Rejoice Church Project

The Architects

Anna Dowhower: (405)432–8998, adowhower@ou.edu

Anthony Houston: (972)838-5567, anthony.houston@ou.edu

MIS 3373 – 001

December 10, 2021

Table of Contents

Project Report ……………………………………………………………..............3

Project Overview………………………………………………………………………….3

Executive Summary…………………………………………………………………….3

Assumptions………………………………………………………………………………..4

Problems Encountered………………………………………………………………..6

Entity Relationship Diagram (ERD)…..7

Data Flow Diagrams (DFD)…..8

Context Diagram...8

Systems Diagram...9

Lower Levels...10

Data Models...13

Data Flows…..13

Metadata…..17

Process Specs...18

Interface and Output Designs..21

Input Design..21

Output Design...22

Dialog Diagram...23

Prototype...24

Final Project Report

I. Project Overview

When first presented with the scenario and project deliverables, we were slightly overwhelmed with the many different diagrams and forms required for the project. However, as we began developing the project, tensions alleviated as the project came together like puzzle pieces. We quickly realized that all the deliverables, including but not limited to the: Dialog Diagram, ERD, Process Specifications, and Data Flow Diagram, worked together to serve the function of fully modeling the system. Due to unexpected team changes, we weren't able to follow our initial project plan. However, after making some adjustments our team was able to efficiently complete all required deliverables. If we were to manage the team any differently, we would have been stricter on deadlines to prevent tasks from getting backlogged. Overall, *The Architects* developed a great understanding of modeling systems, and we hope to apply the skills learned in future projects.

Executive Summary

Rejoice Church located in Grand Rapids, Michigan has exploded in size since Reverend Timothy Beck took leadership and the church is now looking to accommodate its members by building a scalable system to handle all information processing. Due to the rapid growth of the church, church management has been plagued with the manual processing of paper forms and they'd like to begin automating some of these business functions. Rejoice Church has tasked *The Architects* with designing a platform that is user-friendly, functional, and capable of scaling for future growth.

The Architects' solution to create and implement the system first involved analyzing the functions of the church and highlighting the needs and wants. To provide a network representation of the system, data flow diagrams including: a Systems Diagram, with detailed lower-level processes, and a Context Diagram were developed to determine and model logical requirements. To model the data that the system will collect, an Entity Relationship Diagram was created that depicts the relationships entities have with one another. Additionally, a Data Dictionary and Metadata were put together to define attributes, data flows and data stores. To describe the actions taken by the system, a process specification, using structured English, allowed the team to represent the internal structure and functionality of the processes. The Architects also ensured that the interface of the system was functional by creating an input form and a dialog diagram. Lastly, a prototype was created using a Microsoft WPF application, which provided a visual and functional representation of how the user will interact with the application.

The Architects are committed to providing their clients with the best experience when it comes to handling and managing their information and our team is excited to see where this newly implemented system takes Rejoice Church in the near future!

II. Assumptions

In completing this project, we had no choice but to make assumptions regarding certain aspects of the project. These assumptions were all carefully thought out. Before making them, we looked for alternative solutions. If there was no clear solution, we considered what would be most logical concerning the client and their users.

Assumption 1: In the final working system, all reports will be automatically generated and uploaded to the “View Reports” window of the user interface. This implies that the user will not have to upload or create the reports. A specific format will be designed for each report type, and the system will automatically fill in the data. This will save the users time and reduce data errors and mistakes.

Assumption 2: Ceremonies are performed for members only. In other words, for a reverend to conduct a wedding, funeral, or baptism, the participants must be a member of the church. This allows the church to document ceremonies with greater ease. It also ensures that they will not be overwhelmed with ceremony requests that cannot be fulfilled due to overscheduling. If this was not the case, members might be denied ceremonies because non-members reserved a timeslot first.

Assumption 3: All loose collections from the collection plate are cash and change only. This means that the donations cannot be traced back to a specific person. This may also account for any donations from guests and visitors who are not members.

Assumption 4: Any course information is added to the “Ceremonies” data store through a process unrelated to this system. Although information regarding the completion of a course is required for this system, the process of scheduling and signing up for them is not part of this system. Therefore, we assume that in another system, course completion will be input into “Ceremonies” for use in this system.

