Software Specification

for

Campus Ministry Software

Version 1.0 approved

Prepared by Zahara Kazmi

Loyola University Maryland

May 2nd, 2017

Table of Contents

Table of Figures

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

The Software Specification document is produced to accompany the software and it includes the development process and documentation for Campus Ministry Software. This document will provide details for how the software is developed, how it has been tested and how to use the software. The document will be a graphical and narrative description of the software design as well as a user manual and reflection for this project.

1.1 Purpose

The purpose of the document is to specify the development for Campus Ministry Database Software. This project will include a Web interface which will be connected to a database as well as a Machine Learning Algorithm for Event Planning. This project will expand over a period of semester and it must be completed before the end of the semester with given mentioned requirements.

1.2 Document Conventions

This document is divided into different parts and the details of the project get more specific as you move further into the document. Each new section is labeled with larger font and new number. Furthermore, the priority of each task will be defined in the order the tasks are listed. All figures included in the document show projection completion only up to this point.

1.3 Intended Audience and Reading Suggestions

This document is intended to for developer, project manager, documentation writer, primary users, and administration. This documentation contains estimation of task management, requirement breakdown, system features and any additional requirements. The Overall description of the Product is important for primary users and developer because it will outline the limitations and requirements of this software. Project Estimation is intended for the administrator and Project Manager to confirm and provide a time estimation by when this project should be completed. External Interface requirements are intended for Administrator as they will be responsible for managing the system once it is deployed. The document will give a breakdown of the project followed by a breakdown and detail of the requirements. The breakdown will provide overview of project breakdown thus reading that beforehand will give you an overview followed by details. The document should be read in the order that follows as it will become more and more complex as the document progresses.

1.4 Product Scope

The software being developed will serve several different purposes and needs for campus ministry. The database will offer a single access point of storage and access of data for the staff. Information collected at various campus events will all be accumulated and will be saved into database as entered to be later queried by Staff as needed. The web interface will also allow staff and interns to login to add hours they have spent working during the week and what projects they are working on. The Web interface will also have guest pages only to be used at events and by people who are only providing the information. They will simply enter their information and submit it so it can be saved in the database. This step will make the usability and lifetime of the software because data will not need to be entered manually but instead will accumulate from the

forms. The Machine Learning Algorithm will run on the database as requested by user to predict a time of day and day of the week to hold an event for specified attendance. This will not only allow for the flow of the office and work to improved but also for campus ministry to serve the campus needs better.

1.5 Software context

This Software will enable Campus Ministry to store and transform data into information to support work flow. This software will replace previous work flow of using a file system. This software will allow staff and interns to have concurrent access to the database as well to guest to enter information. The database structure will improve integrity of the data as well providing a single location access point for all staff and guest members from any location. Data redundancy and inconsistency would be removed and data will come directly from source instead being manually entered by interns. It will increase the security and reliability of the office in collected information. Improvement in the various qualities will improve data quality as well thus resulting in improved and faster work flow.

Another aspect of this software is the Machine Learning Algorithm which will be providing suggestions for planning future events based on information from previous event. This will allow interns to have targeted audience to improve quality of the events.

1.6 Major constraints

Machine Learning Algorithm implementation of this software does not have data to be trained on and tested on. The office has already combined their collected data into one file and lost data for over the years. Thus, the algorithm needs to be confirmed via pseudo-data or use unsupervised learning algorithm. The Algorithm will be confirmed based on accuracy for the specified algorithm and its functionality to work with the different features.

1.7 References

In order to develop this document several different online resources are being implemented for estimation and timeline. Trello is being used to keep track of tasks and hours in a burn down format. TeamGnatt is being used to keep track of project in a Gantt chart and is included in the Project Estimation Section under timeline. A repository will exist on GitHub for all the code as well as all the documentation. The links for all these references is included in the Appendix section.

2. Software Requirement Specification

This section provides a detailed description of all the data structure which will be implemented within this software. The software includes a Database and the users will be entering information via web interface thus we have several different structures where data is being passed from one point to another. The data is being stored as values pairs. When the user inputs data and hits submit, it was read in using the name assigned to field and stored into database in its specified position

2.1 Project Estimate, Schedule, and Risks

2.1.1 Software Development Process Used

The software development process model which will be implemented for this project will be the Waterfall model. This project does not require the need to reevaluate requirements nor the need to repeat a step once we have developed in the process thus waterfall is the best model for it. In following the Pure Waterfall Process model, we have the strength of reducing our planning overhead as we would have already completed the documentation. This also minimizes wasted effort as we have already confirmed and documented the exact process and content of what we are building. Although this model does not provide flexibility but as this is something which would not be usable until it is completed with all the requirements thus we have no need for flexibility. Our only non-documentation deliverable is once the project is completed done and we have tested it as well. Waterfall will allow us to create a product with clearly understood requirements thus making a more effective software.

2.1.2 Project Task Set (Statement of Work or SOW)

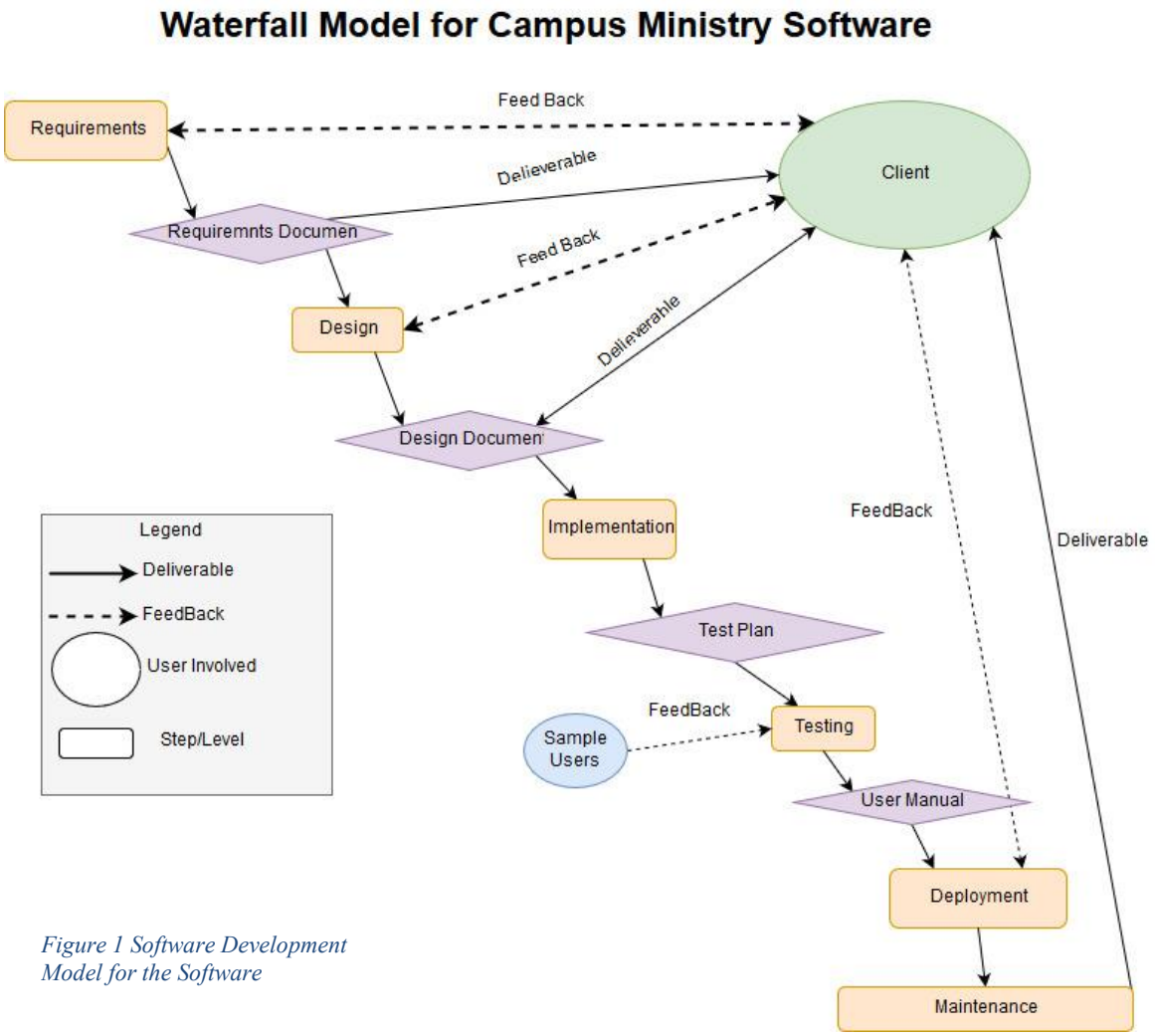

Figure 1 Software Development Model for the Software

The Statement of Work offers a broad overview and breakdown of project tasks. All the tasks have been divided into Waterfall levels and within each level we have list of tasks. The start date is estimated start date based on due date from when the previous task should be completed. The due date of the project indicates the date by which this specific task is estimated be completed. These dates indicate that these tasks should be completed either by this date or within that week. These dates allow us to estimate project tasks and work break down. The percentage represents number of hours spent on this task in comparison to all the tasks within the project. The due date for document is represented as a yellow diamond and it also has a check mark once the document is completed and has been submitted.

Campus Ministry Software

▼ Requirements	2%		Start	Due
Meet with Client to Discuss needs	2%		Jan 19, 2017	Jan 25, 2017
Come up with Requirements	2%		Jan 26, 2017	Feb 1, 2017
Reevaluate requirements Docuements	3%		Feb 2, 2017	Feb 8, 2017
Create formal SRS	6%		Feb 2, 2017	Feb 8, 2017
Show to Client and Confirm	0%		Feb 2, 2017	Feb 8, 2017
Present SRS	0%		Feb 2, 2017	Feb 8, 2017
SRS Document DUE	☐	◆	Yesterday	Yesterday

Task | Milestone | Group of Tasks

▼ Design	0%		Start	Due
Get Data from Client	0%		Today	Feb 24, 2017
Design Structure for databse	0%		Feb 23, 2017	Feb 28, 2017
Machine Learning Algorithm Research	0%		Feb 17, 2017	Feb 19, 2017
Design Machine Learning	0%		Feb 19, 2017	Feb 21, 2017
Web Interface style/Design Research	0%		Feb 17, 2017	Feb 21, 2017
Meet with Client and confirm design solution	0%		Feb 27, 2017	Feb 28, 2017
Create Design document	0%		Feb 24, 2017	Mar 3, 2017
Present Design Document	0%		Mar 13, 2017	Mar 15, 2017
DESIGN DOCUMENT DUE	☐	◆	Mar 2, 2017	Mar 2, 2017

Implementation	**0%**		**Start**	**Due**
Set up server and software on Campus Min Network	0%		Mar 14, 2017	Mar 16, 2017
Create Database	**0%**			
Create Database Schema	0%		Mar 15, 2017	Mar 18, 2017
Define Relationships	0%		Mar 15, 2017	Mar 18, 2017
Import data into database	**0%**			
Create Connected Driver	0%		Mar 15, 2017	Mar 18, 2017
Create Insert Into for Each Table	0%		Mar 15, 2017	Mar 18, 2017
Create Class to Allow CSV Loading	0%		Mar 15, 2017	Mar 18, 2017
Create web application/site connected	**0%**			
Log in Page	0%		Today	Mar 27, 2017
Home Page to User	0%		Today	Mar 27, 2017
New Project Page	0%		Today	Mar 27, 2017
Project progress and Update Page	0%		Today	Mar 27, 2017
Supervisor Page	0%		Today	Mar 27, 2017
Weekly View Page	0%		Today	Mar 27, 2017
Query Page	0%		Today	Mar 27, 2017
Choir Guest Page	0%		Today	Mar 27, 2017
Wedding Guest User Page	0%		Today	Mar 27, 2017
Program Attendee Page	0%		Today	Mar 27, 2017
Retreats Guest Page	0%		Today	Mar 27, 2017
Connect to Database	0%		Today	Mar 27, 2017
Create Machine Learning Algorithm	**0%**			
Training Machine with Algorithm	0%		Apr 2, 2017	Apr 5, 2017
Run Validation and Testing	0%		Apr 15, 2017	Apr 17, 2017
Confirm Algorithm	0%		Apr 17, 2017	Apr 17, 2017
Run on Query from Database	0%		Apr 17, 2017	Apr 17, 2017
Add Algorithm to Web Interface	0%		Apr 17, 2017	Apr 17, 2017
Write all documentation and show client final result	0%		Apr 18, 2017	Apr 21, 2017

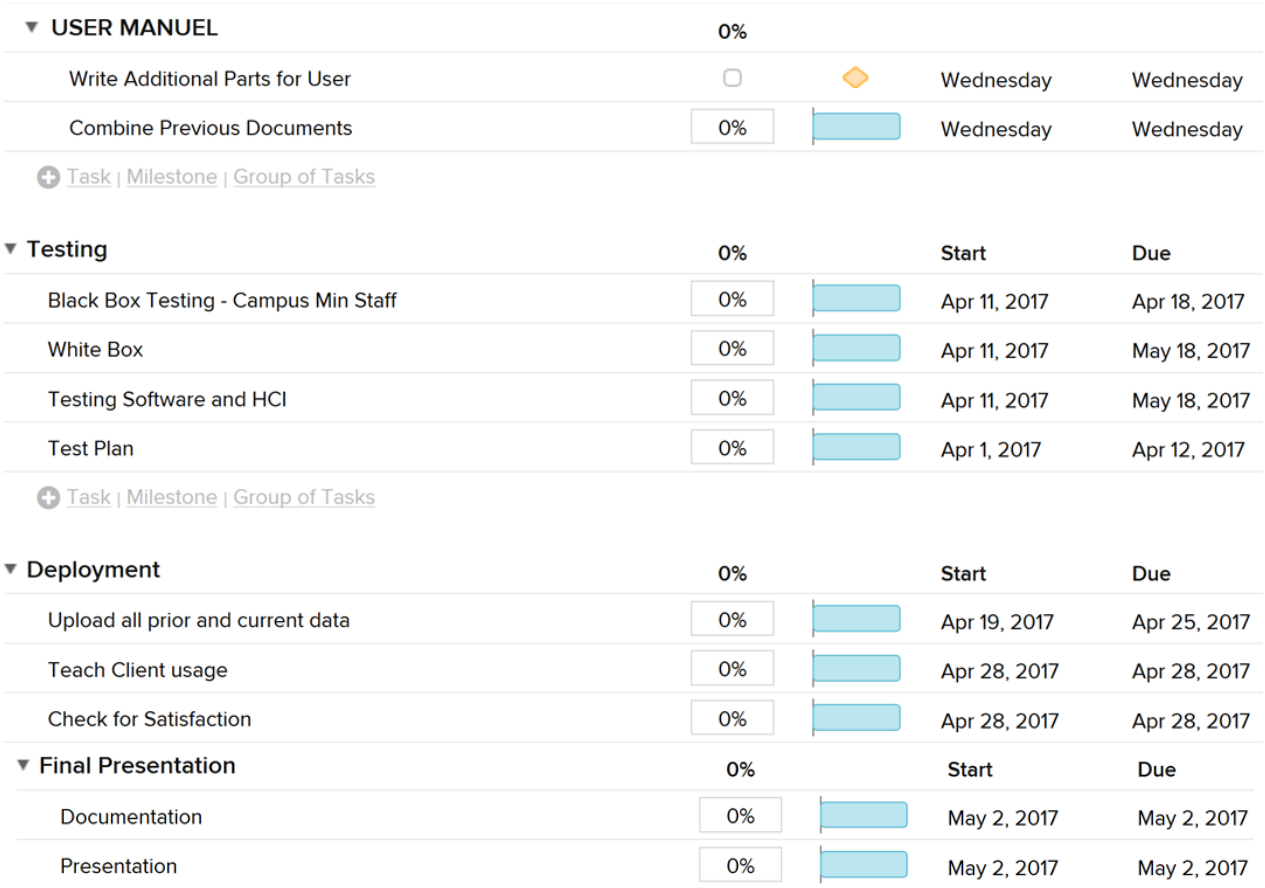

Figure 2 Statement of Work

2.1.3 Estimation Techniques Applied and Results

To estimate the cost of this project, we will be looking at resource costing and estimating how much the software and hardware for this project will cost. Other than resources we will not be accumulating any other costs.

Resource List

Item Name		Type	Cost
MySQL		Software	$0.00
Apache Server		Software	$0.00
Computer with Storage		Hardware	$0.00 (Client has one)

For estimating the effort for this project, we will be using COCOMO. Estimating via Wideband Delphi is difficult as we would only have estimates from one person. Thus, we will run COCOMO with Sources of Lines of Code as well as with unadjusted Function Points. The first result we have is for unadjusted Function point result given our input about programming abilities, knowledge of project and software and implementation needs. The first estimation based on 20 unadjusted function points and second estimation is based on 1000 SLOC. Both estimation give us effort of one person and schedule of about 5 months.

Software Development (Elaboration and Construction)

Effort = 1.4 Person-months
Schedule = 5.3 Months
Cost = $0

Your

Total Equivalent Size = 800 SLOC

Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	0.1	0.7	0.1	$0
Elaboration	0.3	2.0	0.2	$0
Construction	1.0	3.3	0.3	$0
Transition	0.2	0.7	0.2	$0

Software Effort Distribution for RUP/MBASE (Person-Month

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	0.0	0.0	0.1	0.0
Environment/CM	0.0	0.0	0.1	0.0
Requirements	0.0	0.1	0.1	0.0
Design	0.0	0.1	0.2	0.0
Implementation	0.0	0.0	0.4	0.0
Assessment	0.0	0.0	0.2	0.0
Deployment	0.0	0.0	0.0	0.0

Figure 3 COCOMO Testing Estimation Based on Number of Unadjusted Function Points

Software Development (Elaboration and Construction)

Effort = 1.7 Person-months
Schedule = 5.7 Months
Cost = $0

Your pr

Total Equivalent Size = 1000 SLOC

Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	0.1	0.7	0.1	$0
Elaboration	0.4	2.1	0.2	$0
Construction	1.3	3.6	0.4	$0
Transition	0.2	0.7	0.3	$0

Software Effort Distribution for RUP/MBASE (Person-Months)

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	0.0	0.0	0.1	0.0
Environment/CM	0.0	0.0	0.1	0.0
Requirements	0.0	0.1	0.1	0.0
Design	0.0	0.1	0.2	0.0
Implementation	0.0	0.1	0.4	0.0
Assessment	0.0	0.0	0.3	0.0
Deployment	0.0	0.0	0.0	0.1

Figure 4 COCOMO Estimated Based on Lines Of Code

We will also be implementing a burndown chart for time estimation. Although the burndown will also help in task management but mainly it gives us a time estimation given our own knowledge and familiarity with the software and project. The chart below shows the date by which our project should be done based on estimates for all the tasks we created earlier. The estimation for each task is based on the developer's own knowledge of their work and experience of how long it should take based on previous experience.

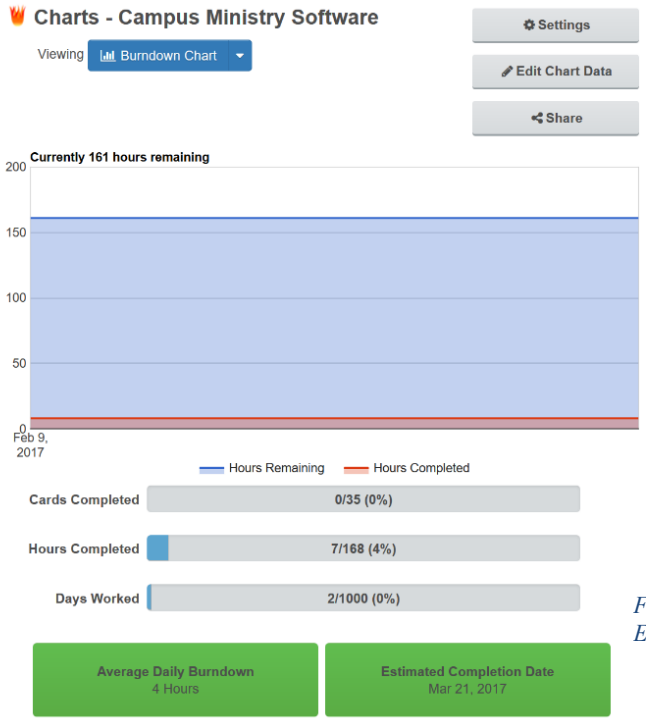

Figure 5 Trello Burn Down Chart Estimation

2.1.4 Timeline Chart

This section of the project will provide us with a timeline for the project and give us breakdown of project. We will be using a Gantt Chart for showing all the tasks and milestones in a timeline as the get completed or come up. This chart will be used throughout the project to show how much time is being spent on each task and it will also give us a percentage of how much work is completed.

Figure 6 Gantt Chart Overall Project Overview.

2.1.5 Project Risks, Risk Impact and Mitigation plan

This project is something which will be replacing the current method of usage at the campus ministry office thus we have a great risk in this project. The functionality of this project needs to be thoroughly developed and tested before it can be utilized by the office for their work. Below we have listed the risk, their priority and how to mitigate the risk. Below we have some of the major risk that can occur at any time during the project and after its completion but since we are not changing existing data or removing previous software we have no risks in terms of loss of data.

Risk	Priority	Mitigation
Database Entries not entered correctly and thus not saved in database	HIGH – very likely that due to some null inputs or not giving inputs database might not accept the entry.	Proper testing of all input values and testing to make sure database accepts any data given and it is stored properly.
Machine Learning Algorithm has low accuracy	HIGH	Try different algorithm and features to find better accuracy during training instead of getting to validation or testing.
Server Turn off	HIGH	Default settings should be to always be on and make all ports needed by server available to it – in documentation the process of fixing such an error should be included

Figure 7 Project Risk and Mitigation Plan

2.2 Overall Description of the Product

2.2.1 Product Perspective

This software in development for this project is a new self-contained product and it will make no changes to already existing work flow at campus ministry. Instead this software will work to

enhance and improve workflow by allowing access to all data via database and web interface. The diagram below shows a diagram of overall system connections and users involved.
In the above diagram, we have a guest user which is outside the system and is only interacting as naïve user. The Actors within the System are Interns and Staff. Interns interact with Databases via Interface while Staff will have access to Raw database if needed. This is just a general overview of how the different components will interact with one another.

