Project Plan
and
Software Requirements Specification

for

Campus Ministry Software

Version 1.1 approved

Prepared by Zahara Kazmi

Loyola University Maryland

March 20, 2017

Table of Contents

Table of Figures

Revision History

Name	Date	Reason For Changes	Version
Zahara Kazmi	03/17/2018	Defining Requirements	1.1

1. Introduction

1.1 Purpose

The purpose of the document is to specify the requirements 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 above 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 is 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 in order 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 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 Gnatt 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. Project Estimate, Schedule, and Risks

2.1 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.

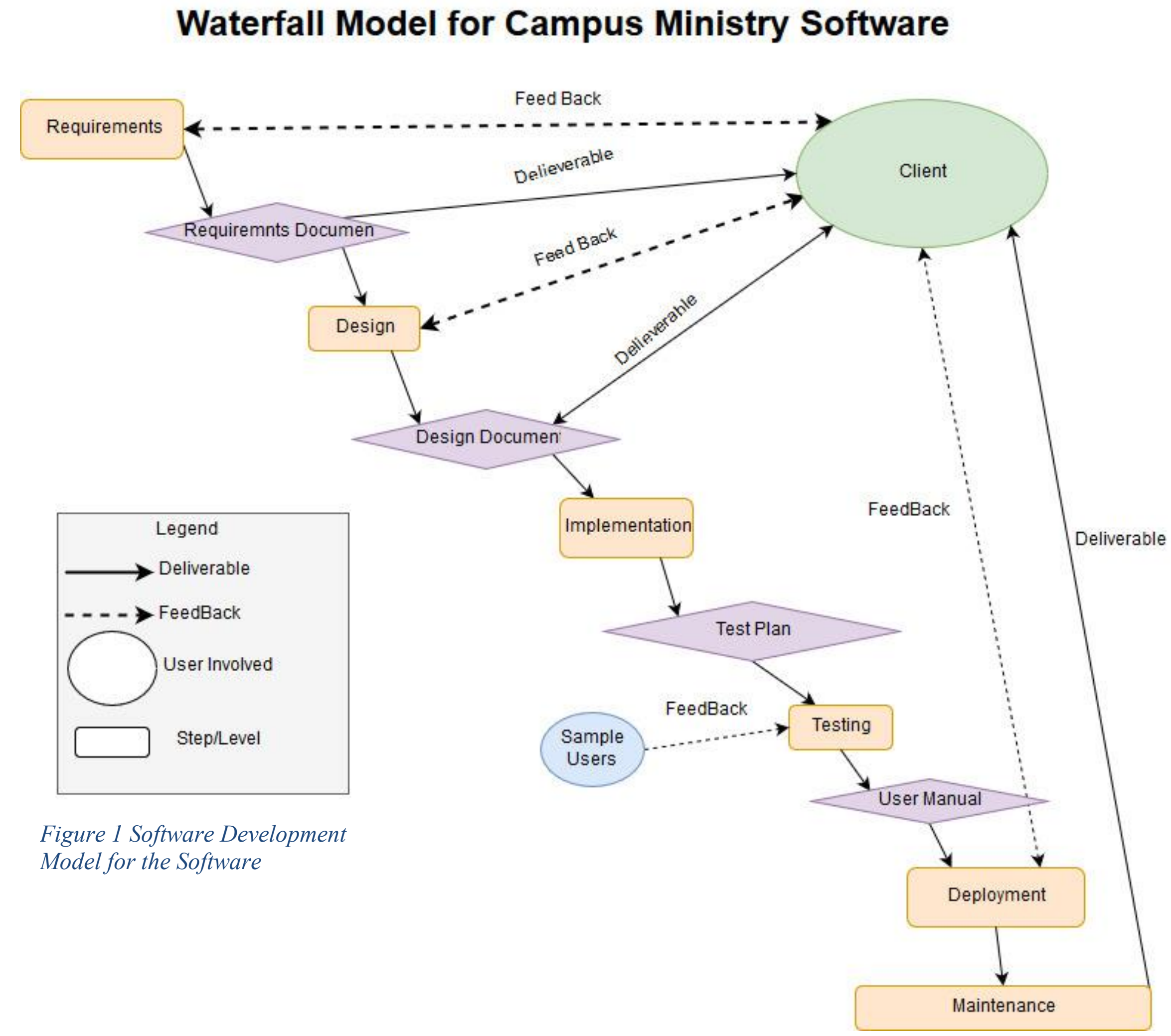
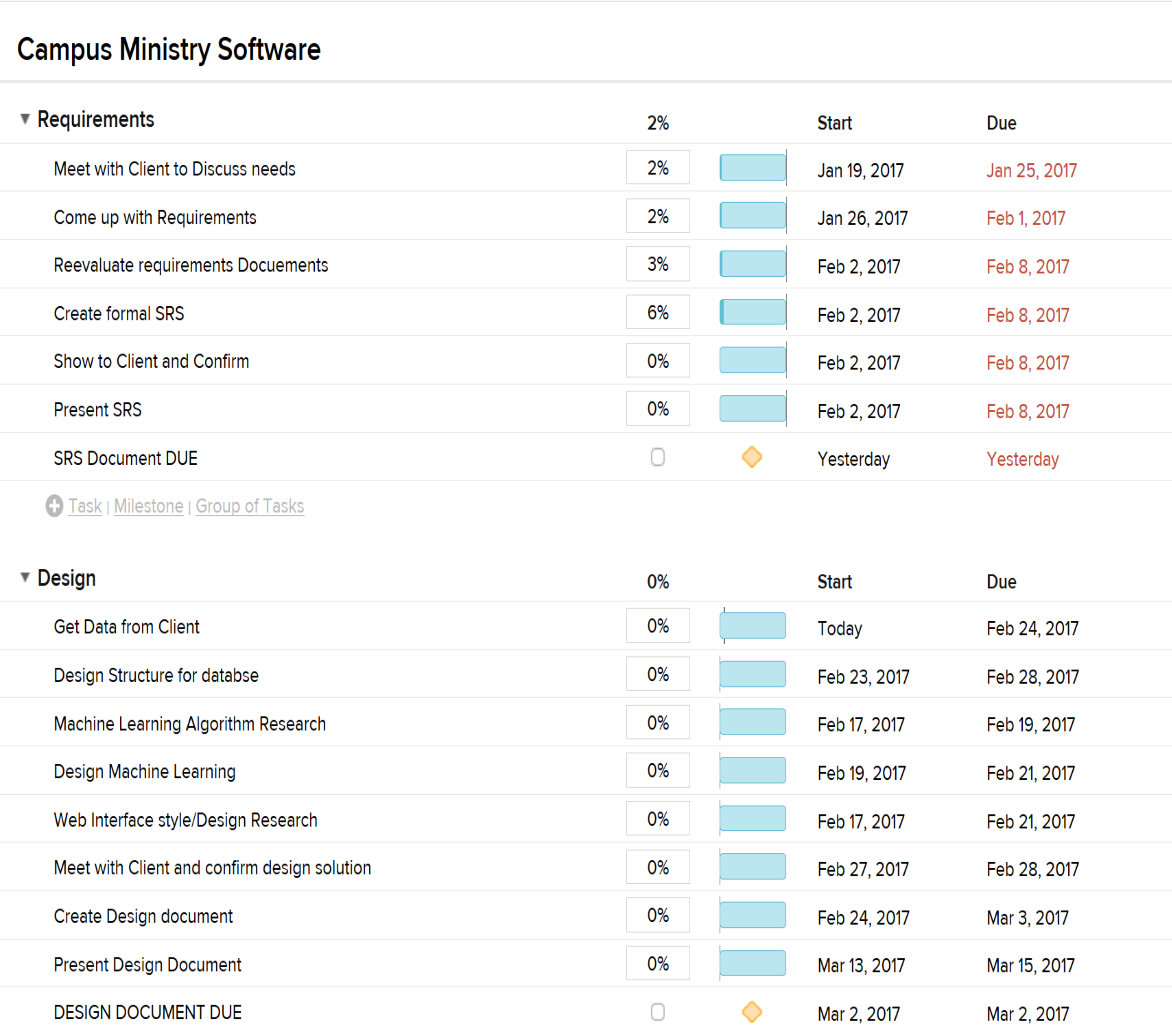


