Inception Spec Note

The following document is a compilation of incremental iterations. From the beginning, we formed an agile team of six people and worked through two iterative processes under the direction of Dr. Barker, which led us to the inception specification. Each iteration addressed specific tasks that were gradually refined as we moved along. We hope to have addressed Adelante Hispanic Achievers business needs, or at least be on the right path of a proposed solution. Your feedback would greatly be appreciated.

Thank you,

S.A.D Associates.

Elaboration Spec Note

Thank you for the opportunity to partner with your organization this semester, our team really appreciates the relationship we have built. We have enjoyed being able to uncover opportunities for changes at Adelante and appreciate your organization fully welcoming us. At times, change can be daunting and uncertain, but through our open communication and your dedication, we have built a solid foundation for revamping the current AHA system. We hope that this semester has been as rewarding for us as it has been for you and look forward to deepening our business relationship in the future. Please, do not hesitate to reach out to us with any questions.

Thank you,

S.A.D Associates

Table of Contents

I. Inceptoion Specification

1 System Request Error! Bookmark not defined.

1.1 Project Sponsor ***Error! Bookmark not defined.***
1.2Business Needs ***Error! Bookmark not defined.***
1.3Business Requirements ***Error! Bookmark not defined.***
1.4Business Value ***Error! Bookmark not defined.***
1.5Issue Constraints ***Error! Bookmark not defined.***

2 System Requirements Error! Bookmark not defined.

2.1Requirement Hierarchy ***Error! Bookmark not defined.***

2.1.1 Functional Requirements **Error! Bookmark not defined.**

2.1.2 Non-Functional Requirements **Error! Bookmark not defined.**

2.1.3 Requirement Details **Error! Bookmark not defined.**

3 Initial Architecture Considerations Error! Bookmark not defined.

3.1Design Viewpoint ***Error! Bookmark not defined.***
3.2Realization Viewpoint ***Error! Bookmark not defined.***

4 Team Charter Error! Bookmark not defined.
4.1Mission Statement ***Error! Bookmark not defined.***
4.2Team Goals ***Error! Bookmark not defined.***
4.3 Team Meetings
Error! Bookmark not defined.
4.4Team Communication ***Error! Bookmark not defined.***
4.5Team Decisions ***Error! Bookmark not defined.***

Appendix A: Vision Document Error! Bookmark not defined.

Appendix B: Process Models Error! Bookmark not defined.

Appendix C: Risk Analysis Error! Bookmark not defined.

Appendix D:High Risk Use Cases Error! Bookmark not defined.

Appendix E: Risk Trace Matrix Error! Bookmark not defined.

Appendix F: Inception Phase Prototypes Error! Bookmark not defined.

Appendix G: Gantt Chart Error! Bookmark not defined.

Appendix A: Vision Document

Inception Specification

Vision Document

Version <3.0>

Adelante Hispanic Achievers

Revision History

Date	Version	Description	Author
01/20/2016	1.0	Version 1.0	Maam Awa Ndaw
02/12/2016	2.0	Updated version 1.0: Appendix B	Maam Awa Ndaw
02/20/2016	3.0	Updated version 2.0: Appendices A &B	Maam Awa Ndaw

Adelante Hispanic Achievers

Table of Contents

1. Introduction **6**

1.1References *6*

2. Positioning **7**

2.1Problem Statement *7*

2.2Product Position Statement *10*

3. Stakeholder and User Descriptions **11**

3.1Stakeholder Summary *14*

3.2User Summary *15*

3.3User Environment *16*

3.4Summary of Key Stakeholder or User Needs *18*

3.5Alternatives and Competition *20*

4. Product Overview **20**

4.1Product Perspective *20*

5. Product Features **21**

6. Other Product Requirements **23**

7. Appendix A- Hardware and Software **25**

8. Appendix B: Feasibility Study **26**

Adelante Hispanic Achievers

Vision Document

Introduction

The purpose of this document is to collect, analyze, and define high-level needs and features for the system to be proposed to Adelante Hispanic Achievers. It focuses on the capabilities needed by the stakeholders and the target users, and why these needs exist. The details of how the proposed system fulfills these needs will be established in our analysis in the upcoming sections of this document. At first, we will define the problem that Adelante in currently experiencing with their current system along with any consequential barriers. Second, we will discuss the stakeholders and end-users, their responsibilities, environment and needs. At last, we will touch upon the specifications requested, as well as additional product features that would be necessary, along with the benefits and potential associated with the proposed system.

1.1 References

This document is prepared upon gathering insightful information from Adelante's 2014/2015 Annual report, IT strategic assessment reports and specifications to the concept of the future system discussed hereon. In addition, various forms were used to gain more information prior to developing a potential solution.

Report 1:	Annual Report- Adelante Hispanic Achievers Jul 2014/Jun 2015
Report 2:	IT Strategic Assessment Report_Bratcher - Dec 07,2015
Report 3:	IT Strategic Assessment Report_Farc - Nov 06- Dec 07, 2015
Report 4:	IT Strategic Assessment Report_Gardner - Dec 07, 2015
Report 5:	Adelante Hispanic Achievers CIS Website Project Specifications
Report 6:	Adelante Hispanic Achievers Registration Form 2015-2016
Report 7:	Permission for Photographs Form
Report 8:	Adelante Hispanic Achievers Behavior Contract
Report 9:	Authorization to Release Education Records and Consent Form
Report 10:	Optional Parental Survey

Adelante Hispanic Achievers

Positioning

1.2 Problem Statement

Adelante Hispanic Achievers is a non-profit organization offering sustainable after-school educational programs to middle school and high school students. The organization offers three programs; School tutoring on Wednesdays at Thomas Jefferson Middle School and on Thursdays at Lassiter Middle School, and a Saturday program open to both middle and high school students. Adelante is increasing in size, which is also increasing operating expenses but unfortunately cash inflows are decreasing affected by its current processing system.

Adelante Hispanic Achievers

The problem of	• Adelante student application process begins with a mandatory in-person registration • Attendance, volunteer and all existing forms are recorded on paper, then later manually processed in a system, which is inefficient as compared to if the entire process was automated • Adelante does not track information pertaining to individual donations, except for date and amount. • Lack of diverse, solid and reliable donor base • Disproportion between sources of revenues as more revenues are generated through grants than donations, which makes the organization easily financially unstable • Accounting Software Quickens is used for financial recordkeeping and reporting purposes but it has very limited capabilities that are not sufficient to the organization • Some fundraising activities lack strategy such as the first Phone-a-thon • Insufficient hardware to accommodate all current students • Lack of software consistency and durability • Lack of disaster recovery plan • Scattered data such as numerous attendance and sign-in sheets, with duplicates circulating between employees • Out-of-date information such as out-of-date contact list from Phone-a-thon fundraising event • Lack of Enterprise Resource Planning, Risk Assurance and Customer Relationship Management Systems led to decentralized information, high exposure to risk of failure and potential FERPA and Privacy Act violations • Insufficient IT funds to hire a Webmaster or IT specialist • Lack of a centralized communication system within the organization

affects	• Executive Members • Staff Members • New Members • Donors • Volunteers • Parents • Current Students • Potential future Students
the impact of which is	• Unable to build and maintain efficiency in day-to-day operations • Barrier to actual and sustainable growths • Threat to continuous operations
a successful solution would be	An integrated System that will ensure data centralization, accuracy, consistency and security. The System will include • Sign-up newsletter form • News transmission page • Direct Donation Link • Donor Database • Mass Communication tool • Storage and retrieval tool for all files • Volunteer sign up and Database • Interactive Calendar • Student Database • Report Generation tool • Database for grants Adelante also needs to take steps towards • Building a diverse and reliable donor base to promote financial stability and growth • Developing and maintaining relationships with donors • Increasing awareness locally

	• Investing in Accounting Software for small size companies such as sage 50 that will ensure ease of recordkeeping and reporting of financial information, while requiring a minimum skill set and some training • Raising funds to be able to support its processing needs, as well as additional staff members prior to serving the rising number of students

2.2 Product Position Statement

The product discussed previously is shaped around Adelante's current processing system needs and primarily intends to better their business processing environment. Unlike their current system, the proposed product will incorporate tools that provide the best service to Adelante. Efficiency will be maximized when conducting day-to-day operations, which ultimately contribute to growth and success in the long-run.

For	Executive and Staff Members, new Members, Students, Volunteers, Donors, Parents and potential future Students
Who	Lead, Plan, Organize, Administer, Coordinate and Recruit
The (product name)	Cloud-based Integrated System
That	• Maximize efficiency in day-to-day operations • Contribute to sustainable growth and success
Unlike	Their current partly web-based, partly manual business processing environment
Our product	• User-friendly, interactive, easy to navigate, Cloud-based integrated system • Provide the best service to the specific primary needs of data conformity, consistency and centralization Adelante.

Adelante Hispanic Achievers

Stakeholder and User Descriptions

Adelante Hispanic Achievers employs a Board of Directors, with an Executive Director who handles day-to-day activities, a Volunteer/ Program Operations and Development Coordinator, a College Readiness coach, an AmeriCorps Representative and Teachers/ Mentors. Occasional staff members such as interns, volunteers are part of the personnel as well. The primary users of to be considered in this project beyond the employees are the donors, students, potential students and parents, who are the end-users.

Stakeholders and users Profile

Executive Director

She is in charge of day-to-day operations, such as recruiting students, communicating with potential donors, schools, teachers, mentors, parents; writing grant proposals, programming volunteering activities, recruiting volunteers, interns, students, communicating with parents and potential parents, mentors, teachers, schools, coordinating and hosting fundraising events. Though, she has limited IT skills and might need training should the current system migrate to a new one.

Volunteer/ Program Operations and Development Coordinator

He is in charge of handling coordinating volunteering activities, writing grants proposals, communicating with potential donors, recruiting volunteers, interns, communicating with schools, parents and potential parents, teachers, mentors and students; tracking students' attendance, recordkeeping of students' information and hosting fundraising events. This employee is in charge of most of the current IT support and operations, thereof, will be able to work with an integrated system.

College Readiness coach

This Employee works closely with students and parents to monitor attendance, progress, prevent disciplines issues and ultimately help the student meet and achieve their career planning goals and objectives such as college admission process, scholarship opportunities, ACT or SAT test preparation, major selection, Financial Aid application process, and Resume building. As most of this employee's duties is web based, s/he will possess solid technical skills

AmeriCorps representative

This employee is in charge of giving students the opportunity to become engaged in community service through projects for a certain period of time. Essentially, this person communicates with the executive employees about available and upcoming projects that students can participate in. We are anticipating that they have some knowledge about online integrated systems.

Teachers/ Mentors

These are individuals that directly contribute to the development of the students by educating, mentoring, guiding and nurturing their intellectual needs and social and cultural heritage. They should have some training and education about similar web-based tools primarily for education as it is their profession to teach and guide students in similar programs.

Interns

These employees would be in charge of assisting the program coordinators with daily-tasks such as programming activities such as volunteering and fundraising events, gathering information for grant proposals, recording students and business information. As they are all educated individuals, it is anticipated that will have a beginner to intermediate technical skill set.

Volunteers

These employees primarily assist employees with programs or any other volunteer areas of interest like special events or translating written materials. They can be family members of students, staff and board members, interns, teachers and even Board of Directors. Most of these individuals are expected to have a minimum knowledge about web-based technologies.

Donors

These individuals donate money towards Adelante's program through fundraising events, occasional requests or voluntarily. They should have some technical skills as they might be prompt to access information regarding donations through a link.

Students

These are the ultimate end-users that benefit from the program through after-school activities such as tutoring, mentoring, cultural development and community service. These users are expected to have minimum knowledge about computers.

Potential Students and Parents

These individuals are new applicants and their parents who will users as they will might be initially looking for information through the system at first. It is anticipated that they will have some knowledge and skills about how to navigate into the system and access the information.

1.3 Stakeholder Summary

Name	Description	Responsibilities
Project Champions Executive Director Volunteer and Program Operations and development Coordinator	Initiated the system request	Support the project with time and resources Communicates and supports the project importance to other keys decision makers within the organization
Organizational Management	Informed about the project	Budget enough money for the project if possible Encourage users to accept and use the system
System Users	Will use the system once installed	Perform hands-on activities for the project

		Ultimately determine whether the project is useful or not by using the system

1.4 User Summary

For the purpose of summarizing users, defining and establishing their roles, employees with similar functions and responsibilities will be grouped together. For example, the Executive Director and the Volunteer and Program Operations and Development Coordinator will be under administrators. Please note that we are not insinuating that these two employees have the same functions or responsibilities.

Name	Description	Responsibilities	Stakeholder
Board of Directors	Individuals in charge of strategic management and some involvement to fundraising and volunteering activities	Provide an overall direction of the program, focusing on the necessity to meet goals and objectives.	Board of Directors
Administrator (s)	Employees primarily involved with operational and tactical management	Plan, organize, manage and control business operations, while meeting goals and objectives	Executive Director Volunteer and Program Operations and Development Coordinator
Staff Members	Employees who assist with volunteering and fundraising activities, teaching,	Assist with activities and events as directed by management	Volunteer (s) Intern (s)

	counseling, guiding and nurturing the students	Educate, Counsel, Encourage and Promote the development of the students	Mentor (s) / Teacher (s) College Coach VISTA Representative Parents
Primary End-users	People who would use the system as an educational and research tool	Utilize and benefit from resources offered within the program Learn and grow as an individual and a student Achieve goals and objectives pertaining to their education	Current students Parents
New and Potential Users	Individuals who recently joined or who are interested in becoming a student and looking for information	Utilize and benefit from the resources offered within the program Acquire information regarding their interest in joining the program	New and Potential students and their Parents
Donors	People who want to financially support the program	Be able to communicate their intent Be able to provide accurate, trustworthy and reliable information pertaining to donations.	Current and Potential Donors

1.5 User Environment

The primary target end-users are the current registered students. They are often assisted by teachers, mentors, volunteers and interns; and extensively utilize educational resources. The students spend about 3-4 hours after school in this program and currently the main devices supported are ipads and other devices. The amount of time spent on each activity depends on the activity and might be changing depending on the student's progress.

The current system's administrators spend most of their time writing lengthy grant proposals with iweb2009, which has very limited capability; it is time consuming and inconvenient for long-term use. In addition, Adelante's donors' information is overall an entry in an Accounting Software that only captures information about the date and amount of the donation.

For each meeting, students and volunteers' attendance are taken on paper then later manually processed in the current system by the Volunteer and Program Operations and Development Coordinator prior to monitoring participation. In addition, forms and applications are similarly processed. This is a burdensome task as it takes time and effort to process all information three times a week.

When it comes to sharing documents, there is no standard or official practice or method of doing so. An original copy can be passed around with eventually different version of the same information. The same goes for recordkeeping, there is no centralized data storage to keep all files.

An Enterprise Resource Planning system could alleviate the hardship associated with managing day-to-day operations and ultimately add value to the organization. The need of sharing files resulting in information inconsistency would be suppressed.

A Communication System with integrated scheduling and reminder functions would support the need for communicating between employees, support planning fundraising and volunteering activities as well as support the need for a scheduling assistant with grant deadlines and reminders.

There are drawbacks to most systems. There could be downtime for different reasons such as maintenance or technical issues. Maintenance could be scheduled at times when there is minimum traffic. The user environment can be also affected by connectivity or weather issues.

1.6 Summary of Key Stakeholder or User Needs

The current environment is not ideal for the purposes of the organization. Most of the business processing is done manually. There is no central storage for files, data and information are inconsistent across the organization and could benefit from a better communication system. There hasn't been a proposed solution to these problems but it will be eventually offered.

It would consider:

- Sign up Newsletter
- News transmission page
- Mass Communication
- Event Page
- Donor Database
- Direct Donation link
- Interactive Calendar
- Volunteer Sign up page
- Volunteer Database
- Storage and retrieval system for all forms
- Student Database
- Report Generation Tool

Need	Priority	Concerns	Current Solution	Proposed Solutions
Sign up Newsletter	13	Admin(s)	None	Sign up form that will request news information to be sent to an email

News transmission page	12	Admin(s)	None	News transmission page to present a brief overview about the program and inform about upcoming events
Mass Communication	8	Admin(s)	None	Mass communication System for chatting between employees, emailing, media sharing
Donor Database	3	Admin(s)	None	Donor Database to facilitate recordkeeping and tracking of donors' information, related fundraising events, contributions, integrated with a payment processor
Direct Donation link	7	Admin(s)	None	Link that would directly lead to donations information
Interactive Calendar	9	Admin(s)	None	Interactive Calendar to assist with scheduling and reminders
Volunteer Sign up page	11	Admin(s)	None	Track people who express interest in volunteering-sign up form, email feedback to administrator
Event (Fundraising) Page	10	Admin (s)	None	Event Page that displays all information regarding recent and upcoming events, fundraising, etc…
Storage and retrieval system for all forms	2	Admin(s)	None	Storage system for all forms
Student Database	1	Admin(s)	None	Database to able to track students' attendance, grade, progress, volunteers' hours, interests, hours served, survey data, as well as extracting data from files

Report Generation Tool	5	Admin(s)	None	Data received from JCPCS cascade system- be able to generate statistical report
Database for Grants	4	Admin(s)	None	Database for Grants information to facilitate recordkeeping and tracking of amounts, timeline, project, manager, due dates, integrated with a calendar/ reminder of deadlines

1.7 Alternatives and Competition

Adelante could abandon this project and continue operating with its current system. However, they are pursuing the option to upgrade their current environment prior to accommodating organizational needs. Adelante might have some competition inside the community but there aren't serious threats to the organization's future, though, not being able sustain future operations might be an internal threat.

Product Overview

1.8 Product Perspective

The new system will integrate all information within the current system in addition to new features requested. It will be web-based, mobile and kid friendly, professional, and reflective of the organizations' mission and objectives.

Product Features

The following are features that are thought of as necessary for the needs expressed in the organization.

A. Sign up Newsletter

A Sign up form that will request news information to be sent to an email

B. Mass Communication

Mass communication System for chatting between employees, emailing, media sharing

C. Event Page

An Event Page to promote recent and upcoming activities primarily about fundraising, volunteering, student projects, etc.…

D. Donor Database

Donor Database to facilitate recordkeeping and tracking of donors' information, related fundraising events, contributions, integrated with a payment processor

E. Direct Donation link

Link that would directly lead to donations information

F. Interactive Calendar

Interactive Calendar to assist with scheduling and reminders

G. Volunteer Sign up page and Database

Track people who express interest in volunteering- sign up form, email feedback to administrator

H. Storage and retrieval system for all files

Storage system for all forms

I. Student Database

Database to able to track students' attendance, grade, progress, volunteers' hours, interests, hours served, survey data, as well as extracting data from files

J. Report Generation Tool

Data received from JCPCS cascade system- be able to generate statistical report

K. Database for Grants

Database for Grants information to facilitate recordkeeping and tracking of amounts, timeline, project, manager, due dates, integrated with a calendar/ reminder of deadlines

Other Product Requirements

1. Applicable standards

The user-interface must compatible with mobile devices and PCs for any form of an operating system.

2. Constraints

It might worthwhile to note that there might be design and quality constraints. When designing the environment, a lot of factors are going to be taken under accounts and ultimately the primary concern would be to address all needs while considering ease of navigation for all current and future users with different levels of computing skills. Quality constraints will arise during implementation as the decision of what information to display comes into play.

Adelante Hispanic Achievers

CIS 320-01
03/01/2016

College of Business
University of Louisville

Appendix A- Hardware and Software

To properly function, the Cloud-based integrated Solution to be proposed will require a set of hardware and software listed below.

1. Computers
2. Laptops
3. Tablets
4. Printer
5. Web Hosting
6. 1TB Google Drive
7. My SQL Support
8. Internet Connection

*Please refer to Economic Feasibility for actual costs of Softwares listed above. Hardware cost is subject to customer's budget.

Adelante Hispanic Achievers

Appendix B: Feasibility Study

CIS 320-01
03/01/2016

College of Business
University of Louisville

1. Organizational Feasibility

In this section we focus on the business needs answered if we build the new system.

* Familiarity with application
* Familiarity with Technology
* Compatibility: Manual processing vs. New Information System
* Project Size - medium, fairly risky

We considered the technical feasibility to be medium too fairly risky based on a few different factors. We are not integrating any older software making this fairly easy and a lower risk. The back-end interface where administration will be updating specified fields so users who don't have much technical skills will still be able to update the data without manipulating the actual code. Familiar with the technical environment used to produce the site. Word Press has many plug-ins available that essentially provide the user interface and usability. We are all very familiar with the web-based technologies incorporated in the project and understand how to apply our knowledge to the needs of Adelante.

2. Technical Feasibility

Can we build the system from a technical standpoint?

* Strong alignment between project and business strategy
* Business process improvement with daily activities
* Organizational management- belief in valuable contribution from proposed business application
* Ultimate end- Users- contribute to the quality of the system by providing feedback

The organizational feasibility measures the project and business strategy making sure these two components align with each other so every aspect of the project is on the same page and everyone is working toward the

same goal. With the implementation of this new site the daily operations will change drastically. For instance, grant writing time could be reduced by 67-83%, registration time could go under 5-10 mins with automation in place, donor could access donations much quicker and they could be processed a lot faster through PayPal. Managing the system will eventually have to be done by a Webmaster and is also dependent on administration as well but with the implementation of a new website this will bring much value to the organization in monetary measurements and in many non-tangible aspects.

3. Economic Feasibility

Should we build the system from a monetary standpoint?

In this section we evaluate the monetary implications of what is considered, measured, and compared. The results of this investment appraisal help us to ensure that the decision making bodies that govern the business investment undertake the right projects. Below is a cost-benefit analysis we've created to better understanding of the economic feasibility. Should Adelante go with this project, the net present value would be $188,500, they would realize a return on investment of 254%, and they will be able to recover their initial investment around 3 years, 6months, 5 weeks.

Adelante Hispanic Achievers

Cost-Benefit Analysis						
	2016	2017	2018	2019	2020	2021
Benefits						
Increase in Donations		500	500	500	500	500
Increase in Grants		16,800	27,600	45,200	74,000	121,200
Total Benefits		17,300	28,100	45,700	74,500	121,700
PV of all Benefits		**$16,900**	**$26,700**	**$42,600**	**$68,000**	**$108,500**
Development Costs						
Labor	(67,900)	-	-	-	-	-
Web Hosting	(200)	-	-	-	-	-
Total Development Costs	($68,100)	-	-	-	-	-
Operational Costs						
Web Hosting	-	-	-	(200)	-	-
1TB Google Drive	-	(100)	(100)	(100)	(100)	(100)
MYSQL Support		(500)	(500)	(500)	(500)	(500)
Printer (Canon MF8280CW)	(200)					
Printing Consumables		(300)	(300)	(300)	(300)	(300)
Internet		(300)	(300)	(300)	(300)	(300)
Total Operational Costs	(200)	(1,200)	(1,200)	(1,400)	(1,200)	(1,200)
Total Costs	-$68,300	-$1,200	-$1,200	-$1,400	-$1,200	-$1,200
PV of all Costs	**-$68,300**	**-$1,200**	**-$1,100**	**-$1,300**	**-$1,200**	**-$1,100**
Total PV	(68,300)	15,700	25,600	41,300	66,800	107,400
Total Cumulative PV	(68,300)	(52,600)	(27,000)	14,300	81,100	188,500
Return on Investment	254%					
Net Present Value	**$188,500**					
Break-Even Point	3.65 years					

*The above table is based on data provided by Adelante, Bureau of Labor Statistics and Research for softwares listed. Please see next page for actual costs.

Grants Per year Current	9
Projected grants per yr	11
Most Recent grant total	73900
Estimated value per grant	8211
Projected grants amt.	90322
grant increase	16422

Initial Labor	
CIS Project Manager	127640
Database Admin	80280
Web Dev	63490
13 week price	67853

3.1. Improvements Quantified.

- 4-6 hours per week in data entry
- ~150 students registered for M&E Program
- under same time frame or 20-25 minutes' total time for registration
- 10-15 minutes Volunteer Recruitment or 120 Volunteers
- under same time frame could be recruited
- 10-15 minutes A.S.P Registration
- or 40-65 students could be registered under same time frame
- 5-10 hours per grant proposal writing time
- ~25-27 grants
- ~15-19 grants approved
- ~60-70% approval rate
- ~67-83% Time Reduction for writing grants

Adelante Hispanic Achievers

Appendix B: Process Models

Adelante Hispanic Achievers

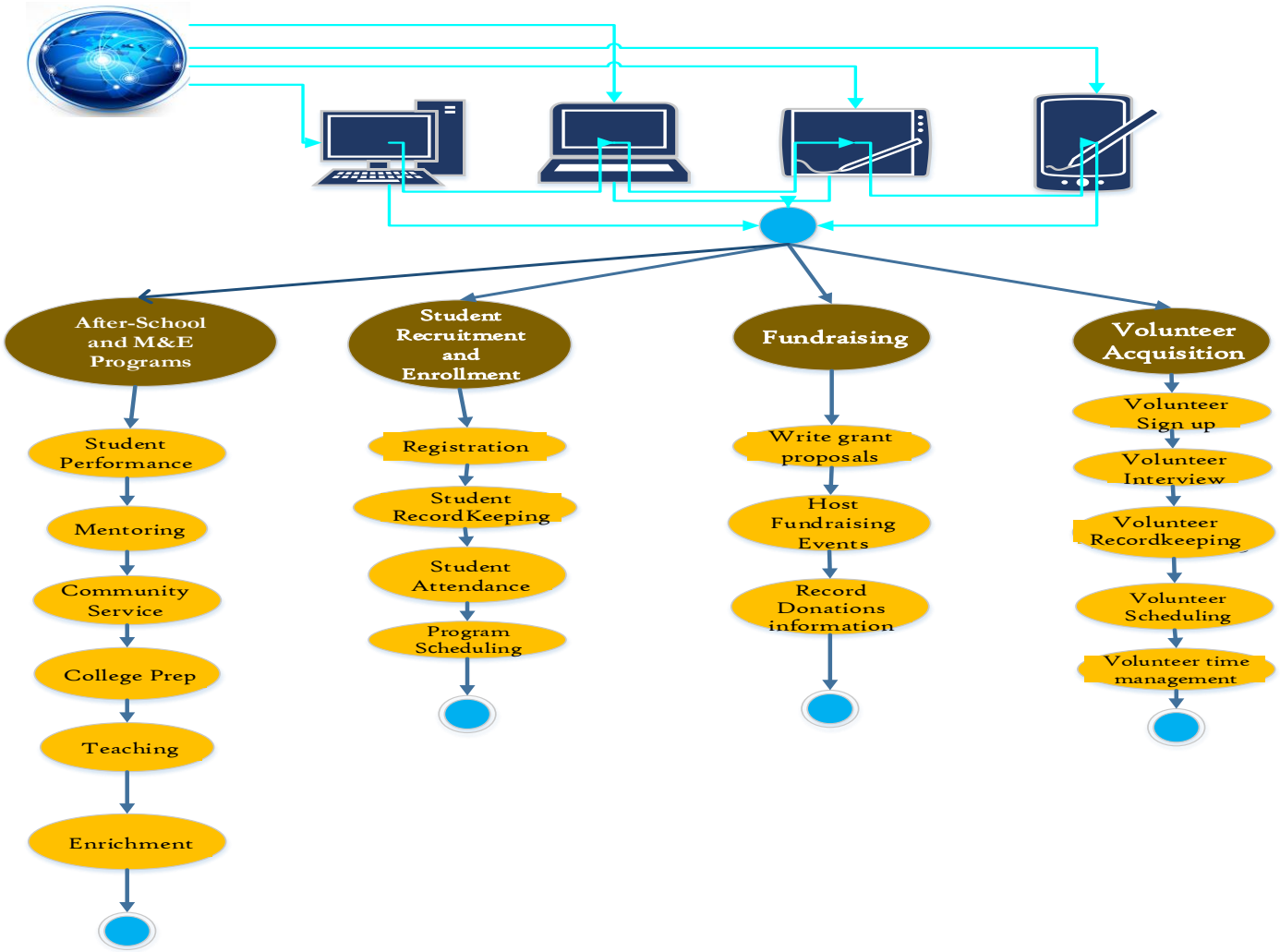
Process Models

CIS 320-01
03/01/2016

College of Business
University of Louisville

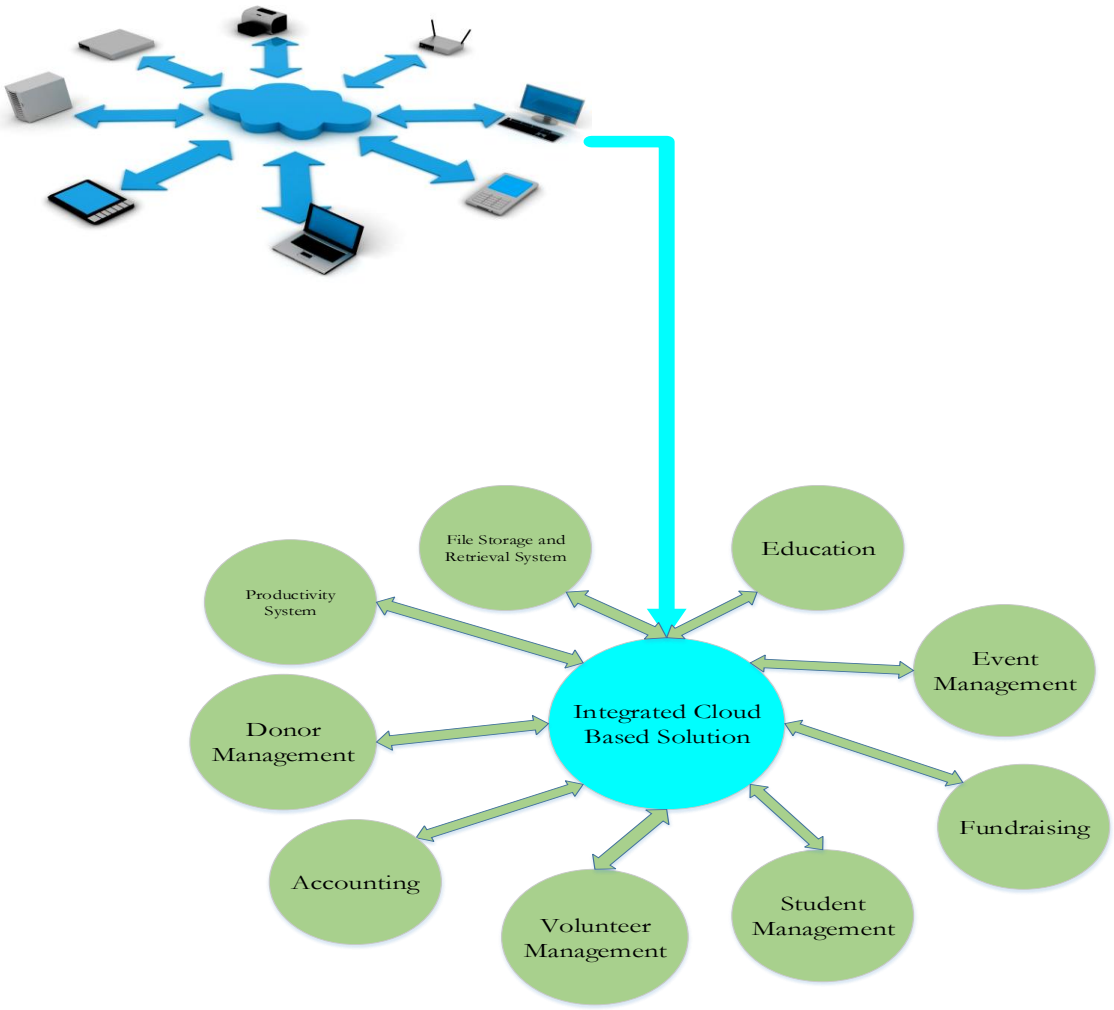
We developed two process models for this project. The As-Is process models depicts to the current business processing environment and gives a visual representation how business is conducted inside Adelante. The To-Be Model projects our vision of the system to be proposed. It gives an overview of all elements that responds to the current business needs.

As-Is Model


After-School and M&E Programs
Student Performance
Mentoring
Community Service
College Prep
Teaching
Enrichment
Student Recruitment and Enrollment
Registration
Student RecordKeeping
Student Attendance
Program Scheduling
Fundraising
Write grant proposals
Host Fundraising Events
Record Donations information
Volunteer Acquisition
Volunteer Sign up
Volunteer Interview
Volunteer Recordkeeping
Volunteer Scheduling
Volunteer time management

Adelante Hispanic Achievers

To-Be Model



CIS 320-01

03/01/2016

College of Business

University of Louisville

Adelante Hispanic Achievers

Appendix C: Risk Analysis

High Risk Use Cases Risk Analysis

CIS 320-01

03/01/2016

College of Business

University of Louisville

Risk Analysis

1. Criteria for assessing use case risk

   Low -- No action required; Low severity of failure
   High -- User input required; Donation transactions; High severity of failure

2. Use case Risk Level Assessment

   Please refer to the following page.

   KEY:
   Yellow = HIGH RISK

3. Other Risk Areas

   Functionality, Security, Usage Frequency, Usability and Payment System for Donations.

   a. Criteria for assessing use case risk

   Subjective assessment based on high/low: severity of failure, likelihood of failure, and complexity of risk area.

	Low	High
Functionality		X
Security		X
Usage Frequency	X	
User Friendliness	X	

Payment System	X

During the iterations of the Elaboration phase, risk may be addressed in several ways. During this phase we will have a better, more detailed understanding of the requirements and the baseline design for the architecture. It is during this period that technical risks can be addressed. During the design, implementation, and testing of the architecture, all essential risks are to be mitigated.

Adelante Hispanic Achievers

1. CreateUserAccount
2. ReadUserAccount
3. UpdateUserAccount
4. DeleteUserAccount

5. CreateEvent
6. ReadEvent
7. UpdateEvent
8. DeleteEvent
9. PostEvent
10. EventPage

11. CreateCalendarDate
12. ReadCalendarDate
13. UpdateCalendarDate
14. DeleteCalendarDate
15. InteractiveCalendarPage

16. CreateNewsletter
17. ReadNewsletter
18. UpdateNewsletter
19. DeleteNewsletter
20. MassEmailPage

21. CreateFile
22. ReadFile
23. UpdateFile
24. DeleteFile
25. UploadFile
26. DownloadFile

27. CreateDonateLink
28. ReadDonateLink
29. UpdateDonateLink
30. DeleteDonateLink
31. DonationPage

32. CreateShareFile
33. ReadShareFile
34. UpdateShareFile
35. DeleteShareFil

36. CreateReport
37. ReadReport
38. UpdateReport
39. DeleteReport

40. CreateSpendingTracker
41. ReadSpendingTracker
42. UpdateSpendingTracker
43. DeleteSpendingTracker

44. CreateDonationTracker
45. ReadDonationTracker
46. UpdateDonationTracker
47. DeleteDonationTracker

48. CreateGrantTracker
49. ReadGrantTracker
50. UpdateGrantTracker
51. DeleteGrantTracker

52. CreateVolunteer
53. ReadVolunteer
54. UpdateVolunteer
55. DeleteVolunteer
56. VolunteerPage

57. CreateStudent
58. ReadStudent
59. UpdateStudent
60. DeleteStudent

61. CreateDonor
62. ReadDonor
63. UpdateDonor
64. DeleteDonor

65. AdminLogin
66. PayPalRedirect
67. CreateArchive
68. ReadArchive
69. UpdateArchive

70. DeleteArchives

71. AssignPermission
72. CreateBackup
73. UpdateBackup
74. PerformSystemRecovery

Appendix D:High Risk Use Cases

High Risk Use Case

CIS 320-01

College of Business

03/01/2016

University of Louisville

CIS 320-01

College of Business

03/01/2016

University of Louisville

The list of use cases below give an overview of the high risk use cases developed in the previous section. Each use case was designated an identifier, a primary actor, though some use cases has multiple actors assigned. Actors are the people that are associated with the functionality that use case ultimately fulfill. Also, each use case was assigned a brief description that help clarify the purpose of the use case and eventually which requirement
it fulfills.

CIS 320-01

03/01/2016

College of Business

University of Louisville

Adelante Hispanic Achievers

Use Case: CreateUserAccount
ID: 1
Brief Description: Allows any student, upon acceptance into the program to create an account that will be further used by admin to create and keep records of students
Primary Actor: Student, Parent

Use Case: UpdateUserAccount
ID: 3
Brief Description: Allows any current student to login into their accounts and update their information such as school, address, etc.…
Primary Actor: Student, Parent

Use Case: CreateEvent
ID: 5
Brief Description: Allows Admin to create an upcoming such as fundraising, volunteering, school visits
Primary Actor: Admin

Adelante Hispanic Achievers

Use Case: UpdateEvent
ID: 7
Brief Description: Allows Admin to update an upcoming event such as fundraising, volunteering, school visits with more or different information
Primary Actor: Admin

Use Case: Event Page
ID: 11
Brief Description: Display all upcoming events and activities in one page.
Primary Actor: Parents, Students, Volunteers

Use Case: CreateCalendarDate
ID: 11
Brief Description: Allows Admin to create a reminder for event or deadlines within the interactive calendar
Primary Actor: Admin

Use Case: UpdateCalendarDate
ID: 13

Brief Description: Allows Admin to update calendar reminders for deadlines or events within the interactive calendar
Primary Actor: Admin

Use Case: CreateNewsletter
ID: 16
Brief Description: Allows members to sign up for newsletters to be sent to their email account
Primary Actor: Parents, Volunteers, Students

Use Case: UpdateNewsletter
ID: 18
Brief Description: Allows members to update information originally submitted in the sign up newsletter form
Primary Actor: Parents, Volunteers, Students

Use Case: MassEmailPage
ID: 20

Brief Description: Allows Adelante employees and board members to be able to send a blast email to all or multiple employees of their choice.
Primary Actor: Staff or Board Members

Use Case: CreateFile
ID: 21
Brief Description: Allows Admin to be able to create a file such as a video, picture, text, grant proposal templates, etc.…
Primary Actor: Admin, Staff Members

Use Case: UpdateFile
ID: 23
Brief Description: Allows Admin or staff member to be able to edit previously created files within the system and make changes accordingly.
Primary Actor: Admin, Staff member

Use Case: CreateDonteLink

ID: 27
Brief Description: Allows members to able to create a link for donation that redirect to the donation page and eventually to PayPal for the collection of funds.
Primary Actor: Admin

Use Case: UpdateDonateLink
ID: 29
Brief Description: Allows members to able to edit and update a donation link with the appropriate information.
Primary Actor: Admin

Use Case: DonationPage
ID: 31
Brief Description: Donation Page with all necessary information regarding donations and links for donors to get more insight about donations.
Primary Actor: Donor

Use Case: CreateShareFile

Adelante Hispanic Achievers

ID: 32
Brief Description: Allows Staff and Admin members to be able to share files amongst them
Primary Actor: Admin and Staff Members

Use Case: UpdateShareFile
ID: 34
Brief Description: Allows Staff and Admin members to be able to update shared files with new information or make changes accordingly
Primary Actor: Admin and Staff Members

Use Case: CreateReport
ID: 36

Adelante Hispanic Achievers

Brief Description: Allows Admin to be able to generate a report whether about a student, monthly spending, volunteering hours, donations by events, etc…
Primary Actor: Admin

Use Case: UpdateReport
ID: 38
Brief Description: Allows Admin to edit and update parameters of a report about a student, monthly spending, volunteering hours, donations by events, etc…
Primary Actor: Admin

Use Case: CreateSpendingTracker
ID: 40
Brief Description: Allows Admin to be able to track spending for continuously by amount, date, type and period.
Primary Actor: Admin

Use Case: UpdateSpendingTracker
ID: 42

Adelante Hispanic Achievers

Brief Description: Allows Admin to be update spending tracker parameters such as period, date, type.
Primary Actor: Admin

Use Case: CreateDonationTracker
ID: 44
Brief Description: Allows Admin to be able to track donations continuously, by amount, date, period, entity.
Primary Actor:

Use Case: UpdateDonationTracker
ID: 46
Brief Description: Allows Admin to be able to update amount, date, period or entities for each donation as well as parameters.
Primary Actor: Admin

Use Case: CreateGrantTracker
ID: CreateGranterTracker
Brief Description: Allows Admin to be able to continuously track grants received by date, amount, source, period, etc…

Primary Actor: Admin

Use Case: UpdateGrantTracker
ID: 50
Brief Description: Allows Admin to be able to update parameters and information in grant tracker.
Primary Actor: Admin.

Use Case: CreaterVolunteer
ID: 52
Brief Description: Allows Admin to be able to create a record and store information about each volunteer
Primary Actor: Admin

Use Case: UpdateVolunteer
ID: 54
Brief Description: Allows Admin to be able to update information about each volunteer

Adelante Hispanic Achievers

Primary Actor: Admin

Use Case: VolunteerPage
ID: 56
Brief Description: Allows Admin to post volunteering activities available for current and prospective volunteers and students volunteering projects
Primary Actor: Admin, Volunteer, Student

Use Case: CreateStudent
ID: 57
Brief Description: Allows Admin to be able to create a student record
Primary Actor: Admin

Use Case: UpdateStudent
ID: 59
Brief Description: Allows Admin to be able to update student record with necessary information
Primary Actor: Admin

Adelante Hispanic Achievers

Use Case: CreateDonor
ID: 61
Brief Description: Allows Admin to be able to create a record for each donor as part of tracking donations
Primary Actor: Admin

Use Case: UpdateDonor
ID: 63
Brief Description: Allow Admin to be able to update information about a donor
Primary Actor: Admin

Use Case: AdminLogin
ID: 65
Brief Description: Gives Admin unique access and permission within the system
Primary Actor: Admin

CIS 320-01

College of Business

03/01/2016

University of Louisville

Adelante Hispanic Achievers

Use Case: CreateArchive
ID: 67
Brief Description: Allow Admin to create a system of records within the system
Primary Actor: Admin

Use Case: UpdateArchive
ID: 69
Brief Description: Allow Admin to be able to edit and update information in the archives within the system.
Primary Actor: Admin

Use Case: AssignPermission
ID: 71
Brief Description: Allows Admin to assign appropriate access permission level to different users and eventually making users have access to specific information instead of all available information.
Primary Actor: Admin

Adelante Hispanic Achievers

Use Case: CreateBackUp
ID: 72
Brief Description: Allow Admin to be able to back up the system at a point in time whenever necessary.
Primary Actor: Admin

Use Case: UpdateBackUp
ID: 73
Brief Description: Allow Admin to be able to perform full, differential or partial backups whenever necessary.
Primary Actor: Admin

Use Case: PerformSystemRecovery
ID: 74
Brief Description: Allow Admin to be able to recover lost data, restore the system to its initial state or reconfiguring it from scratch should system failure occur at some point in time.
Primary Actor: Admin

Adelante Hispanic Achievers

Appendix E: Risk Trace Matrix

Use Case Risk Trace Matrix

CIS 320-01

College of Business

03/01/2016

University of Louisville

The risk trace matrix below primarily shows how each use case fulfils a requirement for the system To-Be. Should there is a requirement omitted, we understand that we would have to make sure that use case is assigned to that requirement, whether or not it is a current use case or a new one that has to be developed. Essentially, all use cases were assigned to the listed requirement based on functionally whether or not is direct or indirect. As some system requirements are dependent, some use cases become dependent, thereof, assigned to not only to the respective requirements that they are expected to fulfill but also to dependent requirements.

Adelante Hispanic Achievers

	Grant Database	Donor Database	Student Database	Volunteer Database	Storage and Retrieval System	Report Generation System	Direct Donation link	Interactive Calendar	Event Page
1. CreateUserAccount		X	X	X					X
2. ReadUserAccount		X	X	X		X			
3. UpdateUserAccount		X	X	X		X			
4. DeleteUserAccount		X	X	X					
5. CreateEvent									X
6. ReadEvent								X	X
7. UpdateEvent								X	X
8. DeleteEvent								X	X
9. PostEvent								X	X
10. EventPage									X
11. CreateCalendarDate								X	
12. ReadCalendarDate								X	
13. UpdateCalendarDate								X	
14. DeleteCalendarDate								X	
15. InteractiveCalendarPage								X	
16. CreateNewsletter									
17. ReadNewsletter									X
18. UpdateNewsletter									X
19. DeleteNewsletter									X
20. MassEmailPage									

Adelante Hispanic Achievers

	Grant Database	Donor Database	Student Database	Volunteer Database	Storage and Retrieval System	Report Generation System	Direct Donation link	Interactive Calendar	Event Page
21. CreateFile	X	X	X	X		X			
22. ReadFile	X	X	X	X		X			
23. UpdateFile	X	X	X	X		X			
24. DeleteFile	X	X	X	X		X			
25. UploadFile					X				
26. DownloadFile	X	X	X	X	X	X			
27. CreateDonateLink		X					X		
28. ReadDonateLink		X					X		
29. UpdateDonateLink		X							
30. DeleteDonateLink		X					X		
31. DonationPage		X					X		
32. CreateShareFile					X	X			
33. ReadShareFile					X	X			
34. UpdateShareFile					X	X			
35. DeleteShareFile					X	X			
36. CreateReport						X			
37. ReadReport						X			
38. UpdateReport						X			
39. DeleteReport						X			
40. CreateSpendingTracker	X	X			X	X			

Adelante Hispanic Achievers

	Grant Database	Donor Database	Student Database	Volunteer Database	Storage and Retrieval System	Report Generation System	Direct Donation link	Interactive Calendar	Event Page
41. ReadSpendingTracker	X	X			X	X			
42. UpdateSpendingTracker	X	X			X	X			
43. DeleteSpendingTracker	X	X			X	X			
44. CreateDonationTracker		X			X	X	X		
45. ReadDonationTracker		X			X	X	X		
46. UpdateDonationTracker		X			X	X	X		
47. DeleteDonationTracker		X			X	X	X		
48. CreateGrantTracker	X								
49. ReadGrantTracker	X								
50. UpdateGrantTracker	X								
51. DeleteGrantTracker	X								
52. CreateVolunteer				X					
53. ReadVolunteer				X					
54. UpdateVolunteer				X					
55. DeleteVolunteer				X					
56. VolunteerPage				X					
57. CreateStudent			X		X				
58. ReadStudent			X		X				
59. UpdateStudent			X		X				
60. DeleteStudent			X		X				

Adelante Hispanic Achievers

	Grant Database	Donor Database	Student Database	Volunteer Database	Storage and Retrieval System	Report Generation System	Direct Donation link	Interactive Calendar	Event Page	Volunteer Sign up	Newsletter Sign up	Mass Communication
61. CreateDonor		X				X	X					
63. ReadDonor		X										
64. UpdateDonor		X										
65. DeleteDonor		X										
66. AdminLogin	X	X	X	X	X	X	X	X	X	X	X	X
67. PayPalRedirect							X					
68. CreateArchive	X	X	X	X	X	X						
69. ReadArchive	X	X	X	X	X	X						
70. UpdateArchive	X	X	X	X	X	X						
71. DeleteArchives	X	X	X	X	X	X						
72. AssignPermission	X	X	X	X	X	X	X	X	X	X	X	X
73. CreateBackup	X	X	X	X	X	X	X	X	X	X	X	X
74. UpdateBackup	X	X	X	X	X	X	X	X	X	X	X	X
75. PerformSystemRecovery	X	X	X	X	X	X	X	X	X	X	X	X

Adelante Hispanic Achievers

Appendix F: Inception Phase Prototypes

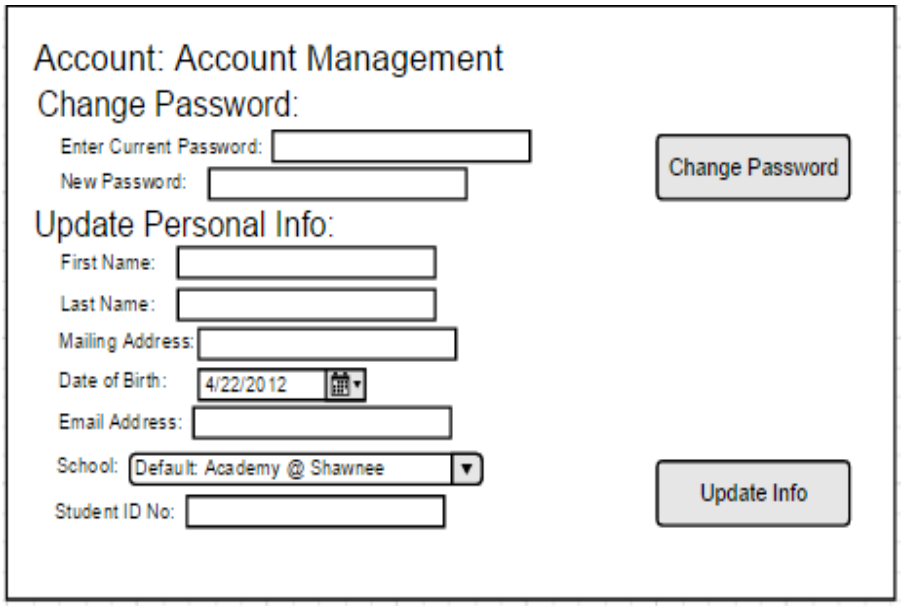
Adelante Hispanic Achievers

Inception Phase Prototypes

Account management

This prototype includes a few functions. The primary function that will likely be used is the password change, which uses a standard “enter current password” box, as well as a new password. It is standard website functionality, and allows a user to update their password.

