t

Iteration 3

Swiftriver

CIS 320-01

Due: 4/2/18

[**Gantt chart for I3**](#_f4duvhx98iea) **3**

[**Trace Matrix**](#_61dmbb2ivh21) **4**

[**Use Case Diagram**](#_pa42liivs6za) **4**

[**High Risk Use Cases:**](#_qiu2bcsy53xj) **5**

[**Use Case: Add Silent Auction Sale**](#_777b1zy92c4o) **5**

[**Use Case: Modify Silent Auction Sale**](#_m3slgemszqt0) **5**

[**Use Case: Delete Silent Auction Sale**](#_vs818u1xpvq4) **6**

[**Use Case: Add Children**](#_6dda1ac48v9l) **7**

[**Use Case: Modify Children**](#_cgnmxvvi5j4v) **7**

[**Use Case: Delete Children**](#_dw7hb3a8pade) **8**

[**Use Case: Add Summer Camp Application**](#_jfu51svyymzu) **9**

[**Use Case: Modify Summer Camp Application**](#_d5ldqfnup4a1) **10**

[**Use Case: Delete Summer Camp Application**](#_sbe1drbyilgw) **11**

[**Use Case: Add One Time Donation**](#_fzbple8h4pxd) **11**

[**Prototypes**](#_4ngfb6d6j2hc) **16**

[**Add & Modify Summer Camp Application**](#_qvs5scqo878d) **16**

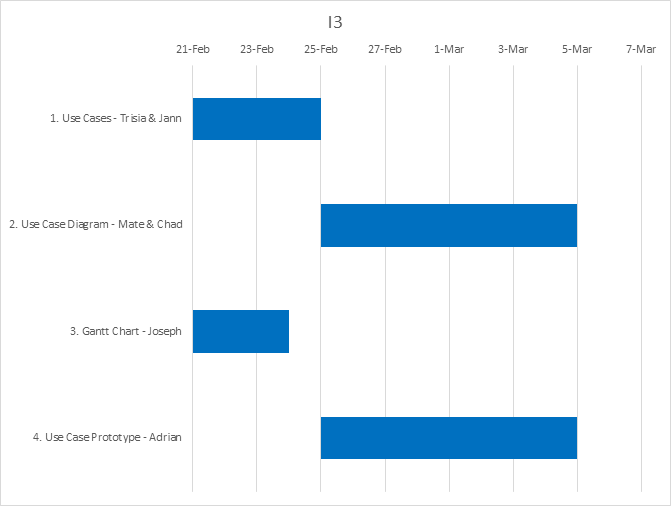
[**Delete Summer Camp Application**](#_1lcv42ug60is) **17**

[**Donation Form**](#_jjfnbg94n7nt) **18**

# Gantt chart for I3

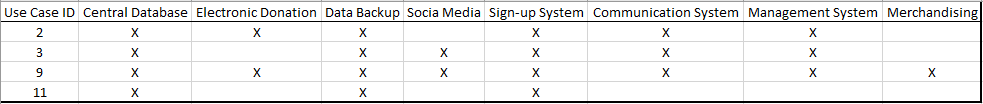
This chart has the name of the task and the name of the group member performing the task to the left of the chart. Each task has a bar that indicated the length of the task, measured in days. Some tasks are independent and can be started whenever while others are dependent and require certain tasks be completed before they can be started. A dependent task is indicated by its starting point being at the end of another task instead of being on the first day like the other tasks. Where the bar starts indicates which task it depends on.

The I3 Gantt chart shows that the dependent tasks are Use Case Diagram and the Use Case Prototype.

**

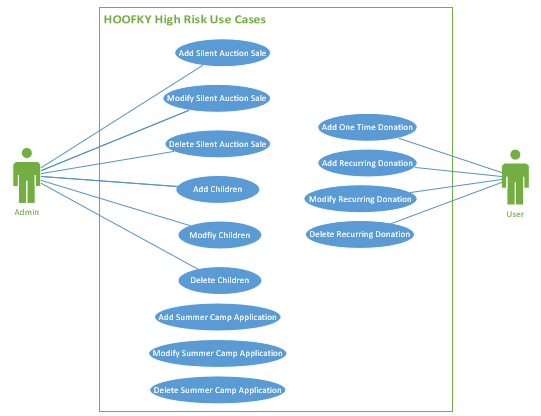
# Trace Matrix

A trace matrix is used to associate use cases with system requirements. Use cases are interactions between the actors and a business process. We require a system that will support these business process.



# Use Case Diagram

The actor is depicted as the stick figures. An actor is a person that benefits and associates with a use case. The use cases are in the blue ovals. A use case is a sequence of actions. The actor and the use case are connected by an association line.