Figure 1 Software Development Model for the Software

2.2 Project Task Set (Statement of Work or SOW)

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.

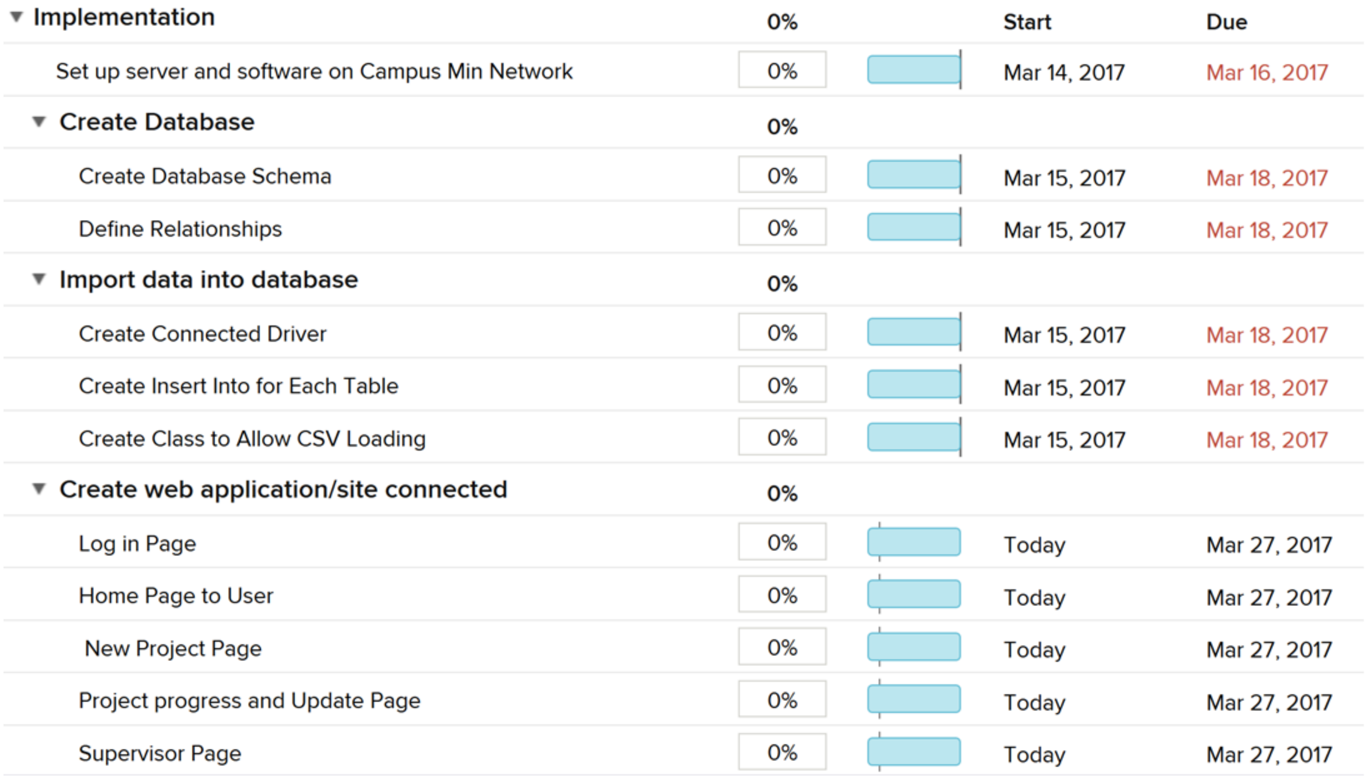


Campus Ministry Software

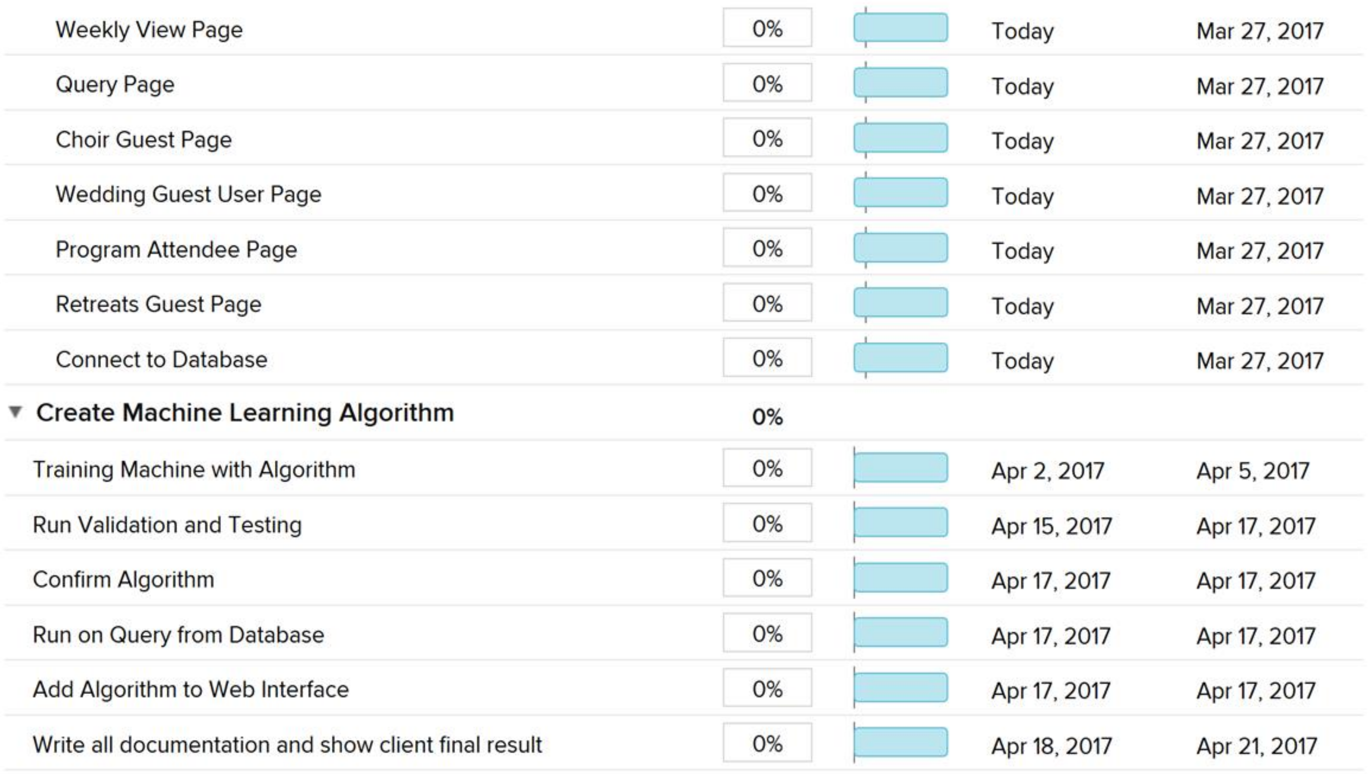
▾ Requirements	2%		Start	Due
Meet with Client to Discuss needs	2%		Jan 19, 2017	Jan 25, 2017
Come up with Requirements	2%		Jan 26, 2017	Feb 1, 2017
Reevaluate requirements Docuements	3%		Feb 2, 2017	Feb 8, 2017
Create formal SRS	6%		Feb 2, 2017	Feb 8, 2017
Show to Client and Confirm	0%		Feb 2, 2017	Feb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