The other component allows them to update personal information in the event that during account creation, mistakes were made, or in the event they have to change their name, such as their parents getting married or divorced.



Account activity

This prototype includes an interactive calendar that allows a user to see dates that they have attended, as well as a list of dates and events in a table. This page has no ability to edit data, with

the exception of future dates that they may select, and rsvp for. This is like many of the other calendar functions that have secondary functionality than the page itself. It provides a way that allows you to track your activity and progress in Adelante's program.

Adelante Hispanic Achievers

Account: Activity

Recent Notifications:

April 22, 2012

Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

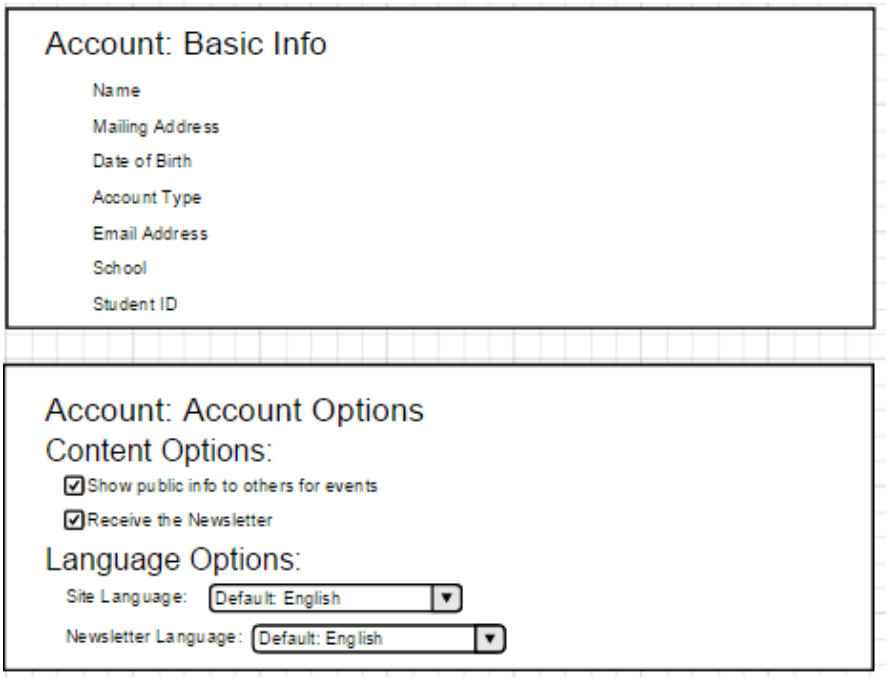
Recent Event Activity:

▼ Date	▼ Event
Cell 1	Cell 2
Cell 4	Cell 5
Cell 7	Cell 8
Cell 10	Cell 11

Adelante Hispanic Achievers

General account options and basic info

These are two separate, small elements. The first element simply displays the users' info to them. It is a fairly simple text menu. The second element allows a user to show their event status, as well as receive newsletter. The language options allow them to select the languages they wish to use for various tasks and publications.

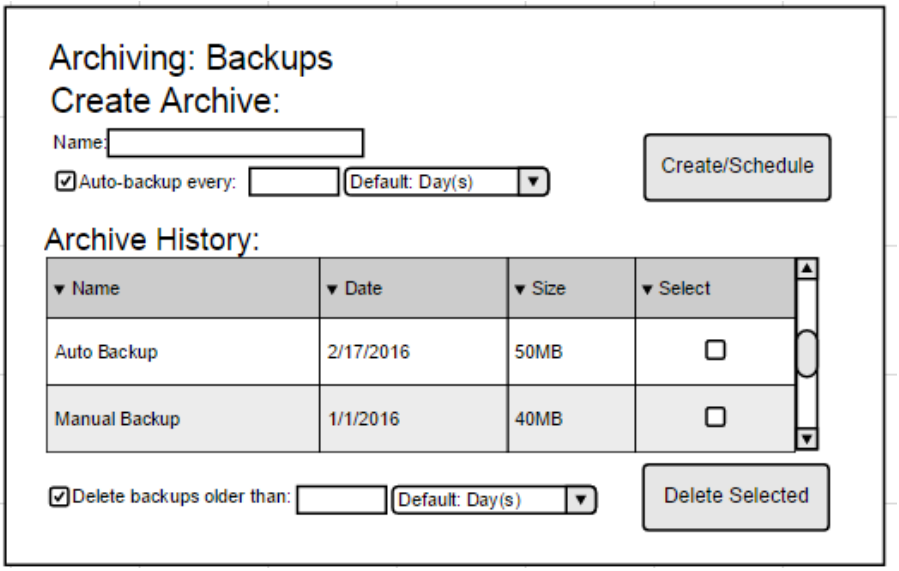


Backups

This is an employee specific function, allowing you to set up backups manually, as well as back up automatically, and manage backup history and deletion. Future functionality could include the ability to roll back or roll forward the entire system or components between backup states in the

event something goes wrong. Our plan of *HostGator* allows us to have two *WordPress* sites, allowing us to perhaps load these to the second site as a testing environment as well.

The basic table functionality allows you to manage a historical archive of backups.



Donation

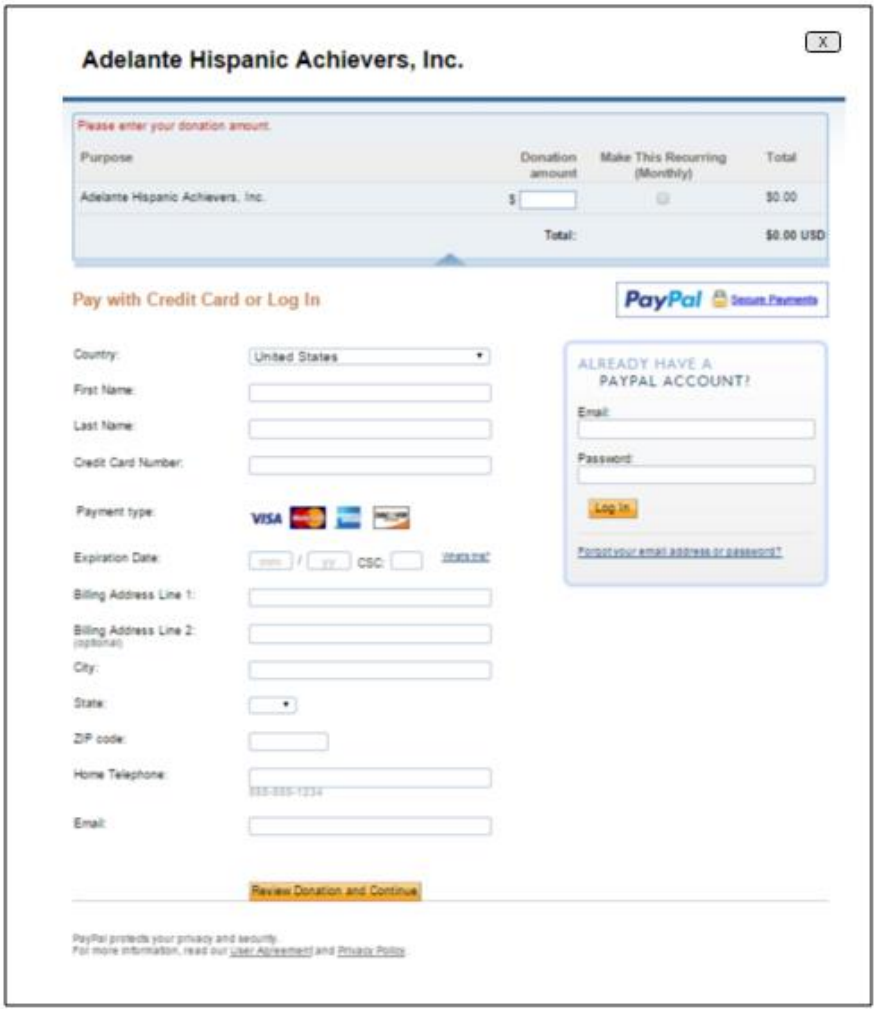
Like other overlay features, these are designed to allow you to perform certain external functionality within the main web page. It is a core functionality of our prototype and possible solution, which is keeping users on the possible website with minimal external interruptions.

Adelante Hispanic Achievers

This page is functionally an embed of their current payment processor for donations, which has reasonable fees, is familiar to Adelante already, and also would require minimal time investment to build.

The current PayPal page is standard, and contains all the information needed to complete a donation transaction.

Adelante Hispanic Achievers

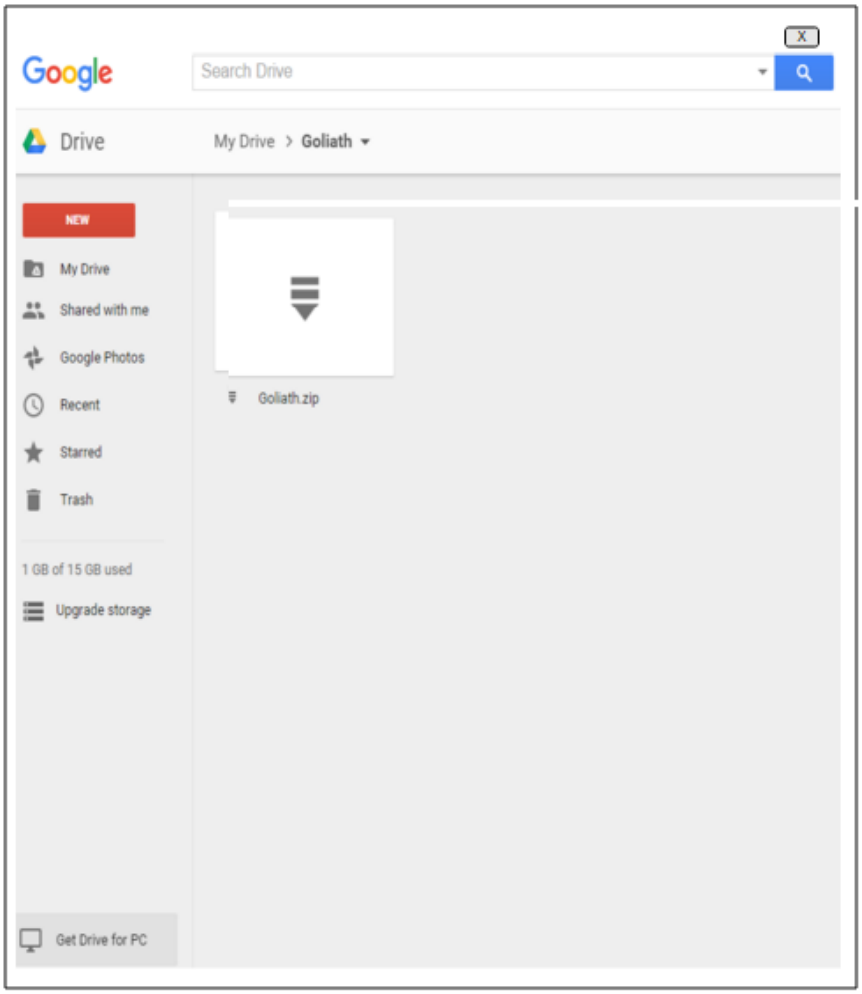


File Storage and Retrieval

Like other overlay elements, this allows you to view your drive files within a webpage.

There really is not much to say about this. Because it uses the google drive page itself, you have access to all of the drive functionality, such as setting up file folders, moving and downloading files, as well as creating and editing documents.

Adelante Hispanic Achievers

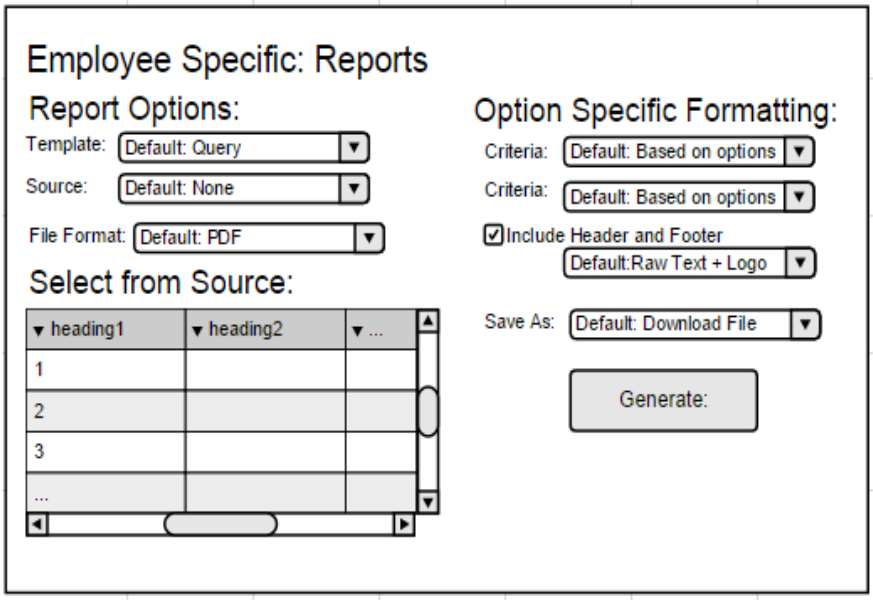


Adelante Hispanic Achievers

Report Generation Tool

This functionality allows you to select data sources, headings and other data from a possible database backend, and format with custom or predefined templates, and output to a file type of choice.

There are a number of options, such as query templates, and default output is a formatted pdf file. Headers and footers can be added, and you could have access to a number of options for saving, such as downloading, or uploading to a specified cloud storage service.

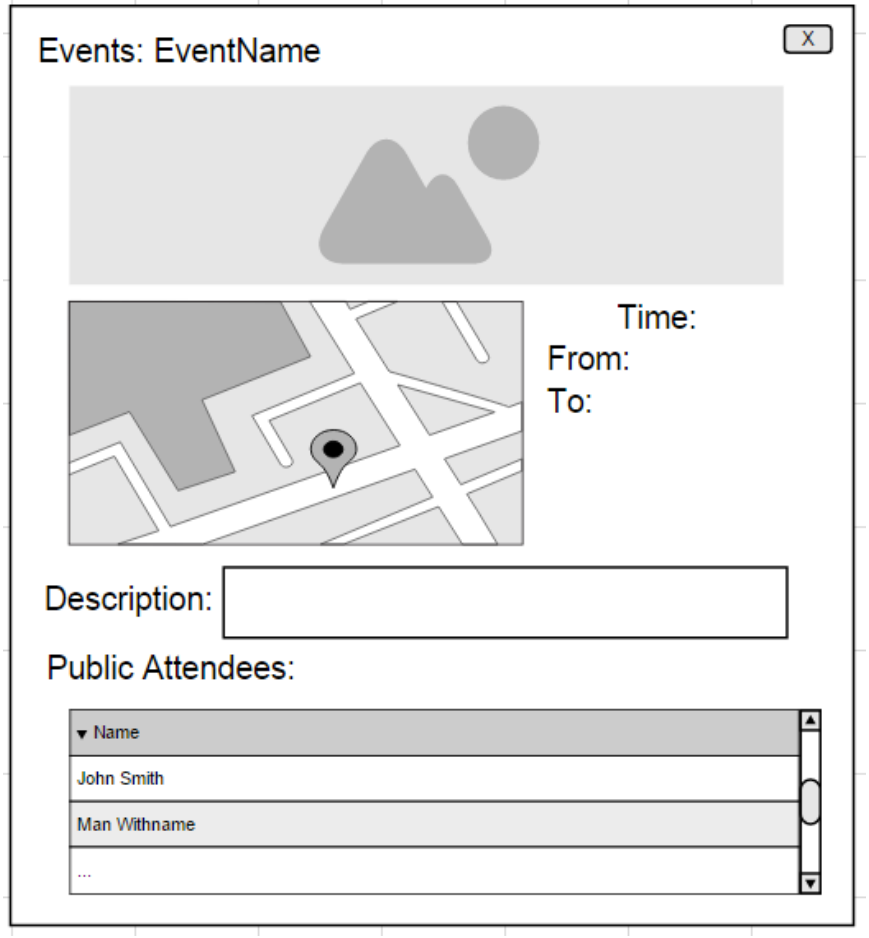

Employee Specific: Reports
Report Options:
Template:
Default: Query
Source:
Default: None
File Format:
Default: PDF
Select from Source:
heading1
heading2
...
1
2
3
...
Option Specific Formatting:
Criteria:
Default: Based on options
Criteria:
Default: Based on options
Include Header and Footer
Default:Raw Text + Logo
Save As:
Default: Download File
Generate:

Adelante Hispanic Achievers

Events

The event overlay provides a general overview of an event for those who may be interested in going.

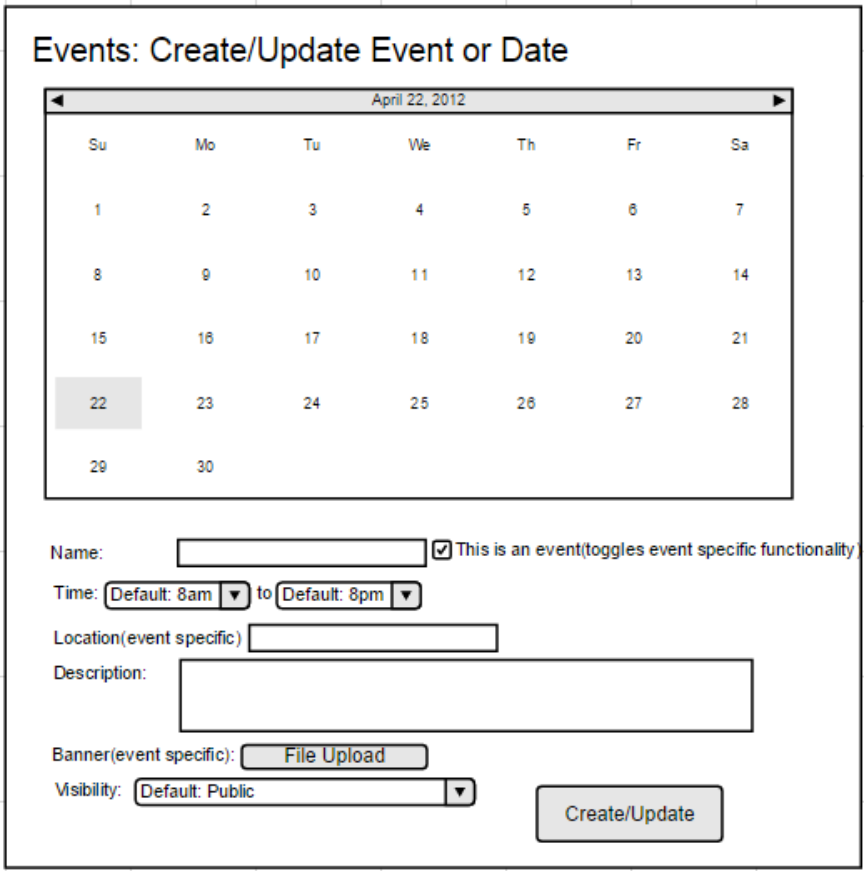
It is a local overlay, and we would build the functionality. It takes a banner image that would give an idea of what the event is about, lists time and dates of an event, as well as uses an embedded maps function to illustrate the location of the event. The description is configurable by the organization, and would describe an event, including things such as dress code and other special consideration. In addition, public attendees display the list of all members who have selected the show public info to others for events, and also RSVP'd for an event.


Events: EventName
X
Time:
From:
To:
Description:
Public Attendees:
Name
John Smith
Man Withname
...

Adelante Hispanic Achievers

Event management

This is the employee version of the events page. The core feature of this element is the calendar. Much like the events activity calendar, it allows you to select events, but that is where the similarity ends. Employees can select an event, and edit it, or select a date, and fill out the required information to create dates and events for the organization. Key features include a description, optional event banner, location address, as well as account visibility.

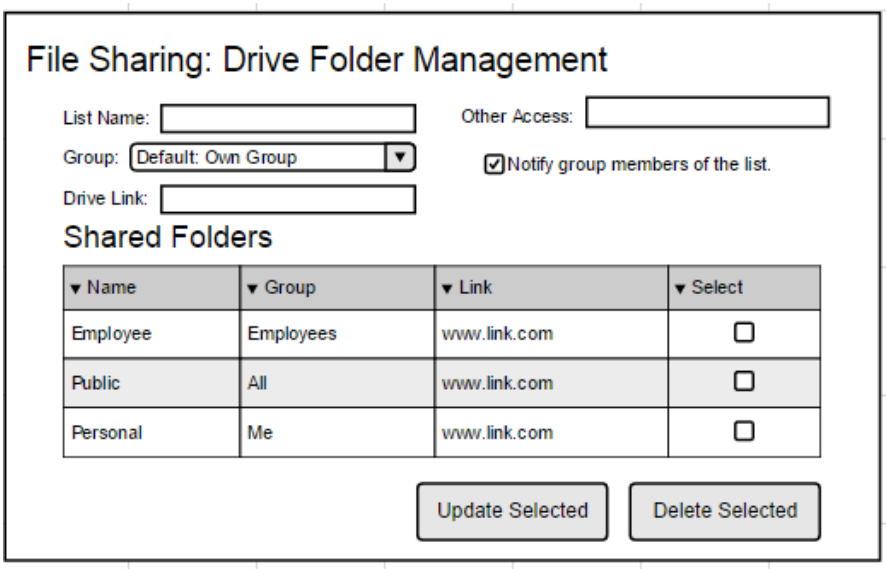

Events: Create/Update Event or Date
April 22, 2012
Su Mo Tu We Th Fr Sa
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30
Name:
This is an event(toggles event specific functionality)
Time: Default: 8am to Default: 8pm
Location(event specific)
Description:
Banner(event specific): File Upload
Visibility: Default: Public
Create/Update

Adelante Hispanic Achievers

File Sharing

This is a core feature for productivity. Adelante employees could add list entries, select availability, and create a google drive folder that allows certain users to access, collaborate, and view files, without compromising other file folders.

Creation features allow you to name a list, edit a list, as well as change the drive link, created in drive on the admin account.


File Sharing: Drive Folder Management
List Name:
Other Access:
Group:
Default: Own Group
Notify group members of the list.
Drive Link:
Shared Folders
Name
Group
Link
Select
Employee
Employees
www.link.com
Public
All
www.link.com
Personal
Me
www.link.com
Update Selected
Delete Selected

Adelante Hispanic Achievers

Email

Everyone can access this. It is a simple mailing box, and allows users with varying permissions to send to varying people. Adelante would likely want to configure the mailing restrictions in order to have full control over internal communication.

The element includes basic recipient, attachment, and subject functionality. However, if Adelante wants to disable the list option, that is fine as well.

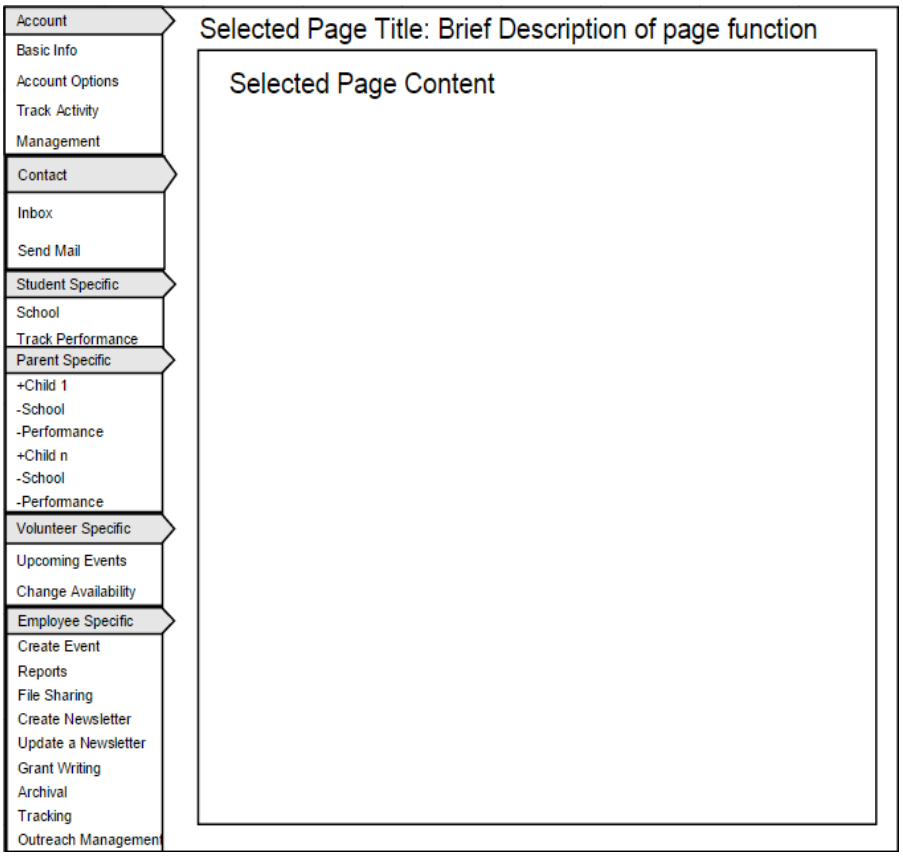
Adelante Hispanic Achievers

Adelante Hispanic Achievers

Master Page

This is the core page of the backend. It has elements contained in a side bar, as well as identifying header and subtitle sections for page content loaded based on your sidebar selection. Many of the other elements are actually members of the content tree, however not all are complete as of yet. The general idea is that the elements are loaded, instead of the entire page refreshing. I wanted to have the website be fast, instead of having to refresh.

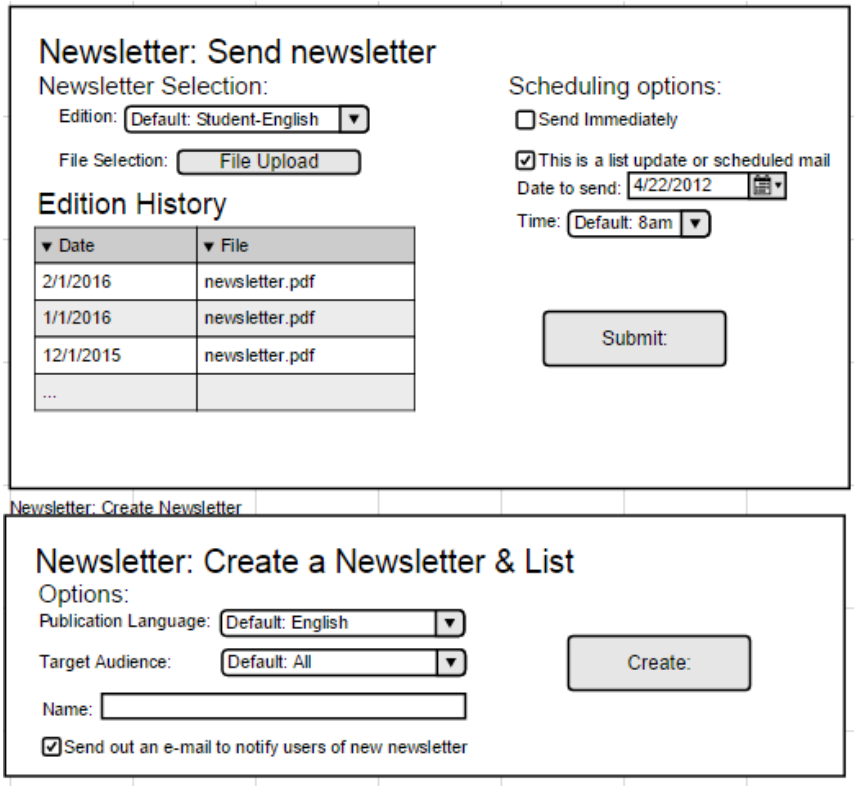
Adelante Hispanic Achievers



Adelante Hispanic Achievers

Newsletter Page

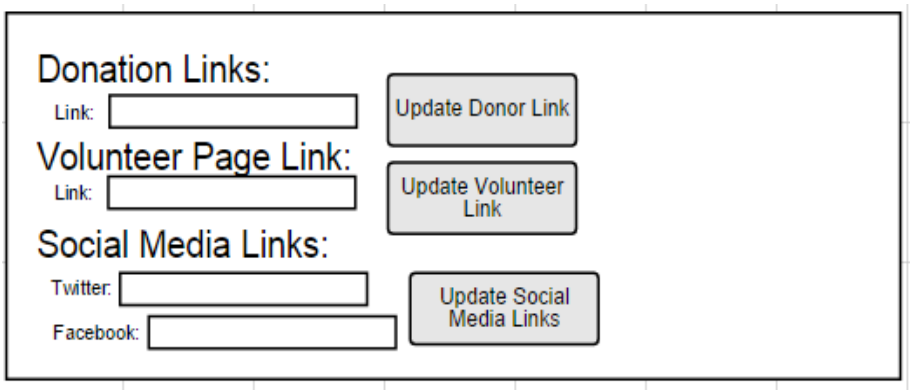
The send newsletter page relies on a list having been created in the newsletter list management. The list management contains elements that allow Adelante to name, as well as select target audience and publication language. The send newsletter page relies on Adelante uploading a newsletter, and allows them to send immediately and schedule sending.



Outreach management

This element is incredibly simple. It allows Adelante to create and update links to overlay page elements, as well as manage the donation button, volunteer button, and social media functions on

the main pages. They simply must enter a link, and update. To delete one, they simply blank it out and update.



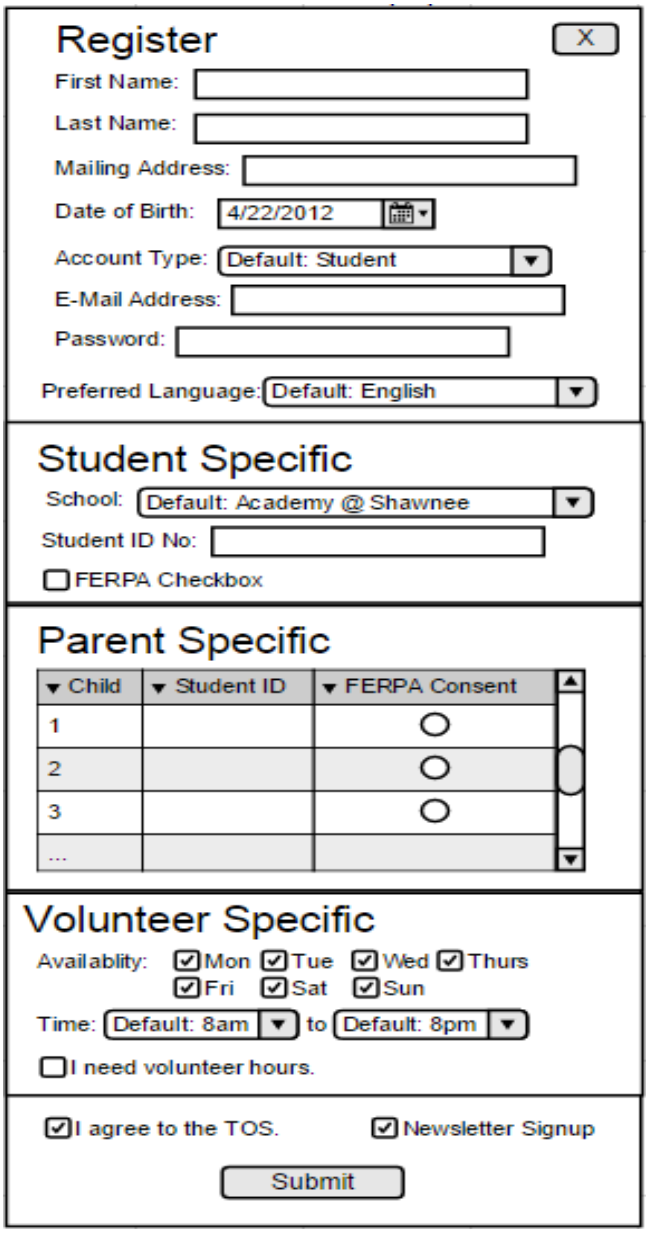
Adelante Hispanic Achievers

Registration

This is an overlay page, like other overlay pages, it is designed to not interfere with browsing, unlike a redirect. This element contains universal elements at the top and bottom, containing basic information common to all account types, such as name, address, date of birth, account type, and contact info as well as default language.

Students get additional elements that allows them to select school and enter student id number. Parents get a table that lets them create a parent-child relationship, enabling them to track their student's progress and contact their children. Volunteers get additional elements geared toward availability, and an option that notifies Adelante that once they volunteer, they will need volunteer hour forms filled out. The page also includes options for everyone to sign up to the newsletter most appropriate for their account type and language.

Adelante Hispanic Achievers

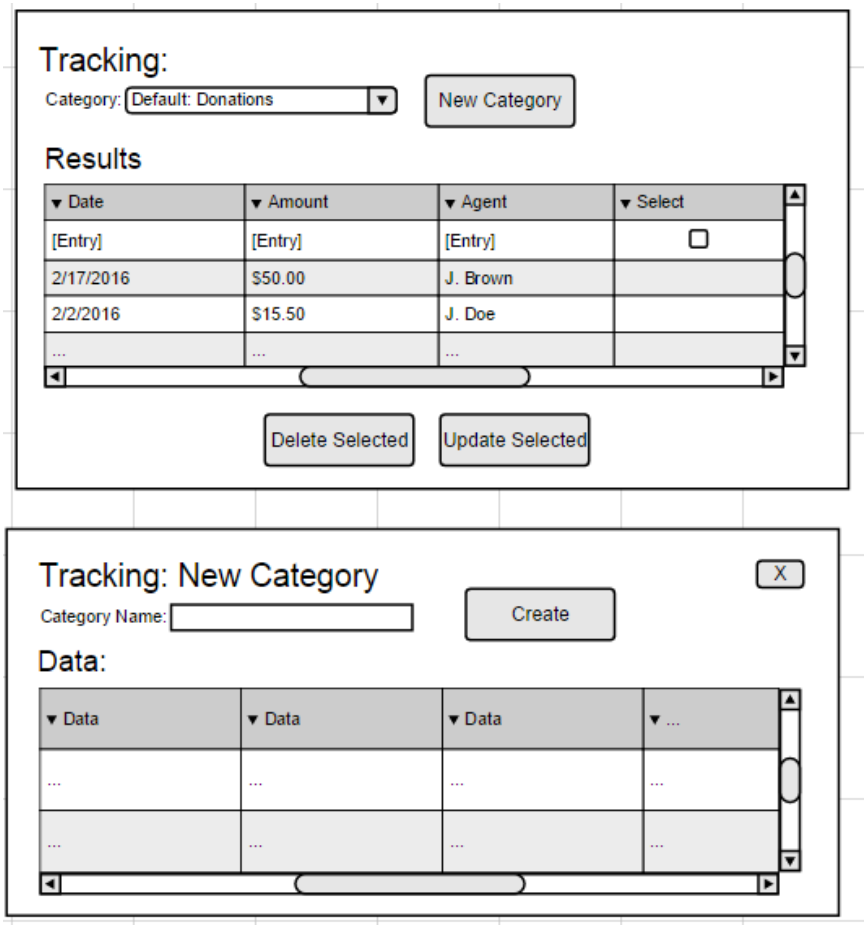


Adelante Hispanic Achievers

Tracker Tool

The tracking element relies on categories being created. When a user clicks on a new category, Adelante can select a data source, and applicable data they wish to track. The create category element is handled like an overlay, and once complete, allows a user to return to the tracking page immediately, and start tracking, sorting, and filtering data. Anything from student info, to financials can be tracked, and access is limited by their account permissions.

Adelante Hispanic Achievers



Appendix G: Gantt Chart

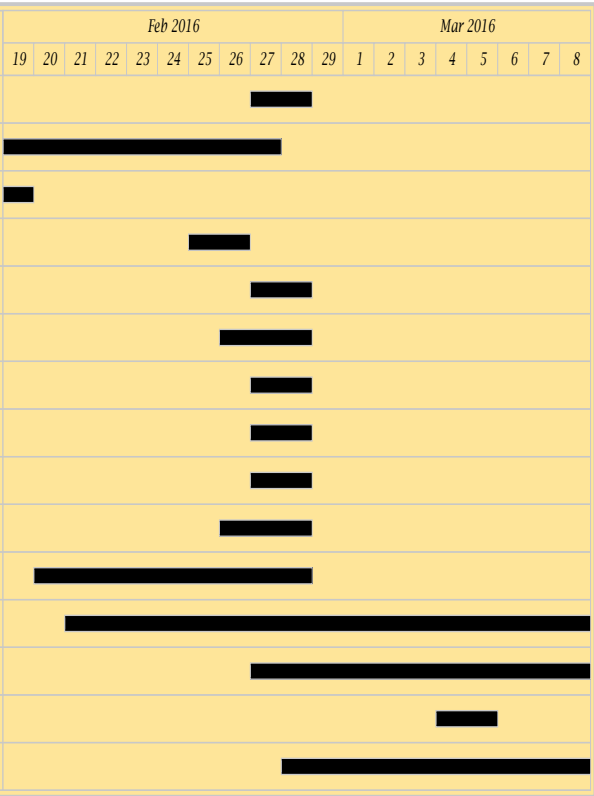
Adelante Hispanic Achievers

Gantt Chart

The Chart below is a graphical representation of all tasks pertaining to Inception Specification and the upcoming Iteration 3. The chart shows all tasks already completed and upcoming, as well as the actual duration it took to complete them and the people that were assigned. It also shows which tasks were dependent and which were independent. In Iteration 3, we will have almost everyone work on the same tasks because we will be concentrating on developing rational use cases and use case diagrams.

Gantt Chart

ID	Task Name	Start	Finish	Duration	Resource Names
1	System Request	2/27/2016	2/28/2016	2d	Chad
2	Vision Document	2/19/2016	2/27/2016	9d	Awa
3	Feasibility Considerations	2/19/2016	2/19/2016	1d	Awa
4	Process Models	2/25/2016	2/26/2016	2d	Awa
5	System Requirements	2/27/2016	2/28/2016	2d	Valencia
6	List of Use Cases	2/26/2016	2/28/2016	3d	Hermilo
7	Initial Architectural Considerations	2/27/2016	2/28/2016	2d	William
8	Use Case Risk Analysis- High Risk	2/27/2016	2/28/2016	2d	Valencia
9	Team Charter	2/27/2016	2/28/2016	2d	Valencia
10	Gantt Chart- Inception Spec	2/26/2016	2/28/2016	3d	Awa
11	Inception Phase Prototypes	2/20/2016	2/28/2016	9d	James/ Hermilo
12	Use Case Detail Construction	2/21/2016	3/8/2016	17d	Awa, Valencia, James, Hermilo, William, Chad
13	Use Case Diagram	2/27/2016	3/8/2016	11d	Awa, Valencia, James, Hermilo, William, Chad
14	Gant Chart- Iteration 3	3/4/2016	3/5/2016	2d	Awa
15	Use Case Prototypes- Version 1	2/28/2016	3/8/2016	10d	James/ Hermilo



II. Elaboration Specification

1. 1. System Requirements

a. *-Functional Requirements*

b. *-Non-Functional Requirements*

c. *-Requirement Details*

2. Appendix A: Use Cases, Use Case Diagram and Sequence Diagram

3. **Appendix B: Trace Matrix**
    a. *-Condensed within Superclasses*
4. **Appendix C: Class Diagram**
5. **Appendix D: Database Design and Data definitions**
6. **Appendix E: User Interface Navigation Diagram and Screen Layouts**
7. **Appendix F: Physical Architectural Design**
8. **Appendix G: Design Procedures**
9. **Appendix H: Gantt Chart**
10. **Appendix I: Elaboration Phase Prototypes**

System Requirements

Functional Requirements

- Newsletter Sign Up
- Mass Communication
- Donor Database
- Donor Donation Link
- Interactive Calendar
- Volunteer Sign Up
- Volunteer Database
- File Retrieval System
- Student Database
- Grant Database
- Report Generation

Non-Functional Requirements

- Easy Navigation
- Differing Security Levels
- Uninterrupted Availability
- Consistency
- Centralization
- Bi-Lingual

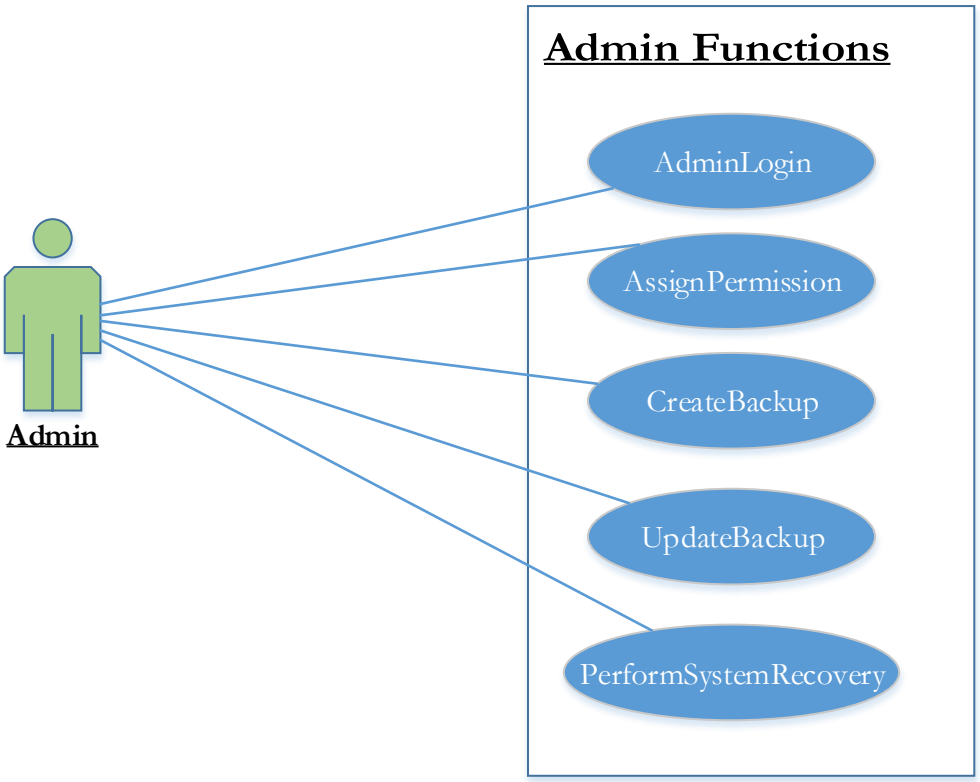
Requirement Details

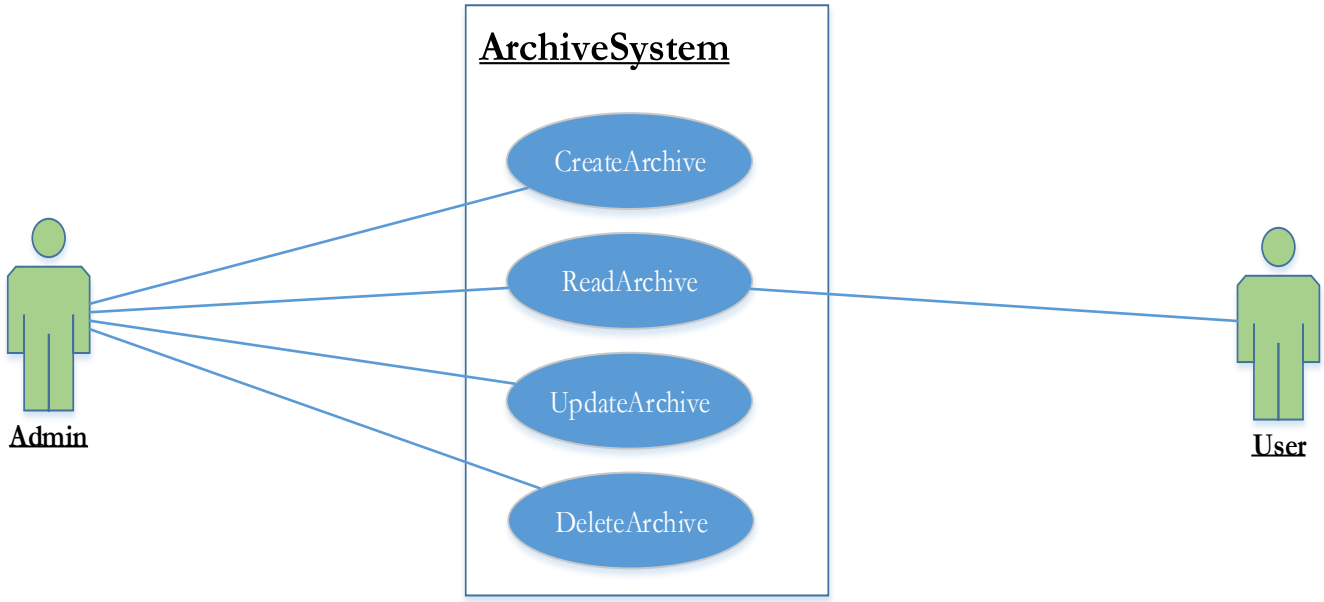
- The system will provide a news transmission page to communicate organization news.
- The system will provide a mass communication tool to communicate to people connected to the connections.
- The system will provide a donor database to connect, and keep track of, to donors of the organization.
- The system will provide a direct donation link for easy donation capabilities.
- The system will provide an interactive calendar to spotlight upcoming events and programs in one location.
- The system will provide volunteer sign up for persons wanting to get involved.
- The system will provide a volunteer database to keep track of, and communicate with, volunteers.
- The system will provide a storage and retrieval tool for all files, in one central location.
- The system will provide a student database to keep track of all students and their attributes.
- The system will provide a database for grants, past and future.

