overview, essential	
Supporting Actors: Other User, Systems Administrator			
Stakeholders and Interests: Religious Leaders – interested in parishioners staying engaged with church activities Parishioners – interested in easy time slots and event modification Church Administrators/Staff – interested in parishioners staying involved with church activities App Developers – interested in creating seamless time slot and event modification functions IT Support – interested in the features acting in the way they were made to function			
Brief Description: The user decides to trade their selected time slot for a different time. It re-publishes the time slot and swaps the user's time slot with the time slot of the user who accepts their trade.			
Trigger: Time-Slot Reservation Use case – The user decides what to do with the time slot they selected			
Type (mark one): <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal (rare)			
Relationships: Association: The User Include: Use Case 4 Extend: N/A Generalization: N/A			
The Normal Flow of Events: <ol style="list-style-type: none"> 1. Initiate Trade Request 2. Submit a Trade 3. Process Trade 4. Notify Parties 			
Sub-flows: Initiate Trade Requests <ol style="list-style-type: none"> 1.1 Display the user's current selected time slot 1.2 User confirms trade 1.3 Time slot is sent to Q&A chat requesting a trade Submit Trade <ol style="list-style-type: none"> 2.1 Another User views the trade request 2.2 Other user selects the time slot in which they want to trade 2.3 Another user confirms the trade Process Trade <ol style="list-style-type: none"> 3.1 System processes trade 3.2 System swaps the time slots being traded to the correct user 3.3 System updates the new time slot in each user calendar Notify Parties <ol style="list-style-type: none"> 4.1 System sends notification of the thriving trade 4.2 System sends the new trade details to each user 			

Alternate/Exceptional Flows:**User Changes Mind During Trade Process**

1. User decides not to trade the time slot after reviewing the details
2. User cancels the trade request
3. The system returns the user to the main menu without making changes

System Fails to Process Trade Request

1. The system encounters an error while processing the drop
2. The system displays an error message
3. User is prompted to try again or contact IT support/Systems Administrator

Special Requirements:**Security Requirements**

1. All user data must be over a secure (SSL/TLS) connection

Usability Requirements

1. The cancellation process must be accessible and usable on both iOS and Android mobile devices
2. The interface must provide clear instructions throughout the cancellation process

Performance Requirements

1. The system must process cancellation requests within 2 seconds to ensure a smooth user experience
2. Notifications about cancellations must be sent within 5 seconds of submission

Backup Requirements

1. Ensure user data is backed up regularly to prevent information loss
2. Implement a recovery mechanism to restore user data in case of system failures

Scalability Requirements

1. The system must handle simultaneous cancelation requests without decreased performance

Compliance Requirements

1. The system must comply with relevant data protection regulations like the GDPR regarding the handling of personal data and storage

Error Handling Requirements

1. Logs errors and notify system administrators if there are critical issues

To do/Issues:

N/A

Use Case Name: Reserve Time-Slot		ID: UC 19	Importance: Must Have
Primary Actor: User		Use Case Type: Overview, Essential	
Supporting Actors: Systems Administrator			
Stakeholders and Interests: Religious Leaders – interested in parishioners staying engaged with church activities Parishioners – interested in easy time slots and event modification Church Administrators/Staff – interested in parishioners staying involved with church activities App Developers – interested in creating seamless time slot and event modification functions IT Support – interested in the features acting in the way they were made to function			
Brief Description: The user decides to reserve the time slot selected as their weekly, reoccurring adoration time slot. This option binds that time slot to that user unless they choose to change it.			
Trigger: Time-Slot Reservation Use case – The user decides what to do with the time slot they selected			
Type (mark one): <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal (rare)			
Relationships: Association: Customer Include: Use case 4 Extend: N/A Generalization: N/A			
The Normal Flow of Events: <ol style="list-style-type: none"> 1. Review the time slot or event chosen 2. Submit Reservation 3. System updates the user's calendar with the reserved time slot 4. System confirms successful reservation 			
Sub-flows: Review Time Slot or Event Chosen <ol style="list-style-type: none"> 1.1 Display the details of the selected event or time slot for confirmation 1.2 Prompt the user to confirm the reservation Submit Reservation <ol style="list-style-type: none"> 2.1 User confirms the Reservation 2.2 System records the reservation request System Removes Selected Event or Time Slot <ol style="list-style-type: none"> 3.1 System processes Reservation request 3.2 System removes the event or time slot from the available events or time slots on the general calendar 3.3 Update the event of time slot status to reflect the user's reservation System Notifies User <ol style="list-style-type: none"> 5.1 System generates a notification about the reservation 5.2 Notification includes details about the newly available time slot of the reserved event 5.3 System sends the user a notification about a successful reservation 			
Alternate/Exceptional Flows:			