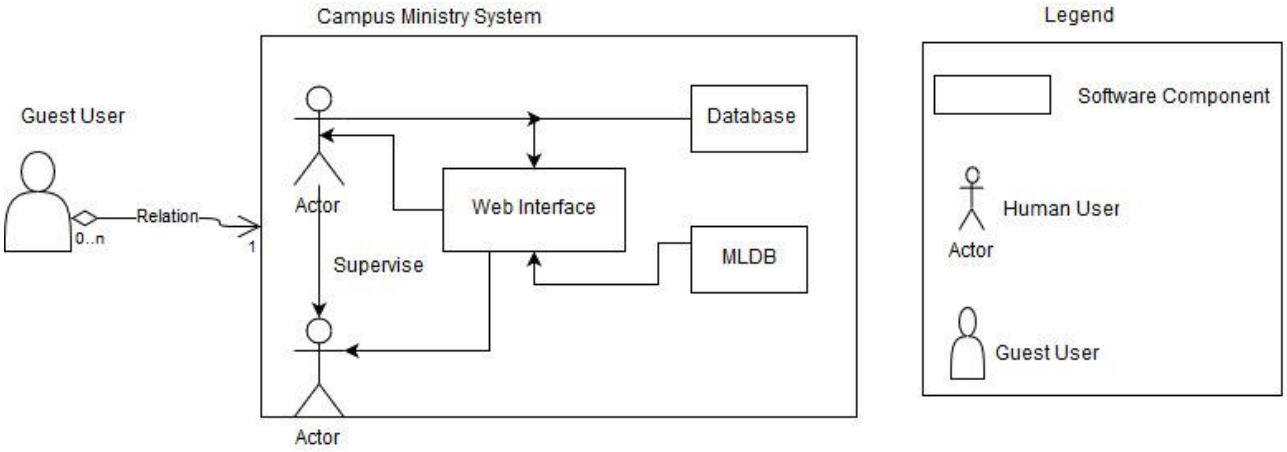

Figure 8 Product Perspective for Users Diagram

2.2.2 Product Functions

This product includes several different functionalities which need to be developed to allow user flexibility in interacting with the software. Below we have listed summary of some functions.

- Log in Network or Guest Form
- Query Page
- Weekly Schedule Overview
- Update project progress and hours worked
- Supervise interns and check hours worked
- Download query results in CSV form or Allow CSV Loading
- Ask Machine Learning Algorithm for Day and time suggestions for Program Planning

These functions represent an overview of all the functionalities this software will have and below we have a top-level flow diagram of how the web interface will accomplish these functionalities. The chart below gives an overview of all the different options available to the users and decisions they would have to make as the progress in the software. As pictured below, the guest user will only

interact with guest forms – they have no knowledge how this information is being stored. This is a very simple and broad overview of the how the system would behave with above functions.

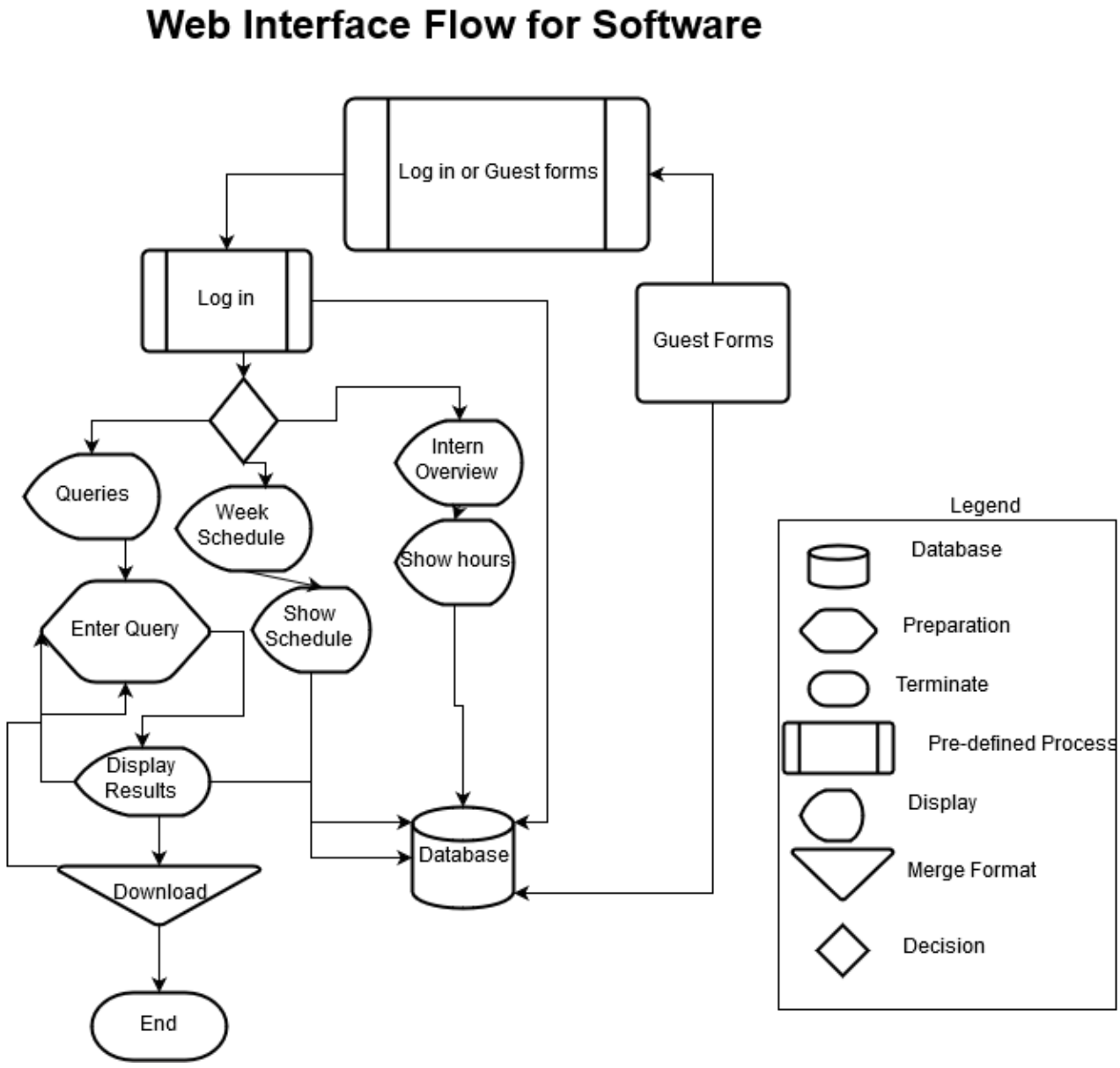

Figure 9 Product Functions Representation Flow

2.2.3 User Classes and Characteristics

For this system, we will have 3 different types of users. Each user is differentiated based on the frequency of use, subset of product function used, and privilege levels. In the table below, each of these different features are described for all the different users who will be interacting with the system.

User Classes

User Type	Frequency	Function Used	Privilege Level	Example
Guest User	Very minimum – for most once	Guest forms to submit data	None	Students at Events Bride/Groom Retreat Students Interested Students Chapel Choir
Interns	Almost every day	Weekly schedules, program and projects. Event attendance and MLDA.	User name and password required	Alex, Brian, Chris – all campus ministry interns
Staff	Every day – several times a day	Check on users, run queries, evaluate data from programs and generate reports.	Username and password required	Elise, Sean, Fr. Murray, George – all Campus Ministry Staff
Administrator	Everyday	Check on server flow and make sure everything is working	Username and password required as well as access to device	Sean – once software is deployed

Figure 10 User Classes and Interaction Chart

2.2.4 Operating Environment

The software will be used via web and will be available from anywhere on the internet. But the server and database will be stored on a device at the Campus Ministry office. The server and database will be implemented on Operating System Windows 7. This system will be developed using MySQL, PHP and MLDA.

2.2.5 Design and Implementation Constraints

This system will have few constraints in terms of the Machine Learning Algorithm, as it only has few technical software available. The Machine Learning Software we will be implementing is Machine Learning Algorithm. There are no specific language requirements set by Client and it is open to developer on what language to be used. The Client will be responsible for maintaining the delivered software thus it needs to be properly documented and explained thoroughly. As the system is not collecting any sensitive information, security is not a top priority but protocols will be created to protect the information. This system will be implemented using MySQL, WampServer64, PHP and Machine Learning Algorithm.

2.2.6 User Documentation

In part with the software, several different documentation components will be developed and delivered to the client. Some of the documentation will be very formal and designed for developers or Engineer to further enhance the system. Since the Client does not have a technical background we will be developed a very detailed and simple to read user manuals as well. Although no online-help will be created for this software but we will create a resource for all other software being used to be referenced by Client. The Client does not request any specific form of the documentation but Client will be receiving a hard copy as well as soft copy.

2.2.7 Assumptions and Dependencies

This project assumes that Campus Ministry has availability for space on their hardware for server and database. This has been briefly discussed but as the length of data is unknown thus estimating necessary space is unconfirmed. The Machine Learning Algorithm depends on having enough existing data on previous programs and events to train and test the machine. Without having prior data no learning can occur thus making our Machine useless. The software will depend on hardware being on all the time as the server it would need to be accepting requests always.

2.3 External Interface Requirements

2.3.1 User Interfaces

This system will heavily implement user interface as this is the only way all the users will interact with the software. The user interface will be implemented to be as simple as possible. One aspect to be implemented on all the pages is a help button which will explain to user all the options available to user on that page. The user will also have ability to move back to the previous page with a back and forward button. There will also be a logout button on each page for user to quit at any point. The Guest form will exact links given to users to input data and they will have no knowledge of

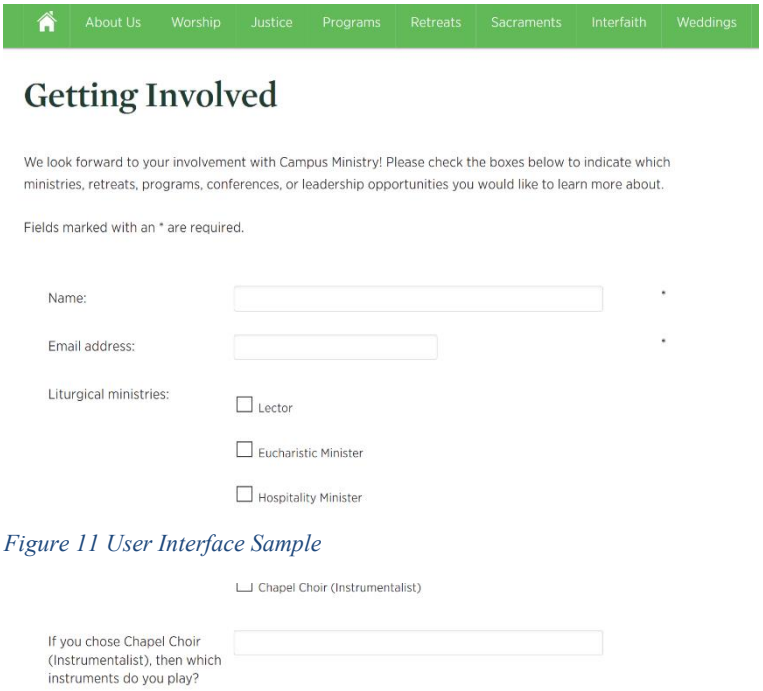

Figure 11 User Interface Sample

other forms. Below is a sample of how the user interface might look like for the Guest to fill out. This is the sample form used by Campus Ministry currently and when the form is filled out an email is sent to campus ministry email for interns to manually enter this information. The screen layout will differ but just like in the sample below there will be tabs at the top for users to easily switch from one screen to another.

2.3.2 Hardware Interfaces

The Software will be only interacting with Hardware at the Server and Database level. The implementation of the Server and Database will be only a Desktop for which the specification have not been decided. The device shall be available on all devices connected to internet.

2.3.3 Software Interfaces

This system is being developed as a database with web interface and machine learning, and this involves communication between several different components. The database implemented in MySQL will communicate with web interface to save new data, run queries and be running to update new information. The Machine Learning Algorithm will communicate with web interface when the user access the page to get a AI prediction for the event. The databases and web pages will also communicate with server into to run. Regardless of when these a specific database is being requested, all databases and servers will need to be running in case a request is made.

2.3.4 Communications Interfaces

This system will implement several different communication interfaces to process its functionality. This application will be run on web browser and will be accessible on any web browser platform. The network server must be always open for communication with the browser and any request which come through to the system. The guest users will be input data into electronic forms which will be communicated via the server to database and stored there but it will also be sent via email to Client specified email. This protocol is being followed because no error checking is implemented in this system thus this will allow interns to check the information if necessary and to confirm if the information has been entered without querying data. The database should and web interface should be synchronized always to not cause data discrepancies and to continuously update. This system will be using HTTP communication standards for all its functionality and for communication with database.

2.4 System Features

This section includes detail of the main features of the system. This section is divided into the three main features of the software which are further explained in smaller components. Each feature is divided because it is a separate implementation and will then be integrated with the system. These features are organized in the order in which it needs to be completed because each feature builds on the previous on. As the feature, will work together to create the system and fulfill the needs of Campus Ministry, the user stories are provided separately to briefly explain how the software will be used.

2.4.1 User Stories

The following User Stories briefly explain what each User of the system wants to accomplish with the system and why they need the functionality. These user stories are primarily for actors who will be using the software as validated and privilege users.

User Stories			
User Story	Feature	Need	Priority
As Elise, I want to be able to search for students with Hindu background who are interested in Interfaith to expand initiative on campus.	Database and Web Interface	can run queries for specific students without knowing MYSQL syntax	High
As Sean, I want to be able query for programs from given dates and type of program to track progress on new programming.	Database and web interface	Keep track of programming at CM	High

As Fr. Murray, I want to be able to print retreat information for specific retreat with all student info so we can confirm all student information has been submitted.	Database and Web interface	Retreat and student information in one location	High
As George, I want to be able to have choir members submit their information online instead on paper and then manually typing so we can instead program for events.	Web interface and database	Guest form for members to type in information which is saved in database so interns do not have to manually type and change information	High
As Megan, I want to be able to get list of choir member names and instrument so we can assign seats.	Web interface and Database	Run query but Choir information should have their instrument listed as well	High
As Elise, I want to be able to check when Zahara, Claire and Precious are in office and what they hours are like so I know when to contact them in office.	Web interface and database	Look at your supervisees schedule and their projects	Low
As Elise, I want to be able to see and hours worked by Claire and what projects she worked on so I can approve them.	Web interface and database	Track hours and projects online	Low
As Nancy, I want to be able to get reports from everyone about hours they worked during the 2 weeks so I can process it for payroll.	Web interface and database	Use online hours tracking for pay rather than manually type in hours to excel and then again for payroll.	Low
As Megan D., I want to be able to get a list of interested student in personal reflection so I can contact them for ignite retreat.	Web interface and database	Run queries to get list of interested students for events	High
As Claire, I want to be able to get a day and time for when to program an interfaith event so that I can have attendance of 20 people.	Machine Learning DB and Web Interface	Use MLDA to get day and time for specific attendance.	Medium
As Sean, I want to be able to get CSV copy of all entries for specific date so that I can know how many new people we received.	Database and Web interface	Have Excel copy for DB to plan next year and printed material.	Medium

Figure 12 User Stories for Overall Software

2.4.2 Database Implementation

5.2.1. Description and Priority

This feature will implement the main database for Campus Ministry. The Schema for this structure will be developed during the design process and the following list of priorities of implementation. All the steps in this process have a high priority because without setting up the database we cannot have a fully functional system. The data loading has lower priority because if we are unable to upload previous it is does not impact the functionality of the system.

5.2.2. Stimulus/Response Sequences

This feature does not have many responses as this will only be taking in data and saving it.

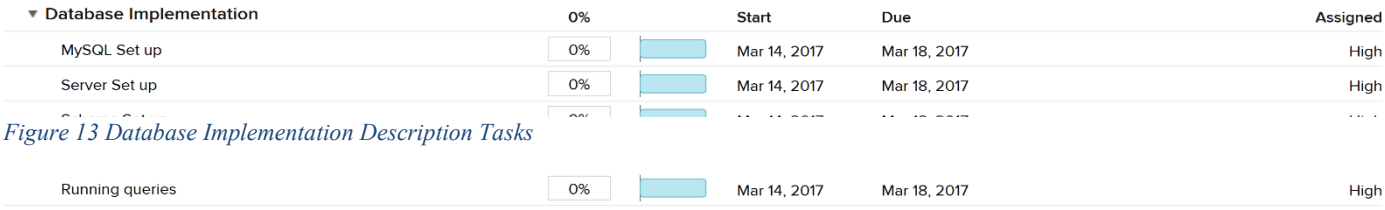

Figure 13 Database Implementation Description Tasks

Below we have included a sample sequence use case for user.

Use Case Sequence	
Name	Choir Input
ID	1
Description	Choir Member fill our Choir guest form
Actors	Choir Member – Guest user
Benefit	Collect data for Choir members
Triggers	Submit button hit after data is entered
Preconditions	User has entered correct information
Main Course	1 User visit link 2 User input data 3 User submit data – email is sent with data 4 Database adds the information
Alternative Courses	1 User visit link 2 User input data 3 User submit data 4 Server off – data not accepted 5 Email received with information but no data 6 Manually add information

Figure 14 Use Case Sequence for Database

5.2.3. Functional Requirements

This feature has several functional requirements because this feature defines all the different categories and types of data which can be collected and stored within the data. All the requirements are listed below in the order of highest priority to lowest priority.

List of Requirements:

REQ-1: A new staff member shall be added with new username and password so that new hires also have access to the system.

REQ-2: Interns shall work on project and events to report their hours and get paid.

REQ-3: Only Staff and interns shall be able to create new projects to maintain the integrity of the software data.

REQ-4: The Programs shall have attendance, location, date, time and type of event being held so the information can be later used for Machine Learning Algorithm.

REQ-5: A retreat shall include list of members attending, semester, year and date so that retreat organizer can keep track of all the students.

REQ-6: An affiliation must exist between bride or groom to Loyola University otherwise wedding is not possible because only Loyola Affiliates can have weddings on Campus.
REQ-7: A wedding entity shall have a bride, groom and all their contact information so wedding coordinator can use information for reservation.
REQ-8: Once logged in, interns shall be able to update hours spent working on project to show progress on project.
REQ-9: When logged in, a supervisor is shall be able to check hours for the interns whom he supervises so that the hours can be paid for.
REQ-10: Any staff member shall be able to run queries on the database so that they can retrieve stored data for events and reporting.

2.4.3 Web Interface

5.3.1 Description and Priority

This feature is specifically for the users the interact with the database and to visually interact with the data. The web interface will provide an interaction tool for users who have no technical experience and it is meant to allow anyone to be able to use and understand the system. It is a gateway between the data and schema to a web page organized to show all the information users want to see. Below we have a list of general task for this feature to be completed. In the figure below we have also included the priority for each task and the need for getting it done for system to be fully functional.

5.3.2 Stimulus/Response Sequences

This feature has the several different sequences available for user to choose from depending on who the user is. Below we have sample for a Staff user.

Web Interface	0%		Start	Due	Assigned
Connect with Database	0%		Mar 19, 2017	Mar 28, 2017	High
User login page	0%		Mar 19, 2017	Mar 28, 2017	High
Staff logged in page	0%		Mar 19, 2017	Mar 28, 2017	High
Intern Logged in page	0%		Mar 19, 2017	Mar 28, 2017	High
Week Schedule page	0%		Mar 19, 2017	Mar 28, 2017	Low
Queries Page	0%		Mar 19, 2017	Mar 28, 2017	High
Projects and Events summary page	0%		Mar 19, 2017	Mar 28, 2017	Medium
Supervision page	0%		Mar 19, 2017	Mar 28, 2017	Medium
Edit Data page	0%		Mar 19, 2017	Mar 28, 2017	High
Guest form for Wedding	0%		Mar 19, 2017	Mar 28, 2017	High
Guest fc					High
Guest fc					High
Guest Form for Attended Events					High
Download CSV of Query result	0%		Mar 19, 2017	Mar 28, 2017	High
log out page	0%		Mar 19, 2017	Mar 28, 2017	High
Guest form for Retreat information	0%		Mar 19, 2017	Mar 28, 2017	High

Figure 15 Web Interface Feature Task List

Use Case Sequence	
Name	Query for Student of Hindu Faith in 2018 Class
ID	1
Description	Elise wants to be able to get a list of students of Hindu faith from class of 2018 for a Holi Preparation
Actors	Elise, who is Staff member at Campus Ministry
Benefit	Involve students of diverse faith background to recognize holidays of different faiths

Triggers	User enter exact query search they want to run
Preconditions	Existing data for Hind students is present – at some point this data has been collection
Main Course	5 Elise Log in 6 Elise go to Queries page 7 Elise enters all the queries into text boxes for query 8 Click submit 9 Query returns list of students of Hindu faith and all their information
Alternative Courses	7 Elise login 8 Go to Query page 9 Enter some data 10 Error of query not possible 11 Ask Elise to reenter information 12 Once accurate information is received, return query results.

Figure 16 Use Case Sequence Chart for Web Interface Feature

5.3.3 Functional Requirements

The primary functional requirements for this feature is to have a page existing. Below we have listed all the necessary functional requirements.

REQ-11: A user log-in page exists so that only staff and interns have access to the database.

REQ-12: A wedding form exists which can be used by Bride and Groom to enter information for wedding so that manual entry is not necessary by wedding coordinator.

REQ-13: A choir entry page exists for guest users to enter information so that manual entry by staff is not necessary.

REQ-14: An interested participant entry page exists for guest users to enter information so that manual entry by staff is not necessary.

REQ-15: An event attendee page exists for guest users to enter information so that manual entry by staff is not necessary. Weekly schedule

REQ-16: Formatted query results shall be displayed for any query run by interns or staff so that users can understand information and retrieve information from database.

REQ-17: Staff and Interns shall be able to download queried results in a CSV format so that they do not have to copy and paste information for offline use.

REQ-18: An interface shall exist for interns to update weekly hours so that they can report how many hours they have worked for payment.

REQ-19: An interface shall exist for supervisor to overview intern hours for the week and progress on projects.

2.4.4 Machine Learning Algorithm

5.4.1 Description and Priority

This feature implements the final part of this system – an Artificial intelligence system. This feature will take the collected data by Campus Ministry and take it a step further to make it more useful. This step will use the collected data via guest forms at the events and after running an algorithm it should be able to suggest date and times for interns to plan events to have a specific audience. Below is a list of task for this feature and their priority. Some of the features have a lower priority because it all depends on the amount of data we previously have collected.

Machine Learning	0%	Start	Due
Data Query from Database	0%	Mar 29, 2017	Apr 1, 2017
Preprocessing	0%	Mar 29, 2017	Apr 1, 2017
Select features to be used	0%	Mar 29, 2017	Apr 1, 2017
Normalize Data	0%	Apr 5, 2017	Apr 5, 2017
Training Data	0%	Apr 5, 2017	Apr 5, 2017
Testing Data	0%	Apr 17, 2017	Apr 17, 2017
Validation Data	0%	Feb 17, 2017	Apr 20, 2017
Connect to Web Interface	0%	Feb 17, 2017	Apr 20, 2017
Showing results on Web	0%	Feb 17, 2017	Apr 20, 2017

Figure 17 Machine Learning Algorithm Tasks List

5.4.2 Stimulus/Response Sequence

This feature is more complicated to have a sequence for because this is an artificial intelligence system and once it has been trained on an algorithm, validating and testing has been complicated it is going to suggest dates based on the inputs. It will only respond and decide based on inputs given and will follow the algorithm developed. This system will respond per the algorithm every time.

5.4.3 Functional Requirements

This feature is a very calculative feature and is just data analysis thus we only have the functional requirements which will build this feature.

REQ-20: The Algorithm shall query data from the database to have data to process the algorithm on.

