

# 

# System Requirements

The system requirements are based on the product features section of the Vision Document to show how the system will implement these requirements. The list is organized into a hierarchy from most to least important features.

SR (1) Backup

1. The system will provide reliable backup of historical data

SR (2) Database Features

1. The system will provide for centralization of data
2. Grant tracking
   1. The system will enable application of grants
   2. The system will create records of grant applications
   3. The system will modify records of grant applications
   4. The system will track grant information

3. Data Manipulation

1. The system will compile data from database for statistical analysis
2. The system will import/export excel data to/from Cascade (JCPS)
3. The system will allow creation of queries and reports
4. The system will allow modification of queries and reports
5. The system will allow lookup and manipulation of data

4. Donor information

1. The system will allow creation of donor records
2. The system will allow modification of donor records
3. The system will link online payments to database
4. The system will delete donor records

5. Student Information

1. The system will allow creation student record
2. They system will allow modification of student records
3. The system will track student attendance
4. The system will export attendance to the database
5. The system will link sign in system with active/inactive classification
6. The system will track student grade improvement
7. The system will link family groups
8. The system will delete student records

6. Volunteer Information

1. The system will track volunteer hours
2. The system will create record of volunteers
3. The system will modify volunteer records
4. The system will allow scheduling of volunteers
5. The system will allow deletion of volunteer schedules
6. The system will delete volunteer records

7. Parent Information

1. The system will create parent record
2. They system will modify parent records
3. The system will delete parent records
4. The system will create progress and attendance reports for parents on website

8. Contact Management

1. The system will create classify contacts by category
2. The system will modify contacts to be grouped by category

9. Board Information

1. The system will create board record
2. The system will modify board record
3. The system will delete board record
4. The system will track board attendance

10. Staff Information

1. The system will create staff record
2. The system will modify staff record
3. The system will delete staff record

SR (3) Website Features

1. The system will allow easy updates and edits to website
2. The system will manage calendar of events and programs
3. The system will modify calendar of events and programs
4. The system will delete calendar of events and programs
5. The system will allow creation of editable and printable PDF registration forms
6. The system will manage website to show volunteer opportunities
7. The system will allow creation of a blog
8. The system will enable picture and video sharing
9. The system will allow requests for availability changes
10. Create mirror site
11. The system will allow staff to administer privileges to users
12. The system will allow users to login to the website
13. The system will allow staff to manage account
14. The system will allow staff to set password requirements to ensure secure accounts

SR (4) Events

1. The system will track event revenues and expenses
2. The system will allow announcement of events to invited guests
3. The system will track event participation
4. The system will track attendance

SR (5) Sign-in System

1. The system will have a digital sign-in system

# Net Present Value

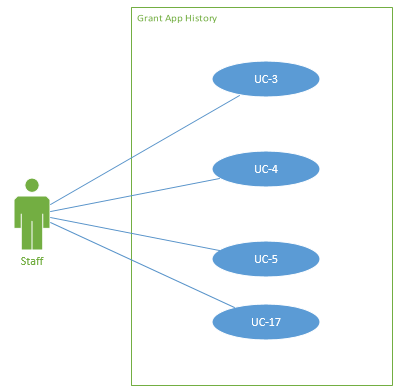
This chart displays the cost-benefit analysis of the system. All costs and expected returns are adjusted for inflation and the time value of money. The overall monetary cost or benefit of the system is listed as the cumulative NPV. ROI is the return on investment which is calculated by dividing the benefit of the system by the costs. The higher the percentage, the higher the net benefit.

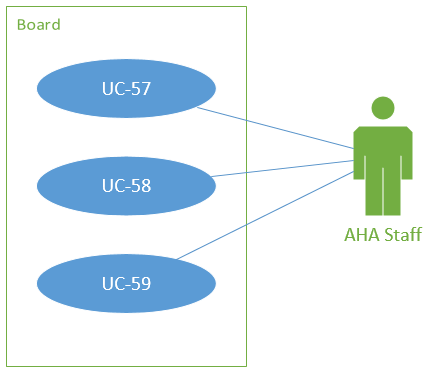
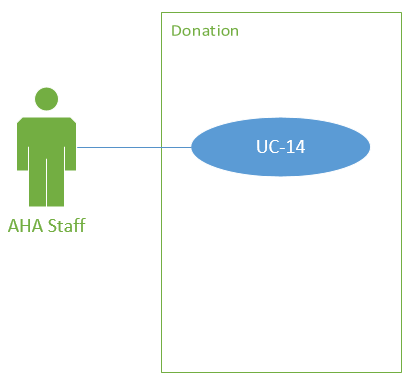
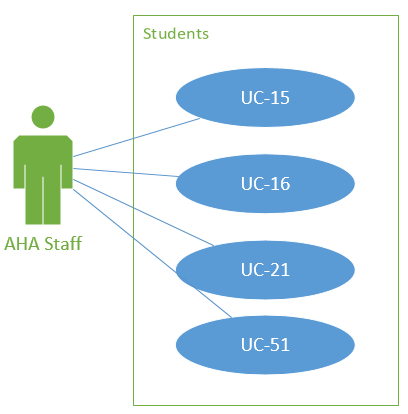
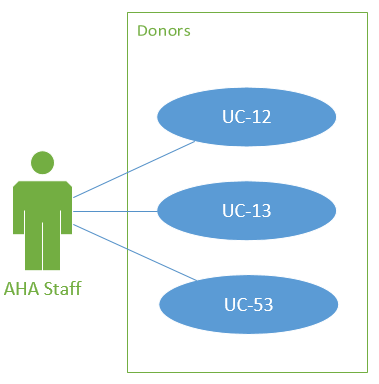
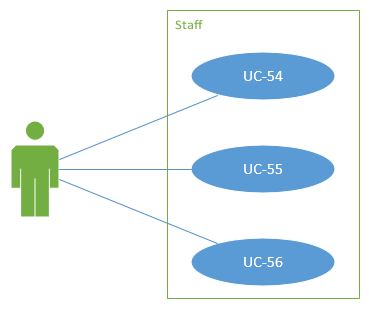
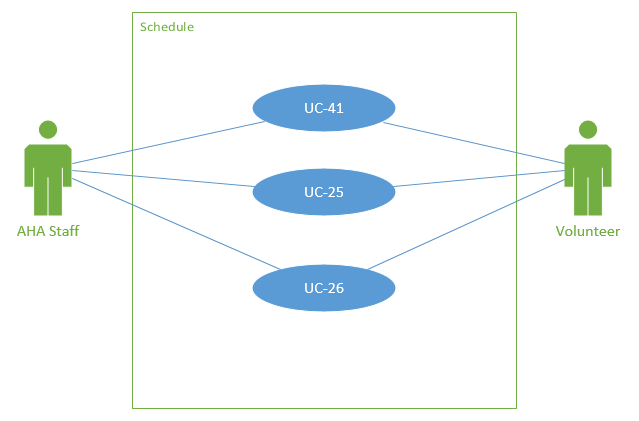
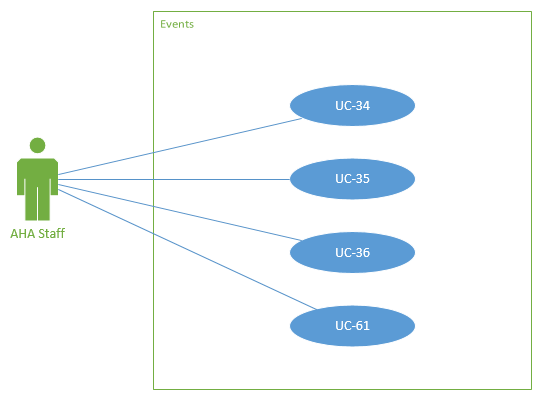
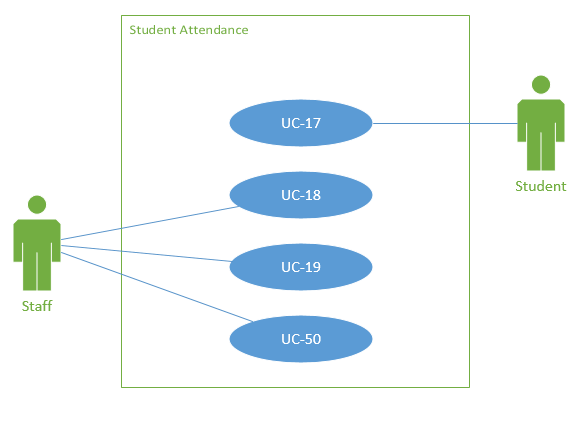
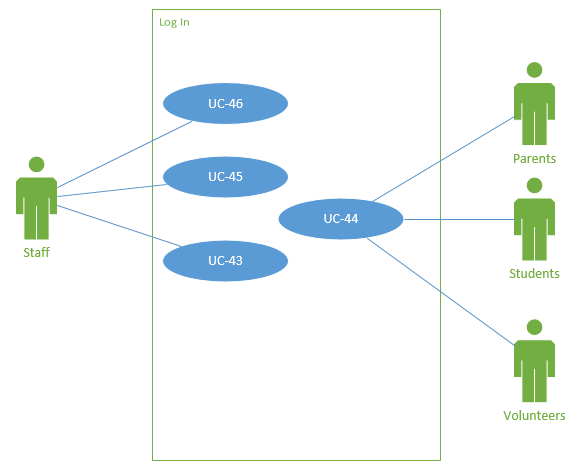
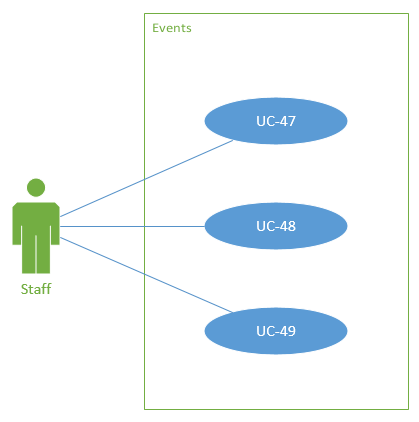
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **This chart represents a ALTERA DESIGN- NPV** | | | | | | | |
|  | **Development** | **2016** | **2017** | **2018** | **2019** | **2020** | **Total** |
| Benefits: |  |  |  |  |  |  |  |
| Increased grant funding |  | 40,000 | 41,200 | 42,436 | 43,709 | 45,020 |  |
| Increased individual donations |  | 1,200 | 1,236 | 1,273 | 1,311 | 1,351 |  |
| Increased fundraiser revenue |  | 1,031 | 1,062 | 1,094 | 1,127 | 1,160 |  |
| **TOTAL BENEFITS:** |  | **42,231** | **43,498** | **44,803** | **46,147** | **47,531** |  |
| **PV OF BENEFITS:** |  | **40,803** | **40,606** | **40,410** | **40,214** | **40,020** | **$202,053** |
| **PV OF ALL BENEFITS:** |  | **40,803** | **81,409** | **121,818** | **162,033** | **202,053** |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| Costs: |  |  |  |  |  |  |  |
| Development: |  |  |  |  |  |  |  |
| Software: | 0 | 0 | 0 | 0 | 0 | 0 |  |
| Hardware: | 0 | 0 | 0 | 0 | 0 | 0 |  |
| Labor: |  |  |  |  |  |  |  |
| DBA | 23415 | 0 | 0 | 0 | 0 | 0 |  |
| Developer | 28580 | 0 | 0 | 0 | 0 | 0 |  |
| Analyst | 24124 | 0 | 0 | 0 | 0 | 0 |  |
| **Total Development Costs:** | **76119** | **0** | **0** | **0** | **0** | **0** |  |
| Annualized: |  |  |  |  |  |  |  |
| Software: |  |  |  |  |  |  |  |
| WordPress CMS |  | 99 | 102 | 105 | 108 | 111 |  |
| Hosting |  | 120 | 124 | 127 | 131 | 135 |  |
| Database (DBMS) |  | 180 | 185 | 191 | 197 | 203 |  |
| **Total Annualized Costs:** |  | **399** | **411** | **423** | **436** | **449** |  |
| **TOTAL COSTS:** |  | 399 | 411 | 423 | 436 | 449 |  |
| **PV OF COSTS:** | 76,119 | 386 | 384 | 382 | 380 | 378 | **78,028** |
| **PV OF ALL COSTS:** | 76,119 | 76,505 | 76,888 | 77,270 | 77,650 | 78,028 |  |
| **TOTAL PROJECT BENEFITS COSTS:** | (76,119) | (35,702) | 4,520 | 44,548 | 84,383 | 124,025 |  |
| **YEARLY NPV:** | **(76,119)** | **40,417** | **40,222** | **40,028** | **39,834** | **39,642** | **124,025** |
| **CUMULATIVE NPV:** | (76,119) | (35,702) | 4,520 | 44,548 | 84,383 | **124,025** |  |
| **RETURN ON INVESTMENT:** | 158.95% |  |  |  |  |  |  |
| **BREAK-EVEN POINT:** | 2.887615054 |  |  |  |  |  |  |
| **INTANGIBLE BENEFITS:** | NONE |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| **Inflation Rate= .03** | |  |  |  |  |  |  |
| PV= Amount/(1+i)^n | |  |  |  |  |  |  |
| i = prime rate (.005) + .03 | |  |  |  |  |  |  |

# 

# Use Case Diagrams

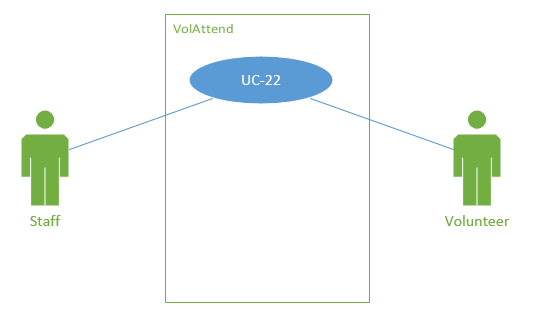
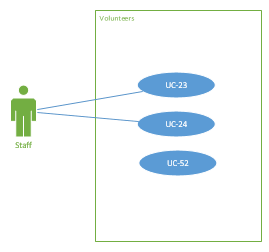
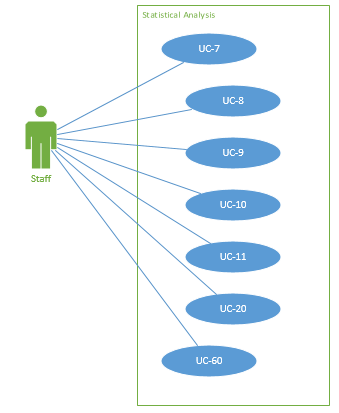
Use Case Diagrams are an analysis tool that displays the business requirements and shows the interaction between the system and the users/environment.



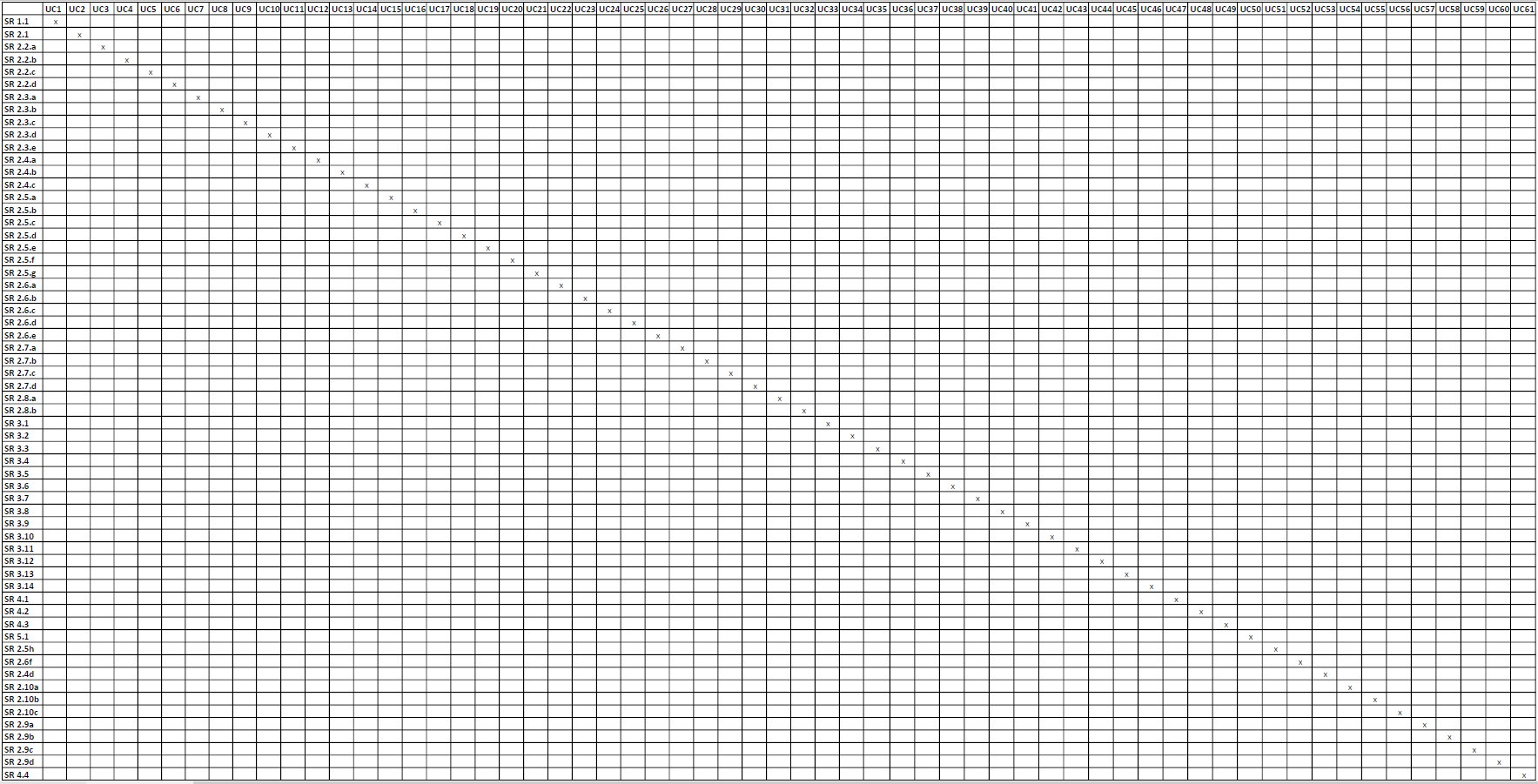


# 

# 



# Trace Matrix

The trace matrix details which system requirements complement the use cases. It is a visual representation of ensuring that each system requirement has a use case to describe it further.

# 

# Use Cases

Use cases explain how the user will interact with the system to perform the necessary business functions. They organize the functions and identify the basic flow of events.

**UC-1**

**1.1 Brief Description**

* Create a backup of all information collected by Adelante including, donors, volunteers, grants, students and parents, financials, events, and documentation, is backed up into an accessible backup storage.

**2. Flow of Events**

**2.1 Basic Flow**

* Database collects all information
* Prepares the following data for backup:
  + Donor
  + Volunteer
  + Grant
  + Student
  + Parent
  + Financial
  + Event
  + Documentation
* Through the database, the servers will automatically and routinely make copies of all of the data
  + Backups will occur daily at 2AM
  + Performed by AHA Staff
* The copied data will then be stored on local storage specific for backups

**Alternative Flows**

**2.2 Manual Backup**

* If we cannot provide AHA a way to automatically backup their data they will need to manually back up their data
  + AHA personnel will need to make a copy of all database data:
  + AHA personnel will then move that copy to a local storage medium

**3. Special Requirements**

* None

**4. Pre-conditions**

**4.1 Up and running**

* System needs to be running

**4.2 Storage Space**

* Before database copies can be stored locally, there must be enough free storage so as to not erase old backups

**5. Post-conditions**

* Copies of the entire system will be kept locally

**6. Extension Points**

* None

**UC-2**

**1.1 Brief Description**

* Create a centralized database for student, donor, grant, and parent information

**2. Flow of Events**

**2.1 Basic Flow**

* Gain access to Mara’s and Dustin’s computers and comb through files
  + Look for *any* relevant information about AHA
  + Briefly look through files
  + Sort files in a collection folders on one PC
    - Sort all relevant information about donors
    - Sort all relevant information about volunteers
    - Sort all relevant information about grants
    - Sort all relevant information about students
    - Sort all relevant information about parents
    - Sort all relevant information about financials
    - Sort all relevant information about events
    - Sort all relevant information about documentation
* After information has been organized it must initially be manually entered into the database
  + All information must be entered by hand into the respective rows and columns

**Alternative Flows**

**2.2 Extra Computers**

* Gain access to any additional computers and comb through files
  + Look for *any* relevant information about AHA
  + Briefly look through files
  + Sort files in a collection folders on one PC
    - Sort all relevant information about donors
    - Sort all relevant information about volunteers
    - Sort all relevant information about grants
    - Sort all relevant information about students
    - Sort all relevant information about parents
    - Sort all relevant information about financials
    - Sort all relevant information about events
    - Sort all relevant information about documentation
* After information has been organized it must initially be manually entered into the database
  + All information must be entered by hand into the respective rows and columns

**3. Special Requirements**

* None

**4. Pre-conditions**

**4.1 Up and running**

* All computers need to be fully functional to access and centralize files

**4.2 Personnel**

* Since this is so labor intensive we must have enough personnel to get the information centralized in a timely manner

**4.3 Compatibility**

* AHA must have met the minimum software requirements
  + This would be to run any DMBS software
  + Also any other software required to run the database
* AHA must have met the minimum hardware requirements

**5. Post-conditions**

* All files will be off of several computers
* All files will be stored in a centralized database

**6. Extension Points**

* None

**UC-3**

**1.1 Brief Description**

* Apply to grants using relevant information extracted from database

**2. Flow of Events**

**2.1 Basic Flow**

* Staff will access the database to see data about past grants
* They will then begin the process of applying to grants
* Throughout the process the database will be utilized to view statistical data
  + Also used to share files
* Throughout each step of the grant process the staff can update the status of the grant
  + Also track its progress through the database

**Alternative Flows**

None

**3. Special Requirements**

* None

**4. Pre-conditions**

**4.1 Up and running**

* System and personal computers needs to be running

**4.2 Statistics**

* Along with the database, the statistical reporting tools need to be functioning properly so that previous stats can be generated

**5. Post-conditions**

* Grants will be sent off for approval
* Database will be updated

**6. Extension Points**

* None

**UC-4**

**1.1 Brief Description**

* Create records of current and completed grant applications including name of grant, date applied, and accepted, rejected, or in progress.

**2. Flow of Events**

**2.1 Basic Flow**

* Access current grant applications
* Access completed grant applications
* For each create instances in the database of:
  + Name of grant
  + Date applied
  + Grant status
    - Accepted
    - Rejected
    - In progress

**Alternative Flows**

**2.2 Paper based system**

* For old paper based grant records:
* Access current grant applications
* Access completed grant applications
* For each create instances in the database of:
* Name of grant
* Date applied
* Grant status
* Accepted
* Rejected
* In progress

**3. Special Requirements**

None

**4. Pre-conditions**

**4.1 Up and running**

* System and personal computer needs to be running

**4.2 Statistics**

* Along with the database, the statistical reporting tools need to be functioning properly so that previous stats can be generated

**5. Post-conditions**

* Grant data will be created

**6. Extension Points**

* None

**UC-5**

**1.1 Brief Description**

* Modify records of current and completed grant applications including name of grant, date applied, and accepted, rejected, or in progress.

**2. Flow of Events**

**2.1 Basic Flow**

* Access current grant applications
* Access completed grant applications
* For each modify all needed records

**Alternative Flows**

**2.2 Paper based system**

* For old paper based grant records:
* Access current grant applications
* Access completed grant applications
* For each modify all needed records

**3. Special Requirements**

* None

**4. Pre-conditions**

**4.1 Up and running**

* System and personal computers need to be running

**4.2 Statistics**

* Along with the database, the statistical reporting tools need to be functioning properly so that previous stats can be generated

**5. Post-conditions**

* Grant data will be modified in the database

**6. Extension Points**

* None

**UC-6**

**1.1 Brief Description**

* Create and track information pertaining to awarded grants including amount received, expenditure of funds, and grantor reporting deadlines

**2. Flow of Events**

**2.1 Basic Flow**

* Staff will go into their DBMS
  + They will create or update database records to track the status of the grant
  + They will look at/update amount received
  + They will look at/update expenditure of funds
  + They will look at/update grantor reporting deadlines

**Alternative Flows**

* None

**3. Special Requirements**

* None

**4. Pre-conditions**

**4.1 Up and running**

* System needs to be running
* Personal computer need to be running

**4.2 Statistics**

* Along with the database, the statistical reporting tools need to be functioning properly so that previous stats can be generated

**5. Post-conditions**

* The database will be managed by:
  + Awarded grant record will be created
  + Awarded grant record will be updated
  + Awarded grant record will be deleted

**6. Extension Points**

* None

**UC-7**

**1.1 Brief Description**

* Use compiled data in database to identify meaningful trends concerning Adelante.

**2. Flow of Events**

**2.1 Basic Flow**

* AHA must identify trends to generate stats for
* Examples
  + Students that come to AHA get better test grades
  + The more a student is with one specific volunteer the more their attendance increases
  + The more a parent of a student visits AHA website, the better attendance is for said student
* Once trend is identified they can find reports in a number of ways:
  + AHA can run SQL queries
  + AHA can run Access reports
  + AHA can build Excel Tables
  + AHA can build Excel charts

**Alternative Flows**

* None

**3. Special Requirements**

* None

**4. Pre-conditions**

**4.1 Up and running**

* System needs to be running

**4.2 Trend ID**

* Trend must be identified before report is ran

**5. Post-conditions**

* AHA now has reports
* AHA can now compare data to figure out next move

**6. Extension Points**

* None

**UC-8**

**1.1 Brief Description**

* Import relevant data about student assessment progress from the JCPS Cascade system.

**2. Flow of Events**

**2.1 Basic Flow - Cascade to AHA**

* Access cascade portal
  + Import Cascade data into corresponding AHA student data
  + Manage Attendance
  + Manage progress
  + Run any reports needed

**Alternative Flows**

**2.2 AHA to Cascade**

* Find relevant student data in AHA database
* Organize by what is necessary to JCPS
* Export relevant AHA student information to a csv
  + Or other cascade compatible file
* Go to import section on Cascade portal
* Import current AHA student data
* Notify JCPS of the new records uploaded

**3. Special Requirements**

* None

**4. Pre-conditions**

**4.1 Up and running**

* AHA system needs to be running
* Cascade system needs to be running

**4.2 Statistics**

* Along with the database, the statistical reporting tools need to be functioning properly so that stats can be generated

**4.2 Compatibility**

* Some sort of compatibility is needed between AHA’s and Cascade’s systems
* Needed for AHA import to Cascade
* Needed for Cascade export to AHA

**5. Post-conditions**

* AHA will have current, updated Cascade information on all relevant students
* Cascade will have current, updated information on all relevant AHA students

**6. Extension Points**

* None

**UC-9**

**1.1** **Brief Description**

* Create reports to show meaningful statistical information for student, volunteers, and donors.

**2.** **Flow of Events**

**2.1** **Basic Flow**

Staff will navigate to reports page

* System will show “New Report” button
* Staff will press “New Report” button
* System will show dialog box to select type of report
* Staff will select type
* System will show Report
* Staff can save and close report

**3.** **Post-conditions**

**3.1** **Extension point**

* **<Log on to server>**
* Invalid server name or password entered
* Abort triggered

**UC-10**

**1.1 Brief Description**

* Modify reports and queries that show statistical information on students, volunteers, and donors.