User Changes Mind During Reservation Process

1. User decides not to reserve the time slot after reviewing the details
2. User cancels the reserve request
3. The system returns the user to the main menu without making changes

System Fails to Process Reservation Request

1. The system encounters an error while processing the drop
2. The system displays an error message
3. User is prompted to try again or contact IT support/Systems Administrator

Special Requirements:**Security Requirements**

1. All user data must be over a secure (SSL/TLS) connection

Usability Requirements

1. The reservation process must be accessible and usable on both iOS and Android mobile devices
2. Interface must provide clear instructions throughout the reservation process

Performance Requirements

1. The system must process reservation requests within 2 seconds to ensure a smooth user experience
2. Notifications about reservation must be sent within 5 seconds of submission

Backup Requirements

1. Ensure user data is backed up regularly to prevent information loss
2. Implement a recovery mechanism to restore user data in case of system failures

Scalability Requirements

1. The system must handle simultaneous reservation requests without decreased performance

Compliance Requirements

1. The system must comply with relevant data protection regulations like the GDPR regarding the handling of personal data and storage

Error Handling Requirements

1. Logs errors and notify system administrators if there are critical issues

To do/Issues:

N/A

Use Case Name: Sign in		ID: UC 20	Importance: Must Have
Primary Actor: User		Use Case Type: Overview, Essential	
Supporting Actors: Systems Administrator			
Stakeholders and Interests: Religious Leaders – interested in parishioners staying engaged with church activities Parishioners – interested in secure login Church Administrators/Staff – interested in parishioners staying involved with church activities App Developers – interested in encrypting login information IT Support – interested in the security of sign-in information			
Brief Description: The Customer has recovered their login information and wants to sign back into their account.			
Trigger: Forgot Password Use case – User tries to recover password Forgot Username Use case – User tries to recover username			
Type (mark one): <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal (rare)			
Relationships: Association: Customer Include: Use case 16, Use case 17 Extend: N/A Generalization: N/A			
The Normal Flow of Events: <ol style="list-style-type: none"> 1. System displays sign-in form 2. User enters credentials 3. Submit Credentials 4. Validate Credentials 5. Successful Sign in 			
Sub-flows: Display Sign-in Form <ol style="list-style-type: none"> 1.1 System displays sign-in page with fields for username and password 1.2 Displays the "Sign In" button below the credentials Enter Credentials <ol style="list-style-type: none"> 2.1 User enters their username 2.2 User enters their password Submit Credentials <ol style="list-style-type: none"> 3.1 User clicks the "Sign In" button 3.2 System captures the entered credentials Validate Credentials <ol style="list-style-type: none"> 4.1 System checks if the entered username exists 4.2 The system validates the password for the corresponding username 4.3 If invalid, the system displays an error message and prompts the user to try again 			