REQ-21: The Algorithm shall pre-process data queried by running LDA, PCA or Feature Engineering to clean data so that the algorithm can provide improve accuracy.

REQ-22: The Algorithm shall divide the data into equal parts to have three sections; Testing, Validation and Training so that the algorithm can have a balanced dataset to improve accuracy.

REQ-23: The Algorithm shall display prediction for time and day of the week to user from the algorithm run so that user can have an answer for their specified attendance.

2.5 Other Nonfunctional Requirements

2.5.1 Non-Functional Requirements

2.5.1.1 Performance Requirements

2.5.1.2 The program will not be very hardware intensive so it shall run well on any web browser and follow HTTP protocol. Due to the program running well on low requirements, scalability is not much of a concern. More resources could possibly allow for faster query results and memory allocation for database inputs.

2.5.1.3 Availability Requirements

This software shall have a mission capable rate of 100 percent because it shall be available always and unavailability shall be zero. The probability for the system to operate satisfactorily at any given point shall be 100 percent unless there is a pre-defined program update. In terms of the nines metric, this software shall have five nines availability which equals 99.99% availability.

2.5.1.4 Maintainability Requirements

This software shall be maintained through least amount of maintenance required since administration who will maintain this will have least amount of effort. The system shall automatically update all the software being used in implemented. If such configuration is not possible for this specific software in this case, the user Manuel must include detailed information and step by step information for updating.

2.5.1.5 Reusability Requirements

This software shall be implemented in organized and layered classes to provide reusability of the software. The software will have various components and objects which will enable reusability of the code on various platforms. If the language requirements for implementation are changed, this software shall still hold reusability component through its design and features elements.

2.5.1.6 Fault Tolerance

This software shall be designed to be a failsafe or fail gracefully software, meaning it will either fail at reduced levels and if it completely fails it will protect the data as well the users. The software shall implement a graceful exit to prevent data corruption after an error has occurred. The user shall be given a message after each error has occurred.

2.5.1.7 Extensibility

This software shall be designed to allow extensibility. The client can add any features as needed and the system shall behave without error regardless of the new functionality working or not. The software shall be able to add new functionality without breaking the previous code and continuing to function without error with new functionality.

2.5.1.8 Security

This software shall only be accessible by those authorized to do so. Administrator shall provide authorization to new users and revoke authorization from employees no longer working at Campus Ministry. The guest form links shall only be available to those to whom they are shared with unless administrator authorizes availability of those forms on public website. This software shall provide security for data accumulation and overall software as some of the information being collected is sensitive information.

2.5.1.9 Usability

This software shall have a high degree of usability for the guest user as well the privileged users. The guest user shall be able to enter the information requested in the correct format within two tries. The privileged user shall be able to learn the process of completing each task within the software after one guided run. Overall, the software shall implement universal standards for buttons and interface transition to have ease of use and learnability to the software.

2.5.1.10 Scalability

The software shall have the capability to hand a growing amount of work as it is database which will continue to grow. The machine learning algorithm shall work regardless of if the queried dataset has only few points compared to thousands of points. The software shall be implemented to keep in mind that the system shall have the potential to be enlarged to accommodate the growth as needed.

2.5.1.11 Reliability

The software shall be reliability of 99.99 percent because user shall be able to depend on the software all the time for the functionality required. For any function being executed, the software shall end with a success and if an error occurs the system shall have error prevention implemented as well as management for risks of failures.

2.5.1.12 Deployment

This software shall have flexible deployment to the servers and database to allow client to change and update deployment as hardware system need to be changed. The software should not be device hardware deployed but instead shall be able to easily deployed from one device to the next.

2.5.1.13 Documentation

This software shall be accompanied by detailed documentation of requirements, design, procedure, testing and step by step process of software implementation. Since the software will be maintained by the client, developer shall provide client with all necessary and used information within the documents as well including online software information for assistance with maintenance.

2.5.1.14 Accessibility

2.5.1.15 This software shall be accessible from any device with connected to the internet. It shall to be configured to appear the same all on Computers regardless off screen size. User shall be able to access the software remotely if they have the accessible links and login information.

2.5.1.16 Readability

The software shall have high readability to allow any new incoming developer to be able to understand the code, edit as needed and change any implementation requested by client. The software design document shall include classes and description of software while the source code shall include detailed documentation for each method as well as class. The developer shall explain the programming style and naming conventions which the code shall follow.

2.5.1.17 Efficiency

The software shall be efficient in its implementation and use. The software shall load within seconds upon a query request and it should be instance for logging in, updating something and form entry. The software shall continue to perform at the efficiency rate regardless of how large the database has become.

2.5.1.18 Effectiveness

The software shall be effective in executing each functionality being implemented by providing the desired result. The software shall behave the way it is intended to and provides the user with an error message if an error occurs and software task does not complete.

2.5.1.19 Correctness

The software shall be correct in its implementation of all the required functionality as well as the non-functional requirements. The software shall be free from error and have a high accuracy rate to confirm its correctness.

2.5.1.20 Robustness

The software shall have robustness such that it shall cope with incorrect input. The software shall be designed to reduce human entry errors as much as possible to have robustness. Robustness shall be tested by giving software incorrect data input and the software shall report to user what software was entered incorrectly.

2.5.2 Software Quality Attributes

This system must qualify as a correct and flexible and maintainable system because once it is delivered to the client more work is not possible and it is only up to client for any changes they would like to make. This system shall also be reliable because all of campus ministry events and programming will go through this event as all the data will be saved here thus depending on the system is very important. The system should return the exact results we are looking for to be deemed reliable. The system should have all the functionality listed in the requirements to be considered correct. To be considered maintainable, the system should be created so minimum maintenance is required and user manual should be simple to read.

2.5.3 Business Rules

As this system is designed for campus ministry staff and intern only, once a staff member or intern is no longer working at the office, the user should no longer have access to the system. New Hired Employees would also need to be registered to be able to have access to the system. Guest Users should only have access to forms when the links are provided and Staff members should only be able access the software with correct username and password. The administrator will be responsible for creating password and username for staff members as well controlling the distribution of the guest forms for data collection.

3. Design Specification

3.1 Data design

This section provides a detailed description of all the data structure which will be implemented within this software. The software includes a Database and the users will be entering information via web interface thus we have several different structures where data is being passed from one point to another. The data is being stored as values pairs. When the user inputs data and hits submit, it was read in using the name assigned to field and stored into database in its specified position.

3.1.1 Internal software data structure

The internal data structures include structures which are passed between the components. Since the users will be entering data into the web forms, that data would needs to be read in and processed to be formatted for entry into MySQL database. The Web Interface component of the software is responsible for reading in the data entered by user from the web forms and it will send the data to connected Server program for processing. The program will process what the information by correctly formatting into MySQL data entry and inserting into MySQL. A detailed description of how the data looks is described in the database dictionary section.

3.1.2 Global data structure

The Global Data Structures for this software include query result selection entered by user. The user will enter the information into web form for query and this is then sent to the Program. From there depending on type of query either results are returned after running on MySQL or user send data to run Machine Learning Algorithm with those constraints. The data from this web form would be available to all the class within the program, sent to MySQL and Machine Learn Algorithm as well as HTML form. The global data structures will be simple key-value which are retrieved through the POST method by PHP files.

3.1.3 Temporary data structure

For this software, the temporary data structure includes the data file which would be created when a query is ran on the database and the results for it will be returned by Java Program via file. Although it is possible to display the results without creating a file, since one of the requirements is that user must be able to download query results in CSV format, creating the file completes this requirement. If the user chooses to not download the file, it will be deleted regardless. Another aspect of the database which would use a temporary file would be when user runs the Machine Learning Algorithm. Whenever the algorithm is requested, first the program will run MySQL query and pull data from database which is saved in file in specific format. The data is the preprocessed as needed and divided into training, verifying and testing. During preprocessing, a new file is created and after algorithm is completed then the results are returned to Java Program via file which are then displayed to user. All the file created in processing Machine Learning Algorithm are deleted once results are displayed to user.

3.1.4 Database Description

The database to be implemented is described here in details using an Entity relationship diagram, Relational Schema, functional Dependencies and a Data dictionary. These figures will describe exactly what type of data will be accepted into database and how it will process and relate to other models. The reference figure below shows the different symbols and their meaning within the figures. For each figure, which references the figure, it will be clearly stated that it references the EER Key Figure.

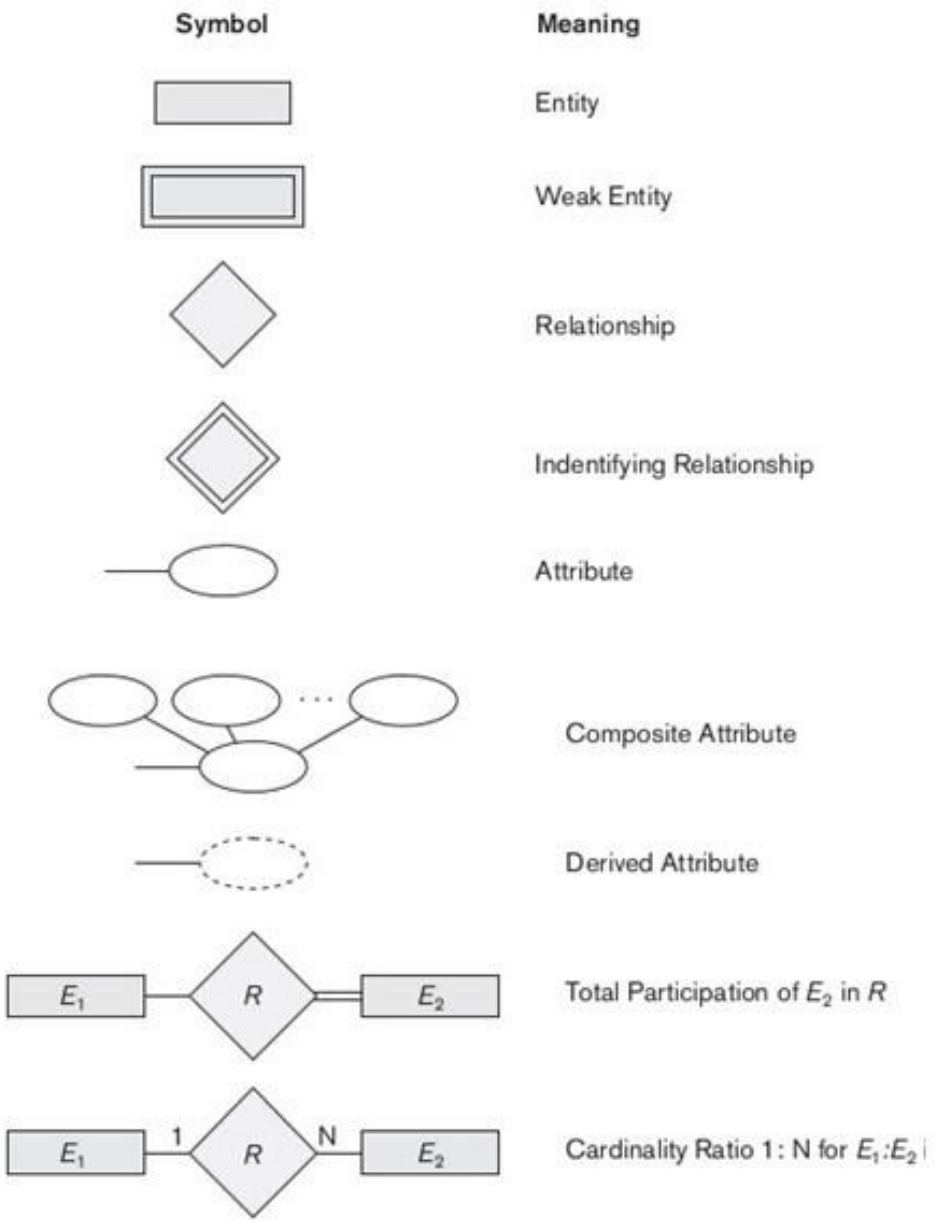

Figure 18 Entity Relation Reference Diagram

3.1.4.1 Entity Relationship Diagram

The Campus Ministry Database has been designed into an Entity relationship diagram and displayed below. The EERD shows all the different entities which are involved within this database and how they relate to one another. Entity diagram references EER Key figure for

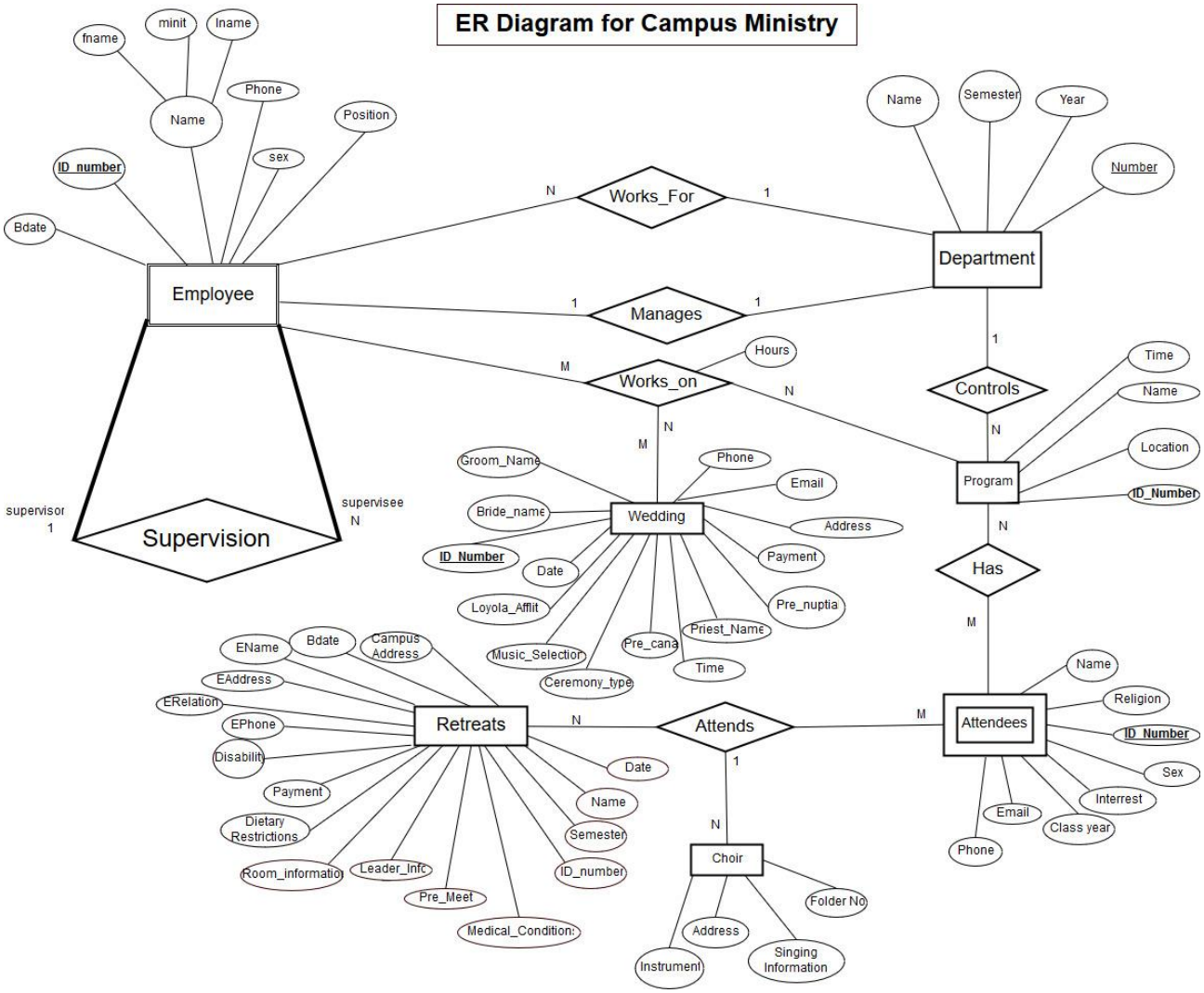

Figure 19 Entity Relationship Diagram for Campus Ministry Database

symbols.

3.1.4.2 Relational Schema

The relational schema takes the ER diagram and dives further into more detail by explaining each entity and its relation to other entities. Below, the schema is divided up by each entity and each one is displayed by itself.

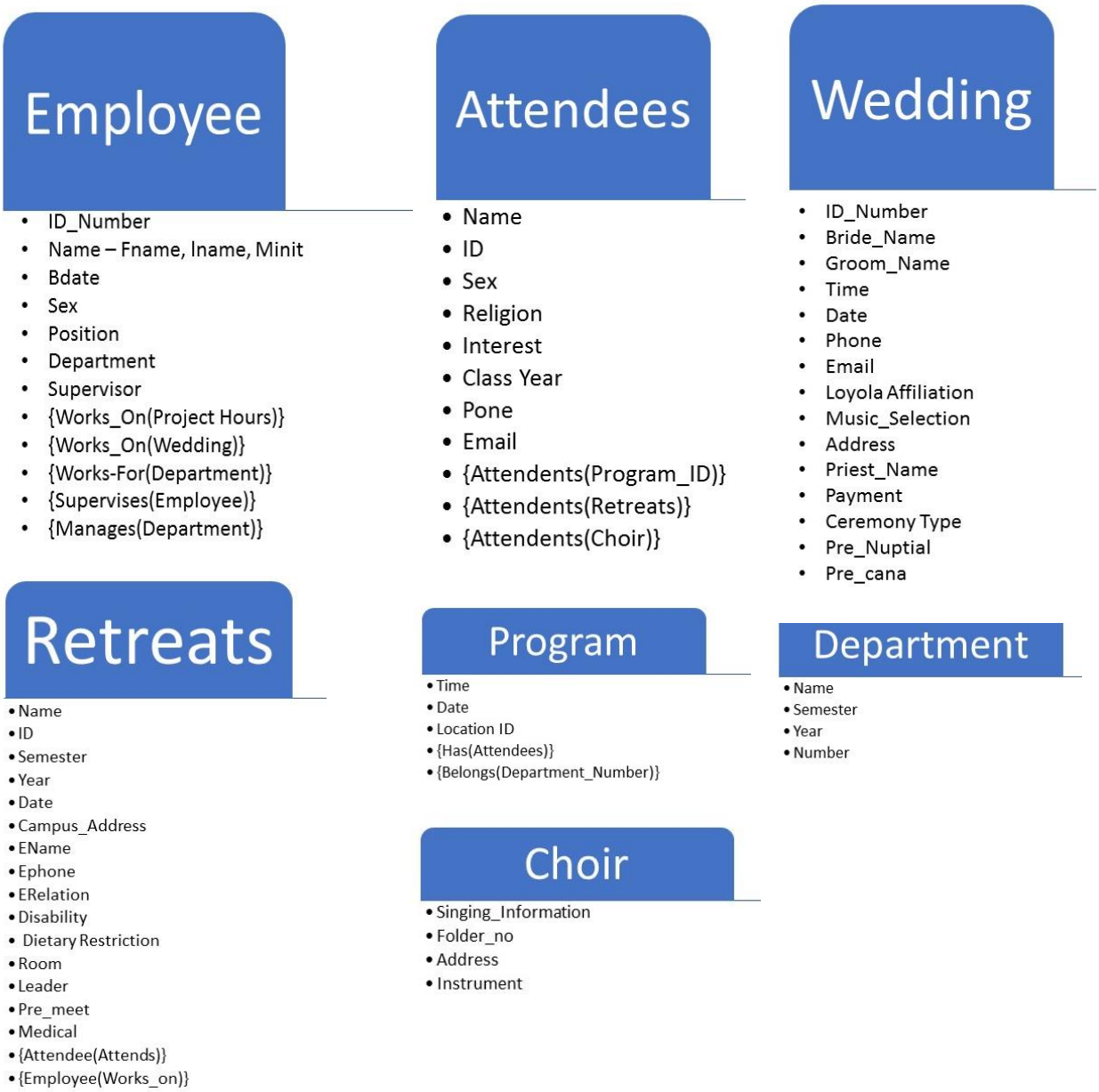

Figure 20 Functional Schema for Entities Department, Choir, Program, Employee, Attendees, Retreats and Wedding

In the above figure, each bullet represents single attribute for each entity and relationship with other objects is defined using brackets following by relationship and entity it has relationship with,

3.1.4.3 Relationship Schema

The relationship schema shows how each entity is related to other entities by displaying its relations as attributes. Below, the relationship schema for Campus Ministry Database. This figure references EER Key figure for symbols.

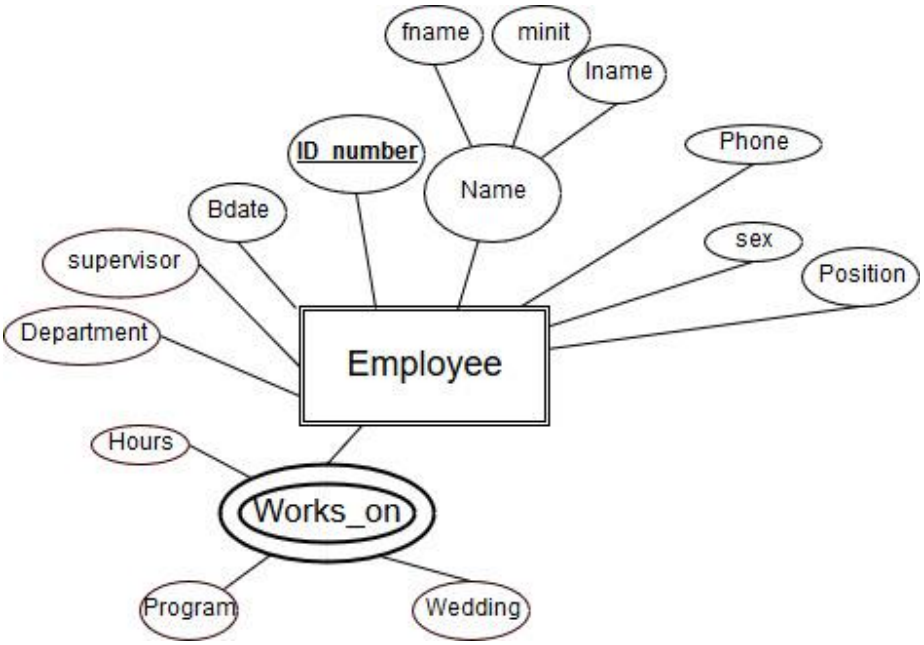

Figure 22 Relational Schema for Employee

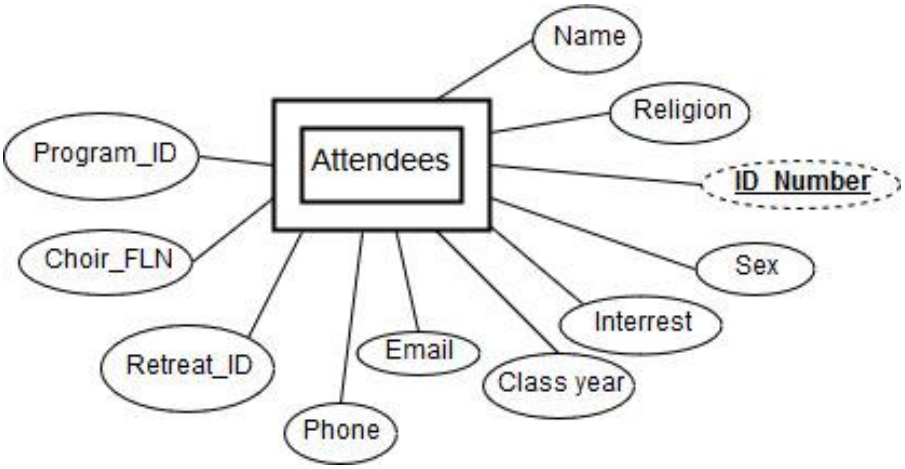

Figure 21 Relational Schema for Attendee

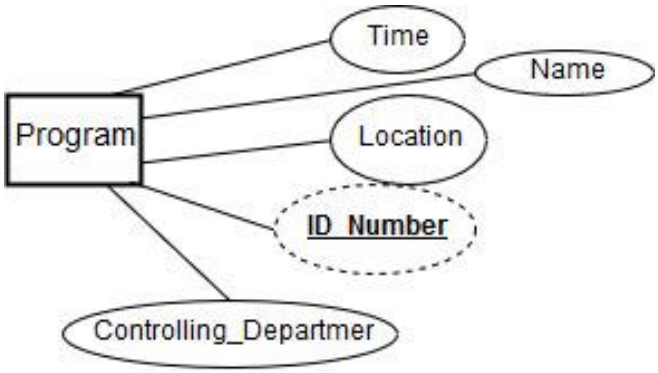

Figure 23 Relational Schema for Program

3.1.4.4 Data Dictionary

The data dictionary for Campus Ministry database states information for each variable within the database. This includes the variable type and its length. These variables reference the database as whole instead of each individual entity, allowing for repeated attributes to be stated at once.