**2.** **Flow of Events**

**2.1** **Basic Flow**

* Staff will navigate to reports page:
* System will show “New Report” button
* Staff will press “New Report” button
* System will show dialog box to select type of report
* Staff will select type
* System will show Report
* Staff can save and close report

**3.** **Special Requirements**

* None

**4. Pre-conditions**

**4.1 Up and running**

* System needs to be running

**5. Post-conditions**

None

**6. Extension Points**

None

**UC-11**

**1.1 Brief Description**

* Allow database to be easily manipulated to filter and sort data to staff’s preferences

**2. Flow of Events**

**2.1 Basic Flow**

* Access the DBMS
  + Filter to preference
  + Sort to preference

**Alternative Flows**

**2.2 Third party**

* Depending on resources available AHA may access the DBMS through a third party client
  + Access the DBMS
    - Filter to preference
    - Sort to preference

**3. Special Requirements**

* None

**4. Pre-conditions**

**4.1 Up and running**

* System needs to be running

**4.2 Statistics**

* Along with the database, the statistical reporting tools need to be functioning properly so that stats can be generated

**4.2 Third Party**

* Any third party tool used must be functioning properly

**5. Post-conditions**

* The data will be sorted
* The data will be filtered
* This will allow for much easier comprehension of what is going on at AHA

**6. Extension Points**

None

**UC-12**

**1.1** **Brief Description**

* Create records in donor table hold donation details

**2.** **Flow of Events**

**2.1** **Basic Flow**

* Access Staff portal:
  + Choose Records
* Select Create records
* Choose Donor to enter the following
  + First name
  + Last name
  + Street Address (if provided)
  + Donation Amount
  + Type of donor (community, staff, volunteer, or parent)
* Click create Records button
  + System will display dialog to confirm completion
  + System will classify donor as new or recurring

**2.2** **Alternative Flows**

**3.** **Pre-conditions**

**3.1** **< Must have connection to Database >**

**4.** **Extension Points**

**UC-13**

**1.1** **Brief Description**

* Modify Donor records in the database. Modification including the personal details of the donor in addition to the details of the transaction such as amount.

**2.** **Flow of Events**

**2.1** **Basic Flow**

* Access Staff portal:
  + Choose Records
  + Select modify records
    - Choose Donor to enter the following
      * First name
      * Last name
      * Street Address (if provided)
      * Donation Amount
      * Type of donor (community, staff, volunteer, or parent)
    - Click Modify Records button
* System will display dialog to confirm completion

**3.** **Pre-conditions**

**3.1** **< Must have connection to Database >**

**4.** **Extension Points**

**4.1** **<Server Authentication>**

* The server authentication must occur when connecting Excel to the Database Server, if it does not the operation cannot be performed. System will send prompt to notify user of failed connection attempt.

**UC-14:**

**1.1** **Brief Description**

* Link online payments from the website to the database to create automated payment history.

**2.**  **Flow of Events**

**2.1** **Basic Flow**

* Donor visits website.
* Navigates to donation page.
  + New donor will enter
    - Country,
    - first name,
    - last name,
    - credit card number,
    - payment type,
    - expiration date,
    - billing address,
    - City,
    - State,
    - zip code,
    - phone number,
    - Email.
* These fields will be submitted via the online payment service (Paypal).
* Paypal transfers the transaction related information directly to the Adelante system database.
* The system will record specific transaction information into the database including
  + payment method
  + donor name
  + donor amount
  + address
  + phone number
  + submission date into the database
* Notify the user that payment has been successfully submitted.

**2.2** **Alternative Flows**

**2.2.1 < First Alternative Flow >**

* If the donor already has a Paypal account, they first enter in donation amount
* The login will include:
  + Email
  + Password
* The Paypal system then transfers donor to another screen
* donor will review details to complete their payment.
* Donor will submit the payment
* Payment transferred to the Adelante system database

**4.** **Pre-conditions**

* Required preconditions:
* Adelante’s website must be functioning
* donate button must be present
* Paypal’s system must be accessible.

**4.1** **< Pre-condition One >**

* Adelante’s website must be functioning so the user can initiate the donation process.

**4.2** **< Pre-condition Two >**

* On Adelante’s website, there must be a donation button/link present so the user can navigate to the donation page.

**4.3** **< Pre-condition Three >**

* Since Adelante will rely on an external online payment system (paypal) for secure money transfers, the paypal service must be accessible.

**5.** **Post-conditions**

* Success: Donor’s payment has been accepted. Donation is transferred via paypal from the donor’s bank account to Adentlante’s bank account. System notifies donor that transfer was successful.
* Failure: Donor’s payment does not go through. System notifies the donor that transfer failed.

**6.** **Extension Points**

* Failed payment

**6.1** **Failed payment**

* After the donor has entered all fields of information to paypal and attempts to submit. If the payment action is not successful the system will abort the action and the donor will be notified.

**UC-15**

**1.1** **Brief Description**

* Create a student record to store information about students.

**2.** **Flow of Events**

**2.1** **Basic Flow**

* Access Staff portal:
* Choose Records
* Select Create record
  + Choose Student to create the following :
    - Student ID
    - Ambassador
    - First name
    - Last name
    - Street Address
    - Zip Code
    - School attended
    - Grade Level
    - Program attended
    - Start date of Program
    - Siblings (Y/N)
* Click Create Record
* System will return dialog to confirm creation

**UC-16**

**1.1** **Brief Description**

* Modify student record including personal information, educational information, and program information.

**2.** **Flow of Events**

**2.1** **Basic Flow**

* Staff will access staff portal:
* Click modify record
* Choose Student to modify any of the following:
  + Student ID
  + Ambassador
  + First name
  + Last name
  + Street Address
  + Zip Code
  + School attended
  + Grade Level
  + Program attended
  + Start date of program
  + Siblings (Y/N)
* System will perform the import and display dialog confirming completion

**3.** **Special Requirements**

* None

**4. Pre-conditions**

**4.1 Up and running**

* System needs to be running

**5. Post-conditions**

* Database will save updated record

**6. Extension Points**

None

**UC-17**

**1.1 Brief Description**

* Delete records of current and completed grant applications including name of grant, date applied, and accepted, rejected, or in progress.

**2. Flow of Events**

**2.1 Basic Flow**

* Access grant applications
* System will show grant records
* User will choose record
* System will show:
  + Grant information
  + Edit button
  + Delete button
* User will click Delete
* System will show message box:
  + “Delete Record?”
  + Confirm button
  + Cancel button
* User will click confirm
* System will delete record
* System will display updated list of grant records

**Alternative Flows**

**2.2 User cancels**

* User clicks Cancel
* System cancels deletion

**3. Special Requirements**

* None

**4. Pre-conditions**

**4.1 Login**

* Staff must be logged onto the system

**5. Post-conditions**

* System will display updated grant record page

**6. Extension Points**

* None

**UC-18**

**1.1** **Brief Description**

* Export attendance records to database

**2.** **Flow of Events**

**2.1** **Basic Flow**

* Staff will Access Attendance 2 app:
  + Export CSV file to excel workbook
  + Export excel workbook to Azure SQL
* System will:
  + Import the Excel CSV file to the Database

**3.** **Preconditions**

* Students have signed in on ipad with attendance 2 app

**3.2 Up and running**

* System needs to be running

**4.** **Special Requirements**

* None

**5. Post-conditions**

None

**6. Extension Points**

**UC-19**

**1.1** **Brief Description**

* Create report from student attendance to classify student as active/inactive in program.

**2.** **Flow of Events**

**2.1** **Basic Flow**

* Staff presses button to view report of student attendance and classification.
* System counts all days student was present at program.
  + If the count / total = >50%, calculated field shows a value of “active”
  + If the count / total = <50%, calculated field shows a value of “inactive”
* System returns to staff user a report including the calculated field.
* System includes an [X] to exit out of report.

**2.2** **Alternative Flows**

* None

**3.** **Special Requirements**

* None

**4.** **Pre-conditions**

**4.1** **Logged in**

* Staff must be logged onto the staff portal

**4.2** **Up and running**

* Both systems need to be functioning

**4.3** **Student ID**

* The student must have a unique student ID that already exists in the attendance and student tables

**5.** **Post-conditions**

* After clicking the [X], staff is taken back to portal home page.

**6.** **Extension Points**

* None

**UC-20**

**1.1** **Brief Description**

* Staff receives inflows of data to/from the Cascade system used by JCPS to track student grade improvement.

**2.** **Flow of Events**

**2.1** **Basic Flow**

* On staff portal, staff user requests to receive data from CASCADE system
* System sends request to CASCADE system
* CASCADE system accepts and sends data to Adelante system
* Inflows of data from CASCADE could include:

o Student name

o Student age

o Student classes

o Student GPA history

* Staff notified data was obtained and system displays data on page

**2.2** **Alternative Flows**

* CASCADE rejects requests
* Adelante system displays message that request blocked

**3.** **Special Requirements**

* JCPS must agree to create a link with their system and distribute/receive information.
* All regulations of student information must be followed.
* Only specified data will be transferred.

**4.** **Pre-conditions**

**4.1** **A Link between two worlds**

* Database and CASCADE system must share a connection.

**4.2** **Up and running**

* Both systems need to be functioning.

**4.3** **Student Existence**

* Student must be an identifiable JCPS student.
* Student must be an identifiable Adelante student

**5.** **Post-conditions**

* System displays homepage of staff portal.

**6.** **Extension Points**

* None

**UC-21**

**1.1** **Brief Description**

* Create relationships between students and parents, thereby creating family groups.

**2.** **Flow of Events**

**2.1** **Basic Flow**

* Staff clicks to open relationship form.
* System responds by displaying the form, which includes:
  + - A box to enter student ID
    - A box to enter parent ID
    - Submit button
    - Cancel button
* On relationship form, staff lists a parent-to-student relationship.
* Staff clicks Submit
* ID data sent to database
* System creates proper relationship in database
* System sends back message to staff that relationship was created

**2.2** **Alternative Flows**

* Staff chooses to not actually create relationship
* Staff clicks “Cancel” button
* System receives cancellation
* System terminates form activity, and form itself.

**3.** **Special Requirements**

* None

**4.** **Pre-conditions**

**4.1** **Logged in**

* Staff is logged in to staff portal.

**4.2** **Up and running**

* System needs to be functioning

**4.3** **Student and parent existence**

* Student and parent IDs must exist within database in correct location.

**5.** **Post-conditions**

* After Submit/cancel made, system displays new blank form to enter a new relationship.

**6.** **Extension Points**

* None

**UC-22**

**1.1** **Brief Description**

* Track volunteer hours and programs worked.

**2.** **Flow of Events**

**2.1** **Basic Flow**

* In Staff portal, staff user navigates to volunteer management page.
* Staff member clicks to run a report on volunteer(s) by choosing either:
  + All Volunteers
  + Specific Volunteer - Enter volunteer ID
* Staff chooses date range to run report, such as
  + Last month
  + Last year
* Underneath is a “Get Report” button
* Also underneath is a “Cancel” button
* Staff member clicks “Get Report”
* System pulls from database information on volunteer(s) including:
  + Volunteer ID
  + Volunteer Name
  + Hours worked
  + Programs worked
* System displays volunteer information in a report to the staff user

**Alternative Flows**

* None

**3.** **Special Requirements**

* None

**4.** **Pre-conditions**

**4.1** **Up and running**

* System needs to be running

**4.2** **Logged In**

* Staff must be logged into staff portal

**4.3** **Volunteer existence**

* Volunteer IDs must exist within database in correct location.

**5.** **Post-conditions**

* System will display area for log in for the next volunteer to sign in refreshed page to run a new report on volunteers.

**6.** **Extension Points**

* None

**UC-23**

**1.1** **Brief Description**

* Create volunteer record of personal volunteer information.

**2.** **Flow of Events**

**2.1** **Basic Flow**

* On portal, staff clicks to open up a form for entering volunteer information
* System displays volunteer information form
* Volunteer information form will include boxes to include:
  + - Volunteer name
    - Volunteer address
    - Volunteer email
    - Volunteer phone number
    - Group affiliations (if applicable)
    - Educational background
    - Tutoring subject interest
* Also present will be:
  + - Submit button
    - Cancel button
* Staff will enter all information into form boxes
* Staff will click “submit”
* System will send information to database, and will record as a new record
* System will send back a message saying “new record created”

**Alternative Flows**

* Staff clicks “Cancel”
* Volunteer form terminated

**3.** **Special Requirements**

* None

**4.** **Pre-conditions**

**4.1** **Up and running**

* System needs to be running

**4.2** **Logged in**

* Staff member must be logged onto the staff portal

**5.** **Post-conditions**

* System will display page with new, blank volunteer information form so user can add another volunteer record.

**6.** **Extension Points**

* None

**UC-24**

**1.1** **Brief Description**

* Modify volunteer record to change volunteer information

**2.** **Flow of Events**

**2.1** **Basic Flow**

* On portal, staff searches for a particular user, and then clicks to open web interface component to update volunteer information
* Web interface component opens up a page displaying (in an editable fashion) a table of current attributes of a particular record including:
  + - Volunteer name
    - Volunteer address
    - Volunteer email
    - Volunteer phone number
    - Group affiliations (if applicable)
    - Educational background
    - Tutoring subject interest
* Also present are two buttons:
  + - Save Changes
    - Cancel
* Staff can choose to click and modify any of the attributes
* After staff is satisfied, they will click the submit button
* Changes are sent through the system and updated in the database
* A message is displayed by the system to the staff user, saying that the changes were successfully made

**Alternative Flows**

* Staff clicks “Cancel”
* Modification interface terminated

**3.** **Special Requirements**

* None

**4.** **Pre-conditions**

**4.1** **Up and running**

* System needs to be running

**4.2** **Logged in**

* Staff member must be logged onto the staff portal

**5.** **Post-conditions**

* System will return to page to search for specific volunteers.

**6.** **Extension Points**

* None

**UC-25**

**1.1** **Brief Description**

* Create volunteer schedules that will include availability and commitment to work.

**2.** **Flow of Events**

**2.1** **Basic Flow**

* On Adelante website, volunteer will click on the calendar app.
* System will open calendar app, which displays in a monthly fashion
* Volunteer chooses which date they would like to schedule on.
* System displays
  + - The various programs available on that day
    - Slots available.
* Volunteer check marks slots they would like to fill.
* System displays all slots volunteer has selected
* Underneath, system shows:
  + - “confirm” button
    - “cancel” button
* Volunteer clicks the “confirm” button
* System updates calendar app.
* System displays the updated calendar with their name filling the slot they chose.

**Alternative Flows**

* Volunteer clicks “Cancel”
* System terminates the scheduling interface
  + System returns to the original calendar showing slots available/taken

**3.** **Special Requirements**

* Volunteer calendar app must be linked/embedded on the Adelante page.

**4.** **Pre-conditions**

**4.1** **Up and running**

* Both system need to be running

**4.2** **Logged in**

* Volunteer member must be logged onto the volunteer portal

**5.** **Post-conditions**

* System will return to overall calendar for the current month.

**6.** **Extension Points**

* None

**UC26**

**1.1 Brief Description**

* Delete volunteer availability and commitment to work

**2. Flow of Events**

**2.1 Basic Flow**

* Volunteer or staff (user) will navigate to calendar
* System will present monthly calendar of programs
* User will navigate to correct date
* System will show program on that date
* User will choose program
* System will show:
  + filled volunteer position for time slot
  + Edit button
* User will click “edit”
* System will show Message Box:
  + “Delete Sign Up?”
  + Radio buttons:
    - “Delete this time slot only”
    - “Delete all future program time slots”
  + “Submit” button
* User chooses to delete one time or repeat
* User submits request by clicking “Submit” button
* System shows “Confirm” button for request and “Cancel” button to cancel request
* User will click “Confirm”
* System updates calendar
* System navigates to time slot calendar page
* User views change

**2.2 Alternative Flows**

**2.2.1 User Does Not Wish to Delete**

* If User chooses not to delete the time slot selected, then they will click the “Cancel” button
* System will cancel deletion
* System will navigate back to time slot page of calendar

**3. Special Requirements**

**3.1 < Calendar Accessibility >**

* Calendar application must be linked to Adelante website

**4. Pre-conditions**

**4.1 < Functioning Website >**

* Adelante’s website must be functioning so the user can initiate the donation process.

**4.2 < Login >**

* User must log into system

**5. Post-conditions**

**5.1 < System Display >**

* System will display time slot calendar page

**6. Extension Points**

* None

**UC27**

**1.1 Brief Description**

* Create records containing personal information for parents in the program.

**2. Flow of Events**

**2.1 Basic Flow**

* Staff will click on Parent Information form
* System will display Parent Information form
* Staff will enter relevant information:
  + Parent name
  + Parent address
  + Parent phone number (if available)
  + Parent cell phone number (if available)
  + Parent work number (if available)
  + Parent email address (if available)
  + Student ID
  + Parent ID
* User will submit form
* System will create record
* System will display new, blank Parent Information form

**2.2 Alternative Flows**

* None

**3. Special Requirements**

* None

**4. Pre-conditions**

**4.1 < Login >**

* User must log into system.

**4.2 < Form>**

* Parent information form must be available

**5. Post-conditions**

**5.1 < System Display >**

* System will display new, blank Parent form after submission

**6. Extension Points**

* None

**UC28**

**1.1 Brief Description**

* Modify records containing personal information for parents in the program.

**2. Flow of Events**

**2.1 Basic Flow**

* Staff will search for specific parent in database
* System will display parent name
* Staff will click on parent record
* System will display:
  + parent information page
  + “Edit” button
  + “Delete” button
* Staff will click “Edit” button
* System will show:
  + completed parent information form
  + “Submit” button
  + “Cancel” button
* Staff will edit parent information
* Staff will click “Submit”
* System will update parent record
* System will display parent information page

**2.2 Alternative Flows**

**2.2.1 User Does Not Wish to Modify Record**

* If User chooses not to modify the record selected, then they will click the “Cancel” button
* System will cancel modification
* System will retain previous record iteration
* System will display parent information page

**3. Special Requirements**

* None

**4. Pre-conditions**

**4.1 < Login >**

* User must log into system.

**5. Post-conditions**

**5.1 < System Display >**

* System will display parent information page

**6. Extension Points**

* None

**UC29**

**1.1 Brief Description**

* Delete records containing personal information for parents in the program.

**2. Flow of Events**

**2.1 Basic Flow**

* Staff will navigate to database parent list
* Staff will click on parent record
* System will display:
  + parent information page
  + “Edit” button
  + “Delete” button
* Staff will click “Delete” button
* System will show Message Box:
  + “Delete Record?”
  + “Confirm” button
  + “Cancel”
* User will click “Confirm”
* System updates parent records
* System returns to parent list

**2.2 Alternative Flows**

**2.2.1 User Does Not Wish to Delete Record**

* If User chooses not to delete the record selected, then they will click the “Cancel” button
* System will cancel deletion
* System will retain previous record iteration
* System will display parent list

**3. Special Requirements**

* None

**4. Pre-conditions**

**4.1 < Login >**

* User must log into system.

**5. Post-conditions**

**5.1 < System Display >**

* System will display parent list

**6. Extension Points**

* None

**1. UC30: Create Progress and Attendance Reports**

**1.1 Brief Description**

* Create progress and attendance reports for parents on website

**2. Flow of Events**

**2.1 Basic Flow**

* Parent will navigate to Parent page
* System will show Parent page with “New Report” button
* Parent will press “New Report” button
* System will show Message Box with box to enter Student ID
* Parent will enter Student ID
* System will show Student Progress and Attendance Report with “Close” button
* Parent will review and close report
* System does not save report
* System closes report

**2.2 Alternative Flows**

* None

**3. Special Requirements**

* None

**4. Pre-conditions**

**4.1 < Functioning Website >**

* Adelante’s website must be functioning so the user can initiate the report process.

**4.2 < Login >**

* User must log into system

**4.3 < Relationships>**

* Parent, student, and valid parent-student relationship must be entered into system

**5. Post-conditions**

**5.1 < System Display >**

* System will display Parent page

**6. Extension Points**

* None

**1. UC31: Create and Classify Contacts**

**1.1 Brief Description**

* Create contacts to be organized into groups such as parents, students, donors, and volunteers. This will allow for mass communication and enable staff to send emails to specific groups.

**2. Flow of Events**

**2.1 Basic Flow**

* User will convert Azure Table to csv format.
* User will import csv file into Outlook
* Outlook will create contacts from csv file
* User will create contact groups in Outlook
* User will sort by:
  + Grade
  + Program
* User will assign contacts to contact groups for targeted email

**2.2 Alternative Flows**

* None

**3. Special Requirements**

**3.1 < Outlook >**

* User must have access to Outlook

**4. Pre-conditions**

**4.1 < Login >**

* User must log into system and Outlook.

**5. Post-conditions**

**5.1 < System Display >**

* System will display Contacts

**6. Extension Points**

* None

**1. UC32: Modify Contacts**

**1.1 Brief Description**

* Modify contacts to be organized into groups such as parents, students, donors, and volunteers.

**2. Flow of Events**

**2.1 Basic Flow**

* Staff will launch Outlook
* System will display log in prompt
* Staff will log into Outlook
* System will display inbox
* Staff will click Contacts icon
* System will display Contacts list
* Staff will choose contact to modify
* Outlook will display:
  + Contact’s information
  + New email button
  + Edit button
* Staff will click edit button
* Outlook shows:
  + Form for editing:
    - Name
    - Address
    - Phone
    - Work
    - Birthday
    - Grade
    - Program
  + “Save” button
  + “Cancel” button
* Staff will edit contact information
* Staff will click “Save”
* Outlook will save new contact information

**2.2 Alternative Flows**

**2.2.1 User Does Not Wish to Modify Contact**

* If User chooses not to modify the contact selected, then they will click the “Cancel” button
* Outlook will cancel modification
* Outlook will retain previous contact iteration
* Outlook will display contact information page

**3. Special Requirements**

**3.1 < Outlook >**

* Staff must have access to Outlook

**4. Pre-conditions**

* None

**5. Post-conditions**

**5.1 < System Display >**

* System will display contact information page

**6. Extension Points**

* None

1. **UC33: Easy website updates and edits**

**1.1 Brief Description**

* Provide staff an easy method to update and edit information on AHA website that doesn’t require extensive coding knowledge.

**2. Flow of Events**

**2.1 Basic Flow**

* Create administrative account on WordPress
* Create AHA website using WordPress
* Staff can now use simple WordPress tools to make updates and edits to site as necessary

**2.2 Alternative Flows**

* Updates or edits fail
* Send relevant information to program participants by email

**3. Special Requirements**

None

**4. Pre-Conditions**

* Functional computer
* Internet connection
* WordPress account
* WordPress installed on computer
* WordPress created website
* Installed on computer most recent versions of:
* Google Chrome or,
* Internet Explorer or,
* Firefox or,
* Safari

**5.**  **Post-Conditions**

* Website displays all updates or edits made through WordPress

**6**. **Extension Points**

* WordPress login authentication
  + Staff users with WordPress administrative level user name and password must enter credentials correctly to be authorized to make site changes.

1. **UC34: Create calendar events and programs**

**1.1 Brief Description**

* Create website with an embedded calendar to display upcoming events, registration dates, and dates for program starts.

**2. Flow of Events**

**2.1 Basic Flow**

* Create a Google account for Adelante Staff
* Staff login to Adelante Google account
* Staff creates calendar to track events and programs
* Copy calendar embed code
* Log in to WordPress
* Paste into website through WordPress
* Save and publish changes

**2.2 Alternative Flows**

* Calendar embed fails
* Use blog functionality in WordPress to share calendar updates

**3. Special Requirements**

* None

**4. Pre-Conditions**

* Google account
* WordPress login
* JavaScript enabled
* Cookies enabled
* Most recent versions of:
* Google Chrome or,
* Internet Explorer or,
* Firefox or,
* Safari
* Functioning computer
* Internet connections

**5. Post-Conditions**

* Website has Google calendar embedded

**6. Extension Points**

* WordPress login authentication
  + Staff users with WordPress administrative level user name and password must enter credentials correctly to be authorized to make site changes.
* Google login authentication
  + Staff users with Google administrative level user name and password must enter credentials correctly to be authorized to make site changes

1. **UC35: Modify calendar events and programs**

**1.1 Brief Description**

* Make changes to embedded calendar in website as necessary

**2. Flow of Events**

**2.1 Basic Flow**

* Log in to Adelante Google account
* Navigate to Google calendar
* Access calendar storing events and programs
* Make necessary modifications to events and programs
* Save changes.

**2.2 Alternative Flows**

* Calendar modifications fail
* Use blog functionality in WordPress to share calendar updates
* Use email to share calendar updates with program participants

**3. Special Requirements**

* None

**4. Pre-Conditions**

* Google account
* JavaScript enabled
* Cookies enabled
* Most recent versions of:
* Google Chrome or,
* Internet Explorer or,
* Firefox or,
* Safari
* Website with Google Calendar embedded
* Functioning computer or mobile device
* Internet connection

**5. Post-Conditions**

* Changes to event or program displayed in embedded calendar

**6. Extension Points**

* WordPress login authentication
  + Staff users with WordPress administrative level user name and password must enter credentials correctly to be authorized to make site changes.
* Google login authentication
  + Staff users with Google administrative level user name and password must enter credentials correctly to be authorized to make site changes

1. **UC36: Delete calendar events and programs**

**1.1 Brief Description**

* Delete items from embedded calendar in website as necessary

**2. Flow of Events**

**2.1 Basic Flow**

* Log in to Adelante Google account
* Navigate to Google calendar
* Access calendar storing events and programs
* Make necessary deletions of events and programs
* Save changes.

**2.2 Alternative Flows**

* Deletion of calendar items fail
  + Use blog functionality in WordPress to share calendar updates
  + Use email to share calendar updates with program participants

**3. Special Requirements**

* None

**4. Pre-Conditions**

* Google account login
* JavaScript enabled
* Cookies enabled
* Most recent versions of:
* Google Chrome or,
* Internet Explorer or,
* Firefox or,
* Safari
* Website with Google Calendar embedded
* Functioning computer or mobile device
* Internet connection

**5**. **Post-Conditions**

* Event program removed from embedded calendar

**6. Extension Points**

* WordPress login authentication
  + Staff users with WordPress administrative level user name and password must enter credentials correctly to be authorized to make site changes.
* Google login authentication
  + Staff users with Google administrative level user name and password must enter credentials correctly to be authorized to make site changes

1. **UC37: Create registration forms in PDF format that can be edited and printed**

**1.1 Brief Description**

* Provide registration forms and applications that can be filled and printed out by each student registering with AHA. Forms must be printed out and taken to registration event held by Adelante each year.

**2. Flow of Events**

**2.1 Basic Flow**

* Staff opens registration form file in Adobe Acrobat DC
* If only physical copy exists, use scanner to create digital copy and open in Adobe Acrobat DC
* Adobe Acrobat DC will automatically place fillable fields in place of static fields detected
* Add, delete, or modify fields as necessary.
* Save file as PDF
* Staff logs in to WordPress
* Use WordPress administrative functions to upload PDF to website
* Create download link with WordPress
* Save and publish website

**2.2 Alternative Flows**

* Users don’t have Adobe or PDF reader program installed
  + Blank physical applications will be provided during registration
* Users don’t have access to a computer
  + Blank physical applications will be provided during registration
* Download capabilities fail
  + Blank physical applications will be provided during registration
  + Forms can be emailed upon request to staff

**3. Special Requirements**

* None

**4. Pre-Conditions**

* Adobe Acrobat Reader or other PDF reader program
* Most recent versions of:
* Google Chrome or,
* Internet Explorer or,
* Firefox or,
* Safari
* Website with Google Calendar embedded
* Functioning computer
* Internet connection

**5. Post-Conditions**

* PDF files are available for access to website visitors by hyperlink

**6. Extension Points**

* WordPress login authentication
  + Staff users with WordPress administrative level user name and password must enter credentials correctly to be authorized to make site changes.

**UC38: Display volunteering opportunities on website**

**1.1 Brief Description**

Provide staff the capability to post available volunteering opportunities and positions to which applicants can apply.

**2. Flow of Events**

**2.1 Basic Flow**

* Staff has available volunteer positions that must be filled.
* Log in to WordPress for administrative access to website
* Navigate to Volunteers: Opportunities page
* Add opportunities to page.