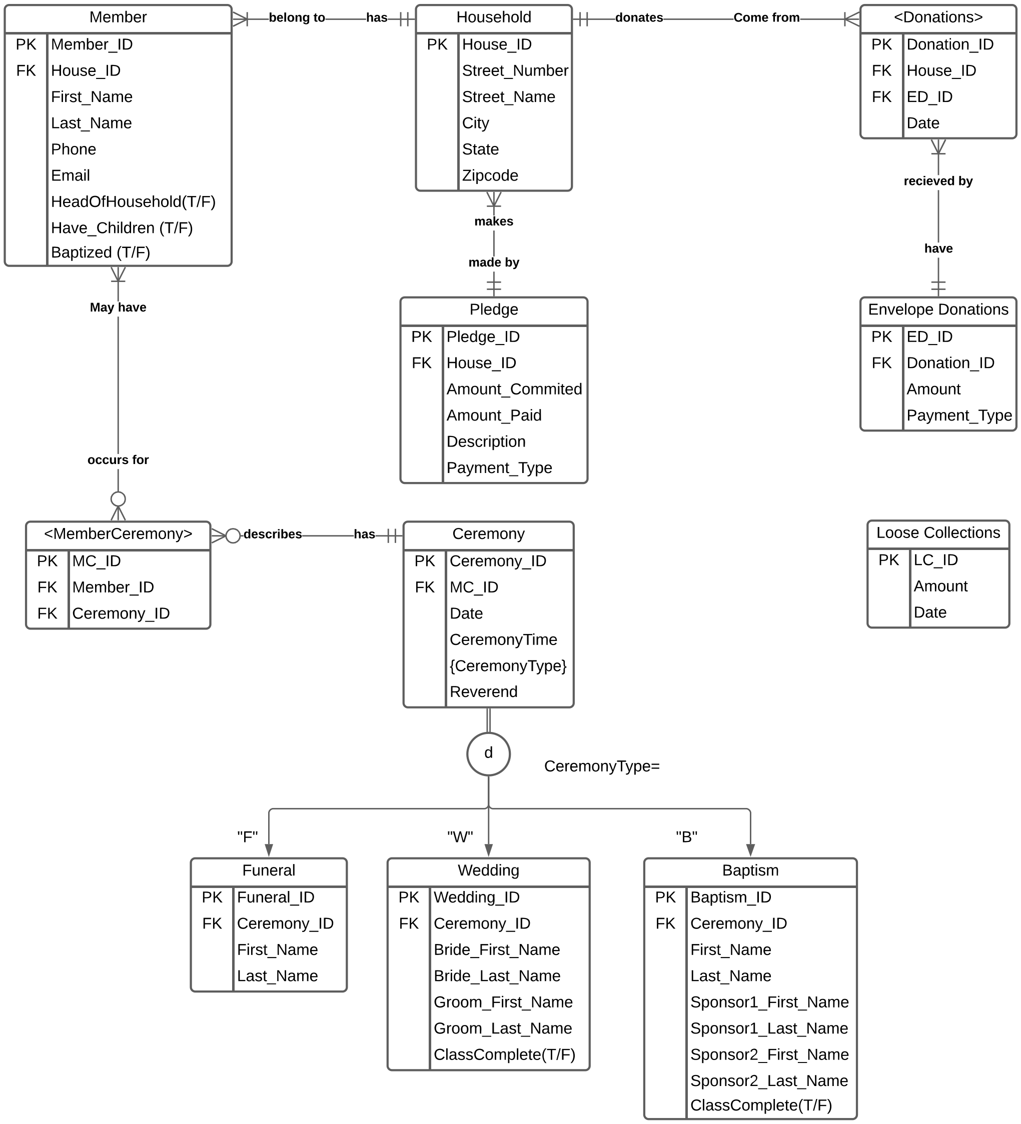
III. Problems Encountered

As all projects go, we did run into some issues when designing this system. Starting off, the amount of team members we had was below average. On top of that, our third member never participated in the class, so we ended up having half the number of members as other teams. This meant that we would have to do twice the amount of work in the same amount of time. Because of this, the scope of our project was reduced. Nonetheless, we still lacked the resource of having multiple opinions whenever we ran into problems. In the end, we made it work and even discovered that communications went much smoother whenever less people were involved.

Another problem we had was making our ERD and DFD align. This was much harder than we had anticipated. Issues related to this and resolving them is most likely what we spent most of our time on. Additionally, we did not notice many of these problems until the last stretch of our project, meaning we would have to adjust a large majority of our system. Our biggest issue was deciding how to incorporate the completion of a course into our "Organize ceremony" process and its lower-level diagram. Additionally, it was identified later in the project that it would be beneficial to have a "Campaign" entity/data store to account for future campaigns that the church could have. However, to make this modification, we would have had to rework a lot of our diagrams.

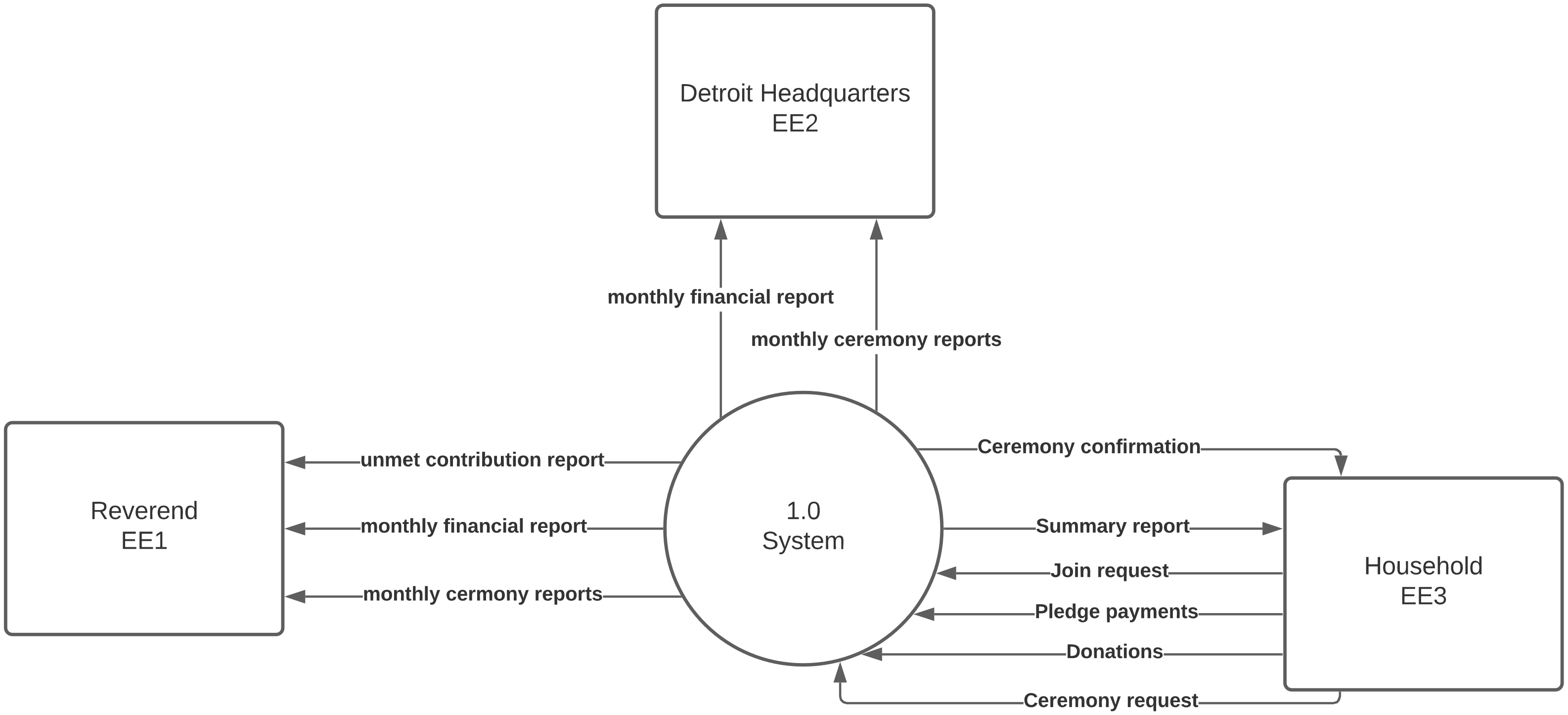
ERD

The Entity Relationship Diagram models how different entities or objects relate to one another. Each entity depicted in the model represents the data that will be collected and stored in the system.