Variable Name	Type	Length	Constraints
Fname	String	30	Characters only
Minit	Char	1	Characters only
Lname	String	30	Characters only
Sex	Char	1	Characters only
Bdate	String	10	Numbers and / only
Id_number	Integer	Varying from 2 to 16 – for different entities	Numbers only – automated by system
Semester	String	4	Two characters and two integers – FA17
Year	Integer	4	Numbers only
Name	String	50	Characters only
Religion	String	30	Characters only
Interest	String	60	Characters only
Phone	String	12	Numbers only and in format of xxx-xx-xxxx
Email	String	40	Characters only
Class_year	Integer	4	Numbers only
Hours	Float	4	Numbers only
Payment	Float	4	Numbers only
Date	String	10	Numbers only and / marks in format xx/xx/xxxx
Pre_meet	Boolean		Either yes or no for attending meeting
Dietary_restrictions	String	40	Characters only
Disability	String	40	Characters only
Leader_info	String	40	Characters only
Medical_condition	String	40	Characters only
Room_information	String	20	Characters only
Ceremoney_type	String	20	Characters only
Time	Float	4	Numbers only
Music_Selection	String	20	Characters only
Pre_nuptial	String	30	Characters only
Priest_Name	String	30	Characters only
Loyola affiliation	String	20	Characters only
Folder_no	Integer	2	Characters only
Instrument	String	30	Characters only
Singing_information	String	30	Characters only

Figure 24 Data Dictionary Table

3.2 Architectural and Component-Level Design

In this section, we will explain the overall architecture of this software as well as its individual components. This section will break down and explain the components in detail and presents limits, restrictions and implementation design features.

3.2.1 Program Structure

This software will be implemented using a Layered Architecture Structure to cater to all the different users and aspects of the software. This software includes web interface which is connected to database on the backend and has machine learning algorithm. The layered architecture will allow us to only show information to specific user which is pertinent to them and keep information hidden as necessary by user privileges. The layered architecture shows how the software would look and how the different parts will interact with one another.

3.2.1.1 Architecture Diagram

The figure below is the overall architecture diagram for the software. This only shows the broad over view for the architecture including servers and domains which will be used from already established applications such as MySQL.

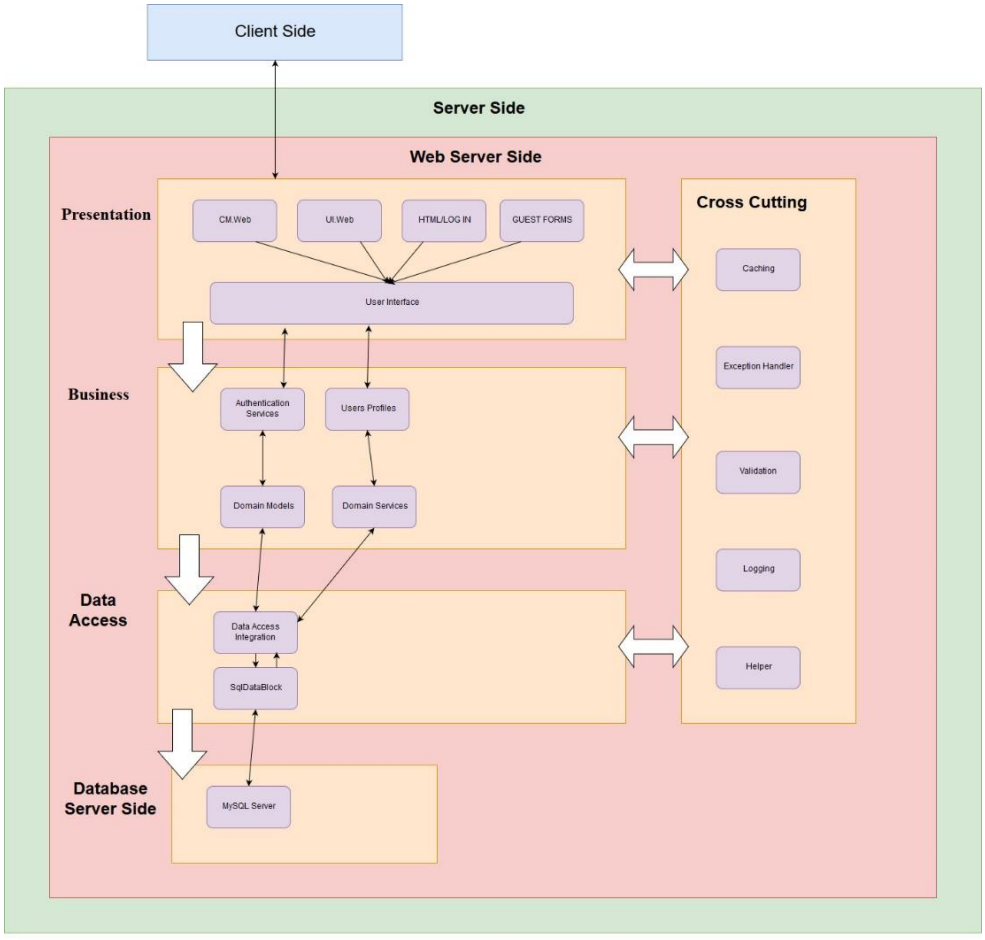

Figure 25 Software Architecture Diagram

3.2.1.2 Alternatives

The figure above is a very broad overview of the different layers of how the different layers will be interacting. Using the layered architecture allows the use of encapsulation and information hiding to only allow authorized users to access to the specified information. Implementing a layered architecture has many benefits such as the software's flexibility in long lasting and adapt to changes over time. An alternative to using layered architecture is using client server architecture style. The system forms as the user request information or inputs information and the system responds accordingly, thus using client-server gives us the control of being the server. The problem which arises in using client server for this software is when we have different user classes. The non-privileged users can be the clients but privileged would not be able to serve as clients at the same time since they need to use data differently.

3.2.2 Component Description

This section of the software will explain the various components of the software and their functionality within the software. A UML class diagram below shows all the components and how they relate to one another. Following the diagram, we have explained each of the components in detail.

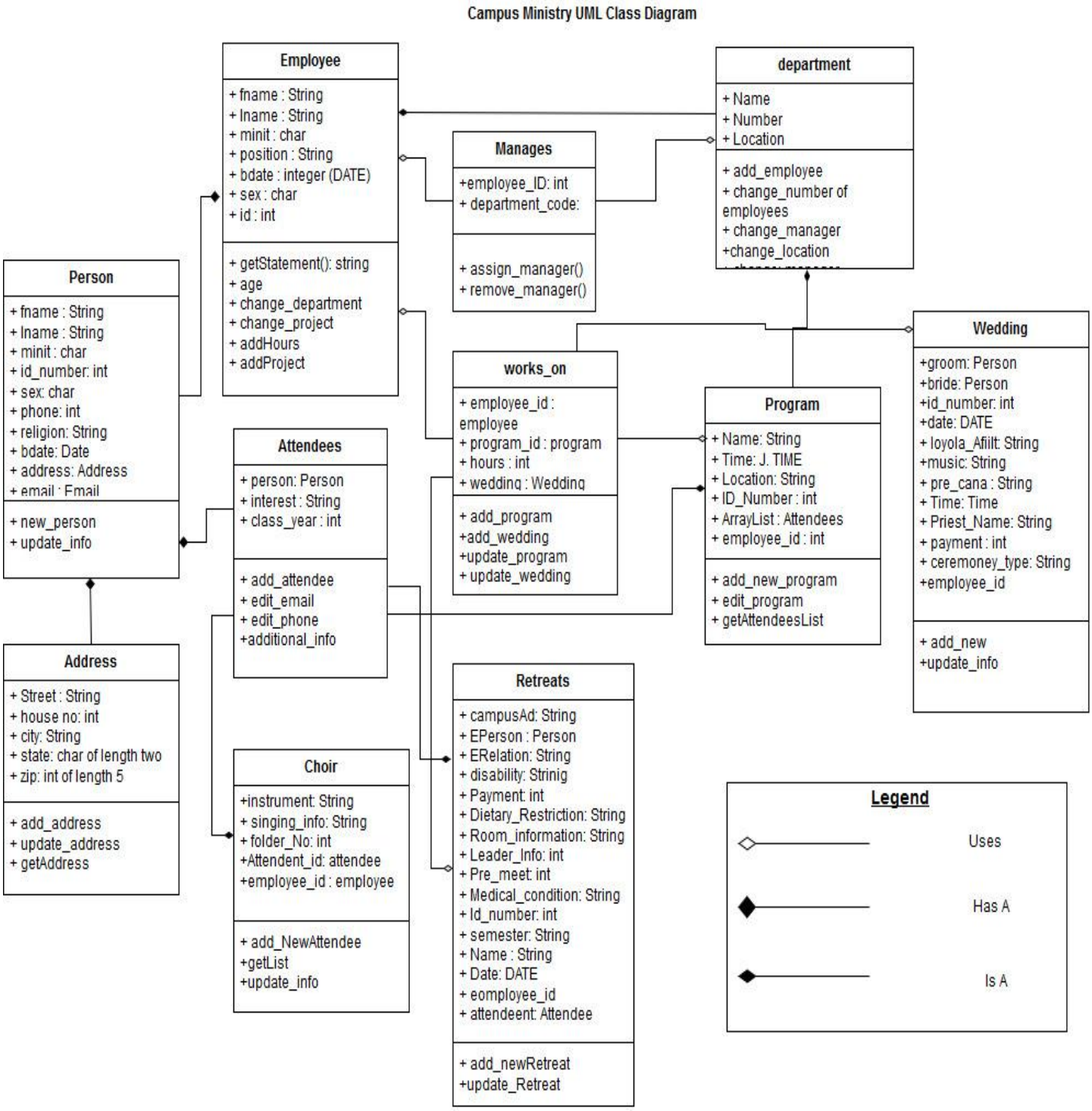

Figure 26 UML Class Diagram

3.2.2.1 Description of Data Collection

The UML diagram above shows all the different components and classes involved in this database and information is being processed. A major part of this software is data collection which campus ministry does at all their various events. The data collection happens using guest

forms which are simple HTML web forms which will be filled out by actors of different types of program.

3.2.2.1.1 *Processing Narrative for Component*

As depicted in the UML diagram, we different classes of users and relationships those users have with the software. This section describes the different components of this system such as Data Collection, Data Processing and Machine Learning Algorithm. The UML diagram gives summary of all the classes involved in the process and the section describes in detail how the components are being used within the system.

3.2.2.1.2 *Component Interface Description*

The Interface for this component will be implemented using HTML and CSS programming to create web pages which can be accessed via web links on the server. The Guest forms will be individual link forms which will be shared with people to collect information. They can be shared via redirect on current campus ministry or will be emailed for people to complete. The design rules and constraints for the Interface is described in the User Interface section below.

3.2.2.1.3 *Sub-Component Details of Data Collection*

As depicted in the UML diagram, the Attendee class is inherited by other classes and this class is directly created for anyone who is not staff or intern who will use the software to enter data. These users are not going to use or process data but merely enter their information for campus ministry for different programs they attend to get email about new programs, wedding reservation processing and contact information. This section is detailed with all the different types of Guest forms which will be implemented for data collection for all the different types of programs for campus ministry.

3.2.2.1.3.1 Sub-Component Processing Detail for Choir Data

The UML diagram shows that the Choir classes is an inheritance of Program class and it has attendees. This class will include chapel choir and Gospel Choir information for campus ministry. Currently, this information is collected on paper and saved on excel by manually typing it in by intern. The forms will allow chapel members to complete forms remotely to provide information to campus ministry to keep for chapel choir for every year.

3.2.2.1.3.1.1 Interface Description

The Choir form will be implemented as Guest form for any type of user to access. The design will adhere to constraints described in the User Interface section.

3.2.2.1.3.1.2 Algorithmic Model

Staff will be able to initiate choir for every year by doing a “new program” and including all the details. The students in the chapel will be added as attendees to the program and they will have information described within the UML.

3.2.2.1.3.1.3 Restrictions and Limitations

This information will be collected for chapel choir every year. If a student is member of chapel choir for more than one year, they would still have to enter new information regardless of new changes. Since this information is being collected via Guest forms, to protect data integrity, a user will only provide information once every semester. If changes need to be made, a privileged user would have to update this information in the database. User’s entering the data will not have access to the data once it is submitted.

3.2.2.1.3.1.4 Local Data Structures

This Webpage will collect and save different information about Chapel Choir members to keep track of members and performance roles. Below we have listed the Data Structures for Webpage and these structures will be saved in the database in the Choir table.

Variable Name	Type	Length	Constraint	Database Name
First Name	String	20	Name of Member	Fname
Middle Name	String	20	Name of Member	Mname
Last Name	String	20	Name of Member	Lname
Address	String	50	Address of Member	Address
Class Year	Integer	4	Graduation year	Cyear
Phone Number	Integer	10	Contact Information	Phone
Email	String	30	Contact Information	Email
Choral Experience	Boolean		Chapel Choir Membership Information	Cexp
Voice Part	Char	1	Placement Information	Vp
Instruments	String	20	Placement Information	Inst
Read Music	Boolean		Placement Information	Rmusic
Cantor	Boolean		Experience Information	Cantor
Folder NO.	Integer	2	Storage Information	flno

3.2.2.1.3.1.5 Performance Issues

The Chapel Choir webpage will be designed to make sure user cannot enter incorrect information and will not allow user to submit without completing the required information. The performance for this form is only impacted by the server speed and its availability. The webpage will perform always and the data must be recorded in the database whenever a user submits the forms. The data must be available when queried and should be saved as entered.

3.2.2.1.3.1.6 Design Constraints

The Chapel Choir Webpage shall be designed to restrict user from entering wrong type of information. User error should be as limited as possible. For example, if the type of data must be a Boolean, user should be given a yes or no option instead of input field. This will protect our data from corruption and possibly from having invalid data.

3.2.2.1.3.2 Sub-component Processing Detail for Retreats

The UML diagram shows that the Retreat classes is an inheritance of Program class and it has attendees. This class will include retreat information for campus ministry. Currently, this information is collected via online form and saved on excel by manually by looking through email received for form submission. The forms will allow Retreat Attendees to complete forms remotely to provide information to campus ministry to keep for every retreat.

3.2.2.1.3.2.1 Interface Description

The Retreat Form will be implemented as a Guest form on the campus ministry website allowing users to access it any time. The interface will implement UI design constraints described in the UI Interface section.

3.2.2.1.3.2.2 Algorithmic Model

Loyola University requires that Retreat payments for Campus Ministry be done on their website. Thus, this form will be used by sending users email about where payment must be made and where to fill out their information. The form submission will save the information in the database under the specific retreat and semester for that retreat. This will allow campus ministry staff to print details of all the retreat attendees and retreat information without having to look through list of emails for each member.

3.2.2.1.3.2.3 Restrictions and Limitations

This form is limited because due to Business Constraint set by Loyola University Maryland, campus ministry must have students fill this form out separately and another form separate to make a payment. This form will save all their information while the other form will be used only for payment records.

3.2.2.1.3.2.4 Local Data Structures

Below we have listed all the information being collected on the Retreat Webpage form and its format along with how it will be stored in the database. The Data dictionary section includes how the SQL format for this data.

Variable Name	Type	Length	Constraint	Database Name
First Name	String	20	Name of Member	Fname
Middle Name	String	20	Name of Member	Mname
Last Name	String	20	Name of Member	Lname
Campus Address	String	20	Address of Member	Caddress
Class Year	Integer	4	Graduation year	Cyear
Gender	Char	1	Gender of Student	Gender
Birth date	Integer	8	Date format – MMDDYYY	Bdate
Student ID	Char	9	Loyola Student ID	lid
Phone Number	Integer	10	Contact Information	Phone
Email	String	30	Contact Information	Email
Religious Affiliation	String	20	Religion for Prayer Purposes	Religion
Retreat Experience	Boolean		Retreat Experience	Rexp
Emergency Contact First Name	String	20	Name of Member	Efname
Emergency First Middle Name	String	20	Name of Member	Emname
Emergency Last Name	String	20	Name of Member	Elname
Emergency Contact Address	String	20	Address of Member	Eaddress
Emergency Phone Number	Integer	10	Contact Information	Ephone
Payment Information	Char	2	Card or Cash Payment	Ask what type of payment
Dietary Restrictions	String	20	Any dietary Restrictions to be listed and allergies	Drest
Disability Support	String	20	Disability Support Services Information	Dsup

How did you hear about us?	Char	1	Selection from Option	Outreach

3.2.2.1.3.2.5 Performance Issues

The Retreat webpage will be designed to make sure user cannot enter incorrect information and will not allow user to submit without completing the required information. The performance for this form is only impacted by the server speed and its availability. The webpage will perform always and the data must be recorded in the database whenever a user submits the forms. The data must be available when queried and should be saved as entered.

3.2.2.1.3.2.6 Design Constrains

The Retreat Webpage shall be designed to restrict user from entering wrong type of information. User error should be as limited as possible. For example, if the type of data must be a Boolean, user should be given a yes or no option instead of input field. This will protect our data from corruption and possibly from having invalid data.

3.2.2.1.3.3 Sub-Component Processing Detail for Weddings

The UML diagram shows that the Wedding class is an inheritance of Program class and it has person – bride and groom. This class will include wedding information for Loyola affiliates who want to have weddings at Loyola's Campus. Currently, this information is collected via pdf form and stored in paper by the wedding coordinator. The forms will allow Bride and Groom to complete forms remotely to provide information to campus ministry to keep for their wedding.

3.2.2.1.3.3.1 Interface Description

The Wedding Reservation Form will be implemented as a Guest form on the campus ministry website allowing users to access it any time. The interface will implement UI design constraints described in the UI Interface section.

3.2.2.1.3.3.2 Algorithmic Model

This Wedding Reservation form is simply use to keep record of wedding information for Bride and Groom for the wedding being held. The Bride and Groom cannot simply fill out the form to reserve a wedding. The Bride or Groom must have an affiliation with Loyola to be able to have a wedding thus the wedding date and rehearsal dates need to be validated by wedding coordinator before this form can be completed. Thus, the Wedding coordinator will share the link with Bride and Groom after wedding has been confirmed.

3.2.2.1.3.3.3 Restrictions and Limitations

As discussed above, this wedding form can only be completed after wedding date is confirmed thus this form will be available online for users to fill. To fill this business model, this form will be shared by wedding coordinator via email.

3.2.2.1.3.3.4 Local Data Structures

Below we have listed all the information being collected on the wedding reservation form and the type of data along with how it will be saved in the database. The database data type is described in the data dictionary section and in the table below only the name within the database for data value is listed.

Variable Name	Type	Length	Constraint	Database Name
Wedding Date	Integer	8	Date Format	Wdate
Wedding Time	String	8	Time/AM/PM	Wtime
Rehearsal Date	Integer	8	Date Format	Rdate
Rehearsal Time	String	8	Time/PM/AM	Rtime

Groom First Name	String	20	Groom Name	Gfname
Groom Middle Name	String	20	Groom Name	Gmname
Groom Last Name	String	20	Groom Name	Glname
Religious Affiliation	String	30	Religion Info	Grel
Address	String	50	Groom Address	Gaddress
Phone Number	Integer	10	Area code + phone	Gphone
Email	String	50	Email Format	Gemail
Class year/Affiliation	String	20	Affiliation Information	Gyear
Parish/Church	String	30	Groom Church Name	Gchurch
Address of Church	String	50	Groom Church Information	Gcadd
Bride First Name	String	20	Bride Name	Bfname
Bride Middle Name	String	20	Bride Name	Bmname
Bride Last Name	String	20	Bride Name	Blname
Religious Affiliation	String	30	Religion Info	Brel
Address	String	50	Bride Address	Baddress
Phone Number	Integer	10	Area code + phone	Bphone
Email	String	50	Email Format	Bemail
Class year/Affiliation	String	20	Affiliation Information	Byear
Parish/Church	String	30	Bride Church Name	Bchurch
Address of Church	String	50	Bride Church Information	Bcadd
Priest First Name	String	20	Groom Name	Pfname
Priest Middle Name	String	20	Groom Name	Pmname
Priest Last Name	String	20	Groom Name	Plname
Religious Affiliation	String	30	Religion Info	Prel
Address	String	50	Groom Address	Paddress
Phone Number	Integer	10	Area code + phone	Cphone
Email	String	50	Email Format	Cemail
Ceremony Type	Integer	1	0 for With Mass 1 for Without Mass 2 for Catholic Ceremony	Ctype

3.2.2.1.3.3.5 Performance Issues

The Wedding Reservation webpage will be designed to make sure user cannot enter incorrect information and will not allow user to submit without completing the required information. The performance for this form is only impacted by the server speed and its availability. The webpage will perform always and the data must be recorded in the database whenever a user submits the forms. The data must be available when queried and should be saved as entered.

3.2.2.1.3.3.6 Design Constrains

The Wedding Webpage shall be designed to restrict user from entering wrong type of information. User error should be as limited as possible. For example, if the type of data is date, a drop-down calendar selection will allow user to pick the correct date instead of having to enter.

3.2.2.1.3.4 Sub-Component Processing Detail for Attendees

The UML diagram shows that the Attendees classes is an inheritance of Person class and they attend programs. This class will include attendee information for events held by campus ministry. Currently, this information is collected on paper by event organizer or no information at all. The forms will allow Attendees to complete forms remotely to provide information to campus ministry to keep for every event.