**2.2 Alternative Flows**

* Volunteer opportunities fail to post
* Redirect potential volunteers to contact staff directly for position availability

**3. Special Requirements**

None

**4. Pre-Conditions**

* WordPress login
* Most recent versions of:
* Google Chrome or,
* Internet Explorer or,
* Firefox or,
* Safari
* Website with Google Calendar embedded
* Functioning computer
* Internet connection

**5. Post-Conditions**

* Volunteer opportunities are available to view by site visitors

**6. Extension Points**

* WordPress login authentication
  + Staff users with WordPress administrative level user name and password must enter credentials correctly to be authorized to make site changes.

**UC39: Create blog**

**1.1 Brief Description**: Add blog to AHA website to allow for posting announcements, events, programs, updates and relevant information by staff for website visitors.

**2. Flow of Events**

**2.1 Basic Flow**

* Staff have information or program and events updates to share with site visitors.
* Using WordPress administrative level access to site, blog can be added.
* Blog can be modified as necessary with WordPress login.

**3. Alternative Flows**

* Blog fails to load
  + Send out Blog updates important to program participants by email

4. **Pre-Conditions**

* WordPress administrative level login
* Most recent versions of:
* Google Chrome or,
* Internet Explorer or,
* Firefox or,
* Safari
* Website with Google Calendar embedded
* Functioning computer
* Internet connection

**5. Post-Conditions**

* Blog is displayed on website that is updated as needed by staff.

**6. Extension Points**

* WordPress login authentication

Staff users with WordPress administrative level user name and password must enter credentials correctly to be authorized to make site changes.

**UC40: Enable picture and video sharing**

**1.1 Brief Description**

Allow staff to post and share photos and videos of events and programs hosted by Adelante Hispanic Achievers with families, students, and visitors of Adelante website.

**2. Flow of Events**

**2.1 Basic Flow**

* Staff have pictures and video from events, programs, and other sources to share on website
* WordPress administrative level access to site provides access to uploading media.
* Media will be displayed on site front end to visitors

**2.2 Alternative Flows**

* Picture and video sharing fail
  + Photos and video can be uploaded to alternative photo and video sharing sites (Facebook, Picasa, YouTube) and corresponding URLs sent by email or posted to website

**3. Special Requirements**

None

**4. Pre-Conditions**

* WordPress administrative level login
* Photo and video media in file format on computer
* Most recent versions of:
* Google Chrome or,
* Internet Explorer or,
* Firefox or,
* Safari
* Website with Google Calendar embedded
* Functioning computer
* Internet connection

**5. Post-Conditions**

* Staff now have the capability to post pictures and video to share with website visitors

**6. Extension Points**

* WordPress login authentication
  + Staff users with WordPress administrative level user name and password must enter credentials correctly to be authorized to make site changes.

**UC41: Request change in volunteer availability**

**1.1 Brief Description**

Provide volunteers with a direct line of communication to Dustin regarding a change in availability and Dustin a solution to tracking volunteer schedule change communications.

**2. Flow of Events**

**2.1 Basic Flow**

* Volunteer has schedule conflict with existing availability
* Volunteer clicks on “Volunteers” tab to open drop down menu
* Volunteer selects “Availability” from drop down menu
* Volunteer is redirected to Availability page
* Volunteer enters dates into “Dates affected” field using dropdown month and day choices
* Volunteer uses drop down “Reasons” menu to select reason for change in availability.
* Volunteer clicks send
* Message is sent as email to Dustin’s email.
* Carbon copy of message is sent to email provided by volunteer on registration.
* Message box confirming transmission of message.
* Dustin reviews availability change for approval.

**2.2 Alternative Flows**

* Volunteer is not registered
  + Attempts to access Availability screen without having registered
  + Volunteer “Availability” page not accessible without log in
* Attempts to log in with screenname and password not existent in system
  + Message box appears informing volunteer they are not registered and to contact Dustin by email.
* Volunteer enters incorrect screenname and or password
  + Authentication error handled with message box that entered password and username are incorrect
* Volunteer “reason for schedule change” not listed
  + Volunteer uses “Other” option and provides reason in text field

**3. Special Requirements**

None

**4. Pre-Conditions**

* Volunteer level login
* Most recent version of:
* Google Chrome or,
* Internet Explorer or,
* Firefox or,
* Safari
* Registered in AHA system as “volunteer”
* Functional computer system
* Internet connection
* Functional website
* Website email capability

**5. Post-Conditions**

* Volunteers can now send changes in availability
* Dustin has electronic paper trail to availability requests

**6. Extension Points**

* Volunteer level login authentication
  + Volunteer users with AHA website volunteer level username and password must enter credentials correctly to be authorized to make site changes.

**UC 42: Create mirror site**

**1.1 Brief Description**

Adelante’s website will be primarily in English with the option to translate into Spanish to accommodate individuals who only speak one of those languages

**2. Flow of Events**

**2.1 Basic Flow**

* <Rewrite code for mirror site>
  + Current website will be duplicated by:
    - Accessing current HTML, JavaScript, or CSS code
    - Rewrite code to display Spanish context, including:
      * Text
      * Images
  + Users click “Espanol” button on main website
  + Website content is translated to Spanish

**3. Special Requirements**

* + None

**4. Pre-conditions**

* Required pre-conditions:
  + Internet access
  + Be able to translate English to Spanish
* < Pre-condition One >
  + Adelante’s website must be accessible for developer or staff member to easily install plug in or access HTML code.
* < Pre-condition Two >
  + Adelante staff must be able to decipher current text in English to Spanish to provide to web developer to translate text.

**5. Post-conditions**

* Post-condition are as follows:
  + Successful translation
  + Inaccurate translation
  + Images not translated
* < Post-condition One >
  + The website is successfully translated so that users can view the site in English or Spanish
* < Post-condition Two >
  + There is a chance that the English text could be inaccurately translated to Spanish. This could be the result of a human error made by the web developer or a glitch with Google Translator. When translating from English to Spanish, there is often a multitude of translation for one word so the best translation may not be produced.
* < Post-condition Three >
  + With the use of Google Translator, sometimes the content inside images cannot be translated due to the text not being able to be recognized.

**6. Extension Points**

* + None

**UC 43 - Administer privileges to users**

**1.1 Brief Description**

Allows staff to change the level of access to the data on the website to certain groups. Level of access and view of the data will be different for students, parents, volunteers, and staff

**2. Flow of Events**

**2.1 Basic Flow**

* Staff clicks on “Administer Privileges” button
* Staff selects user type
* Staff selects which functionality to make changes to
* Staff selects appropriate level of access for that user type and functionality
* Staff selects “Update” button
* User is able to navigate throughout website with appropriate privileges assigned

**2.2 Alternative Flows**

* None

**3. Special Requirements**

* None

**4. Pre-conditions**

* Required pre-conditions:
  + Website must be accessible
  + User must be logged in
* < Pre-condition One >
  + Staff must be able to access website code in order to add conditions to recognize user type.
* < Pre-condition Two >
  + User must be logged into website in order to access administer privileges.

**5. Post-conditions**

* Post-condition are as follows:
  + Correct user types are identified
* < Post-condition One >
  + System successfully assigns user types to each individual who logs onto website to show appropriate content.

**6. Extension Points**

* None

**UC 44 - Login to website**

**1.1 Brief Description**

Allows users to log on to the website with login credentials previously created. System assigns appropriate access privileges to user.

**2. Flow of Events**

**2.1 Basic Flow**

* User accesses website
* User clicks on “login” button
* User enters:
  + Username
  + Password
* System validates the entered information to ensure it is correct
* User is signed in
* System recognizes login information
* System assigns access privileges to user based on if they are staff, parent, student, or volunteer

**2.2 Alternative Flows**

* <User Forgot User Name/Password>
  + System will prompt user to enter in security question entered in Create Secure Login Credentials Stage
  + If answer is correct, the system sends login information to user’s email address
  + If answer is incorrect, the system will prompt user to enter security question again
* < User Enters in Incorrect Information>
  + System will provide message that login information is incorrect
  + System will prompt user to re-enter information
  + User re-enters information
* <User does not have account created>
  + System will prompt user to create account
  + See “Create Account” use case

**3. Special Requirements**

* None

**4. Pre-conditions**

* Required pre-conditions:
  + Website must be accessible
  + User must have account created
* < Pre-condition One >
  + Website must be accessible and properly functioning for user to log in
* < Pre-condition Two >
  + This use case occurs after a user has created an account.

**5. Post-conditions**

* Post-condition are as follows:
  + Successful login
  + User not signed in
* < Post-condition One>
  + User is successfully logged in and able to view website with appropriate access controls.
* < Pre-condition Two >
  + User is unable to log into website based on incorrect information entered.

**6. Extension Points**

None

**UC 45- Manage Account**

**1.1 Brief Description**

Once logged in, website allows users to update their account information stored.

**2. Flow of Events**

**2.1 Basic Flow**

* User navigates to “edit account information” on website
* Website displays user information currently stored in edit mode
* User makes changes to user information
* User clicks “Save” button
* System validates information entered is accurate
* System makes changes to account information to reflect changes
* Website provides confirmation message to user

**2.2 Alternative Flows**

* + <User enters invalid information>
    - If the system determines that the new information entered is invalid, the following occurs:
      * The system provides an error message which says:
        + Information entered is invalid, which could include:
      * Missing information
      * Username already exists in system
      * Password is missing essential information such as:
        + Minimum characters
        + Minimum length
      * Highlights which information entered is incorrect
      * States why information entered is incorrect
    - User makes changes to information to correct mistakes
    - User re-submits information with changes
* <User cancels request>
  + User navigates to “edit account information” on website
  + Website displays user information currently stored in edit mode
  + User makes changes to user information
  + User clicks “Cancel” button

**3. Special Requirements**

* None

**4. Pre-conditions**

* Required pre-conditions:
  + User is logged into website
* < Pre-condition One >
  + User must be logged into website with correct credentials before making changes to their account.

**5. Post-conditions**

* Post-condition are as follows:
  + Successful update
  + Information not updated
* < Post-condition One>
  + The user changes and enters in the correct information to successfully update their account to reflect the changes.
* < Pre-condition Two >
  + User entered in invalid information so changes were unable to be made.

**6. Extension Points**

* None

**UC 46- Create secure login credentials**

1.1 B**rief Description**

System will have set login requirements to ensure accounts are secure.

**2. Flow of Events**

**2.1 Basic Flow**

* Staff sets requirements for login credentials such as:
  + Length
  + Characters
  + Numbers
* Staff accesses website code in HTML, JavaScript, or CSS code
* Staff writes requirements into code
* Requirements are applied to log in portion of website
* System ensures login credentials are authorized

**2.2 Alternative Flows**

* None

**3. Special Requirements**

* None

**4. Pre-conditions**

* Required pre-conditions:
  + Website must be accessible
  + Website must be created
* < Pre-condition One >
  + In order to set login credentials, the staff must be able to access the website to write the requirements into the code.
* < Pre-condition Two >
  + Website must have been created with proper coding in order for staff or web developers to access code and write password requirements.

**5. Post-conditions**

* Post-condition are as follows:
  + Successful application of requirements
* < Post-condition One >
  + The staff or web developer is able to successful update the web coding to set password requirements. Now, when staff or other users creates an account on the website, they will be prompted to enter the set requirements. (See Create Account use case)

**6. Extension Points**

* None

**UC 47- Track Event Revenues and Expenses**

**1.1 Brief Description**

Compile all receipts and expenditures pertaining to event in order to calculate profits or losses.

**2. Flow of Events**

**2.1 Basic Flow**

* Staff will access event revenues and expenses form in database
* Staff will select event to enter in information about
* Also, select date which event occurred
* Staff will enter in profits as received into form
* Staff will enter in expenses into form
* To scan in essential receipts:
  + Click “Scan Receipt” button
  + Upload image
* Click “Save”
* System stores information in database

**2.2 Alternative Flows**

* <Staff enters in incorrect information>
  + Staff can run query to look for uncommon numbers
  + Staff’s data entry will be peer reviewed to catch mistakes
* <Staff cancels request>
  + Staff will access event revenues and expenses form in database
  + Staff will select event to enter in information about
  + Staff clicks “Cancel” button
  + Information is not saved

**3. Special Requirements**

* None

**4. Pre-conditions**

* Required pre-conditions:
  + Event must occur
* < Pre-condition One >
  + In order to track the revenues and expenses of an event, the event must have taken place.

**5. Post-conditions**

* Post-condition are as follows:
  + Valid data is stored
* < Post-condition One >
  + Data is correctly entered in by staff to accurately reflect event profits.

**6. Extension Points**

* None

**UC 48- Announce Events to Invited Guests**

**1.1 Brief Description**

Staff will utilize the contact management system created to provide information to and inform students, parents, volunteers, donors, and community members of upcoming fundraising and outreach events.

**2. Flow of Events**

**2.1 Basic Flow**

* Staff will log onto Outlook
* Staff will draft email message about certain event
* Staff will type name of group to send email to
* Staff clicks “Send” button
* Email will be sent to individuals whose emails are included in list

**2.2 Alternative Flows**

* <Staff cancels request>
  + Staff logs onto Outlook
  + Staff drafts email message about certain event
  + Staff selects previously create group to send email to
  + Staff clicks “Discard” button

**3. Special Requirements**

* None

**4. Pre-conditions**

* Required pre-conditions:
  + Must have internet connection
  + Groups must have been previously created
* < Pre-condition One>
  + Since Outlook is a website, the staff will need a valid internet connection to log on and access email.
* < Pre-condition One >
  + Groups must have been previously set up in Outlook in order to communicate to groups of individuals. See use case 36 for creating groups

**5. Post-conditions**

* Post-condition are as follows:
  + Email is sent
  + Email is not sent
* < Post-condition One >
  + Email is sent to every individual included in the group with content the staff drafted.
* < Post-condition One>
  + Email is not sent due to staff canceling the request.

**6. Extension Points**

* None

**UC 49- Track Event Participation**

**1.1 Brief Description**

Staff will compile data pertaining to event attendance into database in order to track event participation.

**2. Flow of Events**

**2.1 Basic Flow**

* Staff obtains attendance record pertaining to event
* Staff accesses event participation form in database
* Staff selects event to enter information about
* Staff selects date which event occurred
* Staff selects individuals in attendance:
  + Staff
  + Volunteers
  + Students
  + Parents
* Staff enters names of other individuals in attendance not listed
* Staff clicks “Save” button
* Data is saved on database to be accessible to anyone with access

**2.2 Alternative Flows**

* <User cancels request>
  + Staff accesses event participation form in database
  + Staff selects event to enter information about
  + Staff selects date which event occurred
  + Staff clicks “Cancel” button

**3. Special Requirements**

* None

**4. Pre-conditions**

* Required pre-conditions:
  + DBMS must be created
  + Event must have taken place
  + Data from event must have been taken
* < Pre-condition One >
  + Staff must have DBMS that has been created in order to log on and enter data about events.
* < Pre-condition Two >
  + In order to have data accessible about an event, the event must have occurred.
* < Pre-condition Three >
  + Staff must be onsite at event to take attendance. This attendance data will then be transferred to another staff member for recording purposes.

**5**. **Post-conditions**

* Post-condition are as follows:
  + Successful update
  + Unsuccessful update
* < Post-condition One >
  + Data is entered correctly to provide for a successful update of data.
* < Post-condition One >
  + Data is entered incorrectly to provide for an unsuccessful update of data.

**6. Extension Points**

* None

**UC 50- Create Digital Sign-in System**

**1.1 Brief Description**

Paper-based sign-in system will be replaced with digital sign-in. This system will allow volunteers and students to sign in on laptops in order to keep track of attendance data.

**2. Flow of Events**

**2.1 Basic Flow**

* Students and volunteers open log in system on IPad
* Log in system displays page which includes a camera to read QR code
* Log in system also allows user to enter in number
* User enters or scans their respective QR code
* QR code is converted into csv code
* Csv code is sent to database to be recorded as a start time
* User later signs out by entering or scanning QR code on log in page
* QR code is converted into a csv code
* Csv code is sent to database to be recorded as an end time
* System records necessary information:
  + Time elapsed from start to end
  + Date
  + Program
* System adds data to database
* Systems sends message to notify user that QR code was successfully received and recorded
* Staff logs onto website
* In Staff portal, staff user navigates to volunteer/student management page
* Staff clicks on “Run Report”
* Staff selects which user type to run report for:
  + Student
  + Volunteer
* Staff choses date range to run report
* Staff selects “Generate Report” button
* System pulls data from database on user type

**2.2 Alternative Flows**

* <Staff Cancels Request>
  + Staff logs onto website
  + In Staff portal, staff user navigates to volunteer/student management page
  + Staff clicks on “Run Report”
  + Staff selects which user type to run report for:
    - Student
    - Volunteer
  + Staff choses date range to run report
  + Staff selects “Cancel” button

**3. Special Requirements**

* None

**4. Pre-conditions**

* Required pre-conditions:
  + DBMS must be created
* < Pre-condition One >
  + Staff must have a working DBMS to upload information and access it.

**5. Post-conditions**

* Post-condition are as follows:
  + Successful update
  + Unsuccessful update
* < Post-condition One>
  + Database is able to be successfully updated to reflect new attendance information
* < Post-condition One >
  + Database is unable to retrieve information due to a connection error

**6. Extension Points**

* None

**UC-51 Delete Student Record**

**1.1** **Brief Description**

Delete a student record

**2.** **Flow of Events**

**2.1** **Basic Flow**

* Access Staff Portal:
* Choose Student
* System will display:
  + student info page
  + Edit button
  + Delete button
* User will click Delete
* System will show Message Box
  + “Delete record?”
  + “Confirm” button
  + “Cancel” button
* User clicks the “confirm” button
* System updates student records
* System returns to student list

**2.2 Alternative Flows**

* Volunteer clicks “Cancel”
* System terminates the deletion
* System returns to student records

**3.** **Special Requirements**

* None

**4.** **Pre-conditions**

**4.1** **Login**

User must be logged onto the system

**5.** **Post-conditions**

System will display updated list of student records

**6.** **Extension Points**

None

**UC-52 Delete Volunteer Record**

**1.1** **Brief Description**

Delete volunteer information

**2.** **Flow of Events**

**2.1** **Basic Flow**

* Access Staff Portal:
* Choose Volunteer
* System will display:
  + Volunteer info page
  + Edit button
  + Delete button
* User will click Delete
* System will show Message Box
  + “Delete record?”
  + “Confirm” button
  + “Cancel” button
* User clicks the “confirm” button
* System updates volunteer records
* System returns to volunteer list

**2.2 Alternative Flows**

* Staff clicks “Cancel”
* System terminates the deletion
* System returns to volunteer records

**3.** **Special Requirements**

* None

**4.** **Pre-conditions**

**4.1** **Login**

User must be logged onto the system

**5.** **Post-conditions**

System will display updated list of volunteer records

**6.** **Extension Points**

None

**UC-53 Delete Donor Record**

**1.1** **Brief Description**

Delete donor information record

**2.** **Flow of Events**

**2.1** **Basic Flow**

* Access Staff Portal:
* Choose Donor
* System will display:
  + Donor info page
  + Edit button
  + Delete button
* User will click Delete
* System will show Message Box
  + “Delete record?”
  + “Confirm” button
  + “Cancel” button
* User clicks the “confirm” button
* System updates donor records
* System returns to donor list

**2.2 Alternative Flows**

* Staff clicks “Cancel”
* System terminates the deletion
  + System returns to donor records

**3.** **Special Requirements**

* None

**4.** **Pre-conditions**

**4.1** **Login**

User must be logged onto the system

**5.** **Post-conditions**

System will display updated list of donor records

**6.** **Extension Points**

None

# 

# 

# **UC54: Create Staff Record**

# **1.1 Brief Description**

# Create records containing personal information of Staff members within Adelante.

# **2. Flow of Events**

# **2.1 Basic Flow**

# On portal, administrative staff user clicks to open up a form for entering new staff information

# System displays staff information form

# Staff information form will include boxes to enter:

# Staff name

# Staff address

# Staff City

# Staff State

# Staff Zipcode

# Staff position

# Staff email

# Staff phone number

# Staff Description

# Staff Start Date

# Staff End Date

# Staff Wage

# Also present will be:

# “Submit” button

# “Cancel” button

# Administrative staff will enter all information into form boxes

# Staff will click “submit”

# System will send information to database, and will record as a new record

# System will send back a message saying “new record created”

# **2.2 Alternative Flows**

# None

# **3. Special Requirements**

# None

# **4. Pre-conditions**

# **4.1 < Login >**

# User must log into system.

# **4.2 < Form>**

# Staff information form must be available.

# **5. Post-conditions**

# **5.1 < System Display >**

# System will display new, blank Staff form after submission

# **6. Extension Points**

# None

# 

# **UC55: Modify Staff Record**

# **1.1 Brief Description**

# Modify Staff record to change/update personal staff member information

# **2. Flow of Events**

# **2.1 Basic Flow**

# On portal, administrative staff searches for a particular staff member, and then clicks to open editable staff information form.

# Web interface component opens up a page displaying (in an editable fashion) a form of current attributes of a particular staff record including:

# o Staff name

# o Staff address

# o Staff City

# o Staff State

# o Staff Zipcode

# o Staff position

# o Staff email

# o Staff phone number

# o Staff Description

# o Staff Start Date

# o Staff End Date

# o Staff Wage

# Also present are two buttons:

# Save Changes

# Cancel

# Administrative staff can choose to click and modify any of the attributes

# After administrative staff is satisfied, they will click the submit button

# Changes are sent through the system and updated in the database

# A message is displayed by the system to the administrative staff user, saying that the changes were successfully made

# **2.2 Alternative Flows**

# Staff clicks “Cancel”

# Modification interface terminated

# **3. Special Requirements**

# None

# **4. Pre-conditions**

# **4.1 Up and running**

# System needs to be running

# **4.2 Logged in**

# Administrative staff member must be logged onto the staff portal

# **5. Post-conditions**

# System will return to page to search for specific staff members.

# **6. Extension Points**

# None

# 

# **UC-56: Delete Staff Record**

# **1.1 Brief Description**

# Delete personal staff member information

# **2. Flow of Events**

# **2.1 Basic Flow**

# On portal, administrative staff searches for a particular staff member, and then clicks to open editable staff information form.

# System will display:

# Staff information form

# Edit button

# Delete button

# Administrative staff will click Delete

# System will show Message Box

# “Delete record?”

# “Confirm” button

# “Cancel” button

# Administrative staff clicks the “confirm” button

# System removes staff member from Staff Table in database

# System returns to staff list

# **2.2 Alternative Flows**

# Administrative staff clicks “Cancel”

# System terminates the deletion

# System returns to staff records

# **3. Special Requirements**

# None

# **4. Pre-conditions**

# **4.1 Login**

# User must be logged onto the system

# **5. Post-conditions**

# System will display updated list of staff records

# **6. Extension Points**

# None

# 

# 

# 

# **UC57: Create Board Member Record**

# **1.1** **Brief Description**

# Create records containing personal information of Board members within Adelante.

# 2. **Flow of Events**

# **2.1 Basic Flow**

# On portal, administrative staff user clicks to open up a form for entering new Board Member information

# System displays Board member information form

# Board member information form will include boxes to enter:

# o Board ID

# o Board member address

# o Board member City

# o Board member State

# o Board member Zip code

# o Board member position

# o Board member email

# o Board member phone number

# o Board member Description

# o Board member Start Date

# o Board member End Date

# o

# Also present will be:

# o Submit button

# o Cancel button

# Staff will enter all information into form boxes

# Staff will click “submit”

# System will send information to database, and will record as a new record

# System will send back a message saying “new record created”

# **2.2 Alternative Flows**

# None

# **3. Special Requirements**

# **None**

# **4. Pre-conditions**

# **4.1 < Login >**

# User must log into system.

# **4.2 < Form>**

# Staff information form must be available.

# **5. Post-conditions**

# **5.1 < System Display >**

# System will display new, blank form after submission

# **6. Extension Points**

# None

# 

# **UC58: Modify Board Member Record**

# 1.1 **Brief Description**

# Modify records containing personal information of Board members within Adelante.

# 2. **Flow of Events**

# **2.1 Basic Flow**

# On portal, administrative staff user clicks to open up a form for modify new Board Member information

# Web interface component opens up a page displaying (in an editable fashion) a form of current attributes of a particular Board Member record including:

# o Board ID

# o Board member address

# o Board member City

# o Board member State

# o Board member Zip code

# o Board member position

# o Board member email

# o Board member phone number

# o Board member Description

# o Board member Start Date

# o Board member End Date

# Also present are two buttons:

# Save Changes

# Cancel

# Administrative staff can choose to click and modify any of the attributes

# After administrative staff is satisfied, they will click the submit button

# Changes are sent through the system and updated in the database

# A message is displayed by the system to the administrative staff user, saying that the changes were successfully made

# **2.2 Alternative Flows**

# None

# **3. Special Requirements**

# None

# **4. Pre-conditions**

# **4.1 < Login >**

# User must log into system.

# **4.2 < Form>**

# Staff information form must be available.

# **5. Post-conditions**

# **5.1 < System Display >**

# System will display new, blank form after submission

# **6. Extension Points**

# None

# 

# **UC59: Delete Board Member Record**

# 1.1 **Brief Description**

# Delete records containing personal information of Board members within Adelante.

# 2. **Flow of Events**

# **2.1 Basic Flow**

# On portal, administrative staff searches for a particular Board member, and then clicks to open editable member information form.

# System will display:

# Board member information form

# Edit button

# Delete button

# Administrative staff will click Delete

# System will show Message Box

# “Delete record?”

# “Confirm” button

# “Cancel” button

# Administrative staff clicks the “confirm” button

# System removes board member from Board member Table in database

# System returns to board member list

# **2.2 Alternative Flows**

# Administrative staff clicks “Cancel”

# System terminates the deletion

# System returns to Board member records

# **3. Special Requirements**

# None

# **4. Pre-conditions**

# **4.1 < Login >**

# User must log into system.

# **4.2 < Form>**

# Staff information form must be available.

# **5. Post-conditions**

# **5.1 < System Display >**

# System will display new, blank form after submission

# **6. Extension Points**

# None

# 

# **UC-60**

## 1.1 **Brief Description**

Create report from Board Member attendance to classify Board Member as active/inactive in meeting attendance. .

# 2. **Flow of Events**

## 2.1 **Basic Flow**

· Staff presses button to view report of Board Member attendance and classification.

· System counts all meetings Board Members has attended.

· If the count / total = >50%, calculated field shows a value of “active”

· If the count / total = <50%, calculated field shows a value of “inactive”

· System returns to staff user a report including the calculated field.

System includes an [X] to exit out of report.

## **2.2 Alternative Flows**

None

# **3. Special Requirements**

None

# **4. Pre-conditions**

## **4.1 Logged in**

Staff must be logged onto the staff portal

## **4.2 Up and running**

Both systems need to be functioning

## **4.3 Board ID**

The Board Member must have a unique member ID that already exists in the attendance and Board Member tables

**UC-61**

## **1.1** **Brief Description**

Create sign-in system to record guest attendance at events.