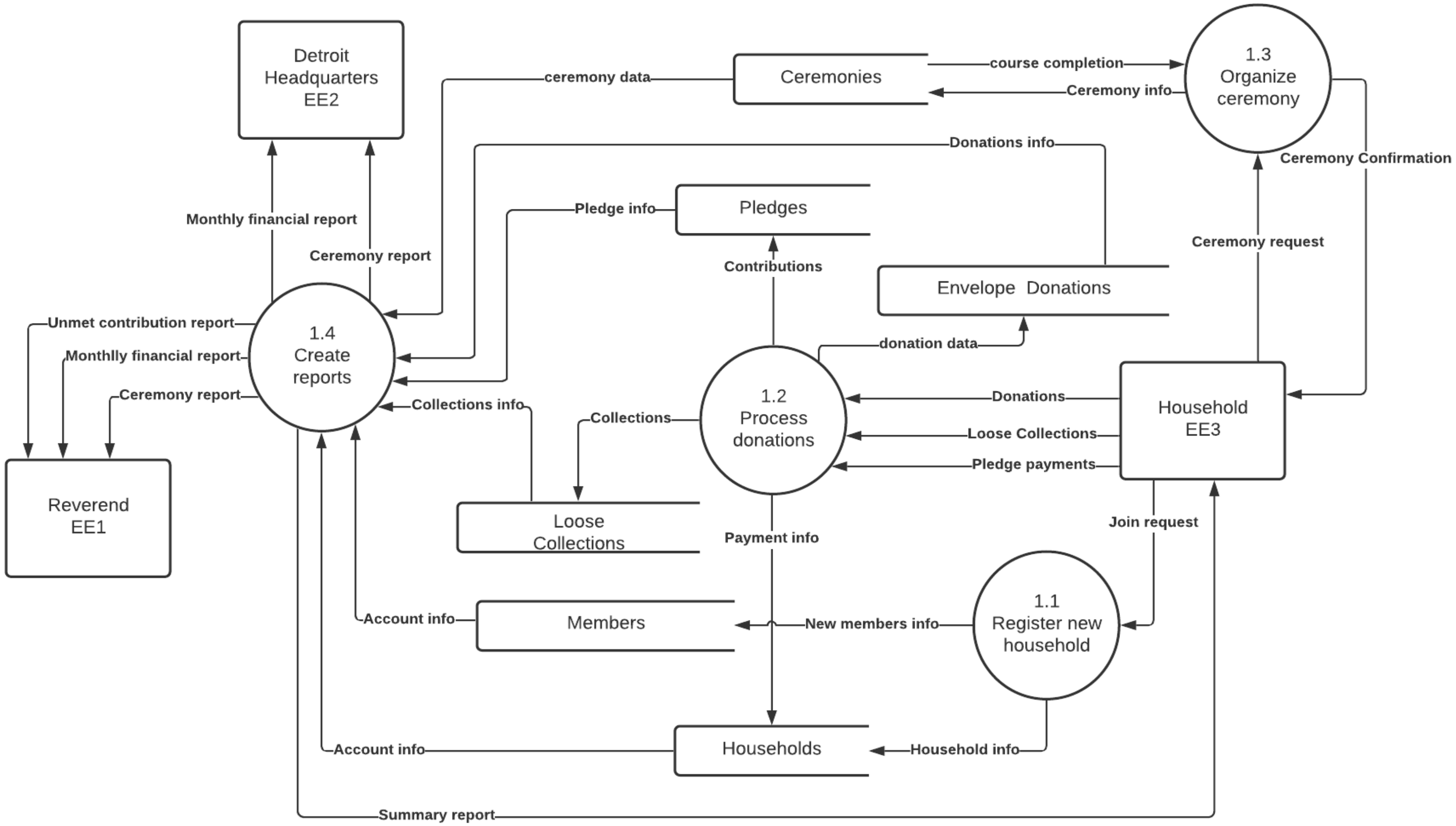
Context Diagram

The purpose of a context diagram is to delineate the system boundary and show the interactions the system has with external entities. The context diagram is useful in helping understand the context which the system will be part of.



Systems Diagram

The systems diagram is a high-level description of the system and is used to visually express the components of a process and its interactions.



Lower-Level Diagrams

Lower-level diagrams are a representation of an exploded process that provide more detail of a process compared to the Systems Diagram.