3.2.2.1.3.4.1 Interface Description

The Attendee Form will be implemented as a Guest form on the campus ministry website allowing users to access it any time. The interface will implement UI design constraints described in the UI Interface section.

3.2.2.1.3.4.2 Algorithmic Model

The Attendee form can be completed at any time by any user. This document does not require any work to be done by the staff. Student and Event attendees will simply complete the form and upon submission the data will be saved in the database.

3.2.2.1.3.4.3 Restrictions and Limitations

This Attendee form is not program specific and is for general campus ministry. This will lead to an accumulated database for all attendees who have come to campus ministry events. Since this is a guest form, the data view is limited to staff users only and once attendee submits the form, they no longer have access to the data.

3.2.2.1.3.4.4 Local Data Structures

Below we have listed data structures implemented within the Attendees form and the name for each variable as it is being stored in database.

Variable Name	Type	Length	Constraint	Database Name
First Name	String	20	Name of Member	Fname
Middle Name	String	20	Name of Member	Mname
Last Name	String	20	Name of Member	Lname
Phone Number	Integer	10	Contact Information	Phone
Email	String	30	Contact Information	Email
Liturgical Ministries	Integer	1	Highest value is 5 – to be interested in all	Interest
Instruments	String	20	Instrument Only	Instrument
Retreats	Integer	1	Max=6	Interest
Programs	Integer	2	Max Value – 12	Interest
Conferences	Integer	1	Max Value -3	Interest
Leadership Roles	Integer	8	Max Value – 8	Interest

3.2.2.1.3.4.5 Performance Issues

The Attendees webpage will be designed to make sure user cannot enter incorrect information and will not allow user to submit without completing the required information. The performance for this form is only impacted by the server speed and its availability. The webpage will perform always and the data must be recorded in the database whenever a user submits the forms. The data must be available when queried and should be saved as entered.

3.2.2.1.3.4.6 Design Constrains

The Wedding Webpage shall be designed to restrict user from entering wrong type of information. User error should be as limited as possible. For example, if the data type is phone and user enters letters, it will give error to user to fix the mistakes.

3.2.2.1.3.5 Sub-Component Processing Detail for Staff Users Log in

The Staff user log in page is designed to allow staff users to run queries and create program and to interact with the database. This will allow staff users to input their username and password to get access to the site which will allow them to interact with the database.

3.2.2.1.3.5.1 Interface Description

The Staff Log-in Form will be implemented as a Guest form on the campus ministry website allowing users to access it any time. The interface will implement UI design constraints described in the UI Interface section.

3.2.2.1.3.5.2 Algorithmic Model

The login page is being implemented as a guest form – meaning it will available on the website all the time but only staff member can utilize by logging in and viewing the data from database. The staff members will choose their username and password during training. The administrator will create the username and password via SQL insert statement or PHP admin.

3.2.2.1.3.5.3 Restrictions and Limitations

This page will work as an access portal to allow staff users to get access to database and to view the data. The users must be previously registered by the admin before they have access to the database. It does not allow users to register through the site but instead a predefined username and password will need to be stored for the user.

3.2.2.1.3.5.4 Local Data Structures

The table below shows all the required information for the Staff users log in page. The table includes how the information will be displayed on the webpage and how it will be stored in the database.

Variable Name	Type	Length	Constraint	Database
Username	Char	10	Characters only	Username
Password	Char	10	Characters and numbers	Password

3.2.2.1.3.5.5 Performance Issues

The Staff login webpage will be designed to make sure user cannot enter incorrect information and will not allow user to submit without completing the required information. The performance for this form is only impacted by the server speed and its availability. The webpage will perform always and the data must be recorded in the database whenever a user submits the forms. The data must be available when queried and should be saved as entered.

3.2.2.1.3.5.6 Design Constrains

The Staff Login Webpage shall be designed to restrict user from entering wrong type of information. User error should be as limited as possible. If the user enters a wrong password or username, they should not be able to access any of the other forms.

3.2.2.2 Sub-Component Details of Data Processing

Another sub-component of this software is interacting with the data and site for staff members to interact with the database. These web forms will allow campus ministry members to search through the database, edit information and delete information. This sections gives detailed information how these functionalities will be executed and implemented.

3.2.2.2.1 *Processing Narrative for Data Processing*

This section describes how the Staff users will interact with database using HTML forms. The staff users, once logged in, will be able to search for students, programs and class years. This will allow campus ministry to invite students and attendees who have previously shown interest in campus ministry.

3.2.2.2.2 *Component Interface Description*

The users will interact with this component via Web interface. The webpage will allow user to pick from table and specific details from the table they are interested in. The query will return the results in table format. The user will be able to view the information and download csv format of the data. Once the user returns to main page the query results will be lost and the query would need to be run again.

3.2.2.2.3 *Sub-component Details for Data Processing*

3.2.2.2.3.1 Sub-Component Processing Detail for Queries

This sub-component capture show the queries will be run the database by the staff members. Once a guest user completes a guest form described in earlier component, the data will be stored in the appropriate table. The staff members will be able to run queries to retrieve the data. The staff member can search for specific people, all the table values and some attributes of the table. When a query is done, the results will be displayed on screen in table format. The user will be given option to download csv format of the table.

3.2.2.2.3.1.1 Interface Description

The query page will need to provide enough data to be able to create MySQL SELECT statements thus the users will be given drop-downs which will allow them to choose what they are looking for. The only input users should select for what table they are looking for. Once the selection is made, a MySQL query will be run in the background and all the data will be printed on screen.

3.2.2.2.3.1.2 Algorithmic Model

The user will be asked to select which table to query data from which corresponds to the different forms filled out by guest users. Upon selecting the table, user has a choice of either getting the whole table to further specify their query. For example, they can simply get the whole choir table or only chapel choir member who have first name Brandon.

3.2.2.2.3.1.3 Restrictions and Limitations

As discussed in Processing details, these classes have been created directly from entities thus the information the values contained in these classes will be defined by those entities. All the variables to be stored in these classes and processed are defined in the data dictionary table.

3.2.2.2.3.1.4 Local Data Structures

The local data structure for the query will be key values pairs which will be retrieved via post method by PHP files which will then process the query and return the table.

3.2.2.2.3.1.5 Performance Issues

The query webpage will be designed to make sure user cannot enter incorrect information and will not allow user to submit without completing the required information. The webpage will have drop downs and only ask for small amount of input from user. The performance for this form is only impacted by the server speed and its availability. The webpage will perform always and the data must be recorded in the database whenever a user submits the forms. The query shall return all the information requested for status of database and it should be correctly formatted in a table.

3.2.2.2.3.1.6 Design Constraints

The query Webpage shall be designed to restrict user from incorrect information, thus they are being asked to choose from drop downs. User error should be as limited as possible. If the user enters incorrect information and no data exists for such query, a message shall be displayed on screen information user. The query webpage shall be designed following the UI design constraints described below.

3.2.2.3 Description of Data Analysis

3.2.2.3.1 *Processing Narrative for Component*

The Machine Learning Algorithm will be used for Staff to get prediction for when an event should be held if they want to have specific attendance. The algorithm will return a time of day, between morning, afternoon and evening along with the day of the week. For example, if the staff enters that they are looking for an attendance of 20 people, the algorithm shall return time of day and day of the week for to have such an event.

3.2.2.3.2 *Component Interface Description*

The Machine Learning Algorithm will involve minimal interface. The only input necessary from user is the attendance which will be accepted from user via Web form and the value returned will be displayed to the user.

3.2.2.3.3 *Sub-Component Details*

A sub-component of the MLA (Machine Learning Algorithm) is the implementation Algorithm being used for MLA. We will be using a multi-layered perceptron to model the features and run the algorithm.

3.2.2.3.3.1 Sub-Component Processing Detail for Data Model

The Multi-Layered Perceptron will allow the implementation of different features such as attendance, program time, date, and location Depending on the amount of attendance for prediction user is looking for that, the AI shall return a different answer.

3.2.2.3.3.1.1 Interface Description

The interface for the AI component will ask user for input on how much attendance they are looking for. The user will be only required to give this input otherwise no information will be required because the program will query data from database automatically. Once the database has enough data to train and learn on, it would no longer need to be repeated thus, there is a waiting period until enough data has been gathered after which the algorithm shall be trained and tested.

3.2.2.3.3.1.2 Algorithmic Model

The data will be queried in Python using MySQL connector and queried from the database. The data will then have to be cleaned and processed. The script would read in the file and check if the values which are meant to be numeric are numeric or not. If the values are acceptable they would be written to a new file. The data to be used for learning at that point would be cleaned and processed data.

3.2.2.3.3.1.3 Restrictions and Limitations

The processing and accuracy of the AI is limited by amount of data available to train the algorithm. Another restriction for this algorithm is that unlike the rest of the system where user can process without human interaction, running this algorithm does require human interaction. As the amount of data increases, the AI would need to be trained and tested again.

3.2.2.3.3.1.4 Local Data Structures

The AI portion of this system is primarily dealing with csv files – it reading from and writing to CSV files. The data is being queried from the database as it is and it is being read in from database and written to csv file. The Data cleaning and processing files reads in csv files and saves the new results in csv file with a different name. The PCA and MLP will use the cleaned file to run the code and return the results.

3.2.2.3.3.1.5 Performance Issues

The performance of the AI is limited to data being used as parameters being used for algorithm. As the database, will be used and accumulate more data the algorithm will be available to improve training and learning.

3.2.2.3.3.1.6 Design Constrains

The data processing is designed as Python scripts and they will automatically collect data, save to CSV file and process the files. Human interaction is required to process each file and the user will simply have to run the files to get the results. PCA results shall be analyzed by a human user and the results will be used to run MLP.

3.2.3 Software Interface Description

3.2.3.1 External Machine Interfaces

3.2.3.2 External System Interfaces

3.2.3.3 Human Interface

The software is being implemented using HTML and CSS. All the users except for the Admin will be interacting with the software via Web Interface. More details about how the user interface is being implemented is explained in User Interface design description.

3.3 User Interface Design

3.3.1 Description of the User Interface

All the users interacting with the software will be using this interface to connect with the software and only the administrator has access to server side if necessary for them to check anything. The UI includes several guest forms to be accessed by non-privileged users and login information system UI for privileged users. In the figures below, some of the sample forms and pages are displayed and their implementation is explained.

3.3.1.1 Screen Images

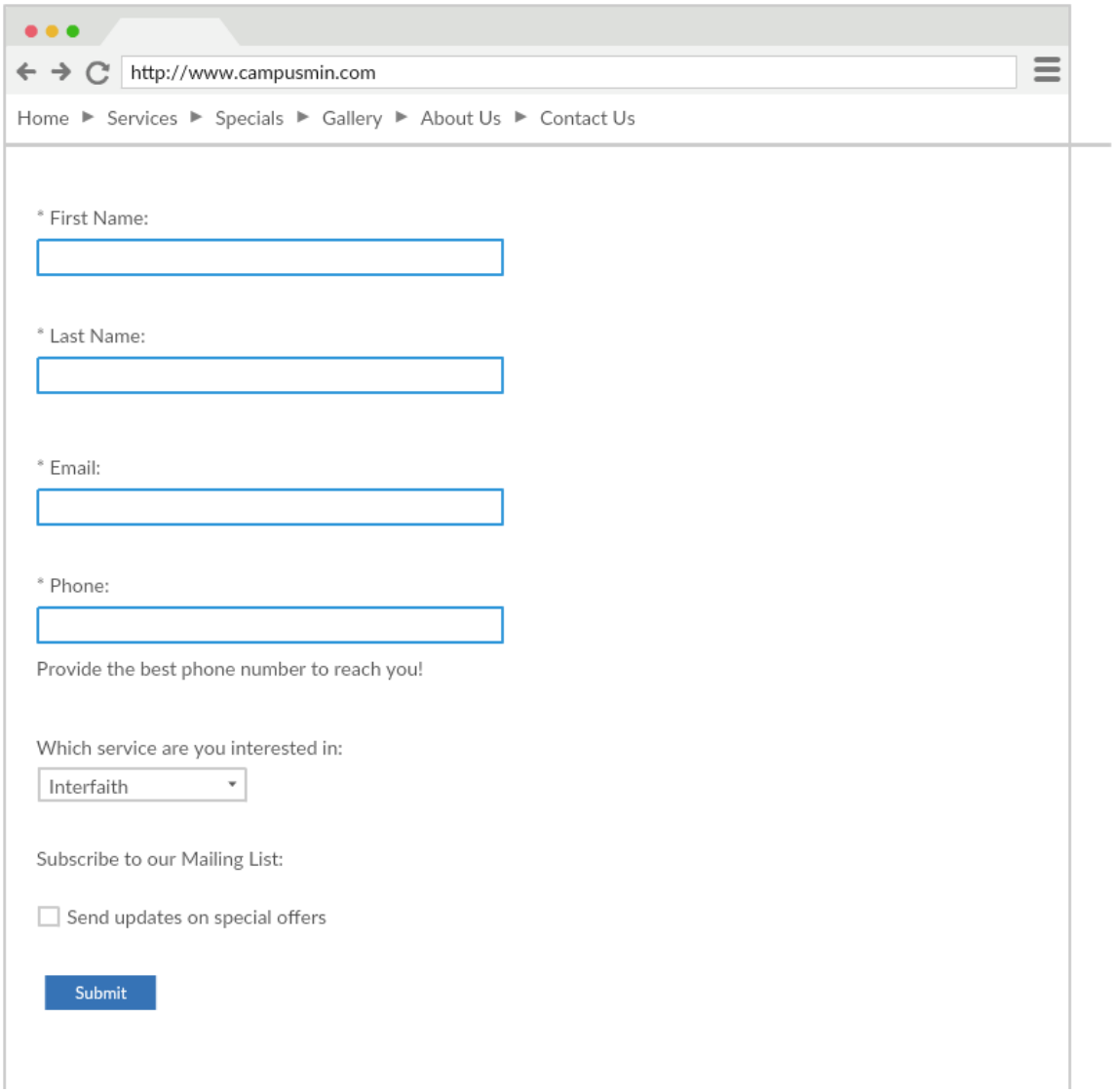

Figure 27 Campus Ministry Attendee Contact Form

The software will implement several different web forms and the template for the forms in general will be as displayed in the implementation above. It will have text field for user to enter information right under the question they are being asked and a submit button to confirm their submission.

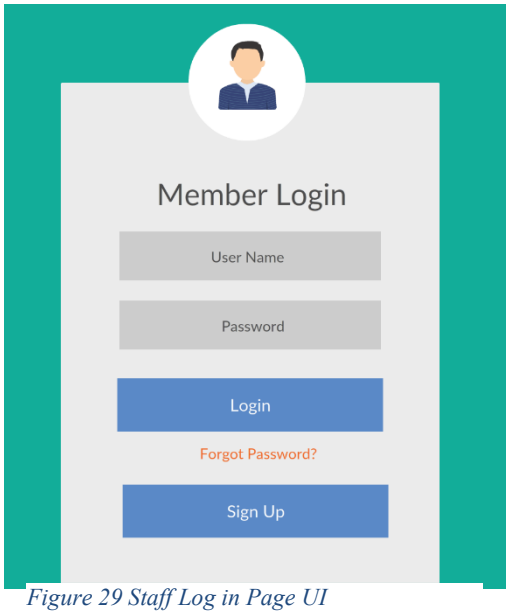

Figure 29 Staff Log in Page UI

The employee login page will be implemented in the style of the template above. It will give user options to login and sign up if that is what is needed.

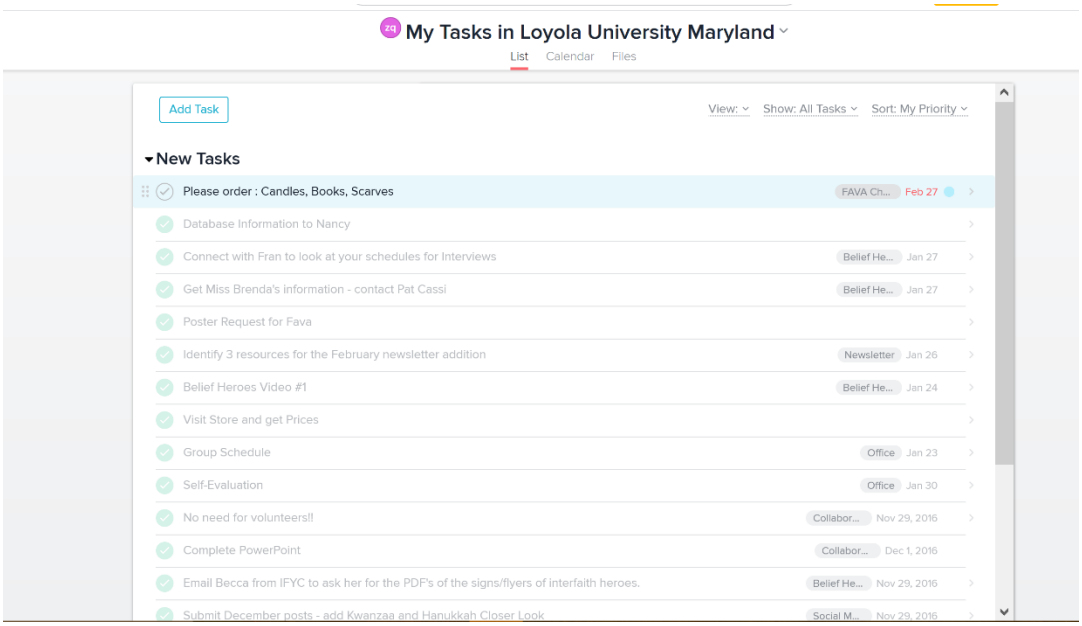

Figure 28 Employee Projects View Page

The page above displays a sample of what the employee would be seeing if they had listed all their programs and it would also show the number of hours spent on that task.

3.3.1.2

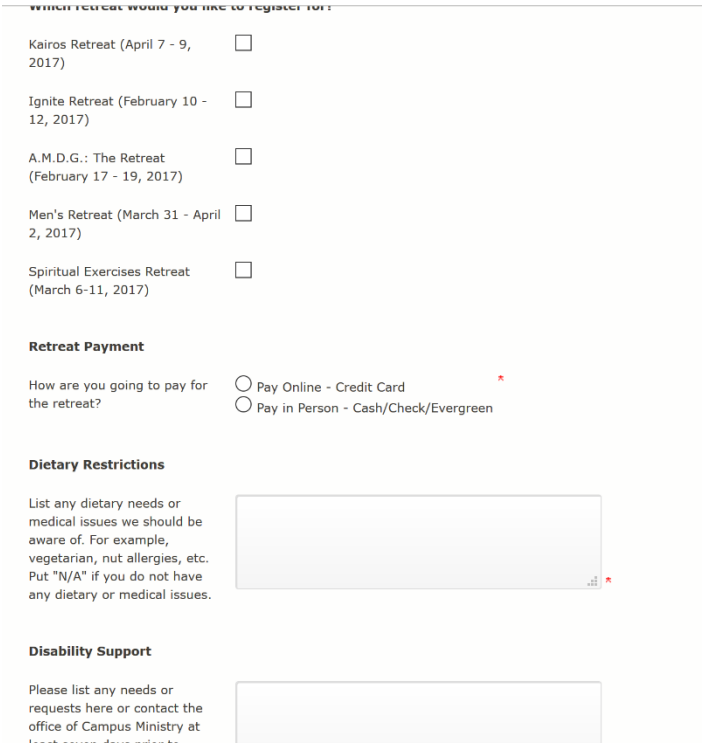

Figure 30 Retreat Sample Form

3.3.1.3 Objects and Actions

The user will have the choice to do different actions depending on where they currently are within the interaction. Since many of users will be guest, they will be primarily interacting with single page or form which will end with a submit button once they have completed the form with all the required information.

3.3.2 Interface Design Rules

To enhance the usability of this software we will be implementing several different rules with in the interface. The interface must implement universally used button and symbols to allow different groups of users to understand the actions which need to be completed. For example, a back button should be implemented using a back arrow instead of forward arrow. The software must have completion of action for the various forms user interacts with – there should be submit button or some sort of confirmation that the information has in fact been submitted and is completed. The interface must allow user to recover easily from errors and retain information needed without user having to remember from one screen to another. The interface must be symmetrical in its implementation and not have un-aligned fields.

3.3.3 Components Available

The interface will be implemented using HTML and CSS, thus we will have a variety of different styles and options to choose from. Any kind of implementation is possible with the two options. Both languages together give the power to the designer to design as needed by the users and by the software. We can make the forms as detailed as needed or as broad. For example, as discussed above, a date will string of length 10, so we can simply have a calendar for user to pick

from or have them insert date. The interface can be implemented using as many components necessary to make the interface within the design rules as well understandable within one sitting.

3.3.4 User Interface Development System Description

The process to develop the user interface will be start to with guest forms such as the Contact form, Wedding forms and Retreat forms. Since those forms are individual entries and do not impact the other parts of the interface, once they are completed. Black Box Testers will test the forms to understand how easy it is for them to use them to evaluate what needs to be changed. Once a general style is selected from working on the guest forms, then employee interface will be created using the standards from the testers for guest forms.

3.4 Restrictions, Limitations, and Constraint

One of the components of this software is the Machine Learning Algorithm which will be implemented to find when an event should be held to plan for specific amount of attendance. Although this component will be implemented, its performance cannot be verified and tested with real data because it is not available by the Client. The Client has already mixed the data into one big excel folder which takes away pertinent information such as event time, attendees and day. Without real data from previous events, the system can only predict on sample data until database is able to accumulate enough data from usage over time.

3.5 Appendices

3.5.1 Requirements Traceability Matrix

The table below lists all the requirements and explains how the different components cover the functional requirements set by the Client.

Requirement	Component	Rational
Database Functional Requirements: REQ-1 to REQ-10	1	Component 1, the different classes created cover all the requirements within the database. • A new staff member can be added • Interns work on projects, retreats and weddings. • Staff supervise interns • Retreats have list of members, semester, year and date. • Bride and Groom person have wedding at Campus Ministry – collect different information about Bride and Groom. • Supervisor can check intern hours.

Web Interface: REQ 11-20	User Interface and Component 1	These functional requirements under web interface are covered by user interface such as: • Log in page exists • Guest forms for Choir, Wedding, Interested Participants and event attendance. • Queries results • Hours updated and viewed. • Overview of projects. • Download query results.
Machine Learning Algorithm: REQ 21-27	Component 2 and User interface	The functional requirements under Machine learning are covered by component 2 and its sub components. • Pull data from previous events • Create weights and preprocess data to add weights. • Run algorithm on data to train, test and verify. • Return results to user - a day and time of day given to user on interface.

3.5.2 Packaging and Installation issues

The Client is still working to confirm where the server and database will be implemented. Initially it was suggested that it would be implemented on a physical machine but due to maintenance concern, the prospects for Cloud hosting are being explored. The actual installation is very simple once the location has been decided.