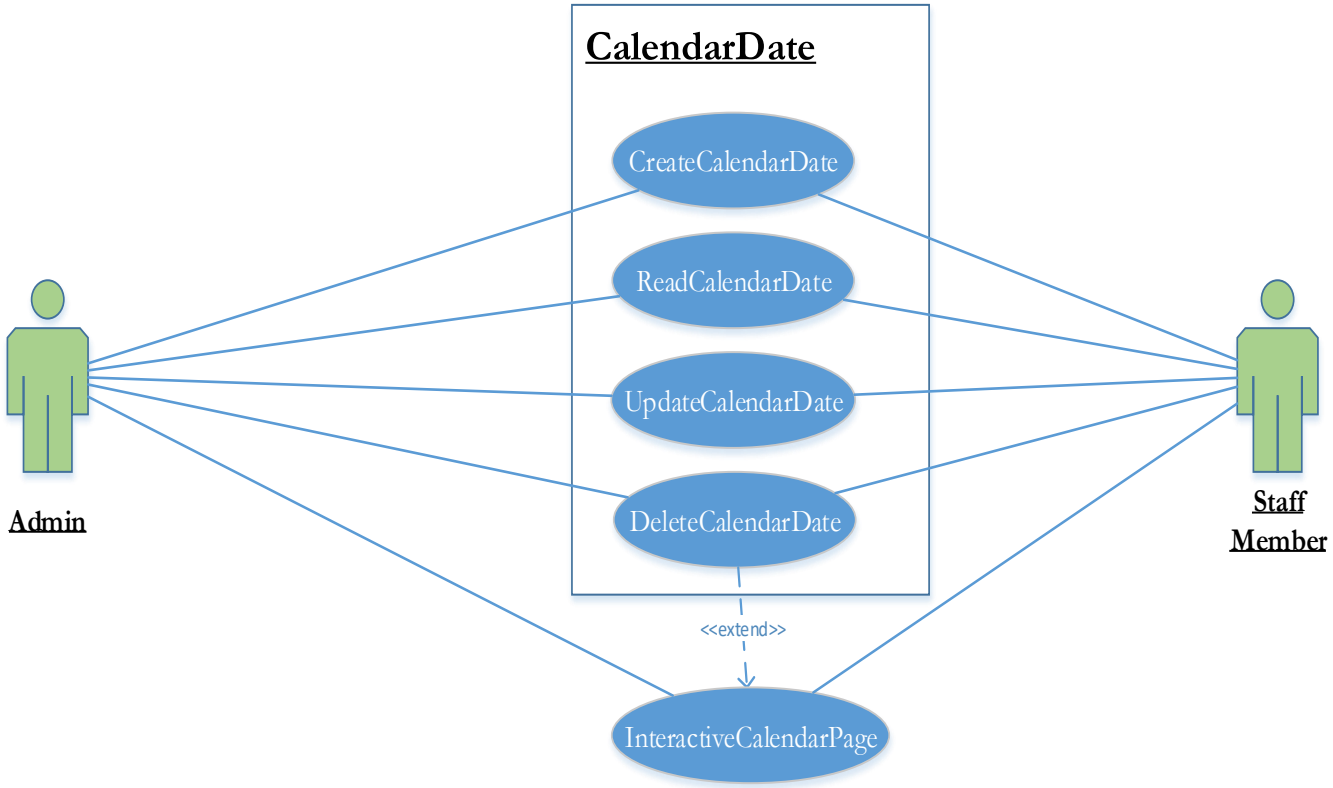
- The system will provide a report generation tool to compile information on each facet of the organization.

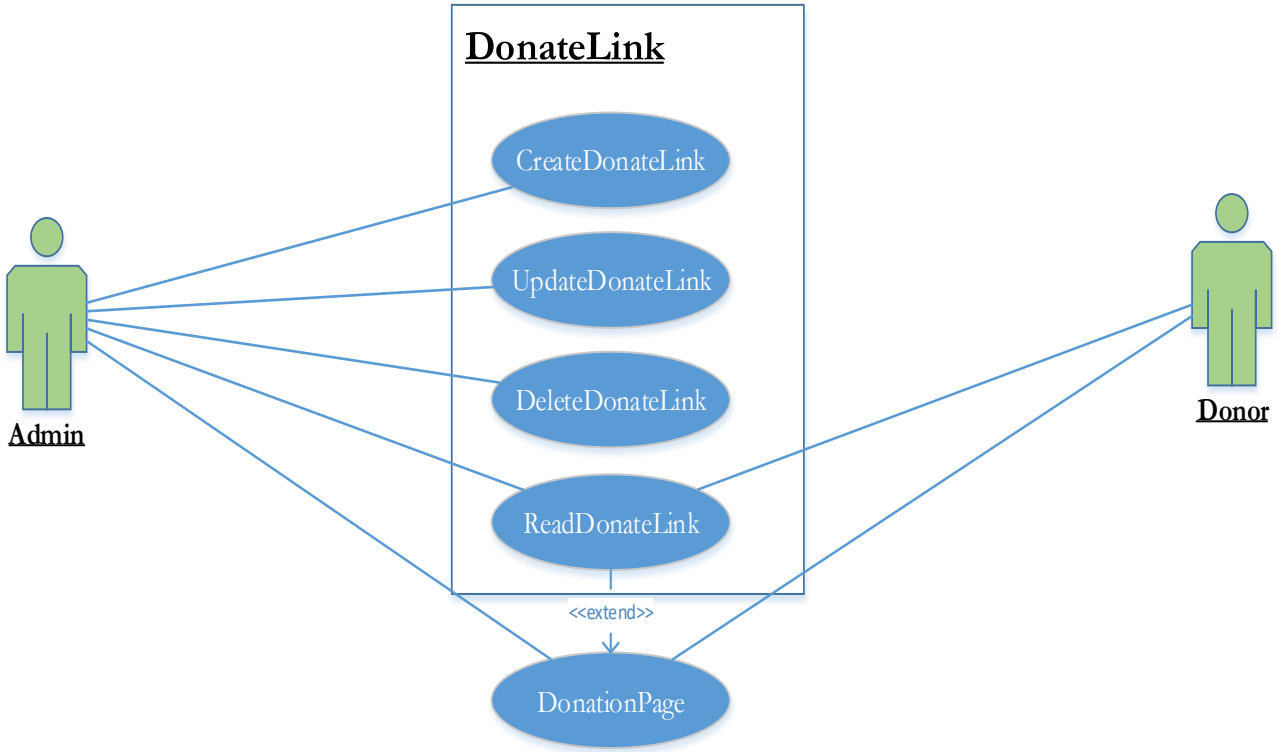
*The graph and requirements above list the specific aspects of the system that need to work together to make the system operate properly. Functional requirements are what the system should do. It is the meat of the system. If these requirements are not met, then the system is lacking in its functionality. Non-functional requirements are limitations on the stated requirements that need to be addressed. They are the higher level requirements that allow each functional requirement to exist. These requirements also state many of the functional requirement parameters and boundaries.

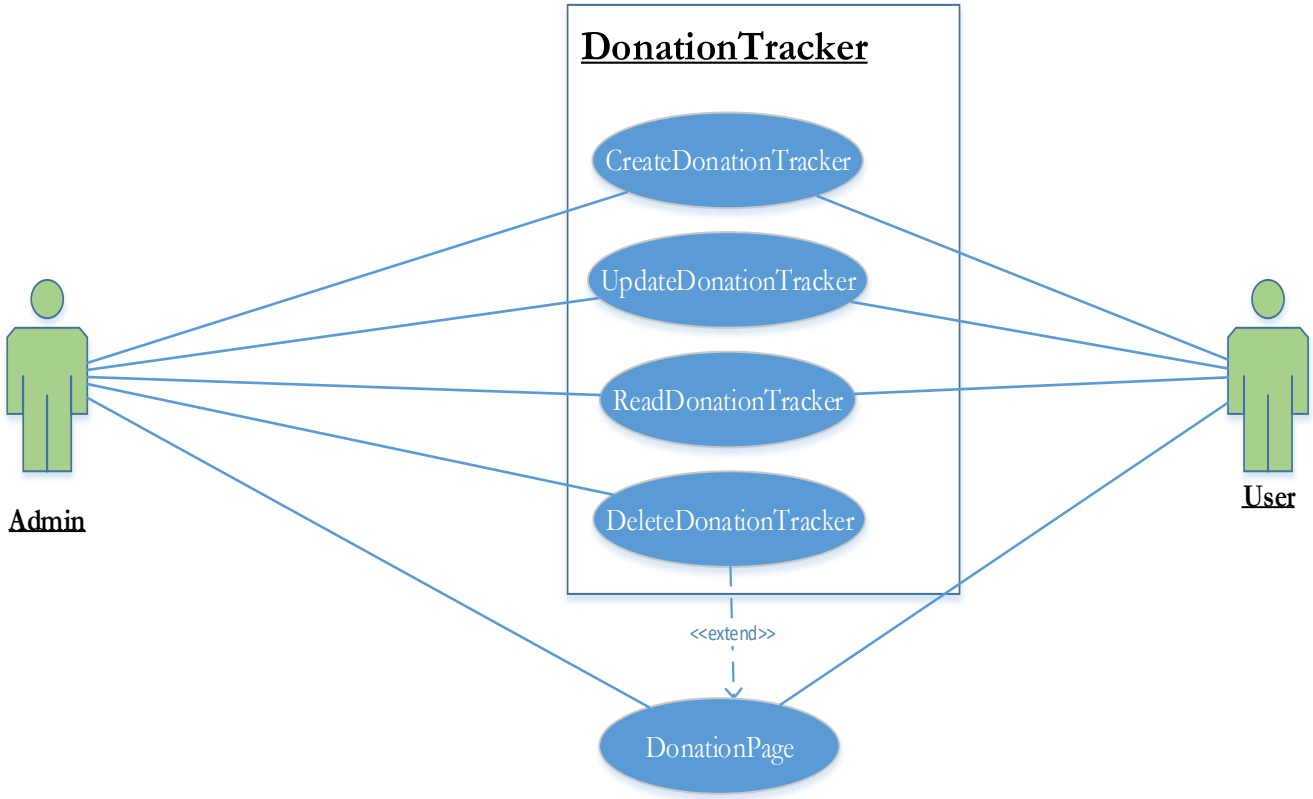
Appendix A: Use Cases, Use Case Diagram and Sequence Diagram

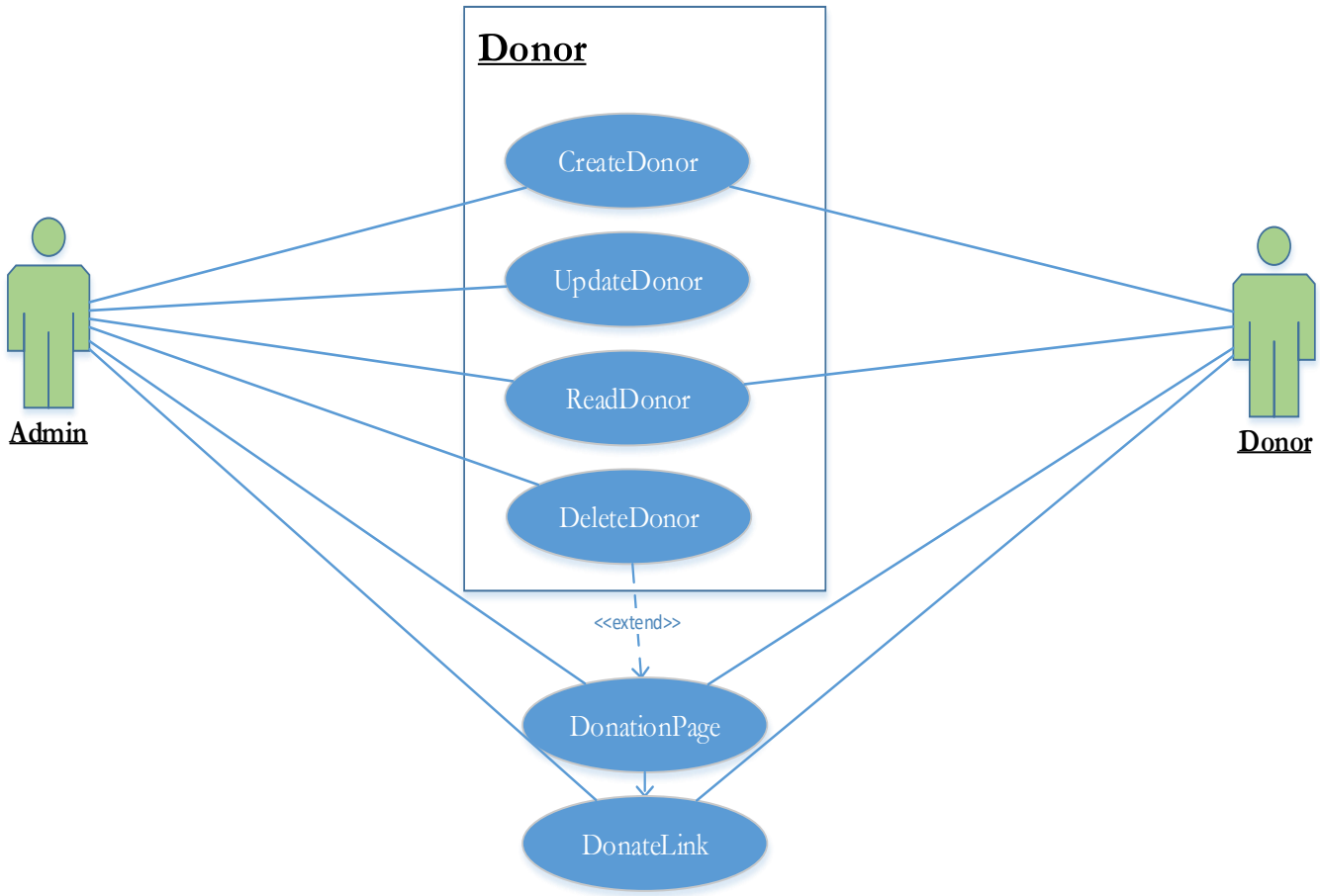


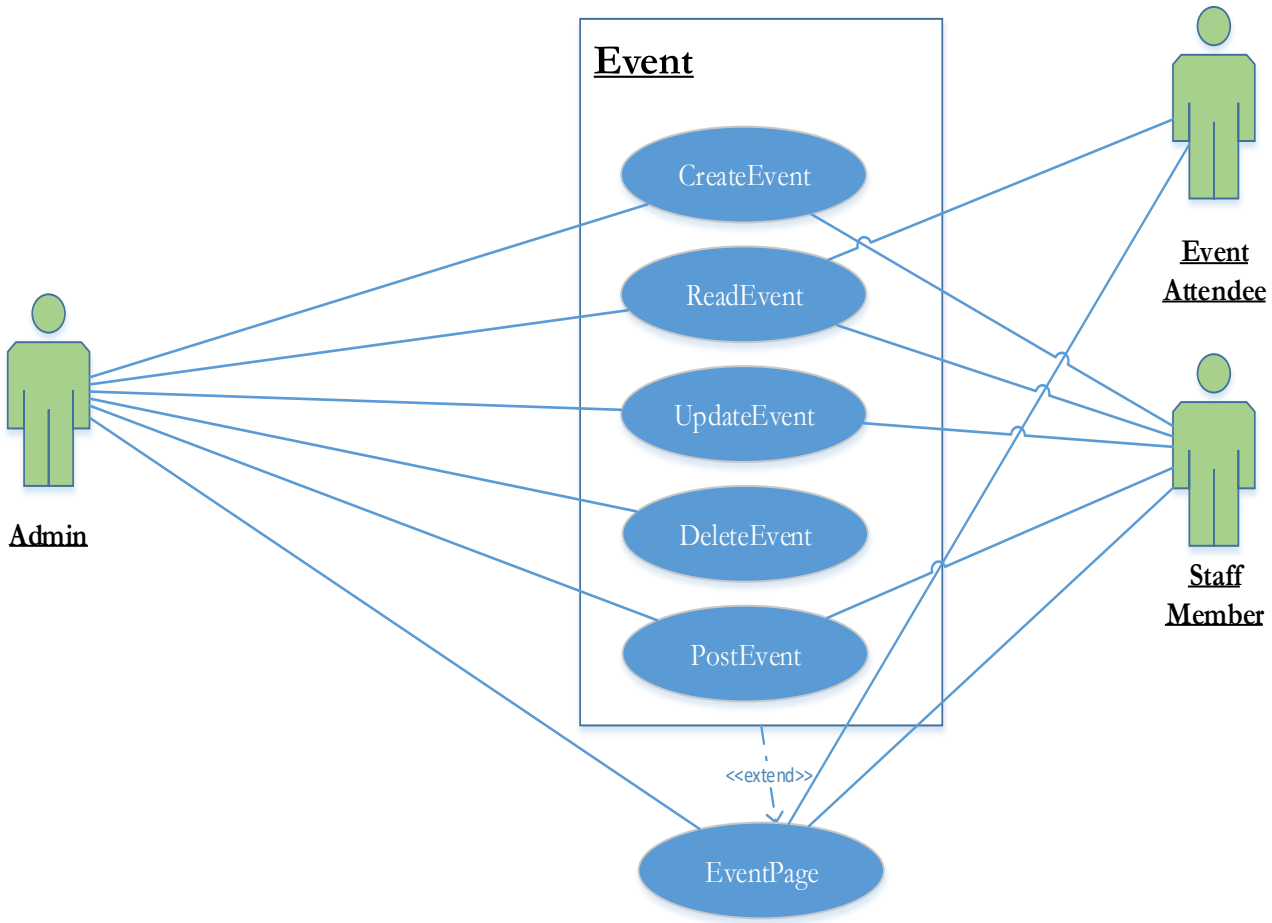

ArchiveSystem
CreateArchive
ReadArchive
UpdateArchive
DeleteArchive
Admin
User

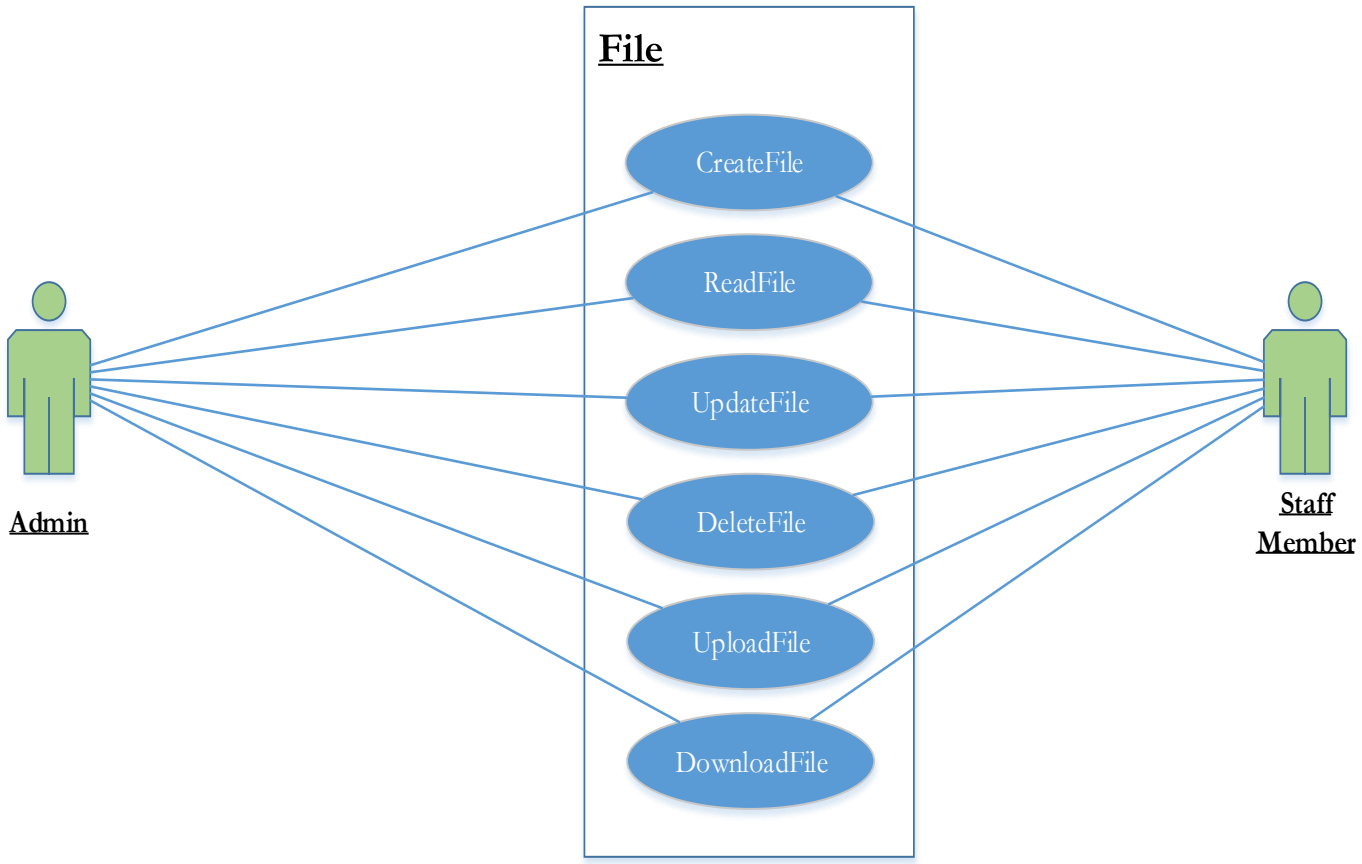

CalendarDate
CreateCalendarDate
ReadCalendarDate
UpdateCalendarDate
DeleteCalendarDate
<<extend>>
InteractiveCalendarPage
Admin
Staff Member

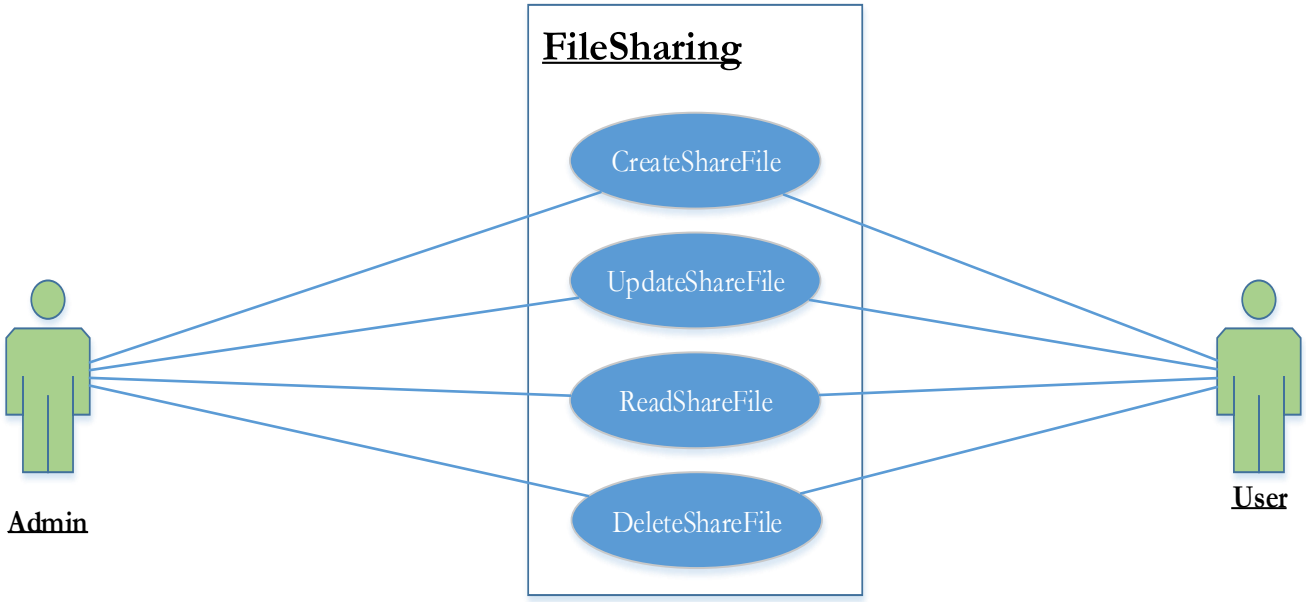

DonateLink
CreateDonateLink
UpdateDonateLink
DeleteDonateLink
ReadDonateLink
<<extend>>
DonationPage
Admin
Donor

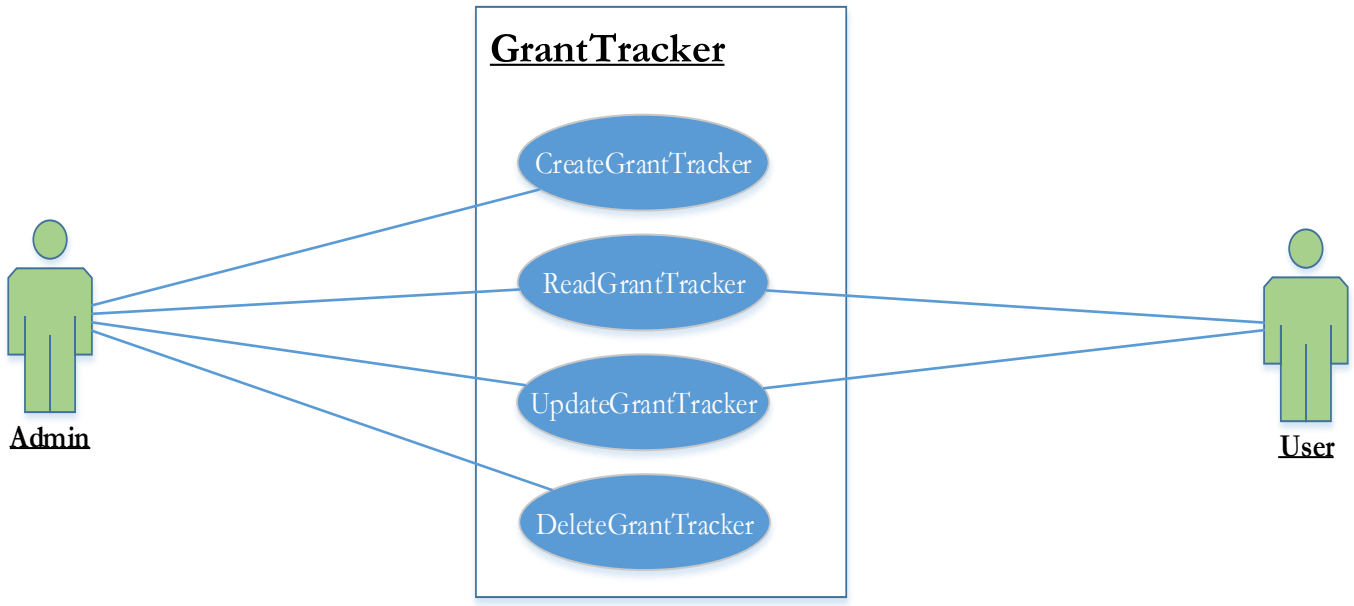

DonationTracker
CreateDonationTracker
UpdateDonationTracker
ReadDonationTracker
DeleteDonationTracker
<<extend>>
DonationPage
Admin
User

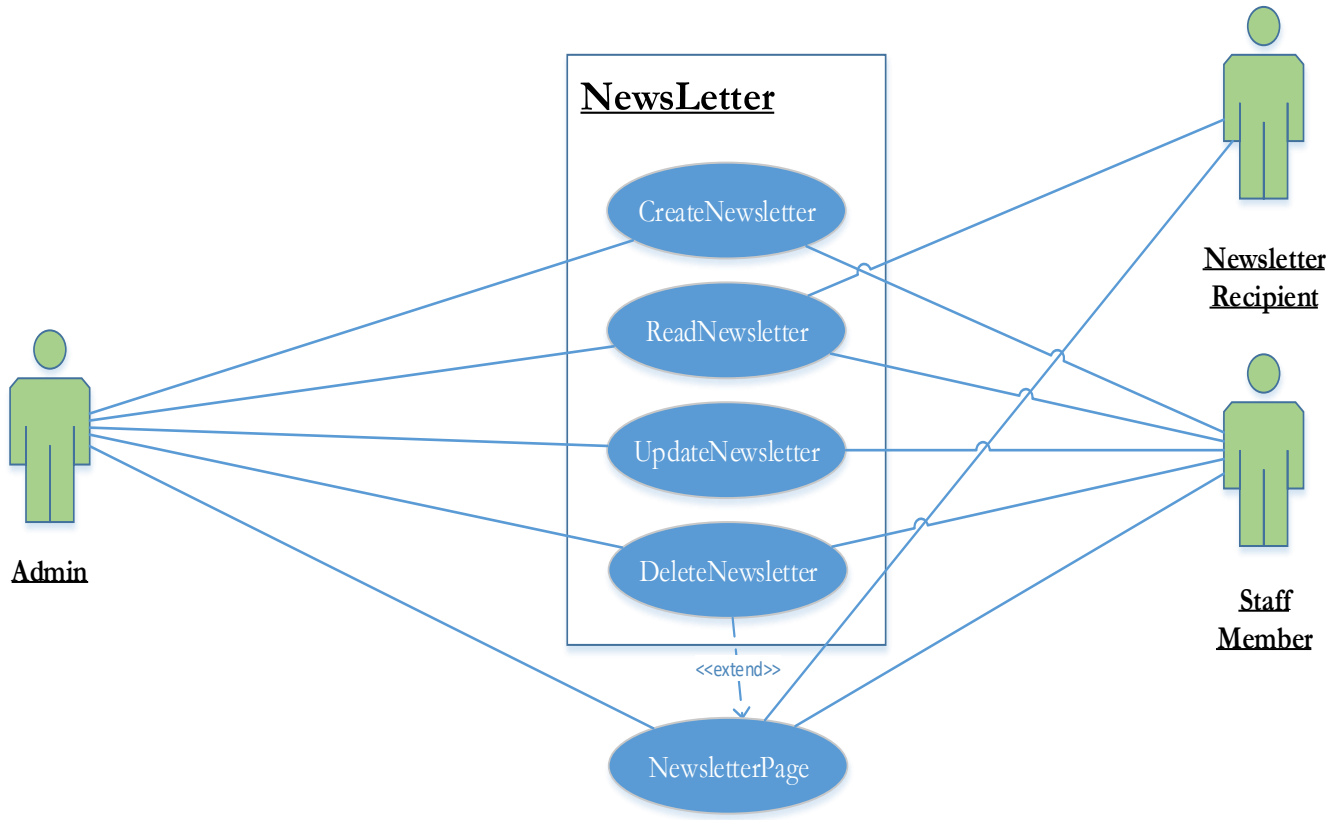

Donor
CreateDonor
UpdateDonor
ReadDonor
DeleteDonor
<<extend>>
DonationPage
DonateLink
Admin
Donor

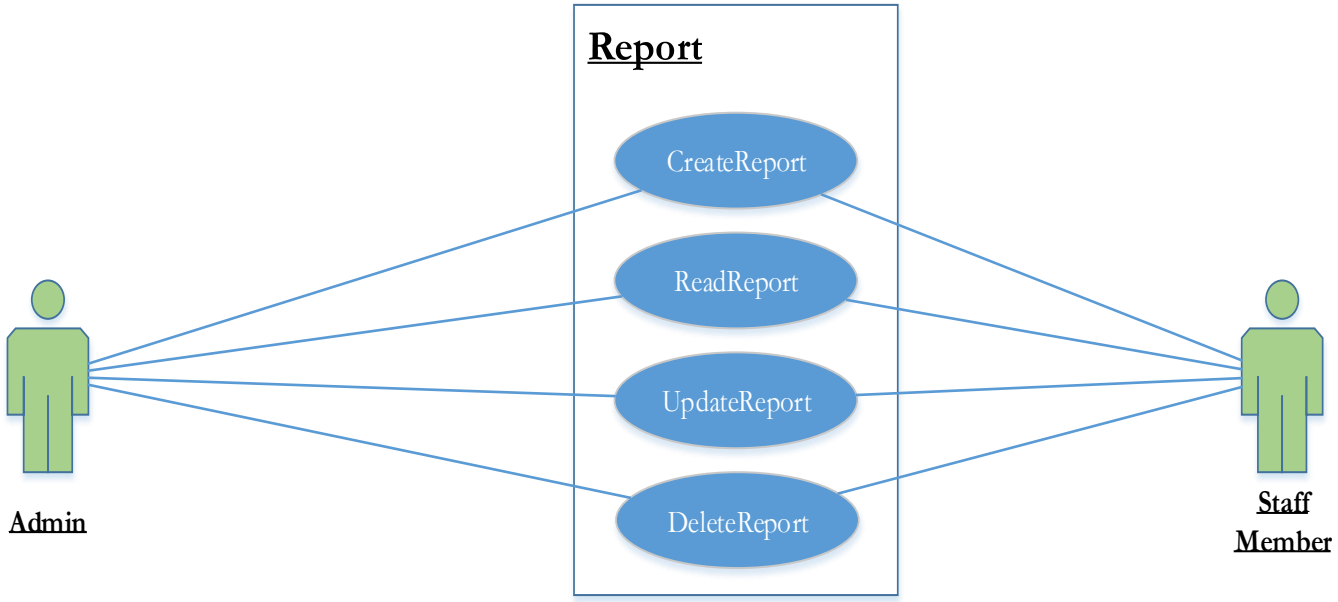

Event
CreateEvent
ReadEvent
UpdateEvent
DeleteEvent
PostEvent
<<extend>>
EventPage
Admin
Event Attendee
Staff Member

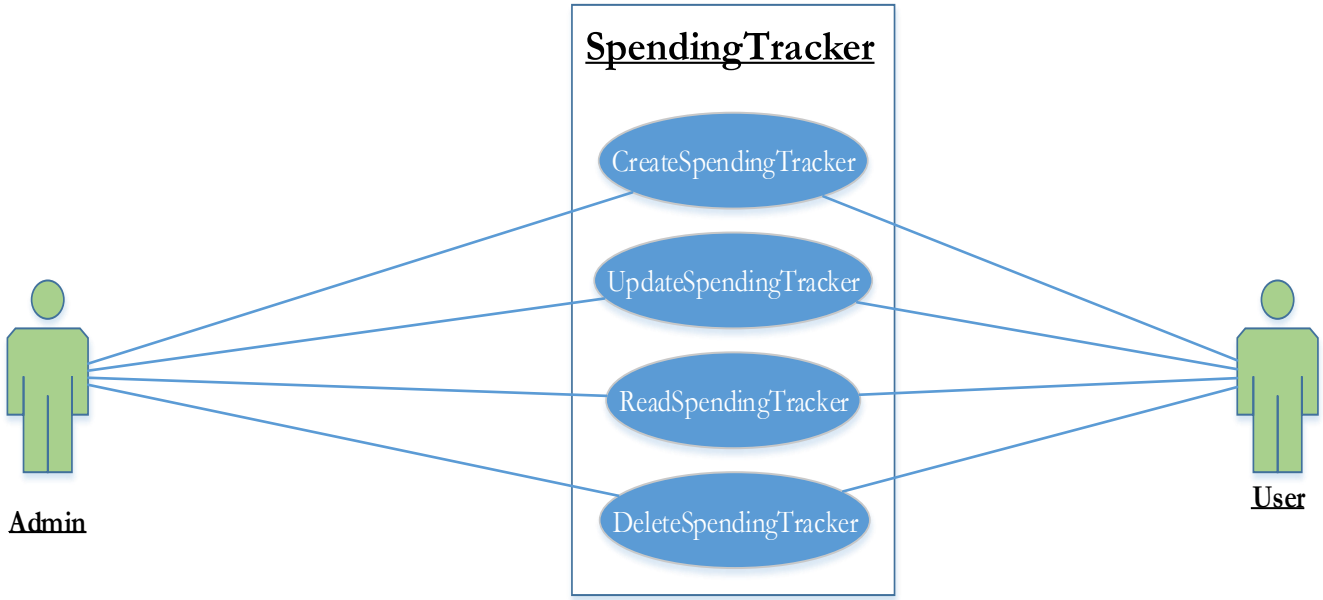

File
CreateFile
ReadFile
UpdateFile
DeleteFile
UploadFile
DownloadFile
Admin
Staff Member

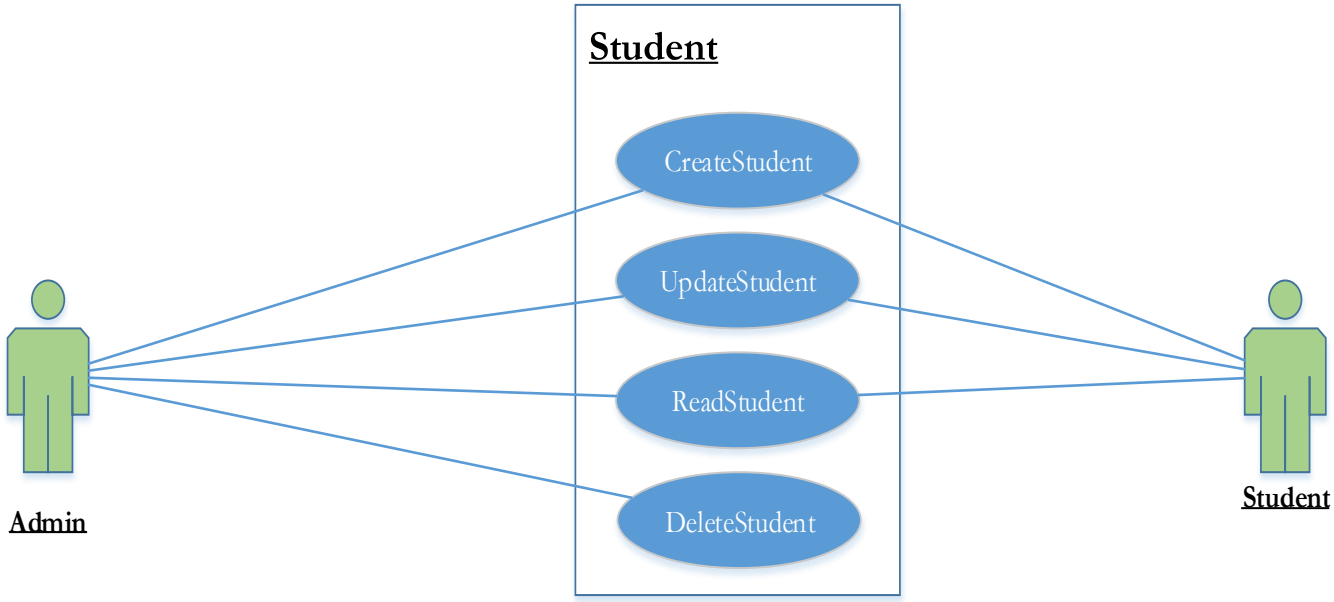

FileSharing
CreateShareFile
UpdateShareFile
ReadShareFile
DeleteShareFile
Admin
User

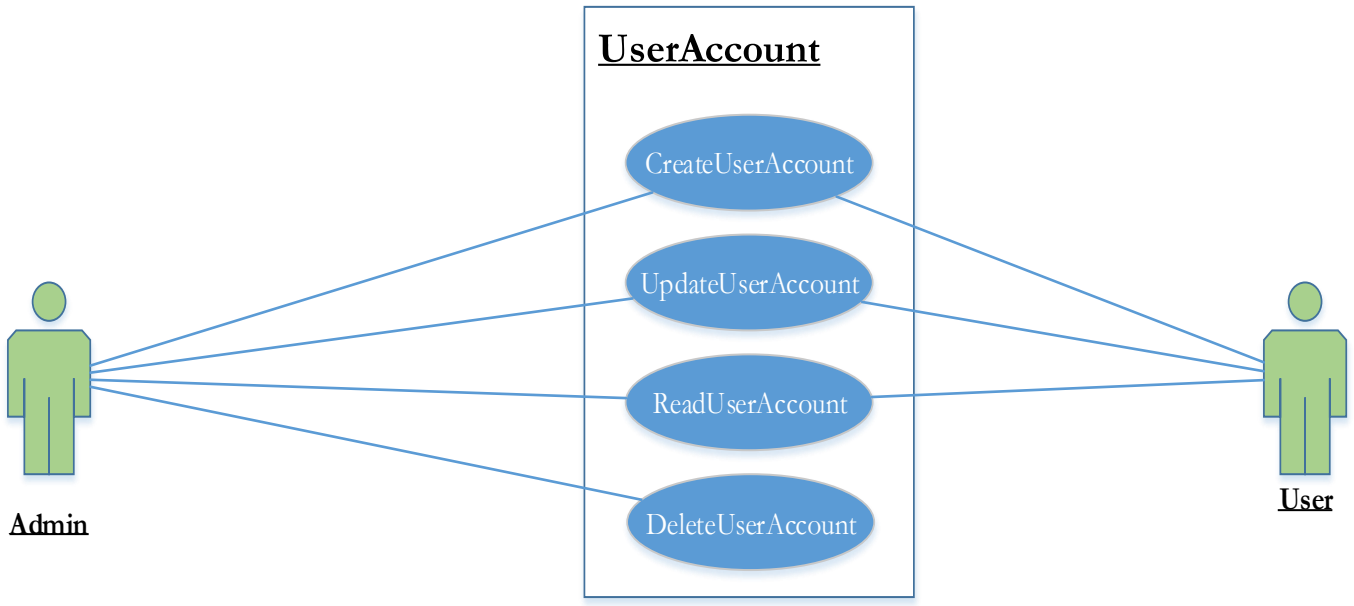

GrantTracker
CreateGrantTracker
ReadGrantTracker
UpdateGrantTracker
DeleteGrantTracker
Admin
User

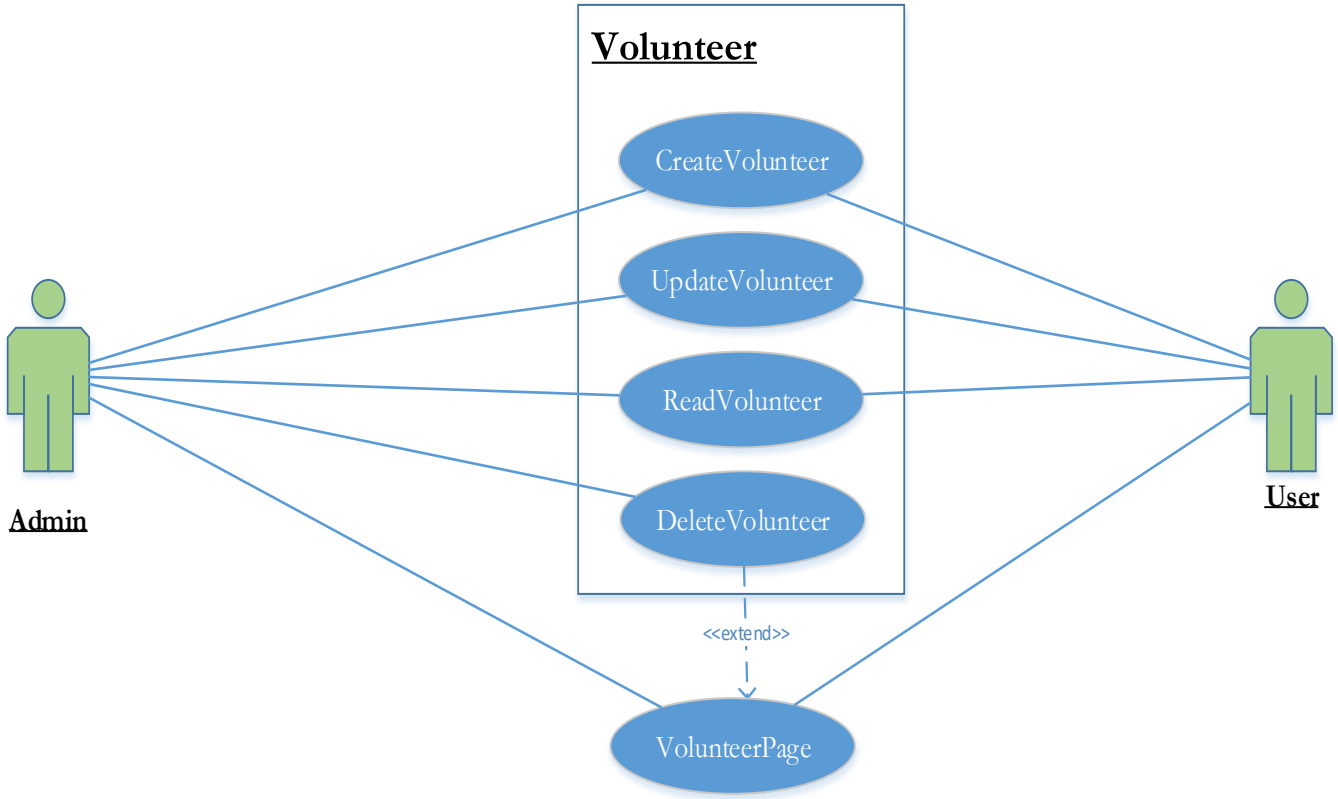

NewsLetter
CreateNewsletter
ReadNewsletter
UpdateNewsletter
DeleteNewsletter
<<extend>>
NewsletterPage
Admin
Newsletter Recipient
Staff Member


Report
CreateReport
ReadReport
UpdateReport
DeleteReport
Admin
Staff Member


SpendingTracker
CreateSpendingTracker
UpdateSpendingTracker
ReadSpendingTracker
DeleteSpendingTracker
Admin
User


Student
CreateStudent
UpdateStudent
ReadStudent
DeleteStudent
Admin
Student


UserAccount
CreateUserAccount
UpdateUserAccount
ReadUserAccount
DeleteUserAccount
Admin
User



Elaboration

Use Case Specification: <AdminLogin>

Version 1.0

Revision History

Date	Version	Description	Author
02/29/2016	1.0	AdminLogin	Maam Awa Ndaw

Table of Contents

1. AdminLogin **21**

1.1 Brief Description 21

2. Flow of Events **21**

2.1 Basic Flow 21

2.2 Alternative Flows 21

2.2.1 < System fails to recognize Admin's credentials > 21

2.2.2 < System could restrict access to Admin > 21

3. Special Requirements **21**

3.1 < Admin's credentials must be already stored in the system prior to logging in > 21

4. Pre-conditions **22**

4.1 < System must be available and in stand by mode > 22

4.2 < Admin's profile must be active in the system > 22

5. Post-conditions **22**

5.1 < Admin is logged in the system > 22

6. Extension Points **22**

6.1 < Admin's Log in credentials could be updated > 22

Use Case Specification: <AdminLogin>

1. AdminLogin

1.1 Brief Description

This use case describes the system's behavior when Admin attempts to log in.

2. Flow of Events

2.1 Basic Flow

The basic flow of events for Admin login is as follow:

- Admin enters username and password
- System authenticates Admin's credentials, username and password
- System grants access to Admin
- System makes all features editable

2.2 Alternative Flows

2.2.1 < System fails to recognize Admin's credentials >

After attempting to authenticate Admin's credentials, the system could fail at granting access to Admin. The issue might be persistent and in that case, Admin could try to reset password and repeat the process of logging in.