Diagram 1.2

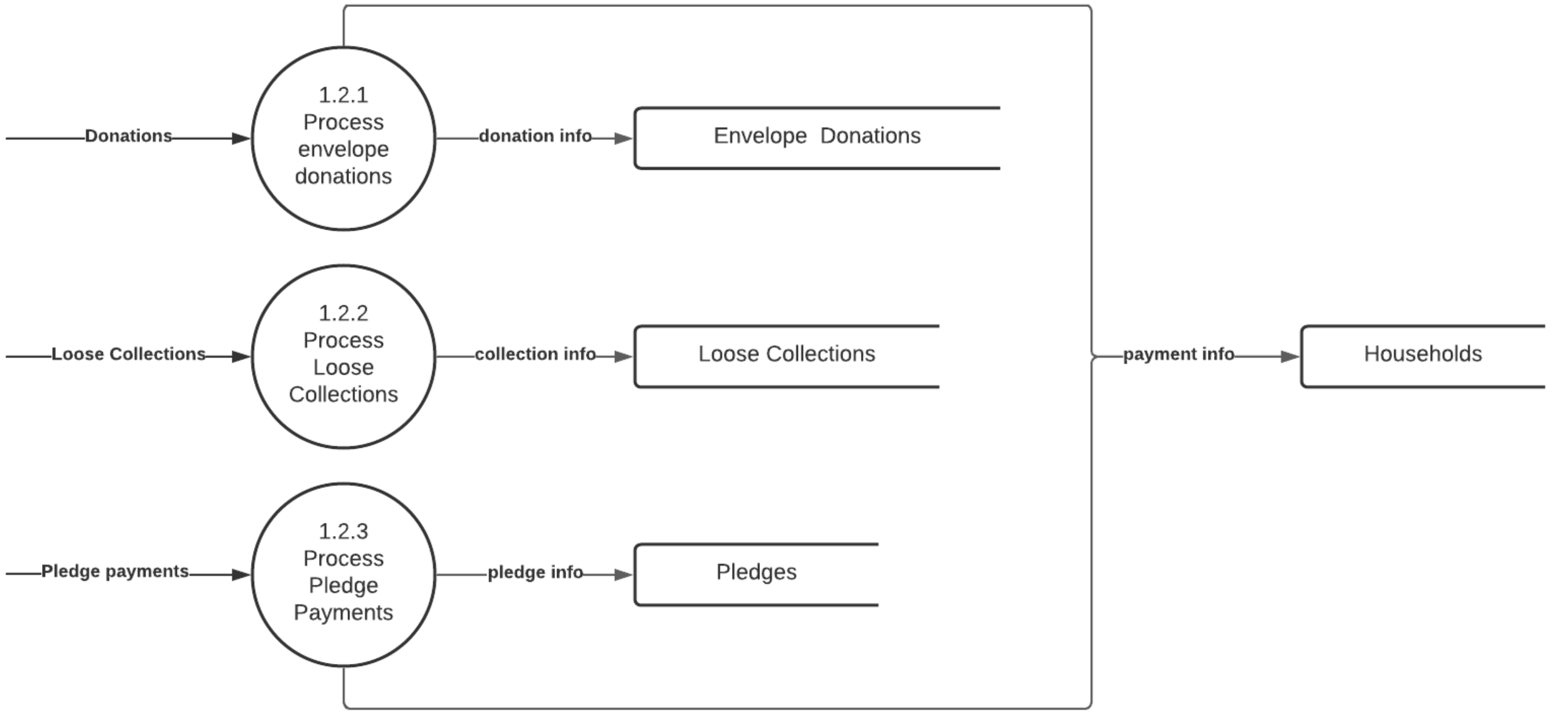


Diagram 1.3

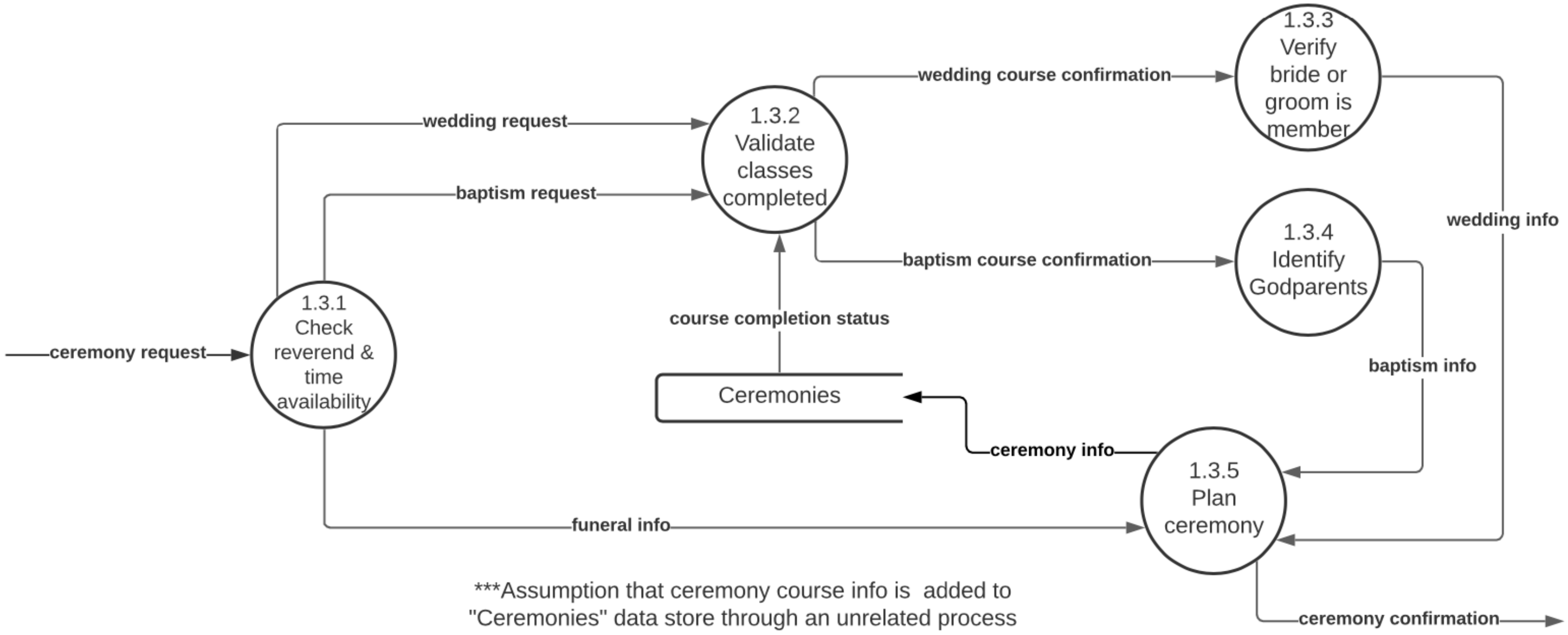
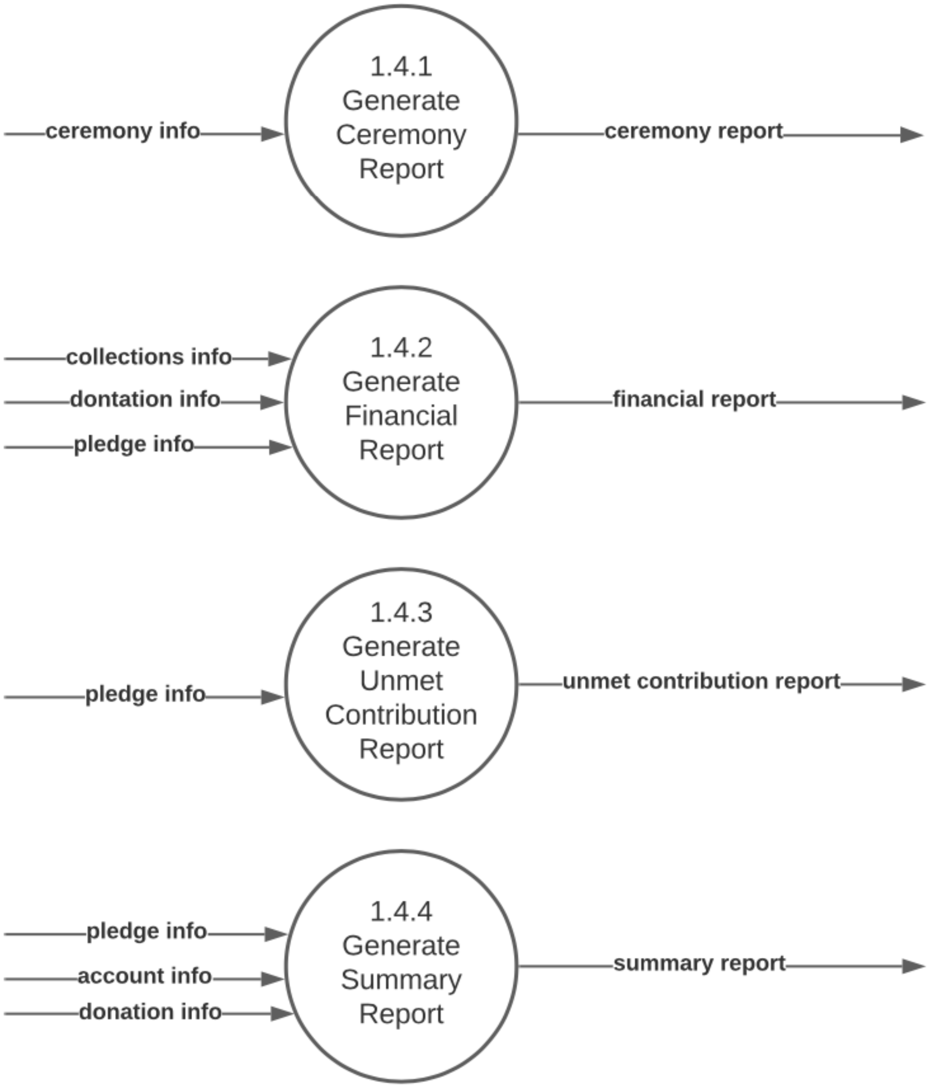