⊕ Task | Milestone | Group of Tasks

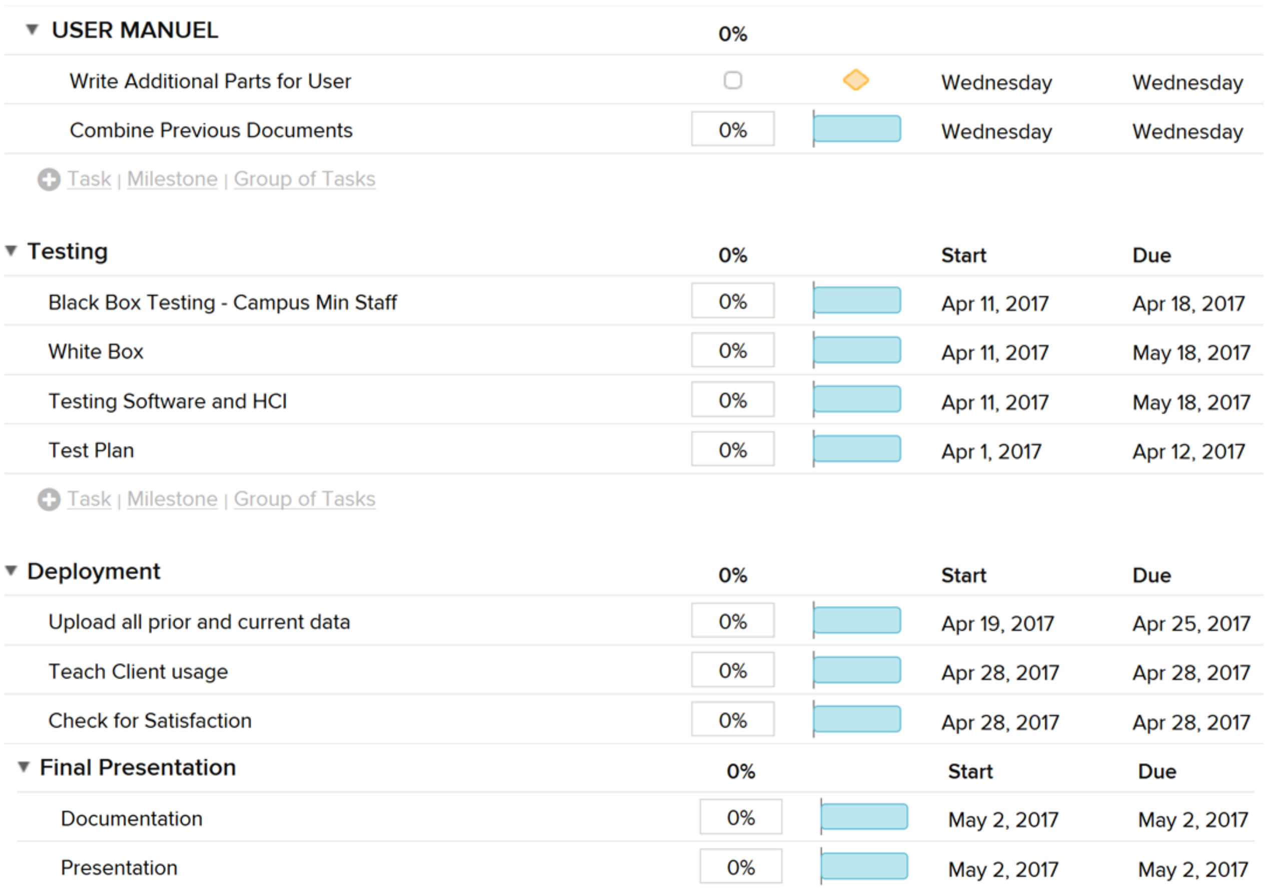
▾ Design	0%		Start	Due
Get Data from Client	0%		Today	Feb 24, 2017
Design Structure for databse	0%		Feb 23, 2017	Feb 28, 2017
Machine Learning Algorithm Research	0%		Feb 17, 2017	Feb 19, 2017
Design Machine Learning	0%		Feb 19, 2017	Feb 21, 2017
Web Interface style/Design Research	0%		Feb 17, 2017	Feb 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



▼ USER MANUEL	0%		
Write Additional Parts for User	□	Wednesday	Wednesday
Combine Previous Documents	0%	Wednesday	Wednesday

Task | Milestone | Group of Tasks

▼ Testing	0%	Start	Due
Black Box Testing - Campus Min Staff	0%	Apr 11, 2017	Apr 18, 2017
White Box	0%	Apr 11, 2017	May 18, 2017
Testing Software and HCI	0%	Apr 11, 2017	May 18, 2017
Test Plan	0%	Apr 1, 2017	Apr 12, 2017

Task | Milestone | Group of Tasks

▼ Deployment	0%	Start	Due
Upload all prior and current data	0%	Apr 19, 2017	Apr 25, 2017
Teach Client usage	0%	Apr 28, 2017	Apr 28, 2017
Check for Satisfaction	0%	Apr 28, 2017	Apr 28, 2017

▼ Final Presentation	0%	Start	Due
Documentation	0%	May 2, 2017	May 2, 2017
Presentation	0%	May 2, 2017	May 2, 2017

Figure 2 Statement of Work

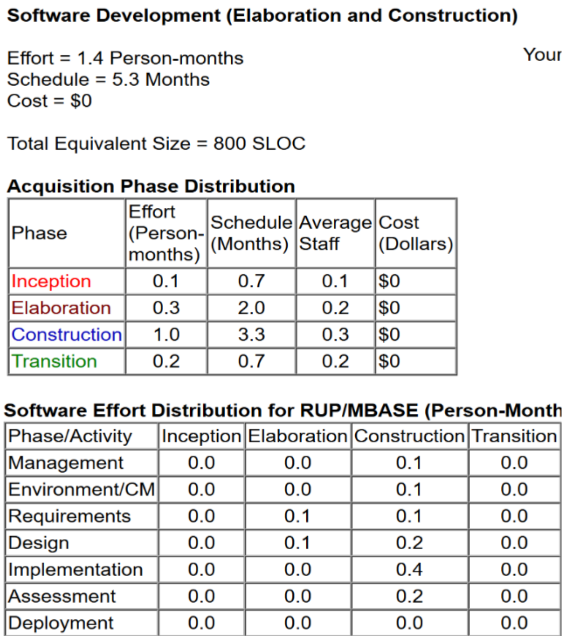
2.3 Estimation Techniques Applied and Results

In order 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)