2.2.2 < System could restrict access to Admin >

Admin could successfully log in the system but is restricted access to certain features within the system. In this case, the system already recognized Admin's credentials and Admin could just navigate to settings and makes necessary changes to edit features settings.

3. Special Requirements

3.1 < Admin's credentials must be already stored in the system prior to logging in >

Admin can only be granted access in the system as Admin if it had already been previously set up.

4. Pre-conditions

4.1 **< System must be available and in stand by mode >**

Admin can log in the system if the system is available and in a waiting state.

4.2 **< Admin's profile must be active in the system >**

Admin must have an active profile in the system in order to be able to access the system as Admin.

5. Post-conditions

5.1 **< Admin is logged in the system >**

Admin has successfully logged in the system and all editable features can be accessed.

6. Extension Points

6.1 **< Admin's Log in credentials could be updated >**

Admin's log in credentials could be assigned to a different user. The new Admin would have different log in credentials.

Elaboration

Use Case Specification: <AssignPermission>

Version 1.0

Revision History

Date	Version	Description	Author
02/29/2016	1.0	AssignPermission	Maam Awa Ndaw

Table of Contents

1. AssignPermission 25

1.1 Brief Description 25

2. Flow of Events 25

2.1 Basic Flow 25

2.2 Alternative Flows 25

2.2.1 < System fails to record user's permission within the system > 25

2.2.2 < Admin assigns permission for user's profile but user can't access system > 25

3. Special Requirements 25

3.1 < User's profile creation request must be reviewed and approved > 25

4. Pre-conditions 26

4.1 < User must request profile creation > 26

4.2 < User's profile creation request must be approved > 26

5. Post-conditions 26

5.1 < User's profile is given access to system > 26

6. Extension Points 26

6.1 < User's profile could be given more access to information within system > 26

Use Case Specification: <AssignPermission>

7. AssignPermission

7.1 Brief Description

This use case describes how admin can assign permission to users of the system and therefore give access to information within the system.

8. Flow of Events

8.1 Basic Flow

The basic flow of events for assigning permission to users is as follow:

- Admin gets notification of user's profile creation request from system
- Admin reviews user's profile information
- Admin assigns permission to system's content to user's profile

8.2 Alternative Flows

8.2.1 < System fails to record user's permission within the system >

After reviewing the user's account and instructing the system to give access to user, there could be a possibility that the system fails to record user's permission within the system. In that case, the issue would have to be resolved then Amin would instruct the system again to assign permission the user.

8.2.2 < Admin assigns permission for user's profile but user can't access system >

There could be a scenario where the user can't access any content within the system after being given access. This issue might be persistent as the user logs in the system and is repeatedly denied access. Admin might have to delete the user's profile and have the user send another profile creation request, then admin could instruct the system to assign permission to the user.

9. Special Requirements

9.1 < User's profile creation request must be reviewed and approved >

Admin can only assign permission to a user's profile after the user's profile creation request has been reviewed and approved.

10. Pre-conditions

10.1 **< User must request profile creation >**

Admin can only assign permission to user's profile if a request had been previously sent and is waiting for review and approval.

10.2 **< User's profile creation request must be approved >**

Admin assign permission to user's profile only if it is approved for access to system's content.

11. Post-conditions

11.1 **< User's profile is given access to system >**

User can access information inside the system upon granted access.

12. Extension Points

12.1 **< User's profile could be given more access to information within system >**

A user could be given higher level of access within the system in the future depending on the circumstances, likewise, a user could be revoked access.

Elaboration

Use Case Specification: <CreateBackup>

Version <1.0>

Revision History

Date	Version	Description	Author
<2/29/2016>	<1.0>	CreateBackup	<James Ryg>

Table of Contents

1. CreateBackup **29**

1.1 Brief Description 29

2. Flow of Events **29**

2.1 Basic Flow 29

2.2 Alternative Flows 29

2.2.1 < Missing data alternative Flow> 29

3. Special Requirements **30**

3.1 < Legal Requirements > 30

4. Pre-conditions **30**

4.1 < Authentication Requirement > 30

5. Post-conditions **30**

5.1 < Cloud storage state > 30

6. Extension Points **30**

6.1 <Cancel> 30

Use Case Specification: <CreateBackup>

13. CreateBackup

13.1 Brief Description

The purpose of the CreateBackup use case is to allow Adelante the ability to quickly, and easily backup various parts of their system. It is an exclusive administrative function, and is not accessible by a standard user. In practice, only the webmaster, or Adelante executives will be able to make use of this for security reasons. CreateBackup allows a user to select multiple options, and will generate a compressed file that contains all data requested for backup, available for direct download and/or for secure cloud backup.

14. Flow of Events

14.1 Basic Flow

- The actor requests the backup menu.
- The window is displayed, if the actors log-in has sufficient permissions to have access to the system as a whole.
- The actor will select one or more from the following options in terms of backup storage.
    a. Website
    b. Database
    c. Logs
- The actor will select one of the options regarding save format.
    d. Download
    e. Cloud
    f. Both
- The actor will submit the request.
- The system will generate the backup based on the options.
- The system will return the backup, and/or upload it securely to the cloud.

14.2 Alternative Flows

14.2.1 < Missing data alternative Flow>

- The actor fails to provide information required to generate the backup.
- The actor views a prompt indicating what data is missing.

15. Special Requirements

15.1 **< Legal Requirements >**

This part of the system deals with sensitive user data that cannot be lost or leaked. Care should be taken to ensure that such data is stored in a secure manner.

16. Pre-conditions

16.1 **< Authentication Requirement >**

The actor must meet the following two conditions to use this feature.

- The actor must be logged in through the system.
- The actor must have authorization to access the feature.

17. Post-conditions

17.1 **< Cloud storage state >**

The actual system does not change, but if the actor selected ‘cloud’ or ‘both’ as an option, some amount of cloud storage will have been used, and the backup will be available for future use.

18. Extension Points

18.1 **<Cancel>**

At any point, the actor can leave the page, and cancel any backup. This can also be selected while the backup is being built.

Elaboration

Use Case Specification: <UpdateBackup>

Version 1.0

Revision History

Date	Version	Description	Author
02/20/2016	1.0	UpdateBackup	Valencia Richard

Table of Contents

1. UpdateBackUp **34**

1.1. Brief Description 34

2. Flow of Events **34**

2.1. Basic Flow 34

2.2. Alternative Flows 34

2.2.1. < Backup(s) Not Created > 34

2.2.2. < Update Not Approved> 34

3. Pre-conditions **35**

3.1. < A Backup Must Exist> 35

3.2. < Login Credentials Verified> 35

4. Post-conditions **35**

4.1. < Donation Link Updated > 35

5. Extension Points **35**

5.1. < Cancel> 35

Use Case Specification: <UpdateBackup >

1. UpdateBackUp

1.1. Brief Description

This use case allows an actor to make changes with systen aspects that have already been backed up. These Backups can be edited by administration. For secuirty and continuitty reasons, this feature is only available to the head of Adalnte. For security, these backups may contain sensitive information that must be guarded. In case of a system failure these backups must be maintained to bring the system back to life.

2. Flow of Events

2.1. Basic Flow

- The Actor initiates the Use Case by accessing the AHA system
- The actor inputs their login information
- The system accepts the information
- The actor locates locates the appropriate Backup file
- The actor attempts the save the changes
- The system accesses the credentials for proper clearance
- Clearance level is high enough to make Updates
- The system updates the Backup

2.2. Alternative Flows

2.2.1. < Backup(s) Not Created >

- The Actor initiates the Use Case by accessing the AHA system
- The actor inputs their login information
- The system accepts the information
- The actors go into the system to locate the code the Backup, but it is missing

2.2.2. < Update Not Approved>

- The Actor initiates the Use Case by accessing the AHA system
- The Actor initiates the Use Case by accessing the AHA system
- The actor inputs their login information
- The system accepts the information
- The actor locates goes into the backend to the line of code representing the Donation Link
- The actor updates code representing the Link
- The actor attempts the save the changes

- The system accesses the credentials for proper clearance
- Clearance level is not high enough to make Updates
- The system alerts the User that changes have not been made

3. Pre-conditions

3.1. < A Backup Must Exist>

Backups must be made previously in order for edits to be performed on them.

3.2. < Login Credentials Verified>

An actor must have the proper login and password to access the system to make updates to go in and change the code and update the Backups

4. Post-conditions

4.1. < Donation Link Updated >

The backup information that will be updated

5. Extension Points

5.1. < Cancel>

An actor can cancel the attempt to Update the Donation Link

Elaboration

Use Case Specification: <CreateArchive>

Version <1.0>

Revision History

Date	Version	Description	Author
<03/18/2016>	<1.0>	CreateArchive	<William Benedict>

Table of Contents

1. CreateArchive **39**

1.1 Brief Description 39

2. Flow of Events **39**

2.1 Basic Flow 39

2.2 Alternative Flows 39

2.2.1 < Adding other newsletters to archive > 39

2.2.2 < Archive already exists > 39

3. Pre-conditions **39**

3.1 < Login & Authorization > 39

3.2 < Newsletter Exists > 40

4. Post-conditions **40**

4.1 < Archive Existence > 40

5. Extension Points **40**

5.1 < Cancel > 40

Use Case Specification: <CreateArchive>

19. CreateArchive

19.1 Brief Description

This use case illustrates how an actor can add newsletter or announcement to a history of previous newsletters and announcements.

20. Flow of Events

20.1 Basic Flow

- The actor navigates to the newsletter page
- The actor selects to add a new newsletter or announcements
- Upon adding the new newsletter or announcement, the actor will be prompted to add the previous newsletter to the archive.
- The actor names the archived newsletter or announcement.
- The actor saves the archived newsletter or announcement.

20.2 Alternative Flows

20.2.1 < Adding other newsletters to archive >

- The actor navigates to the archive page.
- The actor selects where to upload the newsletter from
- The actor navigates to where the newsletter is saved and uploads it.

20.2.2 < Archive already exists >

- The actor adds a newsletter to the archive
- The actor is prompted that a newsletter with the same name already exists
- The actor is redirected back to rename the newsletter.

21. Pre-conditions

21.1 < Login & Authorization >

The actor must be logged in and have access to modify the active newsletter. Any actor that may modify the active newsletter may create a new archive

21.2 **< Newsletter Exists >**

In order to be added to the archive, a newsletter must already be created. The exception to this is newsletters that are added directly to the archive, such as newsletters created before this system's existence

22. Post-conditions

22.1 **< Archive Existence >**

The archive will be created or not created

23. Extension Points

23.1 **< Cancel >**

At any point in the archive creation process, the actor may stop the creation of an archive until it is actually created.

Elaboration

Use Case Specification: <Delete Archive>

Version <1.0>

Revision History

Date	Version	Description	Author
<03/18/2016>	<1.0>	DeleteArchive	<William Benedict>

Table of Contents

1. Delete Archive **43**

1.1 Brief Description 43

2. Flow of Events **43**

2.1 Basic Flow 43

2.2 Alternative Flows 43

2.2.1 < No Permissions to delete Archive > 43

2.2.2 < No selection made > 43

3. Pre-conditions **43**

3.1 < Login & Authorization > 43

3.2 < Archive Existence > 44

4. Post-conditions **44**

4.1 < Change of State > 44

5. Extension Points **44**

5.1 < Cancel > 44

Use Case Specification: <Delete Archive>

24. Delete Archive

24.1 Brief Description

The purpose of this use case is to illustrate how an actor can remove a newsletter or announcement from a list of past newsletters and announcements called an archive. Although all users will have access to view prior archives, only certain actors may have the permissions to delete them

25. Flow of Events

25.1 Basic Flow

- The actor navigates to the list of previous newsletters and announcements.
- The actor selects the previous newsletters and announcements to remove
- The actor receives a prompt asking if they are sure they wish to delete x number of past newsletters/announcements
- The actor confirms by entering their password

25.2 Alternative Flows

25.2.1 < No Permissions to delete Archive >

- The actor is prompted that they do not have permission to delete from the archives
- The actor is redirected back to the archive list

25.2.2 < No selection made >

- The actor does not select any newsletters or announcements to delete
- The actor is prompted they have not made any selection for removal
- The actor is redirected back to the archive page.

26. Pre-conditions

26.1 < Login & Authorization >

The actor must be signed in and have a permission level high enough to delete from the archive

26.2 **< Archive Existence >**

The archive must exist in the first place in order to be deleted.

27. Post-conditions

27.1 **< Change of State >**

The archive may be either deleted or not deleted.

28. Extension Points

28.1 **< Cancel >**

At any point in time, the actor my stop the process to delete an archive, until the archive is actually deleted.

Elaboration

Use Case Specification: <ReadArchive>

Version <1.0>

Revision History

Date	Version	Description	Author
03/20/2016	1.0	ReadArchive	Chad Tracy

Table of Contents

1. ReadArchive **47**

1.1 Brief Description 47

2. Flow of Events **47**

2.1 Basic Flow 47

2.2 Alternative Flows 47

2.2.1 < Credential Error > 47

3. Special Requirements **47**

3.1 < Special Privileges > 47

4. Pre-conditions **47**

4.1 < Archive Exists > 47

5. Post-conditions **47**

5.1 < Archive is Read > 47

6. Extension Points **48**

6.1 <Cancel> 48

Use Case Specification: <ReadArchive>

29. ReadArchive

29.1 Brief Description

Allows actor to view the information stored in the archive. This power should only be for AHA employees

30. Flow of Events

30.1 Basic Flow

- Actor log-in to system with employee credentials
- System grants actor access
- Actor navigates to archive
- Information in archive is available to Actor

30.2 Alternative Flows

30.2.1 < Credential Error >

- Actor attempts to log-in to system
- System rejects actor based on credentials
- System returns error message to actor

31. Special Requirements

31.1 < Special Privileges >

The correct privileges must be given to AHA employees to ensure that they can access the archive

32. Pre-conditions

32.1 < Archive Exists >

Archive must exist in order to ensure that the employees can view the information

33. Post-conditions

33.1 < Archive is Read >

The actor has accessed the information and read the data

34. Extension Points

34.1 **<Cancel>**

At any point the actor should be able to cancel the process and exit the system

Elaboration

Use Case Specification: <UpdateArchive>

Version <1.0>

Revision History

Date	Version	Description	Author
<03/04/2016>	<1.0>	UpdateArchive	Chad Tracy

Table of Contents

1. UpdateArchive **51**

1.1 Brief Description 51

2. Flow of Events **51**

2.1 Basic Flow 51

2.2 Alternative Flows 51

2.2.1 < System Rejects User > 51

2.2.2 < System Error: Update Fails > 51

3. Special Requirements **51**

3.1 < System Must Recognize Admins > 51

4. Pre-conditions **52**

4.1 < Archives Must Exist > 52

5. Post-conditions **52**

5.1 < Archive Is Updated> 52

Use Case Specification: <UpdateArchive>

35. UpdateArchive

35.1 Brief Description

Data that is more historic for the company and not of much use will be put in archive files. When the archive needs to be updated the admin accesses the files within the system and adds any necessary data to the archives

36. Flow of Events

36.1 Basic Flow

- User logs-in to system with credentials (username and password)
- System accepts user and grants them admin privileges
- Admin navigates to the archive within the system
- Admin updates archives with new data and saves it to the archive
- System saves the data entered by the admin
- Archive is updated

36.2 Alternative Flows

36.2.1 < System Rejects User >

- User logs-in to system with credentials (username and password)
- System rejects user
- User is not given admin status
- User can't make changes to archive
- System prompts user to reenter credentials

36.2.2 < System Error: Update Fails >

- Admin successfully logs-in to system and makes updates to the archive
- Admin saves changes and exits the system
- System fails to save changes made by the admin
- Archives remain in the state they were at the beginning of the flow

37. Special Requirements

37.1 < System Must Recognize Admins >

The correct users must be recognized as admins within the system. This allows the admins to navigate through the system and make updates to the archive

38. Pre-conditions

38.1 **< Archives Must Exist >**

In order for the admin to make updates to the archive it must already exist in the system.

39. Post-conditions

39.1 **< Archive Is Updated>**

After the flow of work, the archives should be updated with any data the admin has submitted to the system.

Elaboration

Use Case Specification: <CreateCalendarDate>

Version <1.0>

Revision History

Date	Version	Description	Author
<03/19/2016>	<1.0>	CreateCalendarDate	<Hermilo Acuna>

Table of Contents

1. CreateCalendarDate **55**

1.1 Brief Description 55

2. Flow of Events **55**

2.1 Basic Flow 55

. The actor will submit the request 56

2.2 Alternative Flows 56

2.2.1 < Missing data > 56

2.2.2 < Corresponding Event > 56

3. Special Requirements **56**

3.1 < Event Date Verification > 56

4. Pre-conditions **56**

4.1 < Authentication Requirement > 56

5. Post-conditions **56**

5.1 < Calendar Date Created > 56

5.2 < Cloud Storage Date > 56

6. Extension Points **57**

6.1 <Cancel> 57

Use Case Specification: <CreateCalendarDate>

40. CreateCalendarDate

40.1 Brief Description

Once the actor has been verified, they will create a new date on the calendar for whatever occasion the actor wishes to announce. The actor can create a date at any point in time including previous dates if they wish to maintain for their users. Depending on the selection on the type of calendar date it will sync to the appropriate data.

41. Flow of Events

41.1 Basic Flow

- The Actor request the Create Calendar Date edit page.
- The system verifies the actor's system access
- The page is now unlocked for the actor to create a new calendar date
- The actor will select one or more of the following option to be created
    a. Title
    b. Description
    c. Date
    d. Links

- The actor will submit the request
- The system will create a new calendar date based on the actor's option
- The system will upload the calendar date to the cloud
- The system will check if there is a corresponding event
- The system will sync both dates if an event exist

41.2 Alternative Flows

41.2.1 < Missing data >

- The actor fails to provide information required to generate the calendar date.
- The actor views a prompt indicating what data is missing.

41.2.2 < Corresponding Event >

The system will sync the calendar date with its corresponding event if it exists

42. Special Requirements

42.1 **< Event Date Verification >**

If a calendar date is created the system will need to verify if there is already an event with a matching title and corresponding date.

43. Pre-conditions

43.1 **< Authentication Requirement >**

The actor must meet the following two conditions to use this feature.

1) The actor must be logged in through the system.
2) The actor must have authorization to access the feature.

44. Post-conditions

44.1 **< Calendar Date Created >**

The calendar has now been updated with a new date that the actor has created. Users will now be able to navigate through the calendar and see the dates that the organization has set up.

44.2 **< Cloud Storage Date >**

The system will upload the calendar date to the cloud in order to save the date.

45. Extension Points

45.1 **<Cancel>**

The actor at any point during the create a calendar date page can cancel the process. This will delete any current data that the actor has inputted up to this point and resume o the system as is before the actor began to create a calendar date.

Elaboration

Use Case Specification: <DeleteCalendarDate>

Version <1.0>

Revision History

Date	Version	Description	Author
<03/18/2016>	<1.0>	DeleteCalendarDate	<William Benedict>

Table of Contents

1. DeleteCalendarDate **60**

1.1 Brief Description 60

2. Flow of Events **60**

2.1 Basic Flow 60

2.2 Alternative Flows 60

2.2.1 < Lack of permission to delete flow > 60

3. Special Requirements **60**

3.1 < Event Integration> 60

4. Pre-conditions **60**

4.1 <Login & Authentication Requirement> 60

4.2 <Calendar Date Existence Requirement> 60

5. Post-conditions **61**

5.1 < CalendarDate State > 61

6. Extension Points **61**

6.1 <Cancel > 61

Use Case Specification: <DeleteCalendarDate>

46. DeleteCalendarDate

46.1 Brief Description

The DeleteCalendarDate use case illustrates functionality regarding the removal of events from calendar dates. Only users with access to the event management mechanics will have the ability to delete events from the calendar.

47. Flow of Events

47.1 Basic Flow

- The actor selects the event from the calendar.
- The actor selects delete
- The actor submits deletion request

47.2 Alternative Flows

47.2.1 < Lack of permission to delete flow >

- The actor is displayed a message indicating they do not possess the specific rights to delete an event
- The actor is directed back to the calendar

48. Special Requirements

48.1 < Event Integration>

For proper usage, the system requires event functionality.

49. Pre-conditions

49.1 <Login & Authentication Requirement>

The actor must be logged in, and have a level of permission sufficient to both create and delete events

49.2 <Calendar Date Existence Requirement>

There must be an event linked to a calendar date to delete.

50. Post-conditions

50.1 **< CalendarDate State >**

The event linked to the calendar date can either be deleted or not deleted.

51. Extension Points

51.1 **<Cancel >**

At any point, the actor can cancel deletion until the report is actually deleted

Elaboration

Use Case Specification: <ReadCalendarDate>

Version <1.0>

Revision History

Date	Version	Description	Author
<03/18/2016>	<1.0>	ReadCalendarDate	<James Ryg>

Table of Contents

1. ReadCalendarDate **210**

1.1 Brief Description 47

2. Flow of Events **34**

2.1 Basic Flow 34

2.2 Alternative Flows 34

2.2.1 < Calendar Date does not exist > 34

3. Special Requirements **34**

3.1 <Calendar integration requirement > 47

4. Pre-conditions **35**

4.1 < Login & Authentication requirement> 35

5. Extension Points **48**

6.1 <Cancel> 48

Use Case Specification: <ReadCalendarDate>

52. ReadCalendarDate

52.1 Brief Description

This is a small use case, it does not involve editing, destruction, or creation of data objects. It simply conveys attributes of specific calendar dates. Different dates may have different access requirements. There is no post-condition due to it not modifying any data states.

53. Flow of Events

53.1 Basic Flow

- An actor or object requests date information.
- The actors credentials are checked.
    - If the actor does not have the requisite credentials, do not display the information
- The calendar sends the object or actor information about a specific date or set of dates.

53.2 Alternative Flows

53.2.1 < Calendar Date does not exist >

- If no accessible calendar dates exist, display the calendar as blank.

54. Special Requirements

54.1 < Calendar integration Requirement >

Depending on how Adelante wants to handle dates and times, differing calendar integration services may be used. In that case, it needs to be integrated with the system

55. Pre-conditions

55.1 < Login & Authentication Requirement >

For *certain* conditions, there may be a login requirement or permissions requirement to view dates. This precondition is only valid in those specific cases.

56. Extension Points

56.1 **<Cancel>**

This is only called when there is no valid, or accessible data. There would be no other reason for an actor to simply cancel reading a date.

Elaboration

Use Case Specification: <UpdateCalendarDate>

Version 1.0

Revision History

Date	Version	Description	Author
02/27/2016	1.0	UpdateCalendarDate	Maam Awa Ndaw

Table of Contents

1. UpdateCalendarDate **68**

1.1 Brief Description 68

2. Flow of Events **68**

2.1 Basic Flow 68

2.2 Alternative Flows 68

2.2.1 < Unresponsiveness of Interactive Calendar> 68

2.2.2 < System fail to authenticate Admin credentials> 68

3. Special Requirements **69**

3.1< Admin role and credentials must be created and stored in the system or permission assigned> 69

4. Pre-conditions **69**

4.1 < User must have admin role or permission to update calendar date > 69

4.2 < Calendar Date already exist> 69

5. Post-conditions **69**

5.1 < Calendar Date is updated > 69

6. Extension Points **69**

6.1 < Additional updates to calendar date> 69

Use Case Specification: <UpdateCalendarDate>

57. UpdateCalendarDate

57.1 Brief Description

This use case describes how admin or a staff member can update a date in the interactive calendar, given that the calendar date already exists. This primarily allows someone with admin permission or access to editable features of the calendar, to log in the system and makes changes such as date, time and/ or any reminders, details, etc.… associated with it.

58. Flow of Events

58.1 Basic Flow

The basic flow of events for updating a calendar date is as follow:

- Admin or staff member enters username and password to log in system
- System authenticates admin or specific user
- System or staff member makes all editable features in the system available
- Admin or staff member navigates to interactive calendar
- Admin or staff member searches for a specific date in calendar
- Admin or staff member edits the date and then makes changes such as location, time, description, or parties associated with calendar date

58.2 Alternative Flows

58.2.1 < Unresponsiveness of Interactive Calendar>

There could be a possibility that Admin or staff member logs in the system but the interactive Calendar is not responding to user input and displays the same information after update is made by Admin. This could be due to a bug inside the system. In that case, Admin will have to get the issue fixed before proceeding with the normal flow of updating the calendar date.

58.2.2 < System fail to authenticate Admin credentials>

There could be a scenario where a system error causes the admin to be restricted access whenever s/he enters a username and password to login. It could possibly due to a technical issue that needs to be addressed before proceeding with the normal flow of events for updating a calendar date.

59. Special Requirements

59.1 < Admin role and credentials must be created and stored in the system or permission assigned>

In order for the system to recognize and assign admin role credentials to a user or the user could be permitted to update a calendar date. It must be setup first and recorded in the system. Then, a particular user would be recognized as admin as they are authenticated by the system and would therefore have all editable features activated as soon as they log in.

60. Pre-conditions

60.1 < User must have admin role or permission to update calendar date >

The user must have permission to make updates to the calendar as admin or with access.

60.2 < Calendar Date already exist>

In order for admin or user with permission to update a calendar date, the date must be previously created and is available for updates.

61. Post-conditions

61.1 < Calendar Date is updated >

After updating a calendar date, changes made should be reflected in the appropriate details of the date.

62. Extension Points

62.1 < Additional updates to calendar date>

More information could definitely be added the calendar date in order to reflect more accurate information, similarly information previously added during the update could be deleted from the details.

Elaboration

Use Case Specification: <CreateCalendarDate>

Version <1.0>

Revision History

Date	Version	Description	Author
<03/19/2016>	<1.0>	CreateCalendarDate	<Hermilo Acuna>

Table of Contents

1. CreateCalendarDate **55**

1.1 Brief Description 55

2. Flow of Events **55**

2.1 Basic Flow 55

. The actor will submit the request 56

2.2 Alternative Flows 56

2.2.1 < Missing data > 56

2.2.2 < Corresponding Event > 56

3. Special Requirements **56**

3.1 < Event Date Verification > 56

4. Pre-conditions **56**

4.1 < Authentication Requirement > 56

5. Post-conditions **56**

5.1 < Calendar Date Created > 56

5.2 < Cloud Storage Date > 56

6. Extension Points **57**

6.1 <Cancel> 57

Use Case Specification: <CreateCalendarDate>

63. CreateCalendarDate

63.1 Brief Description

Once the actor has been verified, they will create a new date on the calendar for whatever occasion the actor wishes to announce. The actor can create a date at any point in time including previous dates if they wish to maintain for their users. Depending on the selection on the type of calendar date it will sync to the appropriate data.

64. Flow of Events

64.1 Basic Flow

- The Actor request the Create Calendar Date edit page.
- The system verifies the actor's system access
- The page is now unlocked for the actor to create a new calendar date
- The actor will select one or more of the following option to be created
    e. Title
    f. Description
    g. Date
    h. Links

- The actor will submit the request
- The system will create a new calendar date based on the actor's option
- The system will upload the calendar date to the cloud
- The system will check if there is a corresponding event
- The system will sync both dates if an event exist

64.2 Alternative Flows

64.2.1 < Missing data >

- The actor fails to provide information required to generate the calendar date.
- The actor views a prompt indicating what data is missing.

64.2.2 < Corresponding Event >

The system will sync the calendar date with its corresponding event if it exists

65. Special Requirements

65.1 < Event Date Verification >

If a calendar date is created the system will need to verify if there is already an event with a matching title and corresponding date.

66. Pre-conditions

66.1 < Authentication Requirement >

The actor must meet the following two conditions to use this feature.

3) The actor must be logged in through the system.
4) The actor must have authorization to access the feature.

67. Post-conditions

67.1 < Calendar Date Created >

The calendar has now been updated with a new date that the actor has created. Users will now be able to navigate through the calendar and see the dates that the organization has set up.

67.2 < Cloud Storage Date >

The system will upload the calendar date to the cloud in order to save the date.

68. Extension Points

68.1 **<Cancel>**

The actor at any point during the create a calendar date page can cancel the process. This will delete any current data that the actor has inputted up to this point and resume o the system as is before the actor began to create a calendar date.

Elaboration

Use Case Specification: <DeleteCalendarDate>

Version <1.0>

Revision History

Date	Version	Description	Author
<03/18/2016>	<1.0>	DeleteCalendarDate	<William Benedict>

Table of Contents

1. DeleteCalendarDate **60**

1.1 Brief Description 60

2. Flow of Events **60**

2.1 Basic Flow 60

2.2 Alternative Flows 60

2.2.1 < Lack of permission to delete flow > 60

3. Special Requirements **60**

3.1 < Event Integration> 60

4. Pre-conditions **60**

4.1 <Login & Authentication Requirement> 60

4.2 <Calendar Date Existence Requirement> 60

5. Post-conditions **61**

5.1 < CalendarDate State > 61

6. Extension Points **61**

6.1 <Cancel > 61

Use Case Specification: <DeleteCalendarDate>

69. DeleteCalendarDate

69.1 Brief Description

The DeleteCalendarDate use case illustrates functionality regarding the removal of events from calendar dates. Only users with access to the event management mechanics will have the ability to delete events from the calendar.

70. Flow of Events

70.1 Basic Flow

- The actor selects the event from the calendar.
- The actor selects delete
- The actor submits deletion request

70.2 Alternative Flows

70.2.1 < Lack of permission to delete flow >

- The actor is displayed a message indicating they do not possess the specific rights to delete an event
- The actor is directed back to the calendar

71. Special Requirements

71.1 < Event Integration>

For proper usage, the system requires event functionality.

72. Pre-conditions

72.1 <Login & Authentication Requirement>

The actor must be logged in, and have a level of permission sufficient to both create and delete events

72.2 <Calendar Date Existence Requirement>

There must be an event linked to a calendar date to delete.

73. Post-conditions

73.1 **< CalendarDate State >**

The event linked to the calendar date can either be deleted or not deleted.

74. Extension Points

74.1 **<Cancel >**

At any point, the actor can cancel deletion until the report is actually deleted

Elaboration

Use Case Specification: <ReadCalendarDate>

Version <1.0>

Revision History

Date	Version	Description	Author
<03/18/2016>	<1.0>	ReadCalendarDate	<James Ryg>

Table of Contents

1. ReadCalendarDate **210**

1.1 Brief Description 47

2. Flow of Events **34**

2.1 Basic Flow 34

2.2 Alternative Flows 34

2.2.1 < Calendar Date does not exist > 34

3. Special Requirements **34**

3.1 <Calendar integration requirement > 47

4. Pre-conditions **35**

4.1 < Login & Authentication requirement> 35

5. Extension Points **48**

6.1 <Cancel> 48

Use Case Specification: <ReadCalendarDate>

75. ReadCalendarDate

75.1 Brief Description

This is a small use case, it does not involve editing, destruction, or creation of data objects. It simply conveys attributes of specific calendar dates. Different dates may have different access requirements. There is no post-condition due to it not modifying any data states.

76. Flow of Events

76.1 Basic Flow

- An actor or object requests date information.
- The actors credentials are checked.
    - If the actor does not have the requisite credentials, do not display the information
- The calendar sends the object or actor information about a specific date or set of dates.

76.2 Alternative Flows

76.2.1 < Calendar Date does not exist >

- If no accessible calendar dates exist, display the calendar as blank.

77. Special Requirements

77.1 < Calendar integration Requirement >

Depending on how Adelante wants to handle dates and times, differing calendar integration services may be used. In that case, it needs to be integrated with the system

78. Pre-conditions

78.1 < Login & Authentication Requirement >

For *certain* conditions, there may be a login requirement or permissions requirement to view dates. This precondition is only valid in those specific cases.

79. Extension Points

79.1 **<Cancel>**

This is only called when there is no valid, or accessible data. There would be no other reason for an actor to simply cancel reading a date.

Elaboration

Use Case Specification: <UpdateCalendarDate>

Version 1.0

Revision History

Date	Version	Description	Author
02/27/2016	1.0	UpdateCalendarDate	Maam Awa Ndaw

Table of Contents

1. UpdateCalendarDate **68**

1.1 Brief Description 68

2. Flow of Events **68**

2.1 Basic Flow 68

2.2 Alternative Flows 68

2.2.1 < Unresponsiveness of Interactive Calendar> 68

2.2.2 < System fail to authenticate Admin credentials> 68

3. Special Requirements **69**

3.1< Admin role and credentials must be created and stored in the system or permission assigned> 69

4. Pre-conditions **69**

4.1 < User must have admin role or permission to update calendar date > 69

4.2 < Calendar Date already exist> 69

5. Post-conditions **69**

5.1 < Calendar Date is updated > 69

6. Extension Points **69**

6.1 < Additional updates to calendar date> 69

Use Case Specification: <UpdateCalendarDate>

80. UpdateCalendarDate

80.1 Brief Description

This use case describes how admin or a staff member can update a date in the interactive calendar, given that the calendar date already exists. This primarily allows someone with admin permission or access to editable features of the calendar, to log in the system and makes changes such as date, time and/ or any reminders, details, etc.… associated with it.

81. Flow of Events

81.1 Basic Flow

The basic flow of events for updating a calendar date is as follow:

- Admin or staff member enters username and password to log in system
- System authenticates admin or specific user
- System or staff member makes all editable features in the system available
- Admin or staff member navigates to interactive calendar
- Admin or staff member searches for a specific date in calendar
- Admin or staff member edits the date and then makes changes such as location, time, description, or parties associated with calendar date

81.2 Alternative Flows

81.2.1 < Unresponsiveness of Interactive Calendar>

There could be a possibility that Admin or staff member logs in the system but the interactive Calendar is not responding to user input and displays the same information after update is made by Admin. This could be due to a bug inside the system. In that case, Admin will have to get the issue fixed before proceeding with the normal flow of updating the calendar date.

81.2.2 < System fail to authenticate Admin credentials>

There could be a scenario where a system error causes the admin to be restricted access whenever s/he enters a username and password to login. It could possibly due to a technical issue that needs to be addressed before proceeding with the normal flow of events for updating a calendar date.

82. Special Requirements

82.1 < Admin role and credentials must be created and stored in the system or permission assigned>

In order for the system to recognize and assign admin role credentials to a user or the user could be permitted to update a calendar date. It must be setup first and recorded in the system. Then, a particular user would be recognized as admin as they are authenticated by the system and would therefore have all editable features activated as soon as they log in.

83. Pre-conditions

83.1 < User must have admin role or permission to update calendar date >

The user must have permission to make updates to the calendar as admin or with access.

83.2 < Calendar Date already exist>

In order for admin or user with permission to update a calendar date, the date must be previously created and is available for updates.

84. Post-conditions

84.1 < Calendar Date is updated >

After updating a calendar date, changes made should be reflected in the appropriate details of the date.

85. Extension Points

85.1 < Additional updates to calendar date>

More information could definitely be added the calendar date in order to reflect more accurate information, similarly information previously added during the update could be deleted from the details.

Elaboration

Use Case Specification: <CreateDonateLink>

Version <1.0>

Revision History

Date	Version	Description	Author
<2/29/2016>	<1.0>	CreateDonateLink	<James Ryg>

Table of Contents

1. CreateDonateLink **91**

1.1 Brief Description 91

2. Flow of Events **91**

2.1 Basic Flow 91

2.2 Alternative Flows 91

2.2.1 < No data entered alternative flow > 91

2.2.2 < Bad link alternative flow> 91

3. Special Requirements **91**

3.1 < Donation Link Existence Requirement> 91

4. Pre-conditions **92**

4.1 < Log in and Authentication Requirement > 92

5. Post-conditions **92**

5.1 < Donation Link Data State > 92

6. Extension Points **92**

6.1 <Cancel> 92

Use Case Specification: <CreateDonateLink>

86. CreateDonateLink

86.1 Brief Description

This case allows Adelante employees who have a high enough level of access to change the target link for the donation page. It is exclusive to the webmaster and executive tiers of access, and will also update associated tracking information. Multiple links can be created for differing embeds, such as if donations are being accepted for a specific purpose.

87. Flow of Events

87.1 Basic Flow

- The Actor initiates the CreateDonateLink use case.
- The Actor enters the new donate link they created through their payment processor.
- The Actor concludes creating the donate link.

87.2 Alternative Flows

87.2.1 < No data entered alternative flow >

- The Actor views a message that indicates that no data has been entered in the donate link.

87.2.2 < Bad link alternative flow>

- The Actor views a message that their new donate link returns an error.
- The Actor is prompted to enter a new donate link.

88. Special Requirements

88.1 < Donation Link Existence Requirement>

There must be a donation link created through PayPal or another service for this case to have any usable functionality.

89. Pre-conditions

89.1 **< Log in and Authentication Requirement >**

A user must meet the following conditions to use this feature.

- The user must actually be logged in to create a donate link.
- The user must also have sufficient permissions to access this resource.

90. Post-conditions

90.1 **< Donation Link Data State >**

After this case, a new donation link will have been created for use.

91. Extension Points

91.1 **<Cancel>**

At any point before the final submission, a user can cancel creation.

Elaboration

Use Case Specification: <DeleteDonationLink>

Version <1.0>

Revision History

Date	Version	Description	Author
03/19/2016	1.0	DeleteDonationLink	Chad Tracy

Table of Contents

1. DeleteDonationLink **95**

1.1 Brief Description 95

2. Flow of Events **95**

2.1 Basic Flow 95

2.2 Alternative Flows 95

2.2.1 < Actor Not Authorized > 95

3. Pre-conditions **95**

3.1 < Privileges Must Be Granted to Users > 95

3.2 < Donation Link Must Exits > 95

4. Post-conditions **95**

4.1 < Donation Link is Deleted > 95

5. Extension Points **96**

5.1 <Cancel> 96

Use Case Specification: <DeleteDonationLink>

92. DeleteDonationLink

92.1 Brief Description

Allows those with permission within Adelante's system to delete the link associated with a specified donation page. This power is only for higher Adelante employees and the webmaster.

93. Flow of Events

93.1 Basic Flow

- Actor accesses donation pages with admin privileges
- Actor requests donation link is removed
- System approves request
- Donation link is deleted from page

93.2 Alternative Flows

93.2.1 < Actor Not Authorized >

- Actor attempts to log in to Adelante system
- Actor is either rejected or not given the necessary privileges

94. Pre-conditions

94.1 **< Privileges Must Be Granted to Users >**

The system must recognize the correct individuals as admins and grant privileges accordingly

94.2 **< Donation Link Must Exits >**

In order for the donation link to be deleted, it must exist and be associated with a donation page

95. Post-conditions

95.1 **< Donation Link is Deleted >**

After the process is complete, the link is deleted from the donation page

96. Extension Points

96.1 **<Cancel>**

At any point before the link is deleted, the user should be able to cancel the process

Elaboration

Use Case Specification: <ReadDonateLink>

Version 1.0

Revision History

Date	Version	Description	Author
02/29/2016	1.0	ReadDonateLink	Maam Awa Ndaw

Table of Contents

1. ReadDonateLink **99**

1.1 Brief Description 99

2. Flow of Events **99**

2.1 Basic Flow 99

2.2 Alternative Flows 99

2.2.1 < Potential donor could receive an email with donation link attached > 99

2.2.2 < Potential donor could be given direct access to donation page via email > 99

3. Special Requirements **99**

3.1 < Donate link must be available for viewing and in stand by mode > 99

4. Pre-conditions **100**

4.1 < Donate link previously created > 100

4.2 < Donate link available for reading > 100

5. Post-conditions **100**

5.1 < Donate link viewed > 100

6. Extension Points **100**

6.1 < Donate link accessed > 100

Use Case Specification: <ReadDonateLink>

97. ReadDonateLink

97.1 Brief Description

This use case describes how a donation link can be read or viewed by a potential donor.

98. Flow of Events

98.1 Basic Flow

The basic flow of events for reading a donation link is as follow:

- Potential donor directly navigates to system or through URL link
- Potential donor navigates to donation page
- Potential donor views donation link

98.2 Alternative Flows

98.2.1 < Potential donor could receive an email with donation link attached >

In some cases, a potential donor could be given ease of access to donation link via email. In that case, potential donor accesses email, opens email and reads donation link.

98.2.2 < Potential donor could be given direct access to donation page via email >

There could be a scenario where the potential donor receives an email with a link to donation page. Once accessed, the link would lead directly to the donation page, and there, the potential donor can read or view the donation link.

99. Special Requirements

99.1 < Donate link must be available for viewing and in stand by mode >

For a potential donor to read a donation link, the link would have to present and available for reading or viewing in or out of the system.

100. Pre-conditions

100.1 **< Donate link previously created >**

For a potential donor to be able to read a donate link, the link must have been previously created.

100.2 **< Donate link available for reading >**

Donate link must be available for viewing and in stand by mode.

101. Post-conditions

101.1 **< Donate link viewed >**

Potential donor successfully attempts to read a donate link.

102. Extension Points

102.1 **< Donate link accessed >**

After reading or viewing a donate link, a potential donor may decide to access the link.

<Elaboration >

Use Case Specification: <UpdateDonateLink >

Version <1.0>

Revision History

Date	Version	Description	Author
03/18/2016	1.0	UpdateDonateLink	Valencia Richard

Table of Contents

1. UpdateDonationLink **104**

1.1. Brief Description *104*

2. Flow of Events **104**

2.1. Basic Flow *104*

2.2. Alternative Flows **104**

2.2.1. < Donation Link Not Created > *104*

2.2.2. < Update Not Approved> *105*

3. Special Requirements **105**

3.1. < Donation Link Linked to Donation Database> *105*

4. Pre-conditions **105**

4.1. < Link Must Exist> *105*

4.2. < Login Credentials Verified> *105*

5. Post-conditions **105**

5.1. < Donation Link Updated > *105*

6. Extension Points **106**

6.1. ***< Cancel>***

106

Use Case Specification: <UpdateDonateLink>

1. UpdateDonationLink

1.1. Brief Description

This use case allows an actor with the appripriate permnissionst to go in and update the Donation Link that serves as the intermedry for accepting donations. The linlk m. The link may need to be moved to a different Donation platform and may need to be updated

2. Flow of Events

2.1. Basic Flow

- The Actor initiates the Use Case by accessing the AHA system
- The actor inputs their login information
- The system accepts the information
- The actor locates goes into the backend to the line of code representing the Donation Link
- The actor updates code representing the Link
- The actor attempts the save the changes
- The system accesses the credentials for proper clearance
- Clearance level is high enough to make Updates
- The system updates the Volunteer Link
- The actor goes into the system to link the new Link to Donation database

2.2. Alternative Flows

2.2.1. < Donation Link Not Created >

- The Actor initiates the Use Case by accessing the AHA system
- The actor inputs their login information
- The system accepts the information
- The actors go into the system to locate the code for Link, but it does not exist

2.2.2. < Update Not Approved>

- The Actor initiates the Use Case by accessing the AHA system
- The Actor initiates the Use Case by accessing the AHA system
- The actor inputs their login information
- The system accepts the information
- The actor locates goes into the backend to the line of code representing the Donation Link
- The actor updates code representing the Link
- The actor attempts the save the changes
- The system accesses the credentials for proper clearance
- Clearance level is not high enough to make Updates
- The system alerts the User that changes have not been made

3. Special Requirements

3.1. < **Donation Link Linked to Donation Database>**

After the main flow has been created, the actor must ensure that the database link is connected to the Donation database. If this is not ensured, then donations will go in limbo instead of to Adalante

4. Pre-conditions

4.1. < **Link Must Exist>**

The Link must be coded beforehand in order to make changes and updates to it

4.2. < **Login Credentials Verified>**

An actor must have the proper login and password to access the system to make updates to go in and change the code and update the Link

5. Post-conditions

5.1. < **Donation Link Updated >**

The changes that the actor wants to be made will be reflected in the Updated Donation Link

6. Extension Points

6.1. < **Cancel**>

An actor can cancel the attempt to Update the Donation Link

<Elaboration >

Use Case Specification: <CreateDonationTracker>

Version <1.0>

Revision History

Date	Version	Description	Author
03/18/2016	1.0	CreateDonationTracker	Valencia Richard

Table of Contents

1. CreateDonationTracker **110**

1.1 Brief Description *110*

2. Flow of Events **110**

2.1 Basic Flow *110*

2.2 Alternative Flows **110**

2.2.1 *< System Error in Creation >* *110*

2.2.2 *< Login Fail>* *110*

3. Special Requirements **110**

3.1 *< Donation Information Integration >* *110*

4. Pre-conditions **111**

4.1 < System Capability> *111*

4.2 < Login Credentials Verified> *111*

5. Post-conditions **111**

5.1 < Donation Tracker Created> *111*

5.2 < Donation Information Integration> *111*

Use Case Specification: <CreateDonationTracker>

103. 1. CreateDonationTracker

103.1 1.1 Brief Description

This use case allows an actor to create application that references the Donation object. The Tracker must be coded and maintain a system of domation amount and pertinent infotmation inclidong donator, donation amount, donation date, and how the donation was used.

104. 2. Flow of Events

104.1 2.1 Basic Flow

- The Actor initiates the Use Case by entering credentials in the AHA system
- The system accepts the credentials
- The actor creates object in the System to track Volunteers
- The system accepts the new object

105. 2.2 Alternative Flows

105.1 2.2.1 < System Error in Creation >

In order to create the Tracker object, the system must have the capability.

105.2 *2.2.2 < Login Fail>*

In order to access the Tracker tool you need to have the proper authorization. A situation may arise in which someone may attempt to log into the system but they are not authorized. The system will reject their login.