1.3.3
Verify
bride or
groom is
member
wedding course confirmation
wedding request
1.3.2
Validate
classes
completed
baptism request
wedding info
1.3.4
Identify
Godparents
baptism course confirmation
1.3.1
Check
reverend &
time
availability
course completion status
ceremony request
baptism info
Ceremonies
ceremony info
1.3.5
Plan
ceremony
funeral info
***Assumption that ceremony course info is added to
"Ceremonies" data store through an unrelated process
ceremony confirmation

Diagram 1.4


ceremony info
1.4.1
Generate
Ceremony
Report
ceremony report
collections info
dontation info
pledge info
1.4.2
Generate
Financial
Report
financial report
pledge info
1.4.3
Generate
Unmet
Contribution
Report
unmet contribution report
pledge info
account info
donation info
1.4.4
Generate
Summary
Report
summary report

Data Flows

Monthly_financial_report =	Collections_total + Date + Donations_total + Grand_total + Pledges_total +
Collections_total =	^{4} {SUM(Total_weekly_collections)}
Donations_total =	^{4} {SUM(Total_weekly_donations)}
Grand_total =	^{4} {SUM(Total_weekly_contributions)}
Pledges_total =	^{4} {SUM(Total_weekly_pledges)}
Total_weekly_collections =	SUM(collections_amount)
Total_weekly_contributions =	Total_weekly_collections + Total_weekly_donations + Total_weekly_pledges
Total_weekly_donations =	SUM(donations_amount)
Total_weekly_pledges =	SUM(pledges_amount)

Unmet_contribution_report = Date +
Unpaid_pledges +
Total_unmet_pledge_contributions

Amount_owed = Pledge_amount – Paid_amount

Pledge_info = First_name +
Last_name +
Pledge_amount +
Amount_paid +
Pledge_duration

Total_unmet_pledge_contributions = SUM (Amount_owed)

Unpaid_pledges = {Pledge_info}

Summary_report = Date +
First_name +
Last_name +
{SUM(amount)}

Ceremony_request =

Ceremony_date +
Ceremony_time +
Ceremony_participants +
Reverend +
Ceremony_type = [funeral|baptism|wedding]

Class_completion = [T | F | n/a]

Ceremony_participants = SUM(Name)

Name = First_Name + Last_Name

Reverend = Name

Pledge_payment =

Date +
House_ID +
Amount +
Payment_Type +
Pledge_ID

Metadata

The metadata/data dictionary is an organized listing of all the data elements belonging to the system, with definitions of each attribute so that users will have a common understanding of all inputs and outputs.

Table	Element Label	Alias	Description	Mode	Length	Decimal Places	PIC	Notation Coding	Units	Range of Values or Codes
Member	Member_ID	Member_Number	Unique identification for members	N	6	NA	9(6)	NA	NA	NA
	House_ID	Household ID	Unique identification for household	N	6	NA	9(6)	NA	NA	NA
	First_Name		Member's first name	AN	30	NA	C(30)	NA	NA	NA
	Last_Name		Member's last name	AN	30	NA	C(30)	NA	NA	NA
	Phone	Phone_Number	Members's phone number	AN	12	NA	9(3)"-"9(3)"-"9(4)	NA	NA	NA
	Email	Email_Address	Member's email address	AN	80	NA	C(80)	NA	NA	NA
	Have_Children		Does the member have children?	AN	1	NA	C(1)	NA	NA	T=True F=False
	Baptized		Has the member been baptized?	AN	1	NA	C(1)	NA	NA	T=True F=False
	Head_Of_Household		Is this member the head of the household?	AN	1	NA	C(1)	NA	NA	T=True F=False
Household	House_ID		Unique identification for household	N	6	NA	9(6)	NA	NA	NA
	Street_Number		Household street number	N	15	NA	9(10)	NA	NA	NA
	Street_Name		Household street name	AN	30	NA	C(30)	NA	NA	NA
	City		Household city	AN	30	NA	C(30)	NA	NA	NA
	State		Household state	AN	2	NA	C(2)	NA	NA	NA
	Zip		Household zipcode	N	5	NA	9(5)	NA	NA	NA
Donations	Donation_ID		Unique identifaction for a donation	N	6	NA	9(6)	NA	NA	NA
	House_ID		Household that the donation belongs to	N	6	NA	9(6)	NA	NA	NA
	ED_ID		The ID of the envelope donation	N	6	NA	9(6)	NA	NA	NA
	Date		Date the donation was made	AN	8	NA	9(2)"/"9(2)"/"9(2)	MM/DD/YY	NA	NA
Pledge	Pledge_ID		Unique identification for each pledge commited	N	6	NA	9(6)	NA	NA	NA
	House_ID		The household the pledge belongs to	N	6	NA	9(6)	NA	NA	NA
	Amount_Commited		The $ amount commited by the household for the pledge	N	8	2	$$$9(2) v 9(2)	$	Dollars	1 - 9999.99
	Amount_Paid		The $ amount paid by the household	N	8	2	$$$9(2) v 9(2)	$	Dollars	1 - 9999.99
	Description		Description of the pledge	AN	150	NA	C(150)	NA	NA	NA
	Payment_Type		Payment method for the pledge	AN	2	NA	C(2)	NA	NA	NA
Envelope Donations	ED_ID		Unique identification for each envelope donation	N	6	NA	9(6)	NA	NA	NA
	Donation_ID		Unique Identification for a donation	N	6	NA	9(6)	NA	NA	NA
	Amount		The $ amount of the donation	N	8	2	$$$9(2) v 9(2)	$	Dollars	1 - 9999.99
	Payment_Type	Payment_Method	Payment method for the donation	AN	2	NA	C(2)	NA	NA	C=Cash Ch=Check
Loose Collections	LC_ID		Unique identification for each loose collection	N	6	NA	9(6)	NA	NA	NA
	Amount		$ amount of the loose collection	N	8	2	$$$9(2) v 9(2)	$	Dollars	1 - 9999.99
	Date		The date the loose collection is made	AN	8	NA	9(2)"/"9(2)"/"9(2)	MM/DD/YY	NA	NA
MemberCeremony	MC_ID		Unique identification for each member's ceremony	N	6	NA	9(6)	NA	NA	NA
	Member_ID		Unique identification of the member	N	6	NA	9(6)	NA	NA	NA
	Ceremony_ID		Unique identification of the ceremony	N	6	NA	9(6)	NA	NA	NA
Ceremony	Ceremony_ID		Unique identification of each ceremony	N	6	NA	9(6)	NA	NA	NA
	MC_ID		Unique identification of the member's ceremony	N	6	NA	9(6)	NA	NA	NA
	Date		The date the ceremony is to take place	AN	8	NA	9(2)"/"9(2)"/"9(2)	MM/DD/YY	NA	NA
	CeremonyTime		The time of the ceremony	AN	7	NA	9(2)":"9(2) C(2)	HH:MM AM/PM	NA	NA
	CeremonyType		The type of ceremony	AN	1	NA	C(1)	NA	NA	W=Wedding B=Baptism F=Funeral
	Reverend	Pastor	The name of the requested reverend	AN	30	NA	C(30)	NA	NA	NA
Funeral	Funeral_ID		Unique identification of a funeral	N	6	NA	9(6)	NA	NA	NA
	Ceremony_ID		Unique identification of the ceremony	N	6	NA	9(6)	NA	NA	NA
	First_Name		The first name of the deceased	AN	30	NA	C(30)	NA	NA	NA
	Last_Name		The last name of the deceased	AN	30	NA	C(30)	NA	NA	NA
Wedding	Wedding_ID		Unique identification of a wedding	N	6	NA	9(6)	NA	NA	NA
	Ceremony_ID		Unique identification of the ceremony	N	6	NA	9(6)	NA	NA	NA
	Bride_First_Name		The first name of the bride	AN	30	NA	C(30)	NA	NA	NA
	Bride_Last_Name		The last name of the bride	AN	30	NA	C(30)	NA	NA	NA