Your

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

Total Equivalent Size = 800 SLOC

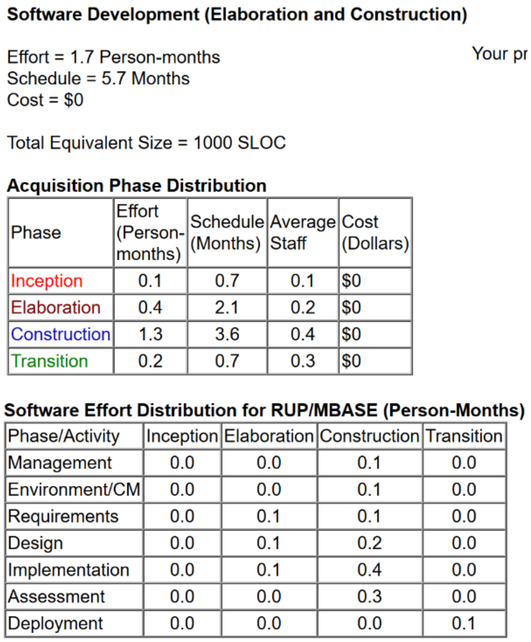
Acquisition Phase Distribution

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)

Your pr

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

Total Equivalent Size = 1000 SLOC

Acquisition Phase Distribution

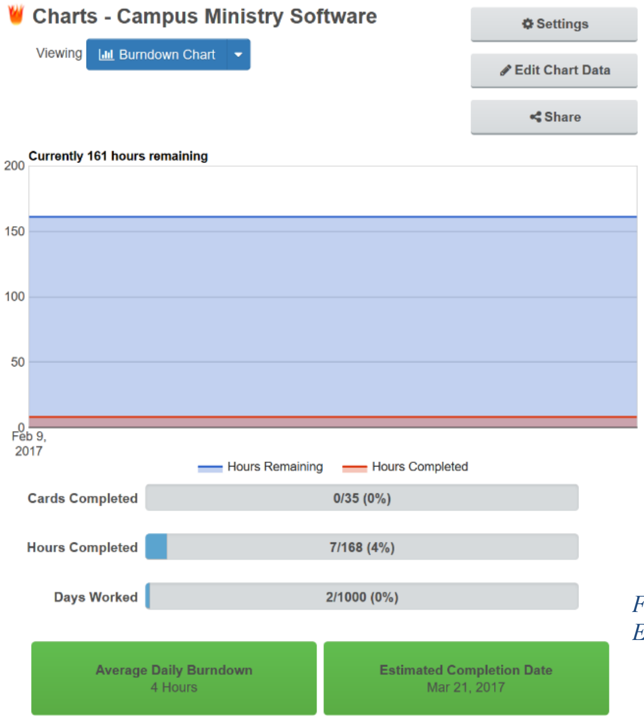
Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	0.1	0.7	0.1	$0
Elaboration	0.4	2.1	0.2	$0
Construction	1.3	3.6	0.4	$0
Transition	0.2	0.7	0.3	$0

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.



Charts - Campus Ministry Software

Viewing Burndown Chart

Cards Completed 0/35 (0%)

Hours Completed 7/168 (4%)

Days Worked 2/1000 (0%)

Average Daily Burndown 4 Hours

Estimated Completion Date Mar 21, 2017

Figure 5 Trello Burn Down Chart Estimation

2.4 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.

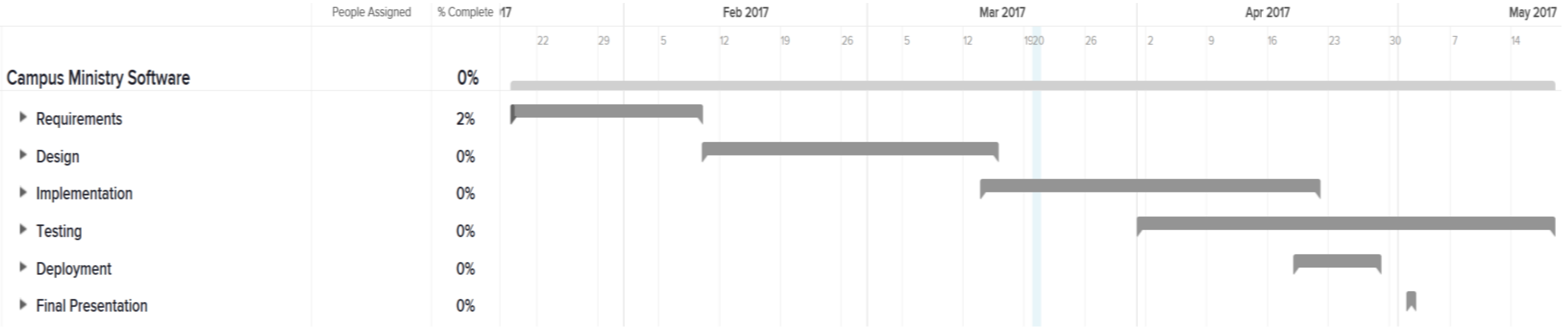


Figure 6 Gantt Chart Overall Project Overview.

2.5 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 in order 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

3. Overall Description of the Product

3.1 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.