# **2.** **Flow of Events**

## **2.1** **Basic Flow**

* Two iPads displayed:
  + one for login credentials
    - Students, parents, volunteers, board members and staff will enter in following information:
      * Username
      * Password
    - System records their presence
  + one for guests
    - Guests will enter following information:
      * First name
      * Last name
      * E-mail
      * Phone number
    - System records guest presence

**2.2 Alternative Flows**

* In case of guest refusal to input information digitally, paper guest book will be available
  + Individual will enter following:
    - First name
    - Last name
    - E-mail
    - Phone number

# **3. Special Requirements**

None

# **4. Pre-conditions**

## **4.1 Attendance recording**

* Attendance recording must be accessible on both iPads

## **4.2 Up and running**

* Systems need to be functioning

## **4.3 Usernames and passwords**

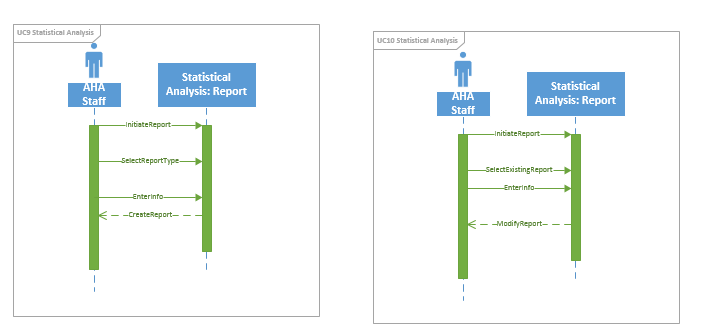
* Students, parents, volunteers, board members and staff must have usernames and passwords

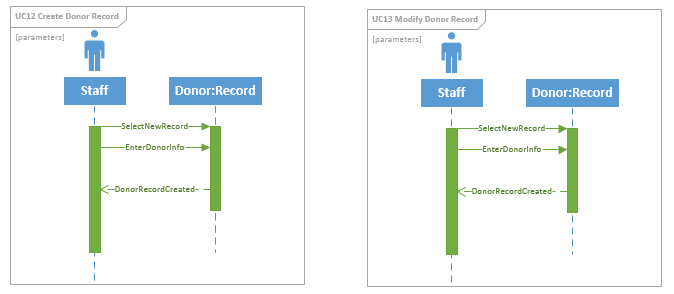
# 

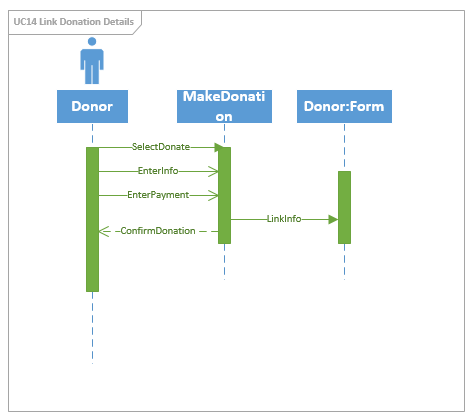
# 

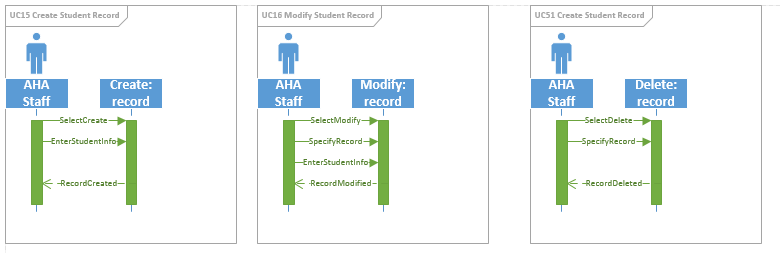
# Sequence Diagrams

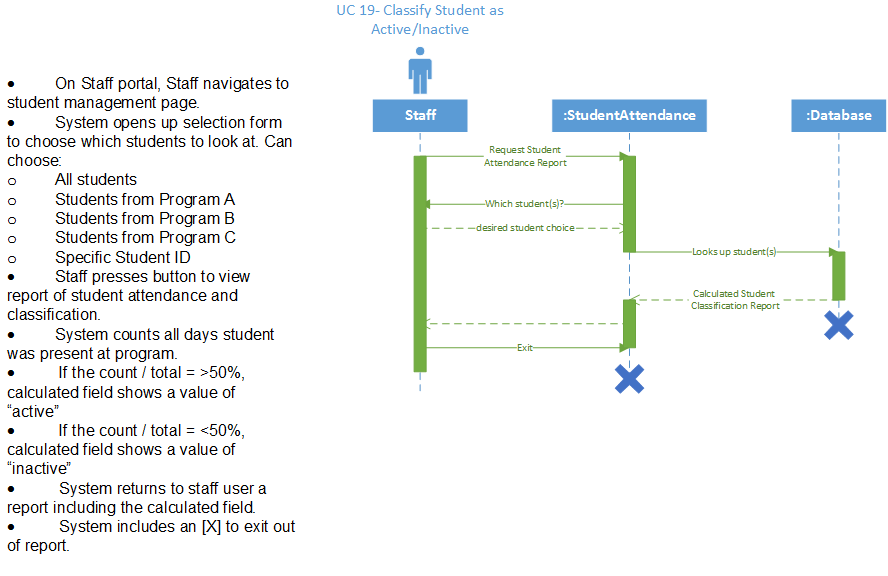
*The figures below are a type of interaction diagram. They illustrate the objects that are involved in the Use Cases and the messages that are sent and received by the system.*

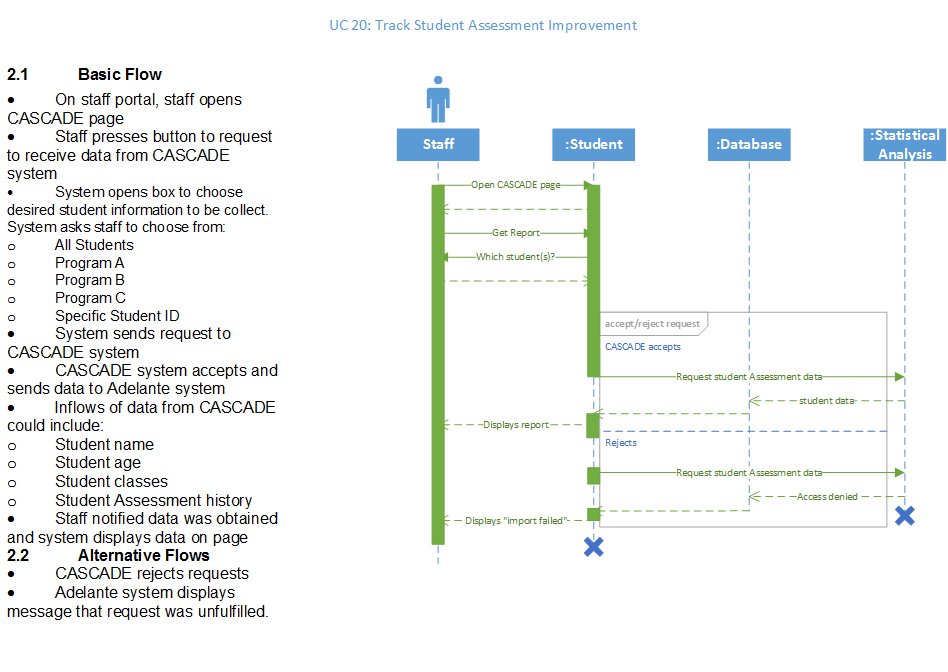


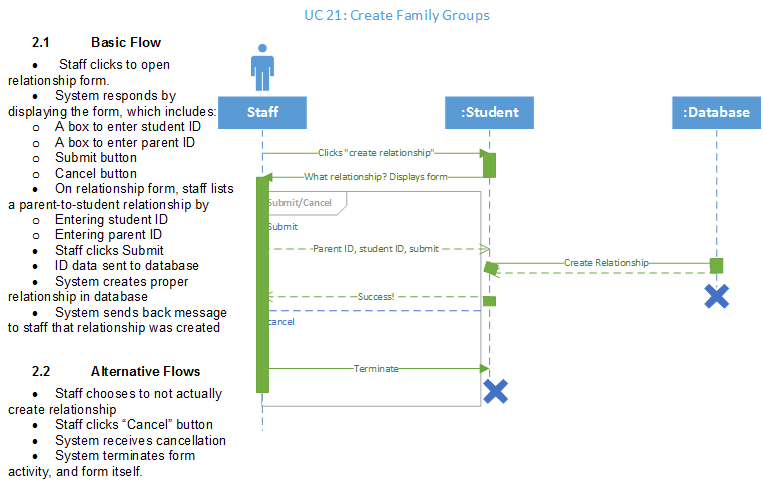


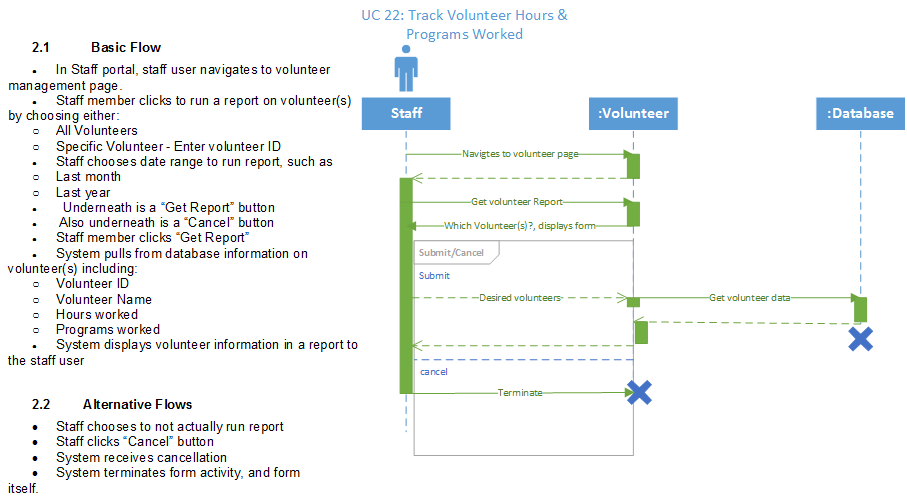


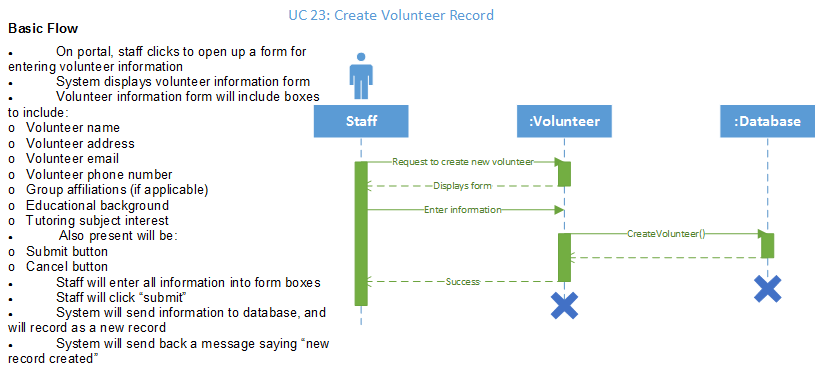


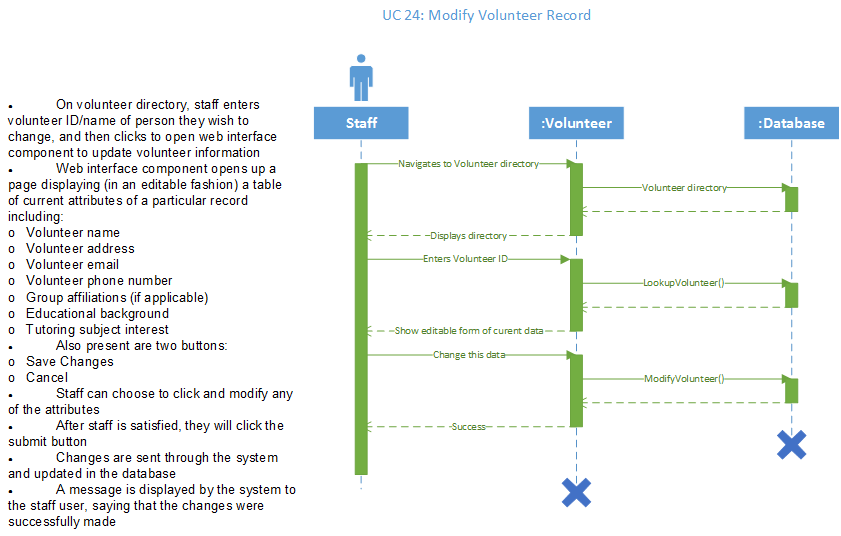


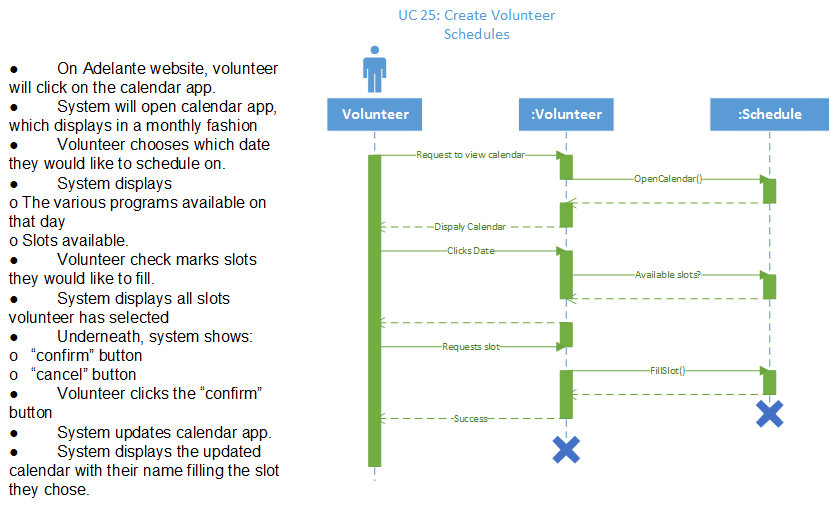


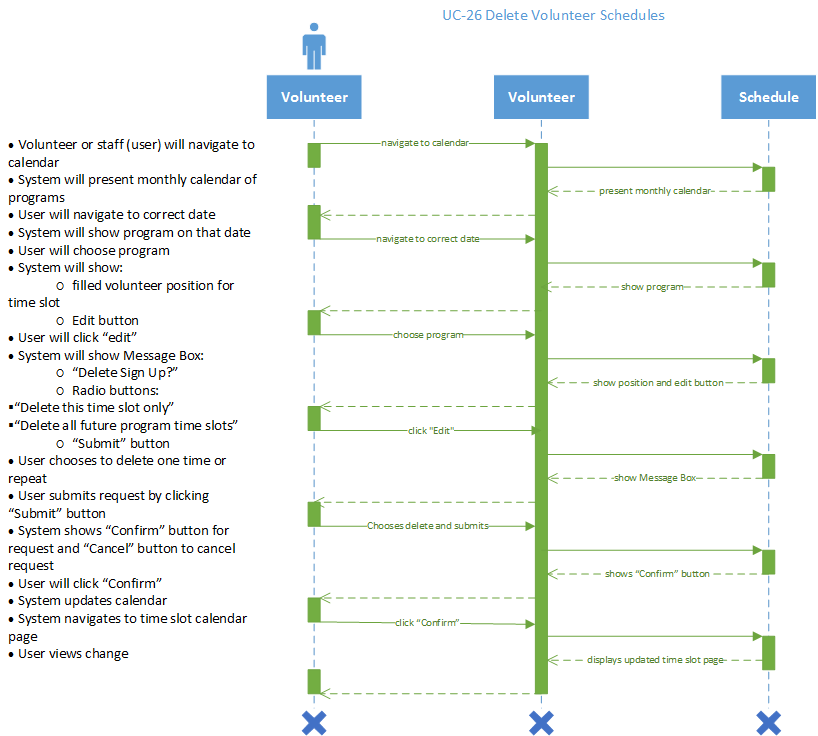


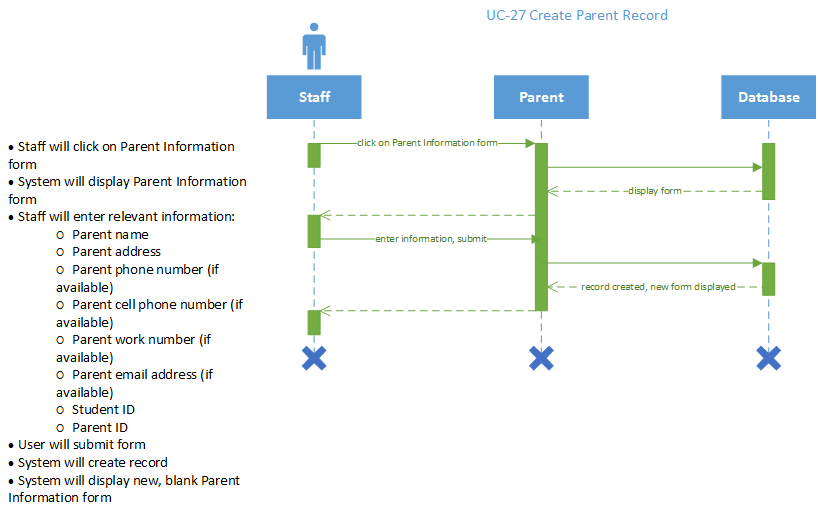


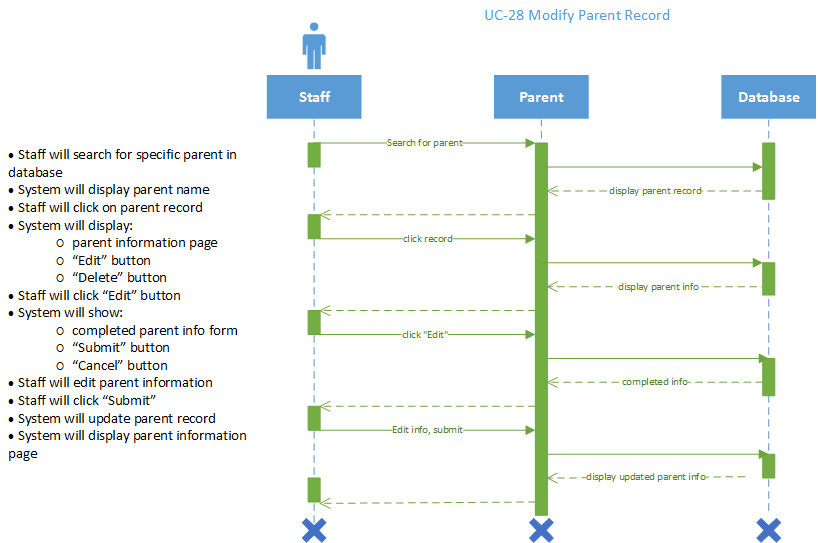


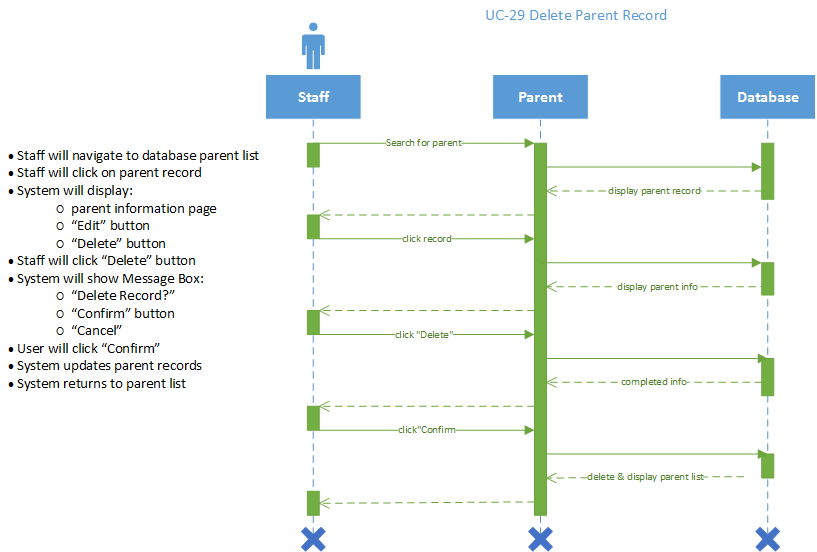


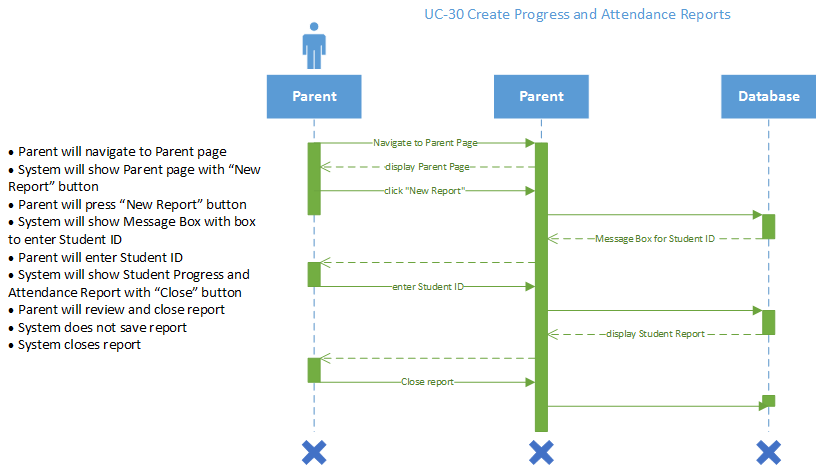


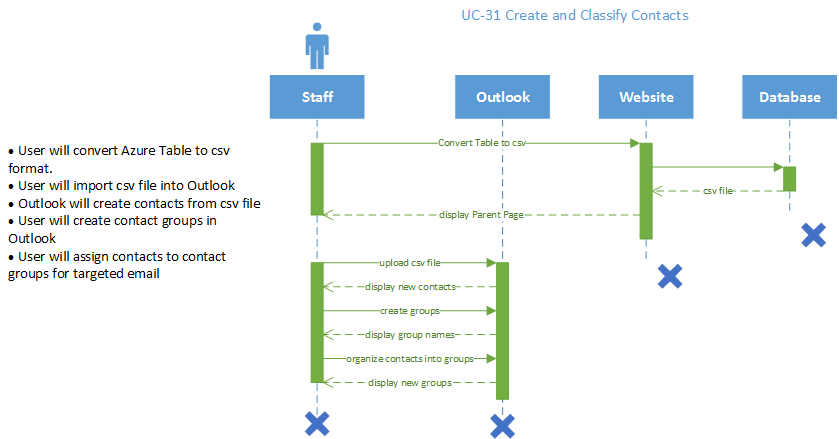


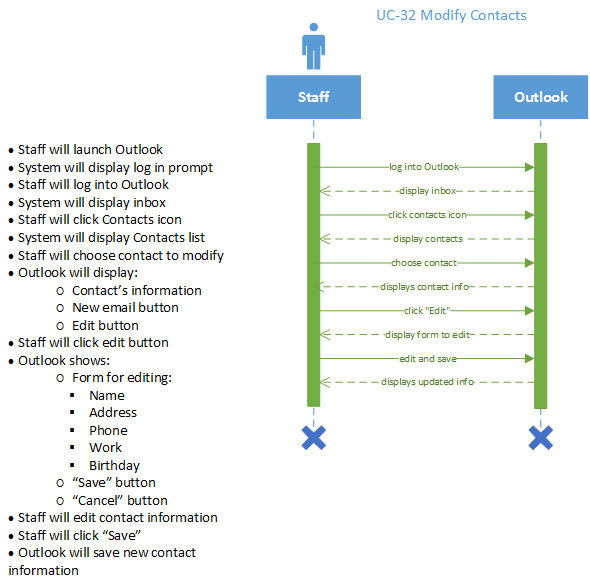




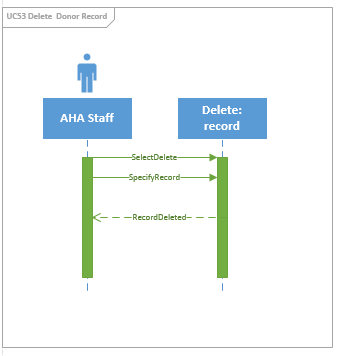






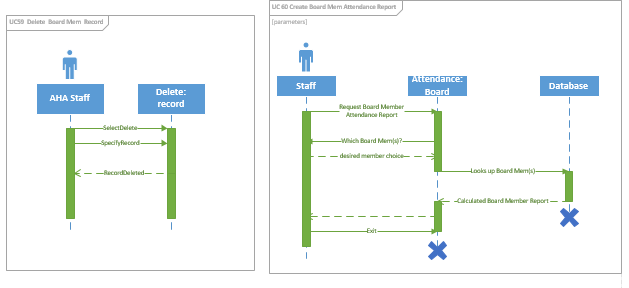


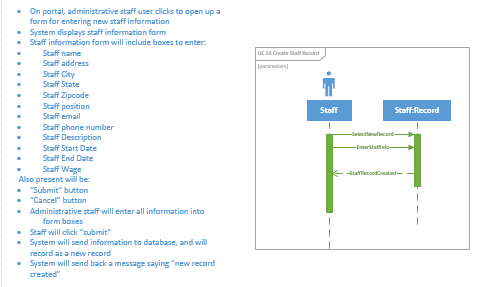


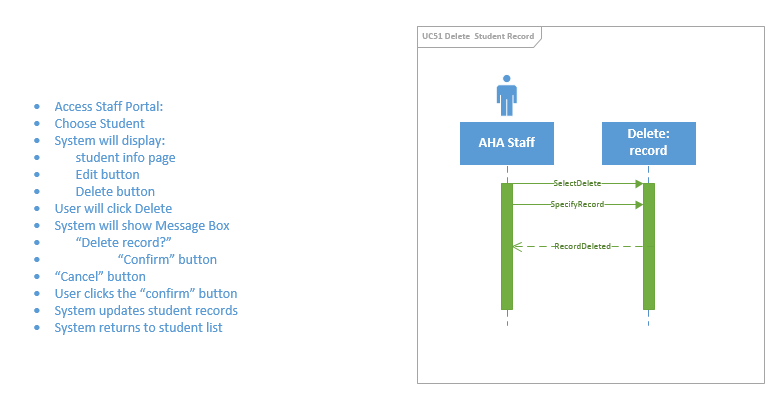
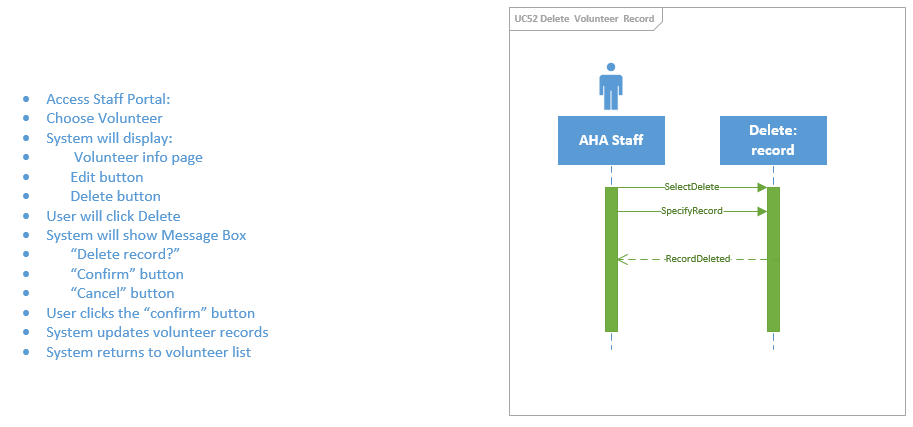