Process Specifications

Process 1.2.1 - Process envelope donations

```
FOR EACH donation
    RECORD last name, first name
    RECORD amount
    RECORD payment type
    RECORD date
    item_record = last name + first name + amount + payment type + date
    STORE item_record in envelope donations
    STORE item_record in households
END FOR
```

Process 1.3.3 - Verify bride or groom is member

```
FOR EACH wedding course confirmation
    IF bride or groom IS NOT member
        THEN member status = invalid
        SEND invalid member status
    ELSE IF bride or groom IS member,
        THEN member status = valid
        SEND wedding info to ceremony planner
    END IF
END FOR
```

Process 1.3.5 - Plan Ceremony

```
FOR EACH ceremony_request
    RECEIVE baptism info OR wedding info OR funeral info
    ceremony_details = ceremony_type, date, time,
                       participant(s)_name, reverend
    POST ceremony_details to Ceremonies
    CREATE ceremony confirmation
    SEND ceremony confirmation to Household
END FOR
```

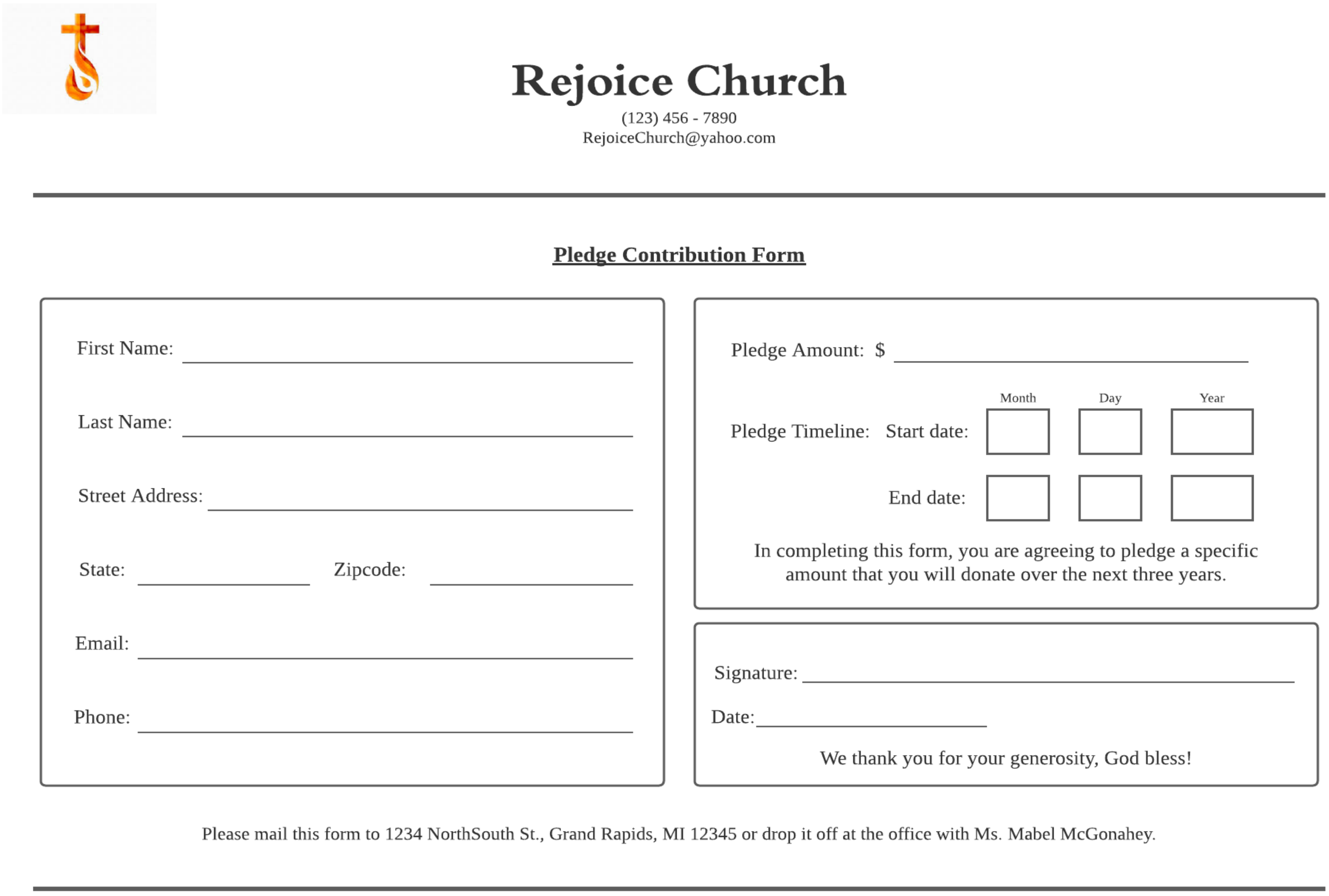
Process 1.4.2 - Generate Financial Report

```
FOR EACH month IN accounting period
        RECEIVE collections_info, donation_info, pledge_info
        collections_info = collections_amount + collections_type + collections_date
        donation_info = donations_amount + donation_type + donations_date +
                                  household_name
        plege_info = pledge_amount + pledge_type + pledge_date +
                              houdehold_name
           FOR EACH week IN month
                Total_weekly_collections = SUM(collections_amount)
                Total_weekly_donations = SUM(donations_amount)
                Total_weekly_pledges = SUM(pledge_amount)
                Total_weekly_contributions = Total_weekly_collections +
                                                               Total_weekly_donations +
                                                               Total_weekly_pledges
           END FOR
        Collections_total = SUM (Total_weekly_collections)
        Donations_total = SUM (Total_weekly_donations)
        Pledges_total = SUM (Total_weekly_pledges)
        Grand_total = SUM (Total_weekly_contributions)
    REPORT Total_weekly_collections, Total_weekly_donations,
               Total_weekly_pledges, Total_weekly_contributions, Collections_total,
               Donations_total, Pledges_total, Grand_total
END FOR
```