<p>Successful Sign in</p> <ol style="list-style-type: none"> 5.1 Username and password are valid, so the System grants access to the user 5.2 System pulls up a user session 5.3 System redirects the user to the homepage
<p>Alternate/Exceptional Flows:</p> <p>Incorrect login credentials</p> <ol style="list-style-type: none"> 1. System displays an error message indicating that the credentials are invalid 2. The user is prompted to re-enter their username and password 3. User attempts to sign in again <p>Technical issues during sign-in</p> <ol style="list-style-type: none"> 1. The system displays an error message indicating that there is a problem with signing in 2. The user is advised to try again later 3. If the issue persists, the system displays a message to contact the IT support/Systems administrator for assistance
<p>Special Requirements:</p> <p>Security Requirements</p> <ol style="list-style-type: none"> 1. The system must ensure that passwords are stored securely using encryption algorithms 2. All user data must be over a secure (SSL/TLS) connection <p>Usability Requirements</p> <ol style="list-style-type: none"> 1. The login form must be accessible and usable on both iOS and Android mobile devices 2. Provide clear error messages to guide users in resolving sign-in issues <p>Performance Requirements</p> <ol style="list-style-type: none"> 1. The system must process sign-in requests within 2 seconds to ensure a smooth user experience <p>Backup Requirements</p> <ol style="list-style-type: none"> 1. Ensure user data is backed up regularly to prevent information loss 2. Implement a recovery mechanism to require user data in case of system failures <p>Scalability Requirements</p> <ol style="list-style-type: none"> 1. The system must handle simultaneous login requests without decreased performance 2. Ensure that the system database architecture can scale to accommodate the growing number of user accounts <p>Compliance Requirements</p> <ol style="list-style-type: none"> 1. The system must comply with relevant data protection regulations like the GDPR regarding the handling of personal data and storage <p>Error Handling Requirements</p> <ol style="list-style-type: none"> 1. Logs errors and notify system administrators if there are critical issues
<p>To do/Issues:</p> <p>N/A</p>

6.0 System Evolution

Future Feature:

1. Barcode Scanner Integration
 - a. System implementation of a barcode for door lock access unique to each user
 - b. Could enhance this feature with Quick Response (QR) codes that include additional data such as event details or user-specific access permissions
 - c. Upgrade barcode scanner to support QR codes and integrate with Internet of Thing (IoT) enabled door locks for secure access management
2. Expand Event Management
 - a. Develop a more comprehensive event management module with functions for setting up recurring events and managing attendee lists
3. Hardware and software upgrades
 - a. As user demand grows, upgrade server infrastructure to accommodate more powerful processors, increased Random-access memory (RAM), and faster storage solutions
 - b. Transition to scalable database solutions like Google Cloud SQL for better performance

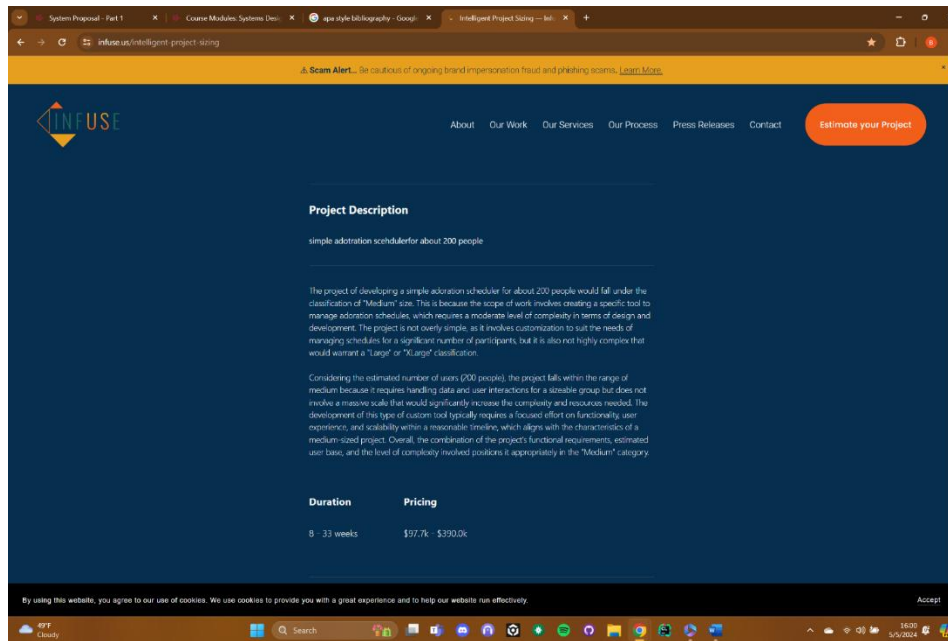
7.0 Conclusions and Recommendations.