# High Risk Use Cases:

# Use Case: Add Silent Auction Sale

|  |  |
| --- | --- |
| ID: | 9.A |
| Use Case Name:: | Add Silent Auction Sale |
| Primary Actor(s): | Admin |
| Risk Level: | High |
| Description: | This use case describes how an admin can add a silent auction profit if item is sold. |
| Preconditions: | Admin is logged into hoofky.org |
| Postconditions: | Admin adds a silent auction sale. |
| Flow of Events: | * Admin clicks on “Manage Sales”. * Admin clicks on “Add/Edit Silent Auction Sale”. * Admin enters item name. * Admin enters date of sale. * Admin enters price. * Admin enters buyer’s name. * Admin clicks submit. * Admin logs off hoofky.org. * Use case ends. |

# Use Case: Modify Silent Auction Sale

|  |  |
| --- | --- |
| ID: | 9.B |
| Use Case Name:: | Modify Silent Auction Sale |
| Primary Actor(s): | Admin |
| Risk Level: | High |
| Description: | This use case describes how an admin can modify a silent auction profit if item is sold. |
| Preconditions: | Admin is logged into hoofky.org. |
| Postconditions: | Admin modifies silent auction sale. |
| Flow of Events: | * Admin clicks on “Manage Sales”. * Admin clicks on “Add/Edit Silent Auction Sale”. * Admin enters item name. * Admin enters date of sale. * Admin enters price. * Admin enters buyer’s name. * Admin clicks submit. * Admin logs off hoofky.org. * Use case ends. |

# Use Case: Delete Silent Auction Sale

|  |  |
| --- | --- |
| ID: | 9.C |
| Use Case Name:: | Delete Silent Auction Sale |
| Primary Actor(s): | Admin |
| Risk Level: | High |
| Description: | This use case describes how an admin can delete a silent auction profit if item is sold. |
| Preconditions: | Admin is logged into hoofky.org |
| Postconditions: | Admin deletes silent auction sale. |
| Flow of Events: | * Admin clicks on “Manage Sales”. * Admin clicks on “Add/Edit Silent Auction Sale”. * Admin enters item name. * Admin enters date of sale. * Admin enters price. * Admin enters buyer’s name. * Admin checks “Check here to delete this item” * Admin clicks submit. * Admin logs off hoofky.org. * Use case ends. |

# Use Case: Add Children

|  |  |
| --- | --- |
| ID: | 3.A |
| Use Case Name:: | Add Children |
| Primary Actor(s): | Admins |
| Risk Level: | High |
| Description: | This use case describes how an Admin can add children information |
| Precondition: | Admin logs into WordPress |
| Postcondition: | Child information is added |
| Main Flow: | * Admin clicks on “My Site” to open hoofky.org editor. * Admin clicks on “Customize.” * Admin clicks “Campers” button. * Admin clicks on blue edit icon. * Admin clicks “Add Camper.” * Admin adds child information. * Admin clicks “Submit” * Child is added to HOOF database * Admin logs out of WordPress * Use Case ends |

# Use Case: Modify Children

|  |  |
| --- | --- |
| ID: | 3.B |
| Use Case Name:: | Modify Children |
| Primary Actor(s): | Parents/Guardians, Admin |
| Risk Level: | High |
| Description: | This use case describes how Admin can modify children information. |
| Precondition: | Admin logs into WordPress |
| Postcondition: | Child information is modified/updated |
| Main Flow: | * Admin clicks on “My Site” to open hoofky.org editor. * Admin clicks on “Customize.” * Admin clicks “Campers” button. * Admin clicks on blue edit icon. * Admin clicks “Edit Camper.” * Admin modifies/updates child information. * Admin clicks “Submit” * Child is modified to HOOF database * Admin logs out of WordPress * Use Case ends |

# Use Case: Delete Children

|  |  |
| --- | --- |
| ID: | 3.C |
| Use Case Name:: | Add |
| Primary Actor(s): | Admin |
| Risk Level: | High |
| Description: | This use case describes how an Admin can delete children information. |
| Precondition: | Admin logs into WordPress |
| Postcondition: | Child information is deleted from HOOF database |
| Main Flow: | * Admin clicks on “My Site” to open hoofky.org editor. * Admin clicks on “Customize.” * Admin clicks “Campers” button. * Admin clicks on blue edit icon. * Admin clicks “Delete Camper.” * Admin selects child to delete. * Admin deletes child. * Admin clicks “Submit” * Child is deleted from HOOF database * Admin logs out of WordPress * Use Case ends |