106. 3. Special Requirements

106.1 3.1 < Donation Information Integration >

Donation information must be available to the new Object. The donation information must now pass through the object to make full use of the Object

107. 4. Pre-conditions

107.1 4.1 < System Capability>

In order to create the Tracker object the system must have the capability to create the object

107.2 4.2 < Login Credentials Verified>

An actor must have the proper login and password to access the system to create the object

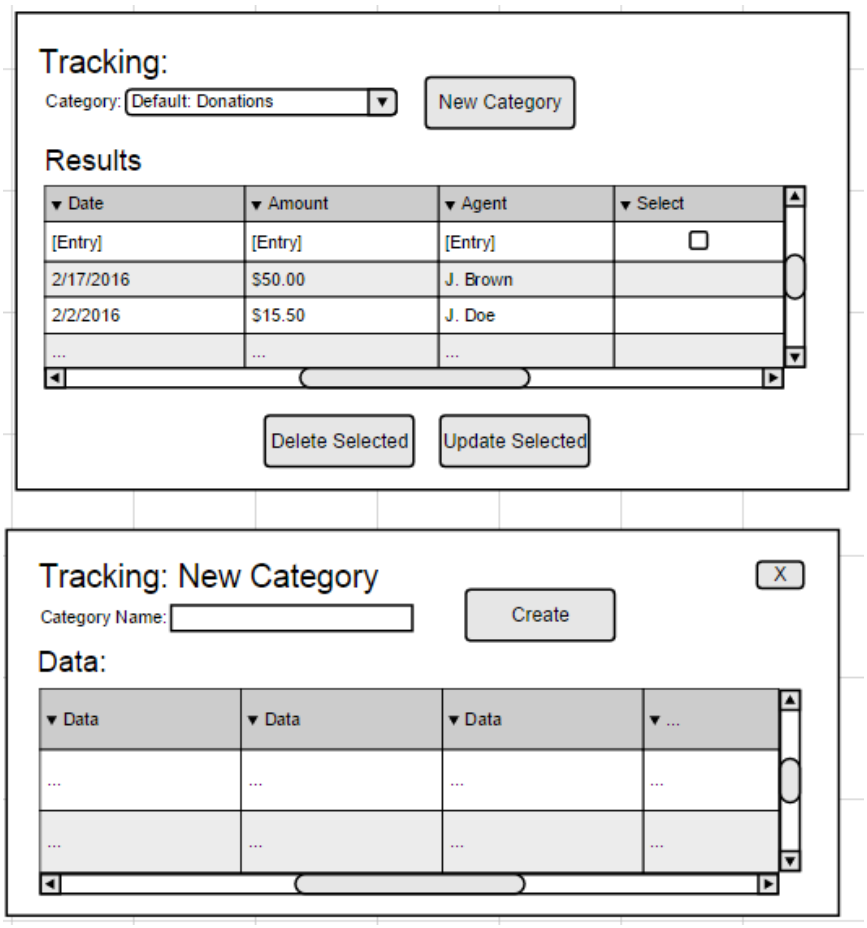
108. 5. Post-conditions

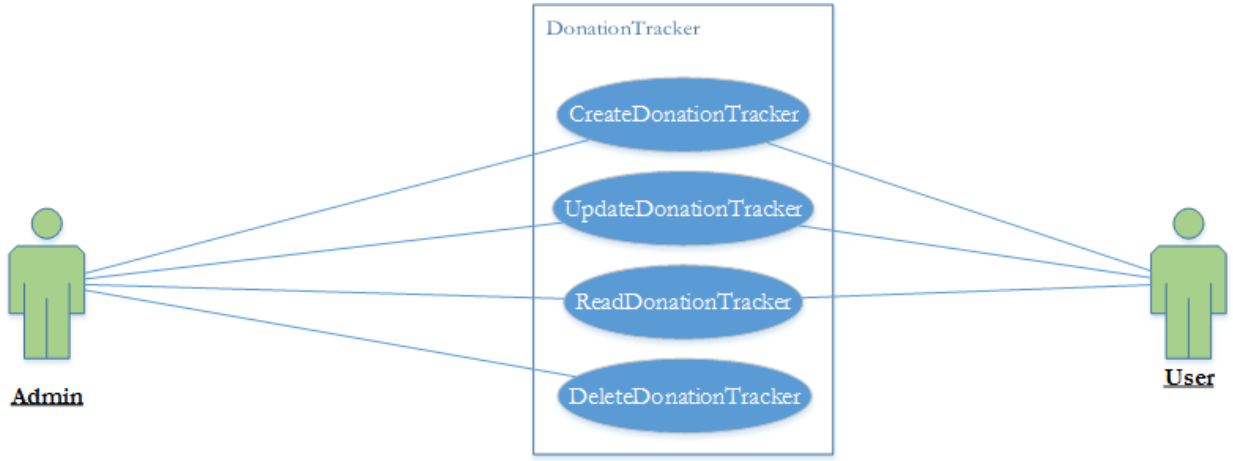
108.1 5.1 < Donation Tracker Created>

The Donation tracker is created and fully functional

108.2 5.2 < Donation Information Integration>

Donation information must be integrated into the system


Tracking:
Category: Default: Donations
New Category
Results
Date
Amount
Agent
Select
[Entry]
[Entry]
[Entry]
2/17/2016
$50.00
J. Brown
2/2/2016
$15.50
J. Doe
Delete Selected
Update Selected
Tracking: New Category
X
Category Name:
Create
Data:
Data
Data
Data
...


DonationTracker
CreateDonationTracker
UpdateDonationTracker
ReadDonationTracker
DeleteDonationTracker
Admin
User

Elaboration

Use Case Specification: <DeleteDonationTracker>

Version <1.0>

Revision History

Date	Version	Description	Author
03/19/2016	<1.0>	DeleteDonationTracker	<Hermilo Acuna>

Table of Contents

1. DeleteDonationTracker **118**

1.1 Brief Description 118

2. Flow of Events **118**

2.1 Basic Flow 118

2.2 Alternative Flows 118

2.2.1 < Lack of Authentication > 118

3. Special Requirements **118**

3.1 < N/A > 118

4. Pre-conditions **118**

4.1 < Authentication Requirement > 118

4.2 < Donation Tracker Existence > 118

5. Post-conditions **119**

5.1 < Donation Tracker Deleted > 119

6. Extension Points **119**

6.1 <Cancel> 119

Use Case Specification: <DeleteDonationTracker>

109. DeleteDonationTracker

109.1 Brief Description

This use case is used to maintain organization and to delete any excess Donation Tracker use cases or outdate ones. The actors with the right access will only be able to utilize this use case as it is a high level task.

110. Flow of Events

110.1 Basic Flow

- The actor is on the donation tracker page
- The actor request delete donation tracker
- The system authenticates the actor
- The system processes the DeleteDonationTracker use case

110.2 Alternative Flows

110.2.1 < Lack of Authentication >

- The user does not have the level of access to utilize this use case and the system rejects the request

111. Special Requirements

111.1 < N/A >

N/A

112. Pre-conditions

112.1 < Authentication Requirement >

The actor must meet the following two conditions to use this feature.

- The actor must be logged in through the system.
- The actor must have authorization to access the feature.

112.2 < Donation Tracker Existence >

The donation tracker must exist in order for the actor to delete the tracker

113. Post-conditions

113.1 **< Donation Tracker Deleted >**

The Donation Tracker is deleted from the system and everything that it contained.

114. Extension Points

114.1 **<Cancel>**

The actor may cancel the delete use case so long as it is applied before the actor clicks the DeletDonationTracker use case.

Elaboration

Use Case Specification: <ReadDonationTracker>

Version <1.0>

Revision History

Date	Version	Description	Author
03/19/2016	1.0	ReadDonationTracker	Chad Tracy

Table of Contents

1. ReadDonationTracker **122**

1.1 Brief Description 122

2. Flow of Events **122**

2.1 Basic Flow 122

2.2 Alternative Flows 122

2.2.1 < Credential Error> 122

3. Pre-conditions **122**

3.1 < Donation Tracker Exists > 122

3.2 < Special Privileges > 122

4. Post-conditions **123**

4.1 < Actor Viewed Donations > 123

5. Extension Points **123**

5.1 <Cancel> 123

Use Case Specification: <ReadDonationTracker>

115. ReadDonationTracker

115.1 **Brief Description**

Allows the actor to view the information on donations via the donation tracker. Allows the actor to view the amount donated, the donator, the date it was donated, and how it was used. Only the donator and AHA employees should have this power.

116. Flow of Events

116.1 **Basic Flow**

- Actor logs-in to AHA system
- System approves actor
- Actor navigates to donation tracker
- Actor can view donations and the information associated with it

116.2 **Alternative Flows**

116.2.1 < Credential Error>

- Actor logs in to AHA system
- Actor is denied access and returned to log-in screen

116.2.1.1 < Lower Permissions >

- Actor logs-in to AHA system
- Actor is granted access, but does not have permission to view donation tracker

117. Pre-conditions

117.1 **< Donation Tracker Exists >**

In order to view the donation tracker, it must exist within the system. It must also be up to date with current information

117.2 **< Special Privileges >**

It must be determined who can view the tracker information from the system before it can be viewed to ensure only the right people can see it.

118. Post-conditions

118.1 **< Actor Viewed Donations >**

After the case is complete, the actor will have viewed the donation and the information associated with it

119. Extension Points

119.1 **<Cancel>**

At any point the actor should be able to cancel the process and exit the system

<CIS320>

Use Case Specification: <UpdateDonationTracker>

Version <1.0>

Revision History

Date	Version	Description	Author
<03/05/2016>	<1.0>	UpdateDonationTracker	<James Ryg>

Table of Contents

1. UpdateDonationTracker 210

1.1 Brief Description 47

2. Flow of Events 34

2.1 Basic Flow 34

2.2 Alternative Flows 34

2.2.1 < Empty Text Exception Flow > 34

3. Special Requirements 34

3.1 < External Donation Tracker Requirement > 47

4. Pre-conditions 35

4.1 < Login & Authentication Requirement > 35

4.2 <Prior Tracker Existence Requirement >

2

5. Post-conditions 35

5.1 < Tracker State > 47

6. Extension Points 48

6.1 <Cancel> 48

Use Case Specification: <UpdateDonationTracker>

120. UpdateDonationTracker

120.1 Brief Description

The UpdateDonationTracker use case illustrates how an actor can change the behavior of the systems donation trackers. Only users who have the ability to edit the site directly, likely the webmaster and high level adelante staff, will have the ability to update donation trackers.

121. Flow of Events

121.1 Basic Flow

- The actor navigates to the donation tracker.
- The actor selects a donation tracker.
- The actor can change the tracker's attributes (there likely may be more than this).
    - Name
    - Target
    - Location
- The actor submits the changes.

121.2 Alternative Flows

121.2.1 < Empty Text Exception flow >

- The actor enters bad, or no data in a required field.
- The actor is prompted with a message indicating the bad field.
- The actor enters the new updated field.

122. Special Requirements

122.1 < External donation tracker requirement >

Adelantes' payment processor must have provided the ability to pull records.

123. Pre-conditions

123.1 **< Login & Authentication Requirement>**

The actor must have access rights to the donation tracker management functionality.

123.2 **< Prior tracker existence requirement>**

A tracker must have been created before you can update it.

124. Post-conditions

124.1 **< Tracker state >**

The tracker assumes the new submitted state.

125. Extension Points

125.1 **<Cancel>**

The actor can cancel prospective changes until they submit.

Elaboration

Use Case Specification: <CreateDonor>

Version <1.0>

Revision History

Date	Version	Description	Author
03/19/2016	<1.0>	CreateDonor	<Hermilo Acuna>

Table of Contents

1. CreateDonor **133**

1.1 Brief Description 133

2. Flow of Events **133**

2.1 Basic Flow 133

2.2 Alternative Flows 133

2.2.1 < Missing data > 133

3. Special Requirements **133**

3.1 < Donation Account > 133

4. Pre-conditions **133**

4.1 < Authentication Requirement > 133

4.2 < User Donate > 133

5. Post-conditions **134**

5.1 < Donor created > 134

6. Extension Points **134**

6.1 <Cancel> 134

Use Case Specification: <CreateDonor>

126. CreateDonor

126.1 Brief Description

This use case allows actors that are Adelante employees or administrators to create accounts for those that have donated to the nonprofit organization. The only way for a user to obtain a donor account would mea that they would have to donate and have the higher level actors confirm.

127. Flow of Events

127.1 Basic Flow

- User donates to the nonprofit through the website
- The actors confirm that this a verified user
- The actor requests a new donor account
- The system verifies credentials
- The system creates a new donor account
- The information to access account is sent to the donor

127.2 Alternative Flows

127.2.1 < Missing data >

- The actor fails to provide information required to generate the donor account.
- The actor views a prompt indicating what data is missing.

128. Special Requirements

128.1 < Donation Account >

The user must have donated to Adelante in order to proceed.

129. Pre-conditions

129.1 < Authentication Requirement >

The actor must meet the following two conditions to use this feature.

- The actor must be logged in through the system.
- The actor must have authorization to access the feature.

129.2 < User Donate >

The user has already donated

130. Post-conditions

130.1 **< Donor created >**

The system now has a new donor in the system that tracks and manages the information going though that account.

131. Extension Points

131.1 **<Cancel>**

The actor at any point during the creation of the donor can exit and all the data being inputted will be erased and the system will return to the original point before the actor clicked on the CreateDonor use case.

Elaboration

Use Case Specification: <DeleteDonor>

Version <1.0>

Revision History

Date	Version	Description	Author
03/19/2016	<1.0>	DeleteDonor	<Hermilo Acuna>

Table of Contents

1. DeleteDonor **139**

1.1 Brief Description 139

2. Flow of Events **139**

2.1 Basic Flow 139

2.2 Alternative Flows 139

2.2.1 < Lack of Authentication > 139

3. Special Requirements **139**

3.1 < N/A > 139

4. Pre-conditions **139**

4.1 < Authentication Requirement > 139

4.2 < Donor Existence > 139

5. Post-conditions **140**

5.1 < Donor Deleted > 140

6. Extension Points **140**

6.1 <Cancel> 140

Use Case Specification: <DeleteDonor>

132. DeleteDonor

132.1 Brief Description

This use case is to delete any donor on the system that is determined by the actor to be no longer active or it is a repetitive account. This allows the system to be maintained and organized and remove the need for unnecessary excess data. The only actors with this access are the Adelante Employees and Administrators

133. Flow of Events

133.1 Basic Flow

- The actor is on the donor page
- The actor request to delete donor
- The system authenticates the actor
- The system processes the DeleteDonor use case

133.2 Alternative Flows

133.2.1 < Lack of Authentication >

- The actor does not have access to be able to delete the donor.

134. Special Requirements

134.1 < N/A >

N/A

135. Pre-conditions

135.1 < Authentication Requirement >

The actor must meet the following two conditions to use this feature.

- The actor must be logged in through the system.
- The actor must have authorization to access the feature.

135.2 < Donor Existence >

The Donor must exist in order for this use case to be utilized. Without a donor to be processed the delete donor use case cannot function

136. Post-conditions

136.1 **< Donor Deleted >**

The donor is now deleted from the system by the actor.

137. Extension Points

137.1 **<Cancel>**

The actor my cancel their action up until the request to the system to delete the donor is used. The actor my go through the all but the last option and still be able to cancel the use case event

Elaboration

Use Case Specification: <ReadDonor>

Version <1.0>

Revision History

Date	Version	Description	Author
<03/18/2016>	<1.0>	ReadDonor	<James Ryg>

Table of Contents

1. ReadDonor **143**

1.1 Brief Description 143

2. Flow of Events **143**

2.1 Basic Flow 143

2.2 Alternative Flows 143

2.2.1 < No Donor Information > 143

3. Special Requirements **143**

3.1 < External Donor Payment Processor> 143

4. Pre-conditions **144**

4.1 < Login & Authentication Requirement > 144

5. Extension Points **144**

5.1 <Cancel> 144

Use Case Specification: <ReadDonor>

138. ReadDonor

138.1 Brief Description

Another read only case, this case specifies how donor information can be accessed by an actor or object. In this case, Adelante would want to restrict access to financial officers or certain employees of their organization for security reasons, as it deals with private donor information.

139. Flow of Events

139.1 Basic Flow

- An actor or object requests donor information.
- If the credentials are sufficient, the system requests the donor data from the database
- The donor database sends the data back.
- The recipient actor or object receives the data.

139.2 Alternative Flows

139.2.1 < No Donor Information >

- If there is no donor information available, the object or actor is returned a message stating so.
- If the actor has the ability to edit donors, they are given the option to redirect to the donor page to troubleshoot.

140. Special Requirements

140.1 < External Donor Payment Processor>

Adelante uses PayPal. Without a functional PayPal link, there is no way to update donors.

141. Pre-conditions

141.1 **< Login & Authentication Requirement >**

To view personal donor information, only a member of the organization who is given the access roles will be able to use this, however metadata (such as total donations) can be used elsewhere without restriction.

142. Extension Points

142.1 **<Cancel>**

In the event an exception occurs, it will cancel itself if the issue is not resolved. On manual donor view, an actor could theoretically cancel viewing and close.

Elaboration

Use Case Specification: <UpdateDonor>

Version <1.0>

Revision History

Date	Version	Description	Author
<03/18/2016 >	<1.0>	UpdateDonor	<William Benedict>

Table of Contents

1. UpdateDonor **147**

1.1 Brief Description 147

2. Flow of Events **147**

2.1 Basic Flow 147

2.2 Alternative Flows 147

2.2.1 < Incomplete information provided> 147

2.2.2 <No change in information> 147

3. Special Requirements **147**

4. Pre-conditions **148**

4.1 <Login & Authentication> 148

4.2 < Donor Existence > 148

5. Post-conditions **148**

5.1 <Change of State> 148

6. Extension Points **148**

6.1 <Cancel> 148

Use Case Specification: <UpdateDonor>

143. UpdateDonor

143.1 Brief Description

This use case illustrates how an actor will Update the information of a donor that has already been created. These information updates are key to maintaining points of contact with donors when they move or change numbers

144. Flow of Events

144.1 Basic Flow

- The actor selects a donor from the list of donors.
- The actor selects the update donor option for the selected donor.
- The actor may update one or more fields with new information.
    - The actor may update the address of the donor
    - The actor may update the phone number(s) of the donor.
    - The actor may update the payment information of the donor (if provided)
    - The actor may update the status of the donor (active, recurrent, inactive, no contact
- The actor submits the updated information

144.2 Alternative Flows

144.2.1 < Incomplete information provided>

- The actor does not provide a complete: Address, Phone number, etc.
- The actor is prompted by a message that the data they have input is incomplete
- The actor is navigated back to the update form to complete the information.

144.2.2 <No change in information>

- The actor makes no visible changes to the date
- The actor is prompted by a message stating that they information provided is exactly the same
- The actor is navigated back to the update form.

145. Special Requirements

146. Pre-conditions

146.1 **<Login & Authentication>**

The actor must be signed in and have permission levels to see and update donors.

146.2 **< Donor Existence >**

The donor must have previously been created in order to be updated.

147. Post-conditions

147.1 **<Change of State>**

The Donor information that is recorded for the donor has been updated.

148. Extension Points

148.1 **<Cancel>**

The actor may end the update process at any time before the information is actually updated.

Elaboration

Use Case Specification: <CreateDonor>

Version <1.0>

Revision History

Date	Version	Description	Author
03/19/2016	<1.0>	CreateDonor	<Hermilo Acuna>

Table of Contents

1. CreateDonor **133**

1.1 Brief Description 133

2. Flow of Events **133**

2.1 Basic Flow 133

2.2 Alternative Flows 133

2.2.1 < Missing data > 133

3. Special Requirements **133**

3.1 < Donation Account > 133

4. Pre-conditions **133**

4.1 < Authentication Requirement > 133

4.2 < User Donate > 133

5. Post-conditions **134**

5.1 < Donor created > 134

6. Extension Points **134**

6.1 <Cancel> 134

Use Case Specification: <CreateDonor>

149. CreateDonor

149.1 Brief Description

This use case allows actors that are Adelante employees or administrators to create accounts for those that have donated to the nonprofit organization. The only way for a user to obtain a donor account would mea that they would have to donate and have the higher level actors confirm.

150. Flow of Events

150.1 Basic Flow

- User donates to the nonprofit through the website
- The actors confirm that this a verified user
- The actor requests a new donor account
- The system verifies credentials
- The system creates a new donor account
- The information to access account is sent to the donor

150.2 Alternative Flows

150.2.1 < Missing data >

- The actor fails to provide information required to generate the donor account.
- The actor views a prompt indicating what data is missing.

151. Special Requirements

151.1 < Donation Account >

The user must have donated to Adelante in order to proceed.

152. Pre-conditions

152.1 < Authentication Requirement >

The actor must meet the following two conditions to use this feature.

- The actor must be logged in through the system.
- The actor must have authorization to access the feature.

152.2 < User Donate >

The user has already donated

153. Post-conditions

153.1 **< Donor created >**

The system now has a new donor in the system that tracks and manages the information going though that account.

154. Extension Points

154.1 **<Cancel>**

The actor at any point during the creation of the donor can exit and all the data being inputted will be erased and the system will return to the original point before the actor clicked on the CreateDonor use case.

Elaboration

Use Case Specification: <DeleteDonor>

Version <1.0>

Revision History

Date	Version	Description	Author
03/19/2016	<1.0>	DeleteDonor	<Hermilo Acuna>

Table of Contents

1. DeleteDonor **139**

1.1 Brief Description 139

2. Flow of Events **139**

2.1 Basic Flow 139

2.2 Alternative Flows 139

2.2.1 < Lack of Authentication > 139

3. Special Requirements **139**

3.1 < N/A > 139

4. Pre-conditions **139**

4.1 < Authentication Requirement > 139

4.2 < Donor Existence > 139

5. Post-conditions **140**

5.1 < Donor Deleted > 140

6. Extension Points **140**

6.1 <Cancel> 140

Use Case Specification: <DeleteDonor>

155. DeleteDonor

155.1 **Brief Description**

This use case is to delete any donor on the system that is determined by the actor to be no longer active or it is a repetitive account. This allows the system to be maintained and organized and remove the need for unnecessary excess data. The only actors with this access are the Adelante Employees and Administrators

156. Flow of Events

156.1 **Basic Flow**

- The actor is on the donor page
- The actor request to delete donor
- The system authenticates the actor
- The system processes the DeleteDonor use case

156.2 **Alternative Flows**

156.2.1 < Lack of Authentication >

- The actor does not have access to be able to delete the donor.

157. Special Requirements

157.1 **< N/A >**

N/A

158. Pre-conditions

158.1 **< Authentication Requirement >**

The actor must meet the following two conditions to use this feature.

- The actor must be logged in through the system.
- The actor must have authorization to access the feature.

158.2 **< Donor Existence >**

The Donor must exist in order for this use case to be utilized. Without a donor to be processed the delete donor use case cannot function

159. Post-conditions

159.1 **< Donor Deleted >**

The donor is now deleted from the system by the actor.

160. Extension Points

160.1 **<Cancel>**

The actor my cancel their action up until the request to the system to delete the donor is used. The actor my go through the all but the last option and still be able to cancel the use case event

Elaboration

Use Case Specification: <ReadDonor>

Version <1.0>

Revision History

Date	Version	Description	Author
<03/18/2016>	<1.0>	ReadDonor	<James Ryg>

Table of Contents

1. ReadDonor **143**

1.1 Brief Description 143

2. Flow of Events **143**

2.1 Basic Flow 143

2.2 Alternative Flows 143

2.2.1 < No Donor Information > 143

3. Special Requirements **143**

3.1 < External Donor Payment Processor> 143

4. Pre-conditions **144**

4.1 < Login & Authentication Requirement > 144

5. Extension Points **144**

5.1 <Cancel> 144

Use Case Specification: <ReadDonor>

161. ReadDonor

161.1 Brief Description

Another read only case, this case specifies how donor information can be accessed by an actor or object. In this case, Adelante would want to restrict access to financial officers or certain employees of their organization for security reasons, as it deals with private donor information.

162. Flow of Events

162.1 Basic Flow

- An actor or object requests donor information.
- If the credentials are sufficient, the system requests the donor data from the database
- The donor database sends the data back.
- The recipient actor or object receives the data.

162.2 Alternative Flows

162.2.1 < No Donor Information >

- If there is no donor information available, the object or actor is returned a message stating so.
- If the actor has the ability to edit donors, they are given the option to redirect to the donor page to troubleshoot.

163. Special Requirements

163.1 < External Donor Payment Processor>

Adelante uses PayPal. Without a functional PayPal link, there is no way to update donors.

164. Pre-conditions

164.1 **< Login & Authentication Requirement >**

To view personal donor information, only a member of the organization who is given the access roles will be able to use this, however metadata (such as total donations) can be used elsewhere without restriction.

165. Extension Points

165.1 **<Cancel>**

In the event an exception occurs, it will cancel itself if the issue is not resolved. On manual donor view, an actor could theoretically cancel viewing and close.

Elaboration

Use Case Specification: <UpdateDonor>

Version <1.0>

Revision History

Date	Version	Description	Author
<03/18/2016 >	<1.0>	UpdateDonor	<William Benedict>

Table of Contents

1. UpdateDonor **147**

1.1 Brief Description 147

2. Flow of Events **147**

2.1 Basic Flow 147

2.2 Alternative Flows 147

2.2.1 < Incomplete information provided> 147

2.2.2 <No change in information> 147

3. Special Requirements **147**

4. Pre-conditions **148**

4.1 <Login & Authentication> 148

4.2 < Donor Existence > 148

5. Post-conditions **148**

5.1 <Change of State> 148

6. Extension Points **148**

6.1 <Cancel> 148

Use Case Specification: <UpdateDonor>

166. UpdateDonor

166.1 Brief Description

This use case illustrates how an actor will Update the information of a donor that has already been created. These information updates are key to maintaining points of contact with donors when they move or change numbers

167. Flow of Events

167.1 Basic Flow

- The actor selects a donor from the list of donors.
- The actor selects the update donor option for the selected donor.
- The actor may update one or more fields with new information.
    - The actor may update the address of the donor
    - The actor may update the phone number(s) of the donor.
    - The actor may update the payment information of the donor (if provided)
    - The actor may update the status of the donor (active, recurrent, inactive, no contact
- The actor submits the updated information

167.2 Alternative Flows

167.2.1 < Incomplete information provided>

- The actor does not provide a complete: Address, Phone number, etc.
- The actor is prompted by a message that the data they have input is incomplete
- The actor is navigated back to the update form to complete the information.

167.2.2 <No change in information>

- The actor makes no visible changes to the date
- The actor is prompted by a message stating that they information provided is exactly the same
- The actor is navigated back to the update form.

168. Special Requirements

169. Pre-conditions

169.1 **<Login & Authentication>**

The actor must be signed in and have permission levels to see and update donors.

169.2 **< Donor Existence >**

The donor must have previously been created in order to be updated.

170. Post-conditions

170.1 **<Change of State>**

The Donor information that is recorded for the donor has been updated.

171. Extension Points

171.1 **<Cancel>**

The actor may end the update process at any time before the information is actually updated.

Elaboration

Use Case Specification: <CreateFile>

Version <1.0>

Revision History

Date	Version	Description	Author
<2/29/2016>	<1.0>	CreateFile	<James Ryg>

Table of Contents

1. CreateFile **173**

1.1 Brief Description 173

2. Flow of Events **173**

2.1 Basic Flow 173

2.2 Alternative Flows 173

2.2.1 < Draft Flow > 173

2.2.2 < Bad File Alternative Flow> 173

2.3 < First Special Requirement > 173

3. Pre-conditions **173**

3.1 < Login & Authentication> 173

4. Post-conditions **173**

4.1 < File State > 173

5. Extension Points **174**

5.1 <Cancel> 174

Use Case Specification: <CreateFile>

172. CreateFile

172.1 **Brief Description**

This case allows Adelante members to create files, with visibility and access limited by their access rights on the system. Various types of files can be created, and Adelante employees would be able to limit access to this tool on an individual and access right basis.

173. Flow of Events

173.1 **Basic Flow**

- The actor begins creating a file.
- The actor either uploads or drafts a file.
- The actor selects visibility of the file (who can access it).
- The actor submits the file for publishing.

173.2 **Alternative Flows**

173.2.1 < Draft Flow >

- The actor is displayed a web form to edit a file.
- The actor selects from site data, or inputs their own data.
- The actor submits the file.

173.2.2 < Bad File Alternative Flow>

- The actor is presented with a message indicating that the file is invalid in some form.
- The actor is prompted to submit a new file

173.3 **< First Special Requirement >**

There must be some system in place for file transfer.

174. Pre-conditions

174.1 **< Login & Authentication>**

The actor must meet the following conditions:

- The actor must be logged in.
- The actor must have access rights sufficient to create files.

175. Post-conditions

175.1 **< File State >**

A file is created and stored.

176. Extension Points

176.1 **<Cancel>**

The actor can cease creating a file before it is submitted.

Elaboration

Use Case Specification: <DeleteFile>

Version 1.0

Revision History

Date	Version	Description	Author
02/20/2016	1.0	DeleteFile	Valencia Richard

Table of Contents

1. DeleteFile **177**

1.1 Brief Description 177

2. Flow of Events **177**

2.1 Basic Flow 177

2.2 Alternative Flows **177**

2.2.1 < Invalid Credentials> 177

3. Pre-conditions **177**

3.1 < File in Existence> 177

3.2 < Login Credentials Verified> 177

4. Post-conditions **178**

4.1 < File Deleted> 178

5. Extension Points **178**

5.1 < Cancel> 178

Use Case Specification: <DeleteFile>

177. DeleteFile

177.1 Brief Description

This use case allows an actor to delete a file in the systen uf it is no longer valid, neccesary, or accurate. Only actors with the appropriate credentials can delete files. Since file deletion is permanet, the system will prompt for verification of the Delete

178. Flow of Events

178.1 Basic Flow

- The Actor initiates the Use Case by entering credentials in the AHA system
- The system accepts the credentials
- The actor locates the file
- The actor deletes the file
- The system recognizes the request for Delete
- The system asks the user to verify the delete
- The system deletes the file

2.2 Alternative Flows

2.2.1 < Invalid Credentials>

- User enters information
- The system receives request to Login
- The system denies the request based on invalid login information

179. Pre-conditions

179.1 < File in Existence>

In order to delete a file, that file must already exist

179.2 < Login Credentials Verified>

An actor must have the proper login and password to access the system to delete a file. Only the top administration has this ability

180. Post-conditions

180.1 < File Deleted>

The file will be deleted from the system

181. Extension Points

181.1 < Cancel>

The actor will be prompted once to verify the desire to delete the File. At this point only, they can cancel the Delete request

Elaboration

Use Case Specification: <DownloadFile>

Version <1.0>

Revision History

Date	Version	Description	Author
<03/19/2016>	<1.0>	DownloadFile	Chad Tracy

Table of Contents

1. DownloadFile **182**

1.1 Brief Description 182

2. Flow of Events **182**

2.1 Basic Flow 182

2.2 Alternative Flows 182

2.2.1 < Unable to Download File > 182

3. Pre-conditions **182**

3.1 < Files Must Exist > 182

3.2 < Files Must Have Different Levels of Permission > 182

4. Post-conditions **183**

4.1 < Actor has Downloaded File > 183

5. Extension Points **183**

5.1 <Cancel> 183

Use Case Specification: <DownloadFile>

182. DownloadFile

182.1 Brief Description

Allows the actor to download files from the Adelante system based on their level of access.

183. Flow of Events

183.1 Basic Flow

- Actor log-in to system with credentials
- System grants actor access to system
- Based on actor's level of access to system, they will be able to view certain files
- The actor will then select their desired file and request to download a copy of it
- System will approve the download
- The actor will have a copy of the file on their device

183.2 Alternative Flows

183.2.1 < Unable to Download File >

- Actor requests file to download (File does not exist, user does not have permission to download, etc.)
- System rejects user request
- User receives error message

184. Pre-conditions

184.1 < Files Must Exist >

In order for the files to exist, they must exist within Adelante's system

184.2 < Files Must Have Different Levels of Permission >

The files within Adelante's system must have varying levels of permissions. This will control who can view and download the files

185. Post-conditions

185.1 **< Actor has Downloaded File >**
The actor has a copy of the requested file on their desired device

186. Extension Points

186.1 **<Cancel>**
At any point before the file is downloaded, the user should be able to cancel the download

Elaboration

Use Case Specification: <ReadFile>

Version <1.0>

Revision History

Date	Version	Description	Author
03/19/2016	<1.0>	ReadFile	<Hermilo Acuna>

Table of Contents

1. ReadFile **186**

1.1 Brief Description 186

2. Flow of Events **186**

2.1 Basic Flow 186

2.2 Alternative Flows 187

2.2.1 < Missing data > 187

3. Special Requirements **187**

3.1 < N/A > 187

4. Pre-conditions **187**

4.1 < Authentication Requirement > 187

4.2 < File Exists > 187

5. Post-conditions **187**

5.1 < Actor Informed > 187

6. Extension Points **187**

6.1 <Cancel> 187

Use Case Specification: <ReadFile>

187. ReadFile

187.1 Brief Description

This use case will allow the actor to retrieve a file that has been generated from the CreateFile Use Case and read / review the file.

188. Flow of Events

188.1 Basic Flow

- The actor requests the ReadFile Use Case
- The system verifies the actor's system access
- The system retrieves the newsletter for the actor
- The actor will read / review the newsletter
- The actor will determine to exit, save, or print

188.2 **Alternative Flows**

188.2.1 < Missing data >

1. The system failed to retrieve a file and prompts the actor that the file no longer exist

189. Special Requirements

189.1 **< N/A >**

N/A

190. Pre-conditions

190.1 **< Authentication Requirement >**

The actor must meet the following two conditions to use this feature.

5) The actor must be logged in through the system.
6) The actor must have authorization to access the feature.

190.2 **< File Exists >**

The file must exist in order for the system to retrieve the data for the actor

191. Post-conditions

191.1 **< Actor Informed >**

The actor has now been informed about anything that could have been on the file and chose to either print, save, or close out of the file.

192. Extension Points

192.1 **<Cancel>**

The actor at any point can exit the read file use case and the system will go back to its page before the actor chose this use case.

Elaboration

Use Case Specification: <UpdateFile>

Version <1.0>

Revision History

Date	Version	Description	Author
<03/18/2016 >	<1.0>	UpdateFile	<William Benedict>

Table of Contents

1. UpdateFile **191**

1.1 Brief Description 191

2. Flow of Events **191**

2.1 Basic Flow 191

2.2 Alternative Flows 191

2.2.1 < Uploading a updated file> 191

2.2.2 <Permission to Update flow> 191

3. Special Requirements **191**

4. Pre-conditions **192**

4.1 <Login & Authentication> 192

4.2 < File must already Exist > 192

5. Post-conditions **192**

5.1 <File Update State> 192

6. Extension Points **192**

6.1 <Cancel> 192

Use Case Specification: <UpdateFile>

193. UpdateFile

193.1 Brief Description

The UpdateFile use case illustrates how an actor, being Adelante staff or admin, can change or update a document that has already been created and saved.

194. Flow of Events

194.1 Basic Flow

- The actor selects which file they wish to edit.
- The actor can either select to edit the file
- The actor makes changes to the file.
- The actor saves and overwrites the old file

194.2 Alternative Flows

194.2.1 < Uploading a updated file>

The actor may select to upload an updated file instead of making edits directly to the text.

194.2.2 <Permission to Update flow>

- An actor without permission to update a file will be given an indicated of such.
- The actor may continue to view the document if they have required permissions, however, no edits will be accepted.

195. Special Requirements

196. Pre-conditions

196.1 **<Login & Authentication>**

The actor must be logged in and have a level of permission in order to update files. Some files may take a higher level of permission than others.

196.2 **< File must already Exist >**

In order to update something, the file being updated must already exist. If the file does not already exist, then the actor is instead creating a new file.

197. Post-conditions

197.1 **<File Update State>**

The file is either updated or not updated.

198. Extension Points

198.1 **<Cancel>**

At any point in the update process, the actor may cancel the process until the file is actually updated.

Elaboration

Use Case Specification: <UploadFile>

Version 1.0

Revision History

Date	Version	Description	Author
02/29/2016	1.0	UploadFile	Maam Awa Ndaw

Table of Contents

1. UploadFile **195**

1.1 Brief Description 195

2. Flow of Events **195**

2.1 Basic Flow 195

2.2 Alternative Flows 195

2.2.1 < System fails grant access to files to user > 195

2.2.2 < File isn't available in system > 195

3. Special Requirements **196**

3.1 < User must have access to files > 196

4. Pre-conditions **196**

4.1 < User has access to files > 196

4.2 < Files reside in system > 196

5. Post-conditions **196**

5.1 < Files uploaded successfully > 196

6. Extension Points **196**

6.1 < Uploaded file could be deleted > 196

Use Case Specification: <UploadFile>

199. UploadFile

199.1 Brief Description

This use case describes how an authorized user can upload files with student data to other systems such as JCPS Cascade.

200. Flow of Events

200.1 Basic Flow

The basic flow of events for Uploading a file is as follow:

- User enters username and password
- System authenticates user
- System grants user access to files
- User selects file to upload
- User instructs system to upload file into a location
- System uploads file

200.2 Alternative Flows

200.2.1 < System fails grant access to files to user >

User can log into system but system can fail to recognize user's credentials and therefore deny access to files. In that case, Admin could fix the issue and user would have to log out and log back in the system to be able to go through a normal flow of events for uploading a file.

200.2.2 < File isn't available in system >

There could be a scenario where user logs into system and the file to be uploaded is not available in the system, or residing outside the system. In that case, the file would have to be imported in the system and then the user would be able to perform a normal flow of event for uploading a file.

201. Special Requirements

201.1 **< User must have access to files >**

Uploading a file is possible only if the user was previously granted access to files within the system.

202. Pre-conditions

202.1 **< User has access to files >**

User's profile must be previously granted access to files

202.2 **< Files reside in system >**

Files must be available in the system and in waiting state.

203. Post-conditions

203.1 **< Files uploaded successfully >**

User's attempt to instruct the system to upload a file has been successfully completed.

204. Extension Points

204.1 **< Uploaded file could be deleted >**

Once a file has been successfully uploaded into a system, the user could delete the uploaded file and replace it with updated version of the file or a different file, that is if the option is available in that system.

Elaboration

Use Case Specification: <CreateShareFile>

Version 1.0

Revision History

Date	Version	Description	Author
02/29/2016	1.0	CreateShareFile	Maam Awa Ndaw

Table of Contents

1. CreateShareFile **200**

1.1 Brief Description 200

2. Flow of Events **200**

2.1 Basic Flow 200

2.2 Alternative Flows 200

2.2.1 < Limited access for listed contributors to the shared file > 200

2.2.2 < System fails to process the file owner's request and denies access to the file > 200

3. Special Requirements **201**

3.1 < Contributors would have to be in the mailing list or have an email address unrestricted within the system> 201

4. Pre-conditions **201**

4.1 < The file sharing feature must be available > 201

4.2 < Intended Contributors to the file must be present in the system or cleared for access.> 201

5. Post-conditions **201**

5.1 < Shared file > 201

6. Extension Points **201**

6.1 < Contributors could be removed or more contributors could be added to the file > 201

Use Case Specification: <CreateShareFile>

205. CreateShareFile

205.1 Brief Description

This use case describes how admin and staff members with permission can create and share documents with other members. It ultimately allows members to work collaboratively with one document.

206. Flow of Events

206.1 Basic Flow

The basic flow of events for sharing a file is as follow:

- Admin or employee creates, imports or selects a file to share
- Admin adds the contributors email addresses
- Admin enable the edit feature that will allow them to make updates to the file.
- System notifies every listed contributor
- Everyone with permission to edit the file can contribute

206.2 Alternative Flows

206.2.1 < Limited access for listed contributors to the shared file >

There could be a possibility that by default the contributors to the shared file have limited access and therefore cannot fully contribute to the document. In this case, the owner would have to rectify the current permission level to full access so that the file can be appropriately shared.

206.2.2 < System fails to process the file owner's request and denies access to the file >

There could be a scenario where the system doesn't share the file with any of the designed contributors. It could be due to a technical issue that is preventing a successful file sharing. The issue would have to be resolved before a normal flow of event for creating a shared file could be possible.

207. Special Requirements

207.1 < Contributors would have to be in the mailing list or have an email address unrestricted within the system>

In order for someone to create a shared file, the designated contributors to the file would have to be in the mailing list or given particular access to the system and eventually listed as a contributor in the file.

208. Pre-conditions

208.1 < The file sharing feature must be available >

For successful file sharing to occur, the system must have the option for sharing available for use.

208.2 < Intended Contributors to the file must be present in the system or cleared for access.>

In order for a file to be shared between two or more parties, all designated parties must be present in the system or given access to the file upon clearance.

209. Post-conditions

209.1 < Shared file >

After creating and sharing a file with other parties, the file become accessible and editable by everyone listed as contributor.

210. Extension Points

210.1 < Contributors could be removed or more contributors could be added to the file >

Depending on the circumstances, the owner of the file could make some adjustments to the list of people that the file was shared to, add or remove would be possible.

Elaboration

Use Case Specification: <DeleteShareFile>

Version <1.0>

Revision History

Date	Version	Description	Author
03/19/2016	<1.0>	DeleteShareFile	<Hermilo Acuna>

Table of Contents

1. DeleteShareFile **206**

1.1 Brief Description 206

2. Flow of Events **206**

2.1 Basic Flow 206

2.2 Alternative Flows 206

2.2.1 < Invalid Permission > 206

3. Special Requirements **206**

3.1 < N/A > 206

4. Pre-conditions **206**

4.1 < Authentication Requirement > 206

4.2 < Share File Existence > 206

5. Post-conditions **207**

5.1 < ShareFile Deleted > 207

6. Extension Points **207**

6.1 <Cancel> 207

Use Case Specification: <DeleteShareFile>

211. DeleteShareFile

211.1 Brief Description

This use case allows the actor with the correct system access to delete a share file. This is used in order to maintain a neat and up to date database without unnecessary outdated or duplicate share files. Everything on the cloud and the system will be deleted.

212. Flow of Events

212.1 Basic Flow

- The actor request share file that will be deleted
- The system will authenticate the actor's privileges
- The actor will select to delete the share file
- The actor will submit selection
- The system will erase all data pertaining to the share file

212.2 Alternative Flows

212.2.1 < Invalid Permission >

The actor requesting this feature will be denied and cannot delete the share file.

213. Special Requirements

213.1 < N/A >

N/A

214. Pre-conditions

214.1 < Authentication Requirement >

The actor must meet the following two conditions to use this feature.

- The actor must be logged in through the system.
- The actor must have authorization to access the feature.

214.2 < Share File Existence >

The share file must exist in order for the actor to delete it from the system

215. Post-conditions

215.1 **< ShareFile Deleted >**

The Share file is now deleted from the system and everything that it might have contained.

216. Extension Points

216.1 **<Cancel>**

The actor can cancel the action until the action has occurred which would then send the actor back before they accessed the DeleteShareFile use case.

Elaboration

Use Case Specification: <ReadShareFile>

Version <1.0>

Revision History

Date	Version	Description	Author
<03/18/2016>	<1.0>	ReadShareFile	<James Ryg>

Table of Contents

1. Use-Case Name **210**

1.1 Brief Description 47

2. Flow of Events **34**

2.1 Basic Flow 34

2.2 Alternative Flows 34

2.2.1 < File Load Error > 34

2.2.2 < Access Error > 51

3. Special Requirements **34**

3.1 < Share Hosting Requirement > 47

4. Pre-conditions **35**

4.1 < Login & Authentication > 35

5. Extension Points **48**

6.1 <Cancel> 48

Use Case Specification: <Use-Case Name>

217. Use-Case Name

217.1 Brief Description

This use case is a standard read use case. However, certain files may have restrictions on who can access them. As such, there are alternate flows for access and loading errors. The end result of this use case is that an actor or object is passed a copy or link to a shared file.

218. Flow of Events

218.1 Basic Flow

- An actor or object requests a ShareFile resource.
- If the permissions are sufficient, the system requests the ShareFile resource.
- The ShareFile storage returns the requested target.
- The system returns the file to the actor or object that requested it.

218.2 Alternative Flows

218.2.1 < File Load Error >

- The storage returns an exception to the system.
- The system displays a message to the actor or object who requested the resource.

218.2.2 < Access Error >

- In the event that the user somehow attempted to access a resource they cannot access, the system returns a message indicating the condition.

219. Special Requirements

219.1 < Share Hosting Requirement>

This use case relies on Adelante having some form of storage, such as a google drive, with valid links.

220. Pre-conditions

220.1 **< Login & Authentication Requirement >**

Certain files may have an access requirement that require an actor to be logged in. Others may not actually have this requirement.

221. Extension Points

221.1 **<Cancel>**

If a file is not accessible for any reason, the use case cancels. An actor would not normally encounter this condition.

Elaboration

Use Case Specification: <UpdateShareFile>

Version <1.0>

Revision History

Date	Version	Description	Author
03/02/2016	1.0	UpdateShareFile	Chad Tracy

Table of Contents

1. UpdateShareFile **214**

1.1 Brief Description 214

2. Flow of Events **214**

2.1 Basic Flow 214

2.2 Alternative Flows 214

2.2.1 <User Fails to Log-In to System > 214

2.2.2 < System Fails to Update > 214

3. Pre-conditions **215**

3.1 < System Must Recognize Staff Members as Admins > 215

4. Post-conditions **215**

4.1 < Shared Files Are Updated > 215

Use Case Specification: <UpdateShareFile>

222. UpdateShareFile

222.1 Brief Description

Allows the AHA staff and admins to make any necessary modifications to the files they share with updated information.

223. Flow of Events

223.1 Basic Flow

- Begins when the admin logs into system with their credentials (username and password)
- System approves admin and they can then navigate to files shared between staff members and admins within the system
- Once they navigate to the files, the staff member can properly update the data shared between the staff
- System saves any changes made to the shared files by the staff member and they are made available to other staff members.

223.2 Alternative Flows

223.2.1 <User Fails to Log-In to System >

- Staff member attempts to log-in to system with credentials (username and password)
- System rejects staff member's credentials and they aren't able to view or make updates to the system.

In this alternative flow the work is ended at the beginning when the staff member can't access the system. The issue can arise from the system not being updated on the proper credentials, a bug in the system, or the staff member making an error. Regardless of the error it must be resolved before the flow of work can continue.

223.2.2 < System Fails to Update >

- Staff member logs in to system with credentials
- System approves staff member and they are able to navigate to the shared files
- Staff member makes updates to the system
- System fails to save updates made to the system

This work flow is stopped at the end and doesn't retain any changes the staff member made to the shared files. The issue will be from a system error that prevents staff members from making updates. Before updates can be made in the future, any errors must be fixed.

224. Pre-conditions

224.1 **< System Must Recognize Staff Members as Admins >**

Having the status of admin allows the staff members to enter in the system with special privileges. These privileges include the ability to edit items within the system, which is required to update files shared between staff members.

4.2 < Shared Files Must Exist in System >

In order for updates to be made to the shared files, they must exist in the system. This way the staff members are able to access the files and view the updates to them after they are made.

225. Post-conditions

225.1 **< Shared Files Are Updated >**

When completed the changes the staff member made to the files will be updated, saved into the system, and available for the other staff members to see.

Elaboration

Use Case Specification: <CreateGrantTracker>

Version <1.0>

Revision History

Date	Version	Description	Author
<03/18/2016>	<1.0>	CreateGrantTracker	<William Benedict>

Table of Contents

1. CreateGrantTracker 219

1.1 Brief Description 219

2. Flow of Events 219

2.1 Basic Flow 219

2.2 Alternative Flows 219

2.2.1 < No Selected Details> 219

2.2.2 <No Grant Selected> 219

3. Special Requirements 220

3.1 <Grants must be listed and recorded> 220

4. Pre-conditions 220

4.1 <Login & Authentication> 220

5. Post-conditions 220

5.1 <Grant Tracker Created> 220

6. Extension Points 220

6.1 <Cancel> 220

Use Case Specification: <CreateGrantTracker>

226. CreateGrantTracker

226.1 Brief Description

This use case illustrates the how an actor can create a new grant tracker after applying for a grant. In order to create a grant tracker, there must have been a grant that has been applied for.

227. Flow of Events

227.1 Basic Flow

- The actor navigates to the grants page.
- The actor selects a grant that they want to track
- The actor selects what type of information they want to track
    - The actor may track money spent towards grant
    - The actor may track dates of when reporting is due
    - The actor may track the submission deadline
    - The actor may track grant status (Approved, In-Progress, Denied)
    - The actor may track when a grant period has opened (yearly grants)
- The actor submits their selection and selects how they want to be notified
    - The actor may be notified by email
    - The actor may be notified by an alert on the website.
    - The actor may be notified by text.