The development of AdoraPlan aims to provide a user-friendly and robust digital platform for managing even scheduling, user interactions, and profile management. The app's MVP version focuses on the essential functionalities of user account creation, login, profile management, time slot or event reservation, event creation, and a general Q&A chat. These core features are designed to meet users' current needs and future enhancements. Through detailed documentation and tedious planning, we ensure that the app adheres to data protection laws, maintains high performance, and offers a seamless user experience; overall, AdoraPlan is well-positioned to launch successfully.

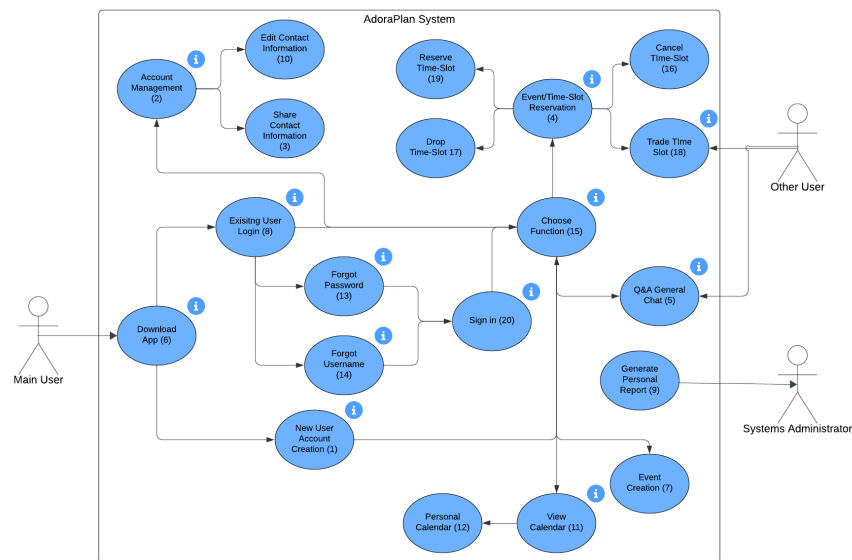
Recommendations:

1. Focus on User Feedback Post-Launch
 - a. Actively gather and access user feedback after the MVP launch to identify areas for improvement
2. Invest in Scalability and Performance Optimization. Implementing cloud-based solutions and architecture will support growth as user numbers grow
3. Plan for Regular Updates and Maintenance
 - a. **Establish a regular schedule for updates and maintenance to address bugs, implement improvements, or roll out new features**

Appendices



Intelligent Project Sizing. (n.d.). Infuse. <https://www.infuse.us/intelligent-project-sizing>



AdoraPlan Use-Case Diagram / Lucidchart. (n.d.). Lucid.app. Retrieved May 22, 2024, from https://lucid.app/lucidchart/fdd76a98-56eb-443d-b375-4ac297ec9798/edit?viewport_loc=-328%2C-206%2C1036%2C1612%2C.Q4MUjXso07N&invitationId=inv_37e51767-eaee-4ba9-b6fb-5c4cfc43561b