# 

# [Use Case: Add Summer Camp Application](https://docs.google.com/document/d/1xTxvNjxT6elOpWh-LB03uLaggccFy_ORGtZGyxBFSkk/edit#heading=h.excrkxs2kaq5)

|  |  |
| --- | --- |
| ID: | 11.A |
| Use Case Name: | Add Summer Camp Application |
| Primary Actor(s): | Admin |
| Risk Level: | High |
| Description: | This use case describes how an admin can add a summer camp application form. |
| Precondition: | Admin is logged into WordPress and is in site editor. |
| Postcondition: | Summer camp application is available. |
| Flow of Events: | * Admin creates Child’s first name textbox * Admin creates Child’s middle initial textbox * Admin creates Child’s last name textbox * Admin creates Child’s age textbox * Admin creates Parent’s first name textbox * Admin creates Parent’s last name textbox * Admin creates Parent’s middle initial textbox * Admin creates email textbox * Admin creates primary phone number textbox * Admin creates secondary phone number textbox * Admin creates emergency contact number textbox * Admin creates address line 1 textbox * Admin creates address line 2 textbox * Admin creates city textbox * Admin creates state textbox * Admin creates zip textbox * Admin creates school name textbox * Admin saves changes * Admin requests submission to publish the application form * Summer application form is submitted * New application form is added onto hoofky * Admin logs off WordPress * Admin closes WordPress * Use case ends |

# [Use Case: Modify Summer Camp Application](https://docs.google.com/document/d/1xTxvNjxT6elOpWh-LB03uLaggccFy_ORGtZGyxBFSkk/edit#heading=h.wz01cx6mshn6)

|  |  |
| --- | --- |
| ID: | 11.B |
| Use Case Name: | Modify Summer Camp Application |
| Primary Actor(s): | Admin |
| Risk Level: | High |
| Description: | This use case describes how an admin can modify a summer camp application form. |
| Precondition: | Admin is logged in WordPress and is in site editor. |
| Postcondition: | Summer camp application files are updated. |
| Flow of Events: | * Admin alters Child’s first name textbox * Admin alters Child’s middle initial textbox * Admin alters Child’s last name textbox * Admin alters Parent’s first name textbox * Admin alters Parent’s last name textbox * Admin alters Parent’s middle initial textbox * Admin alters email textbox * Admin alters primary phone number textbox * Admin alters secondary phone number textbox * Admin alters emergency contact number textbox * Admin alters address line 1 textbox * Admin alters address line 2 textbox * Admin alters city textbox * Admin alters state textbox * Admin alters zip textbox * Admin alters school name textbox * Admin saves changes * Admin requests submission to publish the application form * Summer application form is submitted * Modified application form is added onto hoofky * Admin logs off WordPress * Admin closes WordPress * Use case ends |

# 

# [Use Case: Delete Summer Camp Application](https://docs.google.com/document/d/1xTxvNjxT6elOpWh-LB03uLaggccFy_ORGtZGyxBFSkk/edit#heading=h.omku4vvgmpo5)

|  |  |
| --- | --- |
| ID: | 11.C |
| Use Case Name: | Delete Summer Camp Application |
| Primary Actor(s): | Admin |
| Risk Level: | High |
| Description: | This use case describes how an admin can delete a summer camp application. |
| Precondition: | Admin is logged in WordPress and is in site editor. |
| Postcondition: | Admin is logged in WordPress and is in site editor. |
| Flow of Events: | * Admin requests delete link * Application is deleted * Admin saves changes * Admin requests submission to Publish * Summer application form is deleted on hoofky * Admin logs off WordPress * Admin closes WordPress * Use case ends |

# Use Case: Add One Time Donation

|  |  |
| --- | --- |
| ID: | 2.A |
| Use Case Name: | Add One Time Donation |
| Primary Actor(s): | Donor |
| Risk Level: | High |
| Description: | This use case describes how donors enters a one-time donation amount. |
| Precondition: | User accesses the HoofKY website. |
| Postcondition: | User successfully makes their one-time donation. |
| Flow of Events: | 1.) User clicks “Donate Now” button  2.) User clicks “Log In With PayPal” button  3.) User enters PayPal username  4.) User enters PayPal password  5.) User clicks “Log In”  6.) User verifies one-time transaction  7.) User is redirected back to HOOF website  8.) Use case ends |
| Explanation: | This use case is classified as high-risk because large portions of HOOF’s operating budget come from user donations, and if they were unable to accept donations they would lose significant funding. |

Use Case: Add Recurring Donation

|  |  |
| --- | --- |
| ID: | 2.B |
| Use Case Name: | Add Recurring Donation |
| Primary Actor(s): | Donor |
| Risk Level: | High |
| Description: | This use case describes how donors enter a recurring donation amount. |
| Precondition: | User accesses the HoofKY website. |
| Postcondition: | User successfully sets up their recurring donation. |
| Flow of Events: | 1.) User clicks “Donate Now” button  2.) User clicks “Log In With PayPal” button  3.) User enters PayPal username  4.) User enters PayPal password  5.) User clicks “Log In”  6.) User verifies recurring transaction  7.) User is redirected back to HOOF website  8.) Use case ends |
| Explanation: | This use case is classified as high-risk because large portions of HOOF’s operating budget, and if they were unable to accept donations they would lose significant funding. |