227.2 Alternative Flows

227.2.1 < No Selected Details>

- The actor selects no details to track
- The actor is notified by a message saying that they have not selected enough details to track.
- The actor is sent back to the page to select the details they wish to track.

227.2.2 <No Grant Selected>

- The actor does not select a grant to track
- The actor is notified by a message saying they have not selected a grant
- The actor is sent back to the page to select the grant

228. Special Requirements

228.1 **<Grants must be listed and recorded>**

In order for the grant tracker to work. All grants that can be and are applied for by Adelante must be recorded. Future grants must also be recorded when they are offered. Finally, grants that are no longer given must be removed from the record.

229. Pre-conditions

229.1 **<Login & Authentication>**

The actor must be signed in and have the level of permissions necessary to access the grants page.

230. Post-conditions

230.1 **<Grant Tracker Created>**

The new grant tracker is created.

231. Extension Points

231.1 **<Cancel>**

Until the grant tracker is actually created, the actor may stop the creation of the tracker at any time.

Elaboration

Use Case Specification: <DeleteGrantTracker>

Version 1.0

Revision History

Date	Version	Description	Author
02/20/2016	1.0	DeleteGrantTracker	Valencia Richard

Table of Contents

1. DeleteGrantTracker **224**

1.1 Brief Description 224

2. Flow of Events **224**

2.1 Basic Flow 224

2.2 Alternative Flows **224**

2.2 < Invalid Credentials> 224

2.3 < No Grant Tracker> 224

3. Special Requirements **225**

3.1 < Grant Tracker Connection to Grants> 225

4. Pre-conditions **225**

4.1 < Grant Tracker in Existence> 225

5. Post-conditions **225**

5.1 < Grant Tracker Deleted> 225

6. Extension Points **225**

6.1 < Cancel> 225

Use Case Specification: <DeleteGrantTracker>

232. DeleteGrantTracker

232.1 **Brief Description**

This use case allows an actor to delete the Grant Tracker from the system. At times, the Tracker may become outdates or may no longer function as it originally intended and mat need to be deleted

233. Flow of Events

233.1 **Basic Flow**

- The Actor initiates the Use Case by entering credentials in the AHA system
- The system accepts the credentials
- The actor goes to the area of the system containing the Grant Tracker
- The system deletes the file

2.2 Alternative Flows

233.2 ***< Invalid Credentials>***

- User enters information
- The system receives request to Login
- The system denies the request based on invalid login information

233.3 ***< No Grant Tracker>***

- User enters information
- The system receives request to Login
- The system accepts the credentials
- The actor goes to the area of the system containing the Grant Tracker
- The Grant Tracker does not exist

234. Special Requirements

234.1 **< Grant Tracker Connection to Grants>**

The Grant Tracker will be connected to grants and may be in charge of accounting for said grants. By deleted the Grant tracker steps must be taken to ensure that information about the Grants can still be accounted for

235. Pre-conditions

235.1 **< Grant Tracker in Existence>**

In order to delete the Grant Tracker, it must exist in the system in the first place

236. Post-conditions

236.1 **< Grant Tracker Deleted>**

The Grant Tracker will be deleted from the system.

237. Extension Points

237.1 **< Cancel>**

The actor will be prompted once to verify the desire to attempt to delete the Grant Tracker before the process is complete.

Elaboration

Use Case Specification: <ReadGrantTracker>

Version 1.0

Revision History

Date	Version	Description	Author
02/29/2016	1.0	ReadGrantTracker	Maam Awa Ndaw

Table of Contents

1. ReadGrantTracker 229

1.1 Brief Description 229

2. Flow of Events 229

2.1 Basic Flow 229

2.2 Alternative Flows 229

2.2.1 < User doesn't have appropriate permission to read grant tracker > 229

2.2.2 < Grant tracker isn't available for viewing > 229

3. Special Requirements 229

3.1 < Grant tracker must have been previously created > 229

4. Pre-conditions 230

4.1 < User was granted access to grant tracker > 230

4.2 < Grant tracker exists in system and available for viewing > 230

5. Post-conditions 230

5.1 < System displays grant tracker for viewing > 230

6. Extension Points 230

6.1 < Grant tracker updated > 230

Use Case Specification: <ReadGrantTracker>

238. ReadGrantTracker

238.1 Brief Description

This use case describes how a user with granted access can read the grant tracker.

239. Flow of Events

239.1 Basic Flow

The basic flow of events for reading or viewing the grant tracker is as follow:

- User enters username and password
- System authenticates user
- System grants access to user
- User navigates to grant tracker
- User reads grant tracker

239.2 Alternative Flows

239.2.1 < User doesn't have appropriate permission to read grant tracker >

There could be a scenario where the user hasn't been granted access to the grant tracker and attempts to read the grant tracker. In this case, access would have to be granted before the system can perform a normal flow of event for reading the grant tracker.

239.2.2 < Grant tracker isn't available for viewing >

The grant tracker might not be accessible in some cases and it could be due to maintenance or downtime. The user won't be able to read the grant tracker until the system is up and running and grant tracker is available for viewing.

240. Special Requirements

240.1 < Grant tracker must have been previously created >

For a user with granted access to be able to read the grant tracker, the grant tracker must exist in the system and accessible for viewing.

241. Pre-conditions

241.1 < User was granted access to grant tracker >

A user must be granted access to the grant tracker to be able to read the information in the grant tracker

241.2 < Grant tracker exists in system and available for viewing >

The grant tracker must have been previously created in the system and available for viewing.

242. Post-conditions

242.1 < System displays grant tracker for viewing >

User can read information in grant tracker.

243. Extension Points

243.1 < Grant tracker updated >

A user with appropriate access to grant tracker could decide to update the information in the grant tracker after reading the grant tracker.

Elaboration

Use Case Specification: <UpdateGrantTracker>

Version <1.0>

Revision History

Date	Version	Description	Author
<03/19/2016>	<1.0>	UpdateGrantTracker	<Hermilo Acuna>

Table of Contents

1. UpdateGrantTracker **235**

1.1 Brief Description 235

2. Flow of Events **235**

2.1 Basic Flow 235

2.2 Alternative Flows 235

2.2.1 < Missing data > 235

3. Special Requirements **235**

3.1 < N/A > 235

4. Pre-conditions **235**

4.1 < Authentication Requirement > 235

4.2 < Grant Existence > 235

5. Post-conditions **236**

5.1 < Grant Tracker Updated > 236

5.2 < Cloud Storage State > 236

6. Extension Points **236**

6.1 <Cancel> 236

Use Case Specification: <UpdateGrantTracker>

244. UpdateGrantTracker

244.1 Brief Description

This use case is used to maintain an up to date record of individual grant trackers. The actors in this case Adelante Employees and administrators they will have permission to make changes where the data has changed and once edited will be updated throughout the system using the cloud.

245. Flow of Events

245.1 Basic Flow

- The actor is on the grant tracker page
- The actor request to update the grant tracker
- The system verifies the actor's credentials
- The system will retrieve the grant tracker requested
- The actor will select certain features that pertain to the grant tracker
- The actor will then select update
- The actor will submit the selection
- The system updates the edited grant tracker
- The system uploads it to the cloud

245.2 Alternative Flows

245.2.1 < Missing data >

- The grant tracker doesn't exist and the system informs the actor that the requested grant tracker doesn't exist.

246. Special Requirements

246.1 < N/A >

N/A

247. Pre-conditions

247.1 < Authentication Requirement >

The actor must meet the following two conditions to use this feature.

- The actor must be logged in through the system.
- The actor must have authorization to access the feature.

247.2 < Grant Existence >

The grant must exist in order for the actor to update any of the data

248. Post-conditions

248.1 **< Grant Tracker Updated >**

The actor has now completely updated the grant tracker with the most recent data that they have inputted. The grant tracker will only update what the actor has selected and won't delete anything unless specified.

248.2 **< Cloud Storage State >**

The system will upload the grant tracker to the cloud in order to save the updated data.

249. Extension Points

249.1 **<Cancel>**

The actor at any point during the process of updating the grant tracker. Once the actor has confirmed that they wish to cancel all the current edited data for this session is deleted and the system returns to the previous point before the update grant tracker use case.

Elaboration

Use Case Specification: <CreateNewsletter>

Version <1.0>

Revision History

Date	Version	Description	Author
<3/5/2016>	<1.0>	CreateNewsletter	<James Ryg>

Table of Contents

1. CreateNewsletter **240**

1.1 *Brief Description* *240*

2. Flow of Events **240**

2.1 *Basic Flow* *240*

3. Pre-conditions **240**

3.1 *< Login & Authentication Requirement >* *240*

4. Post-conditions **240**

4.1 *< Newsletter state >* *240*

5. Extension Points **240**

5.1 *<Cancel>* *240*

Use Case Specification: <CreateNewsletter>

250. CreateNewsletter

250.1 **Brief Description**

This use case specifies the process of creating a newsletter to send to a group of people. Only adelante employees with specific access rights can actually utilize this functionality. Different groups of people can be selected, as well as multiple language publications can be queued to send.

251. Flow of Events

251.1 **Basic Flow**

- The actor begins creating a newsletter.
- The actor uploads the version they intend to send
- The actor selects the target audience
- The actor schedules the time to send.
- The actor submits the new newsletter.

252. Pre-conditions

252.1 **< Login & Authentication Requirement >**

The actor must already be logged in, and possess rights to create newsletter functionality.

253. Post-conditions

253.1 **< Newsletter state >**

A new newsletter is created and scheduled to send out.

254. Extension Points

254.1 **<Cancel>**

At any time, the actor can cancel the newsletter creation process.

Elaboration

Use Case Specification: <DeleteNewsletter>

Version 1.0

Revision History

Date	Version	Description	Author
02/20/2016	1.0	DeleteNewsletter	Valencia Richard

Table of Contents

1. DeleteNewsLetter **244**

1.1 Brief Description 244

2. Flow of Events **244**

2.1 Basic Flow 244

2.2 Alternative Flows **244**

2.2.1 < Invalid Credentials> 244

3. Pre-conditions **244**

3.1 < Newsletter in Existence> 244

3.2 < Login Credentials Verified> 245

4. Post-conditions **245**

4.1 < Newsletter Deleted> 245

5. Extension Points **245**

5.1 < Cancel> 245

Use Case Specification: <DeleteNewsletter>

1. DeleteNewsLetter

1.1 Brief Description

This use case allows an actor to delete a a NewsLetter from the newletters that exist in the system. Newsletters that have been previously sent out will not be deleted.

2. Flow of Events

2.1 Basic Flow

- The Actor initiates the Use Case by entering credentials in the AHA system
- The system accepts the credentials
- The actor locates the Newsletter to be deleted
- The actor deletes the NewsLetter
- The system recognizes the request for Delete
- The system asks the user to verify the delete
- The system deletes the file

2.2 Alternative Flows

2.2.1 < Invalid Credentials>

- User enters information
- The system receives request to Login
- The system denies the request based on invalid login information

3. Pre-conditions

3.1 < Newsletter in Existence>

In order to delete a Newsletter, the Newsletter must already exist in the system, otherwise there will be nothing to delete

3.2 < Login Credentials Verified>

An actor must have the proper login and password to access the system to delete a file. Only the top administration have this ability

4. Post-conditions

4.1 < Newsletter Deleted>

The Newsletter will be deleted from the system. Prior Newsletters will remain with the recipient they were originally sent to

5. Extension Points

5.1 < Cancel>

The actor will be prompted once to verify the desire to delete the Newsletter. At this point only, they can cancel the Delete request

Elaboration

Use Case Specification: <ReadNewsletter>

Version <1.0>

Revision History

Date	Version	Description	Author
03/19/2016	<1.0>	ReadNewsletter	<Hermilo Acuna>

Table of Contents

1. ReadNewsletter **249**

1.1 Brief Description 249

2. Flow of Events **249**

2.1 Basic Flow 249

. The actor will read / review the newsletter 249

2.2 Alternative Flows 249

2.2.1 < Print Newsletter > 249

2.2.2 < Close Newsletter > 249

3. Special Requirements **249**

3.1 < N/A > 249

4. Pre-conditions **249**

4.1 < Authentication Requirement > 249

4.2 < Newsletter Exists > 250

5. Post-conditions **250**

5.1 < Actor Informed > 250

6. Extension Points **250**

6.1 <Cancel> 250

Use Case Specification: <ReadNewsletter>

255. ReadNewsletter

255.1 Brief Description

This use case will allow the actor to retrieve a newsletter that has been generated from the CreateNewsletter Use Case and read / review the newsletter.

256. Flow of Events

256.1 Basic Flow

- The actor requests the ReadNewsletter Use Case.
- The system verifies the actor's system access
- The system retrieves the newsletter for the actor
- The actor will read / review the newsletter
- The actor will determine what further action to take

256.2 Alternative Flows

256.2.1 < Print Newsletter >

- The actor will print out the newsletter

256.2.2 < Close Newsletter >

The actor will close out of the newsletter

2.2.3 < Newsletter Nonexistent >

- The system will prompt an error to the actor informing them that the Newsletter doesn't or no longer exist

257. Special Requirements

257.1 < N/A >

N/A

258. Pre-conditions

258.1 < Authentication Requirement >

The actor must meet the following two conditions to use this feature.

- The actor must be logged in through the system.

- The actor must have authorization to access the feature.

258.2 **< Newsletter Exists >**

The Newsletter must exist in order for the system to retrieve the data for the actor

259. Post-conditions

259.1 **< Actor Informed >**

The actor has now been informed about anything that could have been on the newsletter and chose to either print, save, or close out of the newsletter.

260. Extension Points

260.1 **<Cancel>**

The actor at any point can exit the read newsletter use case and the system will go back to its page before the actor chose this use case.

Elaboration

Use Case Specification: <UpdateNewsLetter>

Version <1.0>

Revision History

Date	Version	Description	Author
02/29/2016	<1.0>	UpdateNewsLetter	Chad Tracy

Table of Contents

1. UpdateNewsLetter **253**

1.1 Brief Description 253

2. Flow of Events **253**

2.1 Basic Flow 253

2.2 Alternative Flows 253

2.2.1 < Failure to Authenticate Admin > 253

2.2.2 < Newsletter Fails to Update > 253

3. Special Requirements **253**

3.1 < Update System to Allow Certain Users to Edit Newsletter > 253

4. Pre-conditions **254**

4.1 < User Must Be Granted Permission to Make Edits> 254

4.2 <Newsletter Must Exist> 254

5. Post-conditions **254**

5.1 < Newsletter is Updated > 254

Use Case Specification: <UpdateNewsLetter>

261. UpdateNewsLetter

261.1 Brief Description

This use case describes the process for making updates to the company's newsletter, which is performed by the admin. Allows those with access to log in to the system and make the appropriate changes to the newsletter before it's sent out and posted to the site.

262. Flow of Events

262.1 Basic Flow

- User logs into system with unique username and password
- System authenticates user and grants admin privelages
- The logged in admin navigates to the newsletter files
- Within the newsletter files the admin is then able to make changes to the desired sections of the newsletter
- Admin requests the system save changes made to the newsletter
- System saves changes made by the admin
- Newsletter is updated

262.2 Alternative Flows

262.2.1 < Failure to Authenticate Admin >

- User attempts to log into system
- System rejects user
- User is unable to make updates
- Newsletter remains in its original state
- System prompts user to reenter credentials

262.2.2 < Newsletter Fails to Update >

- Admin logs-in to system and makes changes to newsletter
- Admin requests system save changes
- System fails to save any changes made by the admin

263. Special Requirements

263.1 < Update System to Allow Certain Users to Edit Newsletter >

For this part of the system to operate, the system must be allowed to assign users as admins or grant permission to edit the newsletter. This requirement ensures that the system is able to grant certain users the ability to make changes to the newsletter.

3.2 <System Must Allow Admin to Update When Completed>

In order for the results to show on the website, the admin must be granted permission to update the newsletter when completed. The admin must be in control of this so nothing is accidently updated when it shouldn't be (rough drafts, incomplete work, etc.). This will also allow the admin to save work and complete it at another time if need be

264. Pre-conditions

264.1 **< User Must Be Granted Permission to Make Edits>**

User in charge of making edits must be granted admin status or have permission from the system in order to make edits to the newsletter.

264.2 **<Newsletter Must Exist>**

For the admin to make changes to the newsletter it must exist within the system and in a state to be updated.

265. Post-conditions

265.1 **< Newsletter is Updated >**

Changes made by the admin within the system must be accurately reflected on the website

Elaboration

Use Case Specification: <PayPalRedirect>

Version <1.0>

Revision History

Date	Version	Description	Author
03/19/2016	<1.0>	PayPalRedirect	<Hermilo Acuna>

Table of Contents

1. PayPalRedirect **260**

1.1 Brief Description 260

2. Flow of Events **260**

2.1 Basic Flow 260

2.2 Alternative Flows 260

2.2.1 < Bad Link > 260

3. Special Requirements **260**

3.1 < PayPal Link > 260

4. Pre-conditions **260**

4.1 < Authentication Requirement > 260

5. Post-conditions **260**

5.1 < Redirected to PayPal> 260

6. Extension Points **261**

6.1 <Cancel> 261

Use Case Specification: <PayPalRedirect>

266. PayPalRedirect

266.1 Brief Description

This use case is used during the donation process where a user is in the process of donating money. Once the user is ready to donate this use case redirects you to a PayPal where the transaction can occur safely and is a well-known method of monetary transactions online.

267. Flow of Events

267.1 Basic Flow

- The user request to donate money by using the PayPal link
- The system begins to process the link and redirects the user
- PayPal responds to the request
- PayPal sets the page for the donation

267.2 Alternative Flows

267.2.1 < Bad Link >

- The page fails to load and informs the user that the link did not work correctly

268. Special Requirements

268.1 < PayPal Link >

- The system needs to have a link system with PayPal in order to process the redirection of traffic

269. Pre-conditions

269.1 < Authentication Requirement >

The actor must be on the page where it redirects you to PayPal

270. Post-conditions

270.1 < Redirected to PayPal>

The link has fully completed the process and has now transferred the user to a new PayPal page

271. Extension Points

271.1 **<Cancel>**

The actor can cancel the process up until before they click the link. The system will return you to the start point before you reached the link.

Elaboration

Use Case Specification: <CreateReport>

Version <1.0>

Revision History

Date	Version	Description	Author
03/29/2016	<1.0>	CreateReport	<Hermilo Acuna>

Table of Contents

1. Report **264**

1.1 Brief Description 264

2. Flow of Events **264**

2.1 Basic Flow 264

2.2 Alternative Flows 265

2.2.1 < Missing data > 265

3. Special Requirements **265**

3.1 < PDF Compatibility > 265

4. Pre-conditions **265**

4.1 < Authentication Requirement > 265

5. Post-conditions **265**

5.1 < Report Created> 265

5.2 < Cloud Storage Date > 265

6. Extension Points **265**

6.1 <Cancel> 265

Use Case Specification: <CreateReport>

272. Report

272.1 Brief Description

This use case allows verified actors such as Adelante employees and administrators to create reports for anything they need to keep track of and maintain current data. Reports can be used for anything from public revenue data to their volunteer data and their performance at the company.

273. Flow of Events

273.1 Basic Flow

- The Actor requests the create report page
- The system verifies the actor's system access
- The page is now unlocked for the actor to create a new report
- The actor will write up a new report
- The actor will save the report respectively
- The system will upload it to the cloud
- The system will organize its location base on type of report

273.2 Alternative Flows

273.2.1 < Missing data >

2. The report failed to save properly and is missing from the system

274. Special Requirements

274.1 < PDF Compatibility >

The report needs to be PDF compatible in order to present it in the system for other users to see and download easily without manipulating it.

275. Pre-conditions

275.1 < Authentication Requirement >

The actor must meet the following two conditions to use this feature.

7) The actor must be logged in through the system.
8) The actor must have authorization to access the feature.

276. Post-conditions

276.1 < Report Created>

The report is now created and stored as a PDF on the cloud in order to access anywhere. The system will store it in its corresponding page for better organization.

276.2 < Cloud Storage Date >

The system will upload the report to the cloud in order to save the data.

277. Extension Points

277.1 <Cancel>

The actor at any point during the create a calendar date page can cancel the process. This will delete any current data that the actor has inputted up to this point and resume o the system as is before the actor began to create a calendar date.

Elaboration

Use Case Specification: <DeleteReport>

Version <1.0>

Revision History

Date	Version	Description	Author
<3/5/2016>	<1.0>	DeleteReport	James Ryg

Table of Contents

1. DeleteReport **269**

1.1 Brief Description 269

2. Flow of Events **269**

2.1 Basic Flow 269

2.2 Alternative Flows 269

2.2.1 < Lack of permission to delete flow> 269

2.2.2 < Lack of permission to read flow > 269

3. Pre-conditions **269**

3.1 < Login & Authentication Requirement> 269

3.2 <Report Existence Requirement> 269

4. Post-conditions **270**

4.1 < Report State > 270

5. Extension Points **270**

5.1 <Cancel> 270

Use Case Specification: <DeleteReport>

278. DeleteReport

278.1 **Brief Description**

The DeleteReport use case illustrates functionality regarding deleting reports presets. Physical copies an actor downloads are not deleted however. Only users with access to the reporting mechanics will have the ability to delete the reports. Someone with authorization to merely view reports may not necessarily have the ability to delete them as well.

279. Flow of Events

279.1 **Basic Flow**

- The actor loads the list of reports.
- The actor selects delete.
- The actor submits.

279.2 **Alternative Flows**

279.2.1 < Lack of permission to delete flow>

- The actor is displayed a message indicating that they do not possess rights to delete a specific report.
- The actor is directed back to the list of reports.

279.2.2 < Lack of permission to read flow >

- The actor is displayed a message that the specific report is not visible to them.
- The actor is directed back to the main page or list of reports depending on severity and how they got to that page.

280. Pre-conditions

280.1 **< Login & Authentication Requirement>**

The actor must be logged in, and have a level of permission sufficient to both view and delete reports.

280.2 **<Report Existence Requirement>**

There must be reports already created to delete.

281. Post-conditions

281.1 **< Report State >**
The report can either be deleted or not deleted.

282. Extension Points

282.1 **<Cancel>**
At any point, the actor can cancel deletion until the report is actually deleted.

Elaboration

Use Case Specification: <ReadReport>

Version 1.0

Revision History

Date	Version	Description	Author
02/20/2016	1.0	ReadReport	Valencia Richard

Table of Contents

1. ReadReport **273**

1.1. Brief Description 273

2. Flow of Events **273**

2.1. Basic Flow 273

2.2. Alternative Flows **273**

2.2.1. < No Report > 273

1.1 273

2.2.2. < Invalid Credentials> 273

3. Pre-conditions **274**

3.1. < Report Generated> 274

3.2. < Login Credentials Verified> 274

4. Post-conditions **274**

4.1. < Report Read> 274

5. Extension Points **274**

5.1. < Cancel> 274

Use Case Specification: <ReadReport>

1. ReadReport

1.1. Brief Description

This use case allows an actor to read a report that has been genereated. The actor will pull the report from the Objec created to generate reports

2. Flow of Events

2.1. Basic Flow

- The Actor initiates the Use Case by logging to the system
- The actor locates the proper report
- The actor request to view the report
- The system retrieves the report as a downloaded PDF
- The actor views the report
- The actor can print the report

2.2. Alternative Flows

2.2.1. < No Report >

- The Actor logs to the system successfully
- The actor locates the proper report
- The actor request to view the report
- The system does not retrieve the report because it is missing
- The system alerts the user that the report cannot be retrieved for the corresponding reason

282.2

2.2.2. < Invalid Credentials>

- The Actor logs to the system

- The system denies the login attempt
- The system displays a message with appropriate reasoning

3. Pre-conditions

3.1. **< Report Generated>**

In order to view the report, it must exist. The report must have been created and must be waiting in the system

3.2. **< Login Credentials Verified>**

An actor must have the proper login and password to access the system to create approve the student Create request

4. Post-conditions

4.1. **< Report Read>**

The report will be viewed and can be downloaded or printer.

5. Extension Points

5.1. **< Cancel>**

An actor can cancel their Report Read at any time by exiting the page before the download is complete

Elaboration

Use Case Specification: <UpdateReport>

Version <1.0>

Revision History

Date	Version	Description	Author
<03/18/2016>	<1.0>	UpdateReport	<William Benedict>

Table of Contents

1. UpdateReport **278**

1.1 Brief Description 278

2. Flow of Events **278**

2.1 Basic Flow 278

2.2 Alternative Flows 278

2.2.1 <Permission to update flow> 278

3. Special Requirements **278**

4. Pre-conditions **278**

4.1 <Login & Authorization Requirement > 278

4.2 <Report Preset Existence> 278

5. Post-conditions **279**

5.1 <Report Preset State> 279

6. Extension Points **279**

6.1 <Cancel> 279

Use Case Specification: <UpdateReport>

283. UpdateReport

283.1 **Brief Description**

The UpdateReport use case illustrates how an actor can change report preset, Physical copies of the report downloads the actor has downloaded will not be changed. Only users with access to the reporting feature will have the ability to change report presets. Someone with the ability to view reports may not necessarily have the access to change them as well.

284. Flow of Events

284.1 **Basic Flow**

- The actor selects update preset they wish to edit
- The actor adds or removes parts of the report form
- The actor submits the changes

284.2 **Alternative Flows**

284.2.1 <Permission to update flow>

- The actor does not have a level of permission that allows making changes to report forms.
- The actor is directed back to the report form view

285. Special Requirements

286. Pre-conditions

286.1 **<Login & Authorization Requirement >**

An actor must be logged in and have a level of permission to update report presets.

286.2 **<Report Preset Existence>**

The report preset being updated must already exist.

287. Post-conditions

287.1 **<Report Preset State>**
The report preset is either updated or not updated

288. Extension Points

288.1 **<Cancel>**
The actor may cancel the update process at any point in time, until the preset is actually updated.

Elaboration

Use Case Specification: <CreateSpendingTracker>

Version 1.0

Revision History

Date	Version	Description	Author
03/03/2016	1.0	CreateSpendingTracker	Chad Tracy

Table of Contents

1. CreateSpendingTracker 282

1.1 Brief Description 282

2. Flow of Events 282

2.1 Basic Flow 282

2.2 Alternative Flows 282

2.2.1 < System Error Creation of Tracker Fails> 282

2.2.2 < System Fails to Save Tracker > 282

3. Special Requirements 282

3.1 < All Spending Must be Integrated into System > 282

4. Pre-conditions 283

4.1 < Log-In Authentication > 283

4.2 <Functionality within System to Build Tracker> 283

5. Post-conditions 283

5.1 < Tracker is Created > 283

6. Extension Points 283

Use Case Specification: <CreateSpendingTracker>

289. CreateSpendingTracker

289.1 Brief Description

Admin creates an object within the system in order to track the spending of the company. The spending being tracked includes but is not limited to the amount spent, when it's spent, and the type of money being spent. Allows spending to be tracked, managed, and updated in the future.

290. Flow of Events

290.1 Basic Flow

- User logs-in to system with credentials (username and password)
- System lets user into system and grants them admin privileges
- Admin decides what they want to be tracked
- Admin creates object within system that will track the spending of AHA with the desired specifications (type, amount, date spent)
- System saves data and implements the tracker
- Tracker begins tracking the spending and is available within the system to admins for updating purposes

290.2 Alternative Flows

290.2.1 < System Error Creation of Tracker Fails>

- Admin finishes creation of tracker
- System rejects the tracker and admin is shown an error message containing where the error is
- Admin is prompted to create new input for the creation of the tracker

290.2.2 < System Fails to Save Tracker >

- Admin finishes creation of tracker and saves it within the system
- System fails to save the tracker
- Admin is returned to beginning screen in order to enter data again or begin process to fix bug in system

291. Special Requirements

291.1 < All Spending Must be Integrated into System >

In order for the tracker to accurately track the money spent by the company all of AHA's spending must go through the system.

292. Pre-conditions

292.1 **< Log-In Authentication >**

Admin must be logged in to system and have the necessary privileges in order to create the tracker

292.2 **<Functionality within System to Build Tracker>**

The function to create trackers must be present and functioning within the system in order to make sure the function is able to proceed.

293. Post-conditions

293.1 **< Tracker is Created >**

A new tracker is created and implemented into the system

294. Extension Points

<None>

Elaboration

Use Case Specification: <DeleteSpendingTracker>

Version <1.0>

Revision History

Date	Version	Description	Author
<03/18/2016>	<1.0>	DeleteSpendingTracker	<James Ryg>

Table of Contents

1. DeleteSpendingTracker **210**

1.1 Brief Description 47

2. Flow of Events **34**

2.1 Basic Flow 34

2.2 Alternative Flows 34

2.2.1 < No Tracker > 34

3. Special Requirements **34**

3.1 < Spending Processor Requirement > 47

4. Pre-conditions **35**

4.1 < Login & Authentication Requirement > 35

5. Post-conditions **35**

5.1 < Spending Tracker State > 47

6. Extension Points **48**

6.1 <Cancel> 48

Use Case Specification: <DeleteSpendingTracker>

295. DeleteSpendingTracker

295.1 **Brief Description**

This use case illustrates the mechanics and actions that are required to delete a previously set up spending tracker. It has a few prerequisites, and few alternate flows, because it is a single purpose case. Only employees with proper access will be able to actually delete spending trackers.

296. Flow of Events

296.1 **Basic Flow**

- The actor opens the spending trackers options.
- The actor selects a spending tracker.
- The actor options to delete the spending tracker.

296.2 **Alternative Flows**

296.2.1 <No Spending Tracker>

- The spending trackers do not exist.
- The actor is notified of the tracker state.

297. Special Requirements

297.1 **< Spending processor requirement >**

The spending tracker requires the ability to track external spending. Possibly implementing external software apis.

298. Pre-conditions

298.1 **< Login & Authentication Requirement >**

The actor must be logged in and have sufficient permissions to access the resources required.

299. Post-conditions

299.1 **< Spending Tracker State >**
The spending tracker is deleted.

300. Extension Points

300.1 **<Cancel>**
Until the tracker is actually deleted, the actor can simply cancel changes.

<Elaboration >

Use Case Specification: <ReadSpendingTracker>

Version <1.0>

Revision History

Date	Version	Description	Author
03/18/2016	1.0	ReadSpendingTracker	Valencia Richard

Table of Contents

1. ReadSpendingTracker **291**

1.1. Brief Description *291*

2. Flow of Events **291**

2.1. Basic Flow *291*

2.2. Alternative Flows **291**

2.2.1. < No Tracker Tool > *291*

2.2.2. < Not Authorized to View Tracker> *291*

3. Special Requirements **291**

3.1. < Tracker must be created > *291*

3.2. < Authorized to view Tracker > *292*

4. Pre-conditions **292**

4.1. < First Alternative Flow > *292*

5. Post-conditions **292**

5.1. < Spending Tracker Accessed > *292*

Use Case Specification: <ReadSpendingTracker>

1. ReadSpendingTracker

1.1. Brief Description

This use case highlights how user Reads the Spending Tracker. Users are accessing information only. No changes are made but there are restrictions on the Tracker that must be accounted for.

2. Flow of Events

2.1. Basic Flow

- The Actor access the AHA system
- The actor navigates to the Tracker Tool
- The actor enters credentials
- The system verifies the credentials
- The system allows the actor to access the Tracker if the credentials are valid
- The actor accesses the Tracker Tool

2.2. Alternative Flows

2.2.1. < No Tracker Tool >

In order for a user to read the results from a query by the Tracker Tool, the Tool itself must have first been created

2.2.2. *< Not Authorized to View Tracker>*

In order to access the Tracker tool you need to have the proper authorization. A situation may arise in which someone that should not be able to generate Tracking reports may try to. Since they do they do not have clearance access will be denied

3. Special Requirements

3.1. < Tracker must be created >

A Spending Tracker tool must first exist in order for the results to be read

3.2. **< Authorized to view Tracker >**

An actor must have the proper permissions to log into the Tracker Tool

4. Pre-conditions

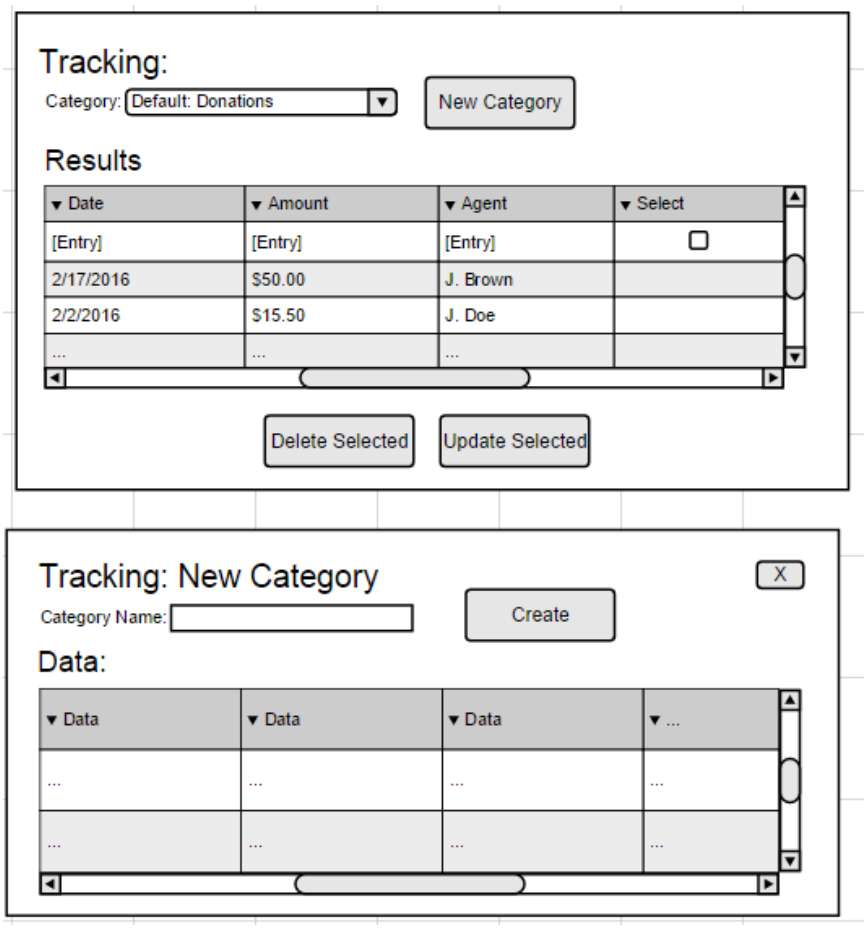
4.1. **< First Alternative Flow >**

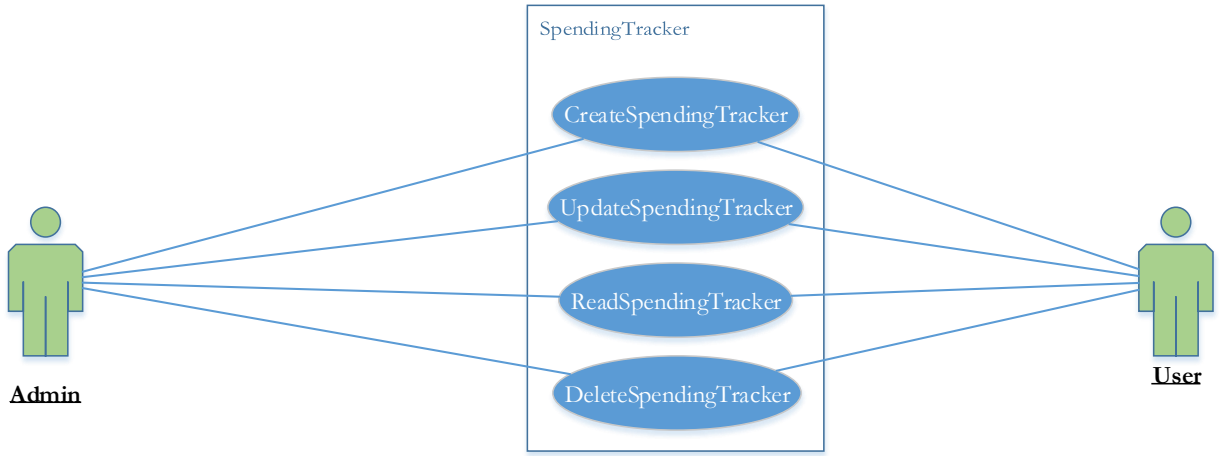
Tracker tool in wait state

5. Post-conditions

5.1. **< Spending Tracker Accessed >**

The spending tracker has generated reports that may be downloaded, but otherwise has not been altered


Tracking:
Category: Default: Donations
New Category
Results
Date
Amount
Agent
Select
[Entry]
[Entry]
[Entry]
2/17/2016
$50.00
J. Brown
2/2/2016
$15.50
J. Doe
Delete Selected
Update Selected
Tracking: New Category
X
Category Name:
Create
Data:
Data
Data
Data
...



Elaboration

Use Case Specification: <UpdateSpendingTracker>

Version 1.0

Revision History

Date	Version	Description	Author
02/29/2016	1.0	UpdateSpendingTracker	Maam Awa Ndaw

Table of Contents

1. UpdateSpendingTracker 297

1.1 Brief Description 297

2. Flow of Events 297

2.1 Basic Flow 297

2.2 Alternative Flows 297

2.2.1 < System error could occur as admin tries to view spending tracker > 297

2.2.2 < Unsuccessful update to spending tracker. > 297

3. Special Requirements 298

3.1 < A spending tracker would have to have been previously created > 298

4. Pre-conditions 298

4.1 < Spending tracker is available and in stand by mode > 298

4.2 < User must have access to edit features of spending tracker > 298

5. Post-conditions 298

5.1 < Updated Spending Tracker > 298

6. Extension Points 298

6.1 < Updated spending tracker could further be updated > 298

Use Case Specification: <UpdateSpendingTracker>

301. UpdateSpendingTracker

301.1 Brief Description

This use case describes how admin can update a spending tracker previously created within this system to give a better and accurate information or reflect a particular change or need.

302. Flow of Events

302.1 Basic Flow

The basic flow of events for updating the spending tracker is as follow:

- Admin navigates to the spending tracker tool
- Admin selects to edit the spending tracker
- System displays all editable features for use
- Admin chooses to make changes to a specific spending tracker such as date or category of spending tracked
- System records changes to spending tracker
- System displays changes to spending tracker

302.2 Alternative Flows

302.2.1 < System error could occur as admin tries to view spending tracker >

There could be a possibility that the system is not displaying the spending tracker due to a technical issue. In that case, the issue would have to be resolved before the a normal flow of event following updating the spending tracker could be possible.

302.2.2 < Unsuccessful update to spending tracker. >

There could be a scenario where the system is unresponsive to user input and fails to update the spending tracker to reflect changes made accordingly. Before a normal flow of event can occur, the root cause of the issue would have to be resolved first.

303. Special Requirements

303.1 **< A spending tracker would have to have been previously created >**

To make an update to a spending tracker, it would have to exist within the system first, that is, it would have to have been previously created.

304. Pre-conditions

304.1 **< Spending tracker is available and in stand by mode >**

To make changes to a spending tracker possible, it would have to be available within the system and ready for use.

304.2 **< User must have access to edit features of spending tracker >**

Spending tracker edit features must be available and accessible by the user for any update to the spending tracker to be possible.

305. Post-conditions

305.1 **< Updated Spending Tracker >**

After making changes to the spending tracker, the system would display an updated spending tracker that reflect the changes made.

306. Extension Points

306.1 **< Updated spending tracker could further be updated >**

Depending on the circumstances, changes could be made to the updated spending tracker.

Elaboration

Use Case Specification: <DeleteStudent>

Version 1.0

Revision History

Date	Version	Description	Author
02/20/2016	1.0	DeleteStudent	Maam Awa Ndaw

Table of Contents

1. DeleteStudent **302**

1.1 Brief Description 302

2. Flow of Events **302**

2.1 Basic Flow 302

2.2 Alternative Flows 302

2.2.1 < System partially deletes student information> 302

2.2.2 < Unsuccessful delete attempt > 302

3. Special Requirements **302**

3.1 < A student's records must exist in the system > 302

4. Pre-conditions **303**

4.1 < Student's account in available in system> 303

4.2 <Student's account deletion has been approved > 303

5. Post-conditions **303**

5.1 < Student's account is deleted from the system > 303

6. Extension Points **303**

6.1 <None > 303

Use Case Specification: <DeleteStudent>

307. DeleteStudent

307.1 Brief Description

This use case describes how Admin can delete information about a student previously stored in the system.

308. Flow of Events

308.1 Basic Flow

The basic flow of events for deleting a student's profile is as follow:

- Admin enters username and password
- Admin navigates to student's profile
- Admin instruct system to delete student's profile
- System deletes student's record
- Student's profile is removed from system

308.2 Alternative Flows

308.2.1 < System partially deletes student information>

There could be a possibility that the system doesn't delete all of the student's information. In that case, admin could instruct the system to delete all information pertaining to student and when successfully done, all student information would be removed from the system.

308.2.2 < Unsuccessful delete attempt >

There could be a scenario where Admin instructs the system to delete the student's account but the system is not responding to the instructions. In that case, Admin will not be able to delete the student's information until the system become responsive.

309. Special Requirements

309.1 < A student's records must exist in the system >

In order for deletion of student's account to be possible, the student's profile must have been previously created and resides in the system.

310. Pre-conditions

310.1 **< Student's account in available in system>**

To be able to delete student's account from the system, the student's profile must be available in the system and either active or inactive.

310.2 **<Student's account deletion has been approved >**

The student's account deletion must be approved by appropriate parties before beginning the account deletion process.

311. Post-conditions

311.1 **< Student's account is deleted from the system >**

After instructing the system to delete the student's profile, the student is removed from the system.

312. Extension Points

312.1 **<None >**

Elaboration

Use Case Specification: <ReadStudent>

Version <1.0>

Revision History

Date	Version	Description	Author
<03/18/2016>	<1.0>	ReadStudent	<William Benedict>

Table of Contents

1. ReadStudent **306**

1.1 Brief Description 306

2. Flow of Events **306**

2.1 Basic Flow 306

2.2 Alternative Flows 306

2.2.1 <No or Some Permission Level to access student information> 306

3. Special Requirements **306**

3.1 <Protected Information> 306

4. Pre-conditions **306**

4.1 < Login and Authorization> 306

4.2 < Student Existence > 307

5. Post-conditions **307**

5.1 < Change in State > 307

6. Extension Points **307**

Use Case Specification: <ReadStudent>

313. ReadStudent

313.1 Brief Description

The purpose of this use case is to illustrate how an actor can view details about a student that has previously been created.

314. Flow of Events

314.1 Basic Flow

- The actor navigates to the list of students
- The actor selects a student they wish to view the information of
- The actor selects view student details.
- The actor can view details regarding the student that had been previously entered in during the creation process

314.2 Alternative Flows

314.2.1 <No or Some Permission Level to access student information>

- An actor, with insufficient permissions, may be unable to access the list of students and;
- An actor, with some permissions, may be able to access only very basic information.

315. Special Requirements

315.1 <Protected Information>

Some of the information regarding students may be protected under FERPA, Therefore, only people who are on a need to know basis can see certain details regarding a student.

316. Pre-conditions

316.1 < Login and Authorization>

An actor must be signed in and have permissions to access student data.

316.2 **< Student Existence >**

The student must have already been created to be read.

317. Post-conditions

317.1 **< Change in State >**

During the read process, there is no change in student data. Therefore, there is no change in state.

318. Extension Points

Elaboration

Use Case Specification: <UpdateStudent>

Version <1.0>

Revision History

Date	Version	Description	Author
03/04/2016	<1.0>	UpdateStudent	Chad Tracy

Table of Contents

1. UpdateStudent **310**

1.1 Brief Description 310

2. Flow of Events **310**

2.1 Basic Flow 310

2.2 Alternative Flows 310

2.2.1 < System Error: Student Record Not Updated > 310

2.2.2 < User Credentials Are Rejected > 310

3. Special Requirements **310**

3.1 < System Must Recognize Admins > 310

4. Pre-conditions **311**

4.1 < Student Record Must Exist > 311

5. Post-conditions **311**

5.1 < Student Record is Updated > 311

Use Case Specification: <UpdateStudent>

319. UpdateStudent

319.1 **Brief Description**

Allows the admin to enter system and update student records with necessary information. Information includes a student's name, address, contact information, status in the system, and another information the user decides is necessary to update student records with.

320. Flow of Events

320.1 **Basic Flow**

- User requests access to system (username and password)
- System approves user credentials and grants them admin privileges
- Admin accesses student data and makes appropriate changes
- Admin saves changes made to student records
- System saves any changes made by the Admin
- Updated student records are updated and displayed to those who have access to them within the system

320.2 **Alternative Flows**

320.2.1 < System Error: Student Record Not Updated >

- Admin makes changes to student record in the system and requests the system save changes made
- System fails to save changes made by admin
- Student record is not updated
- System prompts admin to reenter data

320.2.2 < User Credentials Are Rejected >

- User attempts to log-in to system with credentials (username and password)
- System rejects user and they are returned to log-in screen
- Failure for the user to log-in can come from the system lacking updated credential information or the user not entering their credentials correct

321. Special Requirements

321.1 **< System Must Recognize Admins >**

The correct users must be recognized as admins within the system. This allows the admins to navigate through the system and make updates to the Student record

322. Pre-conditions

322.1 **< Student Record Must Exist >**

In order for the admin to make updates to the student record within the system there must be a student record that already exists

323. Post-conditions

323.1 **< Student Record is Updated >**

After the flow is completed the student record should be up to date with the information the admin submitted

Elaboration

Use Case Specification: <CreateStudent>

Version 1.0

Revision History

Date	Version	Description	Author
02/20/2016	1.0	CreateStudent	Valencia Richard

Table of Contents

1. CreateStudent **314**

1.1 Brief Description 314

2. Flow of Events **314**

2.1 Basic Flow 314

2.2 Alternative Flows 314

2.2.1 < Student Account Creation is Denied > 314

2.2.2 < Unsuccessful Account Creation > 314

3. Special Requirements **315**

3.1 < An identifier, such as Account ID must be assigned to the account profile> 315

4. Pre-conditions **315**

4.1 < Student must be accepted at Adelante> 315

4.2 <Account Sign up form Available> 315

4.3 < Account Creation is requested> 315

5. Post-conditions **315**

5.1 < A New Account Profile > 315

6. Extension Points **315**

6.1 <Edit Profile> 315

Use Case Specification: <CreateStudent>

324. CreateStudent

324.1 Brief Description

This use case describes how a student can create an account with their parents' guidance or not and upon acceptance and registration at Adelante. When the student fills out the account sign up form, a request is sent to the system administrator and subsequently their profile is created upon acceptance.

325. Flow of Events

325.1 Basic Flow

The basic flow of events for creating a student's profile is as follow:

- Student visits Adelante Website and signs up for an account.
- Student fills out sign up form with their personal information such as name, address, school, grade level
- System records and sends student's information to Admin
- Admin reviews the information and accepts the request.
- A confirmation of account creation is subsequently created
- Student's records reside within the system.
- Student's account becomes active.

325.2 Alternative Flows

325.2.1 < Student Account Creation is Denied >

There could be a possibility that the student is not accepted into the program or hasn't completed the process preceding account creation. In that case, admin could deny the account creation because the student isn't officially accepted into Adelante yet. The student's profile will not be created then.