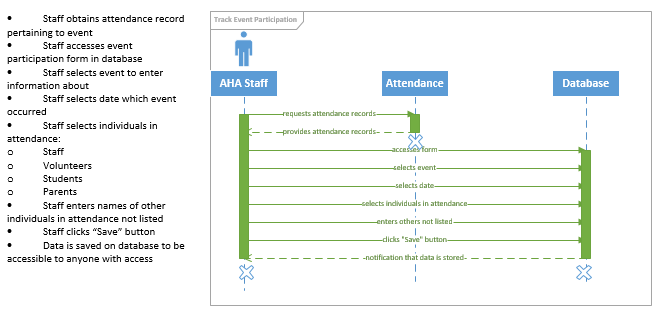
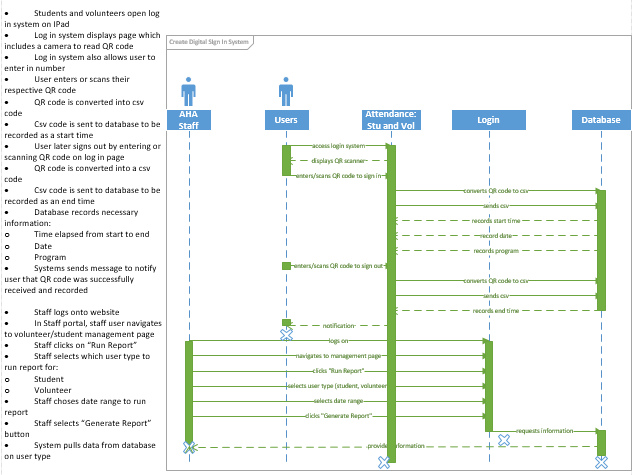
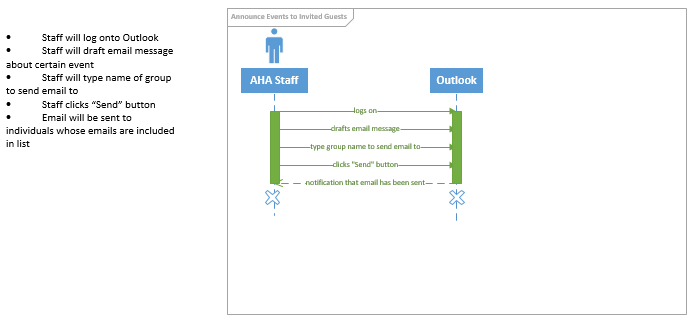
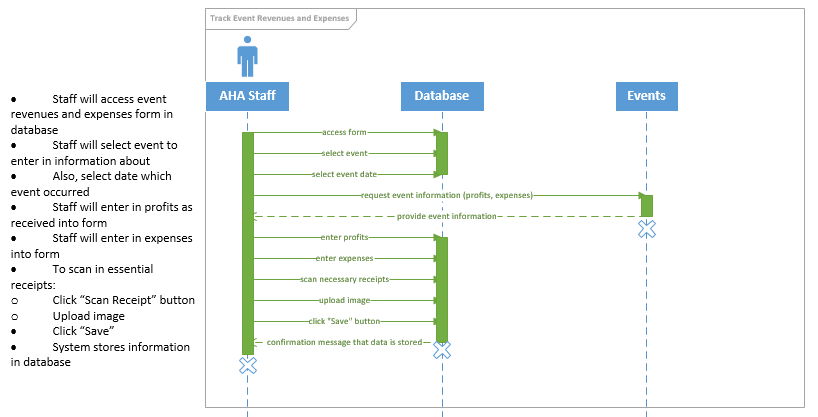
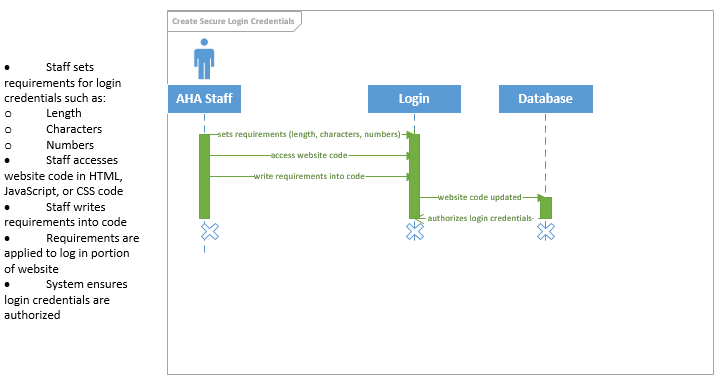
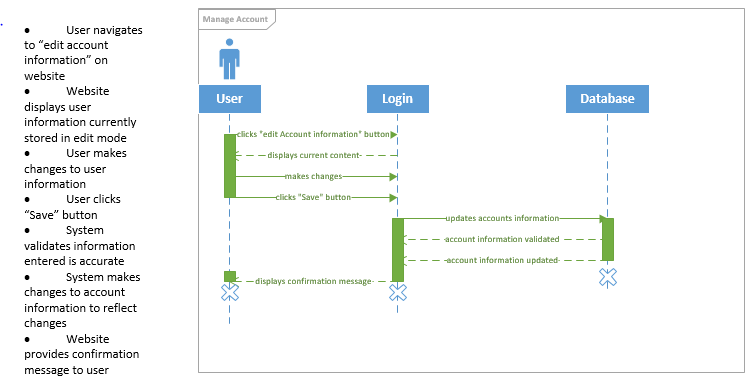
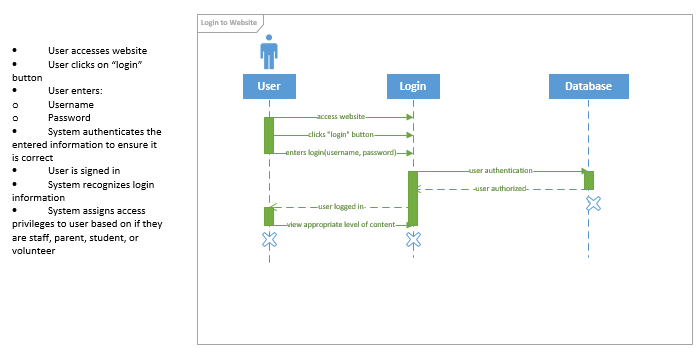
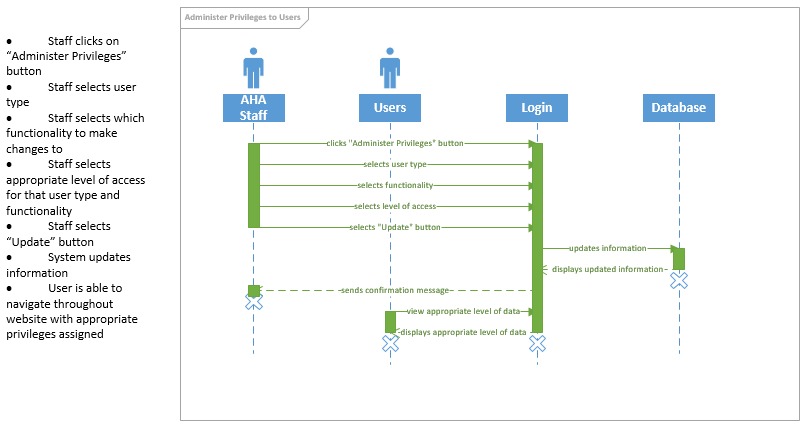
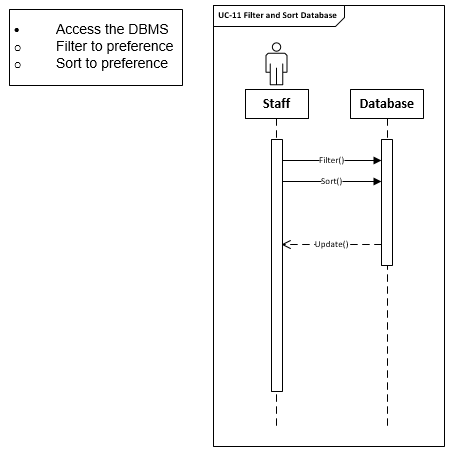
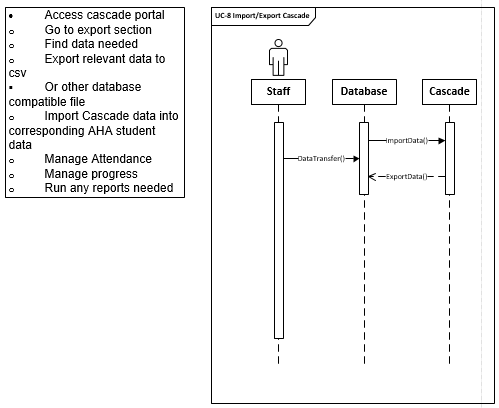
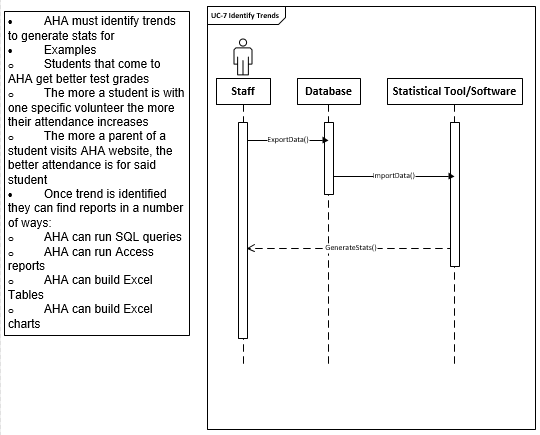
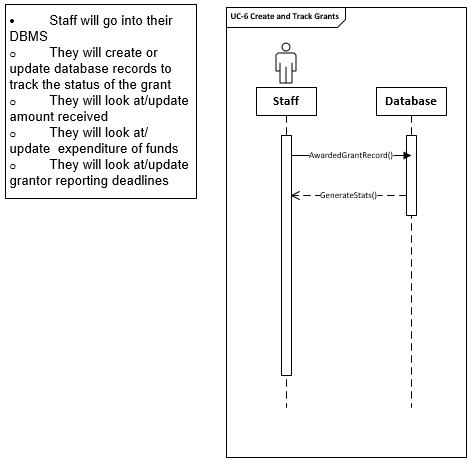
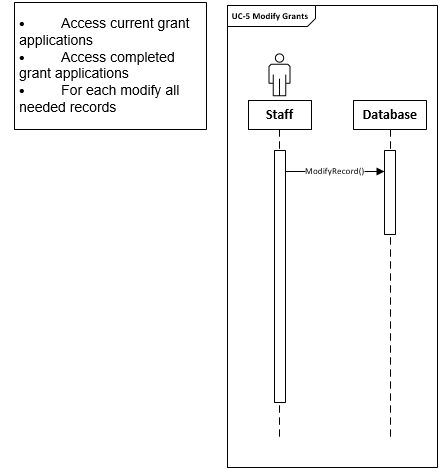
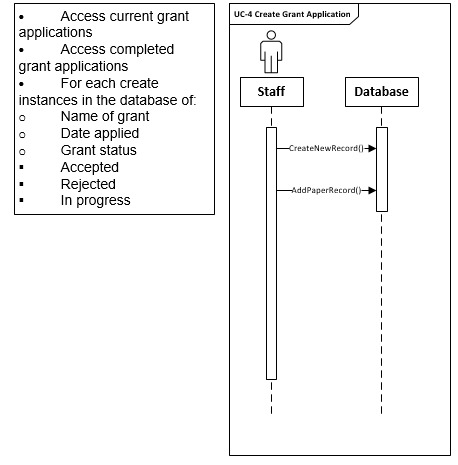
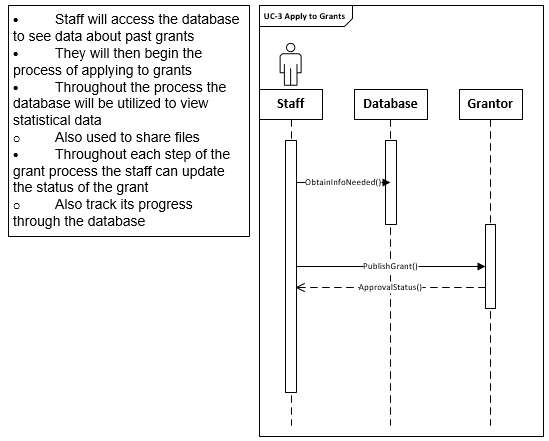


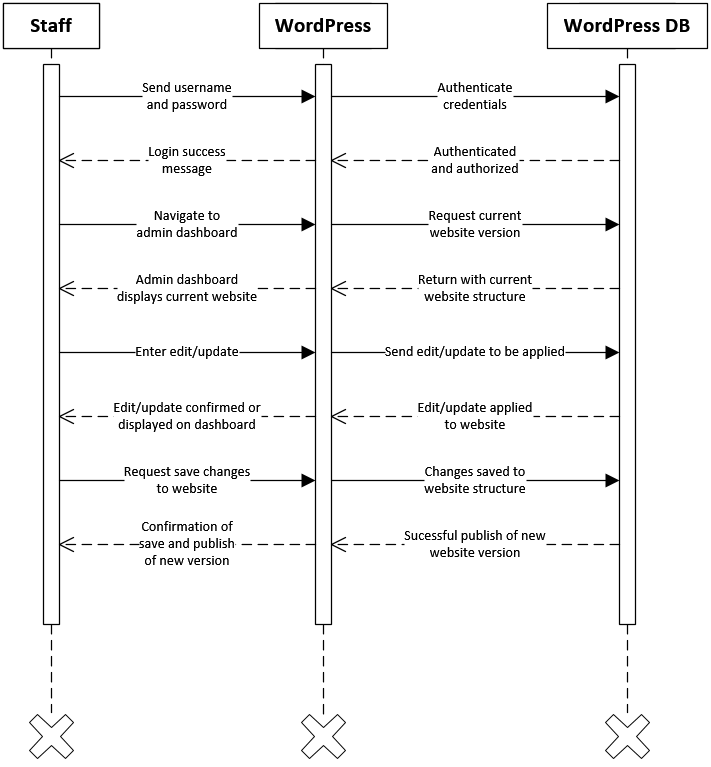






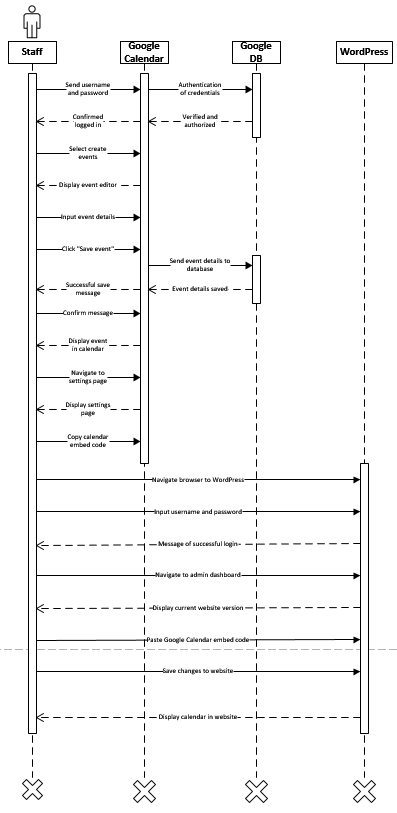
  






**UC33 Easy website updates and edits:**

* Create administrative account on WordPress
* Create AHA website using WordPress
* Staff can now use simple WordPress tools to make updates and edits to site as necessary



**UC34 Create calendar events and programs:**

Create a Google account for Adelante Staff

Staff login to Adelante Google account

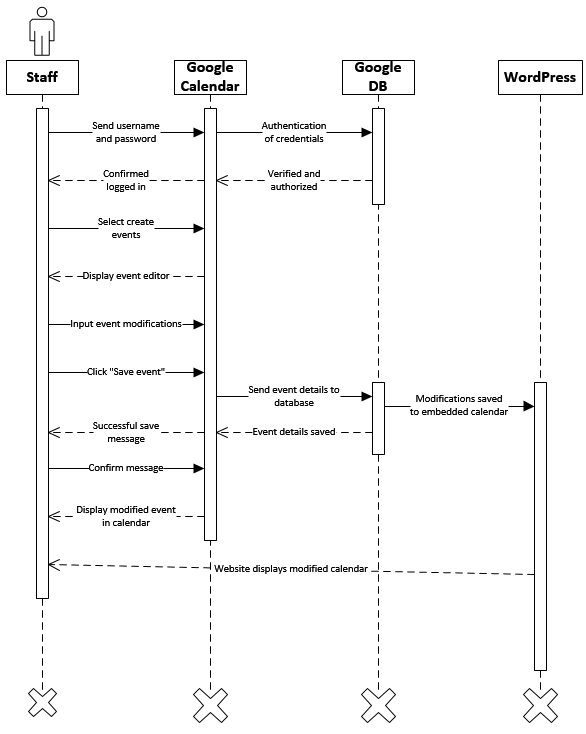
Staff creates calendar to track events and programs

Copy calendar embed code

Log in to WordPress

Paste into website through WordPress

Save and publish changes



**UC 35 Modify calendar events and programs**

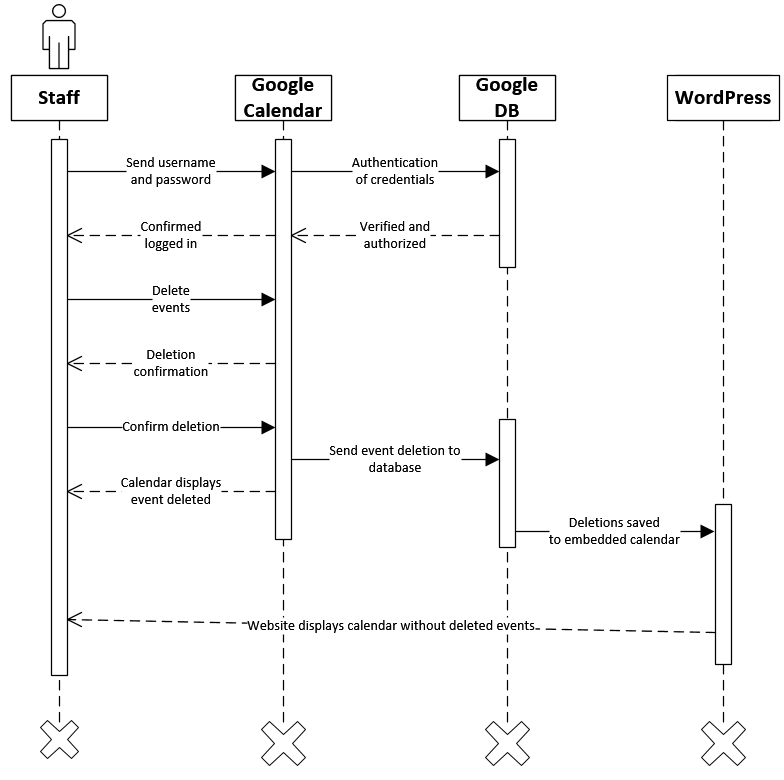
Log in to Adelante Google account

Navigate to Google calendar

Access calendar storing events and programs

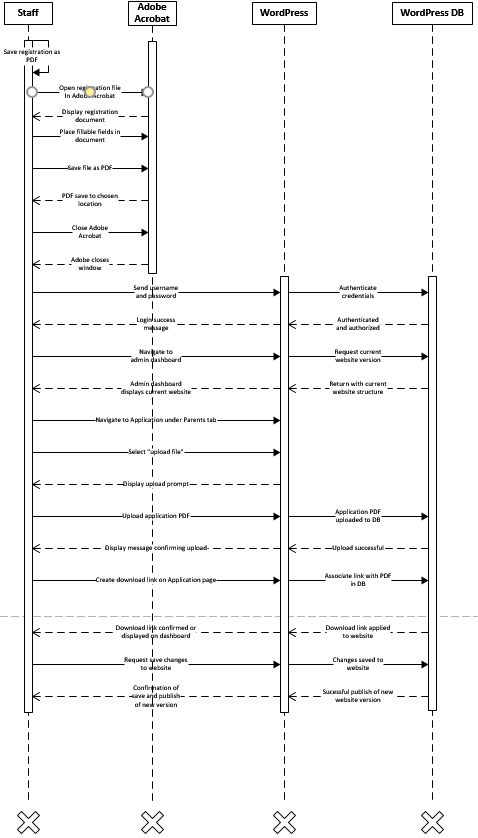
Make necessary modifications to events and programs

Save changes.



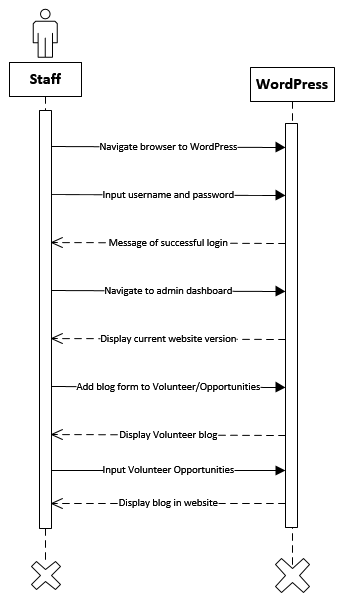
**UC36 Delete calendar events and programs**

* Log in to Adelante Google account
* Navigate to Google calendar
* Access calendar storing events and programs
* Make necessary deletions of events and programs
* Save changes.



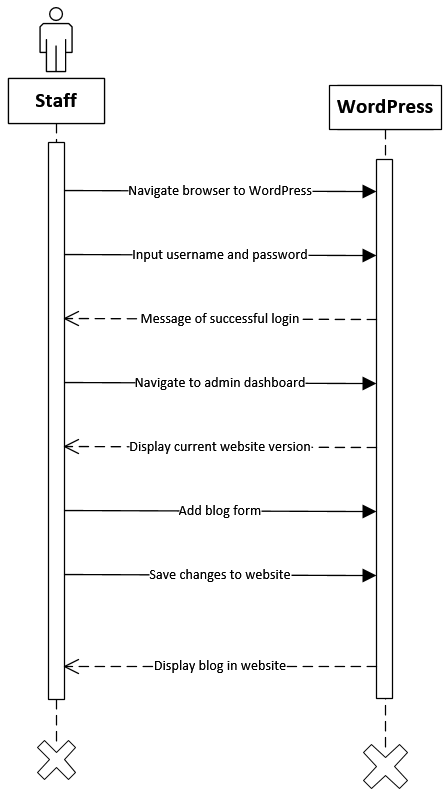
**UC 37 Create registration forms in PDF format that can be edited and printed**

* Staff opens registration form file in Adobe Acrobat DC
  + If only physical copy exists, use scanner to create digital copy and open in Adobe Acrobat DC
* Adobe Acrobat DC will automatically place fillable fields in place of static fields detected
* Add, delete, or modify fields as necessary.
* Save file as PDF
* Staff logs in to WordPress
* Use WordPress administrative functions to upload PDF to website
* Create download link with WordPress
* Save and publish website



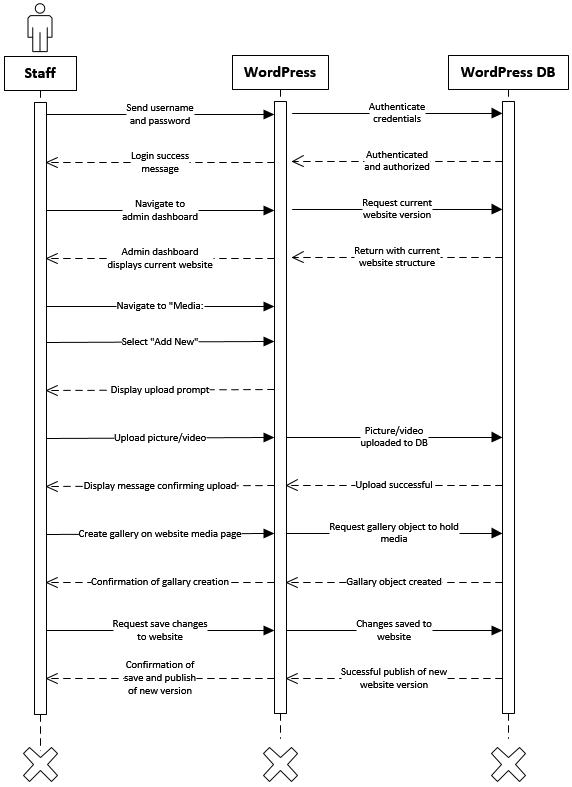
**UC38 Display volunteering opportunities on website**

* Staff has available volunteer positions that must be filled.
* Log in to WordPress for administrative access to website
* Navigate to Volunteers: Opportunities page
* Add opportunities to page.



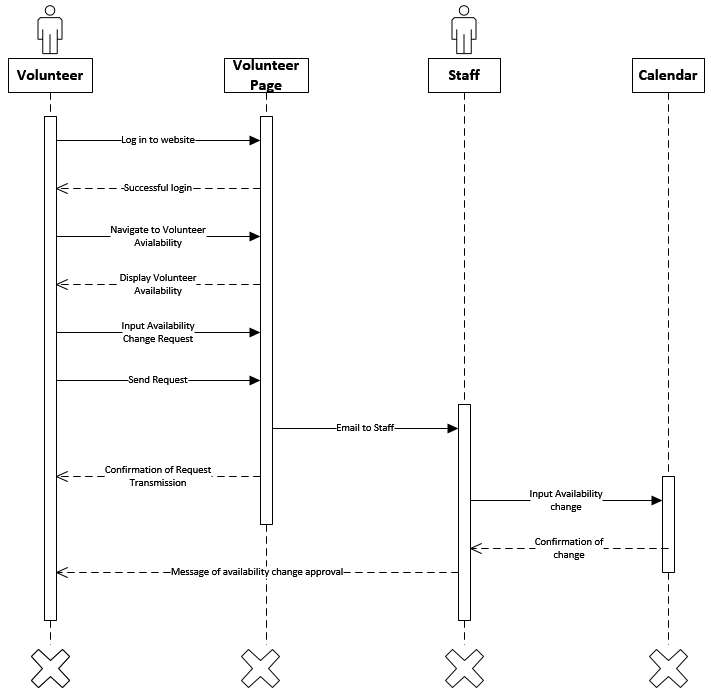
**UC 39 Create blog**

* Staff have information or program and events updates to share with site visitors.
* Using WordPress administrative level access to site, blog can be added.
* Blog can be modified as necessary with WordPress login.



**UC40 Enable picture and video sharing**

* Staff have pictures and video from events, programs, and other sources to share on website
* WordPress administrative level access to site provides access to uploading media.
* Media will be displayed on site front end to visitors

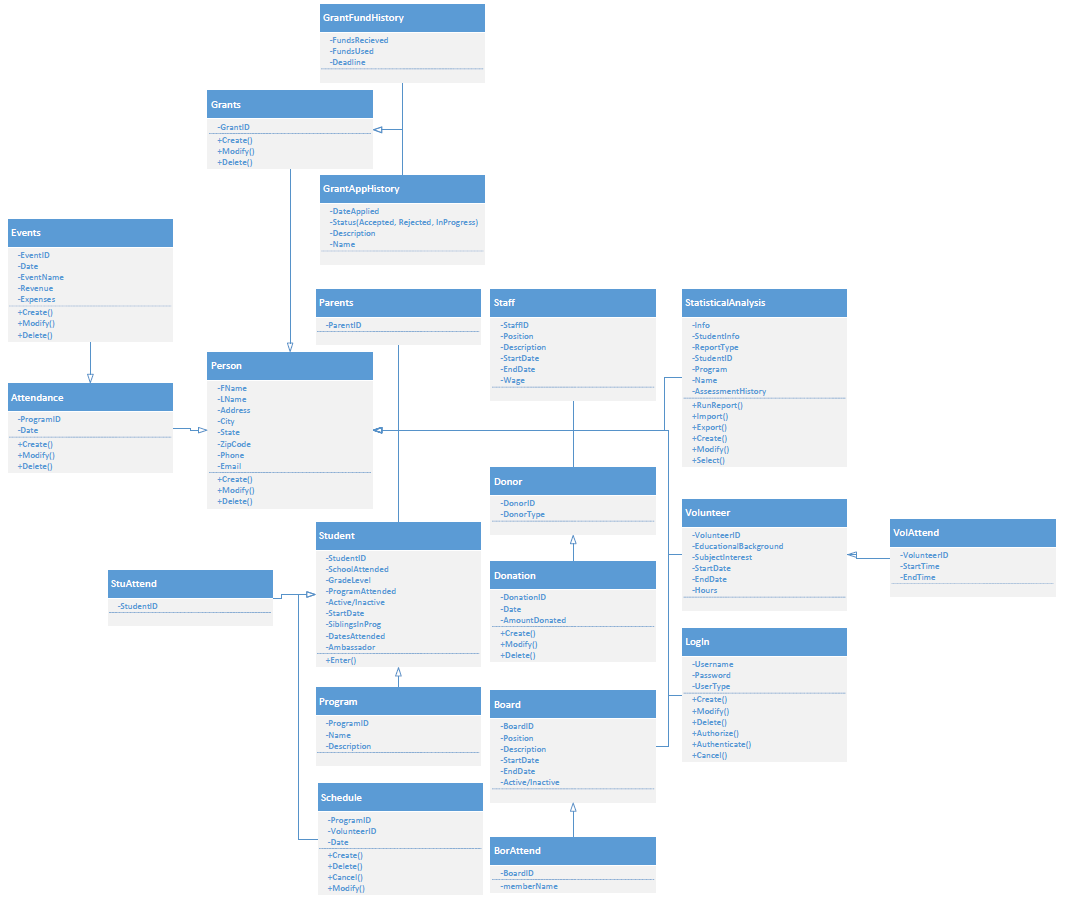


**UC41 Request change in volunteer availability**

* Volunteer has schedule conflict with existing availability
* Volunteer clicks on “Volunteers” tab to open drop down menu
* Volunteer selects “Availability” from drop down menu
* Volunteer is redirected to Availability page
* Volunteer enters dates into “Dates affected” field using dropdown month and day choices
* Volunteer uses drop down “Reasons” menu to select reason for change in availability.
* Volunteer clicks send
* Message is sent as email to Dustin’s email.
* Carbon copy of message is sent to email provided by volunteer on registration.
* Message box confirming transmission of message.
* Dustin reviews availability change for approval.