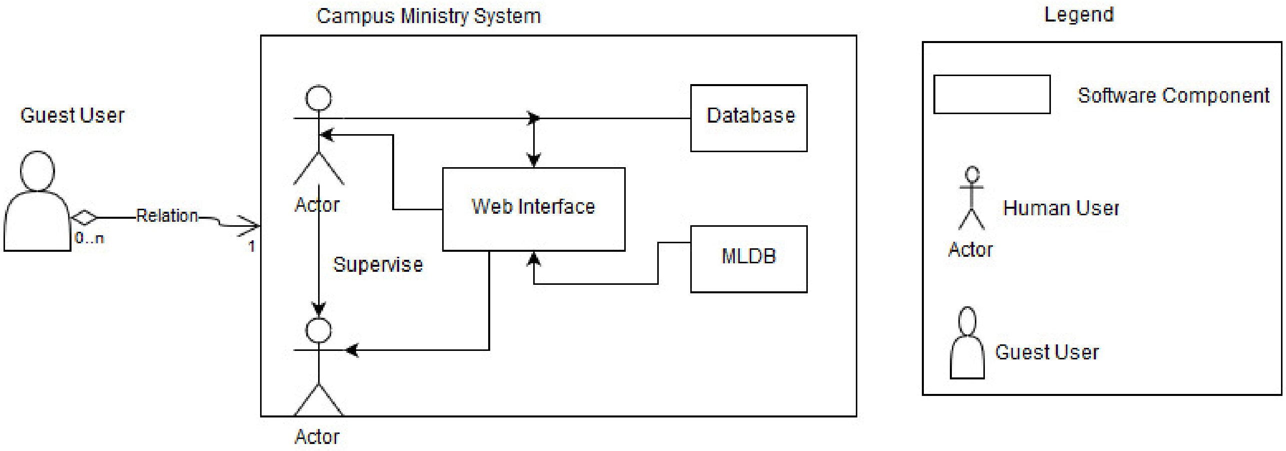


Figure 8 Product Perspective for Users Diagram

3.2 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.

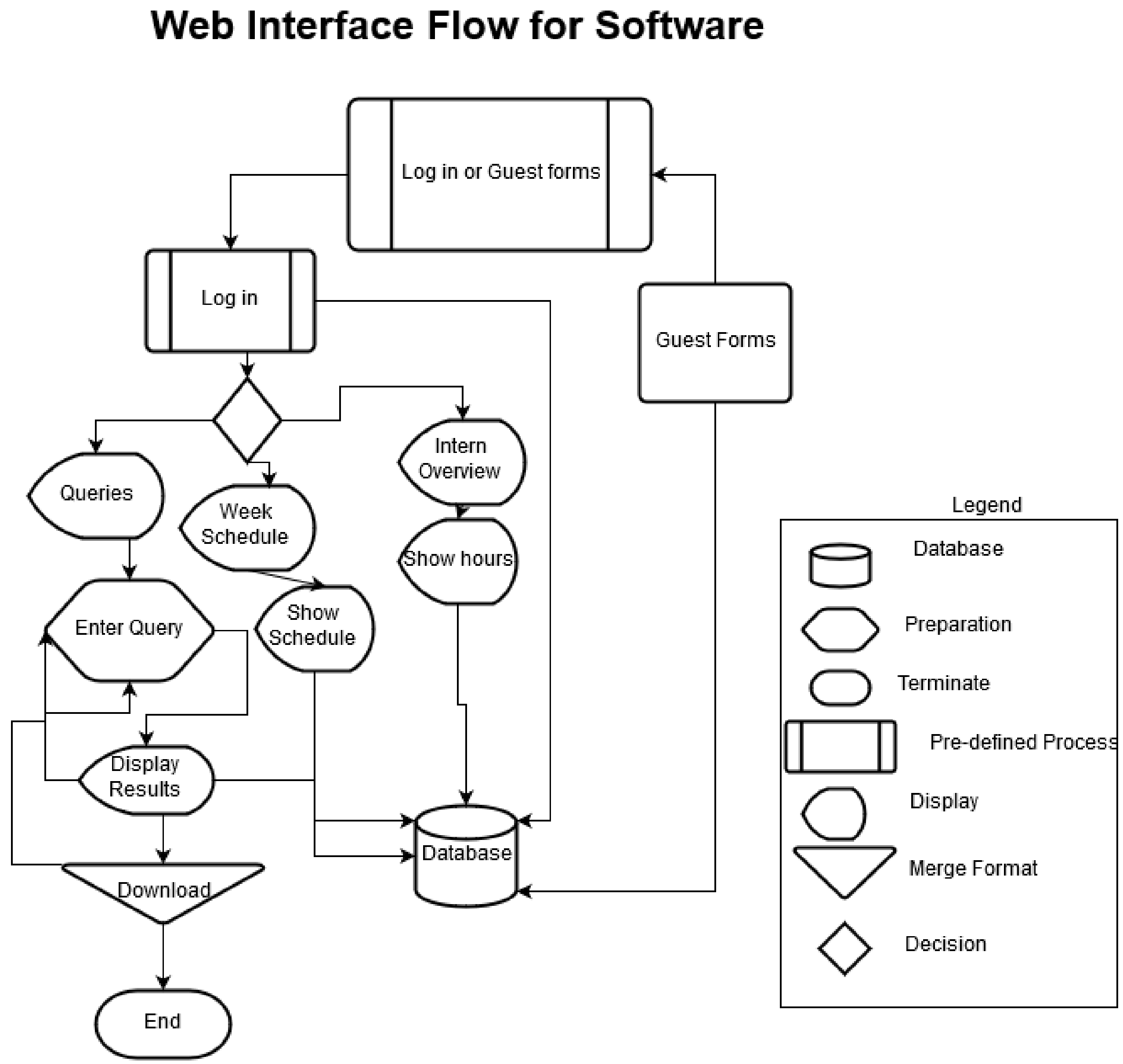


Figure 9 Product Functions Representation Flow

3.3 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

3.4 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.

3.5 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 in order to protect the information. This system will be implemented using MySQL, WampServer64, PHP and Machine Learning Algorithm.