3.5.3 Design Metrics

The software is being designed to implement the database and interface first followed by Machine Learning Algorithm thus more part of the document was spent on database than on MLA. The figures presented in this document will be used to implement the software. More components and data structure might become necessary as we move further in implementation but as they are included in the software, this document will also be updated along with figures and data. In developing this software, one of the design metric which will be considered in coupling between classes. As this software is developed, we want to have low coupling between classes to allow changes to be implemented in one classes without impacting the other. Another design metric to consider is response for a class. This will measure the complexity of the class in terms of method calls. We will calculate this by adding number of methods in the class plus number of distinct method calls made by the methods in the class. Another metric to consider is

message passing coupling which will be done by measuring the number of messages passing among objects of the class. One final design metric to consider in this design implementation is Lack of Cohesion of methods which will measure the correlation between methods and local instance variables of a class. We want to have high cohesion because that would indicate good class subdivision. These metrics will allow us to consider how well the design for the software is and how well the subdivision is between classes.

3.5.4 Supplementary Information

3.5.4.1 Architecture Design

This section includes some sample information which shows smaller components and part of the architecture and possible help for the implementation.

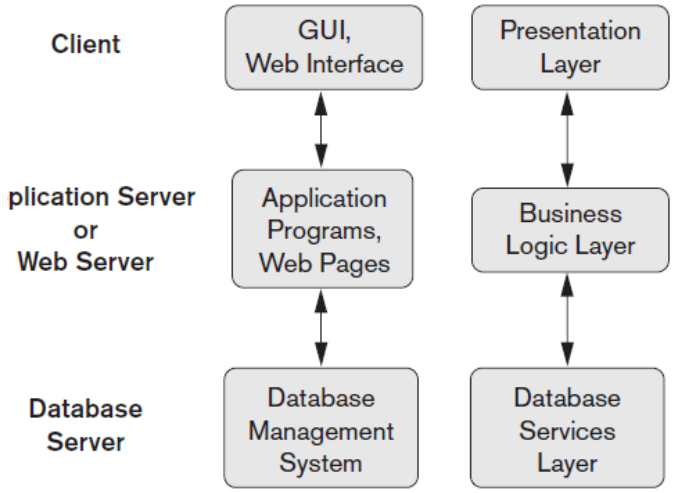

Figure 31 Overview of Architecture Layers

3.5.4.2 DBMS Type

This Database is Homogeneous DBMS because the same database will be used for all the different forms and interfaces. The database is a special purpose database and it can only be used by Campus Ministry because much of information being implemented is only used campus ministry.

4. Testing Specification

4.1 Test Plan

This section describes the overall testing strategy to be implemented for Campus Ministry Software. It describes in detail the project management issues that required to execute the tests effectively.

4.1.1 Testing Strategy

The Campus Ministry Software heavily depends user-interface and how well users are able to use the system. Thus, this software will be tested using a mixture of white and black box testing. Black-box testing will be conducted by sample user who will fill out the forms and run different queries to get the results. The sample user

4.1.1.1 Unit Testing

This software contains three different components and of those components use different types of implementation thus each will require different unit testing mechanism. Unit testing for each of the components will be done by several different tools. The software is a test-driven design and implementation thus it will be tested as it is being made. For each of the different components of the software the testing strategy is described below along with the component.

4.1.1.1.1 *Web Interface/Forms Unit Testing Strategy*

This component of the software is the primary interaction component for all the users of the software. This component encompasses both the Guest forms – to be used by program attendees for - and user log-in pages to be used by campus ministry staff. These forms will be testing tools as well as by sample users. The testing tools to be implemented for the Web forms include using Selenium, a portable software-testing framework for web applications. Another tool that will be used for web interface is Parasoft which will allow for automated peer code reviews preparation, notification and tracking.

4.1.1.1.2 *Database Unit Testing Strategy*

One of the major components of this software and the main back end for this software is the MySQL database. The data entered via web forms will be stored in the data. The two components we need to test for the database is storage of data in each of the tables and running specific queries. The database will be tested using TAP which allow us to write MySQL testing scripts. These test scripts will be written for each of the table and for the queries as well.

4.1.1.1.3 *Machine Learning Algorithm Unit Testing Strategy*

The last component of this software is the Machine Learning Algorithm and this will be programmed on python. In order to evaluate and test each of our function, Python testing scripts will be created. The Python scripts will allow us to do coverage testing for each of the method and we can check using sample data how well our algorithm works.

4.1.1.2 Integration Testing

The Campus Ministry Software involves three different components and integration testing will be a crucial part in its implementation. For integration testing, we would still have to test each of our components but instead of how well each of the function does within these components we have to look at the component as a whole.

4.1.1.2.1 *Web Interface/Forms Integration Testing Strategy*

The data input forms will be tested using a sample user pool to test how well users are able to actual use the web interface. Users will be asked to fill out the forms without any guidance and the amount of time it takes for them to correctly fill out form without any errors will be our measuring tool. An automated tool for integration testing which is to be implemented "TestingWhiz". This will allow us to measure several different aspects of testing and automate forms filling as well.

4.1.1.2.2 *Database Integration Testing Strategy*

The database will be tested by running queries on the whole database. We will request information which combines more than one table. MySQL and Tap scripts will be used to write the queries and run them to test how well and how quick the results are returned. An automated

tool which will be implemented for database testing is “Mydebugger” will allow us to run testing on the database and will provide even more coverage.

4.1.1.2.3 *Machine Learning Algorithm Integration Testing Strategy*

The AI component of this project will have Integration testing built into since our algorithm will provide us an overall accuracy percentage. This accuracy percentage will tell us how well our algorithm is doing based on our sample data. Another testing tool which we can add for testing the algorithm is writing scripts which test the algorithm to get higher accuracy. The Machine Learning will be tested via scripts and the accuracy within the algorithm will be used as measurement tool for how well our code is written.

4.1.1.3 High-Order Testing

4.1.1.3.1 *Whole System Testing*

This software will have three different components combined and working efficiently once it is completed thus we would need to have sample users use this system to evaluate it. Due to the complexity of the software, it is difficult to test the program with real people using this software. As a result, we will have two different groups who will be testing this software. One of the sample group will be the white box tester who would be people who familiar with coding and aware of how this project is developed. The other group will be our black box testers who will have no idea about how the back-end is processed and how it is being developed. Using two different groups will give us an idea about how well people accustomed to programming can use and for those who are not accustomed to programming. Using sample users will also allow us to get input about what component does not work well with the others.

4.1.1.3.2 *Usability Testing*

Usability testing will be done with real users and it will be measured how long it takes for then to learn the system. As a sample user uses the system, it will be measured how long it takes them to really understand and how well they understand the software. This will be done in two different ways, remote and on location. For remote testing, users will be sent the forms and they will be asked to fill them out followed by a survey about how well the software worked for them. On Location testing will be done at Campus Ministry staff meeting and users will be asked to use this software without any explanation. The systems usability will be tested how seeing how long it takes for them to start using system without pauses and unsure about outcomes.

4.1.1.3.3 *Maintainability Testing*

The maintenance of this software is solely dependent on the administrator thus to test the maintainability we would have to test how well he is able to use the back-end of the software. This will be done by providing the admin with a user manual and they will be asked to fix issues and update the software to check how well they are able to complete those tasks. The user will not be provided with any background except for the user manual and they how quickly they are able to learn to use this system.

4.1.1.3.4 *Performance Testing*

The performance of the software is crucial since we are working with several different components which combine and their mobility as a unit is very important. One way to measure the performance of our software will be to execute load testing. This will allow us to check how well our software does when it is being requested and used by multiple people. We will use a Query Surge, a tool which will allow us to run load and stress testing on the software to measure its performance.

4.1.2 Testing tools and Environment

As we have mentioned for each of the components above, we will be using several different tools to test the program. Below we have listed all the different tools and what component they will be used for. These tools are linked in the Appendices section.

Component	Testing Tool	Programming Language of Component
Database	TAP	MySQL
Database	MySQL scripts	MySQL
Database	Mydebugger.com	MySQL
Database	STK/Unit	MySQL
Web Interface	PHPunit.de	PHP
Web Interface	phpt	PHP
Web Interface	Simpletest.org	PHP
Web Interface	Star Tutorial	PHP
Web Interface	Selenium	HTML
Web Interface	Parasoft	HTML
Web Interface	Query Surge	HTML
Machine Learning	Python Scripts	Python

Figure 32 Software Component and Testing Tool Table

4.1.3 Test Schedule

Due to the complexity of this software and its integration of different components, this software must be a test-driven development. The unit testing for the software is completed as it is being developed and the integration testing is done as each section is completed. Since most of the software is user input and user involved, this needs to be tested by users to better measure how well our software does. Below we have included a tentative schedule for when each of the testing should be done.

Testing Type	Starting Date	End Date
Unit Testing	03/10/2017	04/15/2017
Integration	03/29/2017	04/17/2017
System Testing	04/15/2017	04/18/2017
Performance	04/19/2017	04/25/2017
Maintenance	04/24/2017	05/01/2017

Figure 33 Testing Schedule for Software

4.2 Test Procedure

4.2.1 Unit Test Cases

4.2.1.1 Unit test Case Input and Expected Result

The web forms need to be tested for each of their field for acceptable input and result. Each of the forms have been explained in their own section below.

4.2.1.1.1 *Attendee Form Test Case and Expected Result*

This is the form to be used for guest users who attend campus ministry events and to get their information for future use. In the table below we have listed all the information being requested, type of input acceptable, input given and expected result for wrong input. For the attendee form only the information below needs to be tested because users will be checking boxes for the other information. In the table below, in case of error, a message pops up asking user to fix their error.

Test Case #	Field Name	Valid Input	Given Input	Result

1	Name – first and last	Zahara Kazmi	1234456	Error – name must be a string
2	Name – first and last	Zahara Kazmi	23zahara23	Error – must only have letters
3	Name – first and last	Zahara Kazmi	Bobby Jones	No Error
4	Name – first and last	Zahara Kazmi	@xyzml	Error – must contain letters only
5	Phone Number	410-000-0000	410-216-2222	Acceptable input for field
6	Phone Number	410-000-0000	Xyz-xyz-xx23	Error- must be numeric
7	Phone Number	410-000-0000	@@-##-####	Error – must be numeric
8	Email	ex@gmail.com	zqk@loyola.edu	Acceptable input for field
9	Email	ex@gmail.com	123@gmail.com	Acceptable input for field
10	Email	ex@gmail.com	noyes@gmail.com	Acceptable input for field
11	Email	ex@gmail.com	Zwe123.com	Error – must be valid email
12	Email	ex@gmail.com	123@zahara.com	Error – must be valid email

Figure 34 Attendee Form Test Cases

4.2.1.1.2 *Retreat Form Test Case and Expected Result*

The retreat form will be used by students and staff members attending Loyola retreats to provide their information. This information is used by campus ministry to have for each member on the retreat to keep track of their dietary restriction, medical information and emergency contact. Below we have listed all the information being requested and the test cases to be carried for all the fields. We are only running test cases for inputs for fields where user will input information, if user will pick from drop down or check boxes from already existing option, it is not being tested since no wrong input can be given. In the table below, in case of error, a message pops up asking user to fix their error.

Test Case	Field/Information	Input Accepted	Input Given	Result
13	Name – first and last	Zahara Kazmi	1234456	Error – name must be a string
14	Name – first and last	Zahara Kazmi	23zahara23	Error – must only have letters
15	Name – first and last	Zahara Kazmi	Bobby Jones	No Error
16	Name – first and last	Zahara Kazmi	@xyzml	Error – must contain letters only
17	Class Year	####- 2019	2017	Acceptable input for field
18	Class Year	####- 2019	17	Error – must be year format
19	Class Year	####- 2019	Seven	Error – must be numeric

20	Class Year	####- 2019	@17	Error – must not contain symbols
21	Student ID	####### - 1657238	1658238	Acceptable input for field
22	Student ID	####### - 1657238	Xyzsyz	Error – must be numeric
23	Student ID	####### - 1657238	XYZ-143	Error – must be numeric
24	Student ID	####### - 1657238	16582389870	Error – must only be 7 digits
25	Phone Number	410-000-0000	410-216-2222	Accepted
26	Phone Number	410-000-0000	Xyz-xyz-xx23	Error- must be numeric
27	Phone Number	410-000-0000	@@-##-####	Error – must be numeric
28	Email	ex@gmail.com	zqk@loyola.edu	Accepted
29	Email	ex@gmail.com	123@gmail.com	Accepted
30	Email	ex@gmail.com	noyes@gmail.com	Accepted
31	Email	ex@gmail.com	Zwe123.com	Error – must be valid email
32	Email	ex@gmail.com	123@zahara.com	Error – must be valid email
33	Birth date	08/21/1995	08/21/1998	Accepted
34	Birth date	08/21/1995	04/11/2017	Error – must be at least 18
35	Birth date	08/21/1995	04/05/2017	Error – must be at least 18
36	Address Line 1	# Street	2	Error – must be street name
37	Address line 1	# street	2 avenue	Acceptable input for field
38	Address line 2	Letters/#	Apt 1	Acceptable input for field
39	Address line 2	Letters/#	Campus House	Acceptable input for field
40	City	Letters only	Baltimore	Acceptable input for field
41	City	Letters only	B12	Error – must be letters only
42	City	Letters only	@btm	Error – must be letters only
43	Zip code	5 numbers - 21234	2e34	Error- must be numbers only
44	Zip code	5 numbers	21210	Accepted
45	Zip code	5 numbers	Sefhkjsfh	Error – must be numbers only

Figure 35 Retreat Form Test Cases

4.2.1.1.3 *Wedding Form Test Case and Expected Result*

The wedding form will be used by perspective bride and groom to wedding and personal information to campus ministry for wedding preparation. The information which is being asked via selection is not being tested because the user will be given a selection to pick from and any

answer can be chosen. Below we have listed all the test cases for Wedding Reservation form. In the table below, in case of error, a message pops up asking user to fix their error.

Test Case	Field/Information	Input Accepted	Input Given	Result
1	Wedding Date	Month-Day-year	Today's Date	Error- Wedding Date must be in future
2	Wedding Date	Month-Day-year	Past Date	Error- Wedding Date must be in future
3	Wedding Date	Month-Day-year	06/15/2017	Acceptable input for field
4	Rehearsal Date	Month-Day-year	Today's Date	Error- Wedding Date must be in future
5	Rehearsal Date	Month-Day-year	Past Date	Error- Wedding Date must be in future
6	Rehearsal Date	Month-Day-year	06/15/2017	Acceptable input for field
7	Name – first and last	Zahara Kazmi	1234456	Error – name must be a string
8	Name – first and last	Zahara Kazmi	23zahara23	Error – must only have letters
9	Name – first and last	Zahara Kazmi	Bobby Jones	Acceptable input for field
10	Name – first and last	Zahara Kazmi	@xyzml	Error – must contain letters only
11	Religious Affiliation	Words Only	Islam1234	Error – Must be letters only
12	Religious Affiliation	Words only	Islam	Accepted
13	Phone Number	410-000-0000	410-216-2222	Accepted
14	Phone Number	410-000-0000	Xyz-xyz-xx23	Error- must be numeric
15	Phone Number	410-000-0000	@@-##-####	Error – must be numeric
16	Email	ex@gmail.com	zqk@loyola.edu	Acceptable input for field
17	Email	ex@gmail.com	123@gmail.com	Acceptable input for field
18	Email	ex@gmail.com	noyes@gmail.com	Acceptable input for field
19	Email	ex@gmail.com	Zwe123.com	Error – must be valid email
20	Email	ex@gmail.com	123@zahara.com	Error – must be valid email
21	Class Year	####- 2019	2017	Accepted
22	Class Year	####- 2019	17	Error – must be year format
23	Class Year	####- 2019	Seven	Error – must be numeric

24	Class Year	####- 2019	@17	Error – must not contain symbols
25	Address Line 1	# Street	2	Error – must be street name
26	Address line 1	# street	2 avenue	Acceptable input for field
27	Address line 2	Letters/#	Apt 1	Acceptable input for field
28	Address line 2	Letters/#	Campus House	Acceptable input for field
29	City	Letters only	Baltimore	Acceptable input for field
30	City	Letters only	B12	Error – must be letters only
31	City	Letters only	@btm	Error – must be letters only
32	Zip code	5 numbers - 21234	2e34	Error- must be numbers only
33	Zip code	5 numbers	21210	Acceptable input for field
34	Zip code	5 numbers	Sefhkjsfh	Error – must be numbers only
35	Church Parish	Words Only	Chruch1	Error – Must be letters only
36	Church Parish	Words only	Cathedral	Acceptable input for field

Figure 36 Wedding Form Test Cases

4.2.1.1.4 *Log- in Form Test Case and Expected Result*

The log in form will be used by Campus Ministry Staff and interns to gain access to the web forms to run queries and interact with the database. The login form restricts access of other users and only campus ministry staff will be able to access this page. Below we have listed the test cases for this form. In the table below, in case of error, a message pops up asking user to fix their error.

1	Username	Alpha-numeric	Zqkazmi12	Acceptable input for field
2	Username	Alpha-numeric	12345	Error – must be alpha-numeric
3	Username	Alpha-numeric	Zqkazmi	Error – must be alpha-numeric
4	Password	Alpha-numeric (lovely23)	12345	Error – password cannot be all numbers
5	Password	Alpha-numeric	Lovely23	Accepted
6	Password	Alpha-numeric	Lovely	Error – must be alpha-numeric

Figure 37 Login Form Test Cases

One of the underlying assumption for the above test cases is that the accepted username and password must match and be allowed by the administrator.

4.2.1.1.5 *Chapel Choir Test Case and Expected Result*

The chapel choir form will be used by the choir director to keep track of all members of choir and to have members provide their information via the form. Below we have included all the fields in this form which will require testing and other fields are drop down selections. In the table below, in case of error, a message pops up asking user to fix their error.

Test Case	Field/Information	Input Accepted	Input Given	Result
1	Name – first and last	Zahara Kazmi	1234456	Error – name must be a string
2	Name – first and last	Zahara Kazmi	23zahara23	Error – must only have letters
3	Name – first and last	Zahara Kazmi	Bobby Jones	Acceptable input for field
4	Name – first and last	Zahara Kazmi	@xyzml	Error – must contain letters only
5	Class Year	####- 2019	2017	Acceptable input for field
6	Class Year	####- 2019	17	Error – must be year format
7	Class Year	####- 2019	Seven	Error – must be numeric
8	Class Year	####- 2019	@17	Error – must not contain symbols
9	Phone Number	410-000-0000	410-216-2222	Acceptable input for field
10	Phone Number	410-000-0000	Xyz-xyz-xx23	Error- must be numeric
11	Phone Number	410-000-0000	@@-##-####	Error – must be numeric
12	Email	ex@gmail.com	zqk@loyola.edu	Acceptable input for field
13	Email	ex@gmail.com	123@gmail.com	Acceptable input for field
14	Email	ex@gmail.com	noyes@gmail.com	Acceptable input for field
15	Email	ex@gmail.com	Zwe123.com	Error – must be valid email
16	Email	ex@gmail.com	123@zahara.com	Error – must be valid email
17	Birth date	08/21/1995	08/21/1998	Acceptable input for field `
18	Birth date	08/21/1995	04/11/2017	Error – must be at least 18
19	Birth date	08/21/1995	04/05/2017	Error – must be at least 18
20	Address Line 1	# Street	2	Error – must be street name
21	Address line 1	# street	2 avenue	Acceptable input for field
22	Address line 2	Letters/#	Apt 1	Acceptable input for field

23	Address line 2	Letters/#	Campus House	Acceptable input for field
24	City	Letters only	Baltimore	Acceptable input for field
25	City	Letters only	B12	Error – must be letters only
26	City	Letters only	@btm	Error – must be letters only
27	Zip code	5 numbers - 21234	2e34	Error- must be numbers only
28	Zip code	5 numbers	21210	Acceptable input for field
29	Zip code	5 numbers	Sefhkjsfh	Error – must be numbers only
30	Folder no	2 digits	456	Error – must be two digit number
31	Folder no	2 digits	Lik	Error – must be two digit number
32	Folder no	2 digit	02	Acceptable input for field

Figure 38 Chapel Choir Test Cases

4.2.1.1.6 *Program Form Test Case and Expected Result*

The program form will be used by intern and staff to initiate a new event to get list of attendees. This data will be used in machine learning algorithm such as date, time and location. Since this information is carried over for the AI as well it is very important to properly test all the data in this form. The test cases are only for fields where users will be inputting information and no test cases are being done for selection options. In the table below, in case of error, a message pops up asking user to fix their error.

Test Case	Field Name	Input Accepted	Given	Result
1	Event Name	Speed Faithing	12345	Error – name must be a string
2	Event Name	Speed Faithing	Speed2	Error – must only have letters
3	Event Name	Speed Faithing	Mass of Holy Spirit	Acceptable input for field
4	Event Name	Speed Faithing	@mAss	Error – must contain letters only
5	Semester	Letter and two-digit year – FA17	Fall	Error – Must semester year format
6	Semester	Letter and two-digit year – FA17	2017	Error – Must semester year format
7	Semester	Letter and two-digit year – FA17	SP17	Acceptable input for field

Figure 39 Program Form Test Cases

4.2.1.1.7 *Interest Form Test Case and Expected Result*

The interest form will work as sort of contact form for campus ministry. This form will be used when anyone wants to get in contact with campus ministry. They would have to pick reason they want to contact campus ministry and send their information. Below we have listed all the test cases for the requested information for interest form. In the table below, in case of error, a message pops up asking user to fix their error.

1	Name – first and last	Zahara Kazmi	1234456	Error – name must be a string
2	Name – first and last	Zahara Kazmi	23zahara23	Error – must only have letters
3	Name – first and last	Zahara Kazmi	Bobby Jones	No Error
4	Name – first and last	Zahara Kazmi	@xyzml	Error – must contain letters only
5	Phone Number	410-000-0000	410-216-2222	Accepted
6	Phone Number	410-000-0000	Xyz-xyz-xx23	Error- must be numeric
7	Phone Number	410-000-0000	@@-##-####	Error – must be numeric
8	Email	ex@gmail.com	zqk@loyola.edu	Acceptable input for field
9	Email	ex@gmail.com	123@gmail.com	Acceptable input for field
10	Email	ex@gmail.com	noyes@gmail.com	Acceptable input for field
11	Email	ex@gmail.com	Zwe123.com	Error – must be valid email
12	Email	ex@gmail.com	123@zahara.com	Error – must be valid email

Figure 40 interest Form Test Cases

4.2.1.1.8 *Search Query Test Case and Expected Result*

This search form will be used for Campus Ministry staff to run queries on the database and although users will input simple data this needs to be converted into SQL format. The way this form has been formatted is user will get a chance to select from which table they are looking and then they would need to enter specific data. An example of something on this form might is also explained is as follows:
Search (Drop Down Selection from table) where (field selection) equals (user input)
Below the test cases for this form have been listed. In the table below, in case of error, a message pops up asking user to fix their error.