325.2.2 < Unsuccessful Account Creation >

There could be a scenario where the student is accepted into the program and tries to create an account but the system is not storing user input or some other technical issue within the system is preventing the user from creating an account. In that case, the student will not be able to create an account until the issue is fixed.

326. Special Requirements

326.1 < An identifier, such as Account ID must be assigned to the account profile>

Each time a new account profile is created, a unique identifier such as an ID must be assigned to a new record associated with the account profile. ID will be essentially used to access the record within the system.

327. Pre-conditions

327.1 < Student must be accepted at Adelante>

The student's account creation mainly depends on whether or not s/he is accepted at Adelante. For an account creation to be possible, the student must be accepted and enrolled at Adelante.

327.2 <Account Sign up form Available>

The student must be able to sign up for an account, hence the account sign up form must be available and in standby mode within the system.

327.3 < Account Creation is requested>

The student has to submit a filled out sign up form to the admin in order to create a request for an account creation.

328. Post-conditions

328.1 < A New Account Profile >

Upon account creation, a new profile is created and assigned to the student who requested it. Any future information about that student will be stored in their user profile.

329. Extension Points

329.1 <Edit Profile>

It is likely that information stored about the student will change in the future such as address, school, grade level. Admin or student will then edit the records and make changes accordingly.

Elaboration

Use Case Specification: <CreateStudent>

Version 1.0

Revision History

Date	Version	Description	Author
02/20/2016	1.0	CreateStudent	Valencia Richard

Table of Contents

1. CreateStudent **314**

1.1 Brief Description 314

2. Flow of Events **314**

2.1 Basic Flow 314

2.2 Alternative Flows 314

2.2.1 < Student Account Creation is Denied > 314

2.2.2 < Unsuccessful Account Creation > 314

3. Special Requirements **315**

3.1 < An identifier, such as Account ID must be assigned to the account profile> 315

4. Pre-conditions **315**

4.1 < Student must be accepted at Adelante> 315

4.2 <Account Sign up form Available> 315

4.3 < Account Creation is requested> 315

5. Post-conditions **315**

5.1 < A New Account Profile > 315

6. Extension Points **315**

6.1 <Edit Profile> 315

Use Case Specification: <CreateStudent>

330. CreateStudent

330.1 Brief Description

This use case describes how a student can create an account with their parents' guidance or not and upon acceptance and registration at Adelante. When the student fills out the account sign up form, a request is sent to the system administrator and subsequently their profile is created upon acceptance.

331. Flow of Events

331.1 Basic Flow

The basic flow of events for creating a student's profile is as follow:

- Student visits Adelante Website and signs up for an account.
- Student fills out sign up form with their personal information such as name, address, school, grade level
- System records and sends student's information to Admin
- Admin reviews the information and accepts the request.
- A confirmation of account creation is subsequently created
- Student's records reside within the system.
- Student's account becomes active.

331.2 Alternative Flows

331.2.1 < Student Account Creation is Denied >

There could be a possibility that the student is not accepted into the program or hasn't completed the process preceding account creation. In that case, admin could deny the account creation because the student isn't officially accepted into Adelante yet. The student's profile will not be created then.

331.2.2 < Unsuccessful Account Creation >

There could be a scenario where the student is accepted into the program and tries to create an account but the system is not storing user input or some other technical issue within the system is preventing the user from creating an account. In that case, the student will not be able to create an account until the issue is fixed.

332. Special Requirements

332.1 < An identifier, such as Account ID must be assigned to the account profile>

Each time a new account profile is created, a unique identifier such as an ID must be assigned to a new record associated with the account profile. ID will be essentially used to access the record within the system.

333. Pre-conditions

333.1 < Student must be accepted at Adelante>

The student's account creation mainly depends on whether or not s/he is accepted at Adelante. For an account creation to be possible, the student must be accepted and enrolled at Adelante.

333.2 <Account Sign up form Available>

The student must be able to sign up for an account, hence the account sign up form must be available and in standby mode within the system.

333.3 < Account Creation is requested>

The student has to submit a filled out sign up form to the admin in order to create a request for an account creation.

334. Post-conditions

334.1 < A New Account Profile >

Upon account creation, a new profile is created and assigned to the student who requested it. Any future information about that student will be stored in their user profile.

335. Extension Points

335.1 <Edit Profile>

It is likely that information stored about the student will change in the future such as address, school, grade level. Admin or student will then edit the records and make changes accordingly.

Elaboration

Use Case Specification: <DeleteStudent>

Version 1.0

Revision History

Date	Version	Description	Author
02/20/2016	1.0	DeleteStudent	Maam Awa Ndaw

Table of Contents

1. DeleteStudent **302**

1.1 Brief Description 302

2. Flow of Events **302**

2.1 Basic Flow 302

2.2 Alternative Flows 302

2.2.1 < System partially deletes student information> 302

2.2.2 < Unsuccessful delete attempt > 302

3. Special Requirements **302**

3.1 < A student's records must exist in the system > 302

4. Pre-conditions **303**

4.1 < Student's account in available in system> 303

4.2 <Student's account deletion has been approved > 303

5. Post-conditions **303**

5.1 < Student's account is deleted from the system > 303

6. Extension Points **303**

6.1 <None > 303

Use Case Specification: <DeleteStudent>

336. DeleteStudent

336.1 Brief Description

This use case describes how Admin can delete information about a student previously stored in the system.

337. Flow of Events

337.1 Basic Flow

The basic flow of events for deleting a student's profile is as follow:

- Admin enters username and password
- Admin navigates to student's profile
- Admin instruct system to delete student's profile
- System deletes student's record
- Student's profile is removed from system

337.2 Alternative Flows

337.2.1 < System partially deletes student information>

There could be a possibility that the system doesn't delete all of the student's information. In that case, admin could instruct the system to delete all information pertaining to student and when successfully done, all student information would be removed from the system.

337.2.2 < Unsuccessful delete attempt >

There could be a scenario where Admin instructs the system to delete the student's account but the system is not responding to the instructions. In that case, Admin will not be able to delete the student's information until the system become responsive.

338. Special Requirements

338.1 < A student's records must exist in the system >

In order for deletion of student's account to be possible, the student's profile must have been previously created and resides in the system.

339. Pre-conditions

339.1 **< Student's account in available in system>**

To be able to delete student's account from the system, the student's profile must be available in the system and either active or inactive.

339.2 **<Student's account deletion has been approved >**

The student's account deletion must be approved by appropriate parties before beginning the account deletion process.

340. Post-conditions

340.1 **< Student's account is deleted from the system >**

After instructing the system to delete the student's profile, the student is removed from the system.

341. Extension Points

341.1 **<None >**

Elaboration

Use Case Specification: <ReadStudent>

Version <1.0>

Revision History

Date	Version	Description	Author
<03/18/2016>	<1.0>	ReadStudent	<William Benedict>

Table of Contents

1. ReadStudent **306**

1.1 Brief Description 306

2. Flow of Events **306**

2.1 Basic Flow 306

2.2 Alternative Flows 306

2.2.1 <No or Some Permission Level to access student information> 306

3. Special Requirements **306**

3.1 <Protected Information> 306

4. Pre-conditions **306**

4.1 < Login and Authorization> 306

4.2 < Student Existence > 307

5. Post-conditions **307**

5.1 < Change in State > 307

6. Extension Points **307**

Use Case Specification: <ReadStudent>

342. ReadStudent

342.1 Brief Description

The purpose of this use case is to illustrate how an actor can view details about a student that has previously been created.

343. Flow of Events

343.1 Basic Flow

- The actor navigates to the list of students
- The actor selects a student they wish to view the information of
- The actor selects view student details.
- The actor can view details regarding the student that had been previously entered in during the creation process

343.2 Alternative Flows

343.2.1 <No or Some Permission Level to access student information>

- An actor, with insufficient permissions, may be unable to access the list of students and;
- An actor, with some permissions, may be able to access only very basic information.

344. Special Requirements

344.1 <Protected Information>

Some of the information regarding students may be protected under FERPA, Therefore, only people who are on a need to know basis can see certain details regarding a student.

345. Pre-conditions

345.1 < Login and Authorization>

An actor must be signed in and have permissions to access student data.

345.2 **< Student Existence >**

The student must have already been created to be read.

346. Post-conditions

346.1 **< Change in State >**

During the read process, there is no change in student data. Therefore, there is no change in state.

347. Extension Points

Elaboration

Use Case Specification: <UpdateStudent>

Version <1.0>

Revision History

Date	Version	Description	Author
03/04/2016	<1.0>	UpdateStudent	Chad Tracy

Table of Contents

1. UpdateStudent **310**

1.1 Brief Description 310

2. Flow of Events **310**

2.1 Basic Flow 310

2.2 Alternative Flows 310

2.2.1 < System Error: Student Record Not Updated > 310

2.2.2 < User Credentials Are Rejected > 310

3. Special Requirements **310**

3.1 < System Must Recognize Admins > 310

4. Pre-conditions **311**

4.1 < Student Record Must Exist > 311

5. Post-conditions **311**

5.1 < Student Record is Updated > 311

Use Case Specification: <UpdateStudent>

348. UpdateStudent

348.1 **Brief Description**

Allows the admin to enter system and update student records with necessary information. Information includes a student's name, address, contact information, status in the system, and another information the user decides is necessary to update student records with.

349. Flow of Events

349.1 **Basic Flow**

- User requests access to system (username and password)
- System approves user credentials and grants them admin privileges
- Admin accesses student data and makes appropriate changes
- Admin saves changes made to student records
- System saves any changes made by the Admin
- Updated student records are updated and displayed to those who have access to them within the system

349.2 **Alternative Flows**

349.2.1 < System Error: Student Record Not Updated >

- Admin makes changes to student record in the system and requests the system save changes made
- System fails to save changes made by admin
- Student record is not updated
- System prompts admin to reenter data

349.2.2 < User Credentials Are Rejected >

- User attempts to log-in to system with credentials (username and password)
- System rejects user and they are returned to log-in screen
- Failure for the user to log-in can come from the system lacking updated credential information or the user not entering their credentials correct

350. Special Requirements

350.1 **< System Must Recognize Admins >**

The correct users must be recognized as admins within the system. This allows the admins to navigate through the system and make updates to the Student record

351. Pre-conditions

351.1 **< Student Record Must Exist >**

In order for the admin to make updates to the student record within the system there must be a student record that already exists

352. Post-conditions

352.1 **< Student Record is Updated >**

After the flow is completed the student record should be up to date with the information the admin submitted

Elaboration

Use Case Specification: <CreateUserAccount>

Version 1.0

Revision History

Date	Version	Description	Author
02/20/2016	1.0	CreateUserAccount	Maam Awa Ndaw

Table of Contents

1. CreateUserAccount 337

1.1 Brief Description 337

2. Flow of Events 337

2.1 Basic Flow 337

2.2 Alternative Flows 337

2.2.1 < User Account Creation is Denied > 337

2.2.2 < Unsuccessful Account Creation > 337

3. Special Requirements 338

3.1 < An identifier, such as Account ID must be assigned to the account profile> 338

4. Pre-conditions 338

4.1 <Account Sign up form Available> 338

4.2 < Account Creation is requested> 338

5. Post-conditions 338

5.1 < A New Account Profile > 338

6. Extension Points 338

6.1 <Edit Profile> 338

Use Case Specification: <CreateUserAccount>

353. CreateUserAccount

353.1 Brief Description

This use case describes how a user such as a parent, a volunteer, a donor can create an account. When the user fills out the account sign up form, a request is sent to the system administrator and subsequently their profile is created upon approval.

354. Flow of Events

354.1 Basic Flow

The basic flow of events for creating a user account is as follow:

- User visits Adelante Website and signs up for an account.
- User fills out sign up form with their personal information such as name, address, type of user, interest
- System records and sends user's information to Admin
- Admin reviews the information and accepts the request.
- A confirmation of account creation is subsequently created
- User's records reside within the system.
- User's account becomes active.

354.2 Alternative Flows

354.2.1 < User Account Creation is Denied >

There could be a possibility that the user is denied account creation. In that case, it is possible admin could deny the account creation because because of whatever reason such as failure in verification process. The user's profile will not be created then unless Admin approves the account creation.

354.2.2 < Unsuccessful Account Creation >

There could be a scenario where the user tries to create an account but the system is not storing user input or some other technical issue within the system is preventing the user from creating an account. In that case, the user will not be able to create an account until the issue is fixed.

355. Special Requirements

355.1 < An identifier, such as Account ID must be assigned to the account profile>

Each time a new account profile is created, a unique identifier such as an ID must be assigned to a new record associated with the account profile. ID will be essentially used to access the record within the system.

356. Pre-conditions

356.1 <Account Sign up form Available>

The user must be able to sign up for an account, hence the account sign up form must be available and in standby mode within the system.

356.2 < Account Creation is requested>

The user has to submit a filled out sign up form to the admin in order to create a request for an account creation.

357. Post-conditions

357.1 < A New Account Profile >

Upon account creation, a new profile is created and assigned to the user who requested it. Any future information about that user will be stored in their profile.

358. Extension Points

358.1 <Edit Profile>

It is likely that information stored about the user will change in the future such as phone number or address. Admin or user will then edit the records and make changes accordingly.

Elaboration

Use Case Specification: <DeleteUserAccount>

Version <1.0>

Revision History

Date	Version	Description	Author
<03/18/2016>	<1.0>	DeleteUserAccount	<William Benedict>

Table of Contents

1. DeleteUserAccount **342**

1.1 Brief Description 342

2. Flow of Events **342**

2.1 Basic Flow 342

2.2 Alternative Flows 342

2.2.1 < Lack of permission to delete flow > 342

2.2.2 < Permission to delete own account > 342

3. Special Requirements **342**

3.1 < Account Deletion standards> 342

4. Pre-conditions **343**

4.1 < Login & Authentication Requirement> 343

4.2 < Account Existence Requirement > 343

5. Post-conditions **343**

5.1 < Account State > 343

6. Extension Points **343**

6.1 <Cancel> 343

Use Case Specification: <DeleteUserAccount>

359. DeleteUserAccount

359.1 Brief Description

The DeleteUserAccount use case illustrates functionality regarding the removal of User Accounts from the Account records. Only users with access to the editing mechanics of account information, as well as account holders will be authorized to delete accounts. An actor, being an Adelante staff member, may delete an account or an actor, being an account holder, may delete his or her own account.

360. Flow of Events

360.1 Basic Flow

- The actor loads the list of accounts or the individual account page
- The actor selects account(s) to delete
- The actor confirms the deletion using their current password.

360.2 Alternative Flows

360.2.1 < Lack of permission to delete flow >

- The actor is displayed a message indicating that they do not possess the rights to delete a specific account.
- The actor is directed back to the list of accounts or the individual's account page

360.2.2 < Permission to delete own account >

- The actor, being the account owner, has the ability to delete their own account
- The actor is displayed a confirmation message asking for their password.
- Upon correct entry of password, a final confirmation message is sent to actor.

361. Special Requirements

361.1 < Account Deletion standards>

Most organizations allow users to delete their own account at any point and all records of their accounts are removed. This process was included in the alternative flow for that reason.

362. Pre-conditions

362.1 **< Login & Authentication Requirement>**

The actor must be logged in and have a level of permission sufficient to view and delete accounts.

The actor will always have the right to delete their own account at any time after a second authentication.

362.2 **< Account Existence Requirement >**

There must be an account already created to delete

363. Post-conditions

363.1 **< Account State >**

The account can be either deleted or not deleted.

364. Extension Points

364.1 **<Cancel>**

At any point, the actor can cancel deletion until the account is actually deleted.

Elaboration

Use Case Specification: <ReadUserAccount>

Version <1.0>

Revision History

Date	Version	Description	Author
<03/05/2016>	<1.0>	ReadUserAccount	Chad Tracy

Table of Contents

1. ReadUserAccount **347**

1.1 Brief Description 347

2. Flow of Events **347**

2.1 Basic Flow 347

2.2 Alternative Flows 347

2.2.1 < Access Error> 347

2.2.2 < Read Account Error > 347

3. Pre-conditions **347**

3.1 < Account Must Exist Within the System> 347

3.2 < Credential Authentication > 347

4. Post-conditions **348**

4.1 < Actor is Able to Read Account> 348

Use Case Specification: <ReadUserAccount>

365. ReadUserAccount

365.1 **Brief Description**

Allows the actor to view a pre-existing use case within the system.

366. Flow of Events

366.1 **Basic Flow**

- Actor requests access to AHA system with login credentials
- AHA system grants access to actor
- Actor requests to view an account from the system
- System returns the requested account to actor

366.2 **Alternative Flows**

366.2.1 < Access Error>

- The actor is for some reason not authorized to view the requested account
- The system will deny them access
- System returns actor with a message

366.2.2 < Read Account Error >

- The actor requests to read an account
- The account is unable to be read
- System returns actor with an error message

367. Pre-conditions

367.1 **< Account Must Exist Within the System>**

The account the actor requests must already exist within the system

367.2 **< Credential Authentication >**

The system must approve that the actor is allowed to enter the system and read the requested account based on their login credentials

368. Post-conditions

368.1 **< Actor is Able to Read Account>**

After the process is completed the system will have returned the actor with the requested account to read

Elaboration

Use Case Specification: <UpdateUserAccount>

Version 1.0

Revision History

Date	Version	Description	Author
02/20/2016	1.0	UpdateUserAccount	Valencia Richard

Table of Contents

1. UpdateUserAccount **351**

1.1. Brief Description 351

2. Flow of Events **351**

2.1. Basic Flow 351

2.2. Alternative Flows **351**

2.2.1. < User Account Not Created > 351

2.2.2. < Update Not Approved> 352

3. Special Requirements **352**

3.1. < Valid Login > 352

4. Pre-conditions **352**

4.1. < User Account Exist> 352

4.2. < Login Credentials Verified> 352

5. Post-conditions **353**

5.1. < User Account Updated > 353

6. Extension Points **353**

6.1. < Cancel> 353

Use Case Specification: <UpdateUserAccount>

1. UpdateUserAccount

1.1. Brief Description

This use case allows an actor to enter the system and update an existing User Account. Demographic and educational details can be edited with proper credentials. User Account information to be updated includes Account type, contact details, name, address, etc.

2. Flow of Events

2.1. Basic Flow

- The Actor initiates the Use Case by accessing the AHA system
- The actor inputs their login information
- The system accepts the information
- The actor locates the appropriate User Account to be Updated
- The actor updates the User Account
- The system accesses the credentials for proper clearance
- Clearance level is high enough to make Updates
- The system updates the Account

2.2. Alternative Flows

2.2.1. < User Account Not Created >

- The Actor initiates the Use Case by accessing the AHA system
- The actor inputs their login information
- The system accepts the information
- The actor cannot locate the specific UserAccount

2.2.2. < Update Not Approved>

- The Actor initiates the Use Case by accessing the AHA system
- The actor inputs their login information
- The system accepts the information
- The actor locates the appropriate User Account to be Updated
- The actor updates the User Account
- The system accesses the credentials for proper clearance
- Clearance level is not enough to make Updates
- The system does not update the User Account
- The system alerts the user that the User Account has not been updated

3. Special Requirements

3.1. < **Valid Login** >

An actor must have the clearance to go into the system and make change to Account. This power will be held by only the top staff at Adalante. When any other staff that try to make changes that do not have clearance, their update request will be denied and they will need to contact Head Administration

4. Pre-conditions

4.1. < **User Account Exist**>

In order to update a specific User Account, that Account must exist within the system. If the Account does not exist, then it cannot be updated and it must be created

4.2. < **Login Credentials Verified**>

An actor must have the proper login and password to access the system to make the proposed changes. Not everyone on staff will be able to make User Account updates

5. Post-conditions

5.1. < **User Account Updated** >

The User Account will be updated with the appropriate information and will be available for viewing to those with access

6. Extension Points

6.1. < **Cancel**>

An actor has the option of canceling the update before the final details are submitted, to be approved or rejected, in which case the User Account will not be updated.

<Elaboration >

Use Case Specification: <CreateVolunteer>

Version <1.0>

Revision History

Date	Version	Description	Author
03/03/2016	1.0	<CreateVolunteer>	Chad Tracy

Table of Contents

1. CreateVolunteer **356**

1.1 Brief Description 356

2. Flow of Events **356**

2.1 Basic Flow 356

2.2 Alternative Flows 356

2.2.1 < Invalid Information > 356

2.2.2 <Data Isn't Approved by System > 356

3. Pre-conditions **356**

3.1 < Volunteer Creation Tool Must Exist > 356

3.2 < Area to Store Volunteers in System Must Exist> 357

4. Post-conditions **357**

4.1 < Volunteer is Created Within the System > 357

5. Extension Points **357**

5.1 <Cancel> 357

5.2 <Reject> 357

Use Case Specification: <CreateVolunteer>

369. CreateVolunteer

369.1 Brief Description

Allows the admin to create a record to store information about the admin in. Information about the volunteer will include, but is not limited to: name, address, date registered, contact information, username, and password.

This record within the system will also be used in order to track volunteers for events attended.

370. Flow of Events

370.1 Basic Flow

- Volunteer accesses system and navigates to the page where they will request to create an account
- System grants them access and prompts the volunteer to enter information that will be stored into their volunteer record
- Volunteer enters necessary information and submits it to the system
- System approves the submission and stores data in system
- Admin logs-in to system
- System grants access to admin
- Admin accesses volunteer records and approves them
- After approval the system stores the information as a record and the user is recognized as a 'volunteer'

370.2 Alternative Flows

370.2.1 < Invalid Information >

- Volunteer enters information into system
- Information is rejected by system (incorrect format, information already attached to another account, etc)
- System prompts user to reenter information

370.2.2 <Data Isn't Approved by System >

- Volunteer enters information into system
- System error occurs. Data input by user is not saved
- Volunteer is prompted to reenter information

371. Pre-conditions

371.1 < Volunteer Creation Tool Must Exist >

There must be a method in place within the system that allows users to enter information and request to become volunteers

371.2 **< Area to Store Volunteers in System Must Exist>**

There must be a place within the system that is designated to storing volunteer information

Admins must also have access to this in order to approve volunteers

372. Post-conditions

372.1 **< Volunteer is Created Within the System >**

Volunteer is created and recognized in the system as such

373. Extension Points

373.1 **<Cancel>**

At any point the Volunteer is able to cancel their application of data to become a volunteer

373.2 **<Reject>**

The admin is allowed to reject the volunteer's submission, thus rejecting the user of being a volunteer and the system won't create a Volunteer account

Elaboration

Use Case Specification: <DeleteVolunteer>

Version <1.0>

Revision History

Date	Version	Description	Author
<03/18/2016>	<1.0>	DeleteVolunteer	<William Benedict>

Table of Contents

1. DeleteVolunteer **360**

1.1 Brief Description 360

2. Flow of Events **360**

2.1 Basic Flow 360

2.2 Alternative Flows 360

2.2.1 <No Permission to Delete> 360

2.2.2 <No selected volunteers> 360

3. Special Requirements **360**

4. Pre-conditions **360**

4.1 <Login & Authorization> 360

4.2 < Volunteer Existence > 361

5. Post-conditions **361**

5.1 < Volunteer Status > 361

6. Extension Points **361**

6.1 <Cancel> 361

Use Case Specification: <DeleteVolunteer>

374. DeleteVolunteer

374.1 Brief Description

This use case illustrates how an actor can remove a volunteer from the list of volunteers. Some actors may only have permission to read volunteers, however, that does not mean they will have the permission to delete volunteers,

375. Flow of Events

375.1 Basic Flow

- The actor navigates to the volunteer management page
- The actor selects from a list of volunteers who they wish to remove
- The actor is prompted wish a list of names and an "Are you sure you wish to delete these volunteers" prompt
- The actor is prompted to input their password to confirm deletion

375.2 Alternative Flows

375.2.1 <No Permission to Delete>

- The actor does not have the permission level high enough to delete a volunteer.
- The actor is prompted back to the list of volunteers

375.2.2 <No selected volunteers>

- The actor does not select any volunteers to remove
- The actor is prompted by a message saying no volunteers selected for removal.
- The actor is redirected back to the volunteer list

376. Special Requirements

377. Pre-conditions

377.1 <Login & Authorization>

The actor must be signed in and have a permission level high enough to delete volunteers.

377.2 **< Volunteer Existence >**

A volunteer must have been previously created and recorded in order to delete.

378. Post-conditions

378.1 **< Volunteer Status >**

The volunteer is either deleted or not deleted

379. Extension Points

379.1 **<Cancel>**

At any point during the deletion process, the actor may stop the process at any time until the volunteer is actually deleted

Elaboration

Use Case Specification: <ReadVolunteer>

Version 1.0

Revision History

Date	Version	Description	Author
02/29/2016	1.0	ReadVolunteer	Maam Awa Ndaw

Table of Contents

1. ReadVolunteer **364**

1.1 Brief Description 364

2. Flow of Events **364**

2.1 Basic Flow 364

2.2 Alternative Flows 364

2.2.1 < User doesn't have appropriate permission to read volunteer's information > 364

2.2.2 < Volunteer's information isn't available for viewing > 364

3. Special Requirements **365**

3.1 < Volunteer's profile must exist in system > 365

4. Pre-conditions **365**

4.1 < User was granted access to volunteer's profile > 365

4.2 < Volunteer's profile exists in system and available for viewing > 365

5. Post-conditions **365**

5.1 < System displays volunteer's information for viewing > 365

6. Extension Points **365**

6.1 < Volunteer information updated > 365

Use Case Specification: <ReadVolunteer>

380. ReadVolunteer

380.1 Brief Description

This use case describes how Admin, volunteer or a user with access can read or view volunteer's information

381. Flow of Events

381.1 Basic Flow

The basic flow of events for reading or viewing a specific volunteer's information is as follow:

- Admin or user enters username and password
- System authenticates user
- System grants access to user
- Admin or user navigates to volunteer's profile
- Admin or user reads volunteer's information

381.2 Alternative Flows

381.2.1 < User doesn't have appropriate permission to read volunteer's information >

If a user doesn't have appropriate access to volunteer's profile and attempts to read volunteer's information, access would be denied. Access to volunteer's information would have to be granted prior to reading volunteer's information; a normal flow of events for viewing volunteer's information would then be possible.

381.2.2 < Volunteer's information isn't available for viewing >

A user could have appropriate access to volunteer's information but due to system error, volunteer's information isn't available for reading. In this scenario, the error would have to be fixed before a normal flow of events for reading volunteer's information can occur.

382. Special Requirements

382.1 **< Volunteer's profile must exist in system >**

For a user with granted access to be able to read volunteer's information, volunteer's profile must exist in the system and accessible for viewing.

383. Pre-conditions

383.1 **< User was granted access to volunteer's profile >**

A user must be granted access to the volunteer's profile to be able to read the information about a specific volunteer.

383.2 **< Volunteer's profile exists in system and available for viewing >**

The volunteer's profile must have been previously created in the system and available for viewing.

384. Post-conditions

384.1 **< System displays volunteer's information for viewing >**

User can read information about a volunteer.

385. Extension Points

385.1 **< Volunteer information updated >**

A user with appropriate access to volunteer information could decide to update information about volunteer such as address, phone number.

Elaboration

Use Case Specification: <UpdateVolunteer>

Version 1.0

Revision History

Date	Version	Description	Author
02/20/2016	1.0	UpdateVolunteer	Valencia Richard

Table of Contents

1. UpdateVolunteer **5**

1.1. Brief Description 5

2. Flow of Events **5**

2.1. Basic Flow 5

2.2. Alternative Flows **5**

2.2.1. < Volunteer Not Created > 5

2.2.2. < Update Not Approved> 6

3. Special Requirements **6**

3.1. < Valid Login > 6

4. Pre-conditions **6**

4.1. < Volunteer Must Exist> 6

4.2. < Login Credentials Verified> 6

5. Post-conditions **7**

5.1. < Volunteer Updated > 7

6. Extension Points **7**

6.1. < Cancel> 7

Use Case Specification: <UpdateVolunteer>

1. UpdateVolunteer

1.1. Brief Description

This particular use case allows an actor to go into the Adalnte systen and make updats to Volunteer information. The volunteer must exist already to be updated. Volunteer information to be updated include names, contact details, and hours volunteered

2. Flow of Events

2.1. Basic Flow

- The Actor initiates the Use Case by accessing the AHA system
- The actor inputs their login information
- The system accepts the information
- The actor locates the appropriate Volunteer to be Updated
- The actor updates the Volunteer
- The system accesses the credentials for proper clearance
- Clearance level is high enough to make Updates
- The system updates the Volunteer

2.2. Alternative Flows

2.2.1. < Volunteer Not Created >

- The Actor initiates the Use Case by accessing the AHA system
- The actor inputs their login information
- The system accepts the information
- The actor locates the appropriate User Account to be Updated

2.2.2. < Update Not Approved>

- The Actor initiates the Use Case by accessing the AHA system
- The actor inputs their login information
- The system accepts the information
- The actor locates the appropriate User Account to be Updated
- The actor updates the User Account
- The system accesses the credentials for proper clearance
- Clearance level is not enough to make Updates
- The system does not update the Volunteer
- The system alerts the user that the Volunteer has not been updated

3. Special Requirements

3.1. < **Valid Login** >

An actor must have the clearance to go into the system and make updates to any specific volunteer. Only the director and the assistant director roles will have this authority. When any other staff that try to make changes that do not have clearance, their update will not go through

4. Pre-conditions

4.1. < **Volunteer Must Exist**>

In order to update a specific Volunteer, that Account must exist within the system. If the Account does not exist, then additional research will have to be conducted to determine if it was deleted or has not yet been created

4.2. < **Login Credentials Verified**>

An actor must have the proper login and password to access the system to make updates to Volunteer

5. Post-conditions

5.1. **< Volunteer Updated >**

The volunteer will be updated with the information that the user inputted. Upon proper receipt of these updates they will be available to those who have permission to Read Volunteers

6. Extension Points

6.1. **< Cancel>**

An actor can cancel the attempt to Update a Volunteer before submitting to proposed Updates

Appendix B: Trace Matrix

	Grant Database	Donation Database	Spending Database	Volunteer Page	Mass Email Page	Calender Page	Event Page	Donation Page	Volunteer Signup Page	Admin	Archives	Backups
1. CreateUserAccount									X	X		
2. ReadUserAccount										X		
3. UpdateUserAccount									X	X		
4. DeleteUserAccount										X		
5. CreateDate									X			
6. ReadDate												
7. UpdateDate												
8. DeleteDate												
9. PostEvent												
10.Alert						X						
11. Draw						X	x		x			
12. PayPalRedirect								x				
13. RSVP									x			

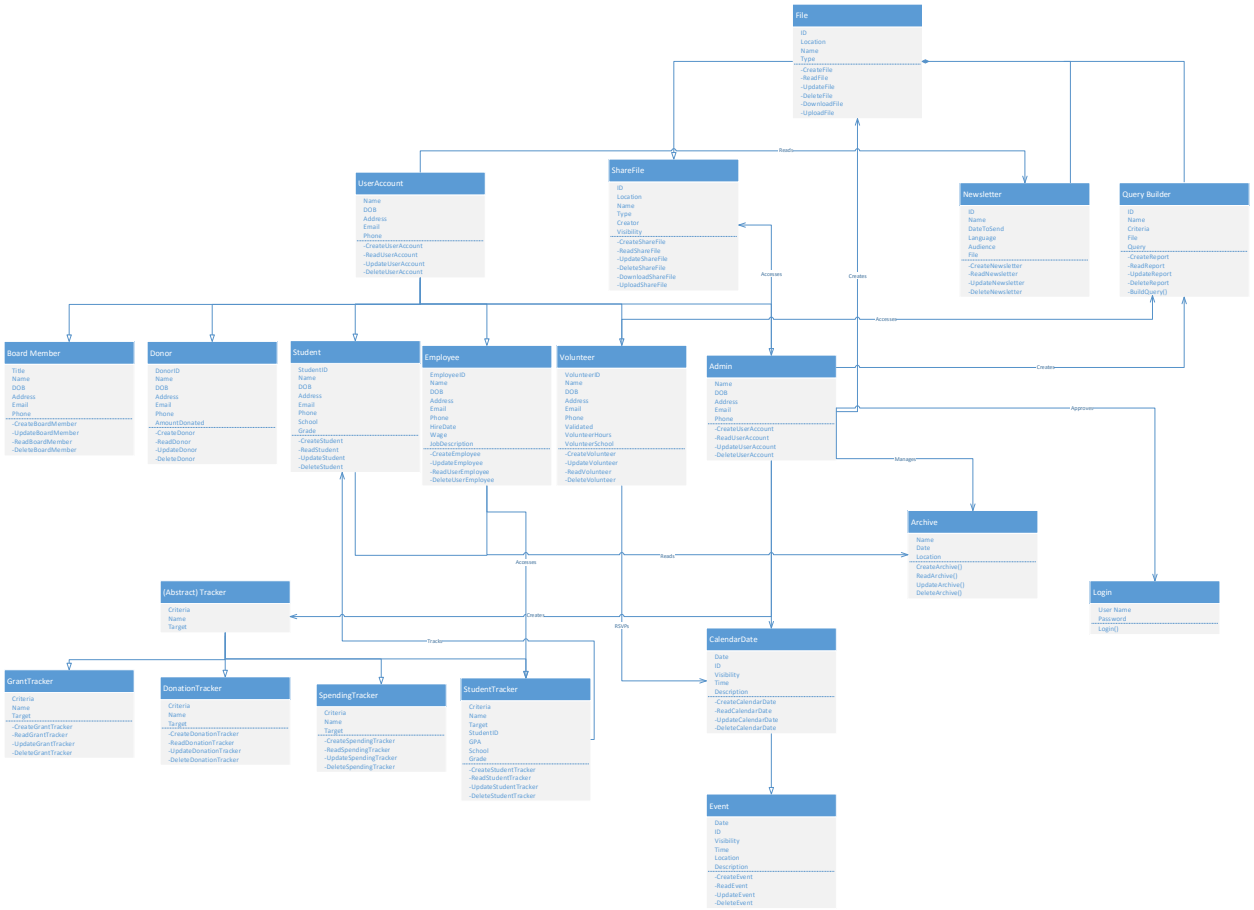
Sheet1

	Grant Database	Donation Database	Spending Database	Volunteer Page	Mass Email Page	Calender Page	Event Page	Donation Page	Volunteer Signup Page	Admin	Archives	Backups
13. RSVP									x			
14. CreateNewsletter												
15. ReadNewsletter												
16. UpdateNewsletter												
17. DeleteNewsletter												
18. CreateFile	X											
19. ReadFile	X											
20. UpdateFile	X											
21. DeleteFile	X											
22. UploadFile												
23. DownloadFile	X											
24.CreateQuery												
25. CreateTracker	X	X	X									

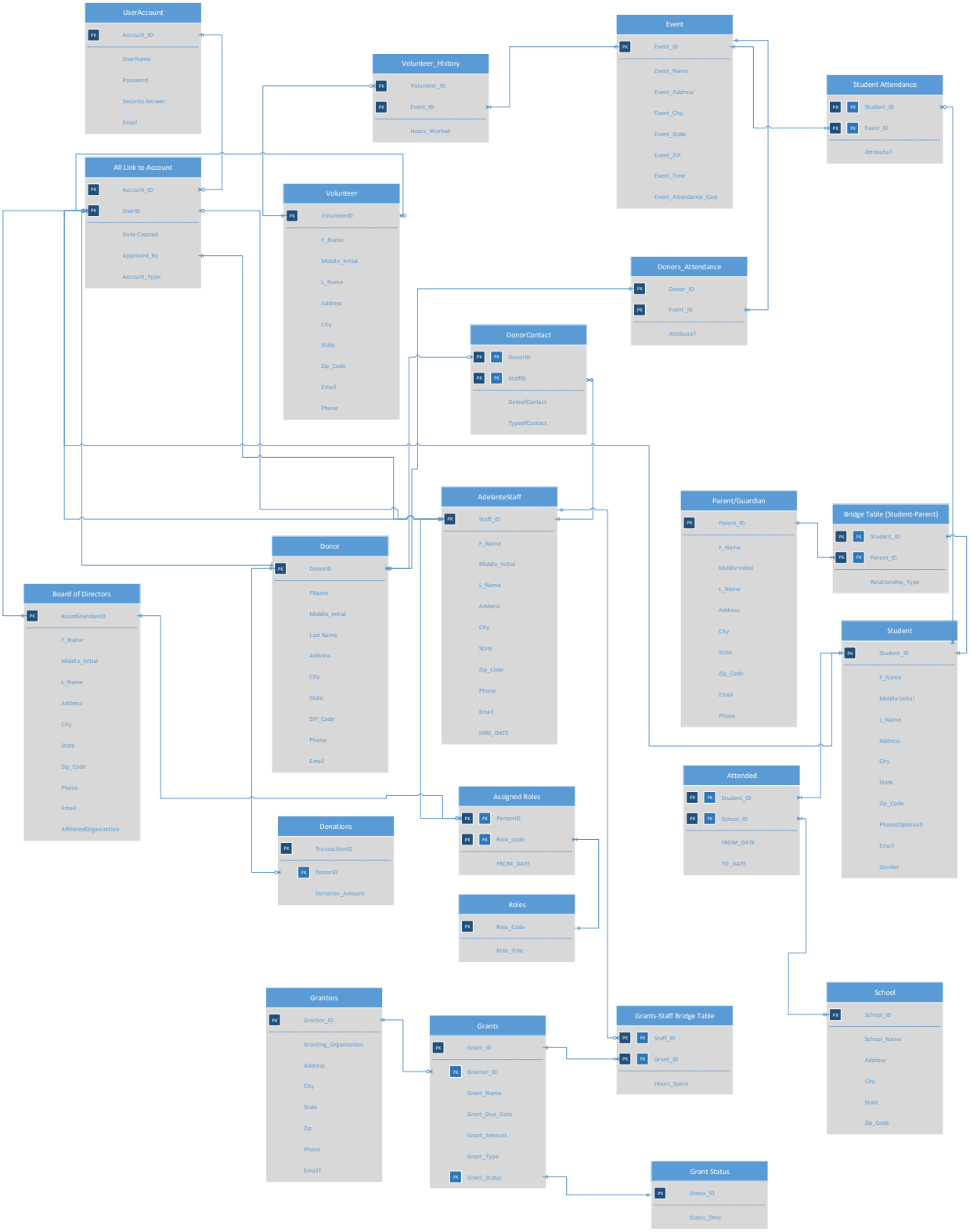
Sheet1

	Grant Database	Donation Database	Spending Database	Volunteer Page	Mass Email Page	Calender Page	Event Page	Donation Page	Volunteer Signup Page	Admin	Archives	Backups
26. ReadTracker	X	X	X									
27. UpdateTracker	X	X	X									
28. DeleteTracker	X	X	X									
29. CreateRecovery	X	x	X	X	X	X	X	X	X	X	X	X
30. ReadRecovery										X	X	X
31. UpdateRecovery	X	X	X	X	X	X	X	X	X	X	X	X
32. DeleteRecovery	X	X	X	X	X	X	X	X	X	X	X	X
33. AssignPermissions										X		

Appendix C: Class Diagram



25. Adalante ERD


UserAccount
PK Account_ID
UserName
Password
Security Answer
Email
Volunteer_History
PK Volunteer_ID
PK Event_ID
Hours_Worked
Event
PK Event_ID
Event_Name
Event_Address
Event_City
Event_State
Event_ZIP
Event_Time
Event_Attendance_Cost
Student Attendance
PK FK Student_ID
PK FK Event_ID
Attribute?
All Link to Account
PK Account_ID
PK UserID
Date-Created
Approved_By
Account_Type
Volunteer
PK VolunteerID
F_Name
Middle_Initial
L_Name
Address
City
State
Zip_Code
Email
Phone
Donors_Attendance
PK Donor_ID
PK Event_ID
Attribute?
DonorContact
PK FK DonorID
PK FK StaffID
DateofContact
TypeofContact
AdelanteStaff
PK Staff_ID
F_Name
Middle_Initial
L_Name
Address
City
State
Zip_Code
Phone
Email
HIRE_DATE
Parent/Guardian
PK Parent_ID
F_Name
Middle Initial
L_Name
Address
City
State
Zip_Code
Email
Phone
Bridge Table (Student-Parent)
PK FK Student_ID
PK FK Parent_ID
Relationship_Type
Donor
PK DonorID
FName
Middle_Initial
Last Name
Address
City
State
ZIP_Code
Phone
Email
Board of Directors
PK BoardMemberID
F_Name
Middle_Initial
L_Name
Address
City
State
Zip_Code
Phone
Email
AffiliatedOrganization
Student
PK Student_ID
F_Name
Middle Initial
L_Name
Address
City
State
Zip_Code
Phone(Optional)
Email
Gender
Attended
PK FK Student_ID
PK FK School_ID
FROM_DATE
TO_DATE
Assigned Roles
PK FK PersonID
PK FK Role_code
FROM_DATE
Donations
PK TransactionID
FK DonorID
Donation_Amount
Roles
PK Role_Code
Role_Title
Grantors
PK Grantor_ID
Granting_Organization
Address
City
State
Zip
Phone
Email?
Grants
PK Grant_ID
FK Grantor_ID
Grant_Name
Grant_Due_Date
Grant_Amount
Grant_Type
FK Grant_Status
Grants-Staff Bridge Table
PK FK Staff_ID
PK FK Grant_ID
Hours_Spent
School
PK School_ID
School_Name
Address
City
State
Zip_Code
Grant Status
PK Status_ID
Status_Desc

ERD Narrative: An ERD is a graphical way to represent the connections present within Adalante. Classes are connect, showcasing how objects within the system operate and interact with oneanother.

Appendix D: Database Design and Definitions

Appendix D: Database Design and Data Definition

1. **Totals Page**
2. Student
3. Student_Family
4. Parent
5. Attended
6. School
7. All_Link_ to_Account
8. User_Account
9. Volunteer
10. Volunteer_History
11. Event
12. Student_Attendane
13. Donor_Attendence
14. Donors
15. Donations
16. Adalante_Staff
17. Assigned_Roles
18. Roles
19. Grant_Work_Record
20. Grants
21. Grant_Status
22. Grantors
23. Board_of_Directors
24. Donor_Contact
25. ERD

Totals Page

Data Dictionary for Adelante				
Prepared by SAD Associates				
	Summary:			
	Total Number of tables	21		
	Total Number of space required for records:	1625	Bytes	
		or		
		1.625	KiloBytes	
	New Tables:	2		
	Total Space from New Tables	236	Bytes	
	Total Space for New and Old tables	1861	Bytes	
		or		
		1.861	KiloBytes	

Student

	Attribute name	Data Type	Can be Null?	Size in Bytes	Description
PK	Student_ID	Char(10)	No	10	This is the student ID provided to the student by JCPS or their School organization. This could also be a generated student ID number if necessary
	F_Name	Char(20)	No	20	This is the student's first name
	M_Initial	Char(1)	No	1	This is the middle intial of the student
	L_Name	Char(35)	No	35	This is the last name of the student.
	Address	Char(40)	No	40	This is the address of the student
	City	Char(15)	No	15	This is the city of the address
	State	Char(2)	No	2	This is the state of the address
	Zip_Code	Char(9)	No	9	This is the zip code of the address
	Phone	Char(10)	Yes	10	This is the phone number of the student, entirely optional
	Email	Char(30)	No	30	This is the email of the student
	Gender	Char(1)	No	1	This is the gender of the student, M for Male, F for Female
			Total Size:	173	Bytes

Student_Family

Attribute	Data type	Can Be Null	Size In Bytes	Description
Student_ID	Char(10)	No	10	This is the student ID attributed to the student.
Parent_ID	Int(6)	No	4	This is the Parent ID attributed to the student.
Relationship_Type	Char(8)	No	8	This is the relationship of the parent to the student, (Father, Mother, Guardian
		Total:	22	Bytes

Parent

	Attribute	Data Type	Can be Null?	Size in Bytes	Description
PK	Parent_ID	Char(6)	No	6	This is the unique ID generated for each parent
	F_Name	Char(20)	No	20	First Name of the parent
	M_Initial	Char(1)	No	1	Middle Initial of the Parent
	L_Name	Char(35)	No	35	Last name of the Parent
	Address	Char(40)	No	40	Address the Parent recides at
	City	Char(15)	No	15	City of the address
	State	Char(2)	no	2	State of the address (Abbreviated)
	Zip	Char(9)	no	9	Zip code of the address (5 digit or 9 digit)
	Email	Char(30)	no	30	Email of the parent
	Phone	Char(10)	no	10	Phone contact for the parent
			Total Size:	168	Bytes

Attended

		Attribute	Data Type	Can be null?	Size In Bytes	Description
PK	FK	Student_ID	Char(10)	No	10	Student ID belonging to a specific student
PK	FK	School_ID	Char(7)	No	7	School ID belonging to a specific school
		FROM_DATE	Date	No	3	Date student started attending
		TO_DATE	Date	Yes	3	Date when student finished (blank if in progress)
				Total	23	Bytes

School

	Attribute	Data Type	Can be Null?	Size in bytes	Description
PK	School_ID	Char(7)	No	7	This is the unqiue ID generated for the school
	School_Name	Char(40)	No	40	name of the school, i.e. Butler High School, Wilkerson Middle School. Etc.
	Address	Char(40)	No	40	address of the school
	City	Char(15)	No	15	City of the school
	State	Char(2)	No	2	State of the school
	Zip-Code	Char(9)	No	9	Zip code of the school
	Phone	Char(10)	No	10	Phone Number of the school
			Total	123	Bytes

All_Link_ to_Account

		Attribute	Data Type	Can be Null?	Size in Bytes	Description
PK	FK	Account_ID	Char(8)	No	8	Account ID belonging to the account
PK	FK	UserID	VarChar(10)	No	11	ID belonging to the user, such as Student ID, Parent ID, Donor ID and Volunteer ID
		Date_Created	Date	No	3	Date the account was created
		Approved_By	Char(7)	No	7	Staff member who approved the account, Staff ID
		Account_Type	Char(15)	No	15	Type of account, Student, Staff, Donor, Etc.
				Total	44	Bytes
*Notes: UserID used in this table refers to StudentID, StaffID, ParentID, DonorID, and volunteerID						
Approved_By will link to StaffID and will be used by Staff with certain assigned roles.						
Staff with these assigned roles will approve accounts for other staff members.						