3.6 User Documentation

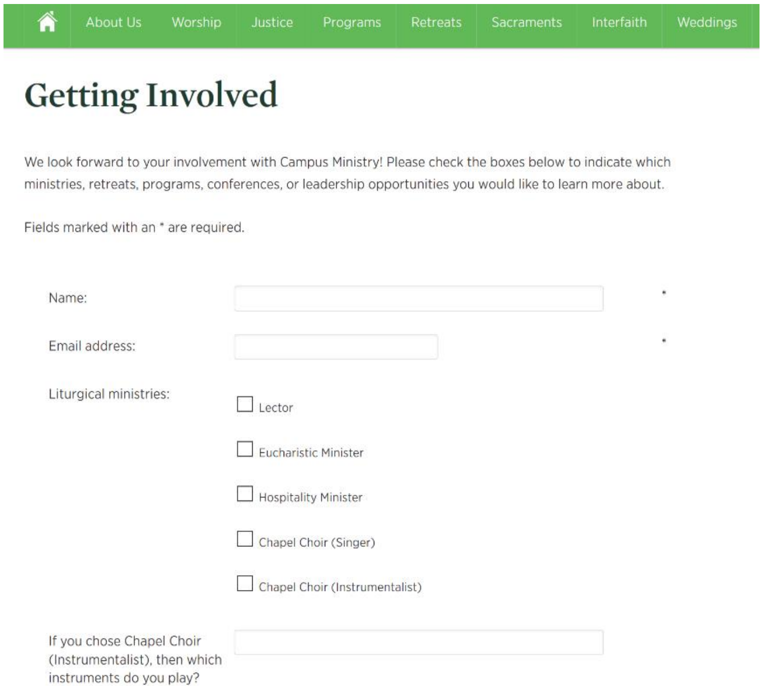
In part with the software, several different documentation compoenets will be developed and delivered to the client. Some of the documentation will be very formal and designed for developers or Engineer to further enhance the system. Since the Client does not have a technical background we will be developed 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.

3.7 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 in order 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 at all times.

4. External Interface Requirements

4.1 User Interfaces



Getting Involved

We look forward to your involvement with Campus Ministry! Please check the boxes below to indicate which ministries, retreats, programs, conferences, or leadership opportunities you would like to learn more about.

Fields marked with an * are required.

Name: *

Email address: *

Liturgical ministries:
☐ Lector
☐ Eucharistic Minister
☐ Hospitality Minister
☐ Chapel Choir (Singer)
☐ Chapel Choir (Instrumentalist)

If you chose Chapel Choir (Instrumentalist), then which instruments do you play?

Figure 11 User Interface Sample

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 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.

4.2 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.

4.3 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.

4.4 Communications Interfaces

This system will implement several different communication interfaces in order 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 at all times to not cause data discrepancies and also to continuously update. This system will be using HTTP communication standards for all of its functionality and for communication with database.

5. 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.

5.1 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.

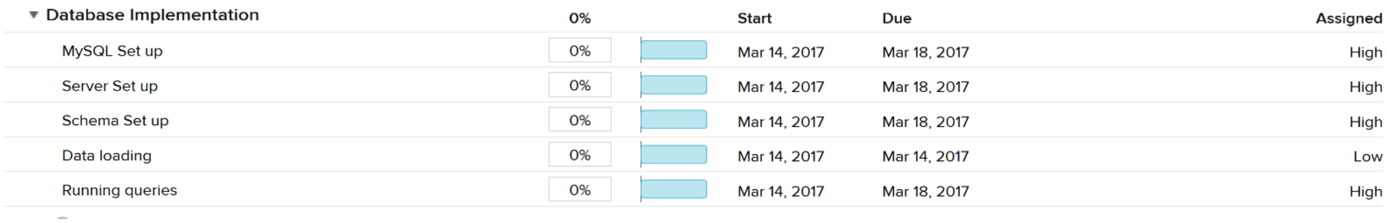
User Stories			
User Story	Feature	Need	Priority
As Elise, I want to be able to search for students with Hindu background who are interested in Interfaith to expand initiative on campus.	Database and Web Interface	Be able to run queries for specific students without knowing MYSQL syntax	High
As Sean, 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 a 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

5.2 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.



▾ Database Implementation	0%	Start	Due	Assigned
MySQL Set up	0%	Mar 14, 2017	Mar 18, 2017	High
Server Set up	0%	Mar 14, 2017	Mar 18, 2017	High
Schema Set up	0%	Mar 14, 2017	Mar 18, 2017	High
Data loading	0%	Mar 14, 2017	Mar 14, 2017	Low
Running queries	0%	Mar 14, 2017	Mar 18, 2017	High