Test Case #	Field Name	Input Accepted	Given	Result
1	Name – first or last	Zahara or Kazmi	1234456	Error – name must be a string
2	Name – first or last	Zahara or Kazmi	23zahara23	Error – must only have letters
3	Name – first or last	Zahara or Kazmi	Bobby Jones	Acceptable input for field

4	Name – first or last	Zahara or Kazmi	@xyzml	Error – must contain letters only
5	Event Name	Speed Faithing	12345	Error – name must be a string
6	Event Name	Speed Faithing	Speed2	Error – must only have letters
7	Event Name	Speed Faithing	Mass of Holy Spirit	Acceptable input for field
8	Event Name	Speed Faithing	@mAss	Error – must contain letters only
9	Class Year	####- 2019	2017	Acceptable input for field
10	Class Year	####- 2019	17	Error – must be year format
11	Class Year	####- 2019	Seven	Error – must be numeric
12	Class Year	####- 2019	@17	Error – must not contain symbols
13	Folder no	2 digits	456	Error – must be two digit number
14	Folder no	2 digits	Lik	Error – must be two digit number
15	Folder no	2 digit	02	Acceptable input for field

Figure 41 Search Query Test Cases

Sample Query: Select (Choir) where Name equals “Brandon”.

4.2.1.2 Stubs and Drivers

4.2.1.2.1 *Web Form Stubs and Drivers*

The forms described above are guest form and this forms input is read from the webpage and stored in the database via PHP scripts. During testing, we cannot save the information in the database since it is not valid information. Thus, we need to make stubs and driver for our form. The stubs for these forms will be read in the data, make sure it is correct format and simply inform user that information has been accepted. These stubs will not save information to database but they will check for proper format for all the different fields. Since these guest forms are simply requesting information, they would not require drivers but only stubs.

4.2.1.2.2 *PHP Script Drivers Stubs and Drivers*

The PHP scripts read in the data from webpages and process the input to MySQL insertion statements. Since these scripts are reading in data from webpages we would need to create stubs and drivers for these scripts. The driver for these scripts will provide sample data to be processed for insertion into MySQL scripts. The driver would simply provide sample data to scripts and we would need to create drivers for each of our MySQL table. The stubs for PHP scripts will accept MySQL statement created by script to make sure it is in acceptable format. If it is accepted MySQL format for insertion statement, the stub would return data entered and if not accepted it will return an error for why it cannot be accepted. We would need to create stubs and drivers for all the different tables within the database because each table accepts different information thus their insertion statements will be different as well.

4.2.2 Integration testing

4.2.2.1 Test cases for Integration Testing and Expected Result

4.2.2.1.1 *Web Form Test Case and Expected Result*

Integration testing for the web forms must be very different from unit testing for this component. Each of the form was requesting different type of information from the user but the result for all the forms was submission. All the forms can be tested by checking if the submission goes through or not. Each form has a submit button at the end and if someone required information is not given or wrong format is given then the form will not submit. Since the integration test will be the same for all the forms, instead of showing it for each form, we have listed all the test cases below for the component altogether. Below we have listed test cases for Web from integration testing.

Test Case #	Field	Input Accepted	Given	Result
1	Submit button	All required information in form	Just one field – name	Error- Please enter required information before submitting
2	Submit button	All required information	All required information	Data saved in database
3	Submit button	All required information	All required +additional information	Data saved in database

Figure 42 Web Form Test Integration Case

4.2.2.1.2 *Search Query Script Test Case and Expected Result*

Integration testing for the Query form has to be to make sure the database returns something to the user. In unit testing, it is already being checked to make sure our input is in MySQL query format thus with integration we must check if it database is returning results to main screen. Below we have listed all the possible inputs and results for integration testing for query form.

Test Case #	Field	Input Accepted	Given	Error
1	Result Button	MySQL statement	MySQL Statement	Result table printed

Figure 43 Search Query Integration Test Cases

4.3 High-Order Testing

4.3.1 Performance and Load/Stress Testing

To conduct performance testing for this software we will be using sample users as well as testing tools. In performance testing, the testing tools to be implemented will be Apache JMeter and Open STA. Both of these tools are used for performance testing and they will test different aspects of the software's performance. Apache JMeter will combine the HTML with database, allow us to run test on the whole test and how well it performs. Using Apache JMeter we can write sample test plans and then in GUI we can pick which plan to run on what type of load. JMeter will let us run heavy load on server, group of sever and specific form. This will test the strength of the forms and server and analyze the overall performance. Similarly, Open STA will also run HTTP stress testing along with Performance testing. Open STA will allow us to record in the browser and produce simple scripts which we can edit. The scripts then can be executed to simulate many

users to generate high performance load on the server. Another tool which can be implemented is Radview which will allow us to easily write scripts and analyze our software. Radview will provide correlation, parameterization, response validation and load generation to test the strength of our software. Using these tools individually and a combination of them we should be able to test how well our software is performing and we would also have the subjective experience of the users to along with numeric performance. Below we have listed set of stress testing which shall be completed on this system.

Test Number	User Load	Rational
1	10 transactions per hour	Testing 10 users per hour on the system
2	100 transactions per hour	Testing 100 users per hour on the system
3	500 transactions per hour	Testing 500 users per hour on the system
4	1000 transactions per hour	Testing 1000 users per hour on the system
5	10000 transactions per hour	Testing 10000 users per hour on the system

4.3.2 Security Testing

Security is a major priority and component of our software thus conducting security testing is significantly important to the software's performance. In designing this software's security testing, we have considered several different principals. One of them is confidentiality and that not only means protecting information provided by users but also who has authorization to see it. Data integrity means that during data transfer nothing is being altered and we are keeping data as it is being provided. The focus is the authentication – users need to have access to be able to look at data and access it. There are several different ways the security can be compromised such as injection, broken authentication and session management, cross-site scripting and missing function level access control. Testing the security is very important and some of the tools we will use to test the security are FxCop and FindBugs. Both tools allow us to create rules for our software and they will try everything to get into our system and see if they are successful. The tools will give feedback on what works and what does not work, giving developer a chance to make changes to better protect the software.

4.4 Usability Testing

4.4.1 Testing Plan

The Usability testing of this software will be done by using a sample of users and the users learning to use the software. Some of measurements for usability testing to be considered includes skill level of user, time required to get used to software, measure of increase in user productivity and user's attitude. Due to the time constraint of this project, the only parameter we could have an issue testing is productivity since that must be a measure over certain period.

4.4.2 Sample Users

Usability testing can only be effective if we have a range of novice and expert users. Since this software will be used authenticated users who have access to the data and by users who will simply be entering information – we have to test the usability for both of those users. The authenticated user sample will be campus ministry staff and interns. The users who will be using this software will get to test and see if it works for them. The other set of users will be picked at random from campus ministry events and will be asked to fill out the form.

4.4.3 Testing

In order to conduct the testing, users will fill out a survey which will ask the about their skill level of computers, the time it took them to learn the software and how well they liked the software. The survey results will allow us to measure how well our software performs in terms of usability.

4.4.4 Analysis

Using the feedback from the testing and survey, we can evaluate the usability of the software. The feedback will also allow us to measure what components in software are complex and are not easy to understand. Knowing about the flaws in our design will provide us with an opportunity to make future changes. The analysis of usability testing should meet the criteria below to be declared usable:

1. A sample user who is novice should be able to get used to the software in about an hour and they should find the software to be easy to enter information into.
2. A sample user who is an expert should be able to get used to the software in less than 30 minutes and they should find the software easy to find and enter information.

If the software usability testing passes the above criteria, it is decla

4.5 Performance Testing

4.5.1 Testing Plan

The performance testing for this software will include running load and stress testing as well as web load. Much of the performance of this software will depend on the type of server we use and its connectivity. For performance testing we will not be using sample user but instead we will implement automated tools which will test the software via provided scripts.

4.5.2 Tools

The tools we will be using for testing the performance of our software include JMeter, Open STA and Radview. These tools will allow us to test the speed, reliability and accuracy of our software. These tools will test the strength of our software by running different types of load and using three different tools gives a good comparison to get an average of how well our software works.

4.5.3 Testing

The testing will be automated and scripts will be written as specified by the tools the test the software. The different loads we need to test include heavy load, medium and low load on server. We will also run multi-threading testing using these tools and recorded tests will provide us with the analysis for our software. Since the testing is automated, we can run as many test as necessary and analyze the results to evaluate the limits for the software.

4.5.4 Analysis

The testing results must adhere to criteria listed below to acceptable as a well performing software.

- The server response time does not degrade to a noticeable point and at all times it has a relatively quick response time.
- The system degradation does not impact the security of the system.
- The reliability and accuracy of the system should not decrease as load increases.

4.6 Testing Specific to Web Application

4.6.1 Content Testing

The content testing for this software will be done using Markup validation service. This validation service will be executed for all the HTML pages. The markup will provide all different types of error with in the page and what is expected and what is returned. The service does not provide any sort of score but the we can measure how well formed the webpages are by counting the number of errors. The webpages will be considered valid and complete if the number of error is less than 10. If the amount of error is greater than 10, the webpage should be changed to fix the error to achieve completeness.

4.6.2 Database Testing

The database queries and database itself is being tested separately, explained under unit testing and integration testing. In order to test the connectivity of the software, the server will continuously check the database server to make sure connectivity is possible.

4.6.3 Compatibility Testing

This software is developed as web application available on any browser. This software is being designed to be used and accessible on any device regardless of OS, Browser or any other constraints as long the device is connected to the internet. The webpages might appear differently on different bowsers and device depending on screen size but their overall design is to fit any screen. This software will be considered to have achieved compatible testing if it can provide queries and save inputted data the same on different platforms. The users who will be tested the software will conduct the compatibility testing as well since they will be running it on their own devices.

4.6.4 Navigation Testing

The system will be hosted on an online commercial server but due to the separation in user type we will have different navigation for the users.

4.6.4.1.1 *Data Entry Form Navigation Testing*

The data entry forms or “guest” forms will be separate individual forms which will be provided to users. These forms will be shared with users via links in emails and provided as redirection through the Loyola website. The user should not need to search these forms anywhere but should be provided to users.

4.6.4.1.2 *Authenticated Site Navigation Testing*

The authenticated site will be private link only accessible by staff and users. The website will be linked on the current campus ministry website so it can be accessed by all staff members. The staff should have previously been registered on the system by the administrator.

4.6.5 Security Testing

The security testing technique for the authenticated site as well the administration side will be done by using the techniques mentioned in section 3.3 under security testing.

5. User Testing Specification

5.1 Executive Summary

The Campus Ministry System is being developed to assist campus ministry in their daily works flow and to execute tasks efficiently. This system shall qualify the following components to be considered usable. First, we shall look at learnability – this asks the question of how easy is it for user to accomplish basic task in first encounter. Next is efficiency which speaks the speed of user performing task once they have learned the system. Then memorability is considered which refers to how easily the user can reestablish using system after not lack of use. Next we look at errors, specifically how many errors a user makes and if they can recover from them easily. Finally, the last one is how pleasant the overall design is. Another quality component is utility which asks you to make sure that your software provides the features needed. The usefulness of product is decided by checking if the product has both usability and utility.

The software includes two main user classes – privileged users and non-privileged users. The non-privileged users shall test the usability of the Guest forms. The Guest forms include wedding forms, retreat form, attendee form and contact form. The privileged users shall test the login process and query process. The privileged users shall test how to interact with database via web interface. This documents explains in greater detail how

5.2 Methodology

5.2.1 Participants

The participants' responsibilities will be to attempt to complete a set of representative task scenarios presented to them in as efficient and timely a manner as possible, and to provide feedback regarding the usability and acceptability of the user interface. The participants will be directed to provide honest opinions regarding the usability of the application, and to participate in post-session subjective questionnaires and debriefing. As discussed in the Executive summary, we have two different user classes, thus different type of users will require us to have different participants.

The privileged user class includes Staff members and those members are the only ones who are allowed access to the database and allowed to interact with the data. Access to the privileged user interface requires a password and username. Participants for this class will be interns and staff members of Campus Ministry. The Participants will include five Staff members and ten interns. These members have no previous experience of working with the system but these participants shall have computer skills. The participants shall require training – explained in the training section.

The non-privileged users class includes attendees of campus ministry program, anyone interested in program information and choir members. These participants can be college students, Loyola Alumni, parents and public. These participants are not required to have any previous experience. The instructions for the interface is given on the form but the users will only fill out the form with information being requested. These participants shall be accumulated by emailing Chapel Choir members and requesting information. At least 50 random participants from Loyola University shall be asked to interact with the interface and the usability tests shall be performed.

5.2.2 Training

The privileged class shall receive training to explain how the system must be used. The users shall be given an overview of functionality available for users to compete and run down of how to complete the functionality. The users shall only require at most 30 minutes of training. The participants shall be allowed to choose their own username and password. Upon gaining access to the interface, participants shall be shown how to use the interface only once. The participants will receive an overview of the usability test procedure, equipment, and software.

5.2.3 Procedure

Participants will take part in the usability test at Campus Ministry Staff Meeting at Campus Ministry Office in Cohn Hall, Loyola University Maryland. A Dell Computer with the interface will be used in a Windows OS. The participant's interaction with the prototype will be monitored by the facilitator seated in the same room. The facilitator will also act as note taker and data logger.

Below listed is a procedure list which must be followed for each participant participating in the testing. These tasks and steps are for privileged user class.

1. Participants is given access with username and Password
2. Participants is given training for the Interface
3. Participants must complete the tasked listed in Usability Task section.

The task list below is for the non-privileged users who are simply being asked to complete the guest forms. These participants shall be picked at random and will not receive any train. They shall be asked to complete the tasks listed in the usability tasks section.

The Administrator shall be provided with the username and password for the server and site hosting. The Administrator shall complete the tasks listed in the usability tasks section.

5.2.4 Ethics

All persons involved with the usability test are required to adhere to the following ethical guidelines:

- The performance of any test participant must not be individually attributable. Individual participant's name should not be used in reference outside the testing session.
- A description of the participant's performance should not be reported to his or her manager, or be otherwise made publicly available.
- Privileged-users shall not share their username and password with anyone else.

5.3 Usability Tasks

The software shall have different tasks to conduct for the different user classes. The participants shall follow the procedure and proceed to complete the tasks listed. The user shall either pass the test or fail it. Below the user classes have been divided into three different classes to test their usability.

Tasks for Privileged Users:

Goal	Task	Pass	Fail
Initiate Action	Log in		
Locating form	Go to Search Page		
Query	Search for all Choir Members		
Query	Search for Brandon in Attendee Table		
Complete Action	Print the list for Ignite Retreat		
Initiate Action	Run AI to suggestion for Attendance of 20		
Query	Search for Bride Name Jessica		
Complete Action	Check on Project Progress		
Update	Update Hours on Project		
Initiate Action	Add New Program to Database		

Figure 44 Task list for Privileged Users

Tasks for Non-Privileged Users:

Goal	Task	Pass	Fail
Find Form	Find Attendee form on Site		
Fill out Form	Complete Attendee form		

Complete Action	Submit the Attendee form		
Find Form	Find Wedding form		
Fill out Form	Fill out wedding form		
Try Navigation	Go Back on Wedding form to previous page		
Complete Action	Complete and Submit Wedding form		
Incomplete Action	Try to Enter email in phone input		
Incomplete Action	Submit form without information		

Figure 45 Task list for Non-Privileged Users

Tasks for Administrator:

Goal	Tasks	Pass	Fail
Navigation	Log in to hosting		
Navigation	Log in to PHP Admin		
Initiate Action	Change password		
Action Completion	Check status of server		
Navigation	Go to Web files folder		
Action Completion	Turn server on		
Action Initiate	Update hosting software		
Action Initiate	Troubleshoot web forms not submitting		
Initiate Action	Add new username and password from database		
Initiate Action	Delete username and password from database		

Figure 46 Task list for Administrator

5.4 Usability Goals

5.4.1 Completion Rate

Completion rate is the percentage of test participants who successfully complete the task without critical errors. A critical error is defined as an error that results in an incorrect or incomplete outcome. In other words, the completion rate represents the percentage of participants who, when they are finished with the specified task, have an "output" that is correct. Note: If a participant requires assistance to achieve a correct output then the task will be scored as a critical error and the overall completion rate for the task will be affected.
A completion rate of 90% is the goal for each task in this usability test. Completion rate testing shall be completed for all the tasks listed in the usability tasks section. Each user class shall have separate completion rate which will be averaged to get a averaged completion rate for the system.

5.4.2 Error-free Rate

Error-free rate is the percentage of test participants who complete the task without any errors (critical **or** non-critical errors). A non-critical error is an error that would not have an impact on the final output of the task but would result in the task being completed less efficiently. An error-free rate of 80% is the goal each task in this usability test. Error-free testing shall be completed for all the tasks listed in usability tasks. Each user class shall have separate error-free rate and it shall be defined for that class only.

5.4.3 Time to Task

The time to complete a scenario is referred to as "time on task". It is measured from the time the person begins the scenario to the time participant signals completion. Below we have listed for different times for some of the tasks participants are being asked complete.

Task	Time	Rational
Form Completion	3 minutes – Typing speed of person	If depends on each user typing speed but at most it should be 3 minutes
Query	2 minutes	This form is mostly drop down selections thus it should be done quickly
Log in	1 minute	Only require username and password and submit
Update hours	2-4 minutes	Depends on if user has to add new project or simply add more hours
Check on server	2-3 minutes	Simply log in and check

Figure 47 Time to Task for Different tasks

5.4.4 Subjective Measure

Subjective opinions about specific tasks, time to perform each task, features, and functionality will be surveyed. At the end of the test, participants will rate their satisfaction with the overall system. Combined with the interview/debriefing session, these data are used to assess attitudes of the participants.

- Participants shall be asked about consistency in color, layout, capitalization and same action for similar situations.
- Participants shall be asked about universal usability – meaning regardless of any variations the software should work the same on all devices, platforms and for people.
- Participants shall be asked to give feedback to user for every action – this includes helpful information and even give visual representation for objects and action.
- Participants shall be asked about dialogs and menus to confirm if we have a completed action meaning it should have beginning, end and middle
- Participants shall be about indictors which allow them to move forward and backwards within the site.
- Participants shall be asked about design and error making – to test how easy or difficult is it for users to make mistakes.
- Participants shall be asked to check if they can undo their action and user control is maintained throughout the system.
- Participants shall be asked about memorizing necessaries pieces of information from one screen to another and if it was possible to do so or not.

5.5 Supporting Documents

Below we have listed the survey users will be asked to complete. All the questions will be Boolean question except for those which require time or number of errors.

1. Forms are clearly displayed and easy to find.
2. Information is clearly visible and understandable.
3. Information is written in style which is easy to follow.
4. Colors on the site are complementing.
5. Screens have the right amount of information.
6. It is easy to find my way around the site.
7. The typography is attractive.
8. It is clear how screen elements work.

9. My mistakes were easy to correct.
10. It is easy to remember where to find things.
11. How long did it take you to complete the form?
12. How many mistakes did you make before submission?

6. User Manual

6.1 Introduction

6.1.1 Purpose

The user manual is supporting document to assist the users in learning to use the software and reference during usage. The manual will provide information about how to use the software as well information on the different functionality available and its maintenance.

6.1.2 Overview

The Campus Ministry System is a software developed to assist Campus Ministry office in collection, storage and retrieval of programming information. The Software includes a Web interface for users to input information and for staff members to retrieve information from the database. The software also includes a Machine Learning Algorithm which retrieves data from database to predict when a program should be held to achieve specific attendance.

6.1.3 Organization of the Manual

The user manual includes information about the system, how to description for each component for the software and software configuration information. The user manual includes information and resources to Troubleshoot problems when the occur.

6.1.4 Acronyms and Abbreviations

This section includes definitions for word usage within this section distinct to this software and the applications used in the development.

- Web Interface – website where all the guest forms are available and user login is also available
- MySQL – Open Source Relational Database Management System
- PHP: server-side scripting language
- Guest – public and students who attend campus ministry events
- Staff – interns and staff employed by campus ministry
- AI – Artificial Intelligence
- MLP – Multilayer Perceptron
- PCA – Principal Component Analysis
- Data Cleaning – remove null and empty values from data
- Python: general purpose programming language

6.2 System Summary

This section provides a general overview of the system written in non-technical terminology. The summary outlines the use of the system and activities supported for guest user and staff.

6.2.1 System Configuration

The Campus Ministry Software is designed to work on any device and any browser regardless of version or screen size. The software requires internet connection either via Wi-Fi or Ethernet. Once the user is connected to internet, the user can access guest forms and log in pages from anywhere. The software is hosted on x10hosting website which provides free hosting service for limited number of databases.

6.2.2 Data Flows

The software has two main different components when interacting with the data. The front end is the web interface where information is being collected through input boxes. The information is then retrieved from the web page using PHP script and reading in the inputs. The information is read and processed into MySQL insert statement and stored in the database. To retrieve the information the once logged in, a staff member will select which table to search from and afterwards they have the option to specify their search further or have the whole table be returned. The user selection is read in using a PHP script, formatted in MySQL Select statement and processed to the database. The database returns results which are printed to screen in a formatted table. The Machine Learning Algorithm retrieves data from database using a Python script which connects to the database and retrieves the information.

6.2.3 User Access Levels

This software includes two different user access groups. The guest users are those who only have access to web input forms where they are being asked to enter their data to provide to campus ministry. The staff users are those who will have access to query data and use this data for programs. These users require authenticated access to the site to be able to retrieve the information from the database.

6.2.4 Contingencies and Alternate Modes of Operation

In the event that the database server or web server is down, the staff user shall have to wait until they become available again to access the data. The Guest users shall also wait until the servers are available again to access the forms. The hosting server should not go down but due to maintenance issue if it is offline, users shall be informed of such occurrences and users will make other arrangements – such as to collect information different way and save to database at later time.

6.3 Getting Started

6.3.1 Guest User Forms

The guest forms are those being completed by guest users to provide their information to campus ministry. These forms can be accessed on the main campus ministry software site and will be available at all times. The user is able to fill out these forms at any point and their information will be saved in the database. There are different forms for different types of guest users – a separate form is available for Program Attendees, Retreat Attendees, Wedding Programs and Chapel Choir Members. Another form is available if a person wishes for campus ministry to contact them which can be filled out. These forms have required fields which must be completed to be able to be submitted otherwise it will prompt user to the missing field. For example, if user forget to enter their email, which is required field, they will be prompted to enter required information before they can continue. The image below shows how the guest forms look.