Process 1.4.3 - Generate Unmet Contribution Report

```
FOR EACH year IN capital campaign
   RECEIVE pledge info
         FOR EACH pledge
                  Pledge_info = first_name, last_name, pledge_amount, paid_amount,
                                      pledge_duration
                  IF paid_amount < (pledge_amount * 70%)
                           POST Pledge_info to Unpaid_pledges
                  END IF
                  amount_owed = pledge_amount - paid_amount
         END FOR
         Total_unmet_contributions = SUM (amount_owed)
   REPORT Unpaid_pledges, date, Total_unmet_pledge_contributions
END FOR
```

Input Design



Rejoice Church

(123) 456 - 7890
RejoiceChurch@yahoo.com

Pledge Contribution Form

First Name: ______________________________

Last Name: ______________________________

Street Address: ______________________________

State: ____________ Zipcode: ______________

Email: ______________________________

Phone: ______________________________

Pledge Amount: $ ________________________

		Month	Day	Year
Pledge Timeline:	Start date:	□	□	□
	End date:	□	□	□

In completing this form, you are agreeing to pledge a specific amount that you will donate over the next three years.

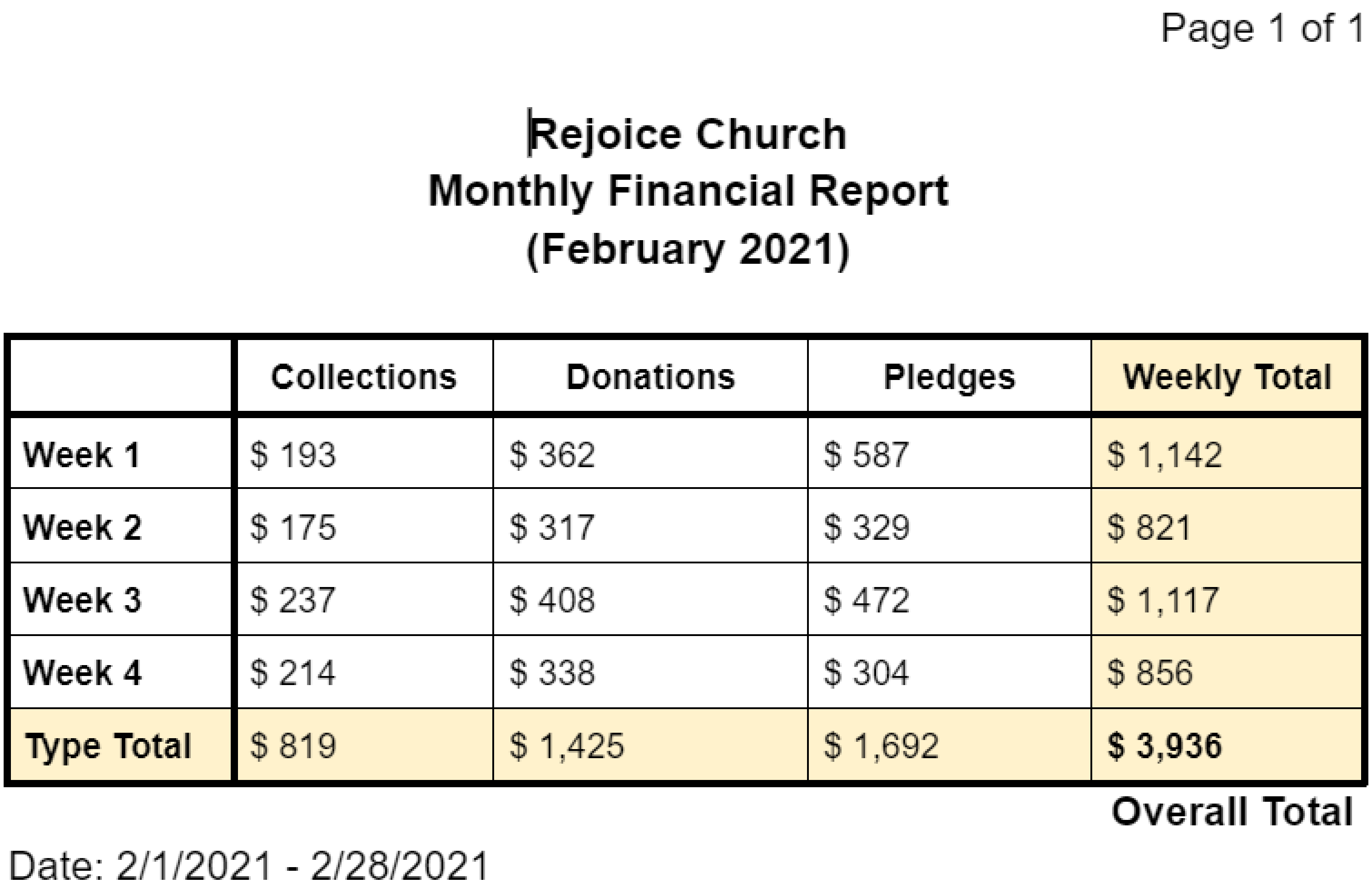
Signature: ______________________________

Date: ________________

We thank you for your generosity, God bless!