Figure 13 Database Implementation Description Tasks

5.2.2. Stimulus/Response Sequences

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

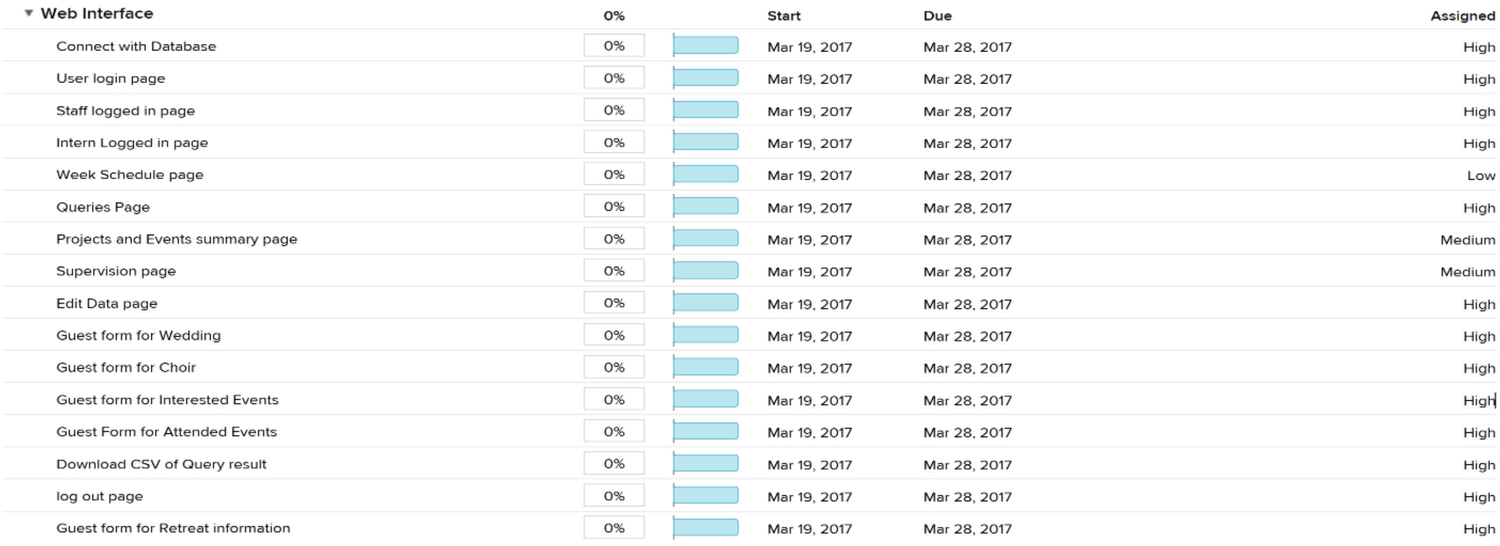
REQ-2: Interns shall work on project and events in order to report their hours and get paid.

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.

5.3 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 a 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.



▾ 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 form for Choir	0%		Mar 19, 2017	Mar 28, 2017	High
Guest form for Interested Events	0%		Mar 19, 2017	Mar 28, 2017	High
Guest Form for Attended Events	0%		Mar 19, 2017	Mar 28, 2017	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

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.

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 goto 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 Goto 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.

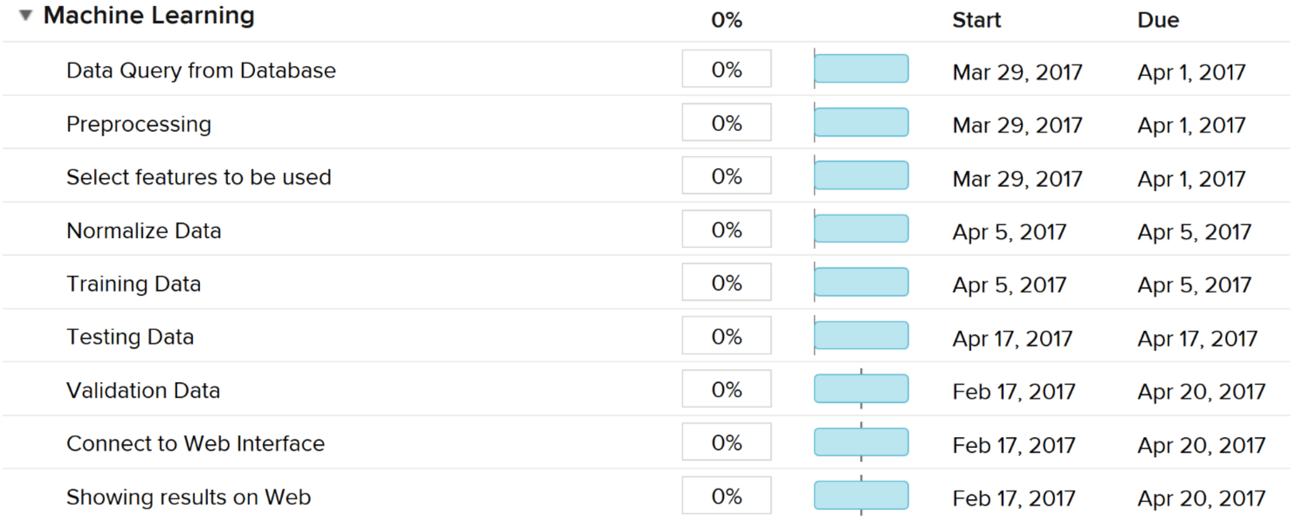
REQ-18: An interface shall exist for interns to update weekly hours so that they can report how many hours they have worked for payment.

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

5.4 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.

6. Other Nonfunctional Requirements

6.1 Non-Functional Requirements

6.1.1 Performance Requirements

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.

6.1.2 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.

6.1.3 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.

6.1.4 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.

6.1.5 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.

6.1.6 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.

6.1.7 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.

6.1.8 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.

6.1.9 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.

6.1.10 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.

6.1.11 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.

6.1.12 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.

6.1.13 Accessibility

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 as long as they have the accessible links and login information.

6.1.14 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.

6.1.15 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.

6.1.16 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.

6.1.17 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.

6.1.18 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 in order 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.

6.2 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 in order to be deemed reliable. The system should have all the functionality listed in the requirements in order 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.

6.3 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.

7. Appendix

Appendix A: Glossary

Guest: Users who will only interact with system to input data

MLDA: Machine Learning Algorithm

Appendix B: Links for Websites

TeamGantt: **www.teamgantt.com**

Burn Down Chart: **https://BurndownForTrello.com/share/dzrxlhf59e**

Draw.io: **www.draw.io**