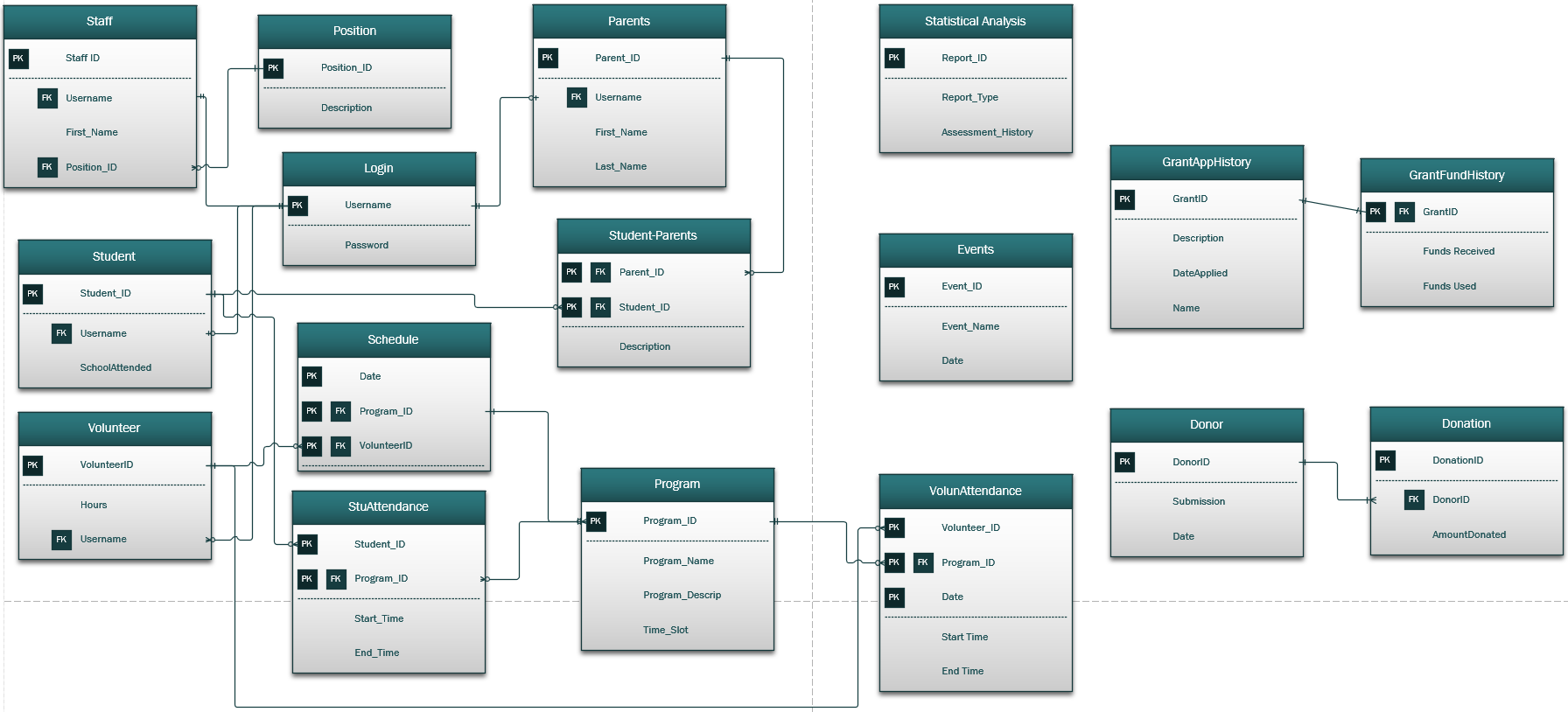
# Class Diagram

A class diagram is a static model that depicts classes along with the relationships among the classes. It is useful in illustrating how the system will store information and relate to other information stored in the system.



# Database Design and Data Definitions

The Database Design depicts the relational management of the system classes. This design is in third normal form, which displays the system classes in a highly structured and organized format. Third normal form eliminates partial and transitive dependencies. These dependencies can create data integrity issues. We combatted these issues by creating a table for each subclass within our class diagram.



Data Definitions represent the attributes that will be present in each table. They are designed in such a way that a database designer can create the structure to fit the needs of the table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Students** | | | | |
| **PK/FK** | **Attribute** | **Definition** | **Type** | **Size (in bytes)** |
| PK | **Student\_ID** | Unique student cell phone number | Alphanumeric | 10 |
|  | FName | Student first name | Alphanumeric | 50 |
|  | LName | Student last name | Alphanumeric | 50 |
|  | StreetAddress | Student Home Street Address | Alphanumeric | 30 |
|  | City | Student Home City | Alphanumeric | 20 |
|  | State | Student Home State | Alphanumeric | 2 |
|  | ZIP | Student Home Zipcode | Alphanumeric | 10 |
|  | Phone | Student Primary Phone Number | Alphanumeric | 13 |
|  | Email | Student primary email address | Alphanumeric | 50 |
|  | SchoolAttended | School student currently attends | Alphanumeric | 50 |
|  | GradeLevel | Student's current grade in school | Alphanumeric | 2 |
|  | ProgramAttented | Program student currently attends | Alphanumeric | 50 |
|  | Active/Inactive | Student classification in Adelante | Binary | 1 |
|  | StartDate | Date Student joined Adelante | Date | 10 |
|  | SiblingsInProg | Does student have siblings outside of program? | Binary | 1 |
|  | DatesAttended | Dates the student attended the program(s) | Date | 10 |
| FK | Username | Username to log in | Alphanumeric | 50 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parents** | | | | |
| **PK/FK** | **Attribute** | **Definition** | **Type** | **Size (in bytes)** |
| PK | Parent\_ID | Unique parent cell phone number | Alphanumeric | 10 |
|  | FName | Parent first name | Alphanumeric | 50 |
|  | LName | Parent last name | Alphanumeric | 50 |
|  | Street Address | Parent home street name | Alphanumeric | 30 |
|  | City | Parent home city | Alphanumeric | 20 |
|  | State | Parent home state | Alphanumeric | 2 |
|  | ZIP | Parent ZIP Code | Alphanumeric | 10 |
|  | Phone | Parent primary phone number | Alphanumeric | 13 |
|  | Email | Parent primary email address | Alphanumeric | 50 |
| FK | Username | Username to log in | Alphanumeric | 50 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Student / Parent** | | | | |
| **PK/FK** | **Attribute** | **Definition** | **Type** | **Size (in bytes)** |
| PK/FK | **Parent ID** | A unique number identifying parents | Alphanumeric | 10 |
| PK/FK | Student ID | A unique number identifying students | Alphanumeric | 10 |
|  | Description | A short description of the relationship between students and parents | Alphanumeric | 50 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Staff** | | | | |
| **PK/FK** | **Attribute** | **Definition** | **Type** | **Size (in bytes)** |
| PK | Staff\_ID | Unique staff cell phone number | Alphanumeric | 10 |
|  | Position\_ID | The different types of staff positions | Alphanumeric | 25 |
|  | StartDate | Date hired | Date | 10 |
|  | EndDate | Date employment terminated | Date | 10 |
|  | Wage | Pay per hour | Currency | 10 |
|  | FName | Staff first name | Alphanumeric | 50 |
|  | LName | Staff last name | Alphanumeric | 50 |
|  | Street Address | Staff home street name | Alphanumeric | 30 |
|  | City | Staff home city | Alphanumeric | 20 |
|  | State | Staff home state | Alphanumeric | 2 |
|  | ZIP | Staff ZIP Code | Alphanumeric | 10 |
|  | Phone | Staff primary phone number | Alphanumeric | 13 |
|  | Email | Staff primary email address | Alphanumeric | 50 |
| FK | Username | Username to log in | Alphanumeric | 50 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Schedule** | | | | |
| **PK/FK** | **Attribute** | **Definition** | **Type** | **Size (in bytes)** |
| PK | Date | Date Scheduled | Datetime | 15 |
| PK FK | ProgramID | ID to identify program | Alphanumeric | 10 |
| PK/FK | VolunteerID | A unique ID identifying volunteers | Alphanumeric | 10 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Donor** | | | | |
| **PK/FK** | **Attribute** | **Definition** | **Type** | **Size (in bytes)** |
|  | FName | Donor First Name | Alphanumeric | 10 |
|  | LName | Donor last name | Alphanumeric | 50 |
|  | Address | Street address | Alphanumeric | 50 |
|  | City | City of residence | Alphanumeric | 30 |
|  | State | state of residence | Alphanumeric | 20 |
|  | ZipCode | Zip Code of residence | Alphanumeric | 2 |
|  | Phone | Phone number | Alphanumeric | 10 |
|  | Email | email address of donor | Alphanumeric | 13 |
| PK | DonorID | ID number to uniquely identify donor | Alphanumeric | 50 |
|  | DonorType | Type of donor: community member, etc. | Alphanumeric | 25 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Donation** | | | | |
| **PK/FK** | **Attribute** | **Definition** | **Type** | **Size (in bytes)** |
| FK | DonorID | ID number to uniquely identify donor | Alphanumeric | 50 |
| PK | DonationID | ID number to identify individual donation instances | Alphanumeric | 25 |
|  | Date | Date of donation | Date | 15 |
|  | AmountDonated | amount donated | Decimal | 20 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Volunteer** | | | | |
| **PK/FK** | **Attribute** | **Definition** | **Type** | **Size (in bytes)** |
| PK | **Volunteer\_ID** | Volunteer Identification code | Alphanumeric | 10 |
|  | FName | Donor First Name | Alphanumeric | 10 |
|  | LName | Donor last name | Alphanumeric | 50 |
|  | Address | Street address | Alphanumeric | 50 |
|  | City | City of residence | Alphanumeric | 30 |
|  | State | state of residence | Alphanumeric | 20 |
|  | ZipCode | Zip Code of residence | Alphanumeric | 2 |
|  | Phone | Phone number | Alphanumeric | 10 |
|  | Email | email address of donor | Alphanumeric | 13 |
|  | Education\_Background | Highest education level of volunteer | Alphanumeric | 25 |
|  | Subject\_Interest | Subject volunteer wishes to tutor | Alphanumeric | 50 |
|  | Start\_Date | Date volunteer joined Adelante | Date | 10 |
|  | End\_date | Date volunteer left Adelante | Date | 10 |
|  | Hours | Total hours calculated from attendance | Numeric | 5 |
|  | Program | Program volunteer attends | Alphanumeric | 50 |
| FK | UserName | Username to log in | Alphanumeric | 25 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Student Attendance** | | | | |
| **PK/FK** | **Attribute** | **Definition** | **Type** | **Size (in bytes)** |
| PK/FK | Program | Unique ID identifying programs | Alphanumeric | 10 |
|  | Date | current date | date | 15 |
| PK | **StudentID** | Unique student cell phone number | Alphanumeric | 10 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Volunteer Attendance** | | | | |
| **PK/FK** | **Attribute** | **Definition** | **Type** | **Size (in bytes)** |
| PK/FK | ProgramID | Unique ID identifying programs | Alphanumeric | 10 |
|  | Date | current date | Date | 15 |
| PK | VolunteerID | Unique ID identifying volunteers | Alphanumeric | 10 |
|  | StartTime | time volunteer started | DateTime | 15 |
|  | EndTime | time volunteer ended | DateTime | 15 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Program** | | | | |
| **PK/FK** | **Attribute** | **Definition** | **Type** | **Size (in bytes)** |
| PK | ProgramID | ID to identify program | Alphanumeric | 10 |
|  | ProgramName | Name of program | Alphanumeric | 15 |
|  | ProgramDescription | Description of the program | Alphanumeric | 50 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Log in** | | | | |
| **PK/FK** | **Attribute** | **Definition** | **Type** | **Size (in bytes)** |
| PK | Username | Username for users | Alphanumeric | 50 |
|  | Passwords | Password for user account | Alphanumeric | 20 |
|  | User\_Type | Defines access level | Alphanumeric | 20 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Events** | | | | |
| **PK/FK** | **Attribute** | **Definition** | **Type** | **Size (in bytes)** |
|  | EventID | ID to identify the event | Alphanumeric | 10 |
|  | EventName | Event Name | Alphanumeric | 15 |
|  | Date | Date and time of event | DateTime | 15 |
|  | Revenue | Money made from fundraiser | decimal | 10 |
|  | Expenses | Money spent on fundraiser | decimal | 10 |

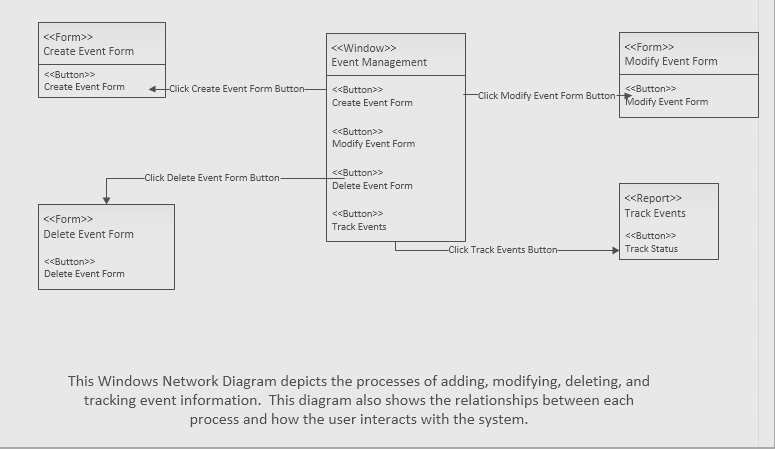
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Position** | | | | |
| **PK/FK** | **Attribute** | **Definition** | **Type** | **Size (in bytes)** |
| PK | Position\_ID | Unique position ID | Alphanumeric | 10 |
|  | Description | Position description | Alphanumeric | 50 |

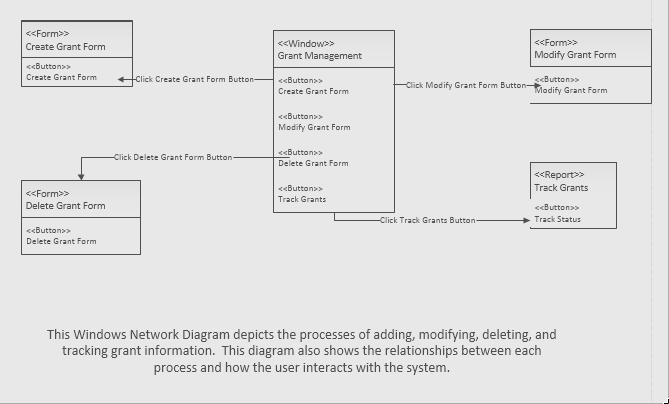
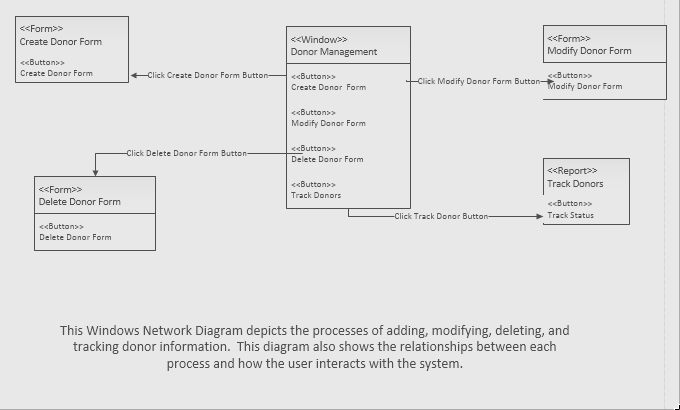
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Statistical Analysis** | | | | |
| **PK/FK** | **Attribute** | **Definition** | **Type** | **Size (in bytes)** |
|  | Report\_ID | Unique report ID number | Alphanumeric | 10 |
|  | Report\_Type | Report on students, grants, parents, etc | Alphanumeric | 10 |
|  | Assessment\_History | A history of student assessments | Alphanumeric | 50 |
|  | Info | Info about the analysis ran | Alphanumeric | 25 |
|  | StudentInfo | Info about each student | Alphanumeric | 15 |
|  | StudentID | unique student ID number | Alphanumeric | 10 |
|  | Program | Programs offered by Adelante | Alphanumeric | 50 |
|  | Name | name of analysis ran | Alphanumeric | 25 |

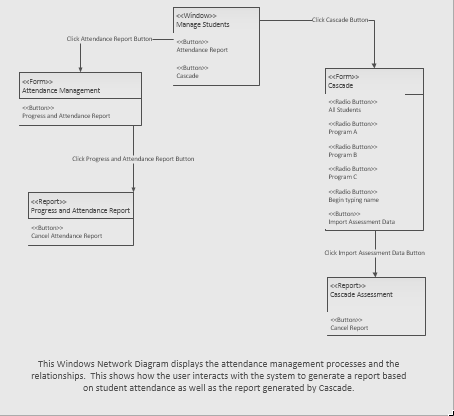
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Grant History** | | | | |
| **PK/FK** | **Attribute** | **Definition** | **Type** | **Size (in bytes)** |
| PK | GrantID | Unique number to identify each grant | Alphanumeric | 10 |
|  | Description | information about the grant | Alphanumeric | 20 |
|  | Name | Name attributed to the grant | Alphanumeric | 10 |
|  | DateApplied | Date grant was awarded | Date | 10 |
|  | Status | Status of grant (Accepted, rejected, in progress) | Alphanumeric | 10 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Grant Funds** | | | | |
| **PK/FK** | **Attribute** | **Definition** | **Type** | **Size (in bytes)** |
| PK | GrantID | Unique number to identify each grant | Alphanumeric | 10 |
|  | Description | information about the grant | Alphanumeric | 20 |
|  | Name | Name attributed to the grant | Alphanumeric | 10 |
|  | FundsRecieved | Amount of grant received | Decimal | 10 |
|  | FundsUsed | Amount of funds used | Decimal | 10 |
|  | Deadline | Deadline for funds to be used | Date | 10 |

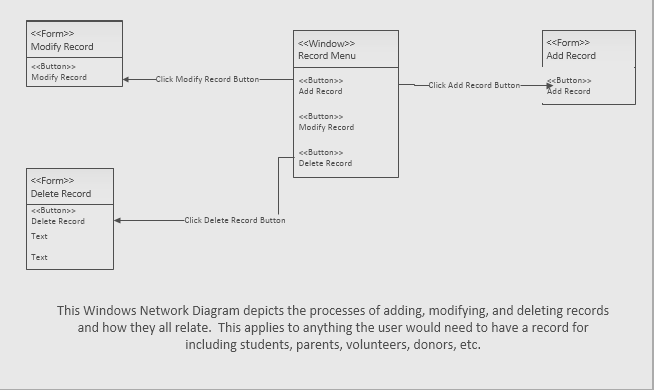
# User Interface Navigation Diagram and Screen Layouts

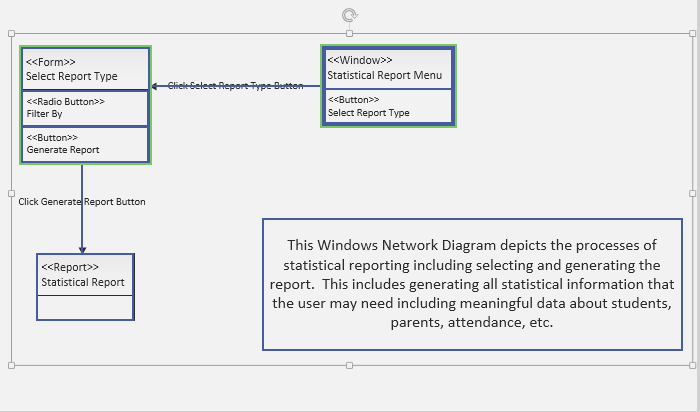
A navigation diagram is used to show how all of the forms and reports used in the system are related and how the user will move from one to the other.





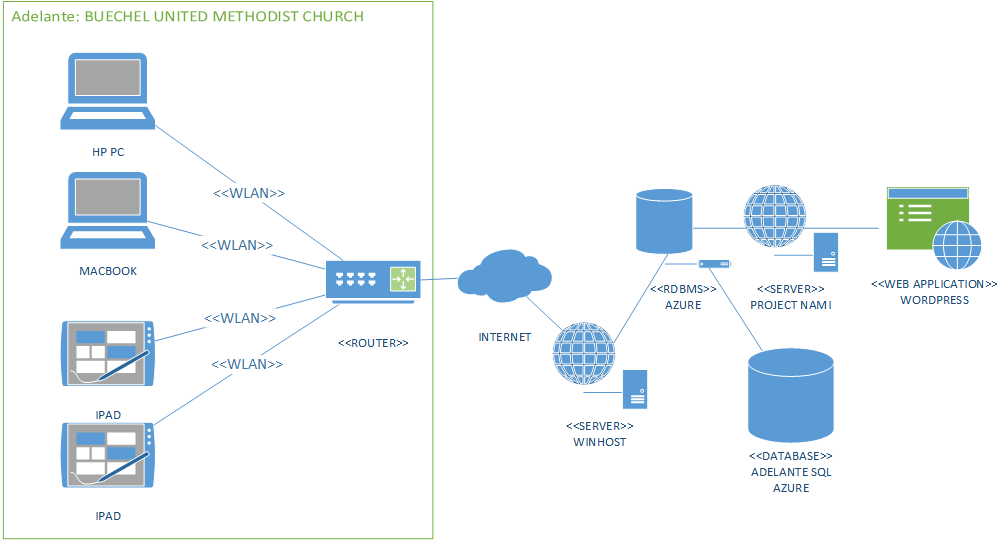
# 





# Physical Architecture Design

* **Deployment diagram**: depicts physical architecture and their software dependencies in the form of **nodes** (hardware compo or computing resources) and **artifacts** (specific software deployed on the node). Additionally, shows relationships between different nodes.



# Design Procedures for Security and Non-functional Requirements

These procedures will outline how to secure the system and how to handle security issues as they occur.

**Maintain security**

These measures will ensure that the system keeps an accurate backup of the system.

* Backup system every two weeks

**Prevent Security Issues**

These measures are to help prevent security issues from occurring.

* Separation of duties, no one individual should have access and control over a functional area of the system.
* Authentication and authorization procedures to verify and allow access on a user level
* Training of all staff on secure procedures including protection of physical devices in addition to creating secure credentials
* System will be protected by a firewall and updated monthly with a patch to the system
* Accept updates from WordPress

**Detecting Security Issues**

These measures will help to detect any security issues that may occur

* System will keep logs of user access including time and date of access

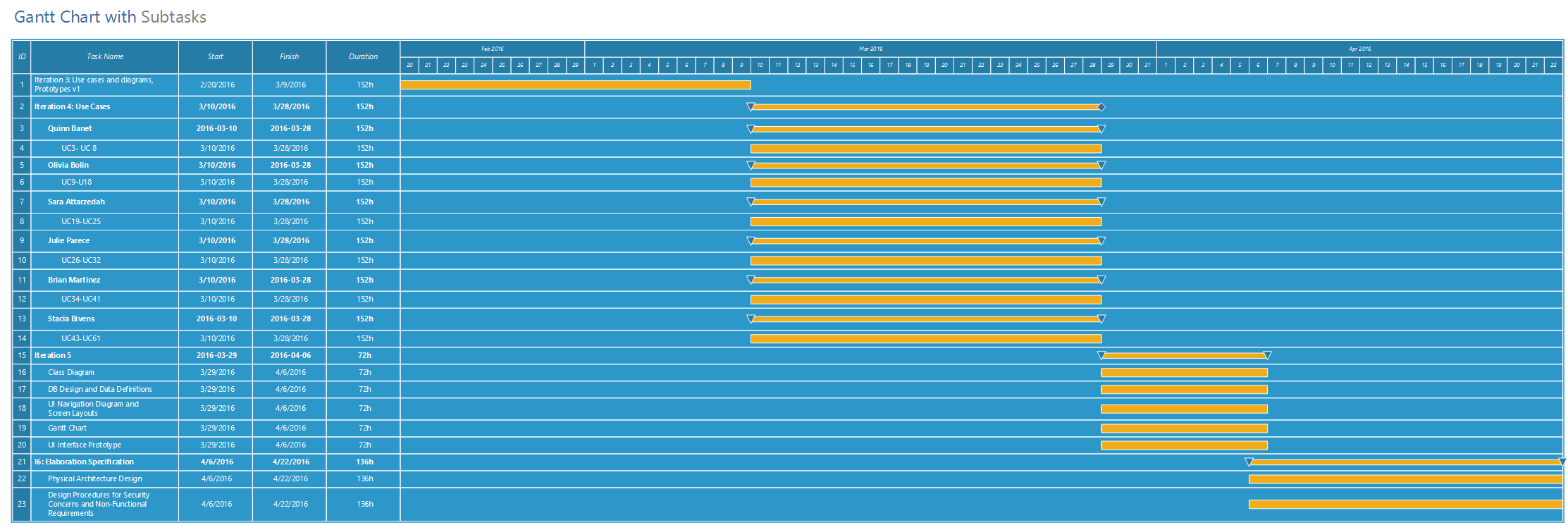
**Correcting Security Issues**:

In the event that a security issue occurs these measures will ensure that the system is corrected to its original state.

* System Restore backup
* Anti-virus software
* OS Upgrades

# Gantt Chart

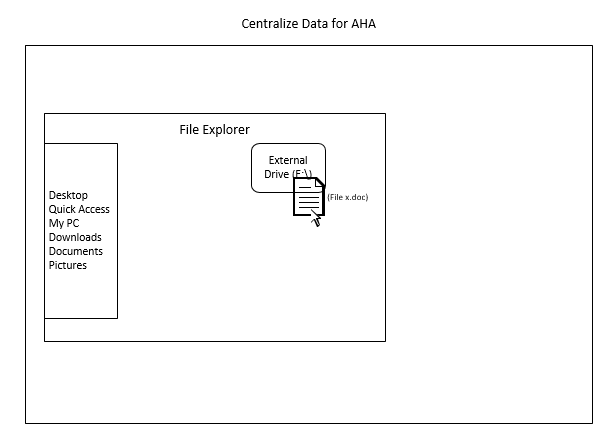
The Gantt Chart illustrates start and finish dates of system deliverables along with the elements that pertain to the project. This chart is specific to iteration four and five of our project deliverables.



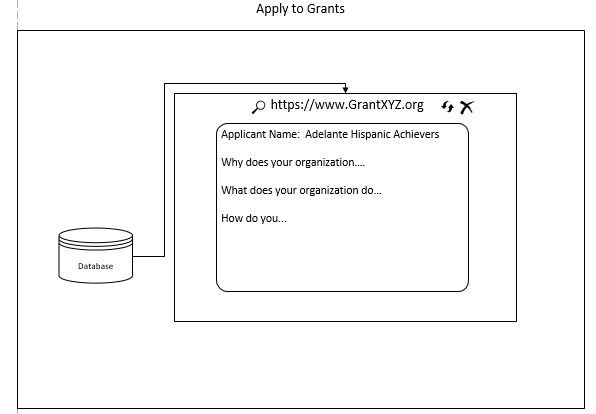
# Elaboration Phase Prototypes

These prototypes represent the data needs and process flows of all use cases. They show the physical screen layout of what the user would see when interacting with the system. These are all high-level views to display a broad overview of the requirements for each use case.