Use Case: Modify Recurring Donation

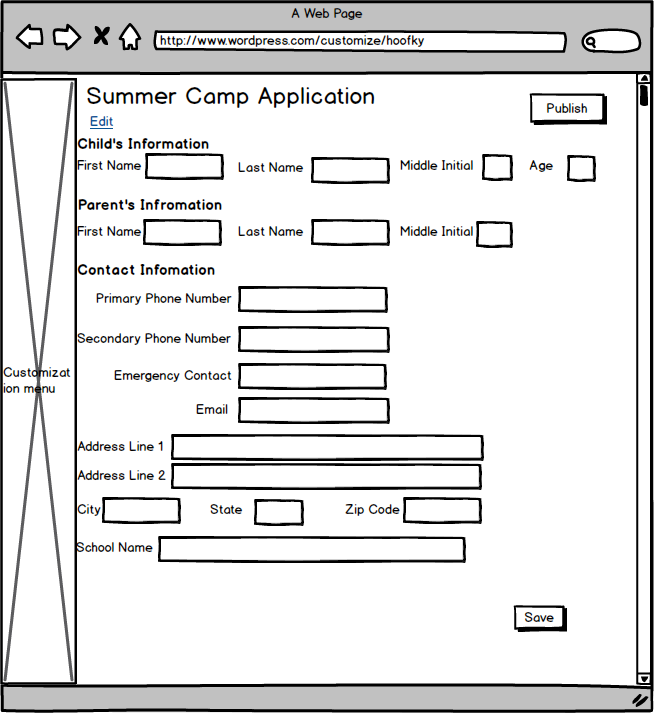
|  |  |
| --- | --- |
| ID: | 2.C |
| Use Case Name: | Modify Recurring Donation |
| Primary Actor(s): | Donor |
| Risk Level: | High |
| Description: | This use case describes how donors can change their recurring donation amount. |
| Precondition: | User accesses the HoofKY website. |
| Postcondition: | User successfully modifies their recurring donation. |
| Flow of Events: | 1.) User clicks “Donate Now” button  2.) User clicks “Log In With PayPal” button  3.) User enters PayPal username  4.) User enters PayPal password  5.) User clicks “Log In”  6.) User clicks “Profile”  7.) User clicks “My Money”  8.) User clicks “Update”  9.) User selects recurring payment  10.) User clicks “Change Amount”  11.) User inputs donation amount  12.) User clicks “Submit”  13.) User is redirected to HOOF website  14.) Use case ends |
| Explanation: | This use case is classified as high-risk as donors need to be able to modify their donations as they see fit, and failing to allow this behavior could result in total withdrawal of their support. |

Use Case: Delete Recurring Donation

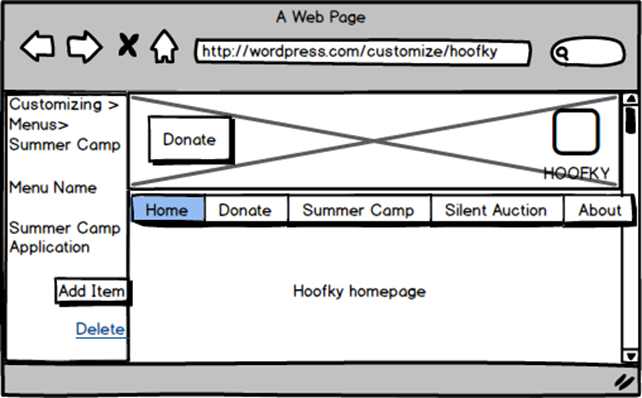
|  |  |
| --- | --- |
| ID: | 2.D |
| Use Case Name: | Delete Recurring Donation |
| Primary Actor(s): | Donation |
| Risk Level: | High |
| Description: | This use case describes how donors can delete their recurring donation amount. |
| Precondition: | User accesses the HoofKY website. |
| Postcondition: | User successfully deletes their recurring donation. |
| Flow of Events: | 1.) User clicks “Donate Now” button  2.) User clicks “Log In With PayPal” button  3.) User enters PayPal username  4.) User enters PayPal password  5.) User clicks “Log In”  6.) User clicks “Profile”  7.) User clicks “My Money”  8.) User clicks “Update”  9.) User selects recurring payment  10.) User clicks “Cancel”  11.) User is redirected to HOOF website  12.) Use case ends |
| Explanation: | This use case is high-risk for reasons similar to use case 2.C; users need to be able to modify or delete their recurring donations as they see fit. |

# Prototypes

# Add & Modify Summer Camp Application



# Delete Summer Camp Application



# Donation Form