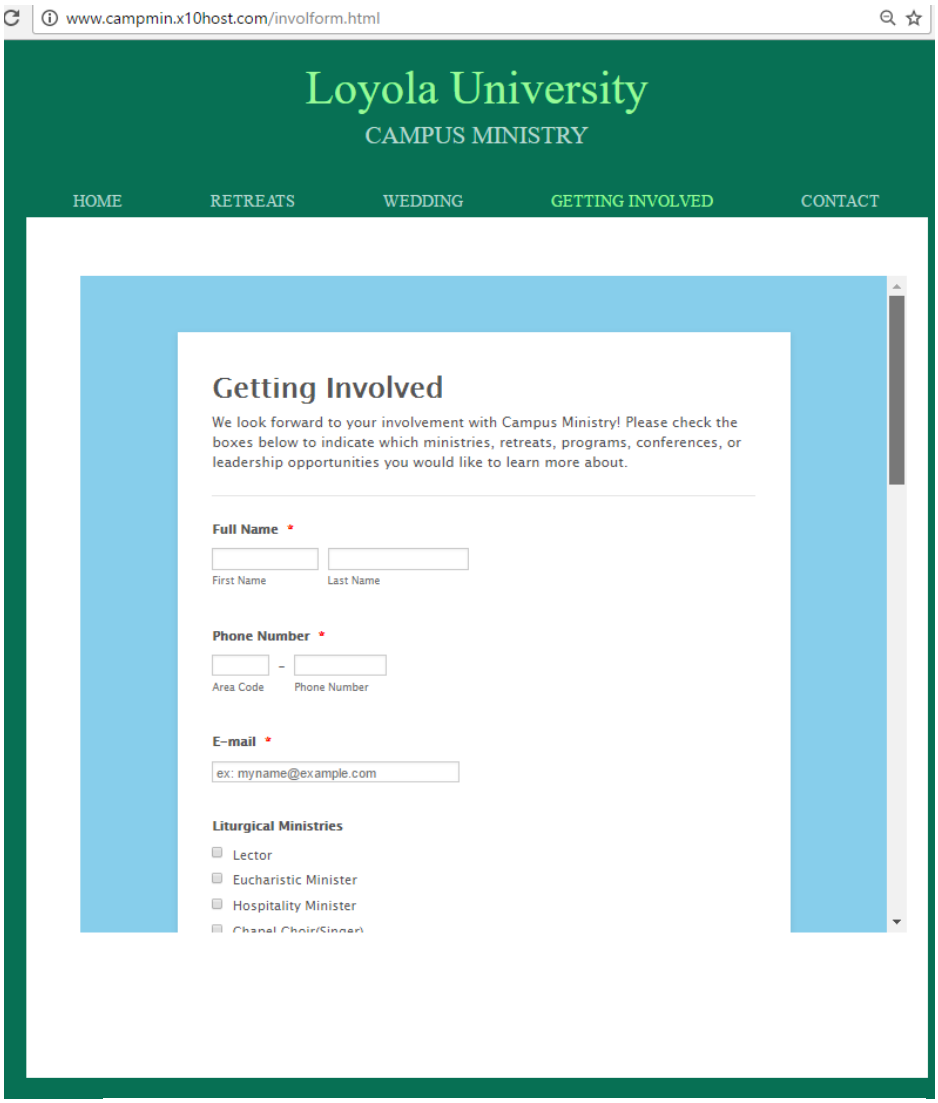

Figure 48 Guest Form Display

6.3.2 Logging On

The login form is only to be used and accessible to campus ministry staff. This form takes the user to different site allowing them query data from the database to print reports and lookup information. Only the staff can have access to this information thus an authentication is required for users to be able to gain access.

6.3.3 Database Query

The staff users only logged in can query data from the database by selecting from a drop down and inputting into an input box exact information they are looking for. For example, if a user is looking for an attendee whose name was Elise. Once logged in and at the query page, the staff would select attendee table, and at where drop down, select fname and in the next drop down select either equals or contains and enter Name Elise in the box. The database will then return all the information for person in the attendee table with the name Elise. Below we have displayed how the log in webpage.

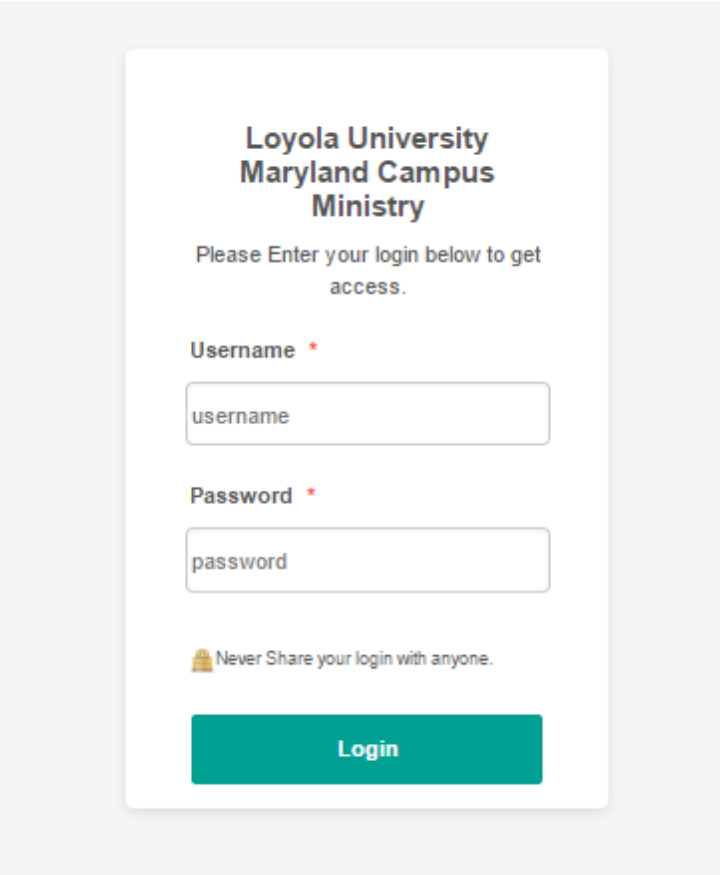

Figure 49 Log in Page Display

6.3.4 Changing User ID and Password

The username and password for the system for each user is stored in database. Only the administrator is allowed access to access the database internally using MyPHPAdmin. The Administrator can change the password and username for a user by accessing the database from MyPHPAdmin. The administrator will access this portal to not only change username and password but also add new users who require access to the system as well as removing users who are no longer have access to the database. Below we have displayed an image of how the interface for changing the password and username looks like.

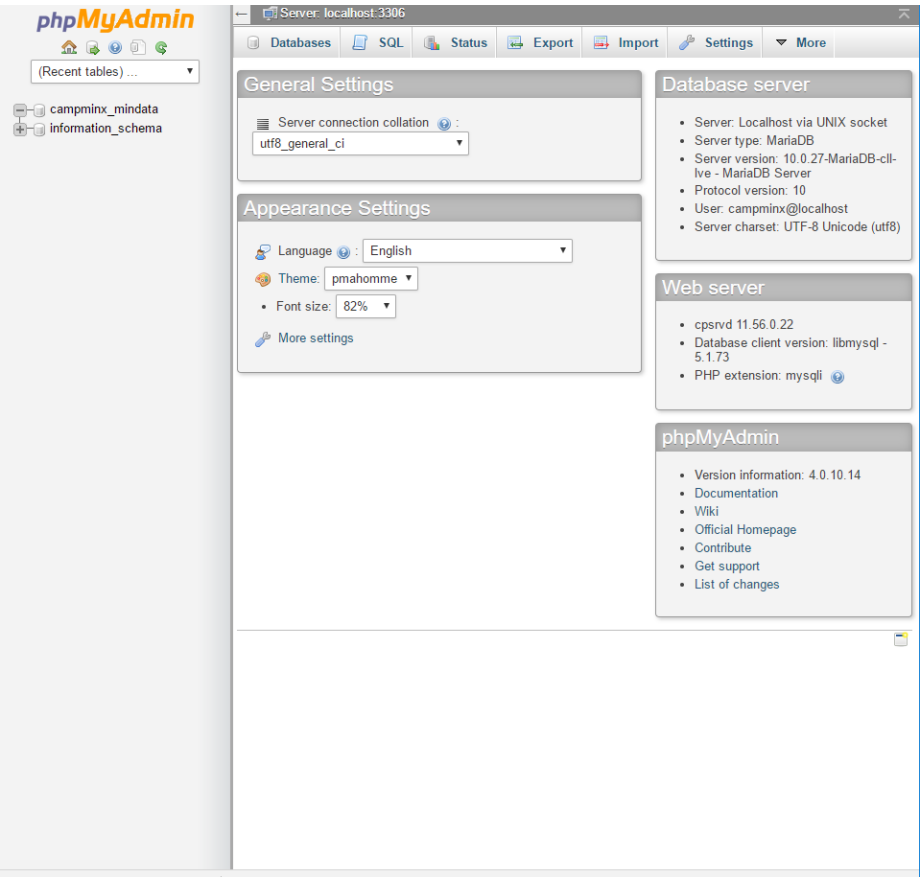

Figure 50 Username and Password Change Display

6.3.5 Exit System

In order the exit the system, both user groups can simply close the browser which was being used to access the site. The staff users can logout if their browser remembers history otherwise simply closing the browser will close the system.

6.3.6 Machine Learning Algorithm

The Machine Learning Algorithm is programmed in Python and is to be run separately from the web interface. The Machine Learning Algorithm requires a human user to run the program thus is its done separately. The algorithm retrieves data from the database and saves it to a csv file, cleans the data to remove any empty or null values. The MLP file will run PCA and MLP on the reduced data. The user shall have to run the MLP file as well the data preprocessing to test out the run the algorithm. Below, there is an image attached which displays the interface for the Machine Learning Algorithm. As depicted in the picture, it is showing a run of the MLP program which has PCA in it as well. The arrow on the image shows where the run button is for user to run the program.

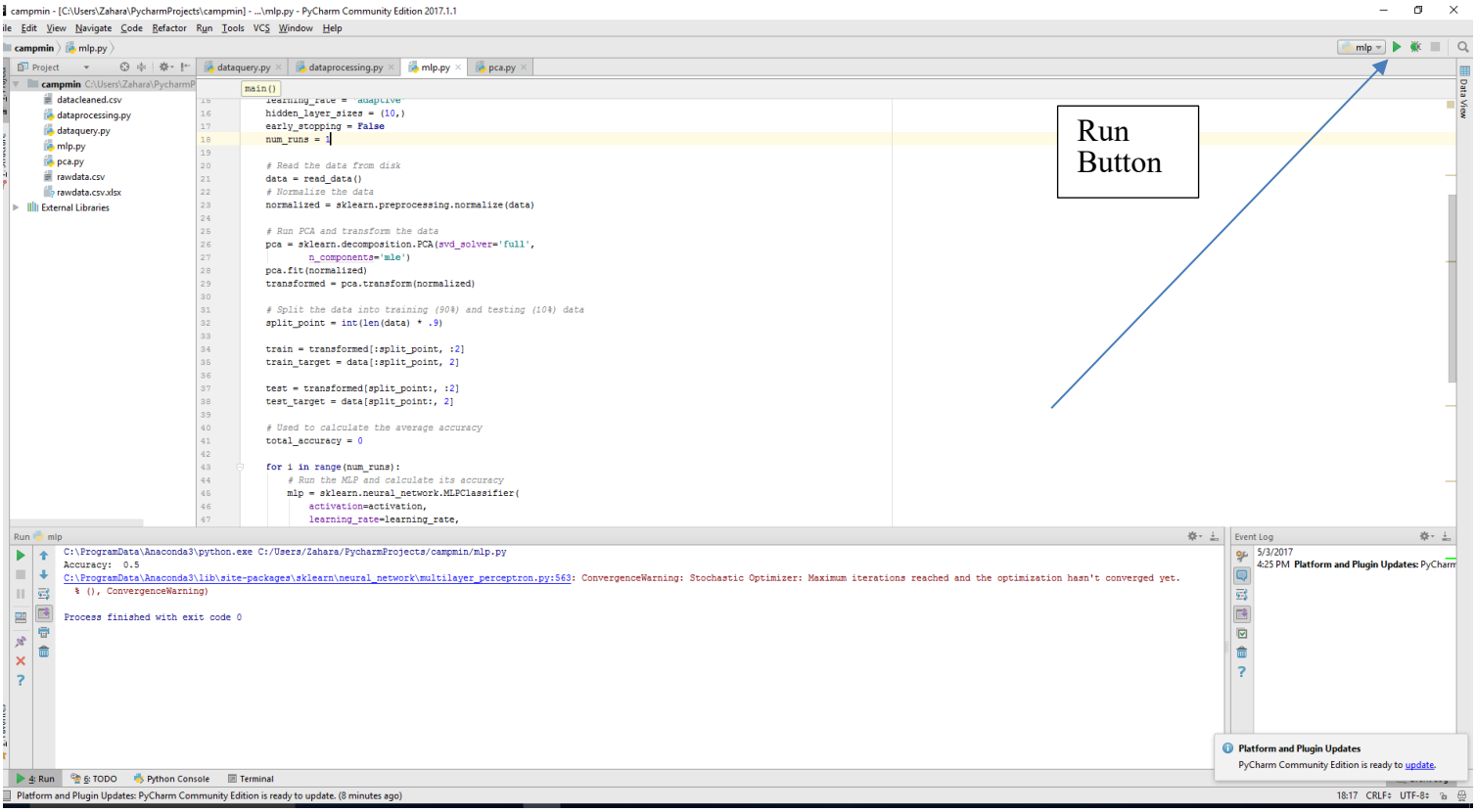

Figure 51 Machine Learning Algorithm Interface Display

6.3.7 Special Instructions for Error Correction

The webpages are developed to restrict errors such as entering wrong type of information but in the case, that user enters wrong information and submits the form. Staff users can go into the system to edit the data and delete it if necessary. The guest user can always resubmit the data with corrected information.

6.4 Project Specific Scenario

In this section, there is a scenario for each of the different components of the software. The scenario explains in details the steps to complete a given scenario.

6.4.1 Web Interface Input

Listed below is a scenario for Guest User to interact with the software and the steps which must be followed to complete an action in the web interface.

Scenario: Justin is a new member of the Chapel Choir and he has been asked to complete the online form to provide his information.

1. Justin will visit the campus ministry website at Loyola university Maryland.
2. Justin will follow the link to campus ministry database web forms.
3. He will navigate to Chapel Choir Form – labeled Chapel Choir.
4. Justin will input the required information such as his name, email, class year, phone and address.

5. Justin will complete the optional information but only which applies to him – such as does he play an instruments or what voice part he has or if he can read music or not.
6. Justin will submit the form and end up at thank you screen.

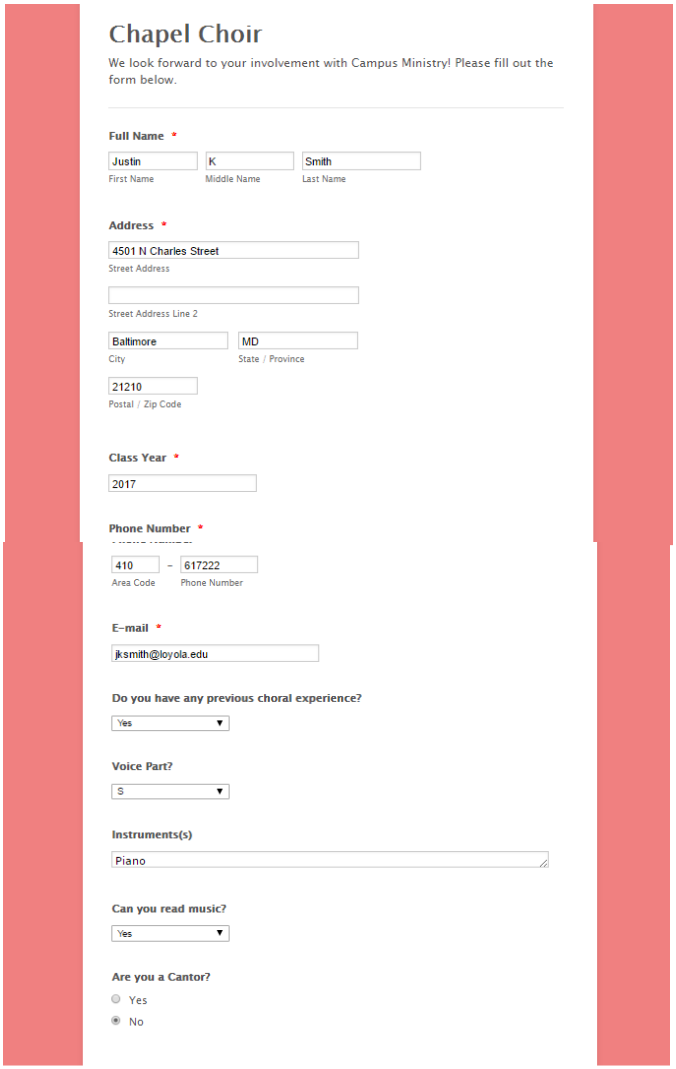

Figure 52 Web Interface Data Input Scenario

6.4.2 Database Information Retrieval

Listed below is a scenario for Staff user to access the log in page and retrieve the given scenario from campus ministry software.

Scenario: George wants to get a list of Chapel Choir Members who are class of 2017.

1. George will access the campus ministry log in page.
2. George will input his username and password.
3. Once login is successful, George will navigate to the query page.
4. George will select choir from drop down menu.
5. George will then select class year in the specification drop down.
6. George will then select equals in choice between equals, contains or not equal drop down.
7. In the input box, George will enter 2017.

8. George will hit Enter.
9. The query will return all the choir information for Chapel Choir members who are class of 2017.

6.4.3 Machine Learning Algorithm

Listed below is scenario for using the Machine Learning Algorithm and steps to follow for similar scenario.

Scenario: Check Prediction for attendance of 25 people.

1. Staff user will open the the Project on PyCharm.
2. Staff will input into MLP file what prediction they are looking for in myArray – if I am looking for attendance of 25.
3. Staff will hit run.
4. The program will return accuracy and result.
5. In result array, the first number is time and second is day.
6. Staff can run again if accuracy is low or end program.

6.5 Troubleshooting

In the section, we have listed some issues which might occur in the software and how to troubleshoot those issues when they arise. The only issues we need to be concerned about are listed below. For more issues, x10hosting can be contacted to assist with offline server or database concern. Administrator can also search online for solution and reference this document for links to helpful websites and tutorials listed in Appendix.

Issue #	Problem	Solution
1	Server not Online	Navigate to Database screen on x10hosting and click check database – it will report any errors. - If Database is online, make sure password and username is correct for database in the PHP connection file.
2	Web Server Offline	Navigate to x10hosting Cpanel - Check status server - If offline, change to status online. - Check availability of all files – should be 644, if not change it.

Figure 53 Troubleshooting and Solution Table

7. Reflection

7.1 Software Development

The process of software development is not one cohesive term and it cannot be defined through one perspective. It is an evolving and changing process. It is not only dependent on the software you are making but the people, your physical capabilities, access to material and time as well. I learned and implemented may different types of software development techniques but while doing our project, I used many different ones and combination of them all. During this development process, I learned not only how to develop a software but that software development is not only about the product and whether you are successful or not. During this development process, I have come to learn much about computer science but about my own working style and habits. I have learned to manage time better and to use the resources around me to better understand my work.

7.2 Working with a Client

In developing a product, working with a Client is very complex and hard to follow along with, especially if your client is computer science oriented. My client, Mr. Sean Bray, is one who can use modern technology but he had no familiarity with the back end of programming and databases. My initial process of gathering requirements was not a difficult one as I not only asked Mr. Sean Bray but interns on type of information they collect and what would be more useful for them. My difficulty came when the implementation and platform needed to be confirmed for the software. After waiting from a response from Client about deployment platform for software I quickly realized I needed to come up with options on my own and continue with my project.

Another concern which came up in working with my Client was that they were forced by Loyola University to change their policy about payment and hours process which then effected my project. Part of my project was to allow interns to input their hours and projects they are working on for supervisors to view and check over. Loyola University Maryland recently deployed the new Time Clock system which allows users to input their hours and projects to be approved by supervisor and processed via Payroll. Since my software is being developed as an external system to Loyola University Maryland, it has no interaction with any other office other than Campus Ministry. My Client, as result of the changed by Loyola, has requested to no longer implement that portion of the software but to instead make the website more user friendly and mobile device accessible. Although it was required for me to remove portion of my code it was adjustable but it has taught me that, this was very much a real-life experience. My Client had the authority to change requirements at any point and my overall experience of working with client was great – it taught me a lot of patience and compromise.

7.3 Designing a Software

As a Computer Science student, we are being taught in our classes to design software's and create program but the level and complexity of this software was far greater. The actual programming for the software was not complicated but the process of developing documents and presenting the project was much more complex. In Software Engineering, we learned to develop these documents but during that project everyone had similar requirements and the reader was aware of the requirement. In producing documents for this project however, all my specifications and components of the project had to be made obvious and clear otherwise it was not understood. In the development of this process I have realized that overexplaining is better than skipping details. The general process for designing this software required me to not think like a Computer Science student – who is looking for a simple solution but a Computer Science. I had to not only think of one way to solve the problem but define the problem, its constraints and several ways to approach and solve the problem. In designing this software, my Client did not limit me to specific language or format for the design but with that added constraint my development process would have become more complex.

7.4 Failures and Successes

The project involved several different components but the most complex piece was to develop the Machine Learning Algorithm. The most complex part of AI component was figuring out which component to use for the data and how it should be processed. After carefully researching different option, I choose MLP because it is used for prediction such as the one I was working with. Since I did not have data available to train or test my algorithm, I had to create random data to test on. The accuracy on pseudo data is around 50% to 59%. This data and accuracy

are not represented of the what the actual data will result in but from running the algorithm as it is, it is obvious that the Machine Learning Algorithm is a failure.

On the other hand, what I do consider a success in this project is to how the forms are designed and how restrict they are in their formatting to not allow wrong input. Designing the web forms takes the longest times as they were being designed to be compatible with any device and browser. The users who have tested the web forms so far have liked the web forms and their organization.

7.5 Tools

It was very interesting to learn and use different tools to accomplish sometimes the same task. One of the most difficult task about these tools was learning them. It became very hard because some of the tools are very new and there is not a lot of help online. Although it was hard it was learn the different tools, I enjoyed being able to know that so many tools exist for testing software and assistance in developing software. I have listed all the different tools I familiarized myself with in the development process for this software.

7.6 Extensibility of Software

The way the software is written it can very easily be extended. I implemented it in a way that if client wanted to add more functionality it would be easy to do so. The only changes we would need to make to code if no new feature is added is to keep up with updates for the software and to maintain accounts with hosting site. Otherwise, the software can function as it is even as new functionality is being added to the software.

8. Appendix

8.1 Appendix A: DBMS Type

This Database is Homogeneous DBMS because the same database will be used for all the different forms and interfaces. The database is a special purpose database and it can only be used by Campus Ministry because much of information being implemented is only used campus ministry.

8.2 Appendix B: Glossary

Below we have listed terms which are mentioned several times throughout the document.

Guest: Users who will only interact with system to input data
AI: Artificial Intelligence
MLP: Multi-Layer Perceptron
PCA: Principal Component Analysis

8.3 Appendix C: Links for Tools

This section includes the names and the links for all the variance tools mentioned in this document which will be implemented to test this software.

TeamGantt: www.teamgantt.com
Burn Down Chart: https://BurndownForTrello.com/share/dzrxlhf59e

Draw.io: www.draw.io
Apache JMeter - http://jmeter.apache.org/
Open STA - http://opensta.org/
Radview - http://www.radview.com/
FxCop - https://www.owasp.org/index.php/FxCop
FindBugs - http://findbugs.sourceforge.net/
Selenium - http://www.seleniumhq.org/
Query Surge - http://www.querysurge.com/
Tap - https://theory.github.io/mytap/
Debugger - http://mydebugger.com/
SimpleTest - http://www.simpletest.org/