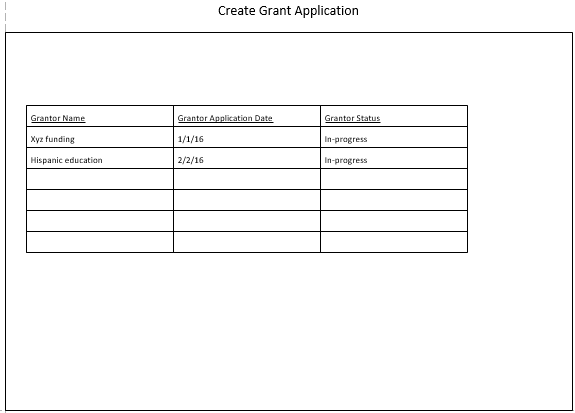
UC-1



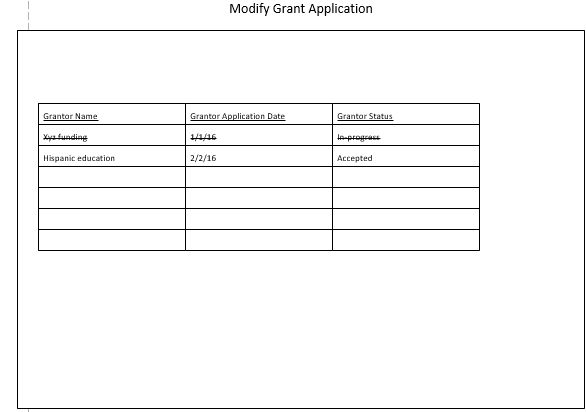
UC2

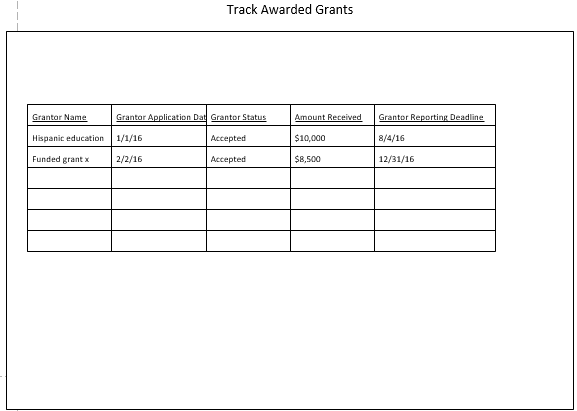


UC3

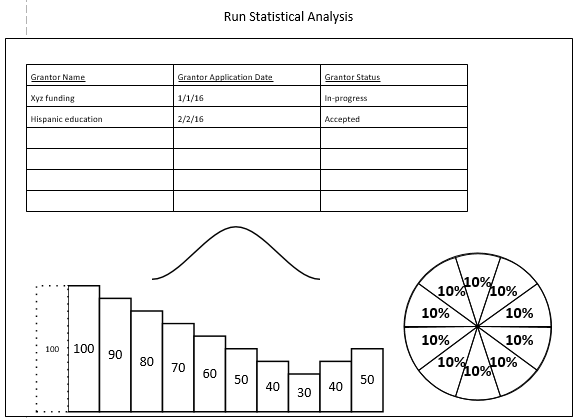


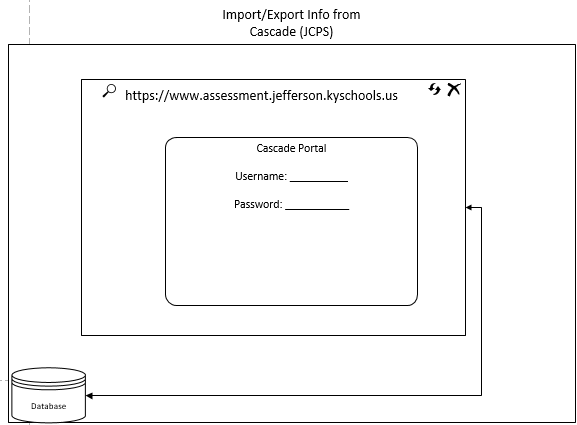
UC4

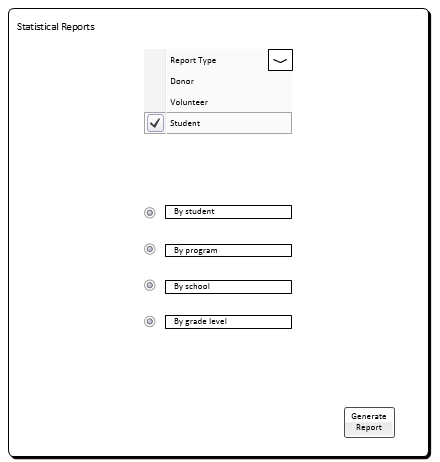
UC5

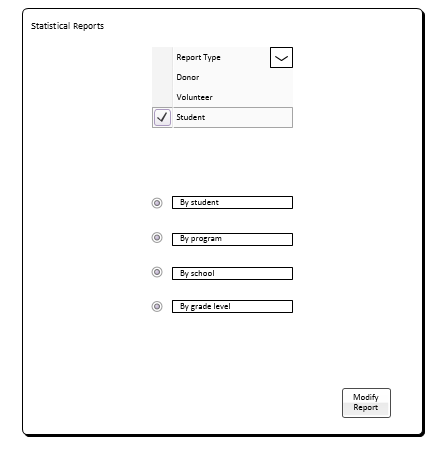


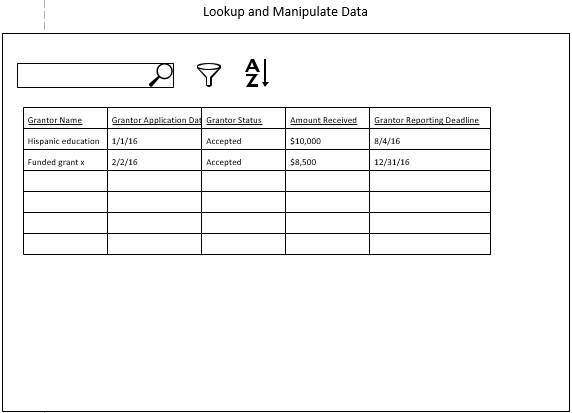
UC6

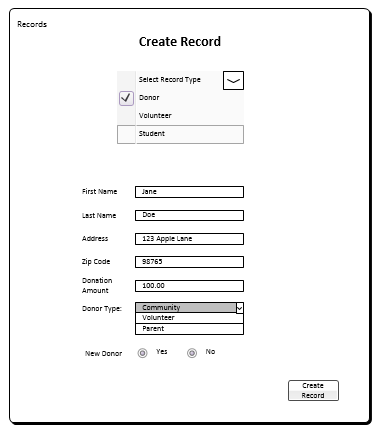
UC7

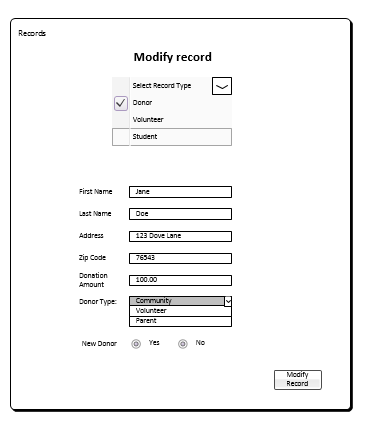
UC8

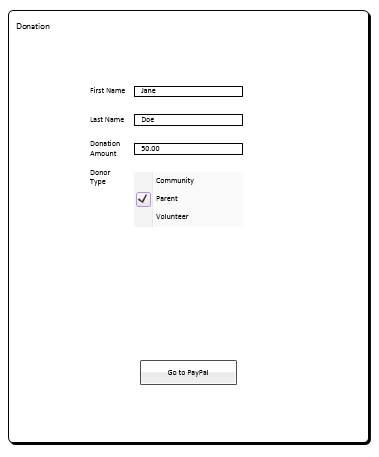
UC-9

UC-10

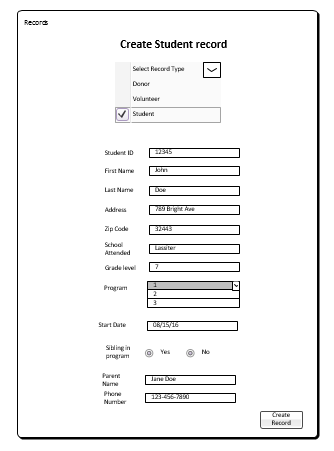
UC- 11

UC-12

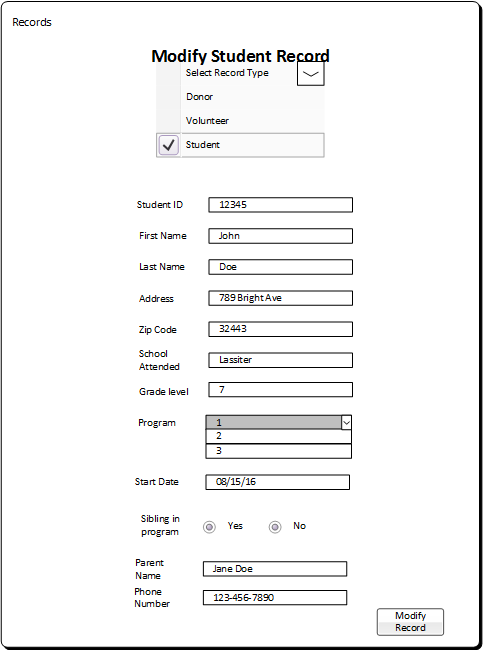
UC-13

UC-14

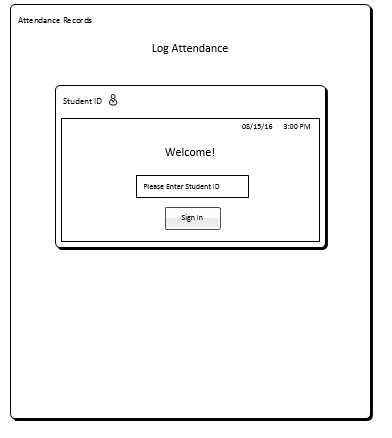
UC-15



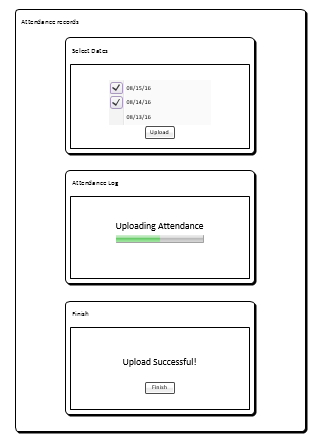
UC-16

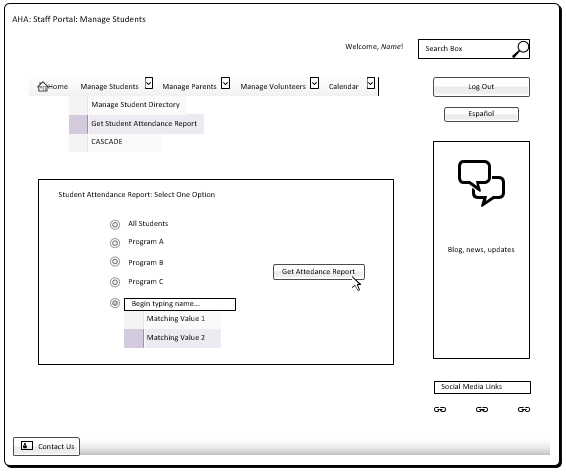
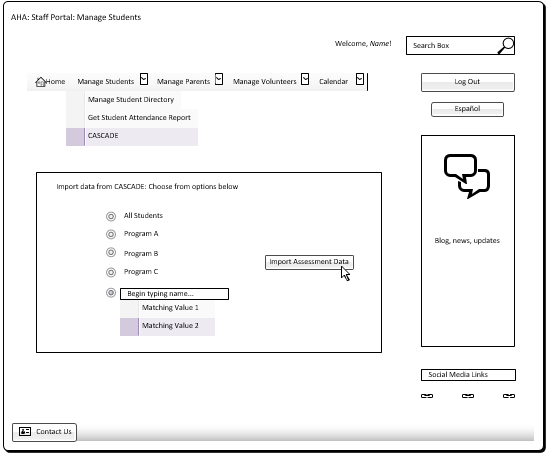


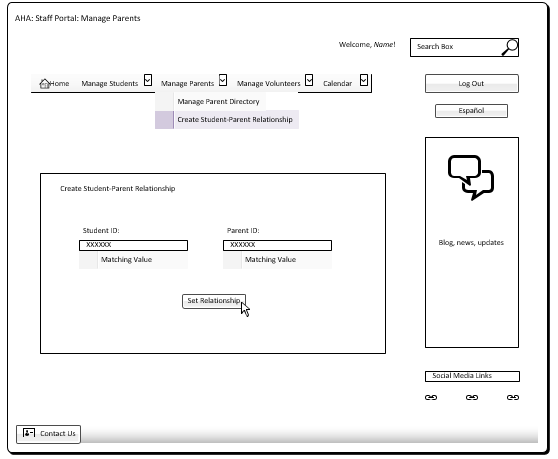
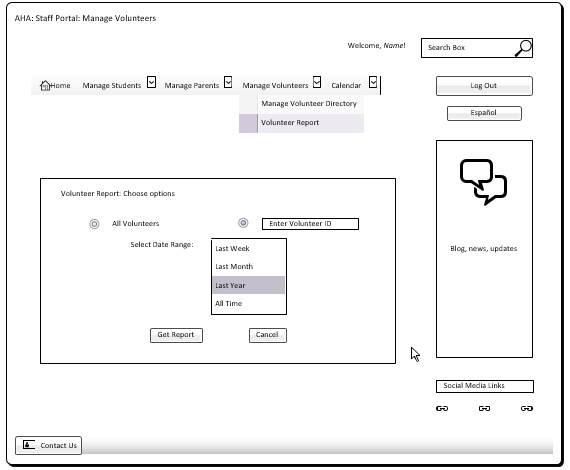
UC-17