User_Account

	Attribute	Data Type	Can be Null	Size in Bytes	Description
PK	Account_ID	Char(8)	No	8	This is the ID generated for the account
	Username	Char(20)	No	20	The name the user uses to sign in
	Password	Char(20)	No	20	The password the user uses to sign in
	Email	Char(30)	No	30	The email linked to the account
			Total	78	Bytes

Volunteer

	Attribute	DataType	Can be Null?	Size in Bytes	Description
PK	Volunteer_ID	Char(5)	No	5	This is the Unique ID generated for the volunteer
	F_Name	Char(20)	No	20	First name of the volunteer
	M_Initial	Char(3)	No	3	Middle initial of the volunteer
	L_Name	Char(35)	No	35	Last name of the volunteer
	Address	Char(40)	No	40	Address where the volunteer recides
	City	Char(15)	No	15	City of the residence
	State	Char(2)	No	2	State of the residence
	Zip_Code	Char(9)	No	9	Zip code of the residence
	Email	Char(30)	No	30	Email contact for the volunteer
	Phone	Char(10)	No	10	Phone contact for the volunteer
			Total	169	Bytes

Volunteer_History

		Attribute	DataType	Can be Null?	Size	Description
PK	FK	Volunteer_ID	Char(5)	No	5	ID linked to a specific volunteer
PK	FK	Event_ID	Char(12)	No	12	ID linked to a specific event
		Hours_Worked	Int(4)	No	4	Number of hours the volunteer worked at a specific event
				Total	21	Bytes

Event

	Attribute	Data-Type	Can be Null?	Size	Description
PK	Event_ID	Char(12)	No	12	This is the Unique ID generated for the event
	Event_Name	Char(30)	No	30	Name of the event
	Attendence Cost	Int(6)	No	4	Cost of an attendee to attend an event, This may be 0 if it is a free event.
	Event Type	Char(15)	No	15	Type of event being hosted, Fundraising, Advertising, Student related, Mentoring, Workshops, Etc.
	Event_Date	DATE	Yes	3	Date the event is planned
	Event_Time	Time	Yes	4	Time the event is planned to begin
			Total	68	Bytes

Student_Attendane

		Attribute	DataType	Can be Null?	Size in Bytes	Description
PK	FK	Student_ID	Char(10)	No	10	ID Belonging to a specific student
PK	FK	Event_ID	Char(12)	No	12	ID Pertaining to a specific event
		Sign_In Time	Time	No	4	Time the student signed in
		Sign_Out Time	Time	No	4	Time the student signed out
				Total	30	Bytes

Donor_Attendence

		Attribute	DataType	Can Be Null?	Size in Bytes	Description
PK	FK	Donor_ID	Char(9)	No	9	ID belonging to a specific donor
PK	FK	Event_ID	Char(12)	No	12	ID pertaining to a specific event
		SignIn_Time	Time	Yes	4	Time the donor arrived, Optional Sign in
		SignOut_Time	Time	Yes	4	Time the donor left, Optional Sign out
				Total:	29	Bytes

Donors

	Attribute	DataType	Can Be Null?	Size in Bytes	Description
PK	DonorID	Char(9)	No	9	This is the unique ID generated for the donor
	F_Name	Char(20)	No	20	Donor's First Name
	M_Initial	Char(1)	No	1	Donor's Middle Initial
	L_Name	Char(35)	No	35	Donor's Last Name
	Address	Char(40)	Yes	40	Donor's Address
	City	Char(15)	Yes	15	City of Address
	State	Char(2)	Yes	2	State of City
	Zip_Code	Char(9)	Yes	9	Zip code of address
	Email	Char(30)	Yes	30	Email contact for Donor
	Phone	Char(10)	Yes	10	Phone contact for Donor
			Total	171	Bytes

Donations

	Attribute	DataType	Can Be Null?	Size in Bytes	Description
PK	Transaction_ID	Char(8)	No	8	ID generated for the transaction
FK	DonorID	Char(9)	No	9	ID belonging to a donor
	Donation_Amount	Decimal(10,2)	No	9	Amount donated
	TransactionID_From_Provider	Char(17)	Yes	17	This is the transaction ID provided by a third party source that links the third party to the
			Total	43	Bytes

Adalante_Staff

	Attribute	DataType	Can be Null?	Size in Bytes	Description
PK	Staff_ID	Char(6)	No	6	This is the Unique Staff ID generated for each staff member
	F_Name	Char(20)	No	20	First name of the staff member
	M_Initial	Char(3)	No	3	Middle initial of the staff member
	L_Name	Char(35)	No	35	Last name of the staff member
	Address	Char(40)	No	40	Address of the staff member
	City	Char(15)	No	15	City of the Address
	State	Char(2)	No	2	State of the City
	Zip_Code	Char(9)	No	9	Zip code of the residence
	Email	Char(30)	No	30	Contact email for the staff member
	Phone	Char(10)	No	10	Phone contact for the staff member
	Hire_Date	Date	No	3	Date the staff member was hired
			Total	173	Bytes

Assigned_Roles

		Attribute	DataType	Can be Null?	Size in Bytes	Description
PK	FK	Staff_ID	Char(6)	No	6	Staff ID for a certain staff member
PK	FK	Role_Code	Char(3)	No	3	Role_Code of the role the staff member belongs to
		FROM_DATE	Date	Yes	3	Date the staff member became assigned the role
				Total	12	Bytes

Roles

	Attribute	DataType	Can be Null?	Size in Bytes	Description
PK	Role_Code	Char(3)	No	3	Unique ID generated for each role
	Role_Title	Char(15)	No	15	Title of each role.
			Total	18	Bytes

Grant_Work_Record

		Attribute	DataType	Can be Null?	Size in Bytes	Description
PK	FK	Staff_ID	Char(6)	No	6	Staff ID of staff member working on grant
PK	FK	Grant_ID	Char(12)	No	12	Grant ID pertaining to grant that is being worked on
		Hours_Spent	Int(4)	Yes	4	Time spent working on grant, May be blank if unknown.
				Total	22	Bytes

Grants

	Attribute	DataType	Can be Null?	Size in Bytes	Description
PK	Grant_ID	Char(12)	No	12	Unique ID generated for each grant
FK	Grantor_Id	Char(6)	No	6	Granting ID pertaining to the granting organization
	Grant_Name	Char(30)	No	30	Name of the grant
	Grant_Due_Date	Date	Yes	3	Date when the grant documents are due
	Grant_Amount	Decimal(10,2)	No	9	Dollar amount of the grant
	Grant_Type	Char(15)	Yes	15	Type of grant (Technology, Mentoring, etc.)
FK	Grant_Status	Char(2)	No	2	Current status of the grant
			Total	77	Bytes

Grant_Status

	Attribute	DataType	Can be Null?	Size	Description
PK	Status_Code	Char(2)	No	2	Unique Code that determines the status of a grant
	Status_Desc	Char(12)	No	12	Description of the status code, (i.e. Complete, In-Progress, Not Started, Approved, Denied)
			Total	14	Bytes

Grantors

	Attribute	DataType	Can be Null?	Size	Description
PK	Grantor_ID	Char(6)	No	6	Unique ID generated for each granting organization
	Granting_Organization	Char(35)	No	35	Name of the granting organization
	Address	Char(40)	No	40	Address for the granting organization
	City	Char(15)	No	15	City of the address
	State	Char(2)	No	2	State of the address
	Zip_Code	Char(9)	No	9	Zip code of the address
	Email	Char(30)	Yes	30	Email of the Granting organization
	Phone	Char(10)	Yes	10	Phone number of the organization.
			Total	147	Bytes

Board_of_Directors

	Attribute	DataType	Can be Null?	Size	Description
PK	BoardMemberID	Char(6)	No	6	Unique ID Generated for the board member
	F_Name	Char(20)	No	20	First Name of the board member
	Middle_Initial	Char(1)	No	1	Middle Initial of the board member
	L_Name	Char(30)	No	30	Last Name of the board member
	Address	Char(30)	No	30	Address of the board member
	City	Char(15)	No	15	City of the address
	State	Char(2)	No	2	State of the City
	Zip	Char(9)	No	9	Zip of the address
	Phone	Char(10)	No	10	Phone Number of the board member
	Email	Char(40)	No	40	Email of the board member
	Affiliated_Organization	Char(35)	Yes	35	Organization the Board member may work for or is affilated through, May be Null
			Total	198	Bytes

Donor_Contact

		Attribute	DataType	Can be Null?	Size in Bytes	Description
PK	FK	DonorID	Char	No	9	Donor ID from Donor table
PK	FK	StaffID	Char(6)	No	6	Staff ID that contacted the donor
		Date	Date	No	3	Date of which the contact was made
		MethodOfContact	Char(20)	Yes	20	This is the method in which the staff member of adelante contacted the donor. Examples are, Phone, In-Person, Email, Letter. ETC

Table Narrative: The Tables contain the data definitions for each potential information bank. For instance, the table labeled Donor Contact will have its attributes and the information that can be entered in each attribute will have to abide by the set parameters, which are the data definitions. These literally define the data.

Appendix E: User Interface Navigation and Screen Layouts

1. **Registration Input**
2. **Registration Output**
3. **Contact Input**
4. **Contact Output**
5. **Event Signup Input**
6. **Event Signup Output**
7. **Create Event Input**
8. **Create Event Output**
9. **Newsletter Signup Input**
10. **Newsletter Signup Output**
11. **Volunteer SignUp**
12. **Volunteer Output**
13. **Login Input**
14. **Login Output**

Registration Input

Registration Output

Contact Input

Contact Output

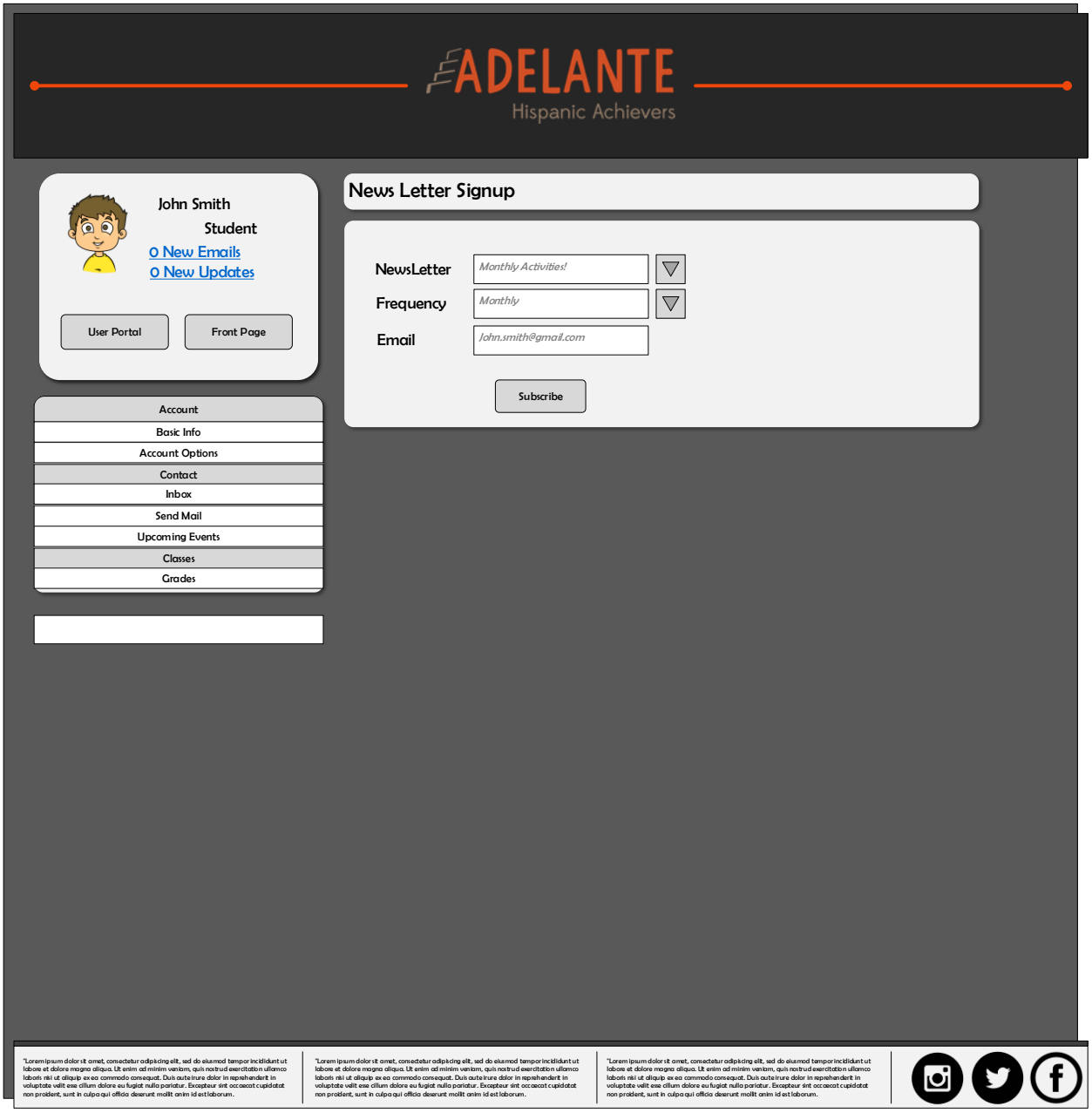
Event Signup Input

Event Signup Output

Create Event Input

Create Event Output

Newsletter Signup Input



Newsletter Signup Output

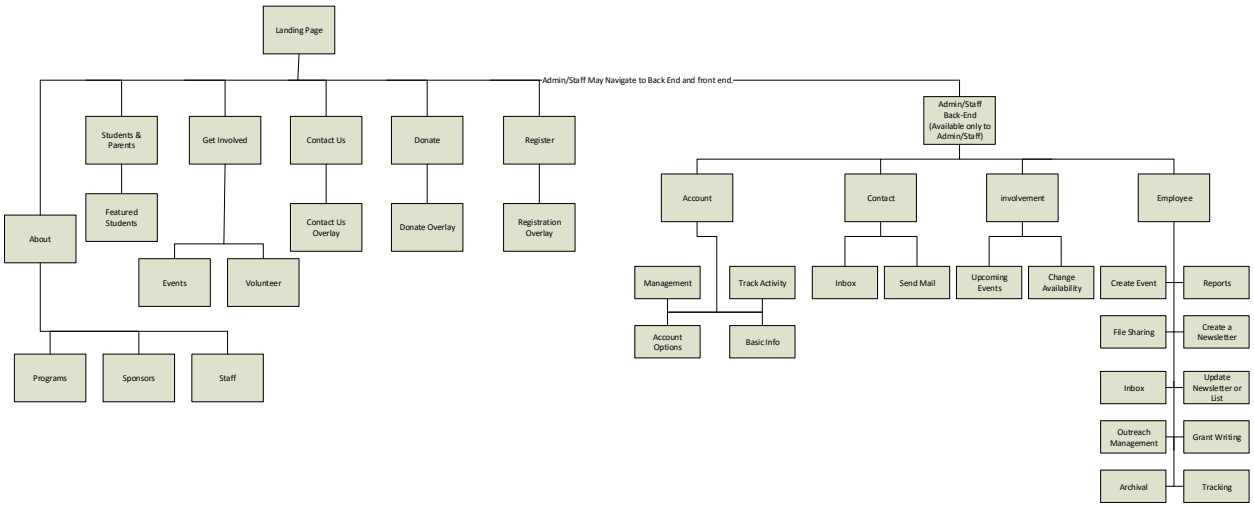
Volunteer Signup

Volunteer Output

Login Input

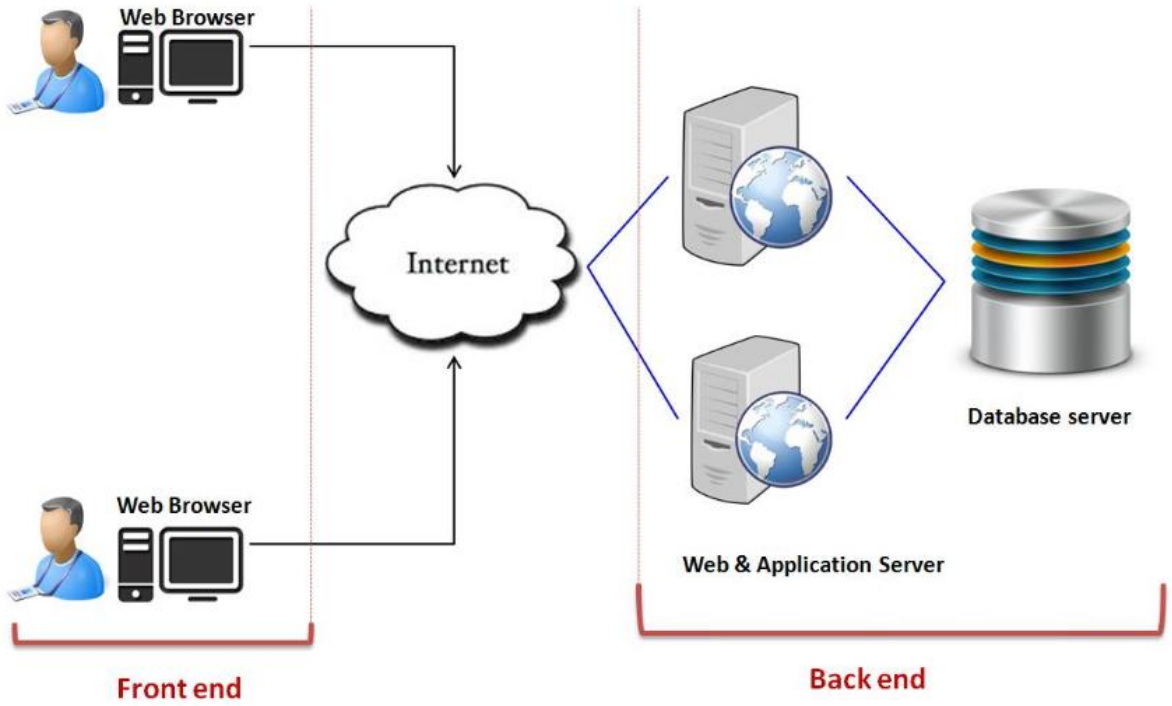
Login
Email Address:
Dustin.Bishop@gmail.com
Password:
Wordpass1
Remember me
Lost your password?
Log In
Don't have an account?
Register

Login Output


Landing Page
Admin/Staff May Navigate to Back End and front end.
Students & Parents
Get Involved
Contact Us
Donate
Register
Admin/Staff Back-End (Available only to Admin/Staff)
Featured Students
Contact Us Overlay
Donate Overlay
Registration Overlay
About
Events
Volunteer
Account
Contact
involvement
Employee
Management
Track Activity
Inbox
Send Mail
Upcoming Events
Change Availability
Create Event
Reports
Account Options
Basic Info
File Sharing
Create a Newsletter
Inbox
Update Newsletter or List
Outreach Management
Grant Writing
Archival
Tracking
Programs
Sponsors
Staff

Appendix F: Physical Architectural Design

Appendix F: Physical Architecture Design



Physical Architecture Design Narrative:

Due to this solution being a cloud based solution. No new critical physical devices will need to be added to the current IT architecture of Adelante. The only hardware requirements for Adelante are devices with internet connection in order to access the website through a web browser. Adelante is currently in possession of these devices, such as their personal computers as well as the Apple IPads. Each of these devices has the ability to connect to the internet.

From that point, a third party, such as HostGator will be hosting the web server for the solution. Adelante will not be able to interact with the hardware itself, however, Adelante will have to maintain the website that is being hosted on the web server.

Adelante's Database server will also be hosted with the third party, HostGator, and this party is responsible for the hardware. The structure of the database itself, however, will be decided upon by Adelante. The Database server and Web server will work with each other and allow for the transmission of data back to Adelante, without Adelante having to interact with the actual database itself, except for future changes or updates to the structure.

Appendix G: Design Procedures for Security Concerns

Appendix G: Design Procedures for Security Concerns and Non-functional Requirements

Design Procedures for Security Concerns and Non-functional Requirements

The following security and protection mechanisms should be considered as the proposed solution will be cloud based computing and thereof all data will be stored and accessed over the internet; in addition to prevention mechanisms for accidental or intentional misuse of the system.

- **Secure Socket Layer-** A standard security technology for establishing an encrypted link between a web server and a browser. It will ensure that all data passed between them will remain private.
- **Access Control-** With each object (person) store information about access privileges, which are designated within the system
- **Authorization-** designate who can perform which operations on which objects. Restricted portions of the site will be locked out via user account authentication. A logged in user will be timed out after a certain period of time. Passwords should be handled in a safe and secure manner. Also, to reduce the likelihood of social engineering, it may be good practice to request that users make use of a mix of different capitalizations, and Unicode symbols in addition to alphanumeric input.
- **Access Enforcement-** Part of the system must be responsible for enforcing access controls and protecting authorization and authentication information.
- **SOX Compliance-** As Adelante is a non-profit organization that reports financial transactions, auditability and confidentiality of such information should be handled accordingly
- **FERPA-** Compliance should be mandatory, especially with the nature of the business and the fact that security with cloud-based systems are critical as confidential data such as student information, emails reside within the system.
- **Backup and recovery capability-** The system should provide reasonable precautions to prevent data loss
- **Intrusion-detection system**- security is an ever-increasing problem in today's internet-based environment, so it is crucial to address possible external infrastructure attacks,

though historically, the greatest security threats has come from internal infrastructure vulnerabilities.

Non-functional requirements specify criteria that can be used to judge the operation of the system, rather than specific behaviors.

- **Operating Requirements**- The physical and technical environments in which the system will operate.

- **Technical Environment**
    - The system will operate over the Web with any browser and users will access the system using through a web browser
    - Portability-The system will need to remain current with the evolving Web standards.
    - Maintainability- No special maintainability requirements are anticipated.

- **Performance Requirements**
    - Speed- Response time should be less than 5 seconds and upload and Download speeds must be acceptable.

    - Capacity- The system will support streaming audio of up to 100 simultaneous users during peak time

    - Availability- The system should be available 24 hours per day, 365 days per year and shall have a 99% uptime performance and sufficient bandwidth

    - Scalability- the system shall increase total throughput under an increased load when resources are added.

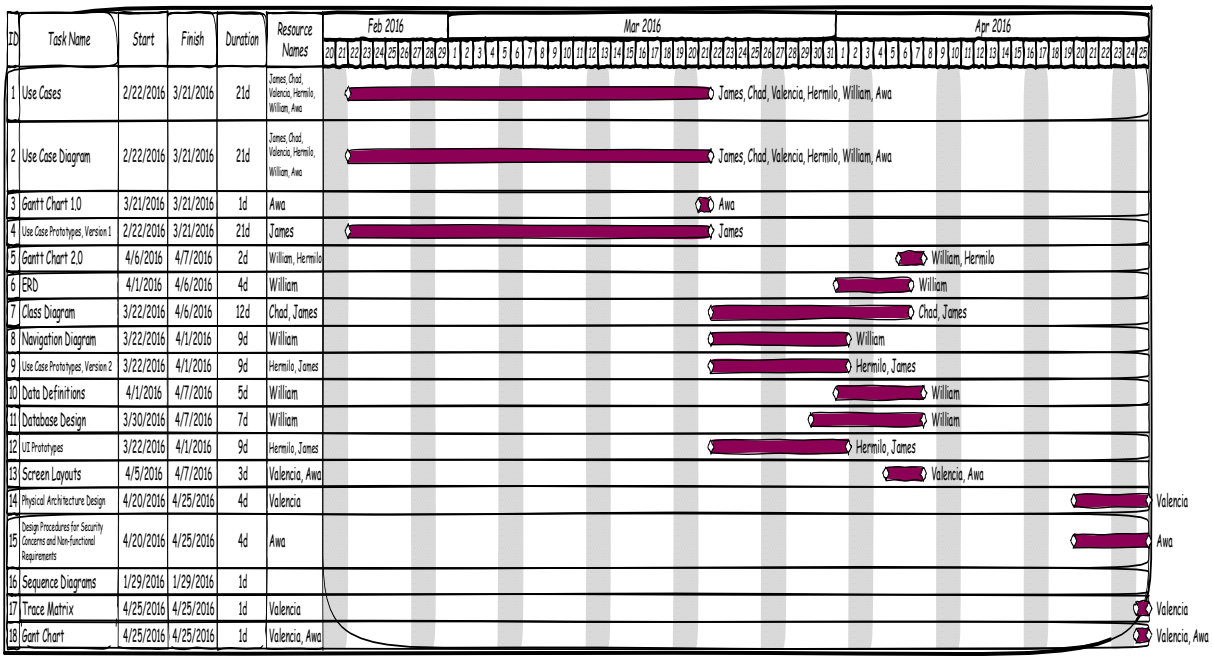
- **Legislative Requirements-** privacy and safety should be addressed during design.

- **Multilingual-** The system shall have Spanish translation capability

- **External Requirements-** Interoperability requirements as Adelante seek donation through its website and donors are redirected to PayPal; and ethical requirements should be addressed as unethical behavior tend to be persistent in non-profits.

Appendix H: Gantt Chart

Gantt chart



ID	Task Name	Start	Finish	Duration	Resource Names
1	Use Cases	2/22/2016	3/21/2016	21d	James, Chad, Valencia, Hermilo, William, Awa
2	Use Case Diagram	2/22/2016	3/21/2016	21d	James, Chad, Valencia, Hermilo, William, Awa
3	Gantt Chart 1.0	3/21/2016	3/21/2016	1d	Awa
4	Use Case Prototypes, Version 1	2/22/2016	3/21/2016	21d	James
5	Gantt Chart 2.0	4/6/2016	4/7/2016	2d	William, Hermilo
6	ERD	4/1/2016	4/6/2016	4d	William
7	Class Diagram	3/22/2016	4/6/2016	12d	Chad, James
8	Navigation Diagram	3/22/2016	4/1/2016	9d	William
9	Use Case Prototypes, Version 2	3/22/2016	4/1/2016	9d	Hermilo, James
10	Data Definitions	4/1/2016	4/7/2016	5d	William
11	Database Design	3/30/2016	4/7/2016	7d	William
12	UI Prototypes	3/22/2016	4/1/2016	9d	Hermilo, James
13	Screen Layouts	4/5/2016	4/7/2016	3d	Valencia, Awa
14	Physical Architecture Design	4/20/2016	4/25/2016	4d	Valencia
15	Design Procedures for Security Concerns and Non-functional Requirements	4/20/2016	4/25/2016	4d	Awa
16	Sequence Diagrams	1/29/2016	1/29/2016	1d	
17	Trace Matrix	4/25/2016	4/25/2016	1d	Valencia
18	Gant Chart	4/25/2016	4/25/2016	1d	Valencia, Awa

Gantt Chart

The Gantt chart is a complete list of tasks that have been divided out for all the iterations, from the beginning to the final Elaboration specification. The tasks each have a unique ID and are arranged chronologically beginning with the first deliverable. Each task also has the date range during which it was worked on along with the main creator(s) of the deliverable. All the tasks are compiled together in an interactive document in a graphical representation. This is useful to hold all group member's accountable and also identify missing requirements.

Appendix I: Elaboration Prototypes

Appendix I: Elaboration Phase Prototypes

1. Home Page
2. About Page
3. About-Programs Pages
4. Get Involved
5. Students and Parents
6. Basic Backup
7. Account Options Back page
8. Track Activity Back page
9. Account Management Back page
10. Inbox Back page
11. Upcoming Events Back page
12. Change Availability Back page
13. Create Event
14. Reports
15. File Sharing
16. Create Newsletter List
17. Send/Update Newsletter
18. Archival
19. Tracking
20. Outreach Management
21. Grants Permission
22. Permissions

1. Home Page

Home: The home page prototype showcases the basic functionality of the front-end pages. They have a post-based content flow, and can have embedded content. We based many of these pages off of Adelante's current webpage content. The bottom of the screen utilizes social media and the theme colors and top, navigation bar, and bottom are all going to be used across the site as a whole. In addition, the front page has placeholders for widgets, and search/login functionality on every page.

2. About Page

About: The about page has many elements from the front page. It inherits login functionality, search functionality, and a common visual theme and navigation menu. Notice that the breadcrumb tree at the top of the page shows your current level on the side, and that it is all organized in a way that uses clean icons, and simple formatting to display the professional side of Adelante. Colors are chosen in order to accentuate the Adelante Logo.

1. **About-Programs Pages**

2. Get Involved

ADELANTE
Hispanic Achievers

Home | About | Students & Parents | Get Involved | Contact | Donate

Get Involved

Get Involved:

Become a Sponsor | March 31, 2016

Adelante operates thanks to our sponsors and volunteers. Adelante is a 501 (c) 3 organization and your tax deductible donation will help provide 135 students and families with Mentoring and Enrichment, Tutoring, and Homework Help, as well as college readiness.

Americorps | Jan 1, 2016

Thanks to the CNCS AmeriCorps Program, Adelante has the support of an AmeriCorps VISTA member for a full year of service from August to August each year. Please apply online to the Corporation for National and Community Service. See AmeriCorps VISTA, advanced search for criteria and online application.

Internships | Jan 1, 2016

Internships are available through: UofL LALS and Bellarmine University School of Education.

LOUISVILLE CARDINALS ™

BELLARMINE UNIVERSITY
IN VERITATIS AMORE

Sample Post | Jan 1, 2016

"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in

culpa qui officia deserunt mollit anim id est laborum.

"Our goal is to guide our students and support their desire, hope, and ambition to be productive citizens in our community" -Stephen Imhoff

SEARCH

Search ... | Submit

Login

Email Address:

YourEmail@email.com

Password:

••••••••

Remember me | Lost your password?

Log In

Don't have an account? | Register

Widget Pane:

Currently Selected Widget Content

1 2 3 4 5 6 7

"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Get Involved: This page is a clean page, inspired by Adelante's current page. On this page we recognize the sponsors and partners that Adelante relies on, or otherwise supports.

3. Students and Parents

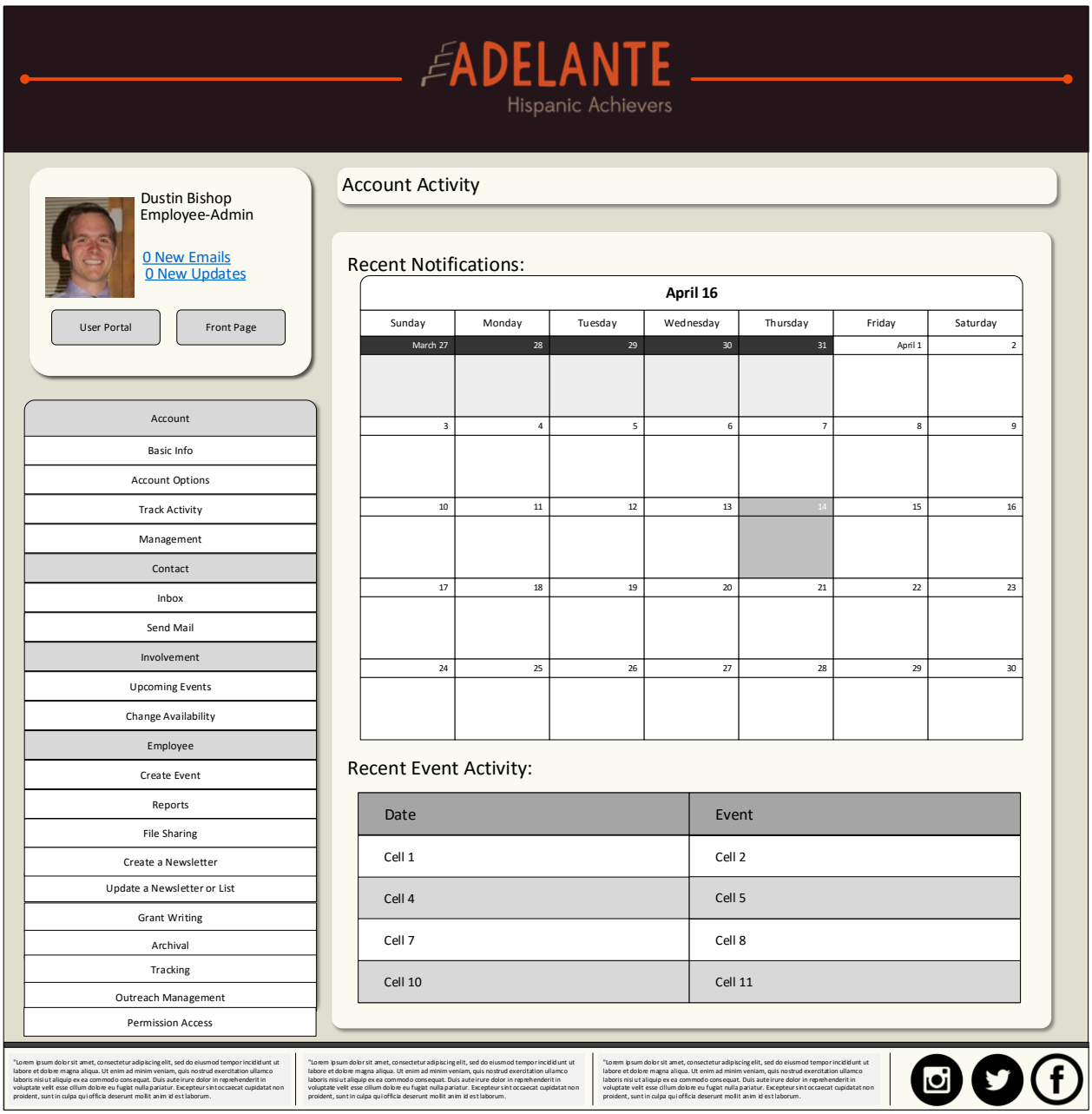
Students & Parents: This page focuses on scholarships and courting a student. On this page we also re-use the file embed with a modern logo, to host the Educational Outlook Report. We want files to be visible, as important documents can be easily seen by most people on the website, and the EER is good at "selling" Adelante to prospective parents.

4. Basic Backup

5. Account Options Back page

Account options: this is a light prototype, and has language options, as well as newsletter and event options. Other than that, it handles nothing else, and is mostly for personalization.

6. Track Activity Back page



* Recent notifications: This uses our calendar system, and tracks events, calendar dates, etc. However, it also shows prior events you've attended.

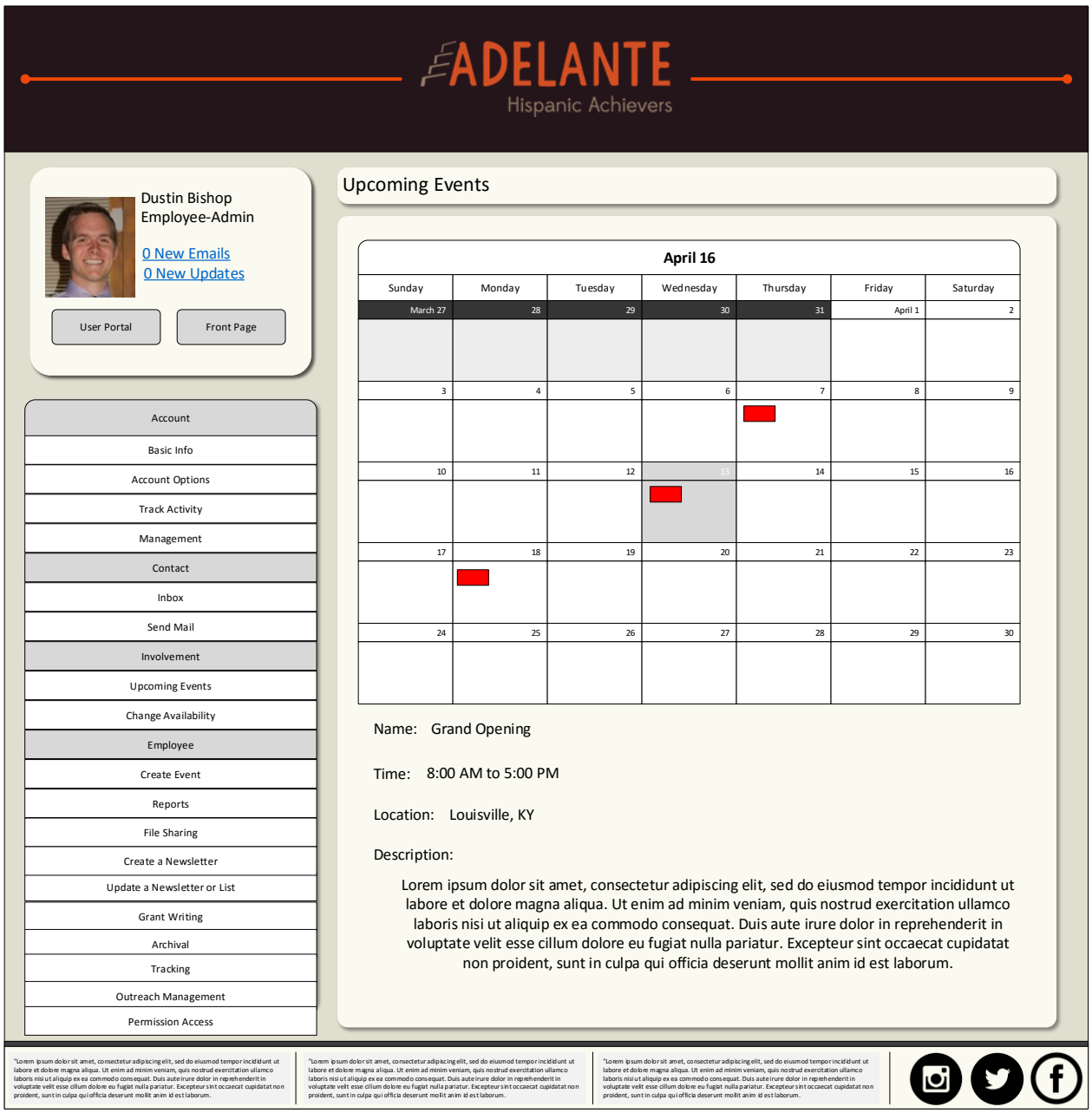
7. Account Management Back page

Account management: This page allows you to change concrete account level things, such as name, address, as well as core functionality, and lets you change passwords as well.

8. Inbox Back page

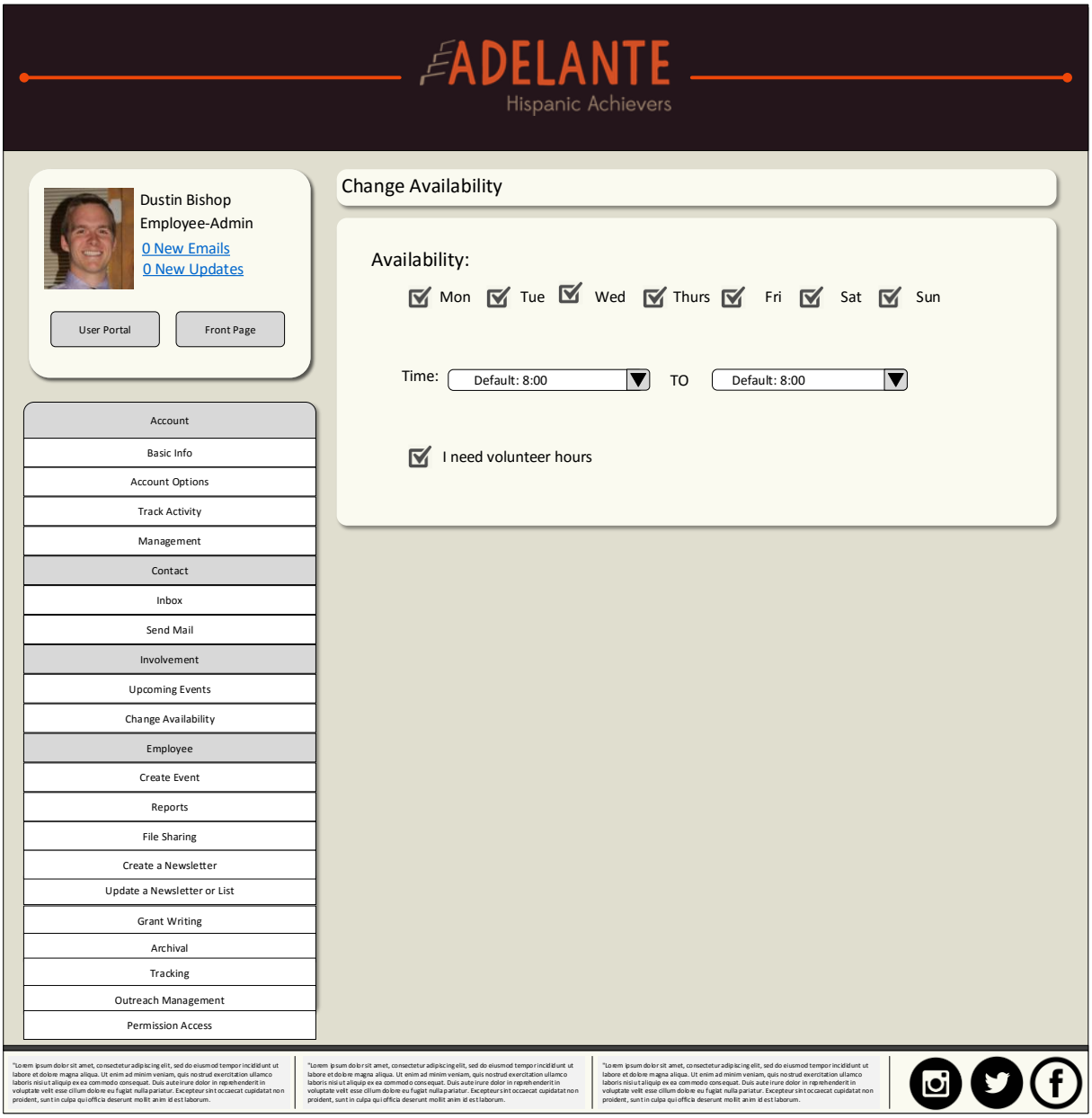
Inbox: This is simply an embedded Gmail page.

9. Upcoming Events Back page



Events: Similar to the main page, this is the discrete calendar version, noting the length and estimated duration of the event that will take place.

10. Change Availability Back page



Change availability: This option allows volunteers to adjust their volunteer hours based on their personal schedule.

11. Create Event

Events & Dates. This allows Employees to select and edit a date, as well as creating new dates and events. The difference between the two is that dates are not trackable on event trackers, as they are not true activities.

12. Reports

Reports: The reports functionality allows you to select templates, data sources, file formats, as well as formatting criteria to organize a given set of data..

13. File Sharing

*File Sharing: This allows Adelante Employees to create file sharing lists, visible to groups, that use a cloud storage provider. They have a choice between major players

14. Create Newsletter List

Create a Newsletter: This is actually slightly inaccurate, as it does not create a letter, but rather creates a list of letters

15. Send/Update Newsletter

Send/Update Newsletter: This uses the create newsletter functionality to create a list and name for a newsletter. By doing this, adelante has the ability to publish multiple newsletters quickly for an audience automatically, as long as they upload the file to do so..

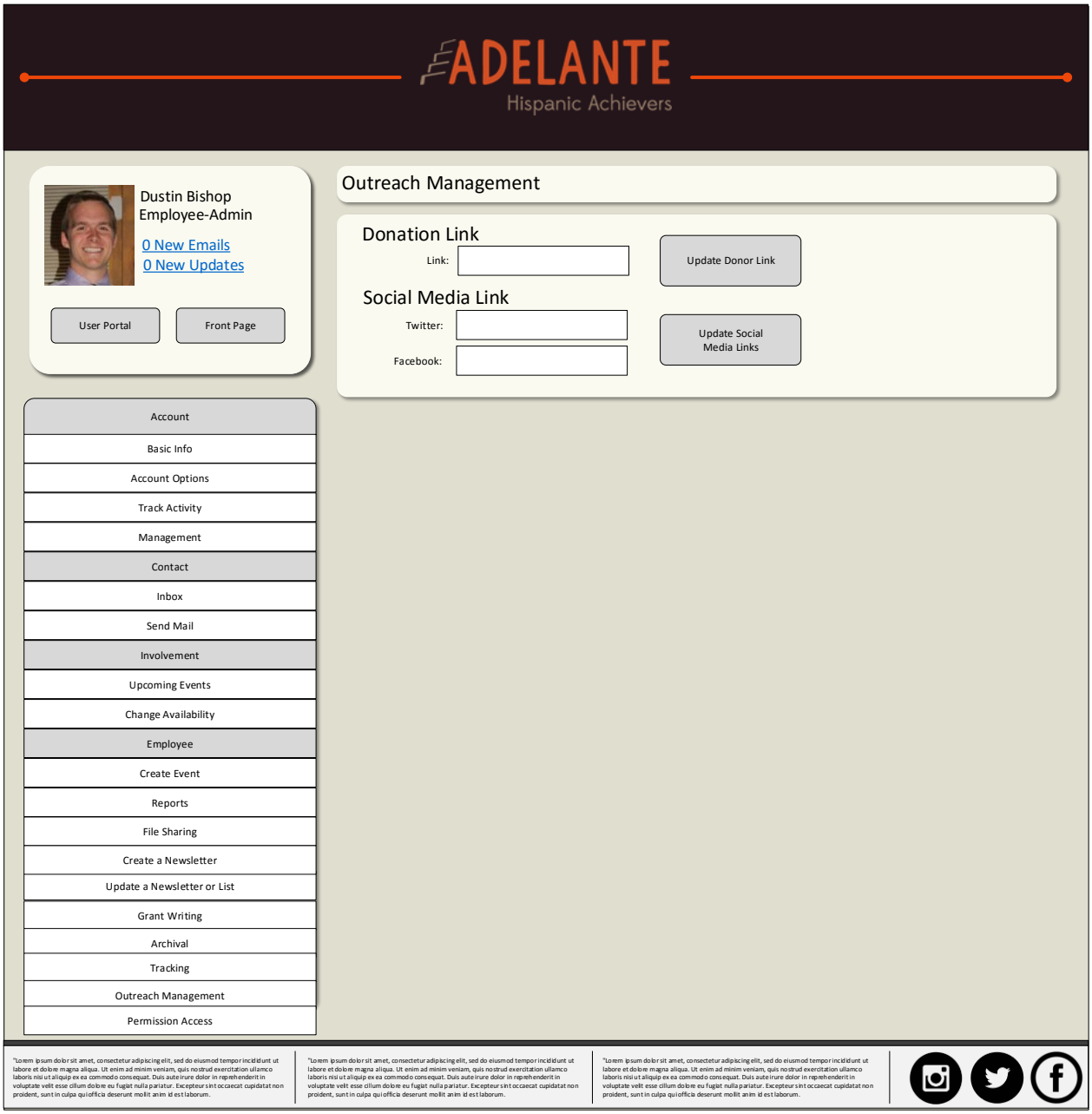
16. Archival

Create Archive: This allows an admin to name their backups, and auto-backup functions. It incorporates a number of features, such as historical data, archives can be automated as well

17. Tracking

Tracking: Allows the user to track in real time, and make custom trackers to encompass a wide variety of data

18. Outreach Management



Outreach Management: This page mostly covers links, and allows the website to update its donate link and Facebook/twitter lins with minimal effort.

19. Grants

ADELANTE
Hispanic Achievers

Dustin Bishop
Employee-Admin

0 New Emails
0 New Updates

User Portal | Front Page

- Account
  - Basic Info
  - Account Options
  - Track Activity
  - Management
- Contact
  - Inbox
  - Send Mail
- Involvement
  - Upcoming Events
  - Change Availability
- Employee
  - Create Event
  - Reports
  - File Sharing
  - Create a Newsletter
  - Update a Newsletter or List
  - Grant Writing
  - Archival
  - Tracking
  - Outreach Management
  - Permission Access

Outreach Management

Donation Link

Link: [ ]

Update Donor Link

Social Media Link

Twitter: [ ]

Facebook: [ ]

Update Social Media Links

"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Grants: Because each grant is different, we made grant writing collaborative like a cloud storage CMS. An employee can create a grant category, and other employees with the right access can join and collaborate on edits, upload files, and generally assist with edits. By using Office 365 or other similar software, we can track changes and allow a democratic process for grant writing with reject/accept changes.

20. Permissions

Permissions: This page allows one to adjust permissions for individuals. Additional functionality included is the ability to assign individuals titles. For instance, an ambassador could be created by giving an account who has student access, some element of volunteer ability.