Please mail this form to 1234 NorthSouth St., Grand Rapids, MI 12345 or drop it off at the office with Ms. Mabel McGonahey.

Output Design



Page 1 of 1

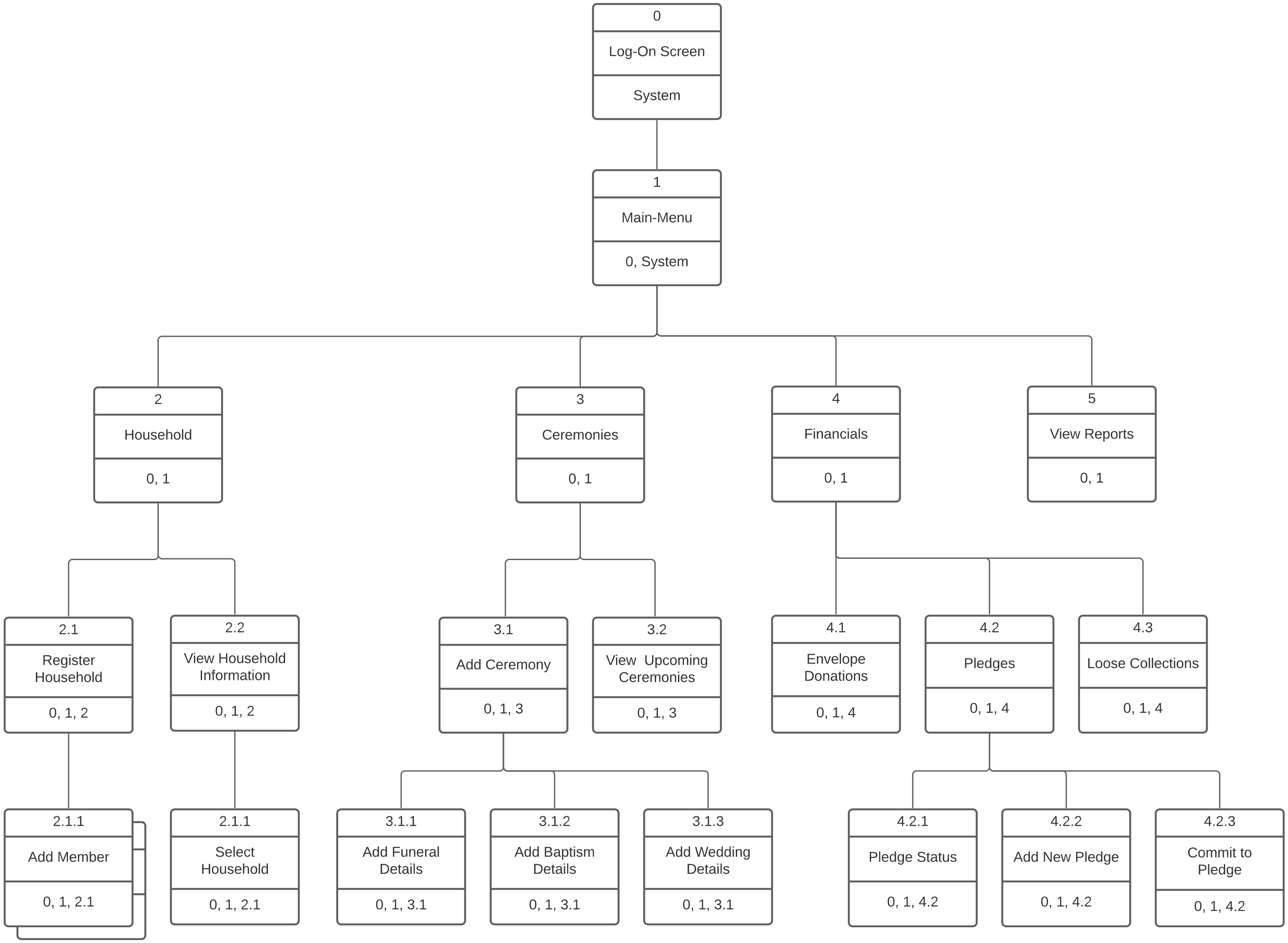
Rejoice Church
Monthly Financial Report
(February 2021)

	Collections	Donations	Pledges	Weekly Total
Week 1	$ 193	$ 362	$ 587	$ 1,142
Week 2	$ 175	$ 317	$ 329	$ 821
Week 3	$ 237	$ 408	$ 472	$ 1,117
Week 4	$ 214	$ 338	$ 304	$ 856
Type Total	$ 819	$ 1,425	$ 1,692	**$ 3,936**

Overall Total

Date: 2/1/2021 - 2/28/2021

Dialog Diagram

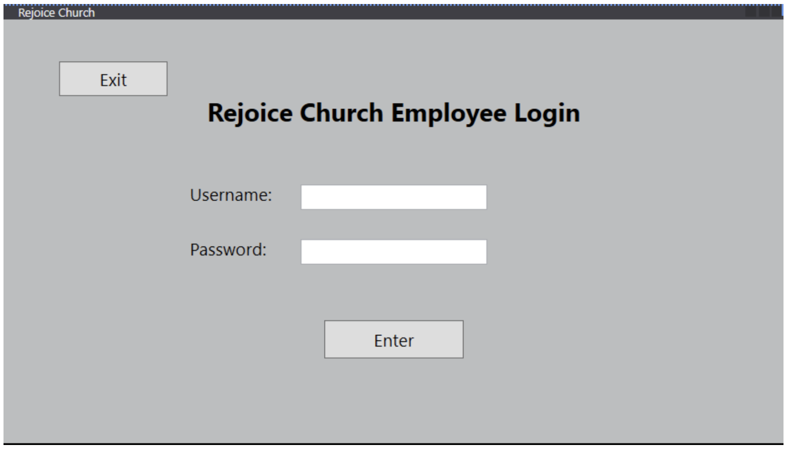


Prototype

To develop the prototype for the system, we created a Microsoft WPF application and deployed it to Microsoft Azure for easy access and download. The prototype provides an understanding of how the end-user will interact with the system.

Download Instructions (Time to download = <4 min):

1. Visit the website: **rejoicechurch.azurewebsites.net**
2. Download the required framework: .NET Desktop Runtime 5.0.12 (if not already installed), by clicking the "Install" button. If the framework has already been installed click the hyperlink "launch" above the Install button.
    a. NOTE: The application has been **SAFELY** deployed to Azure using a "ClickOnce" deployment through Visual Studio and there is **<u>NO malicious content</u>** so proceed through the warning messages.
3. After downloading and proceeding through the warning messages, you should now have access to the application!



Rejoice Church
Exit
Rejoice Church Employee Login
Username:
Password:
Enter