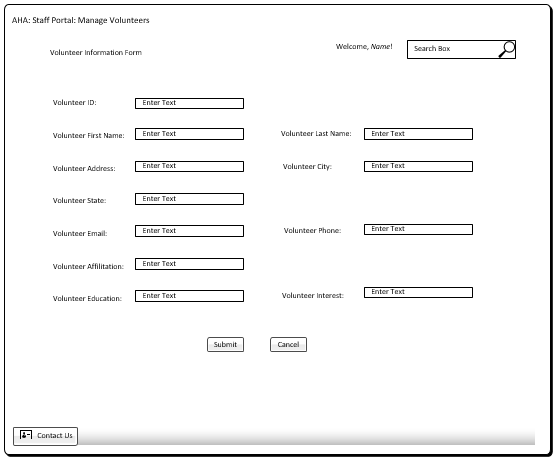


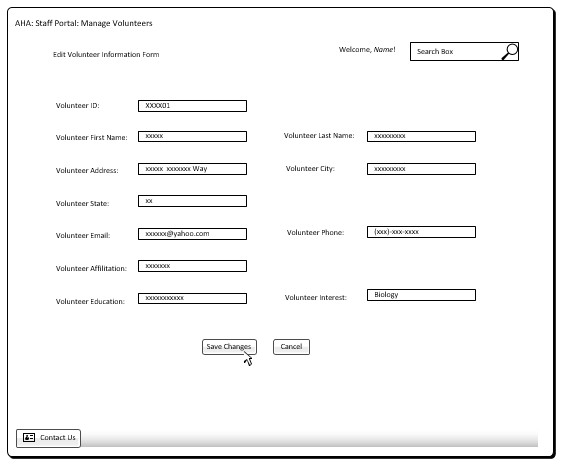
**UC-18**

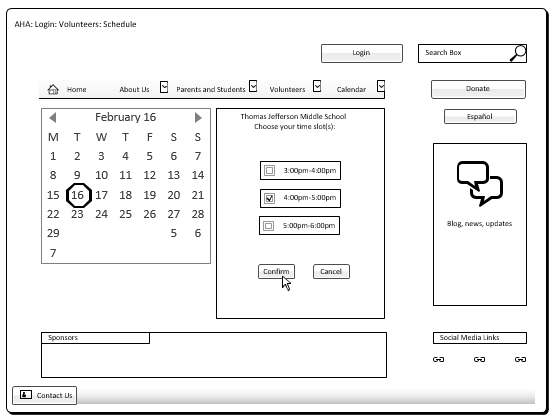


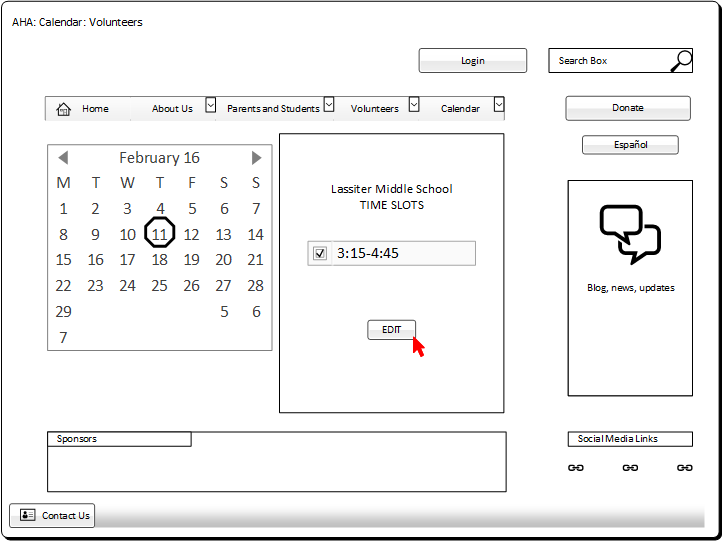
**UC19****UC20**

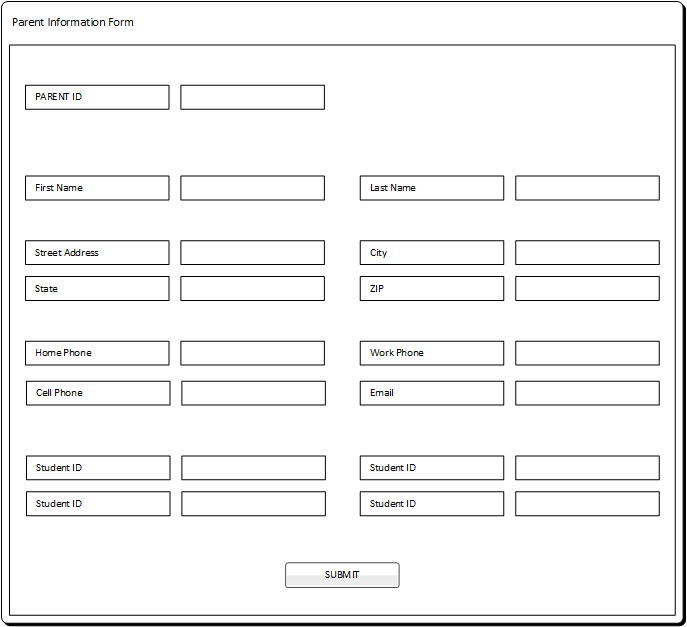
**UC21****UC22**

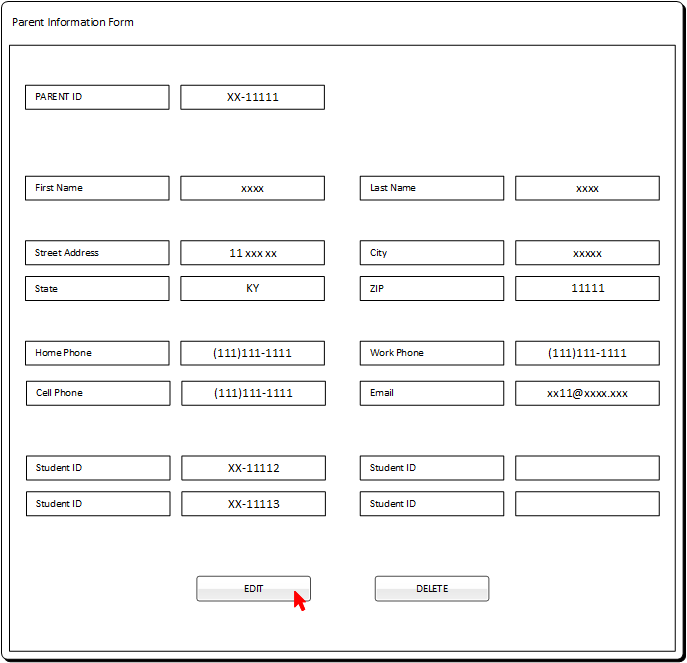
**UC23**

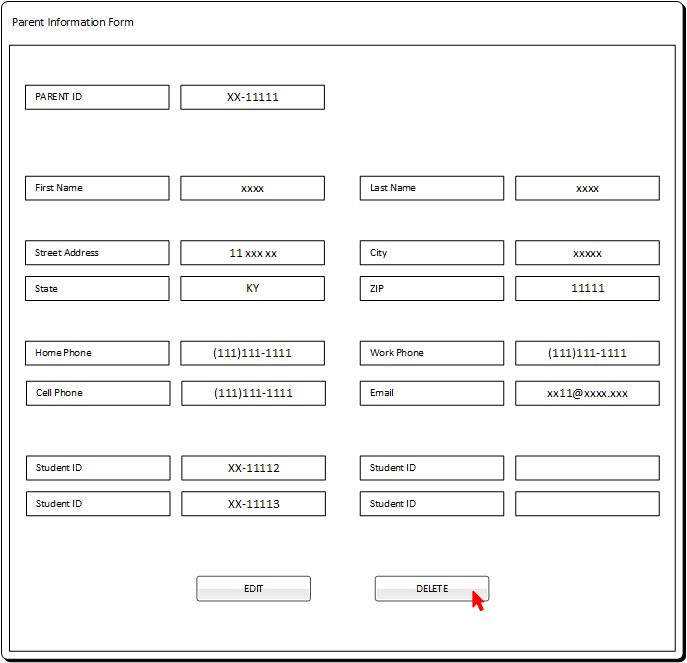
**UC24**

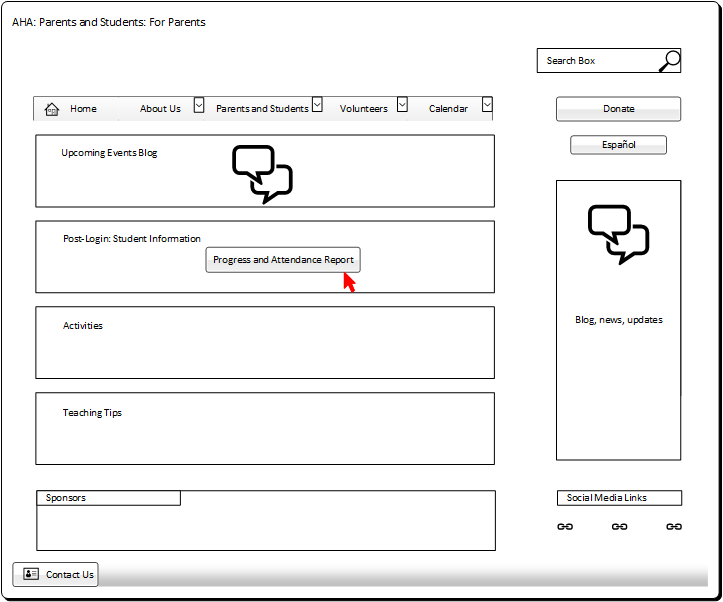
**UC25**

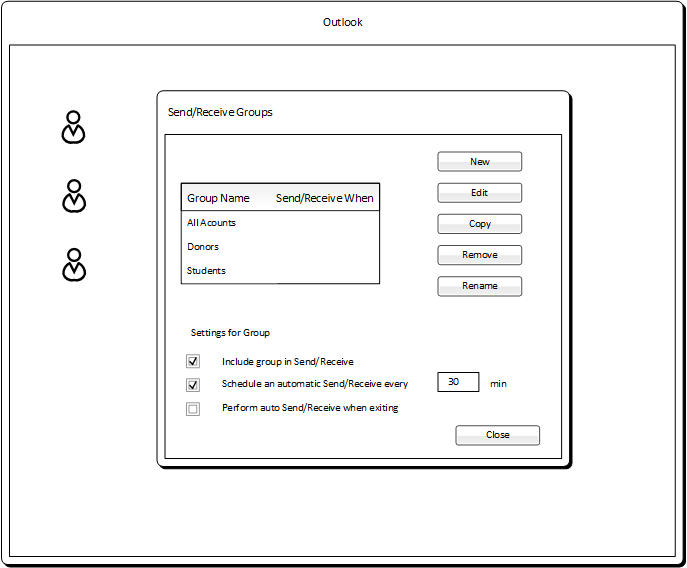
**UC26**

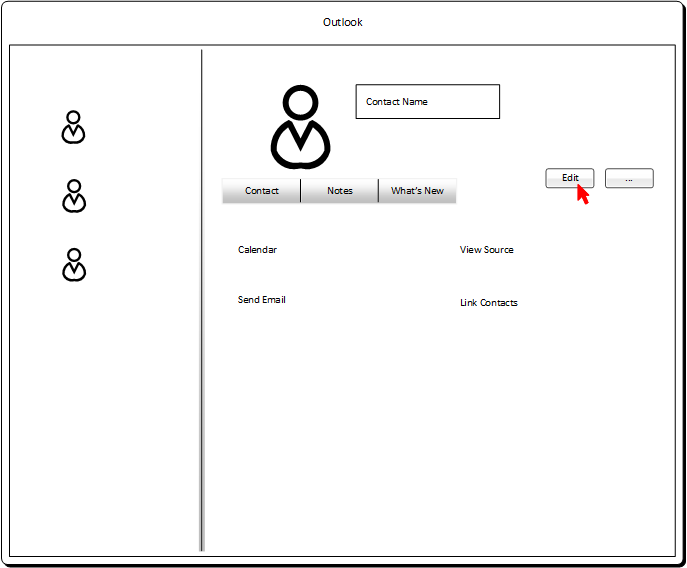
**UC27**

**UC28**

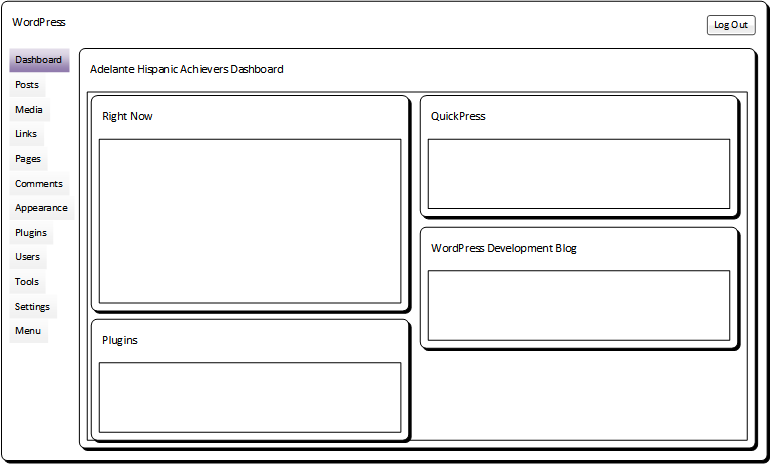
**UC29**

**UC30**

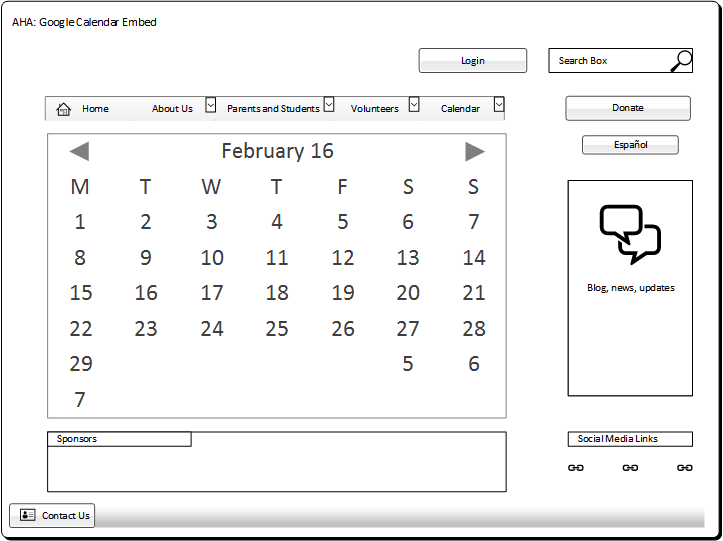
**UC31**

**UC32**

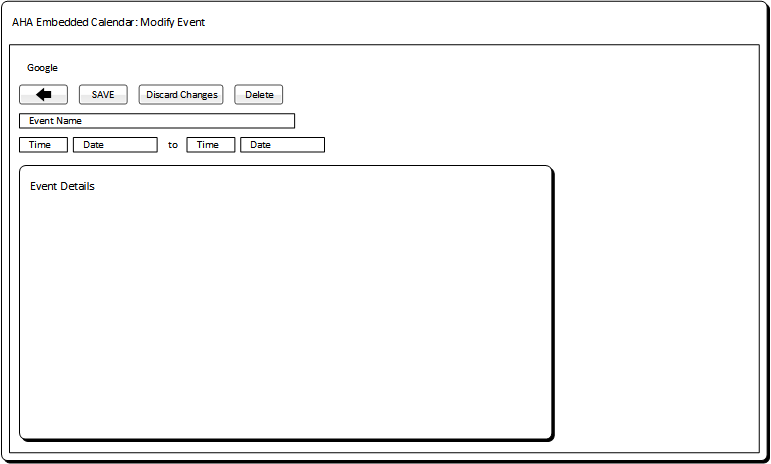
**UC33**



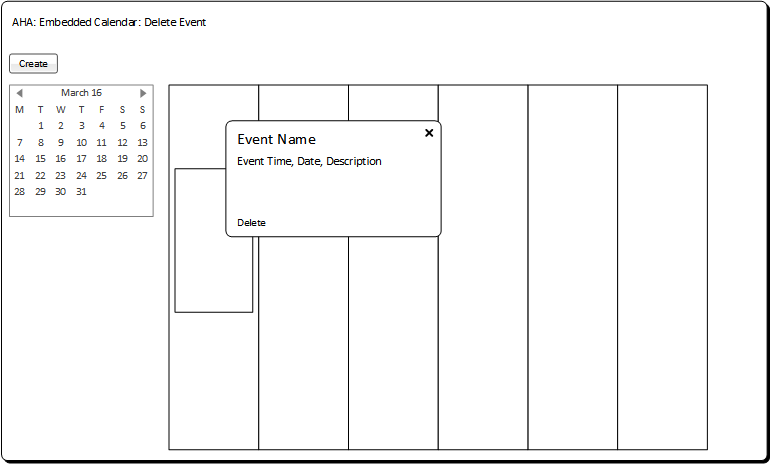
**UC34**



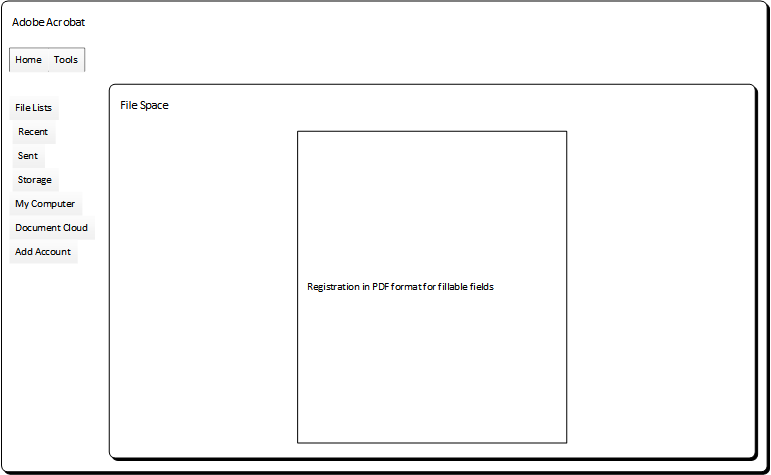
**UC35**



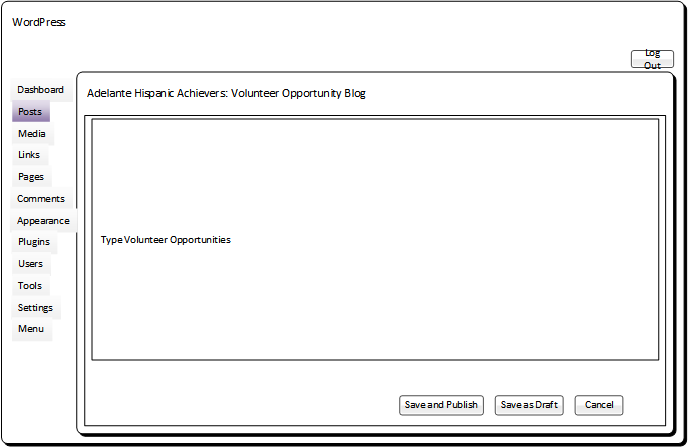
**UC36**



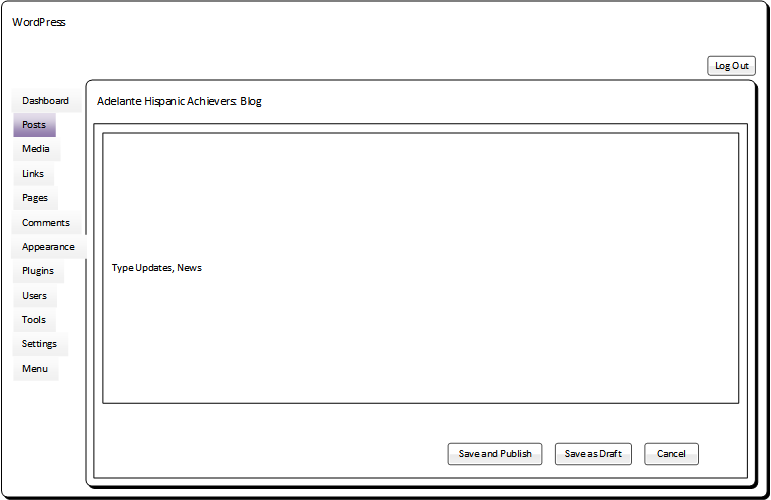
**UC37**



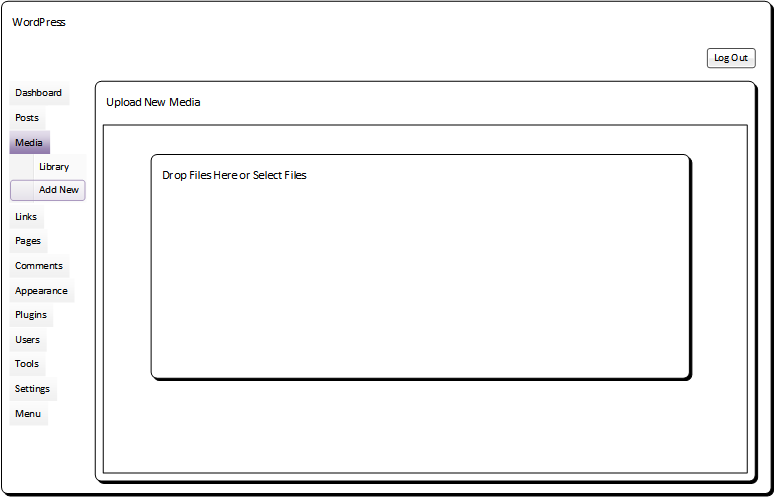
**UC38**



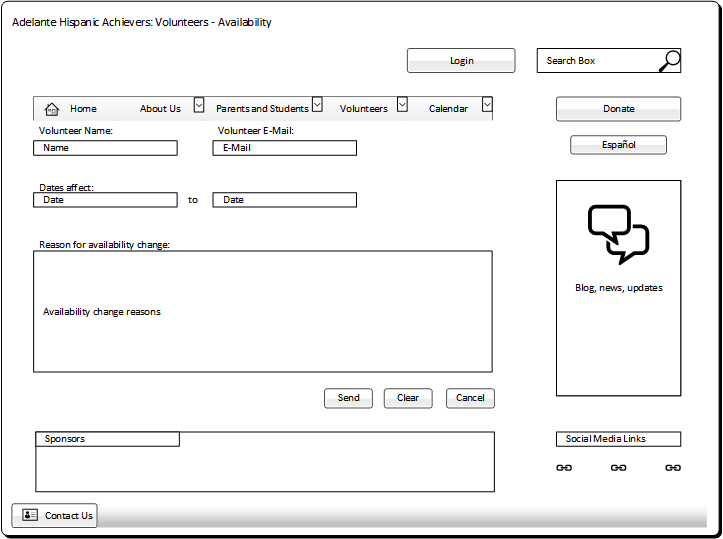
**UC39**



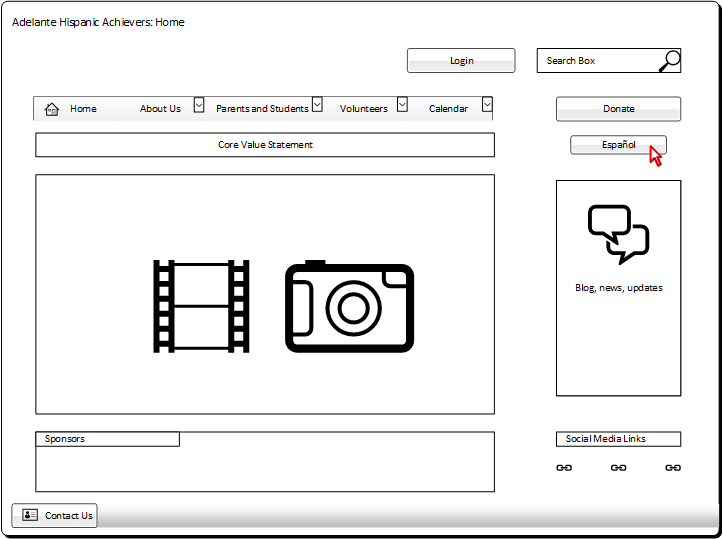
**UC40**

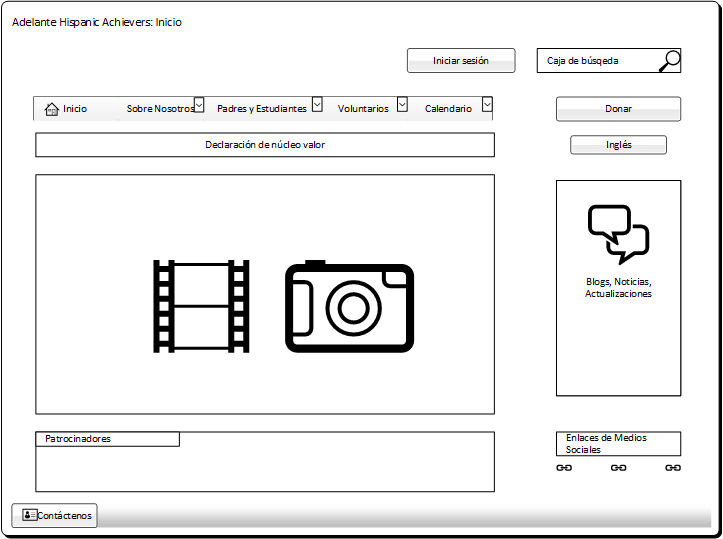


**UC41**

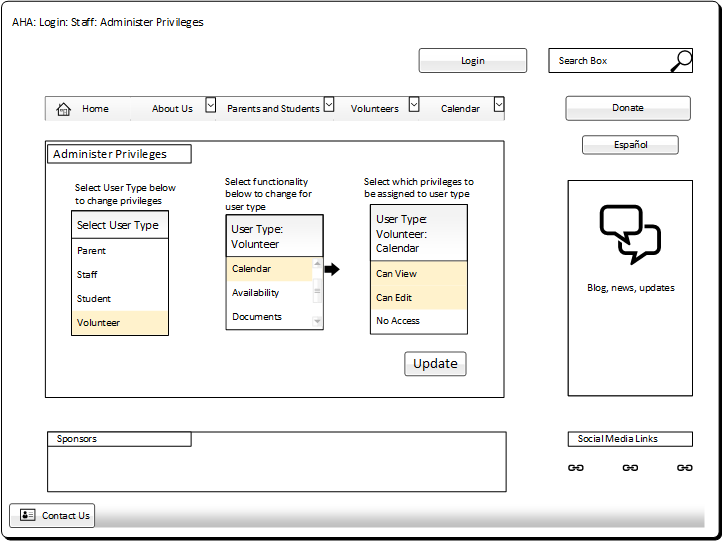


**UC 42**

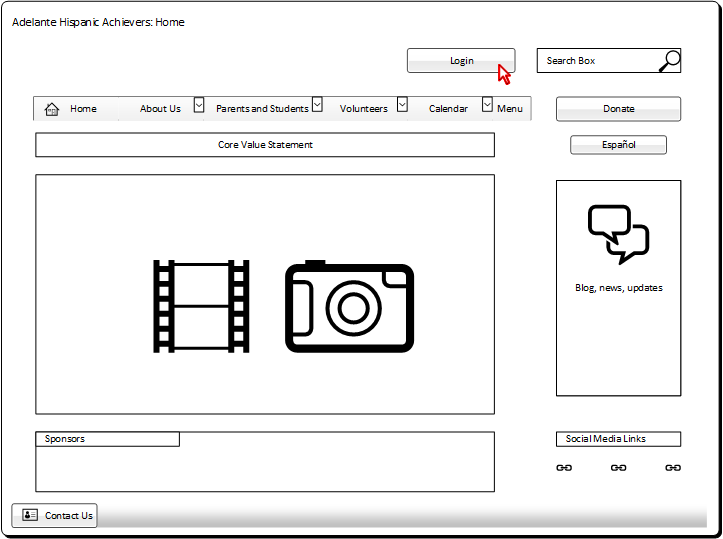




**UC 43**

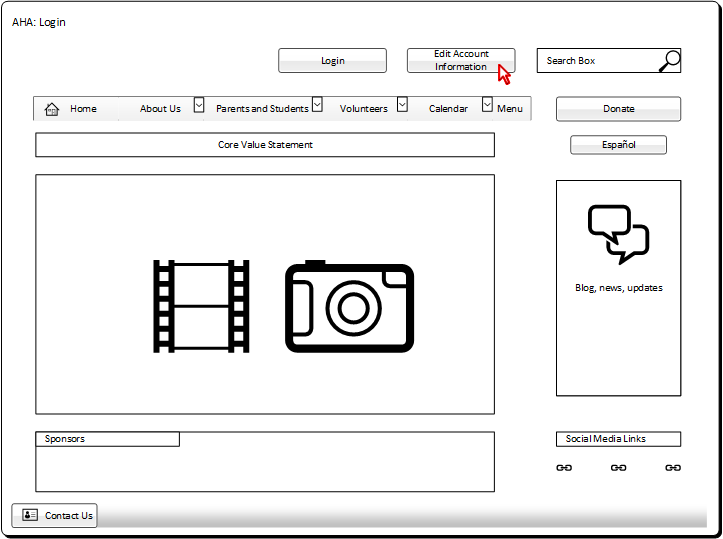


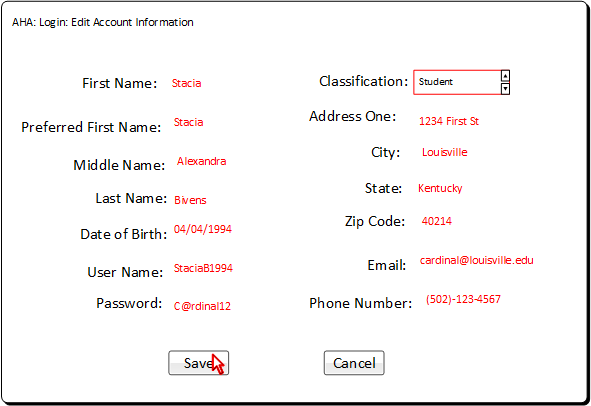
**UC 44**



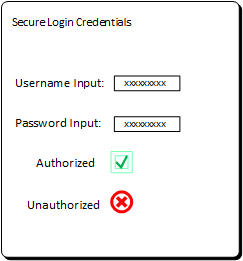


**UC 45**

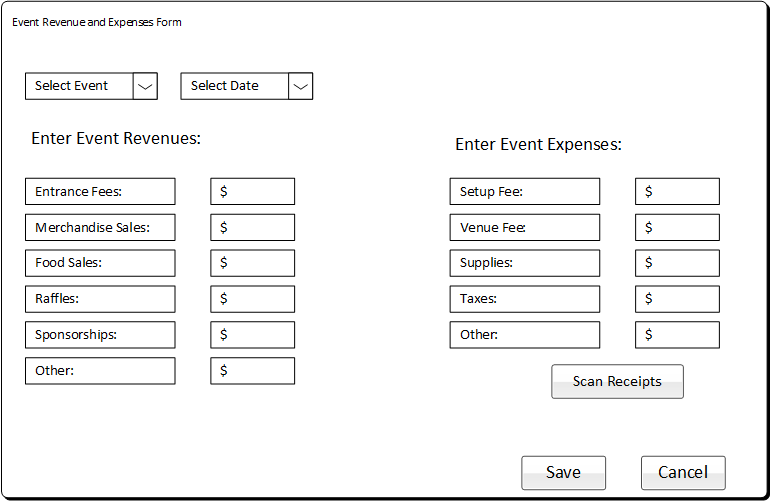


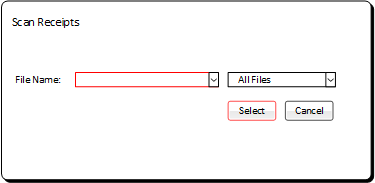


**UC 46**

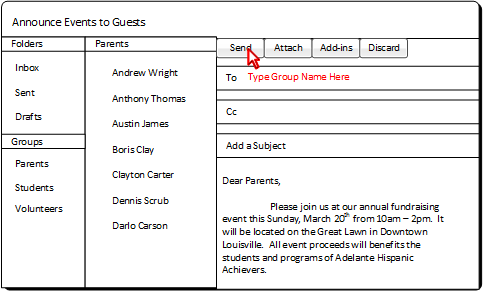


**UC 47**

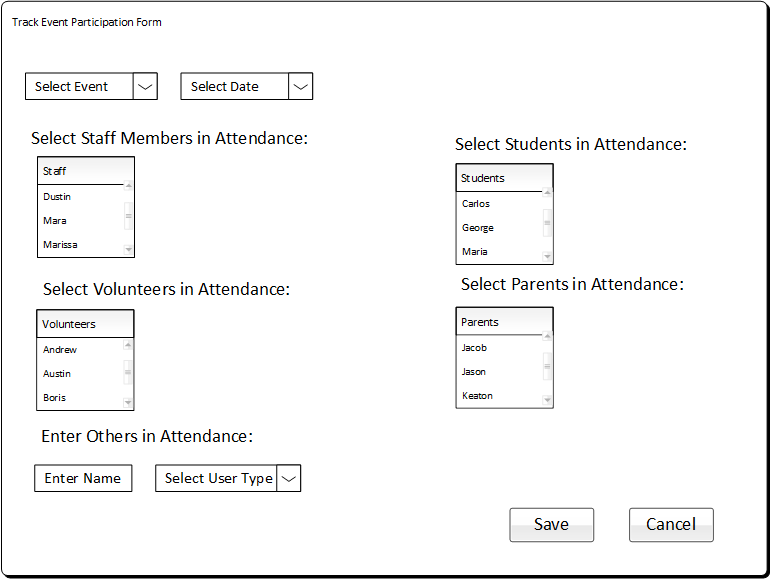




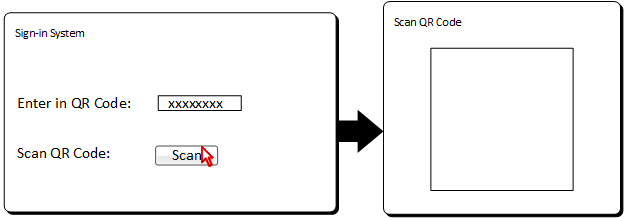
**UC 48**

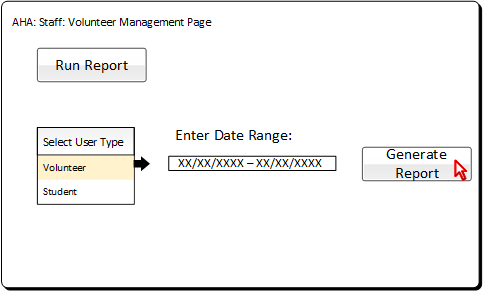


**UC 49**

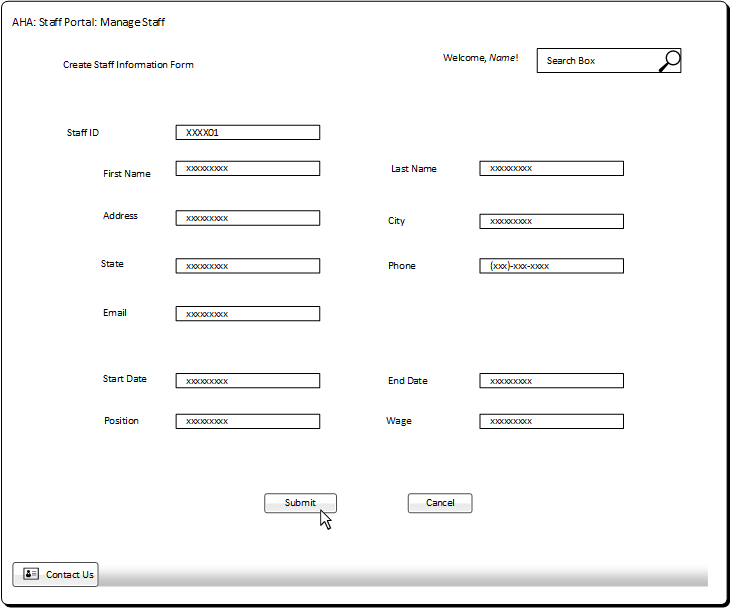


**UC 50**

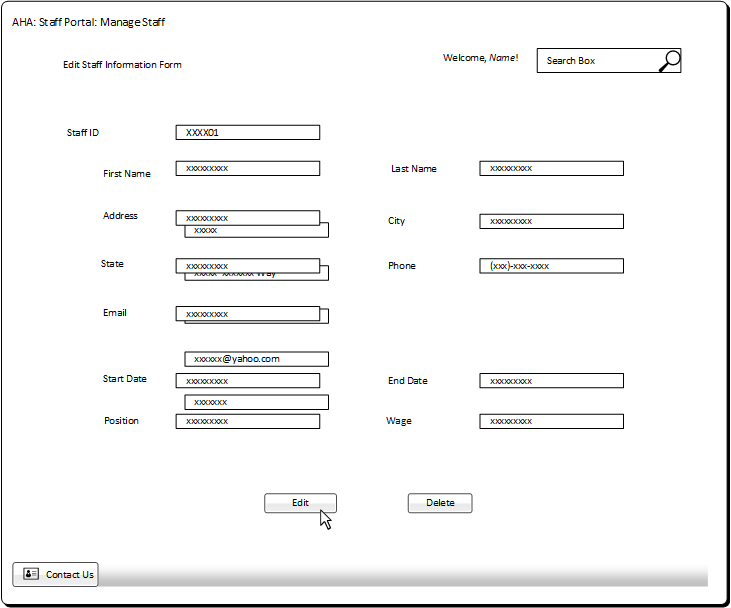




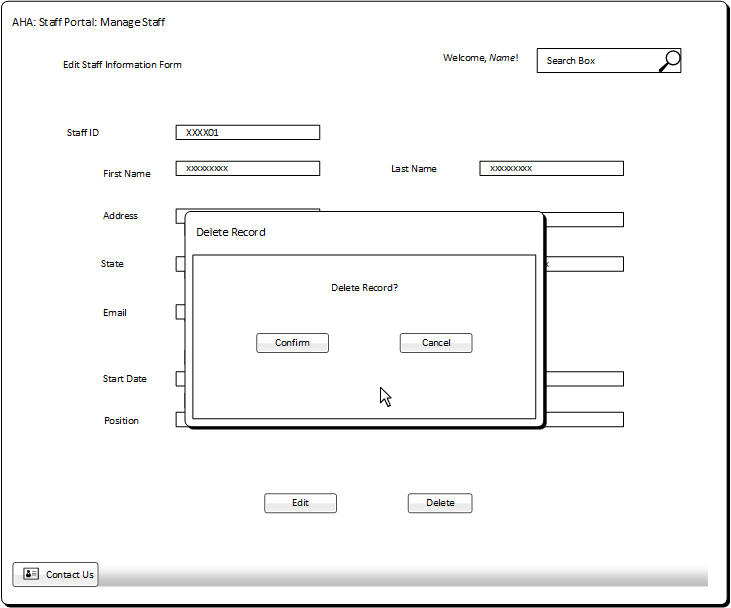
UC-54 Create Staff Information Record



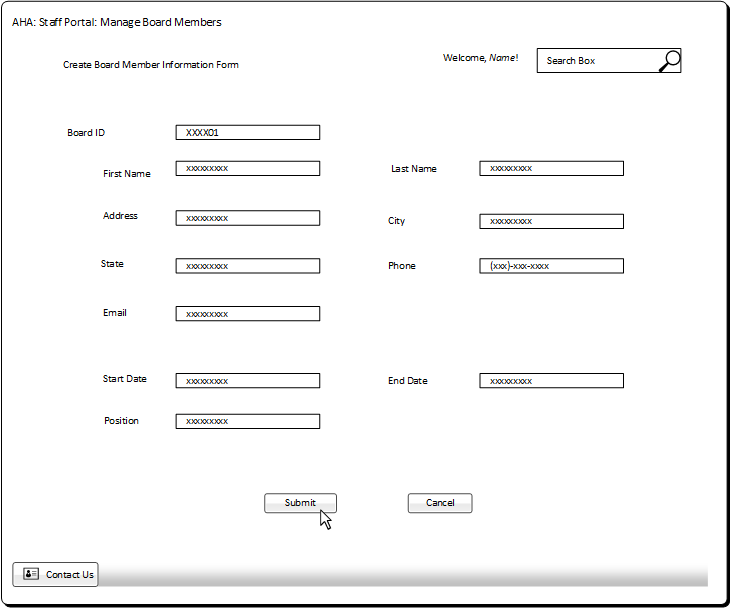
UC-55 Modify Staff Record



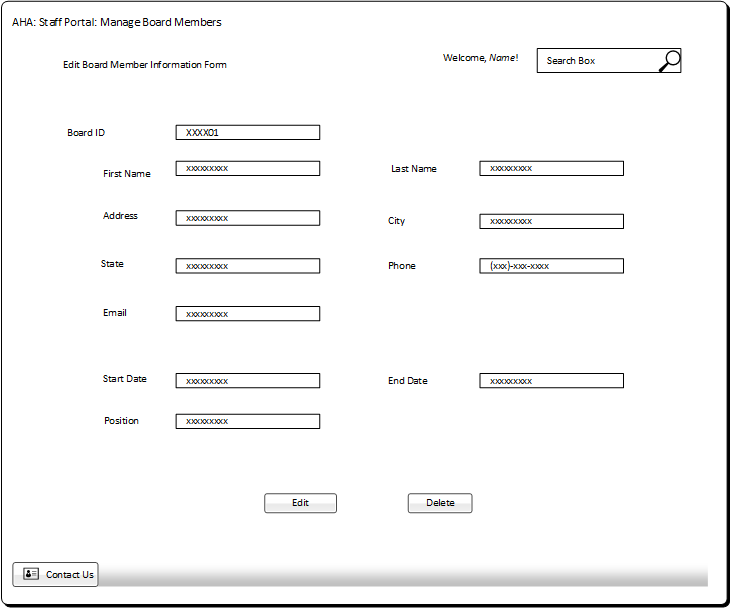
UC-56 Delete Staff Record



UC-57 Create Board Member Record



UC-58 Modify Board Member Record



UC-59 Delete Board Member Record