Glossary

Term :	Definition :
Random Access Memory (RAM)	"RAM is your computer or laptop's short-term memory. It's where the data is stored that your computer processor needs to run your applications and open your files." (Intel)
Question and Answer (Q&A)	is a form where people can pose questions and reply to other inquiries
Portable Network Graphics (PNG)	graphics file format that supports lossless data compression
Metadata	Data that gives more information about other data
Megabyte (MB)	is a unit of digital information storage equal to 1,000,000 bytes (KiloBit to MegaByte Converter)
Joint Photographic Experts Group (JPEG)	is a commonly used method of lossy compression for digital images
Information Technology (IT)	"the technology involving the development, maintenance, and use of computer systems, software, and networks for the processing and distribution of data" (Merriam Webster, 2017)
General Data Protection Regulation (GDPR)	"the world's toughest privacy and security law. Though it was drafted and passed by the European Union (EU), it imposes obligations onto organizations anywhere, so long as they target or collect data related to people in the EU" (Wolford, What is GDPR, the EU's New Data Protection Law)
Fair Labor Standards Act (FLSA)	federal law in the United States that establishes minimum wage, overtime pay eligibility, and child labor practices affecting full-time and part-time workers in the private sector and federal, state, and local government (Overtime/Wage and Hour Activities), (Volume V: Personnel Policies for Administrators and Staff)
Agile	"Agile development focuses on creating working software quickly, collaborating with customers frequently, and adapting to changes easily... especially beneficial for projects that are complex or have uncertain requirements." (Opentext)
AdoraPlan	is the project title, which is short for Adoration Planner
Secure Sockets Layer and Transport Layer Security (SSL/TLS)	"Secure Sockets Layer (SSL) is a communication protocol, or set of rules, that creates a secure connection between two devices or applications on a network. Transport Layer Security (TLS) is the upgraded version of SSL that fixes existing SSL vulnerabilities. TLS authenticates more efficiently and continues to support encrypted communication channels." (Amazon Web Services)

Bibliography

- Amazon Web Services. (2024). *SSL vs TLS - Comparing Communication Protocols - AWS*. Amazon Web Services, Inc. <https://aws.amazon.com/compare/the-difference-between-ssl-and-tls/>
- Cameron, A (2024) CSC3150–System Proposal Part1+Part2–Document Template v9
- Cameron, A (2024) WO4 – Feasibility and Risk
- Cameron, A (2024) WO5–Requirement, What We Are Seeking
- Cameron, A (2024) WO5–Nonfunctional Requirement Triggers
- Cameron, A (2024) Use Case Description Template
- Perpetual Adoration Ministry*. (n.d.). Saint Theresa Catholic Church. Retrieved May 4, 2024, from <https://sugarlandcatholic.com/ministry/perpetual-adoration-ministry/>
- Feasibility Study & Feasibility Study about Business*. (n.d.). ValuStrat. Retrieved May 5, 2024, from <https://valustrat.com/pages/feasibility-studies>
- Intel. (n.d.). *Computer and Laptop RAM*. Intel. <https://www.intel.com/content/www/us/en/tech-tips-and-tricks/computer-ram.html>
- Merriam-Webster. (2017). *Definition of INFORMATION TECHNOLOGY*. Merriam-Webster.com. <https://www.merriam-webster.com/dictionary/information%20technology>
- Optimizing Financial Industry Awards Organizing with CRM Solutions–Lariada.pk*. (2024, January 24). <https://lariada.pk/optimizing-financial-industry-awards-organizing-with-crm-solutions/>
- What is Agile Development and why is it important?* (n.d.). OpenText. <https://www.opentext.com/what-is/agile-development>
- Wolford, B. (2024). *What Is GDPR, the EU's New Data Protection Law?* GDPR.eu. <https://gdpr.eu/what-is-gdpr/>
- KiloBit to MegaByte Converter*. (n.d.). Poco.tools. Retrieved May 6, 2024, from <https://poco.tools/convert/kilobit/megabyte>
- Overtime/Wage and Hour Archives*. (n.d.). Tycko & Zavareei LLP. Retrieved May 6, 2024, from <https://www.tzlegal.com/practice-areas/overtime-wage-and-hour/>
- (2020). Volume V: Personnel Policies for Administrators and Staff. <https://core.ac.uk/download/337399394.pdf>
-